

Phrases of the Kinetic:  
Dynamic Physicality as a Dimension of the Design Process

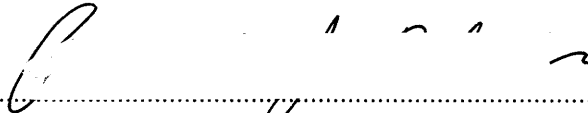
Amanda Jane Parkes

Submitted to the Program in Media Arts and Sciences,  
School of Architecture and Planning,  
in partial fulfillment of the requirements for the degree of  
Doctor of Philosophy in Media Arts and Sciences  
at the Massachusetts Institute of Technology

[JUNE]  
May 2009


© 2009 Massachusetts Institute of Technology. All Rights Reserved.

Author



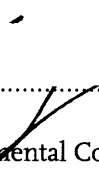
Amanda Jane Parkes  
Program in Media Arts and Sciences

Certified by

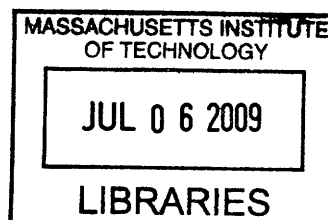


Hiroshi Ishii  
Muriel R. Cooper Professor of Media Arts and Sciences  
Massachusetts Institute of Technology  
Thesis Supervisor

Accepted by



Deb Roy  
Chair, Departmental Committee on Graduate Students  
Program in Media Arts and Sciences



ARCHIVES



# Phrases of the Kinetic: Dynamic Physicality as a Dimension of the Design Process

Amanda Jane Parkes

Submitted to the Program in Media Arts and Sciences,  
School of Architecture and Planning on May 19, 2009  
in partial fulfillment of the requirements for the degree of  
Doctor of Philosophy in Media Arts and Sciences  
at the Massachusetts Institute of Technology

## **Abstract**

At its core, the concept of Tangible Interfaces leverages the idea of using the movement of the body as an inherent part of the human side of a human-computer interaction, assuming that bodily engagement and tactile manipulation can facilitate deeper understanding and more intuitive experiences. However, as an interaction principle in our era of digital design, motion construction and control has been underutilized and little examined as a design tool, leaving open the possibilities of motion's natural ability to draw our attention, provide physical feedback, and convey information through physical change. This dissertation postulates that the ability to experiment, prototype, and model with programmable kinetic forms is becoming increasingly important as digital technology becomes more readily embedded in our objects and environments and need for tools and systems with which to create, manipulate and finesse motion in response to computational and material input remains an under-developed design area. This thesis aims to establish principles of kinetic design through the exploration of two approaches to motion construction and manipulation: motion prototyping as a methodology for design thinking, learning and communication and physically dynamic state memory as a methodology for organic form finding and transformation in the design process. To demonstrate these aims, I present three interface systems: Topobo, a system for motion construction and dynamics physics education with children; Kinetic Sketchup, a system for motion construction and prototyping in architecture and product design; and Bosu, an augmented textile interface offering an experimental approach to digitally augmented organic form finding in fashion and product design.

Thesis Supervisor:  
Professor Hiroshi Ishii  
Muriel R. Cooper Professor of Media Arts and Sciences  
Massachusetts Institute of Technology





Phrases of the Kinetic:  
Dynamic Physicality as a Dimension of the Design Process

Amanda Jane Parkes

Doctoral Dissertation Committee

---

Thesis Supervisor

Hiroshi Ishii  
Muriel R. Cooper Professor of Media Arts and Sciences  
Massachusetts Institute of Technology

---

Thesis Reader

Cynthia Breazeal  
LG Group Career Development Professor  
Associate Professor of Media Arts and Sciences  
Massachusetts Institute of Technology

---

Thesis Reader

Chuck Hoberman  
Founder and President  
Hoberman Transformable Design  
New York, NY



## Contents

Acknowledgements	15
1. Introduction	19
Aim	21
Approach	22
Contributions	22
Thesis Overview	23
2. Theoretical Framework	25
Radical Atoms - the Transformation of Tangible Bits	25
Motion and Emotion	26
Design Language of the Body	28
Transformability and the Future of Interaction Design	29
3. Background and Related Work	31
Kinetic Precedents - Anthropomorphism and Abstraction	31
Kinetic Art & Early Automata	31
Abstracted Motion in Robotic Interaction	35
Actuation in Interaction Design	38
Actuation as dynamic data representation	38
Actuation as embodiment of data	40
Actuation as embodiment of gesture	41
Actuation as form generation	41
State of the Art in Transformable Design	43
4. Motion Prototyping & Dynamic Form Generation	45
Motion Variables	47
Design Parameters	47
Form and Materiality	49
Modularity and Granularity	49
Kinetic Memory and Temporality	49
Repeatability and Exactness	50
Intelligence and Reactivity	50
Emergence	51



5. Topobo: A 3D Constructive Assembly with Kinetic Memory	53
Early Design Studies	55
Passives	57
Actives	59
Queens	61
Technical Specifications	62
Color	63
Initial Evaluation	63
Domains of Knowledge	65
Topobo Backpacks	67
Backpacks Evaluations	70
6. Longitudinal Outreach with Topobo	75
Selected Case Studies	77
After School Enrichment Program	78
Elementary/Middle School Science Classroom	81
After School Robotics Center	84
Urban Science Museum	86
Overall Findings	89
Exhibitions, Courses and Workshops	92
7. Kinetic Sketchup: Evolving Topobo into New Design	95
Realms	
An Architecture Case Study	96
Motion Design Parameters	102
Modes of Operation	106
Module Design	107
Technical Design	111
8. Organicism with a New Actuated Materiality: Glume, Senspectra and Bosu	113
Glume	115
Senspectra	118
Bosu	123
Modes of Operation	125
Module Design	127
Surface for Organic From Finding	129
9. Workshop Evaluation of Kinetic Sketchup and Bosu	133
Overall Findings	136
Inspirational vs. Construction Tools	137
Temporal Design	138



Materiality and Organicism	139
Intuitive Technology	139
Issues and Limitations	140
Project Case Studies	141
Threshold of Controlability	144
10. Looking Forward : Materiality and the Emerging Design	147
Process of Motion Authoring	
The Evolving Tangible Interaction Loop	147
Animation with Verbs and Adverbs	149
Material as Adverb	152
Motion Material Morphology	154
11. Conclusion	157
References	161
Appendix A: Hardware & Software Documentation	169
Topobo & Kinetic Sketchup PCB Schematic	169
Topobo & Kinetic Sketchup Firmware	172
Bosu PCB Schematic & Layout	201
Bosu Firmware	203





*Just as one can compose colors or forms, so one can compose motions.*  
*- Alexander Calder*



## Acknowledgements

My committee members- Hiroshi Ishii, a hero among men, unconditionally supportive and ever provocative, thank you for asking the hard questions and forcing me to push harder for answers. Cynthia Breazeal, kind and thoughtful, calm and perceptive, the lab is lucky to have you and your fascinating perspective. Chuck Hoberman, a visionary with a practical bent, your work has shown that you can make the impossible happen, and work made ever better by a person of kindness and humor behind such great vision

The MIT and broader interaction design communities have offered many for keen guidance of my time here -most especially John Maeda, sorely missed for aesthetic inspiration, humor therapy and eccentric spirit. Allen Sayegh, Kostas Terzidis, Chris Csikszentmihalyi, John Ossendorf, Ivan Poupyrev, Joachim Sauter, Bahktiar Mikhak & Mitchel Resnick. Biggest of thank yous to Linda Peterson who got me through this process with sanity and compassion in the midst of bureaucracy

The Tangible Media Group- most especially Hayes Raffle whose collaboration has been a joy and whose hard work and experience has made so much of my work at MIT possible. Leo Bonanni, the best of friends, the most fruitful of designers, I look forward to a lifetime of collaboration. James Patten for his wise tact, sane advice and true friendship, Vincent LeClerc and Adam Kumpf as the coolest techies around. And all the rest throughout the years- Jamie, Kimiko, Cati, Keywon, Gian, Monzy, Kimiko, Angela, Jason, JB... And a huge thank you to Lisa Lieberson for keeping us a well-oiled machine.

My extended research team - John Pugh, a Future Crafter turned friend who, in late IAP nights, got Kinetic Sketchup on its way. Josh Lifton for his monumental restructuring of code, and super UROPs Andy Goessling, Julie Henion, Judy Ho, Adrian Adames, Brian Mazzeo, Andy Lieserson, Laura Yip, Nick Williams, Jeremy Schwartz, Elysa Wan.

Professional educators & researchers in evaluations and Motorola, Lego, iCampus + Microsoft for financial support



The amazing women whose friendships span decades and continents, thank you for reminding me in the male fest that is MIT how lucky I am to have such a collection of strong independent beautiful women in my life, far and near - Carmen, Hilary, Nicole, Becky, Mannes, Cow, Maggles, Pip & Saoirse.

My best of friends from MIT and all over: Andrew, Zac, Nick, David, Daniel, Kat, Ananda, Josh, Carlton, Dean, Julian, Evan, Sam, Joe, Justin, Amber, Kelly, Alex, Alyssa, Ayah, Daisy, Olli, Brian, Ivan, Jussi, Richard, Connor, Matt, Talia, Ben, Burak, Tak, Shawdee, Dan, Grant, Dave, Marcelo, Marika, Ted, Ricardo, Marcelo, Jeevan, Dave, Jackie, Justin, Jroth, Eric, Mat, Lisa, Di, Shana, Kathleen, Jess, Valerie, Ryan, Dietmar, Jeana, Barbara, Angela Ted, Georgia, Julian

Steen, who has no idea how much influence and inspiration he's given over the years, and Tim, who has taught me that 6000 miles away is right next door in my mind and heart. and last but not least, Eddie O. my companion for so much of this bumpy ride, your support and love kept me afloat.

my family – Mom, for her unfailing and unconditional love and support of all my crazy notions in my quest to live an unconventional life & Ron (for keeping her sane in spite of me), Dre ,for general older sisterly wisdom, & Ryan, her rock, and their newest little Rebecca Quinn, very proud to be your aunt!

This thesis is dedicated to my dad who I know would be delightfully astounded to see what I've been up to since those long ago days we spent building things in the garage...



## *Chapter 1*

### **Introduction**

At its core, the concept of Tangible User Interfaces [Ishii 97] leverages the idea of using the movement of the body as an inherent part of the human side of a human-computer interaction, assuming that bodily engagement and tactile manipulation can facilitate deeper understanding and more intuitive experiences. Utilizing movement is a natural mapping for interaction, reflecting the fact that human beings possess a deeply rooted response to motion, recognizing innately in it a quality of 'being alive.' However, as an interaction principle in our era of digital design, motion construction and control has been underutilized and little examined as a design tool, leaving open the possibilities of motion's natural ability to draw our attention, provide physical feedback, and convey information through physical change. It becomes apparent that our relationship with movement, and its transformative properties needs to be reconsidered and revalued.

Tangible Media is in a state of transition, entering an era where tangible bits are transformed into 'radical atoms', bringing light to a new generation of media. The concept of radical atoms embodies the interaction ideals of Tangible Interfaces with a focus on direct manipulation and bodily engagement but expands on interaction principles to address

changes in physical properties (form, shape, color, stiffness etc) as elements controlled computationally in parallel with hands, body, and physical tools. Radical atoms are physically in sync with their digital/computational models employing bi-directional real-time communication in physically reversible processes. In essence, creating interaction where we think no longer of designing the interface, but thinking of the interface as a material itself.

As we transition to this post-tangible bits era, we are encountering a new range of design problems- how do we visualize, imagine, and design the physical processes of transformation? How do we 'prototype' the metamorphosis of such parallel physical computational interactions through time and space? While designers have numerous techniques and tools at their disposal to improve the interaction and appearance of objects, similar methods for creating ways to model transformation through space and time are lacking. The emerging field of Kinetic Interaction Design creates a foundation on which to guide designers through the physical process of transformation. To mediate this new field, the development of physical media which breaks down the perceptual barrier between tools and materials becomes necessary. These media can supply the complexity of computational capabilities embedded in physical materials with intuitive usability .

With the advent of algorithmic control in our digital systems, the ability to manipulate kinetic behavior and the notion of understanding how to design with sophistication into our kinetic structures becomes crucial. The field of contemporary robotics offers a rich vocabulary from which to draw - in exploring the psychological and functional dimensions of interacting with a self-actuated other and incorporating how human tendencies toward anthropomorphism shapes our perception of motion. As incorporating actuation in interactive systems becomes more commonplace, interaction design can learn from the tenets and examples set by roboticists, additionally leveraging and abstracting from concepts such as fluency, modularity and decentralization.



In the creation of actuated structures, products, and interfaces, designers can take advantage of both the perceptual qualities which motion can offer, creating emotional triggers through abstracted anthropomorphism, as well as the functional qualities with motion as an enabler of transformability. Designers must consider the combined effects of functional and emotional qualities when incorporating motion into the design of objects which interact with the human body, in light of its shifting contemporary identity. The idea that the human body is a fixed standard in design, or that the human body is at the point of departure for everything we make, is part of a recent more conceptual approach to cultural production [Holt 05]. By paying more attention to the body, what it implies by way of its contours and structure not to mention what its physical needs and transformative abilities potentially signify- designers are arriving at a brave new world of fluid and corporeal solutions. Potentially this is world where products and devices respond dynamically, where form dynamically follows the flow of activities of the human body. Instead of form following function, form equals function, paving the way for the a new branch of Organic User interfaces [Vertegaal 07] where objects, no matter how complex, dynamic or flexible its structure, may embody dynamic information.

### **Aims**

This dissertation postulates that the ability to experiment, prototype, and model with programmable kinetic forms is becoming increasingly important as digital technology becomes more readily embedded in our objects and environments and need for tools and systems with which to create, manipulate and finesse motion in response to computational and material input remains an under-developed design area. I aim to create a working vocabulary for kinetic design examining the relationship between functional kinetic behavior and emotional implications of motion in interaction design. This thesis aims to establish principles of kinetic design through the exploration of two approaches to motion construction and manipulation:

- motion prototyping as a methodology for design thinking and communication
- physically dynamic state memory as a methodology for organic form finding in the design process

### **Approach**

This thesis proposes a series of research implementations which embody specific functional uses of programmable kinetic behavior in interface design, ranging from motion prototyping tools as a methodology for design thinking and communication to physically dynamic state change as a methodology for organic form finding:

**Topobo + Backpacks**- system for motion construction and prototyping with children

**Kinetic Sketchup**- system for motion construction and prototyping in architecture and product design

**Bosu** - experimental approach to digitally augmented organic form finding combined with kinetic memory in soft materials for fashion, product, or interaction design

### **Contributions**

This thesis offers the following contributions:

A new approach to kinetic interaction design which creates a stato-dynamic continuum from motion prototyping tools to dynamic form finding, blurring the line between the tools of motion construction and the materials which embody actuation

The design, technical development, and construction of physical design tools for kinetic prototyping and organic transformation and their evaluation when appropriated as materials in varied design environments

Guiding parameters for kinetic design, incorporating the emotional and functional implications abstracted from motion, and ideas for applying them to kinetic products in the future

## **Thesis Overview**

The following chapters describe the evolution of ideas in this thesis:

Chp 2: provides a theoretical framework of contemporary ideas upon which this thesis is founded

Chp 3: describes related work from art, robotics, human-computer interaction and architecture

Chp 4: discusses the parameters of motion design

Chp 5: describes the development of the Topobo system, from initial design through manufacturing

Chp 6: describes the and extended outreach evaluation

Chp 7: describes the development of Kinetic Sketchup, expanding Topobo into new design realms

Chp 8: describes the development of systems for new materiality in form finding and motion design - Glume, Senspectra and Bosu

Chp 9: discusses the evaluation of Kinetic Sketchup and Bosu as put to use in a design workshop

Chp 10: offers a perspective on the future of materiality in motion authoring and what can be learned from existing digital systems in procedural animation and motion interpolation

Chp II: concludes the thesis with a summary and perspective on the technology of future kinetic materials and systems

Much of the work of this thesis is the result of collaborated efforts with colleagues at the Media Lab and the use of the term 'we' throughout this thesis refers to joint ideas and

development. Hayes Raffle has been a long term collaborator on the Topobo project on its development, evaluation, and exhibitions. Josh Lifton did a thorough overall of the Topobo and Backpacks AVR code. Vincent LeClerc collaborated on the early projects looking at organic form-finding- Glume and Senspectra. For Kinetic Sketchup and Bosu, several MIT undergrads, Andy Goessling, Julie Henion, and Judy Ho, have collaborated on development and graduate architecture student John Pugh helped develop the Kinetic Sketchup modules and in running the evaluation workshop.

## *Chapter 2*

### **Theoretical Framework**

#### **Radical Atoms - the Transformation of Tangible Bits**

For the past ten years, the Tangible Media Group at the MIT Media Lab has been pursuing a research vision called “Tangible Bits” [Ishii 97] with the goal of extending access to computation beyond a traditional graphical user interface (GUI), consisting of a keyboard, mouse and screen, to interfaces that use physical objects as tangible embodiments of digital information. The concept of “Tangible User Interfaces” seeks to take advantage of people’s existing skills and bodily knowledge for interacting with the physical world and provide interfaces which allow users direct control of computation through the manipulation of physical objects. Transparency to the intangible world of digital information thus becomes possible through familiar tools and actions of the hand and body.

The development of Tangible Interfaces builds on the premise that hands have long been considered an important player in the human creative, thinking, and learning process, as Malcolm McCullough comments, “Human beings like to make things: they like to use their hands at least as much as their brains;...If we are to tap the increasing visuality and dynamics of computing in order to open new realms of abstraction, we should depend very much on these human

traits to do so“ [McCullough 96]. The relationship of hands and computation has been more formally addressed by the concept of Direct Manipulation, first coined in 1983 by software engineer Ben Shneiderman and later researched in depth by George Fitzmaurice. He argues that “improving the ‘directness’ and ‘manipulability’ of the interface can be achieved by improving the input mechanisms for graphical user interfaces” [Fitzmaurice 95]. The principles of Direct Manipulation have greatly informed the development of Tangible User Interfaces as they work to achieve such manipulability by integrating the input and output spaces, implying a physical manipulation of the digital data itself.

Advances in material science, actuation, nanotechnology are opening Tangible Media to new possibilities, with the power to make atoms dance with computational control, programming matter in the same way pixels can be programmed. Tangible Bits can now be embodied as ‘Radical Atoms,’ transforming physically in 3D space. This new genre of interface design builds on the principles of direct manipulation, engaging the hands and body fluidly and intuitively while allowing the interface to play a more physically active role as a participant in the interaction conversation. This vision sees the future of human-computer interaction where any object, no matter how complex, dynamic or flexible its structure, can display, embody and respond to information, and a design paradigm where we think no longer of designing the interface, but thinking of the interface as a material itself.

### **Motion & Emotion**

As we transition to a realm of ‘Radical Atoms,’ it is important to consider how objects that are physically transformative affect our perception of their identity and objects in motion. Turkle describes objects in motion as a particular kind of ‘computational object,’ in the category of a ‘relational artifact, which allows people ‘closeness to the object’ to imagine themselves ‘inside the system.’ [Turkle 2005]. In 2007, anthropologist Jodi Forlizzi conducted a study on the social nature of robotic products in the home. In her study, different families were given either a regular vacuum cleaner

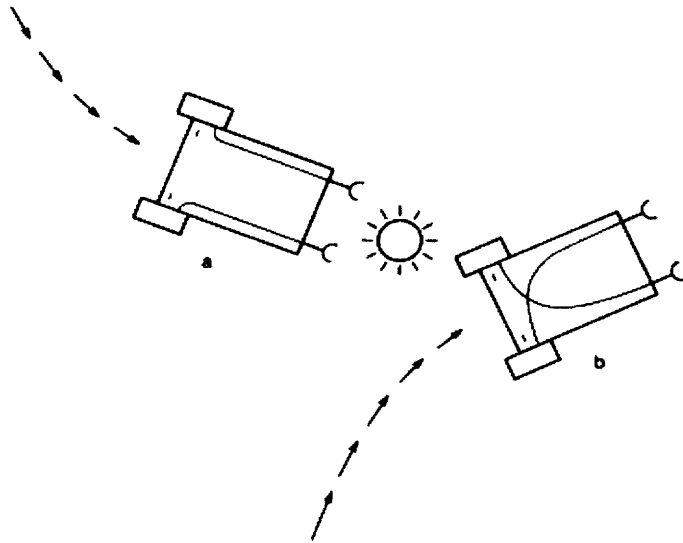


Fig. 2-1 Braitenberg Vehicles with 2 sensors and 2 motors each

or a Roomba, a robotic vacuum cleaner, and compared the families' cleaning patterns as evidenced by their social relationship with the vacuum. Families with the Roomba, noted how their cleaning patterns changed to be collaborative with the Roomba, and cited emotional responses to its behavior, including naming and attributing personality traits to it. Studies such as this highlight the power of kinetic objects in our environments and lend evidence to how people will adopt future robotic systems and how that might affect their design and function.

In the book *Vehicles: Experiments in Synthetic Psychology*, Valentino Braitenberg [Braitenberg 84] considers the conceptual significance of objects in motion from both the perspective of external perception and internal invention. He describes a series of thought experiments in which "vehicles" with simple internal mechanical structures behave in unexpectedly complex ways. Simple motion control mechanisms that generate behaviors that, if we did not already know the principles behind the vehicles' operation, we might call aggression, love, foresight and even optimism [fig. 2-1]. Braitenberg gives this as evidence for the "law of uphill

analysis and downhill invention,” meaning that it is much more difficult to try to guess internal structure just from the observation of behavior than it is to create the structure that gives the behavior. This idea exemplifies an untapped opportunity for designers - to utilize and exploit the emotional effects of motion while allowing the underlying mechanisms to remain hidden. The challenge lies in the deconstruction of this craft for the designer, to abstract perceptual qualities into a vocabulary of design elements, constructed from an expanding array of new technologies to employ for behavioral transformability.

### **Design Language of the Body**

Human-centered design has emerged as one of the tenets of contemporary design, and decades of studies in ergonomics have taught designers to revere the form and abilities of the body as the standard for analysis in interaction [Tilley 02]. Yet our notion of the body is changing. New technologies are allowing our bodies to become enhanced, augmented, expanded in functionality and altered in form. Ubiquitous & embedded technologies are allowing our devices to become more and more a part of us with increasing mobility and pervasiveness. In the space of digital device design, the line between body, clothing and object is blurring. The changing concept of the body, and our associated identities, alters how and what we strive to design for ourselves and the nature of digital products made to be worn and used by the body. With the aid of new digital tools, designers are responding new human body based form languages. Not simply mimetic in the static form and curves of the body but truly mimetic to the nature of the body as a living changing entity. The body's layered construction is able to be transformed because of its unique material properties-- a rigid internal skeleton for structure combined with muscles for movement and skin as a soft malleable covering. It remains up to the designer to utilize new technologies which breath life into computationally enabled material forms.

In the design of interactive objects, the layered multifunctionalism of many devices is removing the ability



of a designer to prescribe a particular ergonomic form appropriate for the device's usage. A device that functions as a camera, a phone and a PDA offers three unique relationships to the body in terms of physical usability. In many cases such a device takes the lowest common denominator form, a rectangular box. Designers must question the usability cost of multifunctionality, as suggested by Buxton [Buxton 01], and apply new technological ideas, in areas such as context awareness or physical transformability, which offer solutions to aid ease of use, engaging technology in the spirit of universal design. We stand in a state of design paradox - where digital tools allow us greater form generation abilities, but multifunctionalism removes an ability to create an appropriate ergonomic prescription. Our forms must respond through physical adaptability, where instead of form following function, form now equals function [Vertegaal 07].

### **Transformability and the Future of Interaction Design**

It is well known that together form and function inform the product design process, but with digital products, and increasing levels of interactivity and information embedded in such products, digital interaction must also contribute intimately to early stages in the design process. The interconnectedness of form, function, and behavior should

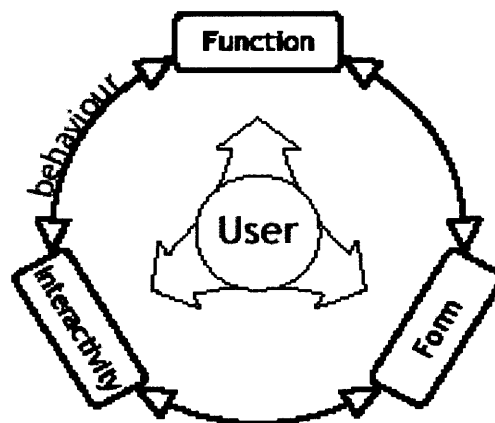


Fig. 2-2 Diagram relating form, function and interactivity by Frens

determine the development of any computational object [fig. 2-2]. In their paper, "Form, Interaction and Function, An Exploratorium for Interactive Products," Frens and Djajadiningrat discuss the need for a mechanism which can incorporate digital-mechanical behavior, through sensing and actuation, into the prototyping process, "Programming the behavior of interactive models goes beyond mere definition of the fact that parts of the model move. It is also about how and when they move. It is about the 'feel' of the interaction" [Frens 03].

Notions of transformable design and perceptual response to motion also relate to a larger changing practice of design in our post-industrial age. The responsibility for the relationship between industry and culture falls, in the modern world, on the shoulders of design. The product is the mediator between manufacture and the consumer, and its design is the container of the message that is mediated." [Sparke 87] The industrial age allowed for mass-production, placing value on replication and multiplicity of the same. In a post-industrial age we can attribute value back to accomplishing uniqueness, difference, individuality, personality, in our production, utilizing the ease of malleability of digital control to be reflected in the variability in creation of our physical environment and objects. We can also design, not just for the life of the product in use, but for its entire lifecycle, from production, use, to disposal. It is the responsibility for the future of interaction design to lead the way toward this ideal.

## Chapter 3

### **Background and Related Work**

#### **Kinetic Precedents - Anthropomorphism and Abstraction**

##### ***Kinetic Art & Early Automata***

To consider motion in interaction design first requires establishing a context for how embodying motion in a machine, whether it be a robot, computer interface or appliance, is an indication of our desire to understand and reflect qualities which we see in ourselves. Human beings have a rich history of designing and utilizing kinetic forms and in considering interactions between man and machine, it is natural and intuitive to reflect on how closely the machine resembles the organic nature of our bodies, either as embodied in physical form or represented in the fluidity and quality of motion. It is our ability to differentiate between lifelike or mechanical qualities in abstracted motion which allows us to employ motion as a tool for communication and engagement, allowing inanimate objects to become embodied partners in our interactions. This capability has been employed and manipulated by a long history of works in art and robotics from which to draw inspiration and analysis of the possibilities for kinetic interaction design.

Going back several centuries, we look to historical robots and automata as a place to extrapolate on the origins of the

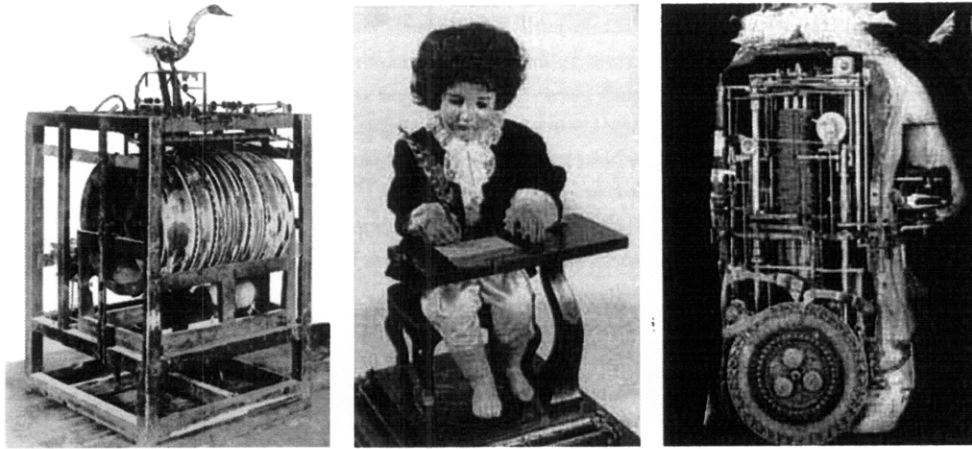


Fig. 3-1 Vaucanson, Mechanical Duck, 1738; Pierre Jacquet-Droz, The Writer 1774

human fascination with motion control. Although present from antiquity, the seventeenth century marked a significant increase in the phenomena of human or animal automatons, self-moving machines, who demonstrated the ideal to perfect the appearance and lifelikeness in a mechanical creature. One of the most famous of these was a mechanical duck by the Frenchman Vaucanson [fig. 3-1]. The duck was described as a marvel which 'drinks, eats, quacks, splashes about on the water, and digests his food like a living duck.' Another similarly spectacular automata most from this period was The Writer by Swiss Pierre Jacquet-Droz [fig. 3-1]. With internal clockwork mechanics, this life size figure of a boy could write any message up to 40 letters. Not utilitarian in nature, these reflect the early human fascination with the novelty of mimicking and embodying human characteristics in machines.

Of the many artistic movements emerging in the early 20th Century, Italian Futurism differentiated itself by attempting to find a modern artistic vocabulary for movement and speed, looking at motion as a 'concept.' While the Futurists did not create mechanical kinetic devices specifically, they were the first to investigate the concept of speed as a plastic value. Their works, while static, demonstrated an achievement of

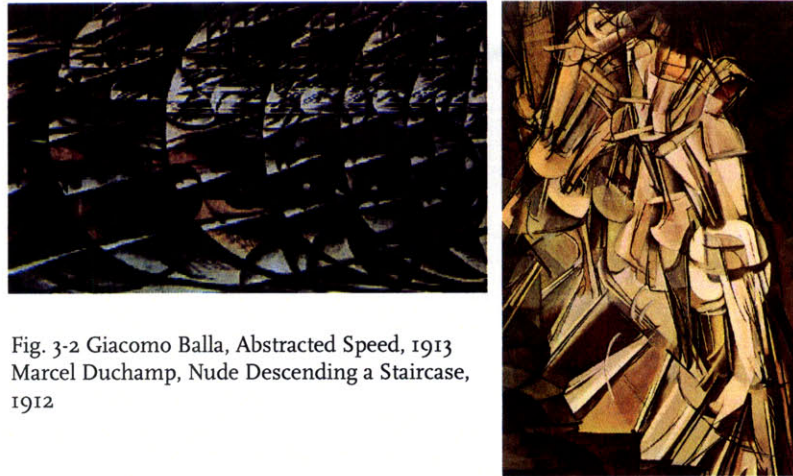


Fig. 3-2 Giacomo Balla, Abstracted Speed, 1913  
 Marcel Duchamp, Nude Descending a Staircase,  
 1912

dynamism which is both representational and conceptual, bringing a sense of human expression and emotion to studies of machine speed. This can be demonstrated by a comparison of Giacomo Balla's *Abstracted Speed*, a Futurist work from 1913 [fig. 3-2] and Duchamp's *Nude Descending a Staircase* from 1912 [fig. 3-2]. Both represent "an expression of time and space through the abstract presentation of movement" [Popper 68] and employ a similar aesthetic of simultaneous views in space, however Duchamp's work presents a human figure in motion while Balla's Futurist work reflects the motion of a machine.

The modern launch of what is considered 'kinetic art', or works which featured real movement in three dimensional space, began in the 1920's with a series of artists working on expressions through highly abstracted forms of motion. The *Light Space Modulator* (1923-1930) [fig. 3-3] by Laszlo Moholy-Nagy builds on traditional mechanical automata, but works with the concept as an abstraction. The sculpture consists of three transparent vertical planes with a central axis of rotating perforated planes through which light creates an environment of reflections and shadow. The *Light Prop* works as a surrogate actor in technological disguise [Krauss 81]. Like a human figure, the light prop has an internal structure that affects its outward appearance. It performs gestures- patterns of projected light that relate to its internal structure - that

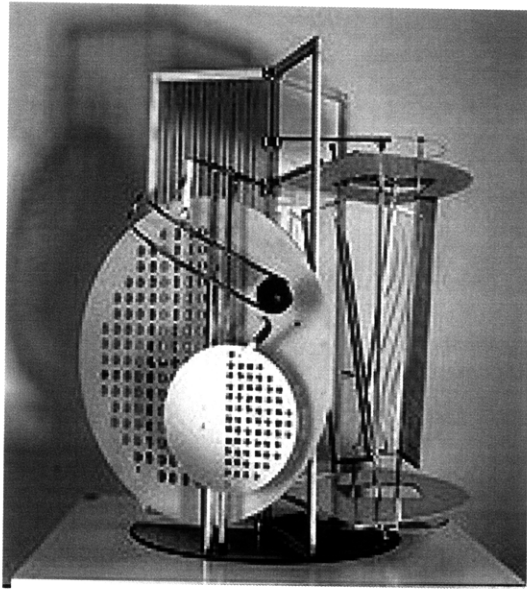


Fig. 3-3 Laszlo Moholy-Nagy, The Light Space Modulator, 1930

change in time and with complexity and it appears to have a quality of intentionality and volition, anthropomorphic in behavior if abstract in appearance.

In contradiction to Moholy-Nagy's technologically mechanized approach to creating motion, Alexander Calder, an American contemporary of Moholy-Nagy, instead utilized natural environmental factors as the source of motion for his sculptures. He is the inventor of the 'mobile' (a phrase coined by Duchamp), seemingly simple constructions of linear arms balanced in a delicate equilibrium capable of being disturbed and set in motion by the wind or air currents in a room, or by the gentle touch of a viewer. When set in motion the elements of a mobile, spin around their points of connection to conjure a sense of virtual volume for the viewer [fig. 3-4]. This generated sense of volume allows the mobile to be seen a metaphor for the body as it displaces space [Krauss 81]. The quality of motion, different in every occurrence, is mimetic of the body's actions, unmechanized and non-repetitive and brings a sense of organicism to the work of Calder.

Jean Tinguely, an artist working in the Dadaist tradition several decades later built 'machines' which combined

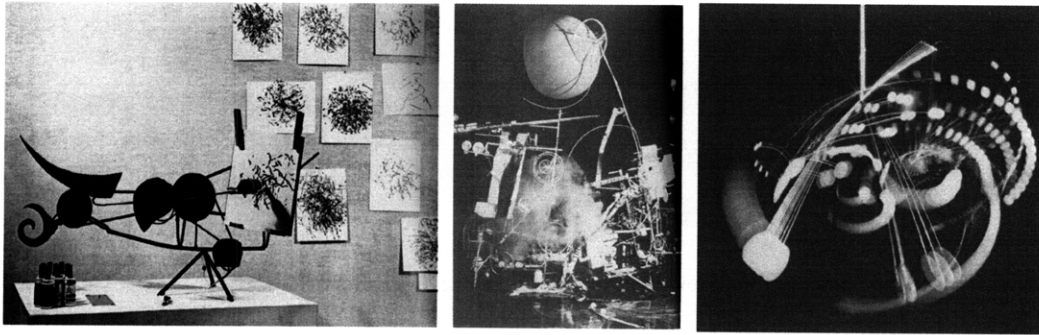


Fig. 3-4 Jean Tinguely, *Homage to New York*, 1950 & *Meta-matic No. 8*, 1958; Alexander Calder, *Mobles in Motions*

the sense of organicism seen in Calder's mobiles with the mechanically performative aspects of Moholy-Nagy's Light Space Modulator. He created a series of self-destructing machines, the most famous which performed in the garden at the Museum of Modern Art in 1960, and a meta-matic series of art-making machines [fig. 3-4]. His investigations into motion arose as part of his curiosity with the complex paradoxes of machines - they are constructed to be utilitarian but simultaneously useless when still; movement itself can demonstrate a kind of stability. The sense of change and movement in his creations obey only the laws of chance. The perception of organicism from Tinguely's creations derives from a sense of mechanical disorder which is in fact discretely ordered through what he terms, 'the functional use of chance' [Popper 68]. He viewed this behavior as very much a collaboration between man and machine, which only together have the ability to create an irrational product.

### ***Abstracted Motion in Robotic Interaction***

In contemporary times, the field of robotics and robotic art offers rich examples of motion design and control both in the functional and perceptual areas. The project *Keepon* [fig. 3-5], developed by Hideki Kozima and Marek Michalowski, was designed to perform emotive and attentive exchange with human interactants (especially children) in the simplest and most comprehensive way [Michalowski 07]. *Keepon* is a small creature-like robots with a soft yellow rubber skin,



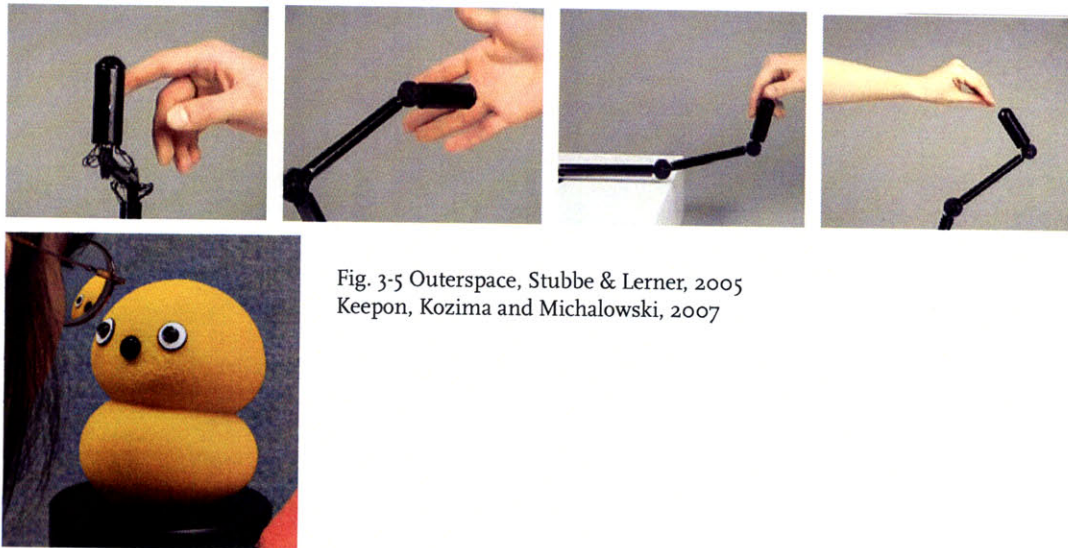


Fig. 3-5 Outerspace, Stubbe & Lerner, 2005  
 Keepon, Kozima and Michalowski, 2007

two cameras in its eyes, and a microphone in its nose. While Keepon possesses basic anthropomorphized characteristics in its form, it is its simple motion vocabulary which is used to communicate: attention is directed by turning and nodding while emotion is expressed by rocking side-to-side and bobbing. As part of the BeatBots project, Keepon has been used to study the underlying mechanisms of rhythmic social communication, focusing on establishing engagement, rapport, and comfort between people and robots that can interact by synchronizing familiar social rhythms.

Employing more abstracted formal qualities, the project OuterSpace [Lerner 05] [fig. 3-5] by Andre Stubbe and Markus Lerner presents a reactive robotic creature resembling an insect antennae, flexible enough to explore the environment. Outerspace appears as a playful, curious creature exploring the surrounding space looking for light, motion, and contact. As Outerspace engages with an observer, its motion patterns, based on body language and human gesture, change in response to stimulus and contact, engaging the observer in a social interaction. Although abstracted, Outerspace's organic motion repertoire allows the user to perceive a sense of intelligence in the creature, changing the nature of the interaction.





Fig. 3-6 The Table : Childhood,  
Andrea & Dean, 2001

Further investigation into the possibilities of perceived intelligence through isolated kinetic behavior are exhibited in the interactive installation *The Table: Childhood* [fig. 3-6] created by the team of roboticist Raffaelo D'Andrea and artist Max Dean. In the work, an ordinary wooden table has been imbued with reactionary behavior characteristics which could be perceived as childlike - playful, teasing, and unpredictable. The table attempts to create a relationship with the viewer, by using only its motion to engage and construct communication. The power of the work arises from the unexpected surprise of such behavior and plays off how easily we can recognize human characteristics, made even more powerful when contrasted with our expectations of the table as a neutral static object. The piece saliently demonstrates how motion designed to be mimetic of a living organism has the power to engage us and create an interactive conversation with an otherwise disembodied object.

Robotics work also provides inspiration on decentralized models of kinetic control and the possibilities of actuation to provide reconfigurability in interface design. Mark Yim's work in modular reconfigurable robotic systems such as PolyBot [fig. 3-7] [Yim 00] demonstrate a approach to building robots for various complex tasks based on one simple

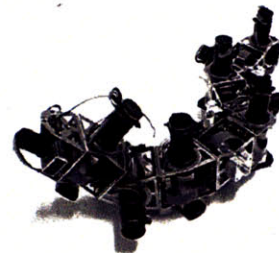


Fig. 3-7 PolyBot, Yim, 2000

repeated module. By connecting many modules together, the system can achieve a functionally complex task by adhering to a set of simple rules governing each module. PolyBot can be programmed to reconfigure itself, change its overall shape and functionality by moving its modules around, based on contextual sensing of different environments or tasks. This decentralized model of control can also lead to the phenomenon known as emergent behavior, where the global behavior of the system creates a perceived centralized intelligence. This allows a system of repeated elements with a simple motion vocabulary to appear more nuanced and naturalistic expanding its possibilities for interaction as a holistic entity.

### **Actuation in Interaction Design**

A growing number of projects in interface design have laid the groundwork for an emerging methodology for kinetic design. Within tangible interfaces, the coupling between the physical and the digital has usually been in one direction; we can change vital information through physical handles but the digital world has no effect on physical elements of an interface. The use of physical motion in a bidirectional relationship, follows as a natural direction relating strongly to the Tangible Interface philosophy, employing the benefits of malleability to the physical world.

### ***Actuation as dynamic data representation***

The first category of kinetic interfaces presents motion within an interface as dynamic data representation. In these systems, actuation is based upon computation feedback and the motion allows the system to become a partner in optimization for a particular condition. Elements in the system can be actuated to change position, speed, direction or quality of motion.

Two such examples of projects in this category are Pico (Physical Intervention in Computational Optimization) [fig. 3-8] [Patten 05] and its predecessor, the Actuated Workbench [fig. 3-8] [Pangaro 02] from the Tangible Media Group which both investigate planar actuation in a two

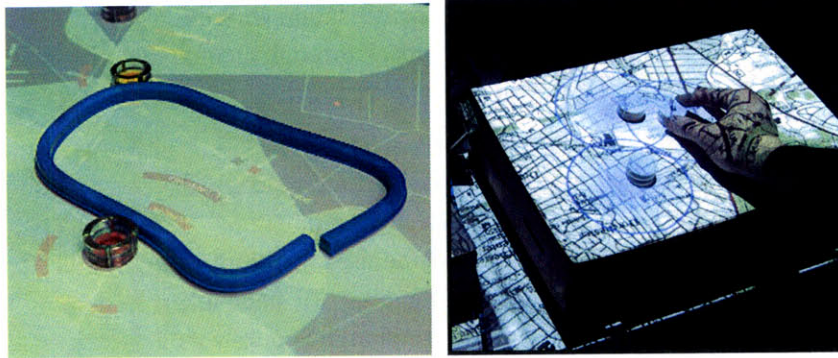


Fig. 3-8 Pico, Patten, 2006 & the Actuated Workbench, Pangaro and Maynes-Aminzade, 2002

dimensional surface. The Actuated Workbench uses an array of electromagnets embedded in a table to physically manipulate the input devices (pucks on a table top) to express computational output. The addition of actuation to an interface allows the computer to maintain consistency between the physical and digital states of data objects. Pico is also a tabletop interaction surface that can track and move small objects on top of it. Pico simultaneously represents and controls the high level structure of a software process with a mechanical process. The user can leverage his or her mechanical intuition about the way physical objects respond to forces and interact with each other to understand how common objects, such as a rubber band or coffee cup, might be used to constrain the underlying software process. The interface provides opportunities for improvisation by allowing the user to employ a rich variety of everyday physical objects as interface elements. The combination of these interactions, all governed by the friction and mass of the objects themselves directly affects the result of the task being performed. The Actuated Workbench and Pico have been used for complex spatial layout problems such as cellular telephone tower layout.

In both Pico and the Actuated Workbench, actuation controls physical elements of the system, the pucks, which are handles to data, representations of parameters in a simulation. The kinetic capabilities of the system allows an inanimate object



to more naturally become part of a conversation with the user, expressing computational feedback through change in position.

### ***Actuation as embodiment of data***

A second category of interface utilizes actuation in which the motion itself embodies data. In such interactions, the movement of the display allows the user to interpret information based on an abstract mapping. In static form, such an interface contains no information, it is purely its kinetic behavior which communicates with the user. An excellent example of motion embodying data exists in Mark Goulthorpe's Aegis Hyposurface [Aegis 03] [fig. 3-9]. This faceted metallic surface has potential to deform physically in response to electronic stimuli from the environment (movement, sound, light, presence etc) or to be controlled by a centralized mathematical function. The system is driven by a bed of pneumatic pistons while the dynamic 'terrains' are generated as real-time calculations.

Another such example comes in the form of the project Pinwheels [Ishii 01] [fig 3-9], an ambient display communicating digital information at the periphery of human perception. In this project, a stream of data, such as stock market activity monitoring, is mapped to the motion of a set of pinwheels, speeding up clockwise if the markets are increasing, for example. Just like the Hyposurface, as static objects, the Pinwheels exist purely as ordinary non-computational objects, it is their motion--its speed and direction-- which allows them to become communication devices.



Fig. 3-9 Aegis Hyposurface, Goulthorpe, 2001 Pinwheels, Ishii, 2001

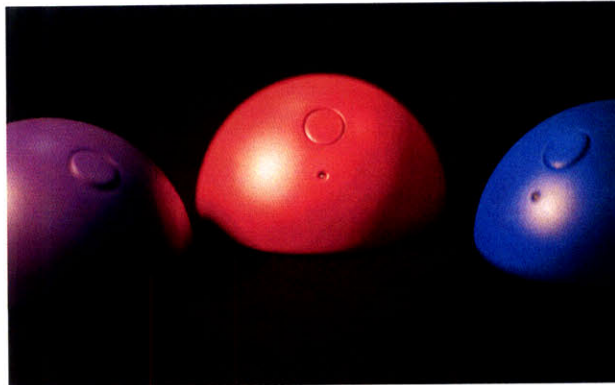


Fig. 3-10 curlybot, Frei, 2000

### ***Actuation as embodiment of gesture***

In a quest to express a sense of natural organicism, a class of kinetic interfaces is emerging which captures motion directly from the gestural language of the human body. curlybot [fig. 3-10] [Frei 00] is a toy that can record and playback physical motion as it is moved along a two dimensional surface.

As one plays with it, it remembers how it has been moved and can replay that movement with all the intricacies of the original gesture; every pause, acceleration, and even the shaking in the user's hand, is recorded. curlybot then repeats that gesture indefinitely creating beautiful and expressive patterns.

### ***Actuation as Form Generation***

Perhaps the most complex and growing category of kinetic interfaces centers around the concept of actuation enabling dynamically changing form. Also known as shape change or shape-shifting displays, these interfaces can be either continuously dynamic, embodying information when in motion, or stato-dynamic, functioning as a static form which moves between states. Projects such as Sandscape [fig. 3-11] [Piper 02] provides the physical affordances of real materials in sculpting and form generation but do not contain a mechanism for the computer to change the physical interface based on the data, the interaction loop is one-directional. Shape change interfaces are a logical progression to extend the capabilities of interfaces such as these, adding actuation to complement visual computational feedback. A combination

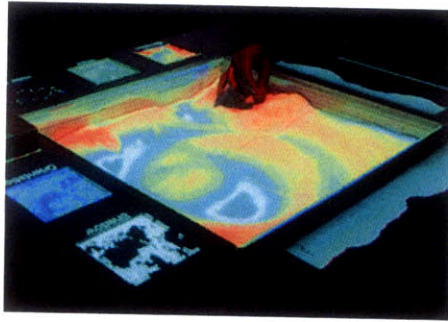


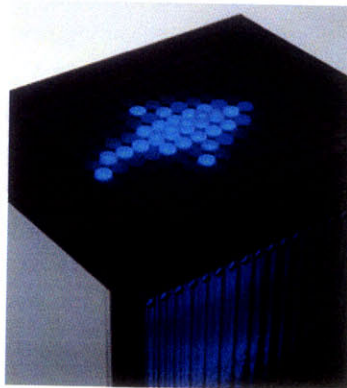
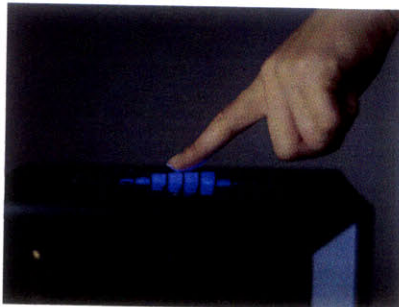
Fig. 3-11 Sandscape, Piper, 2002  
Feelix, Iwata, 2001



of actuation and three dimensional topology is investigated in the project Feelix [fig. 3-11] [Iwata 01] an interface device which provides haptic feedback to animated graphics through a deformable surface. An animated image is projected onto a flexible membrane (screen) and the Feelix system manipulates the membrane as an “actuated pinscreen” that can change the height of the membrane at discrete points on its surface to create relief-like forms. The user can then touch the image directly and feel its shape and rigidity.

The project Lumen [fig. 3-12] [Poupyrev 04] demonstrates a significantly more advanced demonstration of a shape shifting interface. Lumen is a 13 x 13 pixel bit-map display where each pixel can also physically move up and down. The resulting display can present both 2D graphic images and moving physical shapes that are controlled independently and

Fig. 3-12 Lumen, Poupyrev, 2004





can be observed as well as felt by the user, interactive display that presents visual images and physical, moving shapes, both controlled independently. The position sensors built into Lumen's surface allow users to input commands and manipulate shapes with their hands. The smooth, organic physical motions provide aesthetically pleasing, calm displays for ambient computing environments. Cutting edge shape change interfaces such as Lumen begin to explore the real possibilities for how physical transformability can embody the malleability so valued in the digital realm.

### **State of the Art in Transformable Design**

In addition to human-computer interfaces, we also look to the state of the art in transformable design as examples of the potential of dynamically changing form. Chuck Hoberman and Hoberman Associates specialize in transformable design, the development of products, structures, and environments that change their size and shape. Based on new technologies for adaptive building skins, their Emergent Surface [fig. 3-13] [Hoberman 08] is a wall that continuously reconfigures itself with portions selectively disappearing and reappearing. In one condition, the piece appears as a solid surface with three-dimensional curvature. In another, it resolves itself into seven slender poles, running floor to ceiling. And between these extremes lie an infinite variety of configurations. These different states represent the physical embodiment of digital information. As such, Emergent Surface represents a kind of

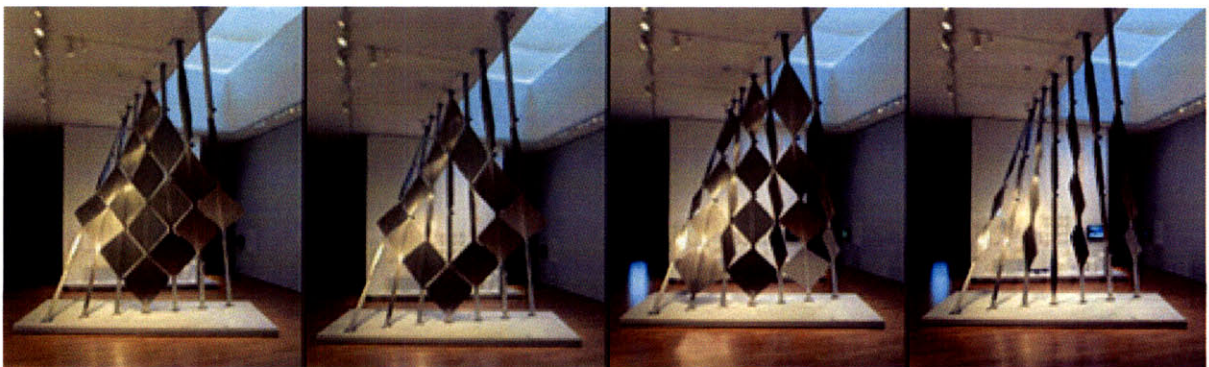


Fig. 3-13 Emergent Surface, Hoberman, 2008



Fig. 3-14 Robotic Dress, Chalayan, 2007

'material media', operating not on bandwidths of light and sound, but in terms of variable solidity and permeability. An example of the state of the art in transformable design for aesthetics exists in the robotic dresses [fig. 3-14] of fashion designer Hussein Chalayan. Chalayan created six mechanical dresses that transform from one era to another, using embedded technology and smart wires, one dress converting in the space of a minute from a high-necked corseted Victorian gown to a crystal-beaded flapper dress, propelling the dress through fashion history.



## *Chapter 4*

### **Motion Prototyping and Dynamic Form Generation**

These categories and examples of state-of-the-art kinetic interfaces have demonstrated a variety of methods to incorporate kinetic behavior as a valuable strategy in interface design. However, they have barely scratched the surface of the possibilities we see available in this relatively untapped arena. One of the biggest hurdles in advancing the development of kinetic interfaces and the products which emerge from them is the struggle to design, prototype and construct the actuated systems. Significant time, energy and commitment is necessary to determine if an interaction has the desired effect. Furthermore the 'language of kinetics'- how we as designers form the kinetic phrases, sentences, or dialogue of an interaction - is new and unknown territory. What is missing from our kinetic equivalent of tools and materials which allow for easy prototyping which is so pervasive in both the static physical world (sculpting) and the digital world (coding).

One possible inspiration exists turning back to the eighteenth century-- in 1772-1779, Swedish engineer Kristofer Polhem, created Letters from a Mechanical Alphabet [fig. 4-1], a series of small wooden objects describing mechanical elements for motion design. The alphabet consisted of 80 letters each demonstrating the simple movement that is contained in

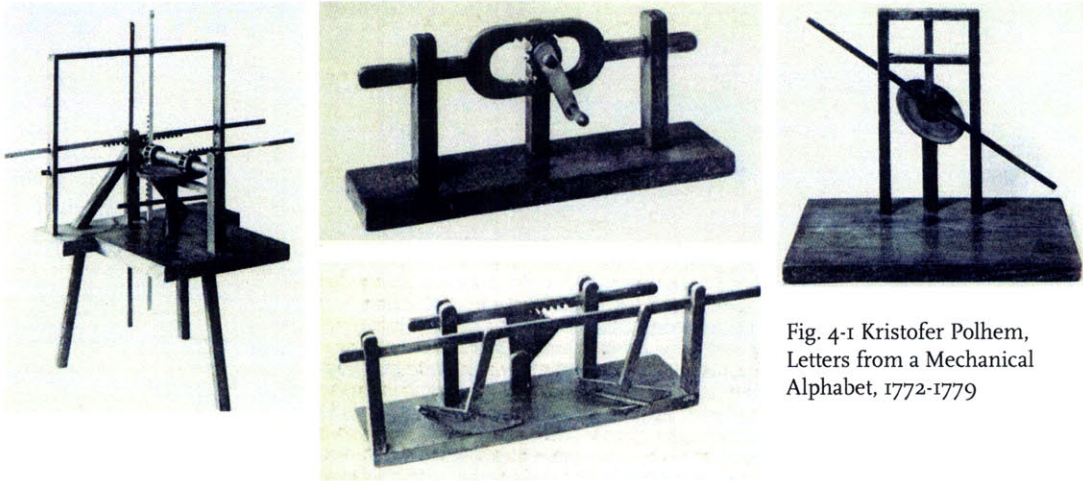


Fig. 4-1 Kristofer Polhem, Letters from a Mechanical Alphabet, 1772-1779

a machine - for example translating rotary movement into reciprocating movement. These objects serve to demonstrate a very direct relationship between form and mechanical motion relationship. However, if this principle of dissecting form and mechanics into an observable behavior was combined with our contemporary digital control structures and a new materiality, it becomes possible to imagine systems where a kinetic behavior could be designed both concretely and in the abstract, with programmability to satisfy functional needs while also manipulating emotional perceptual responses. This thesis explores the design of such systems, with computationally controlled kinetic behavior.

In the design of contemporary kinetic constructions - transformable products, interfaces and structures, it is important to classify the embodied purpose of the kinetic behavior, resulting in two categories of artifacts. Continuo-dynamic objects are objects which fulfill their intended functionality (whether practical or emotional or a combination) only when in motion, and stato-dynamic objects are classed as objects that transform from one state to another to change their functionality but can embody their intended purpose in a static state. I posit a temporal continuum between the two [fig. 4-2] as a new approach to kinetic interaction design where transitions of all different time scales (microseconds to decades) are considered as part of the design process.

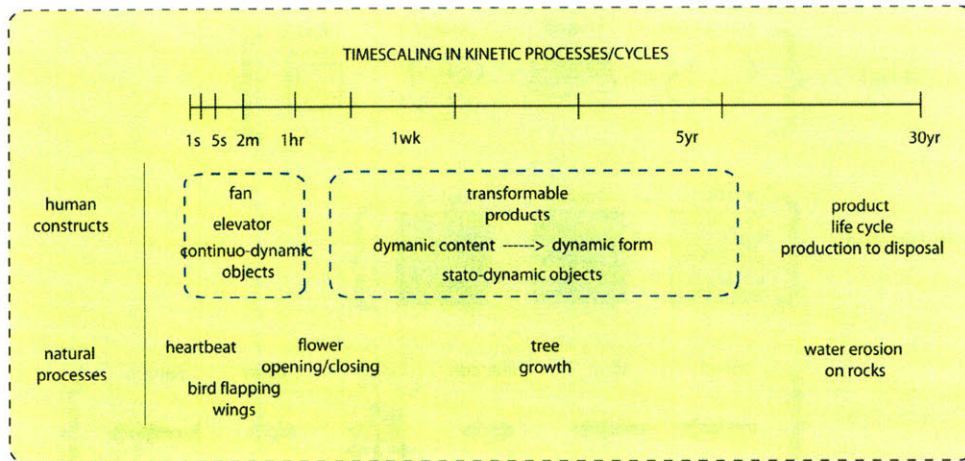


Fig. 4-2 Temporal continuum between continuo-dynamic and stato-dynamic objects

### Motion Variables

In interface design, motion can be delineated with physical components that are actuated in a way that can be detected by and respond to the user. In establishing a design vocabulary for motion, the variable parameters of how motion can be employed and controlled can be categorized in the following ways:

- Change in spatial arrangement of objects - change in position or orientation
- Change in quality of motion of objects - change in speed of rotation, speed of linear motion, or direction of motion
- Change in force applied to the user - change in pressure, amplitude, direction or torque
- Change in surface texture of objects - to be sensed visibly or through touch
- Change in formal characteristics or shape of objects affecting their perceived function and usage - transformability

### Design Parameters

To create a framework for the creation of motion, I present the [fig. 4-3] which I have developed to define the parameters of kinetic design through a set of motions variables. Every



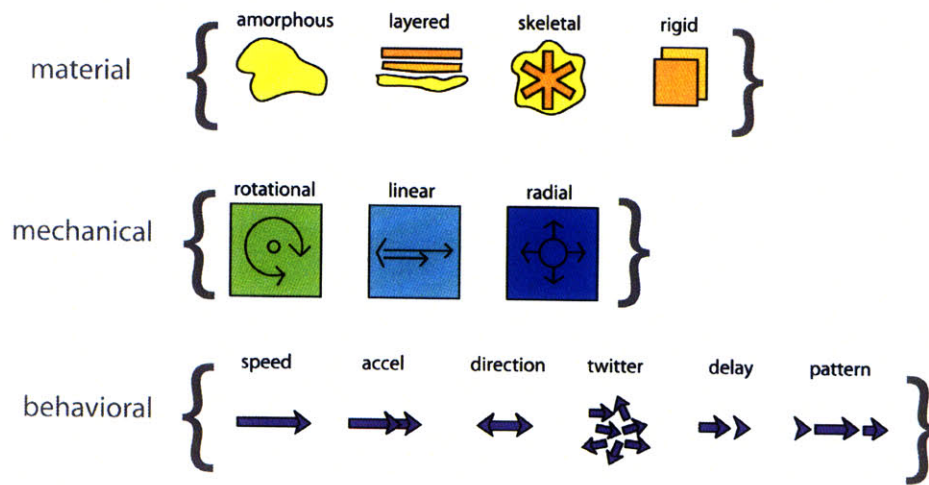


Fig. 4-3 Motion design parameters

motion construct can be broken down into a combination of three control categories:

- mechanical** - physical & spatial design of how the motion is created, based on the view of the user and observer of the system
- material** - physical qualities of the matter in which the motion is embedded, affecting the perceived nature of the motion
- behavioral** - temporal control structure of the motion

By defining for each of these categories, we can create a kinetic composition. We can also use these categories to dissect existing motion constructions, attempting to understand the relational connection between our interpretation of a motion and its qualities derived from its root elements. The interplay of the basic mechanical, material and behavioral elements allow us to create higher level design parameters which help designers to translate physical design choices into perceptual characteristics of a motion construction. I have set forward the following six design parameters to be considered in the design of any motion construction.

### ***Form and Materiality***

Motion itself is a concept, an act which reflects the process of change. In order for us to be able to recognize and comprehend motion, it must be embodied in a material form, as Hegel stated, “There is no matter without movement, any more than there is movement without matter” [Hayward 00]. A most crucial design parameter, and one often sidelined in discussion, is how materiality affects motion perception and control. A very significant perceptual shift can occur with a change in material - a jerky disjointed motion of a series of mechanical motors can be embedded in a soft padded exterior and the quality of motion can be inverted to a smooth oscillation. Materials and form play an important role in kinetic behavior when considering how objects are subject to the natural environmental forces of gravity and friction, or alternatively, the material affordances of an interface can greatly influence the user’s experience. The material characteristics can also become part of a system’s feedback loop if designed into the control structure.

### ***Modularity and Granularity***

Because of the nature of actuator technologies and the systems for digital control, it generally remains necessary for actuated elements to be controlled as discrete units, whether it be array of electromagnets, hydraulic pistons or servo motors. However, it is preferable in many situations for an interface to behave as a continuous surface, mimicking the fluidity and organicism of motion in the natural world. The compromise must be made to achieve a level of granularity which can fool the human senses to discern continuity while maintaining a system that is reliable and controllable as discrete elements. Potential solutions lie in interfacing actuator technologies and materials with appropriate mechanical properties for the desired effect. In the future, advances in smart materials and nanotechnology, allowing for molecular levels of control, will render this issue moot.

### ***Kinetic Memory and Temporality***

While computational control allows actuated systems to provide real-time physical feedback, it also offers the

capability to record, replay and manipulate kinetic data as if it were any other kind of computational data. For designers, this opens up vast potential for the functionality and usability of kinetic interfaces. Kinetic memory can allow a user to fast forward or slow down motion sequence, move it backward or forward in time. Objects could also have a shape change history, or an interface could playback a recorded sequence, exposing it to different material or environmental parameters or superimposing historical or temporally contextual motions onto the present.

### ***Repeatability & Exactness***

We can easily determine lifelike vs. mechanical motions based on exactly repeatable motion. In designing kinetic interfaces, consideration must be given to the desired level of abstracted anthropomorphism and plan the repetition of motion according. With most mechanical systems, repeatable exactness is the simplest control state, and in many behaviors it is easily identifiable. However, it is the perception of the behavior that is important, not necessarily the behavior itself - our perception of a system's exactness can be skewed easily by many factors including material embodiment and control structure. Because of the gestural recording as input for Topobo and curlybot, these interfaces are imbued with a degree of an organicism usually missing from direct digital actuator control.

### ***Intelligence & Reactivity***

Simple systems that perform a single operation repeatedly usually appear devoid of human-like qualities. If a system begins to react and respond to you, a level of anthropomorphism may be determined. However, if the complexity of a system's behavior becomes incomprehensible to an observer, the system will again revert to that of an 'other,' a system operating outside of human intelligence and behavior.

This phenomenon correlates to Masahiro Mori's concept of the 'uncanny valley' [Mori 70]. His hypothesis states that as a

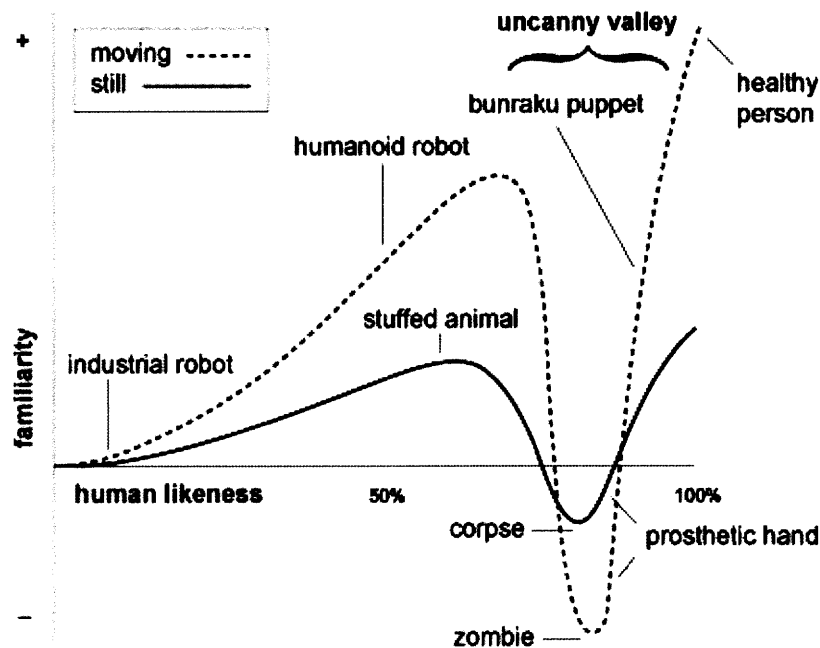


Fig. 4-4 Mori, Uncanny Valley, 1970

robot is made more humanlike in its appearance and motion, the emotional response from a human being to the robot will become increasingly positive and empathic, until a point is reached beyond which the response quickly becomes that of strong repulsion. However, as the appearance and motion continue to become less distinguishable from a human being, the emotional response becomes positive once more and approaches human-to-human empathy levels [fig. 4-4]. In the design of kinetic interfaces, this phenomenon must also be accounted for, whether to avoid the state of the uncanny or to take advantage of the discomfort created by a 'barely-human' other.

### **Emergence**

A certain paradox exists in the case of emergent behavior of kinetic systems. Emergence, defined as the process by which a set of simple rules determine complex pattern formation or behavior, creates systems which contain elements which are thoroughly comprehensible to understand individually

and whose behavior we relate to, (like ants in a an ant colony) while it is difficult to understand the overall behavior of the system functioning with decentralized control. Johnson describes the phenomenon, “The persistence of the whole over time – the global behavior that outlasts and of its component parts – is one of the defining characteristics of complex systems... Generations of ants come and go, and yet the colony itself matures, grows more stable, more organized. The mind naturally boggles at the mix of permanence and instability” [Johnson 01]. Designing for emergence allows kinetic systems to reflect a living state, going in and out of comprehension.



## Chapter 5

### **Topobo: A 3D Constructive Assembly with Kinetic Memory**

The development of Topobo began with a simple question, “What is it like to sculpt with motion?” Is there a way which designing and creating kinetic interactions could be as simple and direct as drawing with a pencil and paper or sculpting static objects in clay?

Topobo is a 3D constructive assembly with kinetic memory, like a set of motorized building blocks with the ability to record and playback physical motion in 3D space. By snapping together a combination of static and motorized components, people can quickly assemble dynamic biomorphic forms like animals and skeletons with Topobo,



Fig. 5-1 The Topobo System



Fig. 5-2 The Topobo System: an Active & Passives

animate those forms by pushing, pulling, and twisting them, and observe the system repeatedly play back those motions. Important and unique to Topobo is the systems' coincident input and output space, the kinetic recording occurs in the same physical space as it plays back. This provides users the a greater understanding of the interface creating a reflection of their own bodies, providing a mechanism to translate the qualities of organic bodily motion with that of mechanical control.

Topobo, named for topology, botanics and robotics, combines the physical qualities of a modular building block system with gestural recording capability producing a means for dynamic expression with the press of a button and the flick of a wrist. Topobo works like a material extension of the body, giving one's gestural fluency computation and memory, to create a three-dimensional dynamic mechanization of

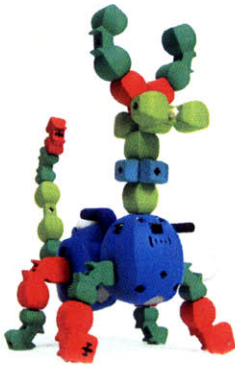


Fig. 5-3 A Topobo Moose

gestural movement. Topobo was designed in aesthetics and functionality to allow children to investigate dynamic and kinematic systems, investigating physical concepts such as knowledge such as dynamic balance, torque, leverage and center of mass. Topobo takes advantage of the combination of the editability of computer data and the physical immediacy of a tangible model and provides a means for expression and investigation of patterns and processes not possible with existing materials.

The Topobo system consists of a physical grammar of ten different primitives, nine of these primitives are called “Passive” because they form static connections. One “Active” primitive is built with a motor and electronics. The motorized components are the only ones that move, so the system is able to faithfully record and replay every dynamic manipulation to a structure.

### Early Design Studies

We began our development by surveying many different types of actuation technologies including magnetic and motorized, both rotary and linear. Due to the high quality and affordability of miniature motors compared to other actuators we chose rotary motion as a kinetic constraint, and initially built dozens of physical prototypes out of plastic and cardboard to study spatial geometries with rotary motion. This led to the development of the current system geometry and a proof of concept using Cricket microcontrollers and servo motors [fig. 5-5]. The Cricket

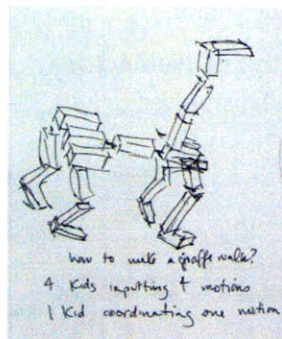


Fig 5-4 Early design sketches - a drawing and model made with LEGOs and rotating joints



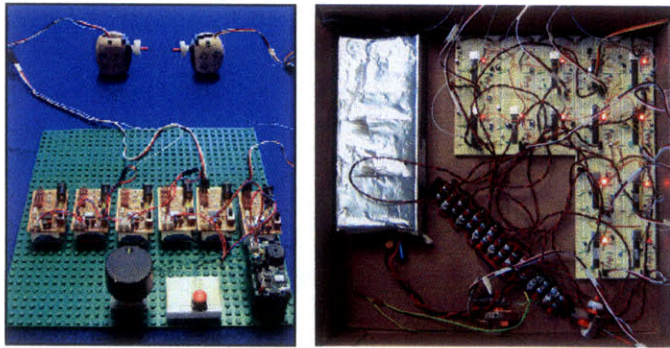


Fig. 5-5 First prototypes - a cricket prototype (left) with servos encased in wood and our first scalable prototype (right) custom breadboarded electronics with daisy-chained power

prototype was extremely fast to implement and allowed us to experiment with the capabilities of the system design. Our first scalable prototype followed, made with wood, hobby servos and breadboarded electronics. Evaluations of this system with kindergartners and second graders helped guide the design of the current system.

For aesthetic and formal considerations of the system, we studied examples of sculpture, patterning in nature, as well as existing constructive assembly systems to inform our design decisions. All the components of the system are intended to be aesthetically consistent both visually and formally. The pieces should individually feel “complete” but be able to combine with other pieces to create unified-looking creations. With the physical design, we also faced a challenge with regards to scale, we needed to create a system that fit comfortably in the hands of children and allowed for small detailed creations while working within the constraint that the minimum size of the Active was determined by the spacial needs of the embedded electronics and motor.

For ease of prototyping, the original pieces, both Active and Passives were lasercut in wood. This proved to be a

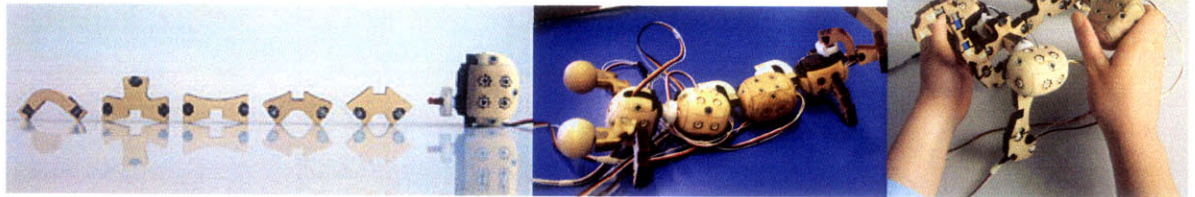


Fig. 5-6 Early prototypes lasercut in wood lent the pieces a hand-crafted feel

comfortable homage to hand-crafted objects while forming a particularly interesting relationship to the embedded electronics. This aesthetic appealed to us and was popular with users although difficulty in molding wood for more three dimensional forms stopped us from continuing with wood as a material when we continued design development and manufacturing. I returned to his aesthetic in one series of designs for Kinetic Sketchup.

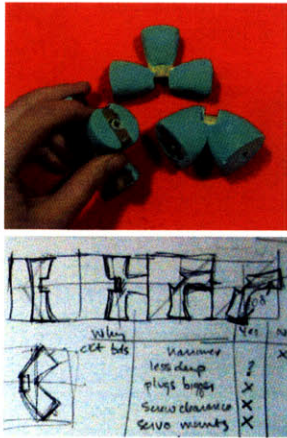


Fig. 5-7 Studies for passives - sketching and in wood and clay

### Passives

We designed nine different Passives to allow a variety of physical structures to be built. Since Topobo is intended to model various natural forms like anthropomorphic skeletons and regular geometrical meshes, the system allows branching and spatial looping. The Topobo geometry is based on a deconstruction of cubic and tetrahedral crystalline geometries [Thompson 1942]. Topobo has five different primitives shapes, four of which come in 2 scales: a “straight” piece, a “T”, an “L” (90°), a “tetra” (108°), and an “Elbow” (offset 90°). The “elbow” (offset 90°) comes in one size. The “straight,” “T,” “L” (90°), and “tetra” (108°) shapes come in two sizes with a scale ratio 2:3, based on the Fibonacci ratio that describes scaling in growing systems like mammalian skeletons [Fig 5-8]. All the pieces except the elbow have a hermaphroditic notch across their center, allowing any two pieces to connect and branch at a right angle. For example, two straight pieces will form a “+” shape, or two tetras will form a tetrahedron. This arrangement allows the formation of regular meshes like a tetrahedral lattice or simple forms like a pentagon or square and this regularity allows for easier creation of large, interconnected forms.

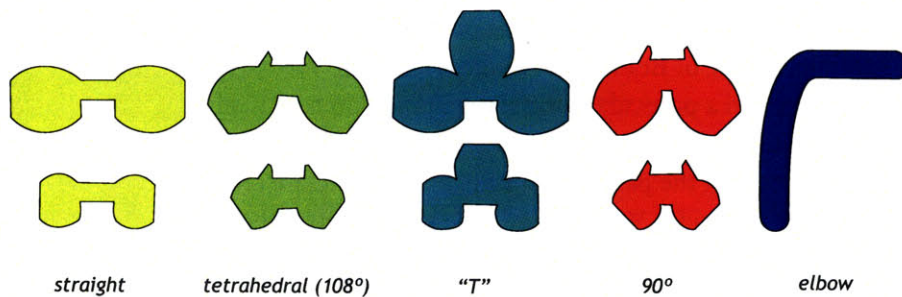


Fig. 5-8 Nine different shapes of the Passives



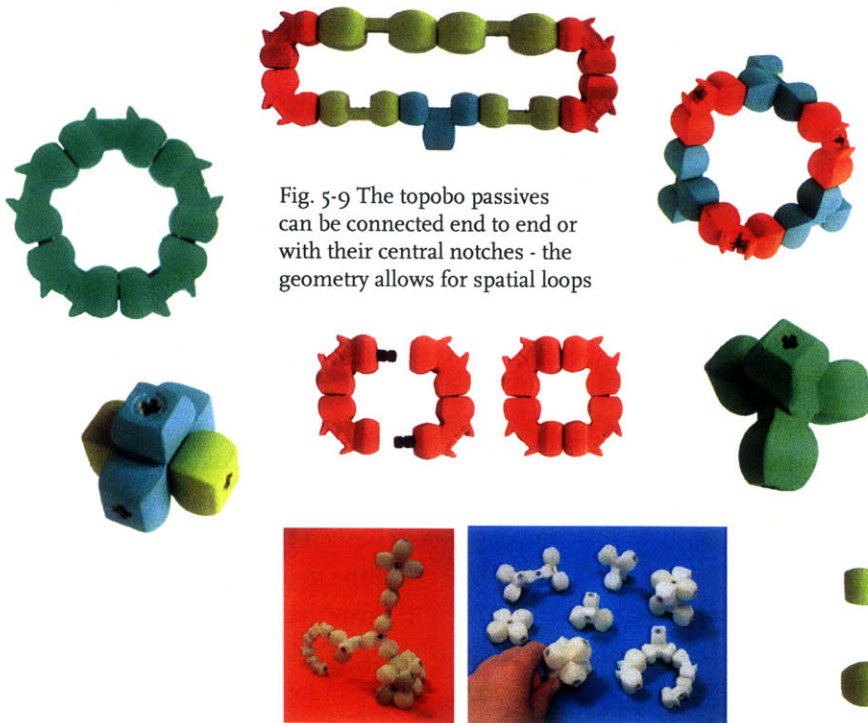
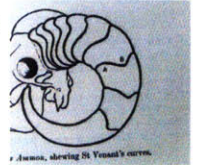


Fig. 5-9 The topobo passives can be connected end to end or with their central notches - the geometry allows for spatial loops



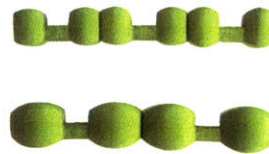
A set of 11 bodies, to show close-packing. From [unclear]

Fig. 5-10 Growth patterns and crystal structure inspired the geometry of Topobo



r. Jensen, showing St. Yonah's curve.

scaling 3:2



Our early wooden prototypes provided a two dimensional skeletal basis. When moving the products into a more three dimensional form, we wanted to give the silhouette of the pieces a natural soft curve while maintaining a square cross section which intended to imply the possible 90 degree orientations of the notch connections. The biggest challenge of the passive design was considering how the ends of the pieces would meet to give the impression of unified creations. We experimented with different curvatures - extruded cones or bell shaped flares ending in a circle. We decided that we would segment each side of the passives into a "bulb" much like those of Brancusi's "Endless Column" [fig. 5-11]. This provided a particular formal inspiration on how to give the system an organic yet geometrically regular feel, while allowing each of the passives to have the same square cross section on the ends. With this design, all the parts in both scales could match end-to-end with a consistent silhouette.

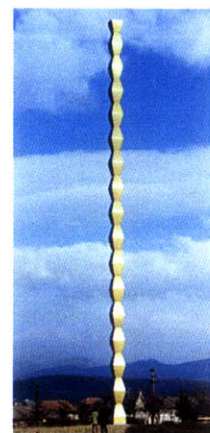


Fig. 5-11 Brancusi, Endless Column, 1938

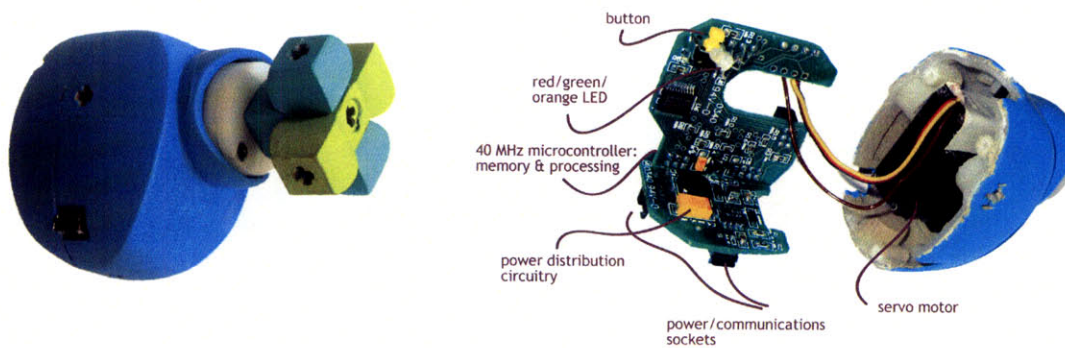


Fig. 5-12 the Topobo Active

### Actives

The Actives are motorized, networkable, plastic pods with a single button and an LED for indicating whether the system is in record (red) or playback (green) mode [fig. 5-12]. The housing has six points of mechanical connection, three sockets to connect power/communication cables and one button that is backlit by a red/green LED. One of the mechanical connectors is connected to the output shaft of the servo motor and rotates 170°. On board custom electronics handle power distribution, memory and processing, and peer-to-peer, multichannel serial communications. Each Active is identical and autonomous, and only needs power to function. The original Topobo actives were built by encasing servo motors in wood housings embedded with LEGO plugs for mechanical connections. The electronics were on separate bread boards, attached to each servo by wires. The electronics were then converted to a PCB and we conducted several design iterations on the Active case, attempting to minimize the size of the Active while incorporating the PCB.

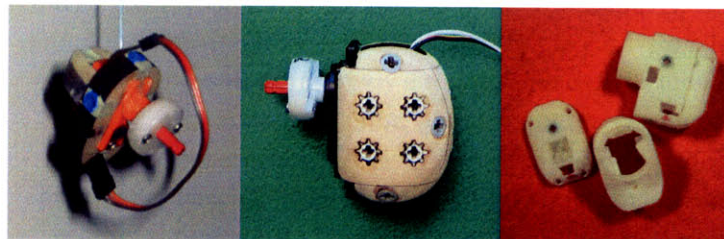
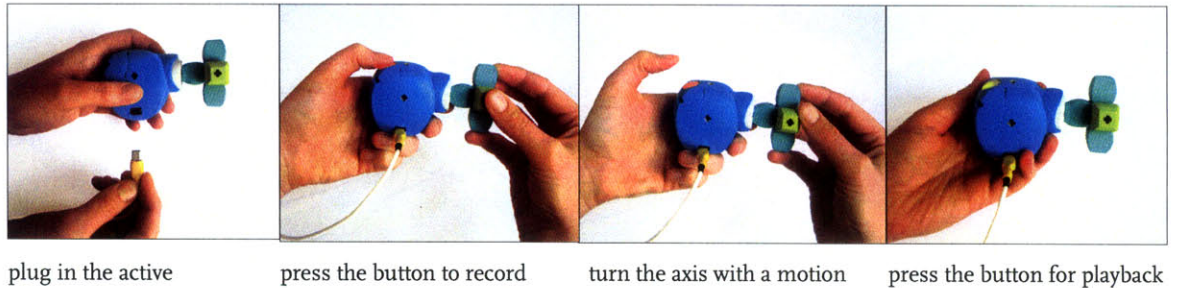


Fig 5-13 First Topobo actives were servo motors encased in wood with off-board electronics

Fig 5-14 First 3D printed case with embedded electronics - aesthetically too "bulky"



Fig. 5-15 Programming an active



A Topobo Active is programmed by direct manipulation, where each Active synchronously records its own motion. To record a movement, the user presses the button on an Active, twists and moves the Active to program a sequence of behaviors, and then presses the button again. The Active immediately goes into playback mode, which repeatedly replays the users input until the button is pressed a third time, which makes the active stop moving [fig.5-15]. A ‘double click’ of the button replays the last recorded motion of the Active. A one-button interface was inspired by curlybot [Frei, 2000] and chosen because it is extremely easy to use. This makes the system accessible to young children, and it allows older children to focus on structure and kinematics rather than on learning a new programming interface. While the one button interface is limited, 3D motion concepts are complex and the immediacy of the interface design encourages rapid experimentation with motion. Physical programming by example also results in natural looking, emotionally engaging motions because they are the reflection of the user’s own body movements.

In a creation with many Actives, all of the Actives will record and playback at the same time. The system treats each button identically; a user can start a recording with any button, and stop the recording with any button. Topobo’s distributed design allows it to be a “high level” interface for thinking about kinematic systems because it lets the user focus on the global behavior of their creation. When a button is pressed, all of the Actives synchronously record their local motions. If a user makes a circular ring of Actives and teaches it to



Fig. 5-16 A Topobo “wheel” which rolls like a tank tread





Fig. 5-17 Programming with a queen: all the Actives synchronously copy the motion of the Queen

roll across the floor like a tank tread (the “wheel”) [fig. 5-16], he or she only needs to understand and program the overall deformations of the ring. The Topobo system automatically decomposes the global motion in to local motions.

### Queens - Centralized Control

The Topobo system also includes special orange Actives called “Queens” which provide a means for centralized control by commanding other Actives to copy their motion. If a recording is started by pressing the button on a Queen, that Queen controls the entire network. The Queen transmits a direct copy of motion: the user turns the output shaft on the Queen and all of the other Actives synchronously mimic the Queen’s motion. For example, suppose that one constructs a linear structure of actives with a Queen at one end. When the Queen is recording, all of the other Actives will mimic its angular position. Thus, increasing rotations to the Queen cause the entire structure to begin to curl into a circular form [fig. 5-19]. Eventually, the ends will touch. A Queen can also



Fig. 5-18 Actives connected with tetrahedrons and programmed with a queen create a spring-like helix

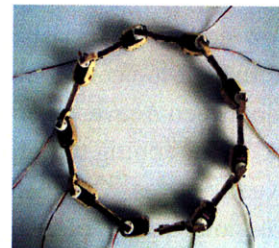


Fig. 5-19 A linear sequence with direct copy creates a circle

make tangible ideas in spatial translation- within a creation such as two facing Actives, such as the legs of a symmetrical animal, the Actives will exhibit opposing, mirrored motions, often surprising to children.

Since a Queen does not need to be mechanically attached to the creation it is programming, it can also be used as a remote controller. Remote programming with a Queen gives a user synchronous input and output feedback during programming, allowing a user to actively debug their creation's motion while they are composing it. Using a queen as a remote control, the creation can respond to the physical conditions of the real environment, not interrupted by the user's grip.

### Technical Specifications

The Actives' on-board custom electronics handle power distribution, memory and processing, and multichannel serial communications. A 24V power bus is locally stepped down to 6V with a buck converter and then is dropped to 5V with linear regulator that powers the digital electronics. This minimizes the effects of power losses in the system, limits noise transfer between Actives and reduces current requirements. A 40 MHz AVR microcontroller handles local processing and network communications. A one-time calibration sequence measures the range of motion of the servo and correlates input and output position data. During record, the microcontroller reads the servo's internal potentiometer at 20Hz using a 10 bit ADC and writes scaled 8 bit values to local memory. This provides about 30 seconds of record data at  $3/4^\circ$  output resolution, which is accurate compared to the backlash in the servo's 4 stage gearbox. A custom peer-to-peer serial networking protocol can transfer data between Actives at 9600 BPS. Specialized line drivers allow hot-swapping power/communication connections between Actives. Originally, a TowerHobbies HS81-MG servo motor with  $170^\circ$  rotation was chosen for its high strength to weight, robust metal gears, ease of back driving, and included sensor and drive circuitry. The servos' output



Fig. 5-20 3D model of the Topobo Active case

shafts are outfitted with a custom clutch to protect the gears from excessive torque. In manufacturing development, we developed an entirely custom servo motor, optimized with gearing specifically designed to withstand back driving. Further technical specifications of the final design can be found in Appendix A.

### **Color**

The color palette of Topobo consists of secondary and tertiary colors which were chosen to lend visual sophistication but with a fun and playful edge. We wanted the colors to retain a relationship to nature although imbued with greater saturation. We developed a palette of cool colors (blues and greens) with one accent color (a deep orange) to give characters a visual “pop.” These colors are all tonally consistent so that none is much brighter or darker than another. The system specifically avoids the traditional primary palette (red, yellow, blue) of many children’s toys and the colors were intended to appeal equally to both genders. The parts are color coded by shape to be able to easily distinguish between forms and to lend themselves to playful, unified looking creations.

### **Initial Evaluation**

Our first evaluation with Topobo was conducted in classrooms at Shady Hill School in Cambridge, MA with 25 kindergartners (5-6 years old), 22 second graders, and 32 eighth graders to evaluate Topobo’s effectiveness as an educational tool for children at various educational levels. Our evaluations with two eighth grade “Physics by Design” classes focused on Topobo’s role supporting design, experimentation and conceptual abstraction. These students normally engage in group projects using manipulatives like LEGO Robolab, so the evaluation was designed to be like familiar classroom activities. We met with four groups of 8 students twice over two weeks, and students worked in pairs or groups of three. These sessions included three homework worksheets and interviews with students. Our



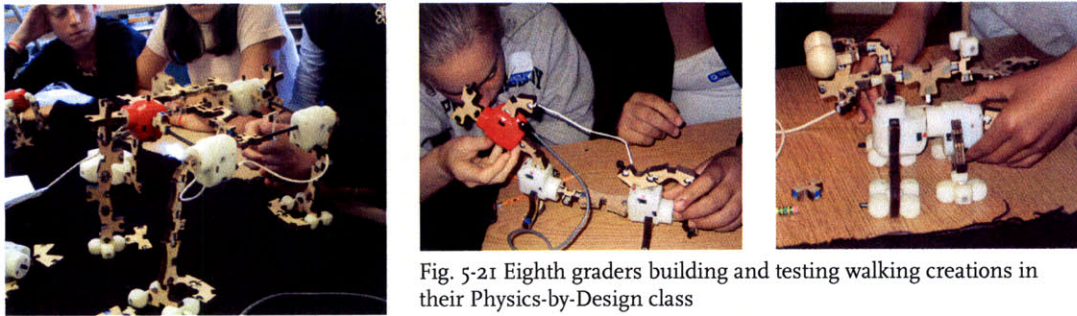


Fig. 5-21 Eighth graders building and testing walking creations in their Physics-by-Design class

first evaluation session introduced the system. Using a preliminary worksheet, students described different types of motion related to their bodies based on both their preexisting conceptual models of motion and then based on activities we designed. The next day, we explained how to use Topobo with demonstrations and examples. Students began by freely exploring the system. Many students built anthropomorphic creations, programming them to tell stories or wiggle around. Their creations often did not move as they expected. Falling creations elicited exclamations like “add more legs” and “make it lower, like a baby.” For most of these students, Topobo quickly became a tool to experiment with center of gravity and dynamic balance.

The second evaluation session a week later focused on a task to construct a “walking creature.” Students first planned and drew their creature and then tried to build it and make it walk. We observed two different methods of design. The first method involved “active iteration” during the creative process. Students built a small part of a creation, programmed it repeatedly until the desired motion was found and then added components, testing how the new components changed the dynamic balance of the creation. This process continued until they had their desired creation. The second method involved students who would “compartmentalize” the processes of structural building and programming motion. Students who compartmentalized would build a creation in its entirety and then program its movement only at the end of their process. Students who employed active iteration were more successful at building creations which walked and balanced. These

students' creations tended to be very different from their original designs on paper and the students were generally able to explain how physical constraints had influenced their designs. In comparison, students who compartmentalized building and programming usually ended up deconstructing their creation and trying to rebuild it using a more iterative process. These findings show that an interface design should support active iteration by allowing users to switch between interdependent processes. Users often need to test many ideas to incrementally develop a successful design. Students who initially compartmentalized the design of form and motion eventually adopted active iteration, suggesting that Topobo supports rapid experimentation with these interdependent processes. However, these findings also suggest that Topobo would benefit from an ability to modulate and tweak recorded motions while in playback which led to the development of the Backpacks.

### **Domains of Knowledge**

From our initial evaluation, we have identified several education concepts which Topobo can help students ages 7-13 to learn. Many of these concepts are usually not introduced to children until they are much older, we believe Topobo can make them accessible earlier by the physical intuitive on which the tangibility of the experience relies:

**Dynamic Balance:** When objects move, their center of gravity changes. Topobo draws attention to this fact when children make things that fall over. Learning how to control falling can lead to an understanding of familiar dynamic processes such as walking.

**Center of Mass/Center of Gravity:** Several groups of students built creations that were initially very tall and tended to fall over when they moved. One student described shortening the creation's legs to keep its weight closer to the ground. He referenced how it is easier for babies to crawl than walk.

**Coordination:** When Topobo is directly manipulated, sequential motions are easy to record. A child might shake his Topobo dog's head, and then wag

his Topobo dog's tail. However, shaking the dog's head and wagging the dog's tail at the same time is difficult because the child needs both hands to do either one of the activities. In order to coordinate these motions, it is necessary either to cooperate with other children (coordinating people) or to use a Queen (which coordinates movements in time). The Queen encourages developing an understanding of how coordinated movements can change a whole system.

**Relative motion:** A second grader built a long string of static parts with an Active part at each end. He programmed each end to wiggle back and forth and observed the ends shaking. Upon suggestion from an adult, he tried holding a shaking end, and was amazed to see his entire creation wave wildly back and forth. This drew his attention to the idea that movements in a connected system are relative to one's frame of reference.

**Movement with Multiple Degrees of Freedom:** A Topobo Active provides motion in one degree of freedom. One pair of eighth grade girls quickly figured out how they could connect two Actives with an elbow piece to create 2 degree of freedom rotational motion. By applying this technique they were able to quickly create a walking moose. While they could not explicitly describe how it worked, their implicit knowledge of these dynamics was evident when they refined the same kind of motion in a different creation a week later.

**Relationships between Local and Global Interactions:** The educational value of understanding relationships between local and global interactions has been investigated at length with object-oriented programming languages such as AgentSheets and StarLogo [Resnick 1999]. Topobo makes certain systems concepts tangible with the Topobo Queens. One group of 8th graders discovered that faster legs (local) do not make a faster animal (global). Another group of three boys figured out quickly that they could create two separate networks of legs on either side of an

animal, each governed by a Queen. Using this concept, they would be able to program each pair of legs with different motions but the legs in each network would have the same repeated motion.

We were excited to see children responding favorably to Topobo and have been easily and intuitively able to comprehend how to operate the system. We are particularly encouraged that children from ages 5-13 when asked what age range they thought Topobo was designed for, they responded their own. We found that Topobo offered an entry point for children at all stages of development and contained enough depth to engage deeper and more meaningful interactions with children as they grow. This depth of experience also suggested Topobo's potential to move beyond the realm of children's toy into different realms of motion design and exploration for adults.

### **Topobo Backpacks**

Backpacks are modular physical components that users can incorporate into Topobo creations to modulate recorded motions while in playback. The Topobo backpack provides a means to physically embody a varying system behavior by moving and forming in response to a mathematical function. The challenge backpacks address is to create a tangible interface that can retain the immediacy and emotional engagement of "record and play" and incorporate a mechanism for real time and direct modulation of behavior during program execution.

The Backpacks are small "discs" which connect to an Active both mechanically (through a LEGO connector) and electrically (through one of the communication ports) and feature a knob (potentiometer) for control of the behavior that they send. Backpacks can be connected either to a regular Active, where their behavior affects only that Active, or to a Queen, where their behavior is sent through the system based on a specific transforming function. Backpacks have three different modes that give children tools to explore their creations' interactions in detail:

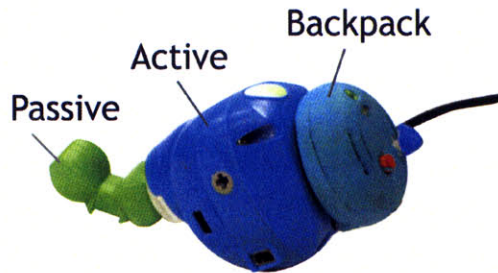


Fig 5-22 a Backpack attached to an Active

Local: When a Backpack is attached to an Active, it affects only that Active.

Global: A Backpack is attached to an Active, and its button is pushed. Or, the backpack is attached to a Queen. The Backpack identically affects every Active in the structure.

Distributed: A Backpack is attached to a Queen and its button is pushed. The backpack affects all Actives and its modulation is proportional to an Active's number of network hops from the Queen. Here, the rate of change is controlled with the Backpack's knob.

We have developed four types of backpacks:

Bigger/Smaller (Amplitude) Backpack causes the Actives motion to be incrementally scaled as it is passed from the source. A linear string of Actives can gradually curl in to a spiral.

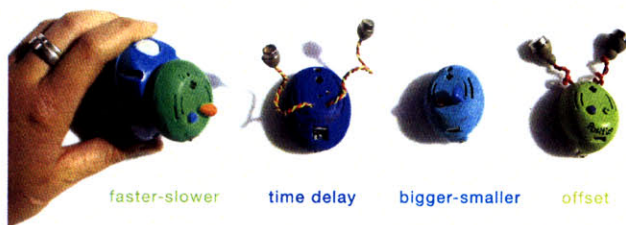


Fig. 5-23 4 Topobo Backpacks - Faster-Slower, Time Delay, Bigger-Smaller, Offset



Faster/Slower (Speed) Backpack increments a change in period as a motion is passed. Due to Topobo's looping playback, a linear string of parts can exhibit harmonic interference patterns.

Time Delay (Phase) Backpack aggregates a time offset before playback of the Queen's motion. A linear string of Actives can move with wave-like motions.

Offset (Orientation) Backpack changes the position of playback relative to the position of the Backpacks potentiometer

Backpacks make tangible some of the benefits of symbolic abstraction, and introduce sensors, feedback and behavior modulation to record and play in a physical model-making paradigm. Backpacks extend the conceptual limits of record and play with an interface that is consistent with both the physicality of educational manipulatives and the local-global systems dynamics that are characteristic of complex robots. The combined functionality of Queens and Backpacks creates the capability to pass a kinetic behavior with a transforming variable through the system, presenting a means to represent and understand mathematical based simulations in a physical material. As part of a distributed decentralized system, this can demonstrate physically how a simple set of rules can lead to complex form and behavior.

#### *Ambient Sensors and Conditional Behaviors*

Offset Backpack has two antennae with light sensor "eyes" in place of its knob [fig. 5-23]. It demonstrates conditional behavior and environmental responses when children can use it to build creatures that can change their posture in response to ambient light. For instance, a child can design an ant that walks towards light. By manipulating the orientation of the antennae, children can discover principles about sensors and control; a creature that walks towards light can be made

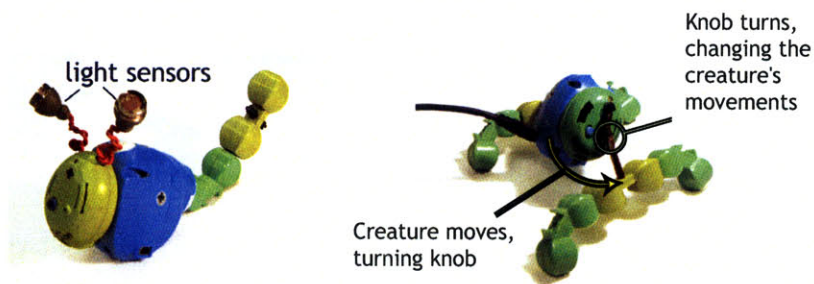


Fig. 5-24 Backpacks with Environmental Feedback, offset backpack with light sensors (left), and a mechanical feedback knob (right)

to walk towards darkness by crossing the two antenna to opposite sides of the Backpack.

#### *Feedback*

Backpacks can also be used to experiment with feedback. The Backpack's knob is fitted with a mechanical connector that allows it to become part of a creature's body. Now, the creature will behave differently when its posture changes. If the backpack is modulating the same motion that is affecting the position of its input knob, it presents a type of physical feedback mechanism [fig 5-23].

### **Backpack Evaluations**

Evaluations of the Backpacks took place in classrooms at Shady Hill School as well as a variety of informal settings (afterschool playgroups) with children aged 6-15. Throughout our design process, we frequently showed the system to children to determine its ease of use and affordances for manipulating its controls and combining it fluidly with the Topobo system. These sessions informed the final physical and interface design of the Backpacks.

#### *Kindergarten—Third Graders*

We evaluated the Backpacks to explore their effectiveness in how tangibly manipulating motion parameters could facilitate the development of abstract ideas about motion. We conducted several informal afternoon sessions in a home

environment, with eight children ranging from K-3rd grade, a mixture of boys and girls. The children were first introduced to the Topobo system, demonstrated how to use it and shown several Topobo creations which took advantage of the Backpack capabilities. They then had an afternoon of free play with the Topobo system and Backpacks with help available from researchers accustomed to working with children and Topobo. Most of the children in the session had not played with Topobo before, except for one third grade girl who had experienced early Topobo prototypes in her kindergarten class, and another seven year old boy who had evaluated Topobo informally in approximately six sessions in the previous two years.

#### *Eighth Graders*

Our next evaluation took place in the eighth grade classroom, in a physics-by-design class. We conducted two sessions with two separate classes, with a total of 26 students. These students had no previous experience with robotic or programming systems and had not been taught a foundation in dynamics or kinematics. However, the school they attended had a hands-on approach with manipulative materials available as part of the curriculum. In the first session, the students were introduced to the Topobo system and Backpacks and given free play with the system.

In the second session, the children were shown successful walking creations we had built, some of which utilized the Backpacks. We demonstrated how the Backpack parameter control could manipulate walking. Following the introduction, half the class was given these built creations to analyze—take apart, change, rebuild—while the other half were instructed



Fig. 5-25 Eighth graders experimenting with the Time Delay Backpack and the Offset Backpack with light sensors

to create their own walking creatures. In between the sessions the classes were given homework workshops to test their conceptual understanding of the Backpacks and all the students were interviewed at the end of the last session.

In both of our evaluations, we found that the Backpacks were an accessible interface for children to explore different parameters and introduced a set of concepts that ranged in complexity. All of the children were able to use the Backpacks, although a greater conceptual understanding was articulated by the eighth graders. Showing the children built creations with the Backpacks in use and allowing them to deconstruct their behavior greatly accelerated the children's conceptual understanding. This was a necessary first step with the younger children to engage totally with the Backpacks.

The Backpacks that described more concrete physical concepts—moving Faster-Slower or Bigger-Smaller—were easier for all the children to observe, understand, utilize and describe. One eighth grade boy commented on how the Faster-Slower Backpack made getting his creature to walk easier. “You could probably do it without it, but it makes it a lot easier...rather than having to rerecord it every time you want to change the speed...you can also get it a little bit more precise with the Backpack.” When employed in a creation, the children were able to understand that the Delay Backpack made the Actives move one after another, thus dissecting a fluid motion into its constituent parts. However, they did not articulate a direct connection to wave-like motion. The Offset Backpack proved to be the most difficult for the children to dissect; children could interpret that the sensor made the creation move toward the light, but only one group of eighth graders was able to articulate an obvious correlation with how the motion of the motor was changing (offsetting to one side) in relationship to the overall walking behavior that the creature demonstrated.

#### *Fluid Integration Into Play*

An important attribute of the Backpacks was observed in how the Backpacks were integrated into the creative process of

using Topobo. In past studies with Topobo, researchers found that users who worked iteratively—going back and forth between building the creation and programming motions—had more success in making a creation walk. We found that the Backpacks integrated seamlessly with this iterative process, while adding a new element with which to iterate. In one session, two eighth grade boys were working on a walking creation with the Faster-Slower Backpack. Throughout their process they explored adding and removing passives to change the weight balance of their creature, reprogramming its motion, and changing the speed with the Backpack knob — all in a fluid and experimental manner. They cited the Backpack as being a necessary part of their creature, because it allowed them to control the speed of their creation without having to also reprogram (and thus overwrite) the motion pattern.

#### *A Logical Next Step*

In one situation, two eighth grade boys had built a creation with a single active that walked forward and then attempted to make their creation turn in one direction. Through experimentation they found that they could successfully change the form of the structure, adding and subtracting passives to its legs, or could manipulate its motion, adding a new Active to its back which functioned to offset the motion like a steering column. In essence, these boys had struggled to discover the principle embodied in the Offset Backpack, which could have easily facilitated their iterations. This situation supports the idea that the Backpacks are building on motion principles already inherent in the system, but are providing a more abstracted and flexible form for students to approach and investigate the concepts they demonstrate; the Backpacks' functionality is a logical inclusion in the Topobo system.

#### *Beyond Children*

Throughout our research, dozens of adults (some of them leading robot designers), have experienced the Backpacks with the Topobo system. All of these users expressed enthusiasm for the Backpacks, especially those people who are

professionally focused on examining the relationship between geometry and movement. Scientists and experts possessed a particular excitement about the distributed Backpacks, recognizing the importance and extensibility of them as a tool to understand the applications of concepts such as wave propagation or system dynamics. They described Backpacks as reflecting the real high level ways of thinking about robotics and motion control, viewing Backpacks as a tool for intuitive manipulation within a control structure.

#### *From Play to Abstraction*

A central question to different kinds of design tools concerns ease of entry (the “learning curve”) and the potential complexity and sophistication of models created with a tool (the “ceiling”). One of the original pedagogical arguments with Topobo was that children of widely ranging developmental levels became engaged with Topobo because it was easy to learn and there were many points of entry for different learners; many levels of complexity were embedded in the system. However, children who were adept with manipulating abstract ideas [Cole 2001] wanted to manipulate their recordings in different ways. Backpacks increase the complexity with which children can design, control and understand their creations.

Whereas an informal system like Topobo can lead to accidents and discovery, a pedagogical benefit of providing parameterized control via manipulatives is that advanced learners can fluidly transition between building, dissecting, and controlling their model. Control is one level removed from spontaneous creation, and Backpacks may help children to discover what, exactly, makes a behavior successful. This may benefit learning, since, as Ackerman argues, effective learning often involves temporarily standing back from the learning experience to reflect on it in more objective terms [Ackerman 99].

## *Chapter 6*

### **Longitudinal Outreach with Topobo**

Concurrent with the development and evaluation of the Topobo Backpacks, we won a grant from iCampus, the MIT-Microsoft alliance, which allowed us to have Topobo re-engineered and mass produced in a two years of extensive collaboration with an Asian toy manufacturer. This allowed us to begin a series of longitudinal studies in which sets of manufactured Topobo were distributed to educators (teachers, museum developers, educational researchers, graduate students) in the United States and Europe. The sets included Actives, Passives, basic Queens, power supplies and cables, and simple booklets. The booklets described the project concept, design and technical details, instructions for programming, and three sample creations with basic assembly instructions. The educators were also directed to the Topobo website which contains additional videos, published papers and visual materials.

The focus of the long term study was on the conception of the educator, and their use of the tangible interface in the absence of an inventor or HCI researcher and addressed the following research questions:

- In what contexts and environments can Topobo succeed?

Educator	Context	Student Age	No. students	Time Span	Interaction
teachers	after-school enrichment program	13-15	18	3 months	themed sessions, free play
science teacher	4th & 7th grade science classrooms	9-10, 12-13	36	8 months	goal-oriented lessons, free play
educational researcher	after-school robotics center	4-6, 8-14	32	5 months	guided sessions
exhibit developers & programmers	science museum	4-adult (target 9-15)	200+	4 months	on-the-floor museum activities, demos, internal conversation
graduate architecture students	architecture course/studio	24-29	12 (focus on 1 specifically)	8 months	self-directed thesis design work

Fig. 6-1. Breakdown of the five selected case studies

- Over what time period will children use Topobo, and how will their use and interpretations of the system evolve?
- What age children will benefit from Topobo, and how will their experiences differ?
- What uses will other educators invent with Topobo?
- Can Topobo be used to illustrate higher level abstract concepts related to motion design?

The results identify design and pedagogical issues that arise in response to distribution of a tangible for learning in different educational environments. We focused on case studies in the following environments:

- Morse School, 3rd grade, (public), (Cambridge, MA)
- Shady Hill School, 4th & 7th grade (private, project-based focus) (Cambridge, MA)
- Brookline High School, Engineering science class (public) (Brookline, MA)
- Exploration After-school Enrichment Program (Boston, MA)
- Kids' Club After-school Robotics Center, (U. Joensuu, Finland)
- Tufts Center for Engineering Educational Outreach (Medford, MA)
- Boston Museum of Science (Boston, MA)
- Harvard Graduate School of Design (Cambridge, MA)





Fig. 6-2 Children play with Topobo at a festival in Denmark

Extensive data was collected over a year and a half, mostly in the form of interviews with educators and educational researchers working with Topobo. We sought to examine the perspective of the educators, and their reactions and plans when presented with Topobo as a new educational toy or kinetic material. We report how Topobo was used by various educators and what kind of initiatives, programming, or curricula they developed in these different environments when the researcher was removed entirely from designing a study or guiding the technology. In this respect, the teachers (not their pupils) are the “users” we are addressing here.

### **Selected Case Studies**

The following five case studies [fig. 6-1] represent a sampling of our research findings in diverse educational contexts with varying aged populations. They represent a cross section of usage environments, target age user and target user scenario. They were chosen because they are representative of common findings while at the same time offer significant depth and layered complexity from which to draw analysis. We aim to highlight the specific issues associated with using a tangible technology in different environments, and to identify the common issues that arise for educators in all environments. Four are presented as the remainder of this chapter while case study 5, at the Harvard Graduate School of design, is included

at the beginning of chapter 7, as it became a direct launching point for the development of Kinetic Sketchup.

### **After-school Enrichment Program**

Over the summer, sets of Topobo were loaned to an after-school enrichment program for middle and high school students. The director, Lori first saw Topobo in use in a local classroom and inspired by its potential, sought out to procure sets for her summer program. She intended to provide the system as an inspiration material for her program teachers with the hope of incorporating it in a more structured way the following summer. We provided a basic explanation of the system but did not set expectations of what we thought it should be used for or how we saw it fitting into her program. As curriculum director, Lori became the liaison to the teachers, explaining the system. Her enthusiasm for Topobo was shared by Dale, a middle and high school technology teacher in the program who used Topobo in his class.

Dale conducted two sessions, two hours long, each of 7-9 students aged 13-15. Students elected to join both sessions and the second session contained many repeat students from the first session, which Dale interpreted as a sign that the students had made progress with Topobo and wanted to learn more. After some quick initial experimentation on his own, Dale began by giving the students a challenge of which he participated, "I'm having trouble getting something to walk [in reality, he was], can you make it walk?" Three boys in the session ended up making a walking robot but did a lot of purely structural experimentation until they began to use the Actives to actually connect, control and locomote the structure.

In the second session, Dale decided to present a series of scientific concepts to enrich the experience of Topobo, but by his own admission, he got carried away with what he wanted to achieve, frustrating himself as well as the students. In the first half hour he used only the passives, looking to explore the systems' geometry and angles, wanting to instill an overall sense of 'engineering platonic solids.' Then he brought in the

Actives and shifted to how the system could mimic molecular reactions, like breaking and creating chemical bonds. He described that upon first seeing Topobo, it immediately reminded him of a PBS special he had seen that showed DNA being spliced. In this vein, he wanted to teach chemical bonding with it, explore crystalline structure, and on a different scale, tensegrity. Dale figured out midway through the session that the material was too dense and presented too quickly for the students.

Dale's sessions ignited both excitement at the possibilities of what Topobo could demonstrate and frustration at his own inability to immediately put them into action. At multiple times during our interview, he suggested the need for a teacher's guide which would provide advice on building creations that walked successfully. He was careful to stipulate that the guide should not didactically provide exact instructions, but rather that it should provide general design guidelines and examples on how to obtain a particular kinetic behavior, combining structure and programming. He described the guide as scaffolding for the teachers to gain a deeper understand of Topobo's possibilities, as opposed to a series of lessons plans to implement in the class. The guide should also feature common mistakes students make when working with Topobo, to keep teachers preemptively informed. Dale and Laurie also suggested running a workshop for teachers, possibly at an education conference, combining teachers of all disciplines.

#### *Pedagogical Ideas*

Even after limited initial exposure to Topobo, Dale, Laurie and other teachers at the program were overflowing with curriculum ideas of what Topobo could be used for in the classroom. A language arts teacher suggested using Topobo to find the rhythm of poetry, almost like a metronome, programming a creature to move in a particular rhythm and asking the students to write a poem about this creature matching the rhythm of the poem to Topobo's. In addition to his ideas about chemical reactions, Dale mentioned that his 8th grade technology class made Rube Goldberg devices in

which Topobo could be easily incorporated. “We could connect it to a ramp or some kind of switch then we could set a whole bunch of other events in play.” He discussed several scenarios for creating real world models for math and science concepts, such as parabolas, using a Topobo construction to knock a ball into the air, like an automated golf club, observing a parabola created in a real world situation. He also envisioned Topobo to be of use in discussing elementary circuit design: he wanted to figure out a way to create a logic relationship, like an and/or gate, between a Queen and the Actives. He struggled with how he would design it but had a sense that by mimicking a programming structure in a physical behavior, it could become more intuitive and easier to comprehend for the students.

#### *Discussion*

Dale begins by using Topobo as a holistic system, creating walking creatures with his students, but soon transitions into a mind set envisioning Topobo as a tool for simulations ranging in scale and time: it becomes an enabling technology for kinetic behavior. This shift shows how Dale has come to recognize Topobo as a flexible and open-ended modeling tool. However, he recognizes the limitations in time and effort of putting those models to work in a classroom, “In general, education is something where you want the fastest and easiest solution, and if it’s something you have to stretch your imagination to make something work for a specific situation, that’s not something people usually do in a classroom.”

Laurie offered a more theoretical perspective on Topobo’s suitability for a classroom situation, “What Topobo offers is that surprise element...It’s intriguing just in its design and its newness, it has that cool factor... maybe I’ve been taught parabolas before but maybe now that I can make one happen with Topobo, it may sink in. Teachers have to teach and reteach and do it in different modalities and do it in different intelligences in hopes that you hit the one of every kid.” She cites its novelty as a factor which can help draw students in, resonating with students of alternative learning styles, and references a multi-modality that is often a specific design



Fig. 6-3 Girls work with Topobo during a lesson on locomotion in a 4th grade science class

principle of tangible technologies.

### **Elementary/Middle School Science Classroom**

Jane, an elementary and middle school science teacher at a Montessori-inspired school, borrowed sets of Topobo to use in her 4th and 7th grade science classes for 8 months. The school had a hands-on approach to learning and she was accustomed to using manipulative materials in her classes. Our goal with Jane was to learn if Topobo could succeed as a formal educational tool: could it fit within a lesson plan, state educational guidelines and other constraints that teachers juggle daily in designing their class material.

Jane incorporated Topobo in her classroom in two ways, first as part of a lesson plan with a curricular goal with her 4th grade class, and second, as a free play activity (for recesses on rainy days) for both her 4th and 7th graders. Jane initially experimented with Topobo in her home and watched her own elementary-age children, nieces, and nephews play informally with Topobo. She tested some of her pedagogical ideas on them, and based on these observations Jane designed a formal lesson plan for her 4th graders about locomotion.

Jane's students worked with Topobo as part of a unit on structures. Lessons took place in two sessions. First, Jane isolated the activity of programming, and set up a specific task all the students could accomplish: children were given

identical pre-built creatures and challenged to get the creation to walk 30 cm, timing for speed. Jane focused on measurement and data collection as part of this exercise, as well as concepts such as friction, gravity and balance. The children expressed desires for free play and experimentation, and it was difficult to keep them focused on a structured task.

In their second session, students were shown video clips of Muybridge's horse and walking robots as background material on natural and mechanical locomotion. They were asked to build their own four legged creature and make it walk a meter as quickly as possible, and describe the order of the leg movements. In building their own creations, a lot of kids started with a creature very similar to what they had used in the previous session. Jane explained, "its always easier to take a model and tweak it." Overall, she was satisfied with the children's success in the activity and Topobo had engaged the attention of her students the entire time, particularly notable with a student who usually displayed attentional disorder issues in extended exercises.

Jane also provided Topobo as a material for free play, during rainy or bad weather days. Deep engagement characterized students in her 4th and 7th grades. "They really, really, really wanted to play with it. It was unbelievably attractive as a play toy – whoever saw it, whatever the age range, from 19 or 20 to 8, people loved to play with it, but they had a hard time unless they had a model to follow." Topobo was more popular as a play toy than as an educational material for Jane, and this evidence suggests that attractive tools can reach students in school outside the context of formal lessons.

### *Discussion*

Jane represents a teacher who has put in considerable time and effort to understanding Topobo's potential and being able to communicate it to her students successfully in the classroom. She described the time put in as essential for her own understanding. Knowing that she could make basic things gave her the confidence to teach it to the children. However, she still did not feel she had a deep enough understanding of how to start working with Topobo in more



complex ways, nor as a teacher did she have time. “It would be really cool if I could make it do that, but I don’t have time to figure that out.” Jane was enthusiastic about her results using Topobo in her structures lesson, but did not use it for formal teaching again. She felt that one of the most important issues with using Topobo in the classroom was educating the teacher on how to think about Topobo and the opportunities it provides.

When asked if Topobo has a place in the classroom, Jane described her philosophy toward activities. “I go back to simplicity. It’s the efficiency question, like the efficiency of straws and paperclips” to explore structures. Simple materials that are easy to work with can get a salient message across in a very direct way. While Topobo provides a certain ease of entry to use, the newness and novelty of the technology is actually a hurdle to identifying and focusing on underlying science concepts.

Like her students who found it easier to tweak the Topobo model she had built, Jane would have found it easier to tweak lesson plans we had provided her. Supplementary materials such as a booklet of basic constructions, and principles behind why and how they work (not just examples of full activities), would be very helpful to give teachers confidence to push forward with making their own activities for Topobo. This finding echoes Dale’s comments from his experience in the after-school center. One challenge will be to teach sufficiently interesting and new ideas (or old ideas in new ways) so that the cost of learning the technology is outweighed by the benefits of the students using it. From

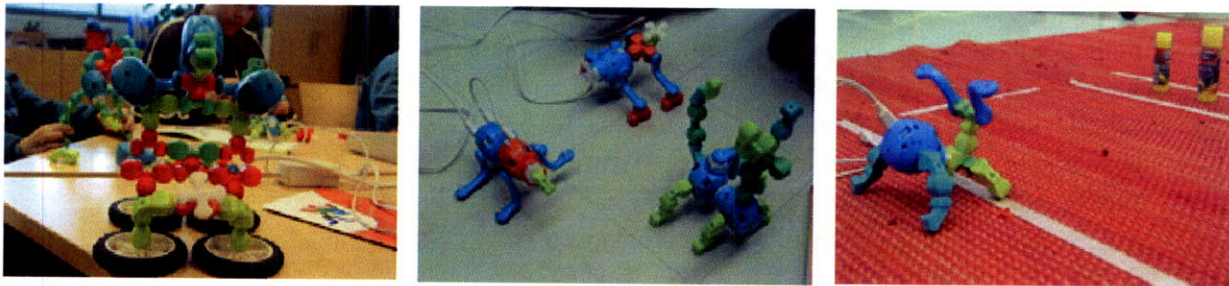


Fig. 6-4 Creations and play by special needs children at an after-school robotics center





Fig. 6-5 Creations by kindergartners in an after-school robotics center after many weeks of play

Jane's perspective, it's hard to compete with the simplicity and economy of straws and paperclips.

### **After-school Robotics Center**

Several sets of Topobo were sent to Mary, an educational researcher studying the advantages and disadvantages of educational robotics for learning with normal and special needs children. Mary conducted her research in an after-school robotics center where children could participate in semester-long courses in which they could engage in somewhat unstructured play with technological tools. She requested Topobo as part of a study investigating how a robotics kit - and a tangible interface in particular - could benefit children in special needs education.

Mary worked with two groups of children, one group aged 8-14 with mixed attentional disabilities including ADHD and Asperger's syndrome, the second, a group of kindergarten school children (non-special needs) ages 4-6. The study looked at 32 children in 13 sessions over a period of 5 months. Each child participated repeatedly in at least 6 sessions, and Mary focused on collecting longitudinal data of children's uses of Topobo.

Both groups of children expressed immediate attraction to Topobo and they engaged continuously with it for long time periods (up to an hour), something very unusual for both populations. With special needs children, Mary found that Topobo kept them very focused but that they needed directed and guided tasks, such as small specific problems to solve or very detailed instructions to follow. With kindergarten children, all kids engaged with Topobo over long time periods

(typ. 30-60 minutes) but some children needed initial scaffolding to understand the programming model.

For both groups, Topobo had a very easy point of entry, different from other robotic systems, and children could quickly and easily build what they desired because the system did not use a on-screen programming environment. Younger children and children who had difficulty with programming could still easily be successful at programming motions for their creations. Over the course of the study, however, Mary observed that Topobo was more suited to the kindergarten. It kept these younger children continuously engaged throughout the sessions, while the older children began to request added functionality such as sensors to build more difficult or complicated programs and scenarios.

As a classroom tool, Mary believed Topobo touches on a number of pedagogical themes including information and communication technology, mechanics, modeling of environments (interdependencies) and procedural thinking. Mary cited that her country's national curricula states that information and communications technology (ICT) should be integrated into all subject matters, but doesn't specify the tools. In this respect, she saw Topobo as a tool that could be integrated into many subjects with younger children. However, children didn't experience these pedagogical ideas directly from Topobo: core technology concepts would need to be introduced in other ways by teachers first, and Topobo could then become a concrete example of the concept.

One area in which Topobo excelled was in promoting collaboration and cooperation between students in both groups. She described that children would first build and program their own creations but then would share and try to program each other's work. They could then use the knowledge gained from each other's experiences to figure out how to make their own creations work better. Why did children collaborate more with Topobo than with other tools? She believed it was because Topobo was easy for everyone to use and understand: not only could a student easily create and program their own model, but they could also easily look

around and understand what everyone else is doing. This transparency facilitated group learning and unstructured collaborative design processes.

### *Discussion*

Mary had success with much younger children than in previous Topobo studies. Although she didn't believe that Topobo was necessarily more attractive to kindergartners than static manipulatives, all young children in her study engaged deeply with it. Where technology-related concepts are sought as part of a young child's experience, she noted that Topobo, with a tangible programming model, allowed for extended play and engagement with technology at a much younger age than systems which required screen-based (GUI) programming models.

Mary's conception, as well as her specific uses, of Topobo stress the importance of establishing in teachers a deep understanding of the system, in order for teachers to be able to present salient concepts to their students. She conceived of Topobo as a "computer" or "technology" system with which children could play with computer-related concepts. Mary sees Topobo as a technology to play with ideas similar to educational-technology work like Logo or LEGO Mindstorms.

This indicates that tangibles may make certain common technology concepts accessible to children at younger ages than non-tangible technologies, as argued by Frei [Frei 2000]. However, in failing to identify concepts from biology which her students pursued in building creatures and investigating walking motions, Mary illustrates that preexisting conceptions of technology education can limit an educator's perspective on what technology is actually capable of teaching. If this is true, researchers in educational technology should focus on broadening the scope of themes that technology is "supposed" to teach.

### **Urban Science Museum**

Sets of Topobo were loaned to a large urban science museum for four months. Topobo had been displayed at many

exhibitions in the past but the interactions with visitors were generally very short and while the exhibitions may have been themed in areas such as innovation in play or robotics, no framework had been built around Topobo to guide its pedagogical context. Thus, sets of Topobo were turned over to teams of exhibit developers and programmers to find out how, or if, Topobo could be incorporated into their development process or inspire new experiences in informal education. Use of Topobo would be voluntary, based on interest in the system. Much internal discussion and two different scenarios incorporating Topobo on the museum floor emerged over a period of five months.

#### *Topobo in 'Design Challenges'*

The first group to work with Topobo was the development team for 'Design Challenges,' a program which features drop-in activities on the museum floor, staffed for 2 hours everyday, looking to provide "gender neutral non traditional engineering experiences." During the activities, children would build with provided materials to accomplish an engineering goal. The museum staff were present as guides but the focus was on allowing the children to engineer the projects on their own. The activities were planned for children aged 9-15. However, with the varying nature of museum visitors, a much wider range of children and adults participated. The team, led by Leah, took Topobo out on the museum floor for four sessions over a period of 2 months. The activity around Topobo was relatively unstructured but focused on making creatures walk, or if that was too difficult in the time frame, making them wave. She noted that visitors played with it for an average of

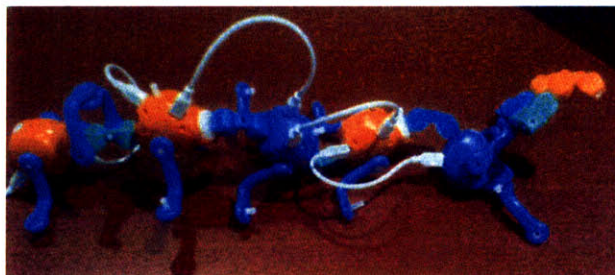


Fig. 6-6 A 'space caterpillar' built by a visitor and volunteer at the science museum's 'Computer Place' exhibit

20 minutes, considered a very long time for a museum floor experience.

When discussing the concept of the Topobo design challenge, Leah described what they had been investigating as biomimicry, attempting to make a connection to how animals walked. But she stated ‘I don’t think we went into it thinking that there was a science concept that we wanted to get across.’ She described their initial aim as showing people a new technology that they wouldn’t get to experience somewhere else, citing Topobo’s novelty as a big draw for museum visitors. The process of designing a ‘design challenge’ involved brainstorming a concept, prototyping solutions and narrowing the appropriate materials to make available, leaving the experience open enough to make four or five things that are totally different but can still accomplish the same goal.

If she were to design a deeper experience for a Topobo Design Challenge, she found the nature of Topobo as a well designed ‘kit’ to be a limitation, because the limited range of pieces could make it hard for students to arrive at diverse solutions. It had not occurred to her to mix Topobo with various other materials (cloth, LEGO®, etc.) as it seemed to go against the nature of the how the system ‘should’ be used. When asked if providing Topobo Actives that had the appearance of a raw motor, she thought ‘it would feel like a material, a raw craft experience as opposed to a kit.’ While the ‘construction kit’ might be seen here as a limitation, the attractiveness and completeness of Topobo’s design also drew in a wider age group than their usual audience, especially younger children. They were not accustomed to running a design challenge that spanned such a wide age range.

#### *Topobo in ‘Computer Place’*

Topobo was also incorporated into a staffed exhibit entitled ‘Computer Place’ whose goal was to introduce visitors to new computer technologies and present emerging computational concepts. Recently they had been moving into demonstrating robotics technologies, as this was seen as an emerging area in computation. Sonia, one of the program coordinators, brought Topobo into Computer Place for a week of continuous use.

She and other staff would demonstrate Topobo and then allow visitors to build creations of their own. To visitors, she described the activity with Topobo as biomimicry, with the goal of “making a computer act more like an animal.” In referencing Topobo, she also discussed concepts in computing such as programming (Topobo programming occurred with the body instead of code), networking, and swarm behavior, based on visitors’ varying interest and engagement.

Sonia’s relationship with Topobo focused on its identity as an emerging technology. Based on her area within the museum, the concept of teaching people about creating locomotion and biomimicry was an engaging experience which functioned as a stepping stone to draw people into a second and perhaps more fundamental goal of demystifying and teaching people about technology. Sonia thought it would be good to take Topobo apart, to show people what the sensors and motors look like, citing that they had a Robosapien® that was deconstructed and was very popular and engaging for visitors. As others had indirectly done, Sonia was directly tapping into the novelty of the system as one of its educational values. While this was clearly unintended in Topobo’s design, it an interesting paradigm for researchers to consider how Topobo’s identity will change as it (and perhaps robotics in general) transition into more commonplace technologies.

#### *Discussion*

In these two scenarios, and throughout conversations with other developers in the museum, it was evident that Topobo’s novelty and ‘cool’ design was a big attraction in a busy space with many experiences vying for attention. But to make a system like Topobo successful in the context of the museum floor, it becomes necessary to constrain it. For tangibles to contribute to the museum experience, one guideline is to create an experience that is constrained enough so people can absorb an idea in under two minutes, and open-ended enough so that people can make the discovery for themselves. One approach may be to appropriate the Exploratorium model of exhibit design in which an idea is made accessible by providing many different exhibits that all isolate and provide a



different way to “discover” an idea.

### **Overall Findings**

In all contexts - museum, classroom, after-school center, robotics center, graduate school - Topobo was regarded as a useful or provocative tool by the educators who worked with it. However, as a construction kit it seemed to excel in contexts that allowed for longer periods of engagement. Jane used it more as a play toy than a curriculum material. The museum asked to use it again, but in the context of a day-long activity. (They would like to use it in computer place, but in a more limited context, e.g. pre-built or somehow constrained in use.) Students and teachers in the after-school robotics center, who have more time to play with the technology, continue to work with it with success.

#### *Designing for multiple environments: Time and Age*

The idea of constructive learning or self-discovery came through in every context. As an open ended system the level of success with different age groups was directly determined by (a) the amount of time children spent with the system and to some degree (b) age. The longer kids may play with it, the younger they can be. When Mary used Topobo as a completely open-ended system, kindergartners (previously considered too young for such a complex system) engaged with it meaningfully if given enough time. Conversely, in the science museum, Topobo was used as a simple demonstration or inspirational piece (not at all an open-ended interaction with Topobo) with all ages, but visitors had only one or two minutes to engage with an idea. Somewhere in between we find Jane’s example of providing her students with pre-built models, so that they might constrain their efforts on programming motion. Universally, less time to interact with the system required it to be more constrained in scope. We believe central issues in designing interfaces or toolkits for multiple audiences will be for the designers to provide means for users to adapt the interface to scenarios of varying time scales, and potentially to different levels of complexity for different aged users.



### *Support for Educators*

Perhaps the most consistent and salient message from educators themselves is that educators need prior experience with the system, to gain confidence in their ability to teach with it. Jane is a teacher who put a lot of time and effort into learning the system and developing a lesson plan so that she could confidently communicate and teach new ideas to her students. In contrast, Dale jumped right into a lot of exciting, but difficult concepts and ended up frustrating himself and his students. Clearly all teachers needed support, and creating one's own lessons is too difficult for teachers to improvise.

Educators all requested similar kinds of support: to be taught examples they could use in their teaching, but they must learn the underlying principles of the examples. Here, the format of the examples was not prescribed, but printed materials in the form of an instruction / activity book may have met many educators' needs. Such a booklet might be similar between a teacher's standard activity guide, but the computational aspect of tangibles requires a level of systems-thinking that is not often specified in teaching with static materials. Certain challenges will arise, such as representing dynamic information (like movement) using a static printed page. Perhaps the booklet would have a companion on-line component of animated examples.

### *Inspiring the Use of Toolkits*

Many researchers like to develop "toolkits" that can be appropriated by teachers or students in a variety of ways. This contrasts with an interface designed to make a specific idea or application salient. For toolkits like Topobo, it seems especially important to provide educators with an inspirational example of an application scenario. Nearly everyone in our study was interested in making small robotic animals walk, and this provided both an emotional and a pedagogical "hook" to get people started thinking about and working with the system.

The inspirational scenario did not confine the range of ideas people explored with Topobo. Sonia and Mary saw Topobo as an entry to more general computing concepts like networking

and communications; Jane compared the system to materials like straws and paperclips (suggesting a general view of it as a material rather than an application); Ray actually used it as a prototyping material in a unique context; Dale envisioned learning conic sections and logic with the system. These digressions from the inspirational example of walking robots encourage us that toolkits can be reappropriated (which allows a user to get more out of their investment in the tools), but we believe the inspirational example application (walking robots) was critical to engage people's interest in the first place.

Dale's conceptions of investigating DNA, parabolas and logic principles suggest that educators are seeking the things that tangibles are already working toward: a more transparent programming and control structure, the ability to physically play with math and science ideas, and putting in people's hands the dynamic simulations that are increasingly an important part of scientific teaching. Mary's observation that transparency allowed collaborative work further supports teachers' goals in constructivist education. In terms of this transparency, accessibility and ability to model dynamic processes, the tangibles paradigm seems an obvious fit to education.

Topobo's highly refined physical design helped it succeed with a broad range of educators in such a hands-off manner because the parts were robust, reliable and approachable. However, the novelty of the system has both pros and cons: on one hand, its uniqueness invited people to explore and play with Topobo, catching people's attention in competitive environments like the science museum. But on the other hand, it is equally valuable to make tangibles seem "familiar" by referencing existing products and interactions. Familiarity allows the researcher to more quickly test the reactions and interactions of a seasoned user.

### **Exhibitions, Courses and Workshops**

In addition to our extended outreach, we have also conducted a series of public exhibitions, workshops and courses with Topobo to gauge children's, teachers' and researchers'

reactions and interpretations of the system. Although not intended to be formal evaluations, the results and impressions formed from these interactions, helped inform the development of Kinetic Sketch-up and Bosu as extensions of the concept of programmable kinetic behavior to new domains and paradigms. We participated in the following workshops, exhibitions, and courses with users totalling over 100,000:

2007: Kitakyushu Innovation Gallery (Japan); SIGGRAPH Educators Forum (San Diego, CA); Future Film Festival (Bologna, Italy); Robots at Play Festival (Odense, Denmark)

2006: MIT Museum (Cambridge, MA); SIGGRAPH E-tech (Boston, MA) ArtSpace Gallery, Threads Exhibition (New Haven, CT)

2005: ZKM Art and Media Center(Karlsruhe, Germany); Universite der Kunst (Berlin, Germany); Collaborative Artifacts Interactive Furniture Workshop (Switzerland)

2004: ARS Electronica Museum (Linz, Austria); SIGGRAPH E-tech & educator Forum (Boston & San Diego) Cooper-Hewitt Design Museum (New York); Wired Nextfest (San Francisco); Artbots (New York), Ivrea Interaction Design Institute (Ivrea, Italy)



## *Chapter 7*

### **Kinetic Sketchup: Evolving Topobo into New Design Realms**

As part of the extended outreach with Topobo (as discussed in Chapter 6), Topobo was introduced in the kinetic architecture course of Prof. Kostas Terzidis at the Harvard Graduate School of Design. My experiences of working with Topobo in an architectural design setting directly launched the development of Kinetic Sketchup. Topobo was originally designed to give children an intuitive physical experience learning dynamic physics concepts, such as torque or leverage, and allow them to deconstruct the complexities of creating walking creatures. However, the idea of 'kinetic memory' speaks to a larger interaction concept, What is it like to sculpt with motion?

For designers, computational control give rise to new possibilities in actuated products and transformable environments but this transition also produces a new range of design problems- how do we visualize, imagine, and design the physical processes of transformation? What are the tools for intuitive motion investigation to train and develop our motion sensibilities in 3D space, towards an interface that makes sketching with motion as easy as drawing with paper and pen? With its single button interface and gestural

recording capabilities, the Topobo Active begins to embody this concept of a motion primitive, a fundamental unit of kinetic memory. In this realm, Topobo transitions from a tangible interface for physics education with children into the broader applications of a design tool for motion exploration, expression and communication.

During an initial studio session, student designers experimented with Topobo in an open-ended fashion. As part of the course, students were using the Arduino [Arduino] programming environment to control sensors and actuators, so they were accustomed to the idea of embedding kinetic behavior physically into their models. However, these students were more comfortable working with physical materials like foam core or paper than with embedded technology. Topobo thus became part of their hands-on modelling and design processes to quickly and easily experiment with movement in their models. The system was presented as a “material” for students to prototype motion concepts in designs of transformable and deployable structures, reappropriating Topobo into a new design context.

Following the experience of the class session, one student, Ritchie, continued working with Topobo over the following six months, utilizing it in the design stages of his Master’s thesis project. Ritchie’s thesis work involved the design of a conceptual transformable opera house [Fig 7-1] set on Potsdamerplatz in Berlin. The building morphs between

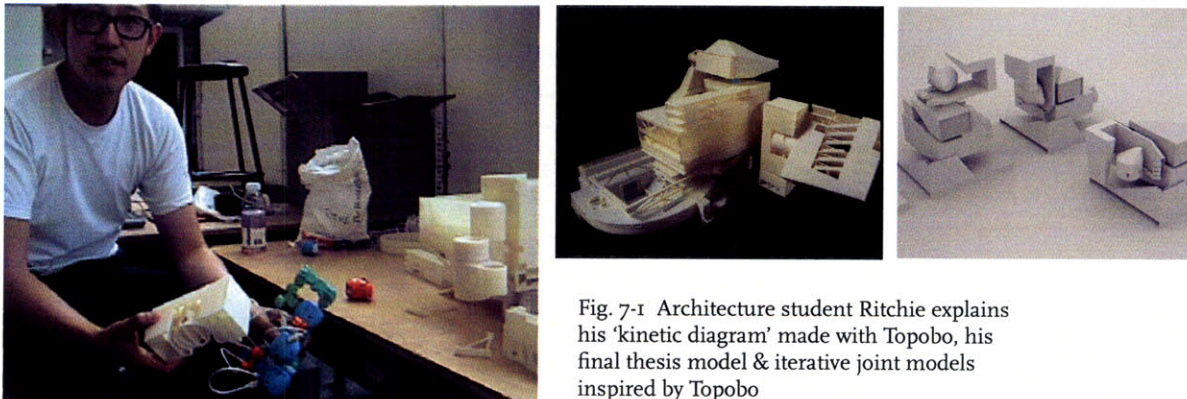


Fig. 7-1 Architecture student Ritchie explains his 'kinetic diagram' made with Topobo, his final thesis model & iterative joint models inspired by Topobo

two physical states, representing two alternate realities: one represents its form in the 1980's before the Berlin wall fell, and the second fictional state represents the building as imagined if the Germans had won WWII.

Ritchie used Topobo as a kinetic prototyping tool as part of the initial design phases for the project. He describes his process: "The most important part for Topobo for me architecturally has been toward the use of diagrams. This model is a representation of some of the kinetic movements in the final project...I used it very early on in the project but as my building started becoming more spatial [modeled in detail & scale] the use for Topobo was eliminated. In the very first stage of a project,...Topobo was instantly these modular parts which I could bring into a kinetic state for discussion."

Ritchie used Topobo as one medium among many in which he communicated his design, with the most useful part for Topobo being early on in the research, "getting my kinetic idea across." When discussing the limitations of Topobo and why he had not continued to use it further along in his design process, Ray cited that he felt constrained by form factor, specifically the joints being a single degree of freedom which made his kinetic model bulky and spatially more complex as he had to offset each joint. As he continued with his design, however, he cited one wing of the building's mechanical design being directly inspired by this constraint, "[this area of joints] came about when I had to keep offsetting the Topobo and I noticed that the axis of rotation could be elongated." [fig. 7-1]. What began as a limitation became part of his design language.

Topobo did not become part of Ritchie's more detailed design phases. While we had given him permission to modify the parts and embed them into his model, Ray preferred to begin 3D modeling in a GUI as the next phase of his design process, "Physically I could take it apart and try to build a chip board model around it but that isn't the method I usually work in, I usually go straight to the computer, draw it in 3D, send the file to the 3D printer. It's just faster." He also mentioned the issue of Topobo as a 'polished kit' not offering



him the flexibility of a raw material to blend into his model. Behavioral control issues also arose, he used Topobo often to discuss and explain the ideas in his models during critiques. For this, he needed the ability to advance, reverse or replay his motion easily, and on command while speaking, like stepping through the motion composition of his model.

The advantages of the physicality and immediate access to kinetic behavior had now been outweighed with a more detailed oriented and familiar tool, 3D modeling. However, the Topobo models Ritchie had made directly influenced many joineries in the final model. He found it useful to think about the design modularly, like Topobo, designing in segments and then connecting them with Lego-like attachments. It helped to work with a physical kinetic material first, when thinking about what would work mechanically in space before attempting to draw it on screen. The building took on a very toylike playful aspect to it, rare in architecture, which he felt may have come from his interactions with Topobo. Ritchie also used Topobo in one unexpected way, mapping the colors of the passives in different areas of his model to denote their spatial functionality, he described it as his 'legend.' The color mapping that began with Topobo continued into his 3D onscreen model to become part of the design language in communicating the project.

My experience with Ritchie directly inspired the development of Kinetic Sketch-up, a series of physically programmable modules for quickly expressing and prototyping the visual language of motion in 3 dimensions. The Topobo Active takes a first step at creating a malleable kinetic prototyping and sketching tool and showed promise as useful in the architectural context. Topobo's gestural recording capabilities allow designers the ability to quickly construct motions based entirely on physical intuition. However, in interactions with Ritchie, it became clear that more variety in mechanical, material and behavioral controls were needed to offer him the flexibility and inspiration he needed for his design. Expanding the form and capabilities of the system for use in a wider variety of interactions offers the possibility of a extending the kinetic prototyping space beyond the

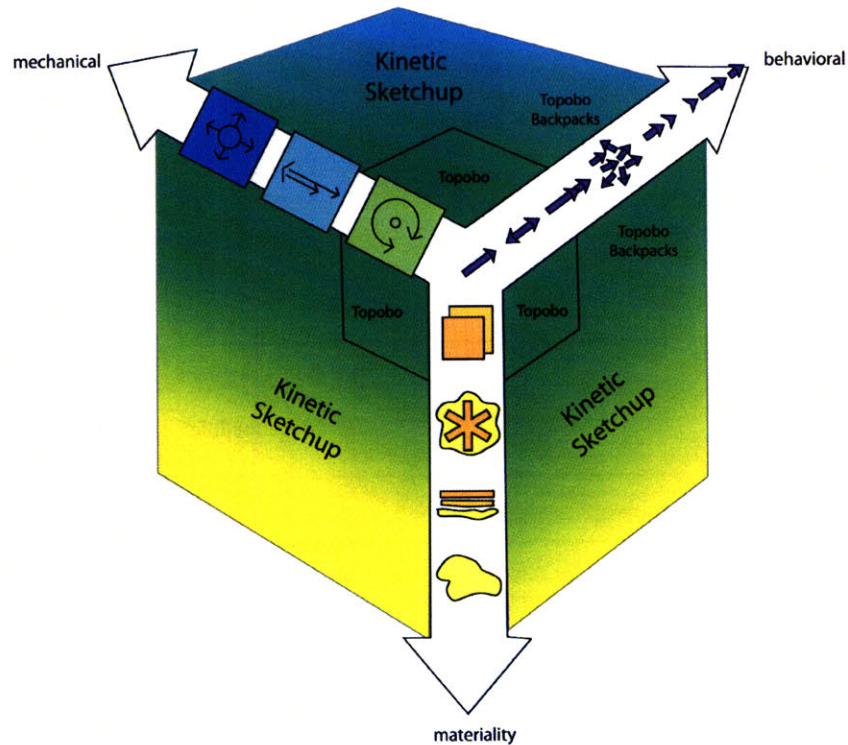


Fig. 7-2 The Kinetic Prototyping Space: Kinetic Sketchup expands beyond the capabilities of the Topobo Active in differing mechanics, form, materiality and behavioral control

limitations of Topobo [Fig 7-2] into a new range of physical attributes for quickly expressing and prototyping the language of motion in three dimensions.

After my initial interactions with Ritchie proved promising, I began a semester long independent study to establish an iterative development process working in tandem with MIT graduate architecture students, John Pugh and Marika Kobel in order to establish appropriate needs, functionality, and affordances for their design process with transformability. During the independent study, John and Marika developed new concepts in transformable architecture and product design. Mimicking Ritchie's process with Topobo, John and Marika used in-progress prototype modules as one element

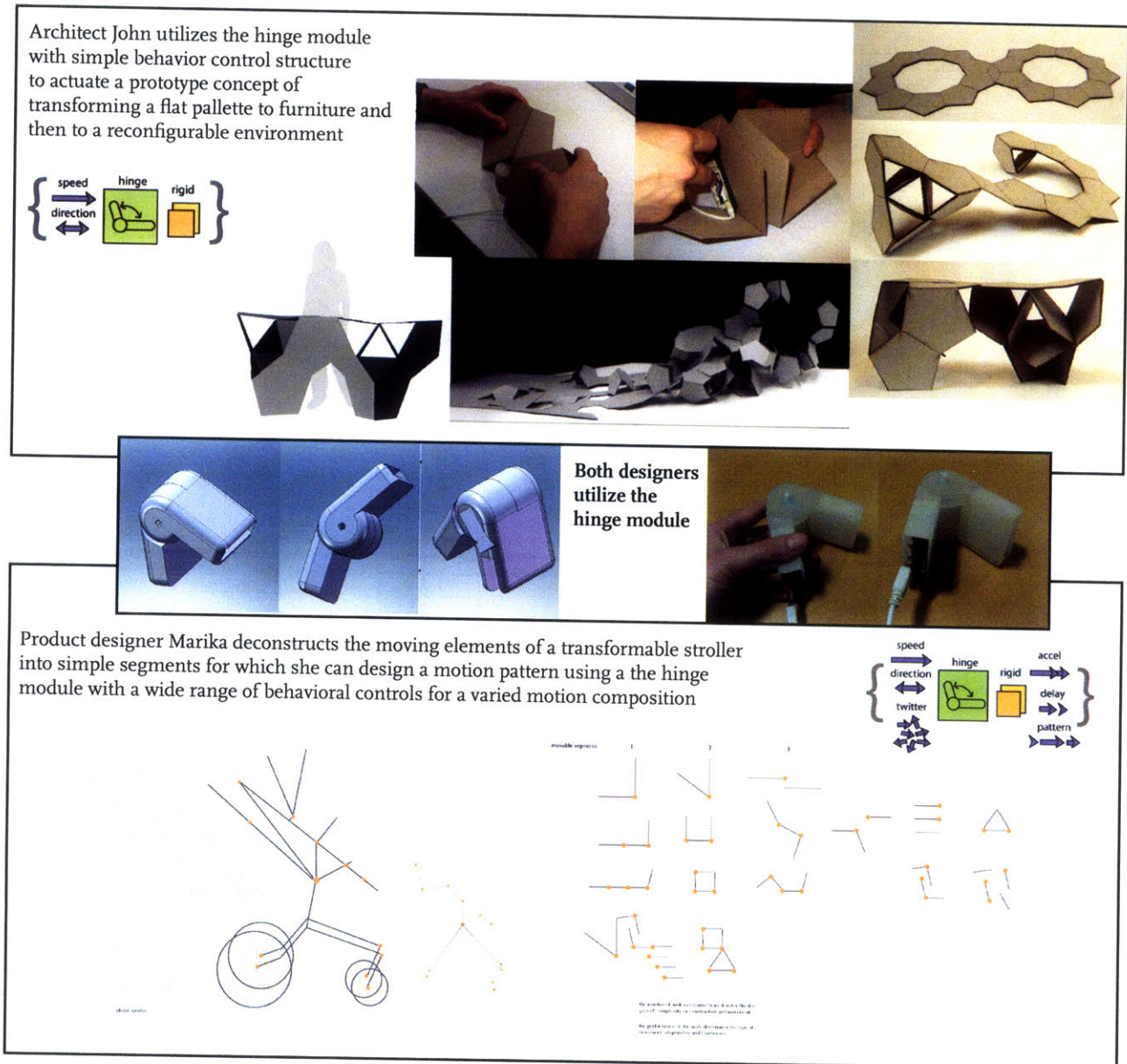


Fig. 7-3 Designer case studies with the hinge module

of many design diagrams and strategies to illustrate and communicate their ideas. Figure 7-3 shows design concepts and documentation of John and Marika early in the study, when they are working with a deconstructed Topobo Active, essentially functioning as a hinge, offering a single degree of freedom rotation. In this situation, the designers were constrained mechanically, resulting in their focus on two very different concepts. Marika worked on a stroller design where



she deconstructed the design into a series of axes in a two-dimensional plane and investigated different possibilities for speed, reactive control and choreography of movement in the abstracted system John, however, focused on the complexity of transformations in a physical geometry when converting flat two dimensional elements to three dimensional structures.

Throughout the semester, I was looking to discover how the design process of the user is altered by use of a new physically programmable tool, in terms of functional characteristics, affordances, collaborative thinking, and the capacity for supporting a creative, expressive and inspirational user experience. In turn, this iterative process inspired, informed and evolved my own process of tool design. This offered what I found to be an appropriate approach to iterative evaluation for a system that should be constantly evolving and expanding as we looked to establish functionality and usability in a base line set of tools for an emerging design paradigm. The independent study allowed me to construct a working vocabulary and design language for designers interested in working with motion.

The final Kinetic Sketchup system consists of a series of physical modules representing a cross section of mechanical, material, and behavioral parameters, matched as logical

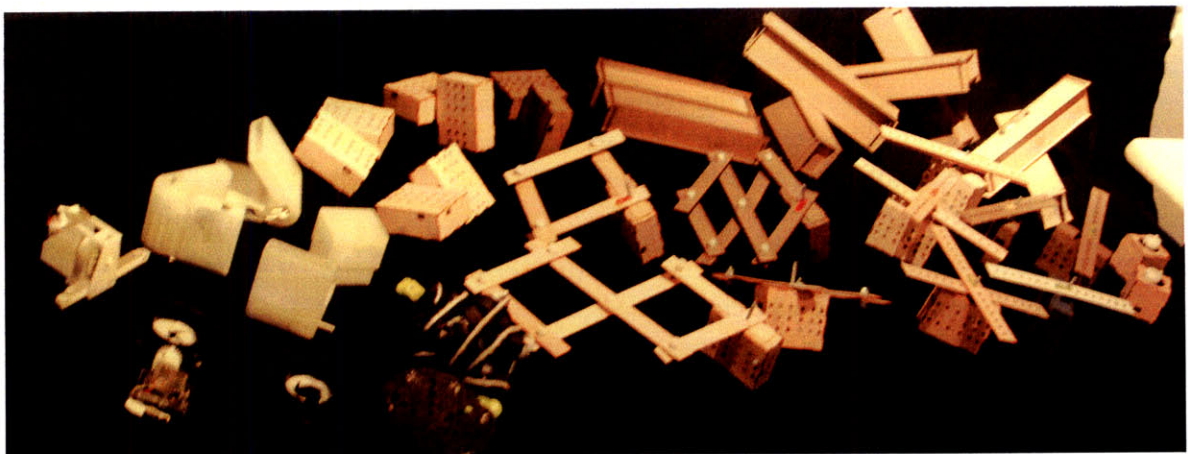


Fig. 7-4 A sampling of Kinetic Sketchup modules

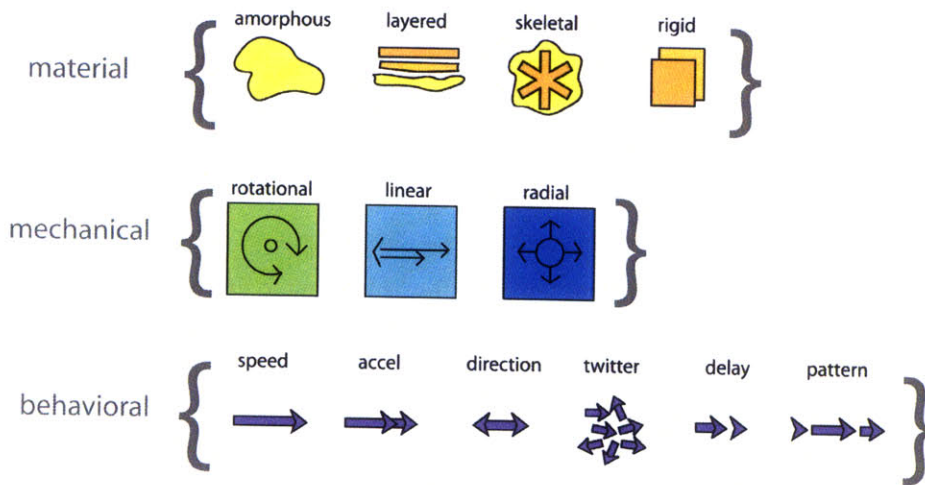


Fig. 7-5 Motion design parameters

pairings for use in transformable structures of varying scales and purposes. Figure 7-5 shows a deconstruction of the basic motion design parameters as first proposed in Chapter 4 and which I have incorporated into the Kinetic Sketchup modules. I view this deconstruction as a kind of road map guiding designers to the fundamentals and basic elements of kinetic design, but it is no way meant to be wholly inclusive of all possible ways to categorize observed or perceived motion.

### Motion design parameters

#### *Mechanical*

**Rotational** - motion moving around a central axis

**Linear** - motion expanding outwards along a straight path

**Radial** - motion expanding outward and inwards from a central point of a circular form

Each category of mechanical design, can be further dissected into a families of built elements, designed and named to conjure up familiar objects or scenarios with transformative properties as shown in fig 7-6. These families include elements with a singular actuator as well as structures formed through compounding multiple actuators. Rotational motion includes elements mimicking the motion of a hinge, a fan, or

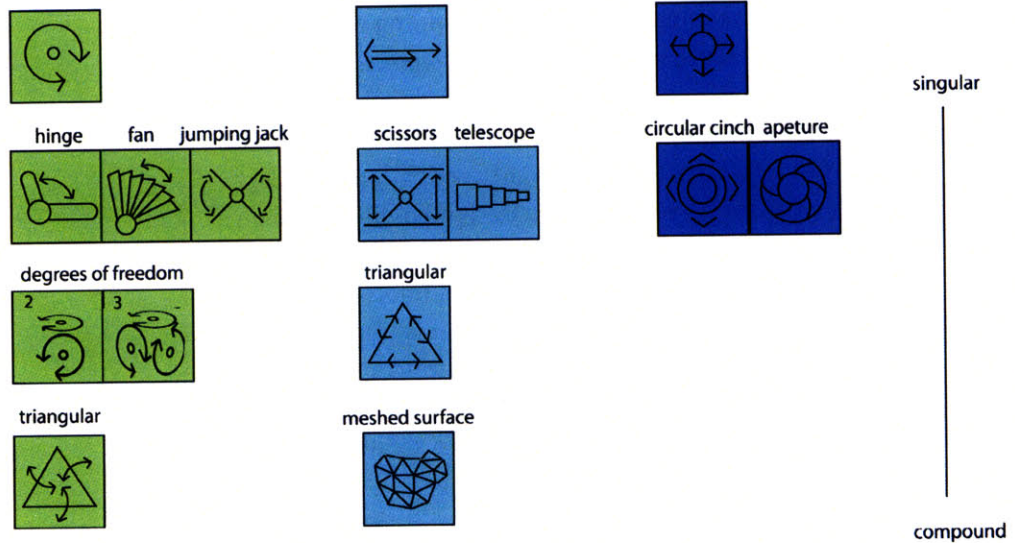


Fig. 7-6 Families of mechanical elements

a jumping jack, all created from a single actuator, as well as compound structures of a 2 degree of freedom joint (joystick) and 3 degree of joint (a shoulder). The final compound structure of the family is a 2D triangular structure which converts to a 3D pyramid structure through use of 3 actuators. The linear family of elements includes a telescoping box structure, and an expanding scissor mechanism each created from a single actuator as well as a triangle with shifting side lengths (3 actuators) and a meshed surface made up of multiple triangular elements which will be further discussed in chapter 8, as part of the Bosu system. The radial family includes an aperture mechanism, which actuates to open and close creating a hole in the center of a static outside frame and a circular cinch mechanism which expands and contracts from a central axis, like an umbrella.

The designed mechanisms are intended to create a representative range of possibilities offering familiarity with common objects or phenomena displaying kinetic behavior (our bodies, nature and environment), while also remaining abstract enough to inspire open design thinking in scale and genre of transformation. While many of the elements can be

created by changing, utilizing or combining with others, we have chosen to dissect them out as *perceptually* different. This is important to note as we are creating a tool to stimulate design thinking in a domain that is potentially unfamiliar. For example, presenting a designer with a rigid rotating shaft conjures up very different design possibilities than a telescoping pole or a collapsing fan, although technically they could all be created from the first, with informed mechanical design. As Braitenberg suggested [x], it is important to isolate the perceptual response a motion can trigger, simplifying the mental leap and engineering knowledge necessary to arrive at a potentially new and innovative design solution.

#### *Material*

**Rigid** - solid structural material

**Layered** - mixture of rigid and flexible materials

**Skeletal** - structure of rigid interior with malleable exterior (like the skin, muscles and bones of our bodies)

**Amorphous** - entirely malleable

The Kinetic Sketchup modules were created using a range of materials including cardboard, wood, acrylic, ABS, polypropylene, rubber, foam, felt, twine and fabric. The range of materials and combinations was intended to give the system broader aesthetic and tactile bounds than Topobo had provided. During our extended outreach sessions, users in creative domains (architecture, exhibition design) had cited the limitation of Topobo as a 'polished kit,' prescribing a particular kind of usage, as opposed to feeling like a raw material. The challenge in the design of the Kinetic Sketchup modules was to make material, form and color choices which allowed for a feeling of open-endedness of interaction (the ability to combine easily and fluidly with many different kinds of materials), while shielding the users from the raw technology of the system (the PCB and servo) to avoid distraction or fear of the system as overly techy. This required a minimal, abstract aesthetic appropriate to the visual language of designers, ideally allowing the system to perceptually disappear within the design of a holistic construction and



aesthetic as created by user. The result is families of modules in varying materials from loose and temporary (cardboard) to more permanent (acrylic) allowing for multiple methods of attachment and combination (screws, glue, LEGO, pins, velcro etc) by the users. Modules which had a 'softer' quality, combining malleable and semi malleable materials, were implemented using the Bosu hardware infrastructure described in Chapter 8, where materiality is discussed more extensively and material composition becomes an important element in determining the interaction feedback loop.

#### *Behavioral*

**Speed** - basic velocity control

**Direction** - basic directional control

**Acceleration** - increase or decrease in velocity, can be cumulative with sequenced playback

**Twitter** - addition of 'noise' into the motion playback, adding a randomized variability to playback

**Delay** - creates an intentional pause in playback

**Pattern** - allowing a motion composition to be sequenced through during playback

These behavior controls were chosen based on much of our research done with the Backpacks in determining appropriate levels of complexity with tangible modulators for kinetic behavior [Raffle 2006]. Differing from the experience of using Topobo to create walking creatures, which implies a repeated looping structure of motion, the behavioral control in Kinetic Sketchup was designed to allow for a more sequenced approach to kinetic design, incorporating the concept of stato-dynamicism as introduced in Chapter 4, where objects transform from one state to another to change their functionality or structure but can embody their intended purpose in a static state. This provides the ability to experiment with and observe structures in different physically static states as well as use of the system playback tool to perform and communicate kinetic concepts to other designers.

## **Modes of Operation**

The Kinetic Sketchup Modules are programmed through the coincident single button interface. Additionally, the modules can be controlled remotely through an off board control knob and button for motion modulation during playback.

### *Direct Manipulation*

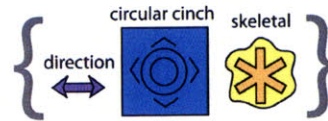
Like Topobo, each of the modules is initially programmed through gestural recording. A single click turns the light to red for recording the motion of the user, a second single click turns the light to green for playback, and a third single click pauses the motion. Additional playback functionality was programmed into the certain modules as a series of sequenced 'double clicks' based on the intended behavioral control. Double clicks on 'speed' behavior modules sequence through doubling the original rate of playback, halving the original rate, and returning to the original. Double clicks on the 'twitter' control modules incrementally increase the randomized variability in the playback through four states until returning to the original. As a stato-dynamic controller, the 'pattern' control modules utilize a series of single clicks to move the kinetic construction to its next state and then pauses, while a double click turns the module off.

### *Remote Control*

A second mode of control was programmed into the modules to allow users to observe the playback state of modules from a distance, without the intervention of the body, while still allowing for slight modulation of kinetic behavior. The design of an off board remote control emerged from experiences with the Topobo backpacks in which users at times expressed frustration that they could not directly and immediately observe the slight changes a Backpack was causing in their creation while they were still in contact with the creation. The remote control also offers a more finely tuned range of variability, instead of sequencing through a series of finite states, the knob on the remote control box offers continuous variability for control of speed, acceleration, twitter and delay. The pattern controller utilizes a remote with a single button, stepping through finite states of playback.

### Module Designs

The following examples represent different module design combinations with descriptions of possible usage scenarios. As a reference, each module can be described symbolically as a 'motion phrase.' For example, following is the phrase for a module of skeletal construction moving radially with simple directional control:



### Hinge with position control

In recording, users can program a sequence of finite states into the hinge which are played back sequentially with button presses. The hinge is intended to be attached to planar surfaces, resulting in motions such as opening a door, or be used to convert 2D flat structure into 3D. By combining many together, one user mocked up what he conceptualized as 'hyperbolic-paraboloid mobius strip' in which the actuated hinge joints between paraboloids allow the mobius strip to twist.

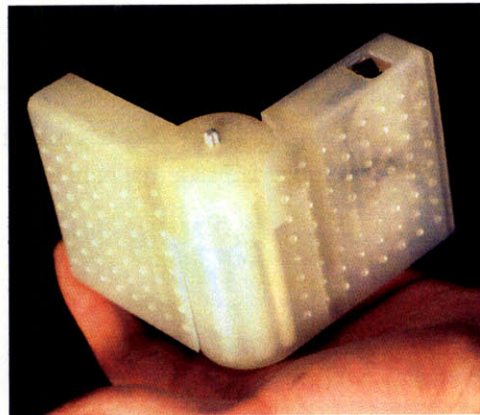
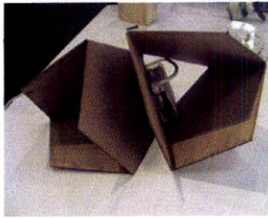
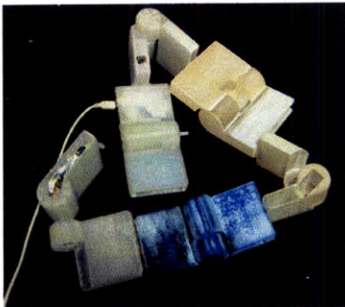
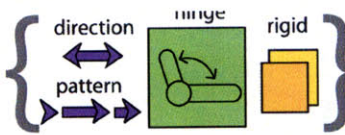
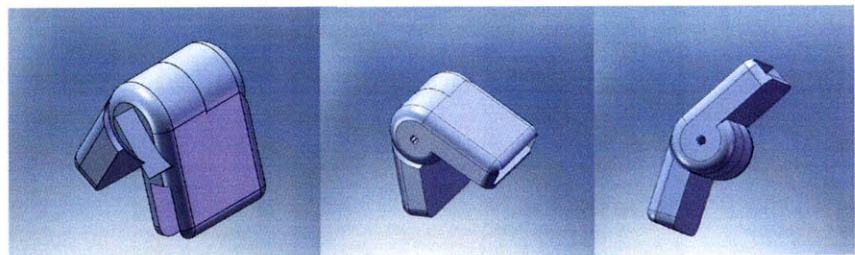


Fig. 7-7 Hinge module



### Linear Telescoping with speed control

This linear module slides in and out and is intended to prototype situations where extension or retraction of length is required, for example, in raising the height of a surface. It can be used horizontally or vertically.

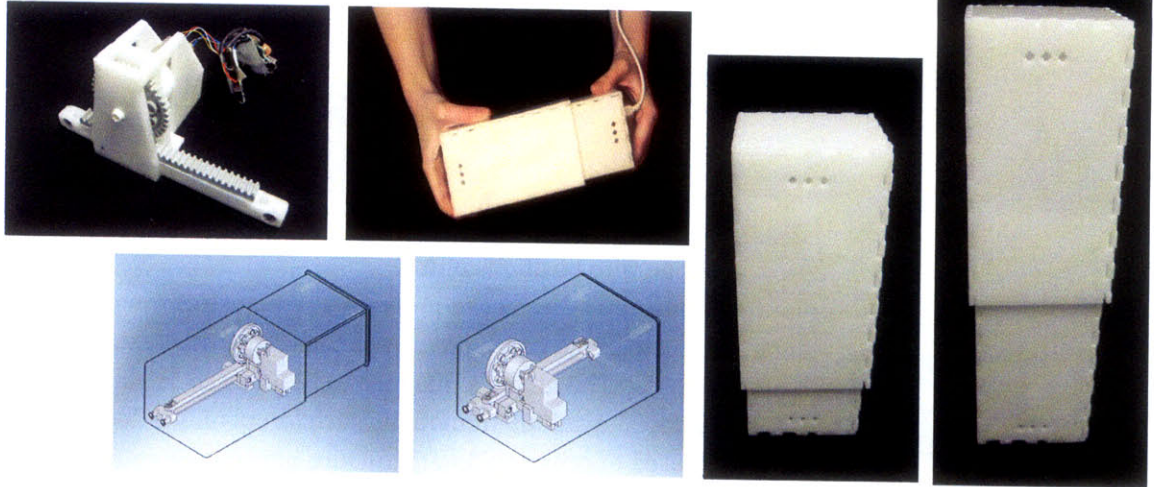
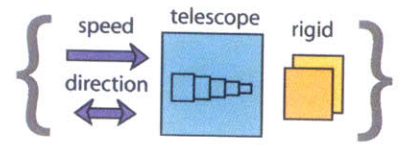


Fig. 7-8 Linear Module

### Scissors with speed control

Also offering a form of linear motion, the scissors are used to extend and contract in length. Different from the telescoping mechanism, however, the structure of the scissors changes inversely in length and width, meaning as it becomes longer, it narrows, while contracting, it adds width. This offers the potential to create structures that appear to have a conservation of surface area or volume, when embedded in a malleable form.

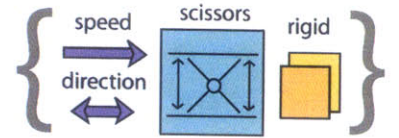
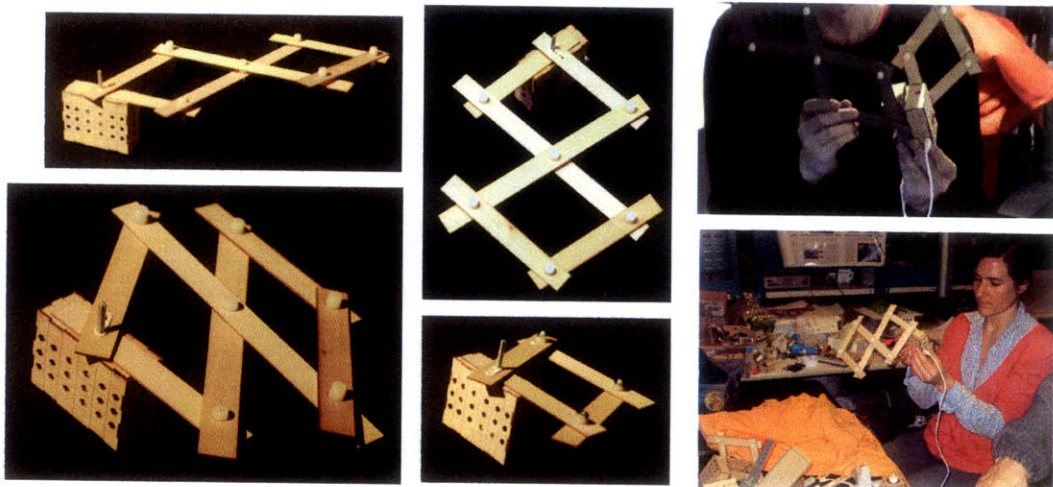


Fig 7-9 Scissors module





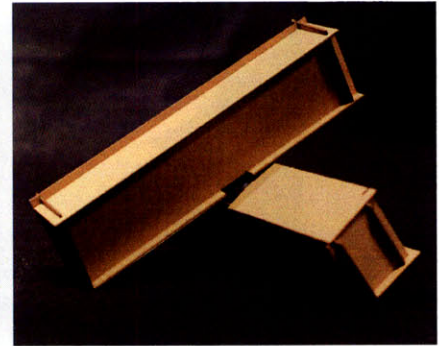
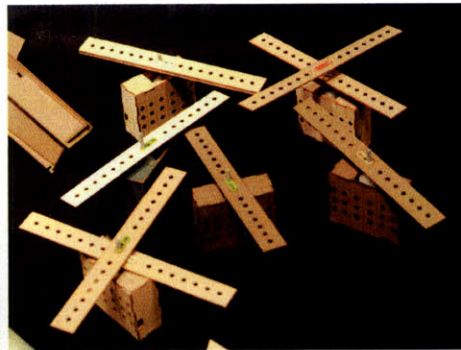
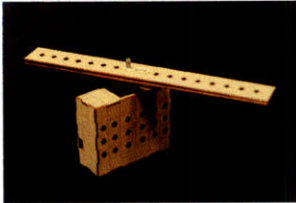
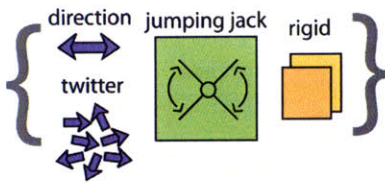


Fig. 7-10 Jumping Jack module

### Jumping Jack with twitter control

The jumping jack is intended to be used when linear elements move in unison but opposing directions around a centralized axis. The increased jitter in the behavior of this module can be easily observed by comparing the motion of linear elements as related to each other.

### Aperture with directional control

The aperture mechanism opens and closes in a radial motion to reveal and conceal area within a 2D surface. It is intended to be used to create dynamic surface perforations of varying sizes that can, for example, control light or air flow, such as in the wall design of the Institut du Monde Arabe in Paris [Fig 7-11].

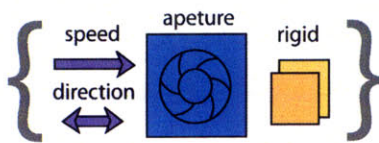
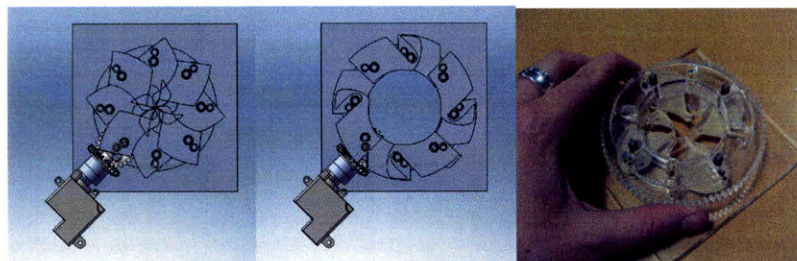
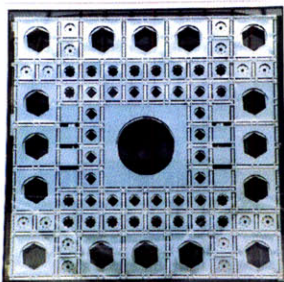
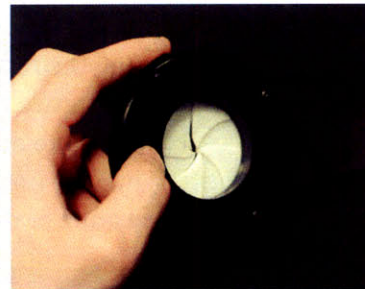


Fig. 7-11 Aperture module



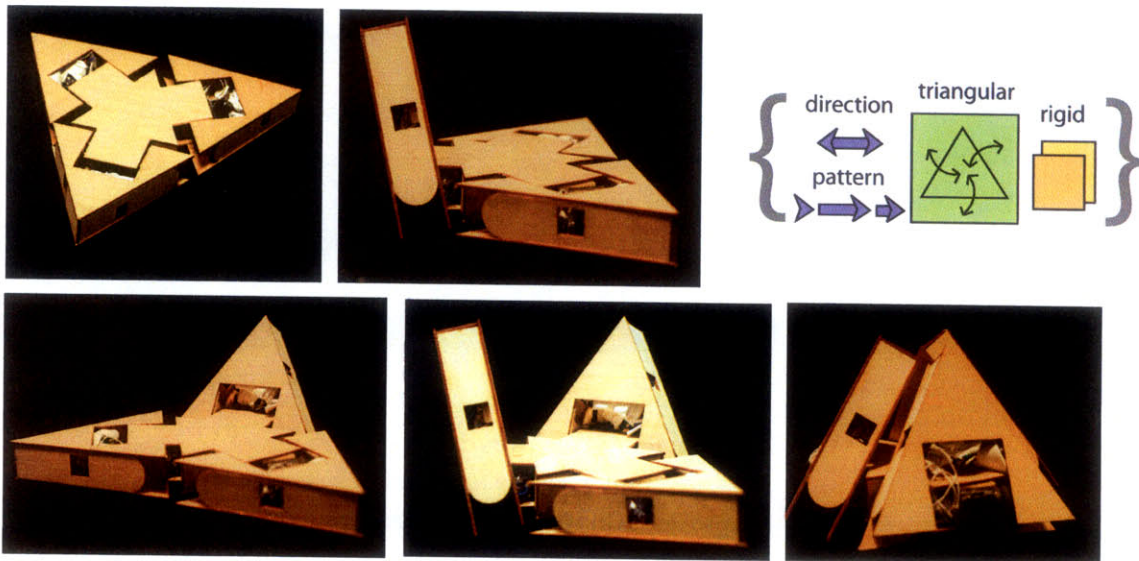


Fig. 7-12 Triangle Module

### Triangular with position control

This offers an example of a compound module which essentially combines 3 rotational hinges modules. Each side of the 2D triangle can fold up to form the sides of a pyramid in 3D space, with position control for in-between states. It is intended to allow for prototyping the morphing of transformable structures moving from 2D to 3D.

### 3 Degrees of Freedom in a skeletal structure

Also a compound module, this construction combines 3 rotational actuators into a structure mimicking a ball-and-socket joint offset in 3D space. The structure is embedded in a woven skeletal structure made of varying densities of laminated foam. While offering a support structure for the motors, the flexible nature of the foam offers a perceptual

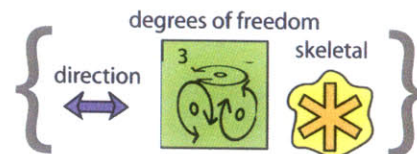
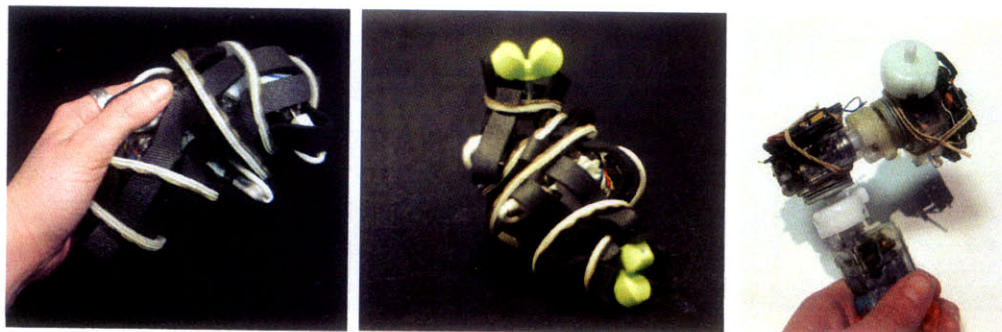


Fig. 7-13 3 DOF Module





shift for this module, where the mechanized motion of the actuators displays an organic quality to the user, like that of a human shoulder. Further discussion of modules utilizing skeletal and layered materiality and how materials can influence motion perception are discussed in Chapter 8.

### **Technical Design**

Technically, the modules use two different electro-mechanical infrastructures. The modules of rigid materiality, are built with the system used in Topobo Active, including a custom PCB, custom servo motor and gears (Fig xx), while in the modules of soft materials and soft mechanics, sensing and actuation occurs via bend sensors and shape memory alloy (nitinol) and a custom PCB which is described as Bosu in Chapter 8. Each module is also outfitted with locations to attach additional parts and materials as necessary in a model prototype (such as threaded holes for screws on the rigid modules or riveted holes or velcro on the soft modules).

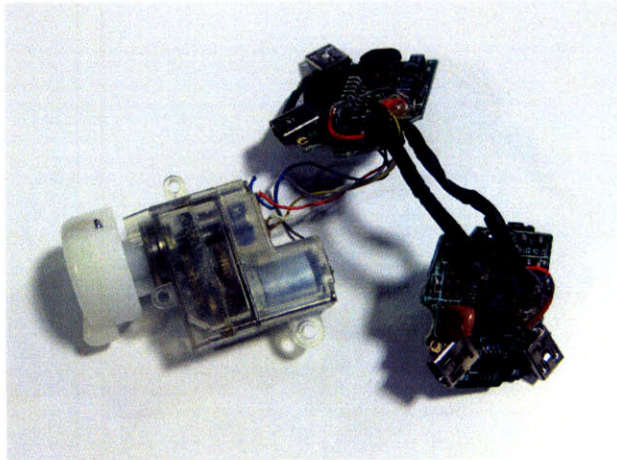


Fig. 7-14 Raw hardware (PCB + servo motor) for the Kinetic Sketchup elements





## *Chapter 8*

### **Organicism with a New Actuated Materiality: Glume, Senspectra and Bosu**

In designing the Kinetic Sketchup modules, materiality emerged as an important design parameter to open up a greater range of aesthetic, tactile and perceptual qualities of kinetic interaction. In interface design, we generally rely on the visual affordances of an object to determine behavior and functionality. However, material affordances can connote a variety of qualities that are the source of rich sensory experiences and occasion for numerous action modalities. From a tactile perspective, the static quality of rigid objects affords unary or binary controls. Hard objects are simply touched or pressed in a singular fashion, while malleable objects have a compliant material quality that invites users to multiple levels of tactile exploration and control. The physical act of deforming a malleable tangible interface can be mapped to a continuum of meanings.

For objects in motion, a very significant perceptual shift can occur with a change in material – a jerky disjointed motion of a series of mechanical motors can be embedded in a soft padded exterior and the quality of motion can be inverted to a smooth oscillation. Materials also play an important role in kinetic behavior when considering how objects are subject to the natural environmental forces of gravity and friction.

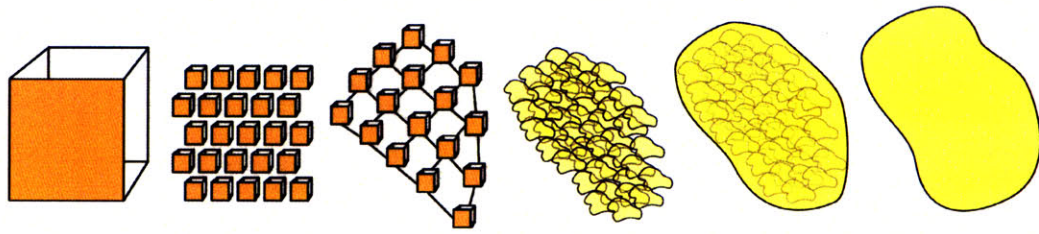


Fig 8-1 Material Transformation - from discrete to continuous

All materials are in essence created of discrete components which combine to create continuous systems. The importance of modularity in creating a material language is evident in our environment from the microscopic level, such as building blocks of biological systems (cells) or chemistry (atoms) to the architectural level, such as refabricated panels. Modularity leads to systems of physical primitives, grammars forming the basis of constructive assembly systems. Our experience with the tactile qualities of a material, are related to the scale of modularity at which we experience it. For example, fine particles of sand, while rigid in nature individually, feel soft to the touch when combined into a large pile and sift through our fingertips. Figure 8-1 demonstrates the shifting scale of material perception, from rigid and discrete to amorphous and continuous.

Part of the perceptual shift gained from materiality relates to the context of abstracted anthropomorphism, or how closely the material quality of an interface relates to human qualities of the body. Our bodies offer a unique structural framework on which to base a kinetic construction- a rigid skeleton of bones with an expected range of motion from our joints, surrounded by muscle, tissue and skin, each of varying densities offering infinite variability in form from one individual to another. The motion of our bodies is interpreted through the surrounding soft tissues, appearing organic in nature, where repeated motions are never exactly the same, a seeming complexity imbued with the familiarity of experience.

I begin this chapter by discussing Glume and Senspectra, two prior projects designed with Vincent LeClerc in the Tangible Media Group, which were conceived as experimental systems to incorporate the organic material natures of our bodies into computational interfaces for form-finding. While static in nature, these projects formed the conceptual basis of the Bosu system. They also mark a shift in thinking for incorporating new material concepts into the interaction loop for TUIs, for which we determined a set of design objectives:

Use texture, plasticity and elasticity to inform the user about functionality of a physical interface

Map the material affordances of a tangible interface to inform and manipulate its control structure

Incorporate the malleable property of a material meaningfully into a distributed digital manipulative.

### **Glume**

Glume [Parkes 06] [fig.8-2], is a modular scalable building system with the physical immediacy of a soft and malleable material. The Glume system consists of soft and translucent augmented modules – six silicone bulbs, embedded with sculptable gel and a full spectrum LED – attached to a central processing “nucleus” [fig. 8-3]. The modules communicate capacitively to their neighbors, via the novel conductive characteristic of hair gel, to determine a network topology and are responsive to human touch. Glume explores a unique area of augmented building materials by combining a discrete internal structure with a soft and organic material quality to relax the rigidity of structure and form in previous tangible building block approaches.

The Glume system was designed to retain the tactile experience of a soft modeling material, while creating a new identity and extended functionality for the material. To achieve this goal we established guidelines for the physical and digital design of the system:

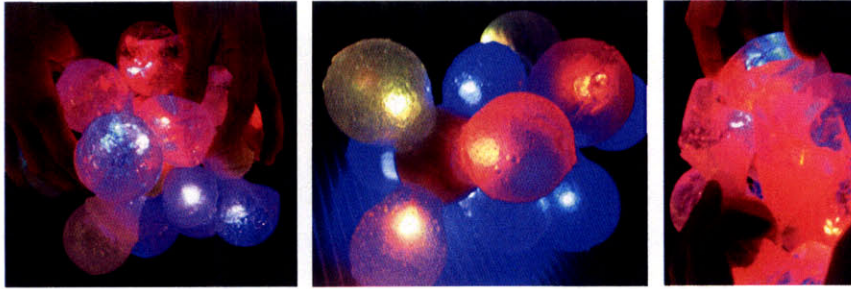


Fig. 8-2 The Glume system in use

Retain a flexible malleable form while incorporating a regular recognizable stacking geometry

Induce a tactile sensation similar to sculpting with a soft moist material

Provide translucency to see inside a model

Allow for distributed or centralized functionality

Incorporate a touch response as feedback to the user

To construct a model, users combine Glume modules, interlocking and shaping the nodes into place. As the user builds, the system determines the model's morphology, first defining an origin point at the base of the model and recursively looking at the neighbors of the 'base' node to define a crude morphology which is then optimized into a 3D mapping of the structure. Once the model is constructed, users can associate it with a predefined semantic model of a specific volumetric map related to the construction. In a hydrologic assessment situation, users construct the 3D geologic map of a terrain while taking into account important physical parameters for the simulation such as soil characteristics and land surface slopes.

To manipulate a model, users modify the parameters of a node or a group of nodes by introducing an object modifier into the system. Object modifiers can affect a single node directly during the building process, and its color is adjusted to reflect the change, mapping different properties (such as types or density of particulates) to different colors. If object



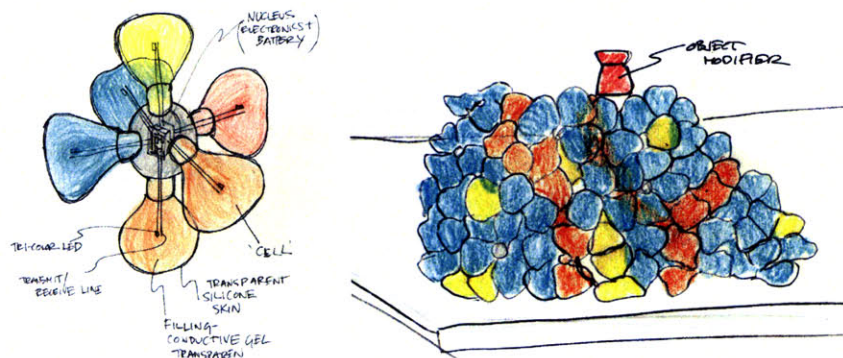


Fig. 8-3 Glume concept sketches

modifiers are added to an existing model, the system will regenerate its semantic model to reflect the new parameters. In a hydrology model, users could simulate the propagation of a pollutant plume and visualize its effects on the geological map, by placing several object modifiers representing pollutant sites on the surface and simulate the pollutants propagation over time.

#### *Technical Implementation*

An individual Glume module consists of six silicone bulbs connected to a central 'nucleus' containing a custom PCB and a 3.8v 1.5mA lithium polymer battery [fig 8-4]. The silicone skin of each bulb has been cast in a translucent silicone rubber. The hollow castings were made from molds modeled in Autocad and then 'printed' using a 3D starch printer. The bulbs are embedded with Softee® Protein Styling Hair gel chosen for its optical clarity and conductive characteristics. The combination of the thin silicone shell and the embedded gel provides the tactile effect that each bulb will retain the shape as sculpted in place by the user.

We devised the system architecture in three distinct modules: a PWM display adapter and six RGB LEDs; an FSK modulator-demodulator and six gel electrodes; and an 8-bit RISC micro-controlling unit. The Glume system is driven by an ATMEGA32L AVR® microcontroller running at 8MHz. The FSK module uses the PWM unit provided by the AVR®



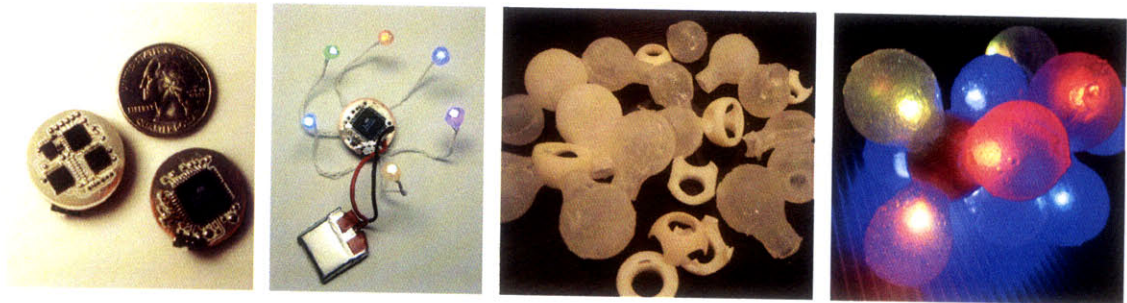


Fig. 8-4 Glume PCBs, unassembled and assembled modules

to modulate two different frequencies into a multiplexer that redirects the signal to one of the six electrodes.

#### *Issues and Limitations*

Based on our interacting with our initial prototype, we observed the main issue of the system to be resolution and accuracy in determining the structural form. Glume works best to simulate the overall global behavior of a model, or as a close up of a particular region. Although each module features a rigid ‘nucleus’ we found the bulbs to be too malleable, not offering enough structure. The hair gel worked to sense the relative position of modules, but when many modules were combined and the bulbs were overly deformed, it was difficult to determine the overall structure as a regular repeated geometry.

#### **Senspectra**

Based on our interactions with the Glume system, we designed a second interface which addressed many of the issues surrounding Glume’s overly malleable materiality. Senspectra [LeClerc 07] [fig. 8-5] is a computationally augmented physical modeling toolkit designed for sensing and visualization of structural strain. The system functions as a decentralized sensor network consisting of nodes, embedded with computational capabilities and a full spectrum LED, which communicate to neighbor nodes to determine a network topology through a system of flexible joints. Each joint uses a simple optical occlusion technique as an omnidirectional bend sensing mechanism to sense and

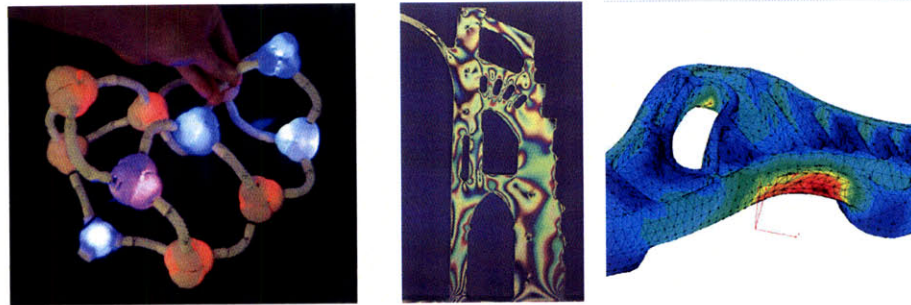


Fig. 8-5 a Senspectra construction, photoelastic Analysis of Chartes Cathedral and Finite Element Analysis on the handle of pliers

communicate mechanical strain between neighboring nodes, while also serving as a data and power bus between nodes. It functions as a tangible interface that utilizes the unique material qualities of its elements as part of a control-feedback loop to collocate a physical model and a visual simulation.

When building with Senspectra, users create a physical model assembling the physical primitives of nodes (solid tetrahedrons) [fig. 8-7] with interconnecting joints (flexible linear connectors) [fig. 8-7]. The Senspectra primitives allow for the creation of regular structures, however by making the joints flexible and elastic, the morphology of the regular structure can be altered to reflect non-regular overall geometries. Sensing through the Senspectra joints allows the system to perform cellular finite element analysis in real-time as models are constructed. Senspectra computes the stresses locally with each individual node integrating the surrounding stresses to obtain a unique local stress vector.



Fig. 8-6 Senspectra nodes change color as the joint between them is bent showing structural strain

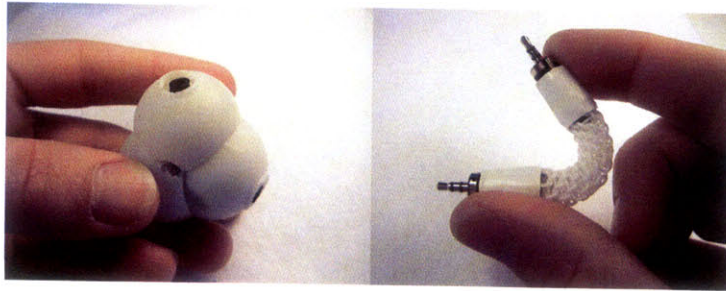


Fig. 8-7 Senspectra elements node (left) and joint (right)

Senspectra provides this real-time FEA functionality and also collocates the output of the visualization directly on the physical model.

The material affordances of the flexible Senspectra joints was chosen to entice users to physically manipulate the digital model in two ways. The bending of a joint between two nodes shows the physical strain on the joint as a mapping of the color in the nodes (red - maximum strain, blue - no strain) [fig. 8-6]. The squeezing of a joint can act to slow the flow between nodes in a model visualizing flow, as pinching a straw would slow the flow of liquid through it. In addition to real-time feedback, Senspectra provides the ability to visualize the resonance of a structure in terms of its elastic stability, by allowing users to record the stresses generated by high frequency oscillations and playback the recordings slowed down as visualizations on the structure.

### *Technical Implementation*

#### *Joints*

The joints are made of silicone tubing and eight conductors wrapped around the tube in a diamond braid [fig. 8-8]. This braiding technique allowed the Senspectra joints to maintain a consistent flexibility and elasticity throughout the designed structures. The braided wires serve for power distribution, peer-to-peer networking, and the sensing of the bending angle of the joint. The tips of the joints are made of two radial connectors [fig. 8-8] which allow for free rotation of the connected nodes, reducing unintended mechanical stress. The silicone tubing within the joints serves as Senspectra's





Fig. 8-8 A Senspectra Joint a) radial connector b) eight conductors woven into a diamond braid c) phototransistor embedded in silicon tubing

omnidirectional bend sensing mechanism. It uses a simple optical occlusion technique measuring the intensity of infrared light coming from an LED on one end of the joint with a matching phototransistor placed at the other end [fig. 8-8]. When the joint is straight, the intensity of the infrared light is at its maximum. As the joint bends the tubing occludes the light to a point where the phototransistor cannot detect any infrared light emitted from the LED. The advantage of using this method over traditional resistive bend sensing is that the joints can bend in any direction and give consistent readings.

#### *Nodes*

The nodes are made of a 3D printed ABS plastic shell embedded with a surface mount PCB [fig. 8-9]. The rigid shell acts as a light diffuser for the embedded full-spectrum LEDs with holes for connectors separated by 109° angles to form a perfect tetrahedron. The heart of the circuitry is an 8-bit RISC AVR® microcontroller running at 8MHz. It controls the color of the LEDs, calibrates the signal coming from the bend sensors and communicates with neighboring nodes through a UART that is multiplexed on four channels and runs at 500kBps.

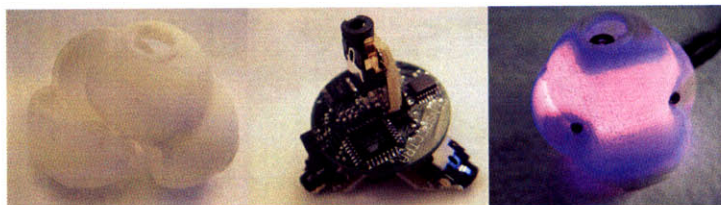


Fig. 8-9 Senspectra Node - casing, PCB and glowing assembled node

While the Senspectra infrastructure provides a flexible modular sensor network platform, its primary application derives from the need to couple physical modeling techniques utilized in the architecture and industrial design disciplines with systems for structural engineering analysis, offering an intuitive approach for physical real-time finite element analysis, particularly for organic forms, utilizing direct manipulation augmented with visual feedback.

Two areas or applications naturally emerged from our interactions with Senspectra. The first is using Senspectra as a teaching tool in structural engineering, for developing an intuition via a physical material for the internal stresses of structures organic in form. The second area emerged as part of a discussion with a leading furniture design company, who upon seeing the system, requested to embed Senspectra into the cushion architecture of an office chair as a way to record strain mappings of a person shifting naturally while seated throughout the day, using it as a testing tool to inform the ergonomics of their design [fig 8-10]. This application marks an interesting shift in intended purpose to using Senspectra as a 2D surface modeller instead of modeling in 3D. Although Senspectra offered limited functionality and resolution in applications, it emerged as an appropriate materiality for interaction, combining a structural repeated geometry of with material malleability for meaningful manipulation.

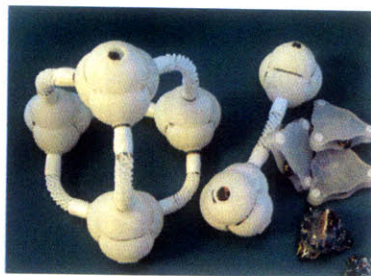
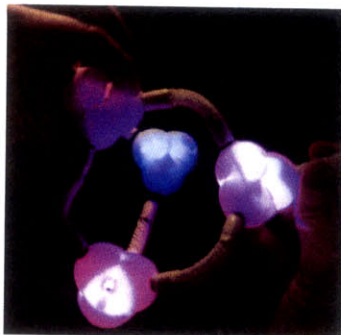


Fig 8-10 Senspectra system and concept of Senspectra in use as an ergonomic testing tool



### **Bosu**

Applying what I learned from the static material explorations of Glume and Senspectra to a dynamic modeling system, I began the design of Bosu. Bosu is a design tool which combines the physical record and playback functionality of Topobo and Kinetic Sketch-up with the organic form finding qualities of Senspectra, bringing kinetic memory to soft materials and marking a new arena of actuation in soft mechanics. It is used for motion prototyping and digitally augmented form finding, combining dynamic modeling with coincident sensing and actuation to create transformable structures. The system consists of varying modular units of bend sensors paired with shape memory alloy (nitinol) actuators woven into a bendable plastic frame and embedded in fabric [fig. 8-11]. Each module can actuate between two positions and together form three dimensional motion pixels.

My experiences with the soft and malleable materialities of Glume and Senspectra informed the structural qualities of Bosu. In Glume, I recognized the interface as too malleable, with unpredictable variability, often rendering control information meaningless. In Senspectra, we determined a more appropriate balance with a semi-rigid structure, constraining the flexibility to areas of the interface which could be modeled and controlled. Bosu is built off the technical infrastructure of Senspectra, utilizing a similar network topology, although the spatial strain sensing is determined via bend sensors instead of the optical occlusion technique.

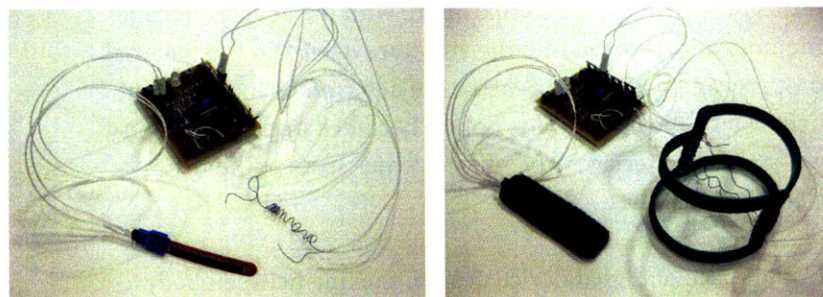


Fig. 8-11 the Bosu System: raw hardware and embedded in a skeletal material structure



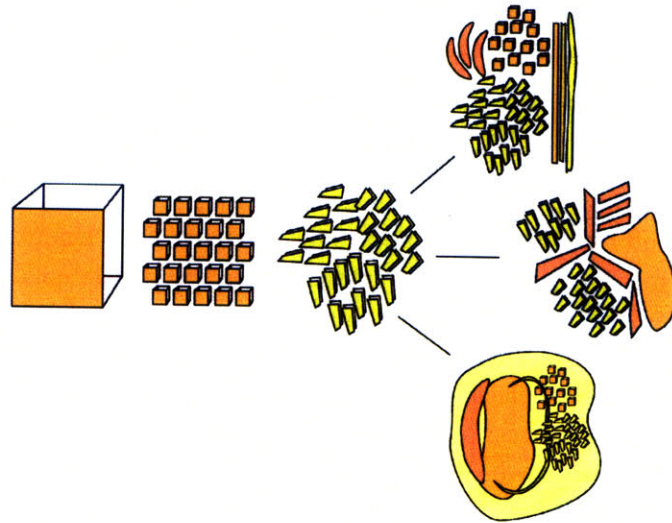


Fig. 8-12 Material transformation - from discrete to skeletal, with different compositions of distinct material qualities

As a kinetic prototyping interface, Bosu moves beyond the domain of Topobo and Kinetic Sketchup, in a shift towards the organism of the body. The structure of the Bosu elements look toward an alternative idea in material transformation where the continuum is no longer a simple repetition of elements descending in scale. Instead, discrete elements move toward smaller repeated elements, dividing into ‘tissues’ of varying densities and structures to prescribe specific behaviors in a separate part of the whole [fig. 8-12]. Designers can start to think of transformable devices in terms of their material composition, with the actuator (shape memory alloy) woven into the structure itself and the material property and geometry of the tissue determines the resultant behavior of the object. This notion becomes important when we consider interfaces and objects which exist in the realm close to our bodies.

In the space of digital device design, the line between body, clothing and object is blurring. Fashion designers have long addressed the notion that what we wear and carry projects

an image of our identities while ubiquitous and embedded technologies are allowing our devices to become more and more a part of us with increasing mobility and pervasiveness. The question is arising of what is human, where the body ends and a device begins, both in terms of assistive and therapeutic technologies, as well as aesthetic statements.

My approach to the design of Bosu considers how the changing concept of the body, and our associated identities, alters how and what we strive to design for ourselves and the nature of digital products made to be worn and used by the body. Transformability can begin to play a bigger role in this design process with a means to physically translate the structural organicism of the body more fluidly and conceptually, blurring the line between design tool and design material.

The material properties of nitinol as an actuator also determine a particular quality of the Bosu system. As an actuator, shape memory alloy does not offer a continuous variability in position, like a servo motor. By applying heat, via a control circuit, it moves between two states - original length and a contracted length where a cool down time is necessary for it to return to its original length. Thus constructions created in Bosu do not lend themselves to a continuously dynamic interaction, such as a Topobo walking creature. Instead the system favors the creation stato-dynamic objects, or object gradually shifting in form. As a design tool, Bosu provides a temporal shift to focusing on stato-dynamicism as a new direction of interaction design.

### **Modes of Operation**

The Bosu PCB features a two button control interface. Like the Kinetic Sketchup Modules, Bosu can be manipulated via direct manipulation or via remote control. Bosu also features a new mode called 'direct control' activated via a second button.

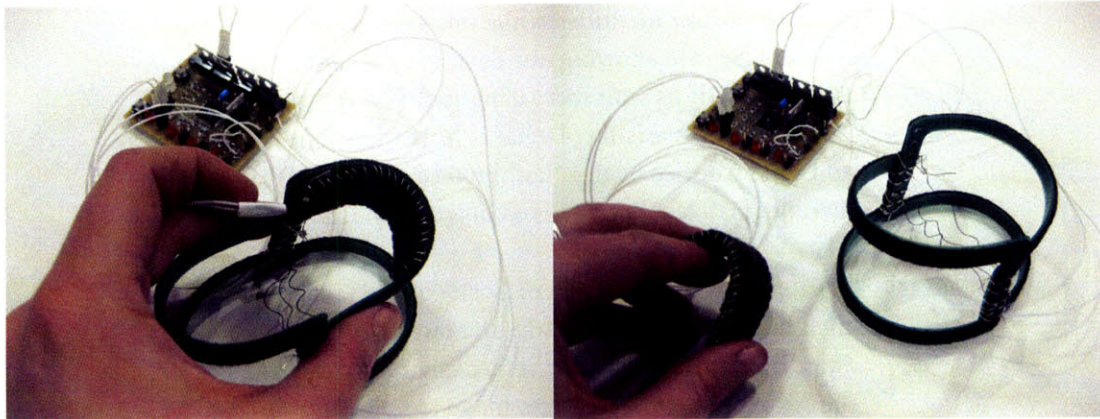


Fig. 8-13 Utilizing Bosu in direct manipulation (left) and Bosu used as in remote control mode (right)

#### *Direct Manipulation*

To record a motion or 'state' into Bosu, the first button is utilized – a single press of the button sets the system into record, a second press into playback and third press to pause – the system toggles between these states. For direct manipulation, the bend sensors of the system (up to 4 per PCB, more if multiple PCBs are connected together) are embedded into the structure in positions corresponding to nitinol springs, creating coincident input and output space for motion recording. During the recording state, a user manipulates the entire structure through a series of positions, which are recorded by the physical bending of the bend sensors [Fig 8-13]. In playback the system sequences through these positions by actuation of the nitinol activated by heat from modulated current in the circuit. A double click on the button plays back the system's last recorded sequence.

#### *Remote control*

In remote control mode, interaction with the system is similar to direct manipulation except instead of the bend sensors being embedded in the structure, they are manipulated away from the structure, attached only to the PCB by thin wires [Fig 8-13]. This mode allows users to record and playback motions without the interference of their hands on the structure itself. This need arose in response to the soft and delicate material nature of many Bosu constructions, making it much easier to control and observe subtle changes in shape.

### *Direct control*

A double click on a second button puts the system into direct control mode in which the moving of a bend sensor directly actuates its corresponding nitinol spring. This mode was designed as an entry point for users to become familiar with the behavior of nitinol as an actuator, allowing for exploration of responsiveness to the sensor controllers and experimentation in timing for actuation and cool down (retraction).

### **Module Design**

Continuing the classification of Kinetic Sketchup, I designed a series of Bosu modules based around the same categorizations of differing mechanical structures. In many cases, these modules can offer a method for material and behavior comparison to their analogous rigid mechanical structures created with Kinetic Sketchup. This offers a simple point of entry for the user, however, the strength and benefits of the Bosu system lie most directly in its functionality as a type of 'raw material,' allowing for its flexibility to be easily embedded and incorporated into custom soft structures. In this way, these modules serve as examples and points of departure for potential behaviors and mechanical



Fig. 8-14 Mixed Bosu Modules



structures but are in no way intended to encapsulate the variability of the system. The ways in which designers flexibly applied Bosu in custom structures is exemplified in the workshop creations described in Chapter 9.

*Hinge*

Fig 8-15 shows examples of the hinge module which actuates between two states, flat and curled up into a bend. The hinge is made from laminating fabric (felt or polyester) and flexible polypropylene sheets, which gives it a soft feel with a spring-back capability. It works like a simple elbow joint, by which many more complex structures can be created. For example, multiple hinges can be strung together to end to end to turn a linear chain into circle.



Fig. 8-15 Bosu Hinges

*Circular Cinch*

Fig 8-16 shows an example of the circular cinch, in which multiple strands of nitinol are sewn in a star pattern across the surface of a fabric circle, with a polypropylene start structure sewn on the reverse, as a return mechanism. The nitinol provides multiple points of linear actuation which



Fig. 8-16 Bosu Circular Cinch

together produce a radial motion, like that of opening and closing an umbrella.

### *Triangle*

The triangle module contains nitinol embedded linearly along its three sides, allowing each side to actuate linearly between two lengths, causing the triangle to change from equilateral to varying isosceles states [fig. 8-17]. Like the hinge, the triangle modules works as a base unit by which many more complex structures can be created. Using triangles to create a surface of repeated patterning, arises as a natural choice because interconnected triangles can describe any freeform shape in physical space, as well as triangulation is employed in the algorithmic process of creating surfaces in 3D modeling software, thus creating a physical surface which could directly mimic a digital model.



Fig. 8-17 Bosu Triangle module

### **Surface for Organic Form Finding**

In creating the Bosu system, my intention was to design one module which would intentionally function as a primitive by which to create a repeated mesh structure. By their nature, meshes take advantage of the structural relationship between the solid and the void, and reference biological paradigms for strength and lightness through spacial looping, such as the bone structure of a bird's wing. The meshed surface is intended to transcend Bosu's functionality as an open ended toolkit, creating a kinetically transformable meshed textile interface which can be used for form finding and capturing organic surfaces. After initial experimentation with repeated triangles, I settled on a solution of repeated trapezoids (functioning like two connected triangles) [fig. 8-18].

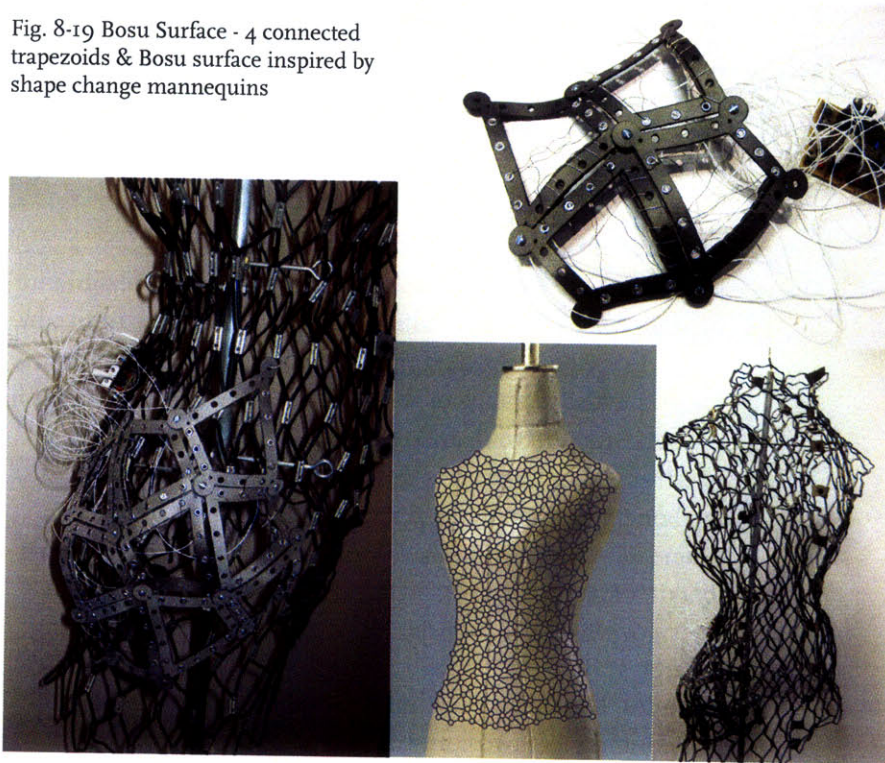




Fig. 8-18 Bosu Surface prototypes

Repeated trapezoids can deform in three dimensions to form overall shapes similar to those created with triangles, but allows a simpler infrastructure where each module utilizes only one embedded strand of nitinol to actuate between two positions, flat and curved. The Bosu trapezoids are created by laminating bendable polypropylene sheet between polyester fabric, the nitinol is threaded through aluminum eyelets, used to insulate the heat in the nitinol to prevent melting in the polypropylene or burning in the fabric [fig. 8-19].

Fig. 8-19 Bosu Surface - 4 connected trapezoids & Bosu surface inspired by shape change mannequins



Together the trapezoids form a 'pixelized' surface which can be deformed in three dimensions surface when draped over an object. The interaction with the surface is similar to working with individual modules with the Bosu hardware in direct manipulation mode. A single button press sets the surface into record mode, and a second into playback. The modules record and playback their state independently but function like a distributed network with a global behavior to produce an overall form.

The Bosu surface is intended to be used as an interface for recording a three dimensional snapshot of curved surfaces, a sort of object surface recorder. Inspiration for the surface was taken from the idea of personalized mannequins which when wrapped around the body, record the body form by deforming the metal mesh [fig. 8-19]. The Bosu surface could potentially replace the need for a 'fit model,' (a person representing a specific size) an idea that emerged while consulting faculty at the Boston School of Fashion Design as to the state of the art in digital tools for fashion design. While technologies currently exist for scanning bodies into a digital model, the nature of fashion design favors working purely in the physical realm. Currently, the body is physically defined as a series of linear measurements, inadequately representing the uniqueness and individual nature of each body. While lower in resolution, the Bosu surface seeks to present an active physical measure of the body surface in three dimensions, the first step to being able to record and transform between multiple physical body forms.

### **Preliminary Usage**

#### *Fashion Metamorphosis workshop*

The first prototype of Bosu was used as part of a two day workshop entitled Fashion Metamorphosis at the Nuova Accademia di Belle Arte Milan, Italy. The workshop focused on the design of 'body objects' whose function lie in between clothing, furniture and appliances. I showed the students Bosu as one example of a physically transformative material (another example was thermochromic color change dyed fabrics). Although the system was not in a developed enough

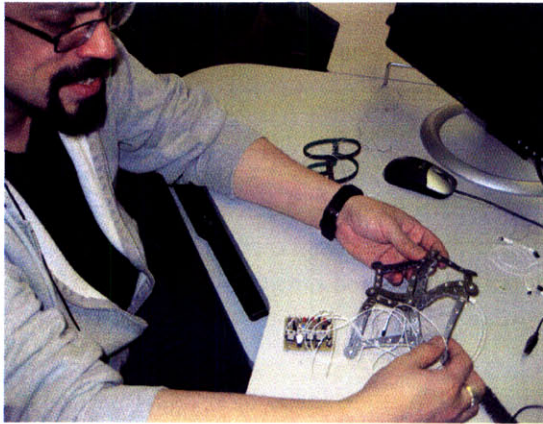


Fig. 8-20 Bosu being evaluated at the Boston School of Fashion Design

state for the students to use it functionally in their designs, Bosu provided an inspirational example embodying the idea of physical shape change and kinetic memory in fabrics or soft structures. The output of the workshops was a series of stop motion animation movies, video prototypes of new concepts in transformative fashion design. Based on Bosu, one student designed a conceptual suit which could electronically change its weave structure from woven to knit, going from a tailored formal fit to a loose stretchy fit.

Bosu was further evaluated in conjunction with the Kinetic Sketchup modules in a week long workshop described in Chapter 9.

## *Chapter 9*

### **Workshop Evaluation of Kinetic Sketchup and Bosu**

The evaluation of Kinetic Sketchup and Bosu took place as a four day workshop entitled 'Prototyping Motion: Transformable Design and Kinetic Behavior in Architecture and Product Design' during January 2009 at the MIT Media Lab. The workshop was organized and run with help from MIT architecture student John Pugh. The workshop focused on stimulating issues and directions for the future of transformable design, addressing questions such as:

- What is the role of kinetic architecture or structures in solving urban infrastructure issues (in energy, for example)?
- How does transformability lead to new ideas in interactivity and interaction design? What is its role in ergonomics or universal design?
- What are new ways to think about motion design using new/smart materials which allow for new motion qualities perceptually and functionally?
- How is perceived anthropomorphization in motion design to be capitalized on, or, inversely, avoided?

The focus for the evaluation was the facilitation of creative expression for new forms of transformability and in evaluating design thinking around the blurring of design tools





Fig. 9-1 Workshop participants engage with Kinetic Sketchup modules

and materials. In evaluating Kinetic Sketchup in a workshop environment for designers of ranging interests, I posited that by using the Kinetic Sketchup and Bosu modules as part of the design process, the participants could reference their own bodies, physical intuition, and material assumptions as tacit knowledge, much as the children did in the design of walking creatures with Topobo. This would in turn ease and expand the process of designing transformation. The evaluation focused on how the design process is altered by use of a new physically programmable tool, in terms of functional characteristics, affordances, collaboration, and the capacity for supporting a creative, expressive and inspirational user experience.

Participants in the workshop were self-selecting, recruited through campus-wide emails and posters at MIT and Harvard. The final workshop totaled 11 participants- 6 male, 5 female -who were professionals or students in the areas of architecture, fashion design, interaction design and mechanical engineering. Participants were minimally familiar with concepts of transformable design, showing an interest in the subject matter, and some came to the workshop with existing ideas which they planned to explore further. Data was collected both in video and personal observation during the workshop as well as written and oral interviews at the end of the workshop. The results are presented as overall findings on the use of the tools and the design process and as case studies of individual projects which emerged from the workshop.





Fig. 9-2 Workshop participants engage with Kinetic Sketchup modules

On day 1, the workshop began with a brief lecture on the state-of-the-art in transformable design including theory and examples from architecture, product design and robotics. After the lecture, participants were given the challenge to, over the next 3 days, design and prototype a concept that employs physical motion as a design parameter. They were then given an hour of free time to brainstorm and discuss ideas and identify overlap or similarities of interest which could result in collaboration on projects. The Kinetic Sketchup and Bosu modules were then introduced as prototyping materials to explore motion concepts as part of their prototyping process and in their final models and diagrams, as they so chose. For clarity, the systems were introduced separately, allowing participants to become familiar with the interface, functionality and limitations of each system in an open-ended fashion before attempting to construct their specific idea. The participants spent the rest of the afternoon experimenting with the systems, in many cases, using the motion variable concepts introduced, such as twitter or acceleration, as inspiration for the basis of new project ideas.

Day 2 through the morning of day 4 were spent working open-endedly on the design and creation of project ideas. Participants has a wide variety of materials available including

plastics, foams, fabrics, wood, paper and wire as well as use of the lasercutter, sewing machine, and varying construction tools. Informal individual check-ins sessions were conducted by John and I with each participant during these days for conceptual and construction advice. The afternoon of day 4, the participants presented their projects to the group and we conducted oral interviews followed by a written questionnaire.

### Overall Findings

All the participants of the workshop were able to successfully utilize the Kinetic Sketchup and Bosu systems and appreciated the consistency of the record-and-playback interaction style of both the systems. The final creations from the workshop included ideas in architecture, furniture, fashion and product, both functional and conceptually abstract [fig. 9-3] with the participants working with materials of mixed properties, hard and soft, in all cases. Participants were generally surprised by the unexpected behavior of soft materials and fabrics when interacting with actuators, highlighting to many the importance of working in the physical realm when first experimenting with kinetic design and unfamiliar materials.

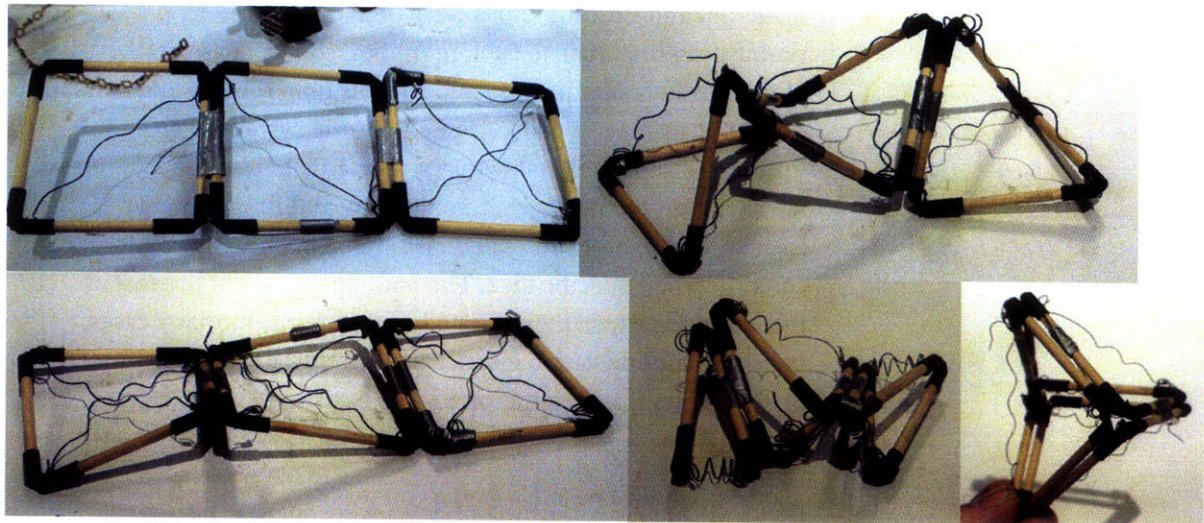


Fig. 9-3 Abstract transformable construction by a workshop participant with Bosu - conversion from three flat squares to a tetrahedron using malleable joints and six nitinol segments



### *Inspirational vs. Construction Tools*

Throughout the workshop, I observed the participants' design process evolve through a series of phases when using the modules. They began in an 'observational' phase in which users experimented with very simple motion constructions of one or two modules and an isolated behavior pattern. The modules essentially served as 'inspirational examples' for varying kinetic behaviors. Key to comprehension for a user was the ability to isolate changes in a motion or mechanical property, allowing a user to clearly dissect the cause and effect of their actions on a system. The rigid modules of Kinetic Sketchup worked best for this behavioral isolation, providing a simplified embodiment of a motion. This allowed users to think broadly about it in context, as one workshop participant commented, "With the Kinetic Sketchup modules, I observed the mechanical behaviors in detail and tried to connect behaviors with real world issues." During the observational phase, the physicality and aesthetic of the modules was very important in conjuring familiar motion qualities of recognizable objects in motion – sliding pistons, spinning fans, blooming flowers, etc. Because motion construction is a relatively unfamiliar form of design, working with a physical medium in the observational phase was essential to understand spatial translation and real world forces surrounding objects in motion and highly influenced the designers' perspectives as they moved forward with their ideas.

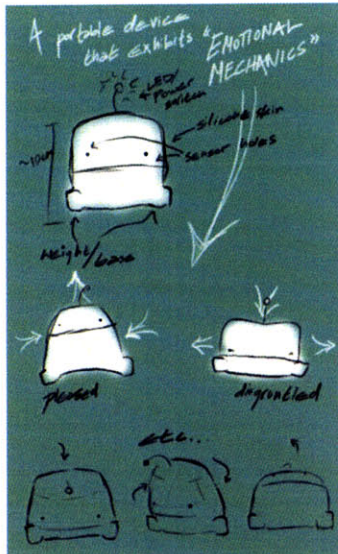


Fig. 9-4 Sketch by a workshop participant of a creation displaying 'emotional mechanics'

Following the observational phase, participants moved onto a 'constructive' phase, in which they designed their own creations from scratch. Some participants honed in on a single behavior they observed, and built an idea around it, such as a twitter module becoming a creature displaying 'emotional mechanics' mapped contextually to actuate based on email and web traffic [fig. 9-4]. Others returned to their original project ideas and began to map their observations about motion control and design to the system they intended to create. Like the successful iterative design style observed with children making walking creature with Topobo, an iterative prototyping style where form and motion

were developed in tandem based on materiality, was most successful for the participants.

As participants moved through the project working on more customized constructions, the modules themselves became less useful as tools. Designers began working directly with the nitinol strands, embedding them into their structures, or stripping Kinetic Sketchup modules down to raw servo motors and PCBs. This raises the question of what is the appropriate level of 'tool' for physical motion prototyping. The designers' ability to rapidly move toward raw actuators potentially negates the purpose of the modules themselves. However, the ability to conceptualize and construct kinetic systems so quickly and intuitively reveals a level of understanding which was derived from the observational phase of the process, a direct result of the mechanical and behavioral experimentation with the modules, the inspirational source for conceptual and functional ideas. One idea which emerged was to another set of modules consisting purely of motors with simple varying mechanical attachments for connectors, although retaining the essential gestural and physical programmability which keeps the system intuitive and accessible for non-expert users. This would take the Kinetic Sketchup modules one step further towards a raw material.

#### *Temporal Design*

Perhaps the most salient and observable shift in the design process occurred around the process of design for physical change through time. Designer's are typically concerned with an object's static presence, functionally and aesthetically, but the introduction of transformation opened up new channels of thinking, both in designing the process of change, and the objects multiple states. One participant observed, "I was thinking in the 4th dimension...with how the object I would create would change over time. So I was not stuck on designing a specific static object. It influenced the materials I chose and used as well as pushed my conceptualization abilities to think of the object as just one state among many. I had no idea how it would really turn out. I'd never had this

kind of process....It felt like I was conceptualizing in video,... but I could touch it and interact with it.” This idea marks a conceptual shift in the design process, where designer’s are able to concurrently think and improvise both physically and temporally, bringing to light the emergence of stato-dynamicism as a potential design strategy.

#### *Materiality and Organicism*

Outside of the temporal shift, what was most unique to the designers’ experience came from the nature of soft materials and the novelty and unexpected behavior of nitinol as a smart material. When presented with a range of material qualities from hard, layered, hybrid to malleable, the shifting nature of materiality became a design inspiration in itself. This stressed the importance of working in a physical medium to challenge assumptions about kinetic behavior and leverage our intuitive understanding of the physical world, “(I learned) how important the possibility to experiment with a lot of different materials is for this kind of design, how important material properties are, even with powerful actuators.” By juxtaposing the mechanical motion of Kinetic Sketchup with the organic motion of Bosu, participants were confronted with a comparison of motion attributes, revealing a relationship between what can be viewed as ‘machine’ motion versus the organic motion of our bodies and the natural world. With tools available to experiment with a range of motion qualities, participants were offered insight into the nature of how to create diversity within motion design. One designer stated, “I began to understand the pure subtleties involved in the process of capturing movement. Particularly how one might attempt to capture human or animal movement over mechanical.”

#### *Intuitive Technology*

While physical temporality and organicism added a shift in design thinking, Kinetic Sketchup and Bosu attempted to ease designers’ explorations into transformable concepts which they were previously hesitant to tackle due to limited technical knowledge. The system lowered the point of entry on mechanical design, providing an infrastructure on which



to build and models to emulate. At the same time, the modules provided simple insight into engineering issues of motion design, related to the spatial nature of mechanical components as priorly unconsidered, “Spatially it helped me to think about designing actuated products with space constraints, where actuators can be placed inside an object to get the desired effect.” The intuitive nature of record and play offered a simplicity and accessibility to the technology of programming motion, as participants noted, “Gestural recording was useful for experimenting with the nitinol, could easily observe how long it takes to move the screen when actuating the nitinol, what properties we want to design – change in speed, time of different stages.” and “The bend switch is a fantastic aide, very efficient way to test ride an idea and also to create a kind of choreography on the fly.”

### **Issues and Limitations**

While Kinetic Sketchup and Bosu added to the kinetic design process in many ways, they also possessed several issues and limitations for working with motion design. The most significant proved to be the number of components available and the level of independent control offered in choreographing motion. Designers naturally desired the multiplicity so inherent in digital systems, where ideas could be generated with complex motions systems involving hundreds of actuators, a limitation commonly noted in tangible systems. Many of the projects developed in the workshop were conceived as distributed systems with simple repeated elements designed to give an emergent global effect. By physically engaging with Kinetic Sketchup and Bosu components, designers were confronted with the reality of designing mechanical systems in physical space and they quickly came to realize the complexity they faced in a real world system. Designers also reached limitations of coordinating motions into more choreographed structures, citing the need for additional sensing and feedback systems to further develop their interaction concepts. While Kinetic Sketchup and Bosu provide a simple point of entry exploiting physical intuition for motion control systems, the next challenge is to balance this with the variability and complex



Fig. 9-5 Body as Pop-up book by a fashion designer

control structures of digital systems. Ideas related to how the kinetic design process could integrate procedural animation and motion authoring techniques for virtual characters are discussed as future work ideas in Chapter 10.

As with all prototype physical systems, Kinetic Sketchup and Bosu suffered from issues of mechanical reliability and sheer number of each different modules available to the participants. While these issues can be directly addressed with further development and production, it emphasizes the mechanically demanding nature of physical systems and the difficulties of deploying tangibles in large scale as a research platform.

### **Project Case Studies**

The following four projects offer a sampling of the ideas emerging from the workshop.

#### *Body as Pop-up Book*

A fashion designer came to the workshop looking to investigate the concept of a transformative garment which used the natural motion of the body's limbs to actuate a structure, like a pop-up book construction or a 'three dimensional suit of armor' [fig. 9-5]. The garment would take the contours of the body, being made of multiple repeated units. She created cubes that moved from 2D to 3D by

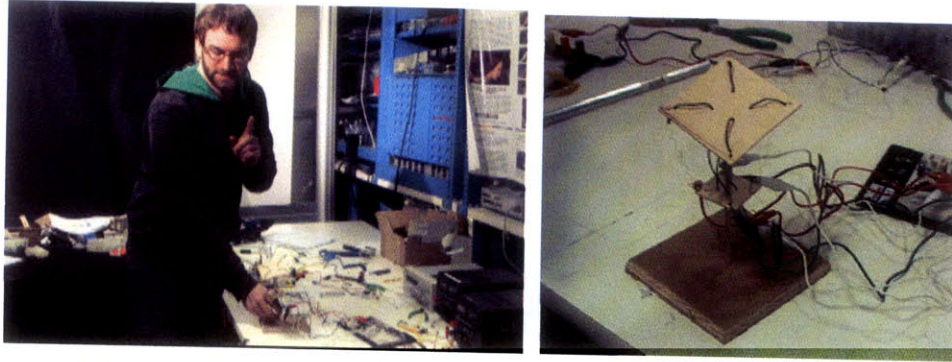


Fig. 9-6 Interaction designer demoing his context aware screen/pen

embedding nitinol in foam laminated with polyester fabric to create a hybrid material that would form malleably to the body while also provide structure to counteract the actuation. She used the Bosu hardware to program the motion using bend sensors by remote control. Although she intended to use the body as the actuator for the final garment, she experimented with designing the kinetic behavior of the units off the body. In different mechanical designs, she was able to isolate variations of motion, and observe and tweak the behavior until arriving at a desired motion effect before adding the variability of the body in motion.

#### *Context Aware Screen/Pen*

This concept by two interaction designers combines a tablet pen with a mini-monitor for spatial co-location when interacting with digital data, illustrating an idea of employing actuation for creating context awareness in a interface. As a user writes with the pen (on a digital tablet), the screen atop the pen stays facing the user by continuously adjusting its angle in relationship to the pen [fig. 9-6]. To illustrate the concept, designers used four nitinol springs attached from the pen to the four corners of a square screen. Using the Bosu bend sensors as direct controllers of the nitinol (off the structure), they were able to improvisationally demo how the screen would stay facing the user. The designers originally conceived of using four motors to rotate the screen but because of spatial constraints decided on the four stranded nitinol actuation. This construction resulted in the



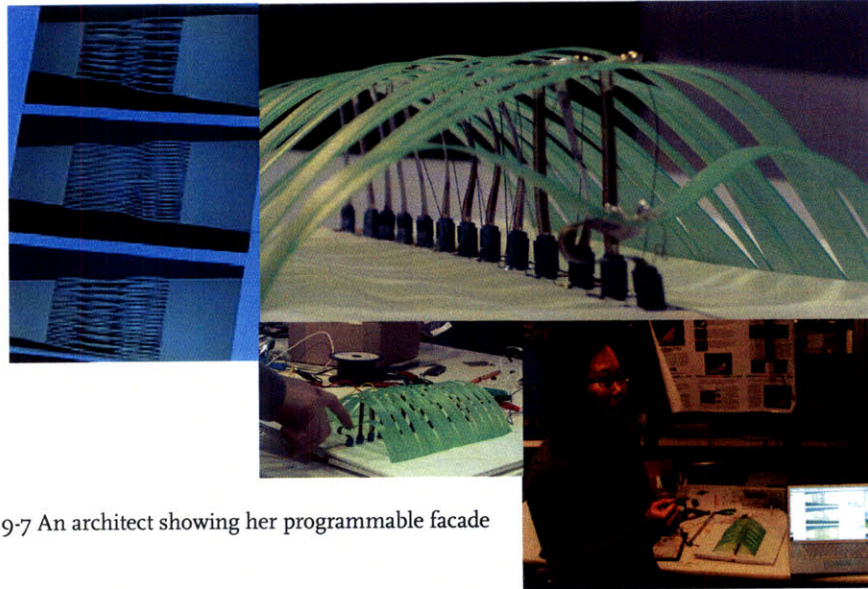


Fig. 9-7 An architect showing her programmable facade

unexpected effect of creating an extremely smooth ball and socket joint (3DOF), with a very organic motion, like a snake following the movement of a charmer.

#### *Programmable Facade*

Designed by an architect, the programmable facade uses the metaphor of record and play in the Bosu system as an interaction scenario scaled up for a kinetic facade [fig. 9-7]. The permeable membrane can be programmed by pushing and pulling on the malleable structure of the wall to record stato-dynamic states or kinetic patterns played back in the surface. In her model, bend sensors are embedded in the structure coincident with nitinol spring actuators in a linear formation. While she originally envisioned the interaction at a 1-to-1 scale of the body to the wall, while working on the prototype, she also conceived of a scaled hand held remote controller, like a musical instrument that could be strummed with the fingers and translated into the architectural surface, changing in materiality as well as scale.

#### *Reconstructing Vase*

A highly conceptual idea developed by an interaction designer, the reconstructing vase addresses issues mapping

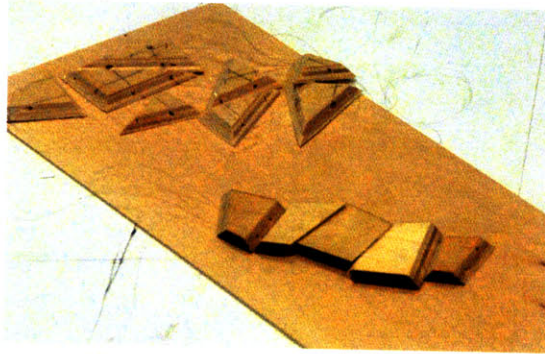


Fig. 9-9 Reconstructing vase concept by an interaction designer

functionality or uniqueness to the destroyed or regenerated state of an object. He was addressing the possibility of using the way things transform (for eg. the violence of smashing) to convey information or emotion as well as the functionality of different intermediary stato-dynamic states (for eg. a lamp shade changing shape to cast light in different ways). To illustrate the idea, the designer used simple wooden elements, with edges cut to specific angles, strung together [fig. 9-9]. A single Kinetic Sketchup unit (servo motor) was used for a large scale motion of coiling the string to contract the object, with subtle small motions of nitinol installed between the pieces to pull them into formation. While extremely low resolution and barely functional in its prototyped state, this project makes an important shift in the development of forward thinking ideas in transformability like programmable matter. By using tools and technologies that are available now (like Topobo and Kinetic Sketchup), he was able to create a comprehensible interaction scenario to shed light on how we develop interaction techniques for materials that are presently out of our physical familiarity.

### **The Threshold of Controlability**

In the workshop, the majority of participants chose to work with Bosu using nitinol based actuation. In some ways, this can be accounted for by the novelty of interacting with nitinol as a smart material and the excitement designers expressed for a system which allowed them to control it without technological overhead. However, for those who did





design but operating in an expressive realm just outside of their plan.

Programmability with Bosu has touched on the very nature of the human fascination with motion as a communication medium, where even in a non-anthropomorphized form, we can identify a quality of being alive. Striving toward organic forms of motion shows a common thread running through the projects, where the most engaging for both for the designers and the viewers were operating at the threshold of controlability. A shift in materiality toward pure malleability can derive an organic feeling in an interaction but the level of controlability falls proportionally [fig. 9-10]. The use of hybrid material structures where we can identify an underlying skeletal structure combined with organic elements, expands the threshold of controlability [fig 9-10], pushing both the natural-like nature of an interaction with the designer's ability to control and designate the interaction. It is within this realm that I envision the future of interaction for kinetic and transformable structures.

## *Chapter 10*

### **Looking Forward : Materiality and the emerging Design Process of Motion Authoring**

The development of Tangible User Interfaces [Ishii 97] was predicated on an important shift in thinking in the nature of human computer interaction with the goal of extending access to computation beyond a traditional graphical user interface (GUI), to interfaces that use physical objects as tangible embodiments of digital information.

#### **The Evolving Tangible Interaction Loop**

In the tangible interfaces vision of interaction, a physical object combined with computation gives both tactile and visual feedback to the user as illustrated by the Tangible Interaction Loop [fig 10-1]. The addition of kinetic behavior through an actuated object or interface creates a second interaction loop, where the user receives tactile and kinetic feedback via motion of the object, as well as visual feedback. As a kinetic memory object, Topobo provides feedback as a reflection of the user's gesture, while in an interface like Pico [Patten 07] the computer becomes a physically active participant in the process of problem solving. Both systems benefit from the physical nature of their kinetic feedback, with the interface responding to the physical forces of the surrounding environment, such as friction and gravity.

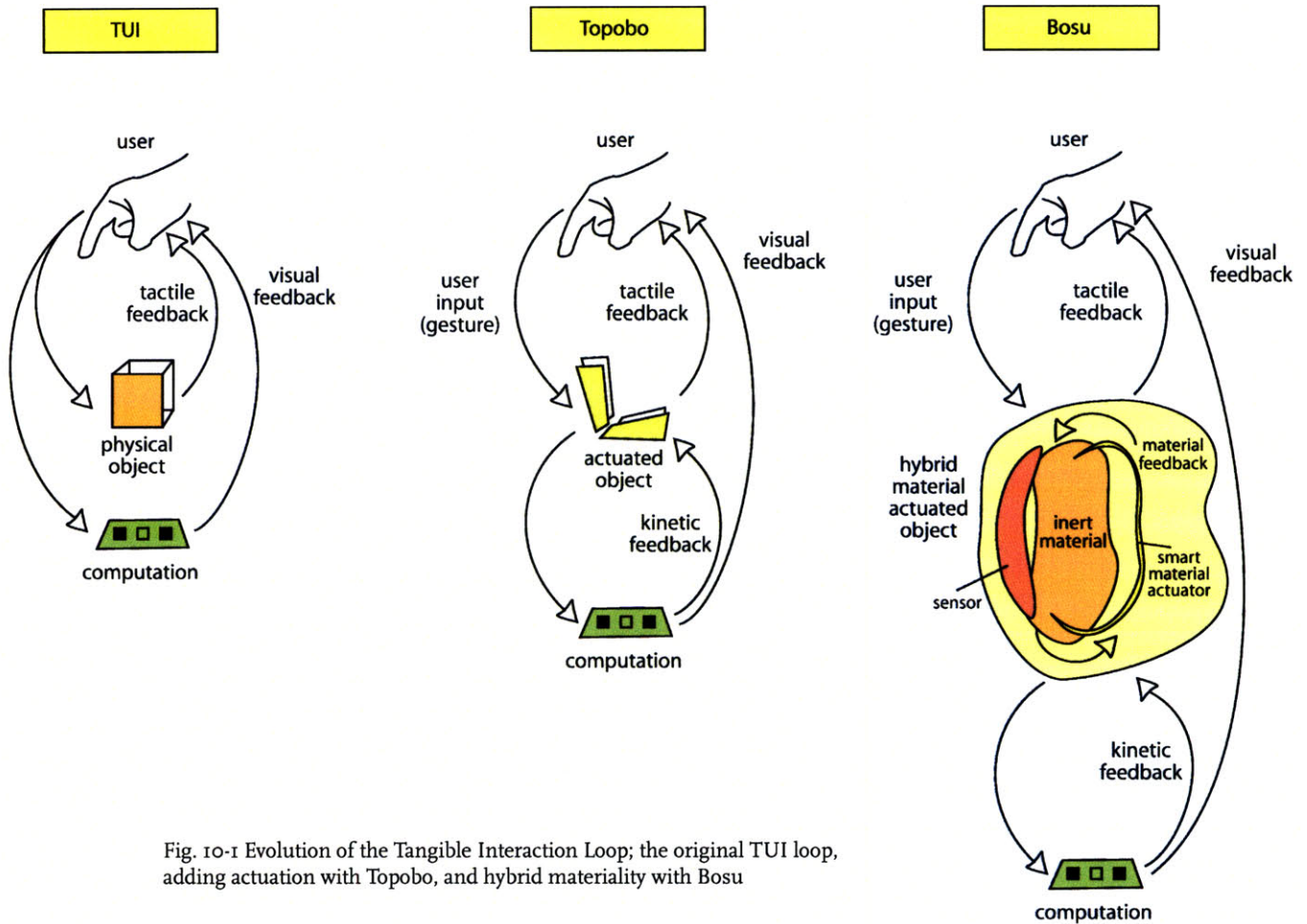


Fig. 10-1 Evolution of the Tangible Interaction Loop; the original TUI loop, adding actuation with Topobo, and hybrid materiality with Bosu

The hybrid material structure of Bosu introduces a new element into the Tangible Interaction Loop [Fig 10-1] where the material itself creates its own internal loop. A hybrid material actuated object receives user input via gesture which is translated both into computational input and material input via the sensor. Once in motion from a user's gestures, the smart material actuator in turn creates material feedback back onto the sensor, resulting in new computational input to the actuator. Designing how this internal material loop affects the tactile and kinetic feedback to the user is an area as yet unexplored and offers interesting possibilities for new perceptual qualities in interactions. This material loop is just outside of the control of the user and yet is affected

by the user's actions, it is in this realm in which we can learn to design how an interface to be kept at the threshold of controlability. The user must easily comprehend the resulting cause and effect of their actions and the logical outcome of the computational feedback but it can also offer an element of surprise, relaxing the rigidity of expectations from computational systems. What can emerge is an intuitive yet slightly unexpected scenario often associated with phenomena of the natural world.

To look forward into the changing role of materiality and motion construction for the physical world, we must first turn to the sophisticated systems that have been developed for motion design and animation in the digital realm. This thesis has addressed designing tools for motion in the physical realm, specifically tapping into the intuitive nature of gestural interaction incorporating material properties. In looking to the future of motion choreography and coordination, however, much can be learned from the conceptual theories of digital animation and their associated algorithmic control structures. The point is not to debate the validity of the physical vs. the virtual but to find a commonality in conceptualization, where ideas from each medium can inspire new connections to translate between the two realms. The hope is to arrive at a coordinated system or set of design tools which allows the digital world to contribute its facility of computational power offering high resolution, multiplicity and complexity while the physical world provides intuitive tactile interactivity through bodily knowledge and engagement for a fluid approach to the kinetic design process.

### **Animation with Verb and Adverbs**

Research into controllable human figure animation can be divided into three major groupings: procedural, simulated, and interpolated [Rose 99]. Procedural animation uses code phrases to generate animation values (DOF) in real times, offering a more diverse series of actions that could be created with predefined animations. Simulated figure animations use the modeled motions of the human through controls to generate motion. Interpolated animation uses sets of example



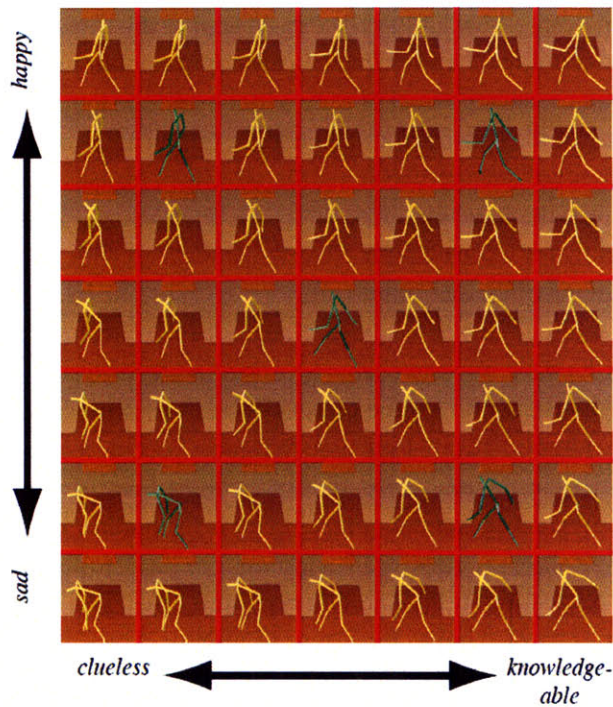


Fig. 10-2 Using the Verbs and Adverbs system, walking sampled across two emotional axes. Green motions are sample motions, the rest are created from the verb/adverb mechanism (figure from ROSE 99)

motions together with an interpolation scheme to construct new motions, essentially combining the benefits of the first two groupings. One popular approach to motion control is entitled 'verbs and adverbs' established by Charles Rose [Rose 99] in which interpolation is formed simultaneously in real-time over multiple dimensions, such as emotional content and physical characteristics. In the authoring state of verb and adverb system, 'verbs' are constructed as controllable motions from sets of examples. The examples could be derived by keyframing or from a motion capture system, assuming a basic structural similarity to the motions. Each motion example is annotated by hand with a set of 'adverb' values as well as a set of 'key times' indicating when important structural elements of a motion occur. Verbs are

then parameterized by ‘adverbs’, or changing interpolation parameters. The adverbs may represent emotional axes such as happy-sad, or physical parameters such as direction [fig 10-2]. The keyframes in the verbs specify periods to engage kinematic constraints. Once a set of verbs and associated adverbs have been established, they are placed into a continuous “space’ of motions parameterized by the adverbs, with the dimension of the space as the numbers of adverbs. Thus, a single authored verb produces a continuous range of subtle variations of a given motion at real-time rates and as a result, simulated figures alter their actions on momentary mood or in response to changes in their goals or environmental stimuli.

In addition, the system features ‘verb graphs’ which provide the means for simulated figures to seamlessly transition from verb to verb within an interactive runtime system, like a kind of blending or tweening between motions. As an interpolated system, verbs and adverbs seeks to address the problem of providing a set of meaningful, high level controls known to the animator or system while maintaining the aesthetic of the source motions. From this system, we deem a possible method of meaningfully translating the Kinetic Sketchup behavioral modules into mapped emotional states through choreographed complexity.

The verbs and adverbs system has been applied as the basis of many iterated systems for higher level synthetic character and robotic controls. For example, in the design of synthetic characters, Blumberg applied the verb and adverb system for motion in real-time of directable creatures in a system of reinforcement learning [Blumberg 02]. Blumberg creates a system in which state, action and state-action space are addressed simultaneously, in which state refers to a specific configuration of the world as sensed by the creature’s sensory system and action refers to how a creature can affect the state of its world. In his system, actions are implemented as discrete verbs with parameterized adverbs, identifiable patterns of motions through time.

Part of the inspiration for the verbs and adverbs system comes from prior work done by Perlin on responsive animation with personality in a system of real-time graphic puppets with defined actions and weights [Perlin 95]. In the system, individual motion are programmed into the puppets beforehand, while also ensuring transitions between any pair of actions are visually correct (like the functionality of Rose's verb graphs). The system allows for the combination of predetermined movements with physical laws where dependencies are implemented by a sequence of conditional expressions with an approach similar to Brooks' subsumption architecture for walking robots, in which more immediate goals block out longer term goals [Brooks 86]. Perlin uses the interpolation technique of noise functions to simulate personality and emotion in existing animations, in order to convey the "texture" of motion.

### **Material as Adverb**

The verbs and adverbs system and the systems which draw inspiration from it has much to offer theoretically to the physical world of motion design and construction. The Kinetic Sketchup modules have a base corollary in verbs and adverbs, where an original gestural motion recording can be considered a verb and the real-time remote control behavior knobs working as adverbs. In the physical world this offers very specific but limited controls. Additions to the system could include a method for key framing within a motion, programming (perhaps with a single button press) places in a motion sequence which mark a specific important transition, and which would be kept in place in spite of any behavioral tweaking with a controller. The idea of combining motions into longer patterns and nested within each other was explored with the Topobo Remix system [Raffle 07]. Raffle had substantial success in allowing children to construct simple motion phrases for walking creature but purposefully limited the complexity of the system to stay within the conceptual threshold of children. Applying the concept of a verb graph for a sequence of motions, which would allow motion sequences to transition and blend seamlessly, could allow for more detailed motion authoring, giving the user

multiple choices motion patterns to sequence through in selection.

Many of these ideas can be implemented within the existing Kinetic Sketchup and Topobo control structure but the question remains where the physical benefits of a programming-by-demonstration system are outweighed by the complexity of controls reach their limits. I posit that the intuitive benefit of working in the physical realm will be outweighed when the comprehension of cause and effect of a user's gesture is lost, much in the way the abstract Topobo Backpacks grew too complex for the children. At this transition point, a combined digital-physical system for motion construction would be preferable, allowing the user to benefit from a simple physical interaction in creating motion but scale up in behavioral complexity and multiplicity of motions in a virtual world.

Perhaps most interesting and unique to the domain of physical motion design is to further investigate the internal material feedback loop created in Bosu, shown in figure 10-1. The material construction of an actuated object can be considered a system of 'adverbs' influencing the 'verb' of the basic motion recording. As materials, the adverbs are

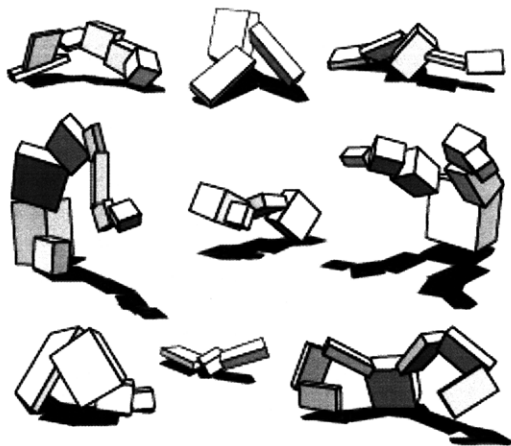


Fig. 10-3 Karl Sims' creatures evolved for walking

physically linked to each other and thus interdependently react to each other. Changing material properties within the system results in a morphological change in both the form and the motion of an object. The key to advancing a framework around this topic is to develop classes of materials (and corresponding forms) which combine to create a particular behavior.

### **Motion Material Morphology**

The work of Kark Sim's evolved virtual creatures [Sims 94] can provide conceptual inspiration in the development of a material morphology for kinetic objects. Sims developed a system for creating morphologies of virtual creatures evolved to move in a directed way - walking, swimming or jumping, for example [fig 10-3]. Referring to biology, the system uses the concept genotype and phenotype for a creature's morphology and a system of virtual sensors, neurons and effectors for control. In the design of a physically kinetic object, embedding sensors in interdependent materials which can actuate themselves, and collecting data to evolve a motion behavior would allow the system to be self-morphing and generative, within a limited reference frame. The designer's role becomes more directly in conceiving of the original genotype and phenotype of an object and in intervening with material construction to move an object out of its limited reference frame.

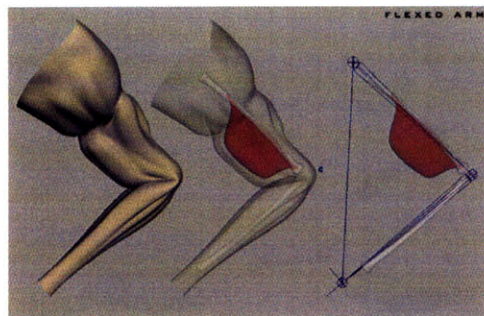


Fig. 10-4 Body memesis as model for transformable kinetic materials and objects



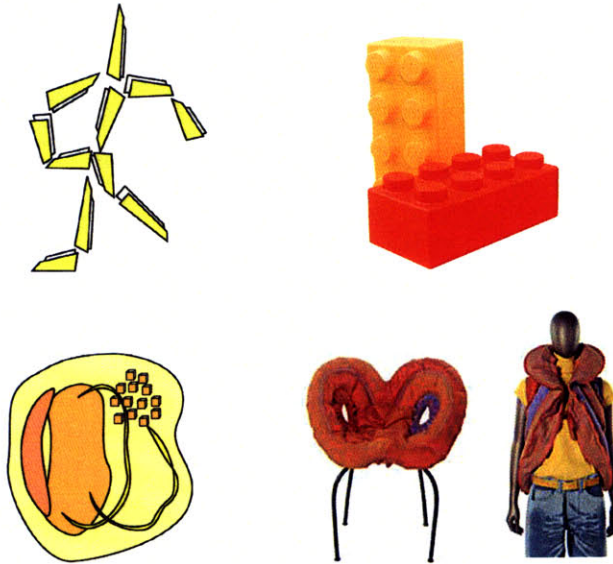


Fig. 10-5 Shifting material construction -from tectonic, like LEGO™ (above) to hybrid like the Apoc Gemini Chair/Clothing System

The ideas of procedural animation and motion interpolation for the virtual world have generally been attributed to characters, looking to express movements to walk, move and respond in their environment as living beings. As we consider the area of transformable objects – products, architecture, fashion – the motion behavior and functionality takes on a new set of requirements and constraints. Motion can deliver a sense of anthropomorphization through abstraction in non-anthropomorphized objects which can be utilized to develop personal relationship with our objects and environments. Material structuring also plays an increasingly differentiating role. We look to our bodies for inspiration, the hybrid nature of the coordinated systems of bone, muscle and skin [fig. 10-4] in new material constructions. We are moving beyond a world where a repeated tectonic structure, like the LEGO™ system, toward hybrid material structures, like the Apoc Gemini Chair/Clothing system from designers Ron Arad and Issey Miyake [fig. 10-5] when seeking a new morphology for transformability and in the technological emulation of future materials.



## *Chapter 11*

### **Conclusion**

At their core, computers are design tools, they allow us to expand our ability to create, think, organize, and learn. The systems which we build to interface with the digital data they provide can be considered in the same paradigm of all tools which we use for non-digital tasks in our daily lives. However, the complication of a human-computer interface lies in the fact that it provides both the raw material and the functionality of use in the same space and of the same material. There is no longer the distinction between organic materiality and the discrete function of the tool with which we manipulate it. This can allow for remarkable advantages in ease of use but also negates many experiences of bodily intuition and feedback.

As Tangible User Interfaces [Ishii 97] become more refined and sophisticated, the future of human computer interaction lies in finding a bridge between the organic structures in the natural world which we seek to emulate and manipulate and the rigidity and exactness of the digital systems which provide the tools for manipulation. As Ellen Lupton comments, “organic forms and materials provide designers with a humanist vocabulary that affirm society’s place within the natural world.” [Lupton 2002] The constructed and the

organic are converging, and the digital materials and tools in development should address this phenomenon by providing an organic material means to engage the tactile senses in the act of creating and modeling.

The rise of ubiquitous computing has brought about the development of innovative systems involving a multiplicity of small computers embedded in everyday objects and the surrounding environment, we are no longer constrained to think of computers as a box on a table. By combining the notion of ubiquitous computing with the approach of direct manipulation, to improve the directness and manipulability of an interface, the possibility for a new class of interaction tools and materials emerges. This new class of materials has its basis in the interaction techniques and tools of Tangible User Interfaces, designed to give physical form to digital information. In order for human computer interfaces to reach a more sophisticated state, they must perform, respond and react in ways that mimic the body and human behavior, not just, or necessarily in their intelligence, but in their materiality.

We can return to Braitenberg's idea of the 'law of uphill analysis and downward invention' where it is easier to create an internal structure of a given behavior, than it is to guess the internal structure from observation. We can apply this concept both with simple kinetic structures but also with a deeper understanding of hybrid material structures and their effects on movement and transformability, expanding our vocabulary of design elements in kinetic construction. The nature of physical tools for motion design creates a shift in procedural design thinking, where designers are able to concurrently think and improvise both physically and temporally, bringing to light the emergence of state-dynamicism as a potential design strategy. Temporal design thinking can be expanded to consider the very nature of product design. Each product is a temporary instantiation of its material form, dynamically changing from raw material to functional product to waste, and its lifecycle can become part of its embodied form. The Nokia Active Disassembly phone,

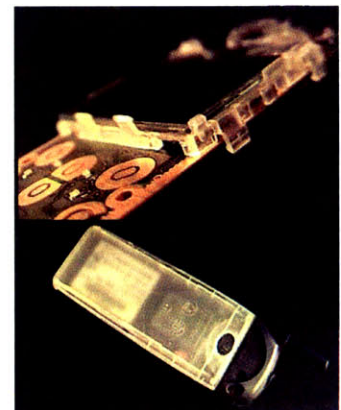


Fig. 11-1 Nokia Active Disassembly Phone

[fig. II-1] designed to separate components for recycling via heat activated nitinol, marks a first step in this design methodology. We also think to the future development of radical atoms, where we can establish intuitive interaction principles for addressing our future abilities to program matter both in the process of transformation and in static-dynamic functionality.

This thesis establishes a path through the development of tools for motion construction in the physical realm. Topobo established the space of kinetic prototyping with a discrete modular system with kinetic memory. With extensive outreach evaluation Topobo, determined that space of research is larger than dynamic physics education for children, and motion design can contribute to new approaches in tangible design thinking. Kinetic Sketchup expanded the space of kinetic prototyping with new dimensions of mechanics, materiality and behavioral control for architecture, product and interaction design and Bosu brought kinetic memory into the realm of soft materials inspiring new design approaches to transformability and form finding. The established systems allow designers to experiment, prototype, and model with programmable kinetic forms, and more over open up novel dimensions of design thinking, engaging new sensibilities for the future of functional and behavioral transformability.





## References

- Ackermann, E. Enactive Representations in Learning: Pretense, Models, and Machines, in Bliss, J., Light, P. and Saljo, R. eds. Learning Sites: Social and technological Contexts for learning, Elsevier, 1999, 144-154.
- Addington, M. and Schodek, D. Smart Materials and Technologies for the Architecture and Design Professions. Oxford: Architectural Press, 2005.
- Aegis Hyposurface, in [http://www.sial.rmit.edu.au/Projects/Aegis\\_Hyposurface.php](http://www.sial.rmit.edu.au/Projects/Aegis_Hyposurface.php)
- Anderson, D., et al., "Tangible Interactions and Graphical Interpretation: A New Approach to 3D Modeling", ACM Siggraph, ISBN: 1-58113-208-5, pp. 393-402, July 2000.
- Arduino Programming Environment: <http://www.arduino.cc/>
- Ashby & Johnson, C. Materials and Design: The Art and Science of Material Selection in Product Design. Butterworth-Heinemann, Oxford, 2002.
- Ball, P. Made to Measure: New Materials for the 21st Century. Princeton: Princeton University Press, 1997.
- Banham, R. Theory and Design in the First Machine Age. New York: Praeger, 1967.
- Blumberg, B. , Downie, M. , Ivanov, Y. , Berlin, M. , Johnson, M. P. , Tomlinson, B. : Integrated learning for interactive synthetic characters. In ACM Trans. on Graphics (2002).
- Braitenberg, Valentino. Vehicles: Experiments in Synthetic Psychology. Cambridge, MIT Press, 1984.
- Breazeal , C., "Towards sociable robots," T. Fong (ed), Robotics and Autonomous Systems, vol. 42(3-4), pp. 167-175. (2003).
- Breazeal, C., Designing Sociable Robots, MIT Press. 2002.
- Brooks, R., "A robust layered control system for a mobile robot, " IEEE J. of Robotics and Automation, vol. 2, no.1 pp. 14-23, 1986.
- Brownell, B. "Transmaterial". Princeton Architectural Press, New-York (2006).

Brooks, R., *Flesh and Machines: How Robots Will Change Us*, New York: Vintage, 2002.

Buxton, W. (2001). *Less is More (More or Less)*, in P. Denning (Ed.), *The Invisible Future: The seamless integration of technology in everyday life*. New York: McGraw Hill.

Cole, M., and Cole, S. *The Development of Children*, Fourth Edition. New York, NY : Worth Publishers, 2001.

Crantz, Galen. *The Chair: Rethinking Culture, Body, and Design*. W. W. Norton & Company; January 2000.

Csikszentmihaly, M. and Rochberg-Halton, E. *The Meaning of Things : Domestic Symbols and the Self*. Cambridge University Press, 1981.

D'Andrea, R. "The Table" and "The Robotic Chair" <http://www.mae.cornell.edu/raff/InteractiveDynamicArt/InteractiveDynamicArt.htm>

DiSalvo, Carl, Gemperie, Francine. "From Seduction To Fulfillment: The Use of Anthropomorphic Form in Design". ACM Press, DPPI 2003.

Dunne, Anthony. *Hertzian Tales: Electronic Products, Aesthetic Experience, and Critical Design*. MIT Press, Cambridge, 2006.

Fenichell, Stephen. *Plastic: the Making of a Synthetic Century*. HarperCollins 1996.

Fitzmaurice, G., Ishii, H., Buxton, W. (1995). *Bricks: Laying the Foundations for Graspable User Interfaces*, Proceedings of Conference on Human Factors in Computing Systems (CHI '95), ACM, Denver, May 1995, pp. 442-449.

Frens, J. W., J. P. Djajadiningrat and Overbeeke, C. J. (2003). *Form, Interaction and Function, An Exploratorium for Interactive Products*. Proceedings of the Journal of the Asian International Design Conference Vol. 1, Tsukuba, Japan, CD-Rom ISSN 1348-7817.

Forlizzi, J. *How robotic products become social products: an ethnographic study of cleaning in the home*. Proceedings of HRI 2007, ACM Press, 2007.

Fox, M., Yeh, B. "Intelligent Kinetic Systems in Architecture."

Managing Interactions in Smart Environments. London: Springer-Verlag, 1999.

Fox, M. "Beyond Kinetic." MIT Kinetic Design Group, 2004.

Frazer, J., , An Evolutionary Architecture. Architectural Association, 1995.

Frei, P., Su, V., Mikhak, B., and Ishii, H. curlybot: Designing a New Class of Computational Toys. Proceedings of CHI 2000. ACM Press (2000).

Gaver, W, Dunne, A., Pacenti, E. "Design: Cultural Probes". Interactions, ACM Press, 1999.

Gropius, Walter, and Wensinger, Arthur, The Theatre of the Bauhaus, Middleton: Wesleyan University Press, 1961.

Haraway, Donna, Simians, Cyborgs, and Women: The Reinvention of Nature. New York: Routledge, Chapman and Hall, 1991.

Hayward Gallery, Force Fields: Phases of the Kinetic. Manchester: Cornerhouse, 2000.

Hoberman, Chuck. Emergent Surface, 2008. <http://www.hoberman.com>

Hodge, Brooke, ed. Skin & Bones, Parallel Practices in Fashion and Architecture, Thames and Hudson, New York, 2006.

Holt, Steven & Skov, Maria. Bobjects and Beyond: The New Fluidity in Design, Chronicle Books, San Francisco, 2005.

Ishii, H, and Ullmer, B., Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms. Proceedings of CHI 1997, ACM Press, (1997).

Ishii, H., Ren, S., Frei, P. Pinwheels: Visualizing Information Flow in an Architectural Space. Proceedings of CHI 2001, ACM Press (2001).

Iwata, H., Yano H., Nakaizumi, F. and Kawamura, R. Project FEELEX: adding haptic surface to graphics . Proceedings of Siggraph 2001, ACM Press (2001).

Johansson, G. Visual Perception of Biological Motion and a Model for its Analysis. Perception & Psychophysics, Vol. 14

(2); 201-211 (1973).

Johnson, M.P., Wilson, A., Kline, C., Blumberg, B., Bobick, A.  
“Sympathetic Interfaces: Using a Plush Toy to Direct Synthetic  
Characters.” Proceedings of the CHI 99 Conference on  
Human Factors in Computing Systems, pp. 152-158.

Johnson, S. Emergence: The Connected Lives of Ants, Brains,  
Cities, and Software, New York: Touchstone, 2001.

Jormakka K., Flying Dutchmen: Motion in Architecture,  
Basel: Birkhauser, 2002

Krauss, R.E. Passages in Modern Sculpture. Cambridge,  
MA: MIT Press, 1981.

Lerner, M., Outerspace: Reactive robotic creature. 2005  
<http://www.andrestubbe.com/outerspace/>

LeClerc, V., Parkes, A. Ishii, H. Senspectra: A  
Computationally Augmented Physical Modeling Toolkit for  
Sensing and Visualization of Structural Strain. Proceeding of  
Conference on Human Factors in Computing Systems (CHI  
'07) San Jose, April 2007.

Lifton, J., et al., “Pushpin Computing System Overview: A  
Platform for Distributed, Embedded, Ubiquitous Sensor  
Networks.”, Pervasive 2002, LNCS 2414, pp. 139-151, 2002.

Loos, Adolf. “Ornament and Crime”, Ariadne Press, 1997.  
(original publication 1908).

Loyall, B., Scott, W., Bates, J., Weyhrauch, P. “System for  
authoring highly interactive, personality-rich interactive  
characters.” Proceedings of the 2004 ACM SIGGRAPH/  
Eurographics symposium on Computer animation (2004),  
pp. 59-68.

Llunblad, S. & Holmquist, L. E. “Transfer scenarios:  
grounding innovation with marginal practices” Proceedings  
of CHI 2007, (p.737 - 746) ACM Press, 2007.

Lupton, E. Skin: New Design Organics. in Skin: Surface,  
Substance and Design. New York: Princeton Architectural  
Press, 2002.

McCullough, Malcolm, Abstracting Craft: The Practiced  
Digital Hand, Cambridge: the MIT Press, 1996.



Michalowski, M., Sabanovic, S., Kozima, H. A dancing robot for rhythmic social interaction. Proceedings of HRI 2007, March, 2007.

Miyake, I & Fujiwara, D. A-POC Making, Vitra Design Museum Catalog, 2004.

Moholy-Nagy, L. Vision in Motion. Chicago: Paul Theobald & Co., 1965.

Mori, Masahiro. Bukimi no tani the uncanny valley. Energy, 7, 33-35. 1970.

Pangaro, G., Maynes-Aminzade, D., and Ishii, H., The Actuated Workbench: Computer-Controlled Actuation in Tabletop Tangible Interfaces. Proceedings of UIST 2002, ACM Press (2002).

Parkes, A., Poupyrev, I, Ishii, H. Dynamic Physical Media: Actuation and Reconfigurability in Interaction Design. Communication of the ACM, Special Issue on Organic User Interfaces. May 2008, in press.

Parkes, A., Raffle, H., Ishii, H. Topobo in the Wild: Longitudinal Evaluations of Educators Appropriating a Tangible Interface. Proceedings of Conference on Human Factors in Computing Systems (CHI 2008). Florence, Italy, April 2008.

Parkes, A., LeClerc, V., Ishii, H., Glume: Exploring Materiality in a Soft Augmented Modeling System, Extended Abstracts of Conference on Human Factors in Computing Systems (CHI '06) Montreal, April 2006.

Patten, J. Ishii, H. Mechanical Constraints as Computational Constraints. Proceedings of CHI 2007. ACM Press (2007).

Perlin, K. "Real Time Responsive Animation with Personality," IEEE Transactions on Visualization and Computer Graphics, 1(1) (1995).

Piper, B., Ratti, C., and Ishii, H., Illuminating Clay: A 3-D Tangible Interface for Landscape Analysis. Proceedings of CHI 2002, ACM Press (2002).

Poupyrev, I., Nashida, T., Maruyama, S., Rekimoto, J., Yamaji, Y. Lumen: Interactive Visual and Shape Display for Calm Computing. SIGGRAPH 2004 Emerging

- Technologies, ACM Press (2004).
- Popper, F., Origins and Development of Kinetic Art. Greenwich, CT. New York Graphic Society: 1968.
- Raffle, H., Parkes, A. and Ishii, H., Topobo: A 3-D Constructive Assembly with Kinetic Memory. Proceedings of CHI 2004, ACM Press (2004).
- Raffle, H., Parkes, A., Lifton, J, Ishii, H. Beyond Record and Play — Backpacks: Tangible Modulators of Kinetic Behavior. Proceedings of Conference on Human Factors in Computing Systems (CHI 2006). Montreal, April 2006.
- Raffle, H., Yip, L., and Ishii, H. 2007. Remix and robo: sampling, sequencing and real-time control of a tangible robotic construction system. In ACM SIGGRAPH 2007 Educators Program (San Diego, California, August 05 - 09, 2007). SIGGRAPH '07. ACM, New York, NY, 35.
- Ramos, V.R. On the Implicit and on the Artificial: Morphogenesis and Emergent Aesthetics in Autonomous Collective Systems, from Architopia Book/Catalogue, Art, Architecture and Science. Maubant, J.L, Moura, L., (Eds.) pp.25-57, Portuguese National Science Foundation, February 2002.
- Resnick, Martin, Berg, et al. "Digital Manipulatives: New Toys to Think With". Proceedings of CHI 1998, ACM Press (1998).
- Rose, C., Bodenheimer, B. Cohen, M.: 'Verbs and Adverbs: Multidimensional Motion Interpolation Using Radial Basis Functions. IEEE Computer Graphics and Applications 18, 5 (1998), 32-40.
- Sims, K. "Evolving Virtual Creatures." Computer Graphics (Siggraph '94 Proceedings), July 1994, pp.15-22.
- Taylor, M. (ed), Surface Consciousness, London: Wiley Academy, 2003.
- Terzidis, K. Expressive Form: A Conceptual Approach to Computational Design. London: Spon Press, 2003.
- Testa, Peter. Carbon fiber prototype tower. <http://www.peter-testa.com/>
- Tilley, A. & Henry Dreyfuss Associates. The Measure of Man

and Woman: Human Factors in Design. John Wiley and Sons, New York, 2002

Turkle, S. The Second Self: Computers and the Human Spirit. the MIT Press, Cambridge, 2005.

Turkle, S. and Papert, S. Epistemological Pluralism and the Reevaluation of the Concrete. *Journal of Mathematical Behavior*, Vol. 11, No.1, in March, 1992, pp. 3-33

Vertegaal, R. Organic User Interfaces (OUI): Designing Computers in Any Way Shape or Form from Unconventional Computation. Heidelberg: Springer Berlin, 2007.

Watanabe, R., et al., "Implementation of ActiveCube as an Intuitive 3D Computer Interface", *Proc. of 4th International Symposium on Smart Graphics*, pp. 43-53, (May. 2004).

Yim, M., et al., Rhombic Dodecahedron Shape for Self-Assembling Robots. Xerox PARC SPL TechReport P9710777, 1997.

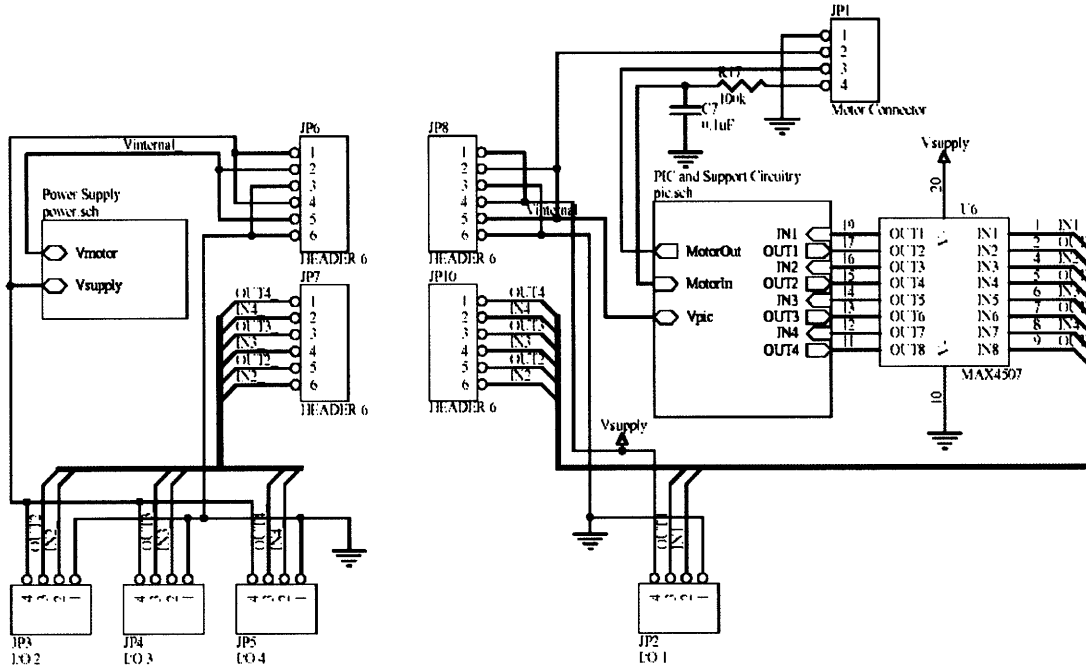
Yim, M. Duff, D. Roufas, C. PolyBot: a Modular Reconfigurable Robot, *IEEE Intl. Conf. on Robotics and Automation*, (2000) San Francisco, CA.

Zuk, W, and Clark, R., *Kinetic Architecture*. New York: Litton Educational Publishing, Inc., 1970.



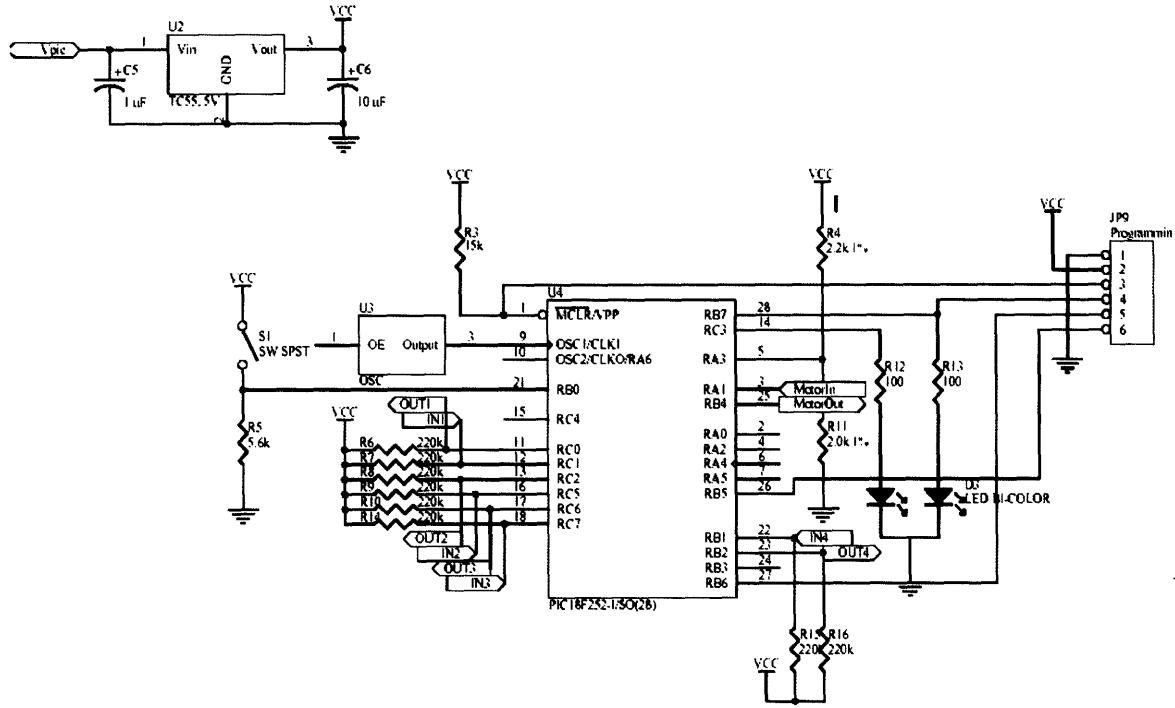
# Appendix A: Hardware & Software Documentation

## Topobo & Kinetic Sketchup PCB Schematic

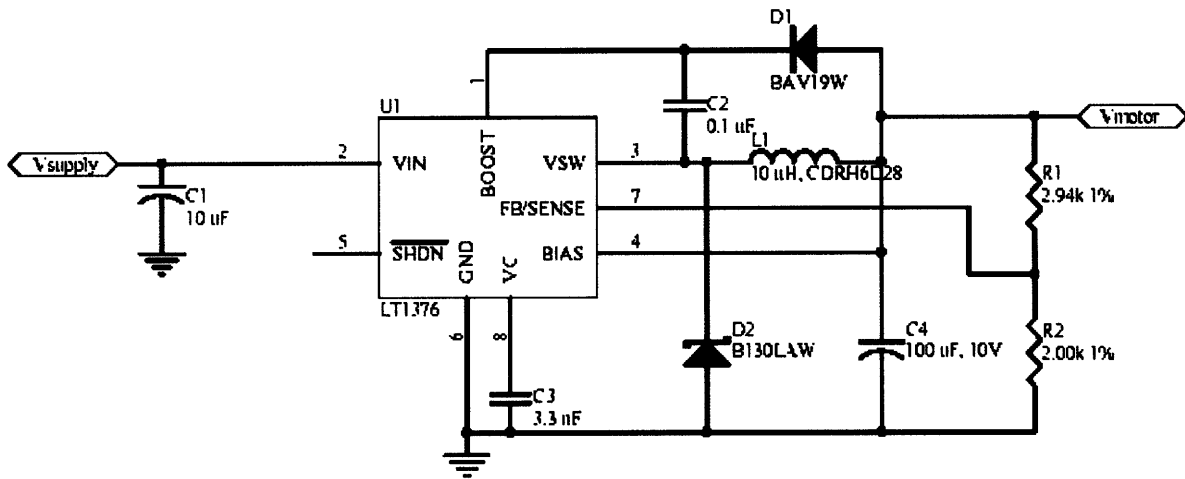




# Topobo & Kinetic Sketchup PCB Schematic



# Topobo & Kinetic Sketchup PCB Schematic



## Topobo & Kinetic Sketchup Firmware

```
/* updated November 2008 by Andy Gossling & Amanda Parkes  
    Added code for Kinetic Sketchup - speed, hinge, twitter, accel & pot  
*/
```

```
/* updated october 2007 hayes raffle
```

### *TO DO*

*realized that reading/writing to external or internal eeprom is extremely slow -  
about 2ms per read/write.  
could optimize by running most functionality from RAM and doing  
background, interrupt-driven writes of  
critical data during runtime. such data would be retrieved on startup.*

### *TIMING*

*main loop:  
7 us when nothing is happening  
15 us when RTC (sig\_output\_compare0) gets called*

*queen messages:  
process a queen message: 5ms  
due to debug log, every 2.5 sec it processes no messages for 250 ms!*

*RTC: 4 us when nothing is happening.*

### *COMM:*

*handshake returned: 20 us, only one channel connected  
handshake complete: 50 us  
bit\_low\_time: 20 us  
bit\_high\_time: 30 us  
handshake timeout: 180 us*

```
*/
```

```
/*updated on January 31st by Michael Fleder  
    Added code for remix interface  
    Added remix.c, remix.h  
    Changes to backpack.c, backpack.h, comm.c, comm.h, main.h, main.c, servo.c,  
servo.h
```

*Restructured the memory storage to make the distinction between recording and playing back*

*(playback and record could involve different memory banks)*

*Most changes are marked with an MF comment*

*NOTE: The message enumeration should be standardized so that all parts have the same message enumerations.*

*\*/*

```
#include <inttypes.h>
#include <avr/io.h>
#include <avr/interrupt.h>
//#include <avr/signal.h>
#include <stdio.h>
#include "servo.h"
#include "util.h"
#include "button.h"
#include "main.h"
#include "spi_eeprom.h"
#include "comm.h"
#include "messages.h"
#include "backpack.h"
#include "remix.h"
#include "debugLog.h"
```

```
uint8_t allegiance = DEFAULT_ALLEGIANCE;
```

```
//eeprom bank variables
extern uint16_t default_bank;
extern uint16_t start_green_address;
extern uint16_t start_teal_address;
extern uint16_t start_orange_address;
extern uint16_t start_blue_address;
```

```
extern uint8_t servo_state;
extern uint8_t should_play_servo_data;
uint8_t dont_record_yet;
```

```

uint8_t just_started_recording;
uint8_t old_data;
uint8_t new_data;

extern uint32_t play_servo_memory_position_precise; //MF Where you are in play-
back (an offset from some base memory address)
extern uint16_t *play_servo_memory_position; //MF
extern uint32_t record_servo_memory_position_precise; //MF Where you are in
recording (offset)
extern uint16_t *record_servo_memory_position; //MF
uint8_t normalized_pos;

extern uint8_t playFromColors;
extern uint8_t recordFromPlayback;
extern uint8_t colorRecording;
extern uint8_t dockHost;
extern uint8_t dockChannel;
extern uint8_t dockValue; //value of the potentiometer

extern uint16_t servo_memory_position_delay, // phase backpack adjustment
record_length_playing, // the length of the current playback //MF
record_length_recording; //the length of the current recording //MF

extern uint16_t servoMemoryPositionIncrement;
extern uint16_t servoMemoryPositionDecrement;
extern uint16_t dockMemoryPositionIncrement;
extern uint16_t dockMemoryPositionDecrement;
extern uint8_t dockExists;
extern uint8_t roboExists;

uint16_t servo_mem_pos_delay = 0;

extern uint16_t play_curr_eeprom_bank_addr;
extern uint16_t record_curr_eeprom_bank_addr;

uint8_t should_stream_data = FALSE;

```

```

// backpack data
extern uint8_t phase_backpack_value;
extern uint8_t playback_amplitude_value;
extern int8_t pos_off_backpack_value;
extern uint8_t pos_backpack_value;

uint8_t checkBackpack = TRUE;

//
extern volatile uint8_t is_ADC_done;
extern uint8_t recvd_pos;
extern uint8_t firstQueenData;
extern uint8_t followerOrigin;

extern volatile uint16_t button_timer;
extern volatile uint8_t button_status;
extern uint8_t button_disabled;
volatile uint8_t last_button = 0;
volatile uint8_t servo_timer = 0;
volatile uint16_t force_calibration_timeout=0;
volatile uint16_t ping_timer = 1;
volatile uint16_t delay_ms_nonblocking = 0;
extern volatile int16_t backpack_timer;
uint8_t remoteDebounceTimer = 0;
uint8_t remoteDebounce = FALSE;
uint8_t RTCflag = FALSE;
extern uint8_t good_channels;
extern uint8_t stream_motor_data_to_PC;
extern uint8_t parentChannel;
uint16_t retry_counter = 0;

#define NORMAL_SPEED 256
#define HALF_SPEED 128
#define DOUBLE_SPEED 512

//bootloader functions
void testBootloader(void);
void goto_boot(void);

// gets called once per 100us, for fine timing stuff like random backoff retries

```



```

ISR(FINE_TIMER)
{
    //this interrupt takes 8us to execute
    randomBackoffDecrementTimeStamps();
}

// gets called once per ms, for general purpose timing stuff (like button)
// haze : put the high priority stuff first, and for low priority, set flags and handle
// them in the mainloop

ISR(GENERIC_TIMER) {
    uint8_t curr_button;
    /*****
SERVO
*****/
    // put this first for precision.

    if (servo_timer > SERVO_SAMPLE_PERIOD)
        servo_timer = 0;
    if (servo_timer == 0) {
        // do the servo thing
        switch (servo_state) {
            case SERVO_RECORD:
                startADC(); // start the ADC, will take a while.
                // ADC interrupt will be called when completed
                break;
            case SERVO_PLAYBACK:
                should_play_servo_data = TRUE;
                break;
            case SERVO_IDLE:
                break; // do nothing;
        }
    }
    servo_timer++;

    /*****
TIMEOUTS - in mainloop
*****/
    RTCflag = TRUE;

    /*****

```

## **BUTTON**

```
*****  
if (! button_disabled) {  
    if (button_timer != 0xFFFF) // start counting, but don't let it roll over!  
        button_timer++;  
  
    curr_button = BUTTON_PRESSED; //poll the button pin  
    // did something just change?  
    if ((button_timer >= BUTTON_DEBOUNCE) && (curr_button != last_  
button))  
    {  
        last_button = curr_button;  
        if (curr_button) { // ie just pressed  
            RED_LED_ON; // a hack to give UI feedback that you did  
indeed press the button  
            if (button_status == FALSE) {  
                button_timer = 0; // reset timer to prep for  
release  
                //uart_putchar('v');  
            }  
            else if ((button_status == SINGLE_CLICK) && (button_  
timer < DOUBLE_CLICK_TIMEOUT)) {  
                button_timer = 0; // ok second doubleclick press  
                //uart_putchar('W');  
            }  
        }  
        else { // most stuff is done on the release....  
            if (button_status == FALSE) // just released from single  
click  
            {  
                button_status = SINGLE_CLICK; // clicked once  
                button_timer = 0; // don't let click again too  
quickly  
                //uart_putchar('^');  
            }  
            else if ((button_status == SINGLE_CLICK) && (button_  
timer < DOUBLE_CLICK_TIMEOUT)) {  
                // just released from double click  
                button_status = DOUBLE_CLICK;  
                RED_LED_OFF; // hack!  
                button_timer = 0;  
            }  
        }  
    }  
}
```



```

        eraseDebugLog(); //erase internal eeprom statistics
        initializeRecordingBanks(); // erase recording banks for remix + robo
        checkCalibration(TRUE); // force calibration
    }
    else
        checkCalibration(FALSE); // use internal memory

    //if it's these characters, it's blank, i.e. never been set
    if (getUniqueID() == 0xffff) {
        generateUniqueID();
    }

    //loadDebugLog();

    // this will make gcc complain but its the right thing:
    // s_m_p_p is a 32 bit int (16 bits of precision data)
    // and s_m_p is just the high 16 bits (not precision)
    // so modifying one will automatically modify the other
    play_servo_memory_position = (uint16_t *)&play_servo_memory_position_pre-
cise + 1; //MF
    *play_servo_memory_position = 2; //MF

    //Note about the above 2 lines: s_m_p is made a pointer so that it's memory ad-
dress (which never changes in this program) is
    //the high 16 bits of s_m_p_p. For normal playback, s_m_p is incremented by
one. To scale playback, the low 16 bits
    //of s_m_p_p are incremented by some value. When the low 16 bits roll over into
the upper 16 (which is s_m_p),
    //s_m_p is incremented and playback advances.
    //Thus, incrementing the low 16 bits can control the speed of playback. MF

    record_servo_memory_position = (uint16_t *)&record_servo_memory_position_
precise + 1; //MF
    *record_servo_memory_position = 2; //MF

    enableButton(); // set up the button interrupt
    readBankAddresses();
    readColorLengths();

    //init these to normal local recording ***
    play_curr_eeprom_bank_addr = default_bank; //MF

```

```

record_curr_eeprom_bank_addr = default_bank;

// alert neighbors you exist
sendOneByteMsg(ALL_CHANNELS, PING_MESSAGE);
sendOneByteMsg(ALL_CHANNELS, NEW_NODE_CONNECTED);

// the main loop
while (1) {
before    randomBackoffRetry(); //retry sending anything that we failed to send

    processAllPending(); //process all messages waiting to be processed

    uint8_t incoming_channel;
    if ((incoming_channel = checkAllChannels()) != 0)
    {
        ReceiveMessagePacket incomingPacket =
receiveMessage(incoming_channel);
        //putstring("\n Just TRIED to receive a message on channel: ");
putnum_ud(incoming_channel);
        //putstring("\nmsg ID = "); putnum_ud(incomingPacket.message[0]);

        if (incomingPacket.messageLength > 0)
        {
            //putstring("\n about to process incomingPacket\n");
            processReceivedMessage(incomingPacket);
        }
    }

// TIMEOUTS
if (RTCflag == TRUE) {
    RTCflag = FALSE;
    doTimeouts();
}

b = getButton();
if (b == SINGLE_CLICK) {

    //putstring("click\n\r");

```

```

        sendOneByteMsg(ALL_CHANNELS, BUTTON_PRESS_MESSAGE);

        switch (servo_state) {
            case SERVO_IDLE:
                if (allegiance == QUEEN) {
                    sendFourByteMsg(ALL_CHANNELS, RECORD_
FROM_QUEEN_MESSAGE,
(int)phase_backpack_value,
(int)(UNITY_FREQUENCY_GAIN/MAX_FREQUENCY_GAIN),
UNITY_AMPLITUDE_GAIN);
// Stream data only when recording starts with
queen.
                    should_stream_data = TRUE;
                }
                else { //you're independent
                    sendOneByteMsg(ALL_CHANNELS, RECORD_
MESSAGE);
                }

                colorRecording = 0; //MF which token recording to
0 = Not recording at all, or in this case recording but not dock related
                record_curr_eeprom_bank_addr = default_bank; //MF
NOT a color start address, is a regular address

                startServoRecord(); //takes 3 ms, does not block inter-
rupts
                break;

            case SERVO_RECORD:
                servoMemoryPositionIncrement = NORMAL_SPEED; //reset speed
                sendOneByteMsg(ALL_CHANNELS, PLAYBACK_MES-
SAGE);
                playFromColors = FALSE; //MF maybe make this line
part of stopServoPlayback()
                startServoPlayback(); //takes 8 ms, does not block inter-
rupts
                break;

```



```

    case SERVO_PLAYBACK:
        sendOneByteMsg(ALL_CHANNELS, STOP_MES-
SAGE);

        playFromColors = FALSE;    //MF

        if(recordFromPlayback)    //MF
        {
            recordFromPlayback = FALSE; //*****If
things don't work w/ button presses and the dock, check this IF statement
        }
        startServoIdle();
        break;

        default:
            //putstring("Strange servo state\n\r");
            halt();
        }
    }

    else if (b == DOUBLE_CLICK) {

        //putstring("kaclick\n\r");
        if ((servo_state == SERVO_IDLE) && (is_data_initialized() )){
            sendOneByteMsg(ALL_CHANNELS, PLAYBACK_MES-
SAGE);

            //servoMemoryPositionIncrement = NORMAL_SPEED;           //reset speed

            if (allegiance == QUEEN)
                allegiance = FOLLOWERQUEEN; //don't stream
data, no one's the boss any more

            startServoPlayback();
        }
    }

```

```

else if(servo_state == SERVO_PLAYBACK)
{
    switch(servoMemoryPositionIncrement){
        case NORMAL_SPEED:
            servoMemoryPositionIncrement = HALF_SPEED;
            break;
        case HALF_SPEED:
            servoMemoryPositionIncrement = DOUBLE_SPEED;
            break;
        case DOUBLE_SPEED:
            servoMemoryPositionIncrement = NORMAL_SPEED;
            break;
        default:
            break;
    }
}

}

}

/*
 * Retrieve and store incoming data from the servo.
 */
if (is_ADC_done) {
    uint8_t normalized_data;

    is_ADC_done = FALSE;

    // get the ADC value and normalize it before storing it
    if ((allegiance == FOLLOWER) || (allegiance == FOLLOWERQUEEN))
    {
        //normalized_data = recvd_pos;

        int16_t temp;
        // set the first position
        if (firstQueenData == TRUE) {
            firstQueenData = FALSE;
            queenFirstPos = recvd_pos;
            temp = followerOrigin;

```

```

    }

    else //set the new position
        temp = (recvd_pos - queenFirstPos) + followerOrigin;

    if (temp > 255)
        temp = 255;

    else if (temp < 0)
        temp = 0;

    normalized_data = temp;

    // ok, set the servo!
    // code doesn't handle phase bp during record.
    setServoPosition(denormalize(normalized_data));

}
else {
    normalized_data = normalizeADC(ADC);
}

/*
string(":", "");
    putnum_uh(*servo_memory_position+curr_eeprom_bank_addr); put-

    putnum_uh(normalized_data); uart_putchar(' ');
    putstring("\n\r");
    */

// check to see if this new data is different than the old data.
// if it is, start saving the data, and tell the neighbors to start saving it too
// putstring("\r\n d_r_y = "); putnum_uh(dont_record_yet);

//decrease the resolution of the data
uint8_t diff;
#define NOISE_THRESH 3 // 3 is ok with COEFF 700 in servo.c ...
if (dont_record_yet == TRUE) {
    // is this the first data?
    if (just_started_recording == TRUE) {
        just_started_recording = FALSE;
        old_data = normalized_data;
    }
}

```

```

        new_data = normalized_data;
    }
    else {
        old_data = new_data;
        new_data = normalized_data;
    }
    //putstring(“ , old = “); putnum_ud(old_data);
    putstring(“ , new = “); putnum_ud(new_data);
    diff = abs(old_data - new_data);
    if (diff > NOISE_THRESH) {
        putstring(“ changed! “);
        dont_record_yet = FALSE;
        sendOneByteMsg(ALL_CHANNELS, USER_IS_NOW_
MOVING_TOPOBO_MESSAGE);
    }
}

// minimum length a recording should be
#define MINIMUM_RECORD_LENGTH 7

if ((dont_record_yet == FALSE) || (*record_servo_memory_position <
MINIMUM_RECORD_LENGTH)) { //we should record
    extEepromWrite(record_curr_eeprom_bank_addr + *record_ser-
vo_memory_position, normalized_data);

}

// stream data to the PC, if requested
if (stream_motor_data_to_PC == TRUE) {
    uint16_t mem_pos, q;
    uint8_t mem_pos_msb, mem_pos_lsb, uniqueID_MSB, uniqueID_
LSB;

    q = getUniqueID();
    uniqueID_MSB = (q >> 8) & 0xFF;
    uniqueID_LSB = q & 0xFF;
    mem_pos = (uint16_t) (play_curr_eeprom_bank_addr + *play_
servo_memory_position);

    mem_pos_msb = (mem_pos >> 8) & 0xFF;
    mem_pos_lsb = mem_pos & 0xFF;

```

```

        /*loadTxMsgBuff(MOTOR_VALUE_MESSAGE);
        loadTxMsgBuff(uniqueID_MSB);
        loadTxMsgBuff(uniqueID_LSB);
        loadTxMsgBuff(mem_pos_msb); //where we are in memory
        loadTxMsgBuff(mem_pos_lsb); //where we are in memory
        loadTxMsgBuff(normalized_data); //unmodulated motor data we
played last time*/

        sendSixByteMsg(parentChannel, MOTOR_VALUE_MESSAGE,
                        uniqueID_MSB,
                        uniqueID_LSB,
                        mem_pos_msb,
                        mem_pos_lsb,
                        normalized_data);
    }

    //allow PC to request pos data during recording
    normalized_pos = normalized_data;

    if (allegiance == QUEEN) {
        if (should_stream_data) {
            sendTwoByteMsg(ALL_CHANNELS, QUEEN_POSITION_
MESSAGE, normalized_data);
        }
    }

    if ((dont_record_yet == FALSE) || (*record_servo_memory_position <
MINIMUM_RECORD_LENGTH)) //we should record
        (*record_servo_memory_position)++;

    if (*record_servo_memory_position >= SERVO_RECORD_BUFFER_
SIZE) {
        if (allegiance == QUEEN) {
            sendOneByteMsg(ALL_CHANNELS, PLAYBACK_MES-
SAGE);
        }
        startServoPlayback();
    }
}

if (should_play_servo_data) { // ie are we playing a motion?

```

```

should_play_servo_data = FALSE;
RED_LED_OFF;

if (backpackPresent(GLOBAL_FREQUENCY_BACKPACK) ||
    backpackPresent(GLOBAL_AMPLITUDE_BACKPACK) ||
    backpackPresent(GLOBAL_POSITION_BACKPACK) ||
    backpackPresent(GLOBAL_POSITION_OFFSET_BACKPACK)) {

    RED_LED_ON;
}
else {
    RED_LED_OFF;
}

// basic functionality (queen does whats in eeprom and streams rest)
if ((allegiance == QUEEN) && !playFromColors && !recordFromPlay-
back &&

    !(backpackPresent(GLOBAL_FREQUENCY_BACKPACK) ||
    backpackPresent(GLOBAL_AMPLITUDE_BACKPACK) ||
    backpackPresent(GLOBAL_POSITION_BACKPACK) ||
    backpackPresent(GLOBAL_POSITION_OFFSET_BACKPACK)))
{

    // check whether there is a position backpack, if so it overrides
    // the internal storage
    // (untested...didnt have a positionbackpack)
    if (backpackPresent(POSITION_BACKPACK))
    {
        normalized_pos = pos_backpack_value;
    }
    else
    {
        normalized_pos = extEepromRead(play_curr_eeprom_
bank_addr + *play_servo_memory_position);
    }
    //uart_putchar('.');
    setServoPosition(denormalize(normalized_pos));
    // loop to beginning (skip first 2 bytes -- record length)

    if (*play_servo_memory_position == 2+servo_mem_pos_delay)

```



```

        {
            if (!dockExists)
//          if (!playFromColors || !dockExists) //if you are playing from colors you don't want to
//          synchronize b/c the dock will tell you when to do so
{          //... for example, if there are multiple colors being played the dock will tell you which one
//          to start next
                sendThreeByteMsg(ALL_CHANNELS, PLAYBACK_SYNC_MESSAGE,

(uint8_t)(record_length_playing >> 8),

(uint8_t)(record_length_playing));
        }
    }
    if (*play_servo_memory_position >= record_length_playing - 1)
    {
        *play_servo_memory_position = 2;
        //putstring("\n\r");
    }
    else
    {
        (*play_servo_memory_position)++;
    }
}

else {
    // you're not a Queen. read recording from memory

    if(*play_servo_memory_position >= record_length_playing)
//MF
    {
        *play_servo_memory_position = record_length_playing -
1;
        } //the above if catches bugs: never want playSMP to be bigger
//then record_length since we're about to read from the eeprom

        normalized_pos = extEepromRead(play_curr_eeprom_bank_addr
+
        *play_servo_memory_position);

```

```

memory bank 1.
// stream data to the PC, if requested. won't work with remix - only
if (stream_motor_data_to_PC == TRUE) {
    uint16_t mem_pos, q;
    uint8_t mem_pos_msb, mem_pos_lsb, uniqueID_MSB,
uniqueID_LSB;

    q = getUniqueID();
    uniqueID_MSB = (q >> 8) & 0xFF;
    uniqueID_LSB = q & 0xFF;

    mem_pos = (uint16_t) (play_curr_eeprom_bank_addr +
*play_servo_memory_position);

    mem_pos_msb = (mem_pos >> 8) & 0xFF;
    mem_pos_lsb = mem_pos & 0xFF;

    /*loadTxMsgBuff(MOTOR_VALUE_MESSAGE);
loadTxMsgBuff(uniqueID_MSB);
loadTxMsgBuff(uniqueID_LSB);
loadTxMsgBuff(mem_pos_msb); //where we are in memory
loadTxMsgBuff(mem_pos_lsb); //where we are in memory
loadTxMsgBuff(normalized_pos); //unmodulated motor
data we played last time*/
sendSixByteMsg(parentChannel, MOTOR_VALUE_MES-
SAGE,
uniqueID_
MSB,
uniqueID_
LSB,
mem_pos_
msb,
mem_pos_lsb,
normalized_
pos);
}

if(recordFromPlayback) //MF
{
    extEepromWrite(record_curr_eeprom_bank_addr + *re-

```

```

cord_servo_memory_position,
                                normalized_pos); //MF Store the current playback
position in the color memory location

                                if(++(*record_servo_memory_position) == SERVO_RE-
CORD_BUFFER_SIZE) //increment the rec. pos. and check if hit the max
                                {
the dock to stop
                                if(dockHost) //If we're the dockHost we need to tell
                                {
                                sendOneByteMsg(dockChannel, STOP_
MESSAGE);
                                }
                                stopRecordFromPlayback(); //MF
                                }
                                }

// PLAYBACK AMPLITUDE SCALING ////////////////////////////////////////
// scale the recorded data with the playback_amplitude_value
// this isnt possible with the position backpack because
// it relies on the first position to scale
if(playback_amplitude_value != UNITY_AMPLITUDE_GAIN) {
    int32_t temp32;
    uint8_t origin_pos;

    origin_pos = extEepromRead(play_curr_eeprom_bank_
addr + 2 + servo_mem_pos_delay);

    if(normalized_pos >= origin_pos) {
        temp32 = (int16_t)normalized_pos - (int16_t)ori-
gin_pos;
        temp32 *= playback_amplitude_value * MAX_AM-
PLITUDE_GAIN;

        temp32 >>= 8;
        if ( (temp32+origin_pos) > 0xff) {
            temp32 = 0xff;
        }
        else {
            temp32 += origin_pos;
        }
    }
}

```

```

else {
    temp32 = (int16_t)origin_pos - (int16_t)normal-
    ized_pos;
    temp32 *= playback_amplitude_value * MAX_AM-
    PLITUDE_GAIN;
    temp32 >>= 8;

    if (temp32 > origin_pos) {
        temp32 = 0;
    }
    else {
        temp32 = origin_pos - temp32;
    }
}

/*
    putstring(" (");
    putnum_ud(normalized_pos);
    putstring("->");
    putnum_ud(temp32);
    putstring("\n\r");
*/
normalized_pos = temp32;
}

```

**// POSITION OFFSET SCALING //**

```

if (pos_off_backpack_value != 0) {
    int16_t temp;

    temp = normalized_pos + pos_off_backpack_value;
    if (temp > 255)
        temp = 255;
    else if (temp < 0)
        temp = 0;

    normalized_pos = temp;
}

```

**// POSITION BACKPACK OVERRIDE //**

```

if (backpackPresent(POSITION_BACKPACK)) {
    normalized_pos = pos_backpack_value;
}

```

```

    }

    // ok set it! ////////////////////////////////////////////////////////////////////
    temp = denormalize(normalized_pos);

    setServoPosition(temp);

    //uart_putchar(' ');
    //putnum_uh(temp);

    // the increment contains 16 bits of 'floating point' precision
    // by default this is UNITYFREQGAIN.

    if(dockExists || roboExists) //MF, HSR -- allows you to use the joysticks before
    copying the recording to a color
    {
        //dockMPI and dockMPD are set when a message is re-
        ceived from the dock
        servoMemoryPositionIncrement = dockMemoryPositionIn-
        crement;
        servoMemoryPositionDecrement = dockMemoryPosition-
        Decrement;
    }

    play_servo_memory_position_precise += ((uint32_t)servoMemo-
    ryPositionIncrement << 8);

    if(playFromColors
        && dockExists
        && (dockValue >= 128)
        && (play_servo_memory_position_precise <
        ((uint32_t) servoMemoryPositionDecrement << 8))) //MF
    {
        //above if: (are we going backwards) && (reachedThebe-
        ginning of a token) -->
        //hold here until we hear about the next token
        if(dockHost)
        { //tell the dock that this has happened and wait for the dock
        to send info about what the next token to play is
            sendTwoByteMsg(dockChannel, HOST_LED_MES-
            SAGE, 0);
        }
    }
}

```

```

        else if(roboExists && (play_servo_memory_position_precise <
((uint32_t) servoMemoryPositionDecrement << 8)))
// you're playing backwards... this is to rollover from the beginning to the end of the record
    {
        *play_servo_memory_position = record_length_playing - 1;
    }
    else //MF
    { //otherwise keep going backwards
        play_servo_memory_position_precise -= (uint32_t) ser-
voMemoryPositionDecrement << 8;
    }

//putnum_ud(servo_memory_position_precise >>16); uart_putchar('.');
//putnum_ud(servo_memory_position_precise & 0xFFFF); uart_putchar(' ')
//putnum_ud(servo_mem_pos_increment); uart_putchar(' ');

//if going forwards && have reached the end
if (*play_servo_memory_position >= record_length_playing)
{
    if(playFromColors
        && dockExists
        && (dockValue < 128)) //the !playFromColors is
not necessary?
    {
        if(dockHost) //tell the dock we've hit the end, wait
for info about what to do next
        {
            sendTwoByteMsg(dockChannel, HOST_
LED_MESSAGE, DOCK_LEDS_PER_QUAD -1);
        }
    }
    else
    {
        *play_servo_memory_position = 2;
        if (*play_servo_memory_position == 2+servo_mem_pos_delay)
        {
//make the local frequency backpack not mess up syncing

```



```

        //only send the sync signal if you're not changing the local frequency
        if ( /*(!playFromColors)*/ (!dockExists) &&
            ((servoMemoryPositionIncrement == UNITY_FREQUENCY_GAIN) ||
             backpackPresent(GLOBAL_FREQUENCY_BACKPACK))) //MF
        {
            sendThreeByteMsg(ALL_CHANNELS, PLAYBACK_SYNC_MESSAGE,
                             (uint8_t)(record_length_playing >> 8),
                             (uint8_t)(record_length_playing));
        }
    }
}

```

```

void doTimeouts(void) {
    // backpack removal
    backpack_timer++;
    if ((backpack_timer & 0xF) == 0) {
        checkBackpackTimeout();

        //testing the ADC for electrical range of pot
        //uint16_t tt;
        /*
        low end: ~300, high end (clockwise) ~1000
        tt = readADC_blocking();
        putstring("\n\n ADC: ");
        putnum_ud(tt);

```

```

    putstring("\r\n");
        */
        //testBootloader();
    }

    // remote (network) debouncing
    if (remoteDebounceTimer != 0)
        remoteDebounceTimer--;
    if (remoteDebounce == TRUE) {
        disableButton(); //is it bad to do this so many times?
        if (remoteDebounceTimer == 0) {
            remoteDebounce = FALSE;
            enableButton();
        }
    }
}

// save the debug log
#ifdef DEBUG
    if (ping_timer == PING_TIMER_RELOAD) {
        saveDebugLog();
        putstring(" gc=0x"); putnum_uh(good_channels);
    }
}
#endif

// PING - goodChannels will get refreshed on handshake timeouts when this message is
sent.
    if (ping_timer++ > PING_TIMER_RELOAD) {
        ping_timer = 0;
        sendOneByteMsg(ALL_CHANNELS, PING_MESSAGE);
        //DEBUGGING STUFF: sendOneByteMsg(ALL_CHANNELS, ORANGE_MES-
SAGE);
    }
}

void initRTC2(void) {
    // setup RTC2 (interrupt 2) to run @ 100us per interrupt
#ifdef IS_644
    TCCR2A = _BV(WGM21); // CTC mode
#endif
}

```

```

        TCCR2B |= 0x3; // clock prescale by 64
        OCR2A = 25; // to make 10KHz clock
        TIMSK2 |= _BV(OCIE2A); // interrupt on compare
    #else
        TCCR2 = _BV(WGM21); // CTC mode
        TCCR2 |= 0x3; // clock prescale by 64
        OCR2 = 25; // to make 10KHz clock
        TIMSK |= _BV(OCIE2); // interrupt on compare
    #endif
    // T_OCR2 = 25; // to make 10KHz clock
    // T_TIMSK |= _BV(T_OCIE2); // interrupt on compare
}

```

```

void enableRTC2(void) {
    #ifndef IS_644
        TIMSK2 |= _BV(OCIE2A); // interrupt on compare
    #else
        TIMSK |= _BV(OCIE2); // interrupt on compare
    #endif
    // T_TIMSK |= _BV(T_OCIE2); // interrupt on compare
}

```

```

void disableRTC2(void) {
    #ifndef IS_644
        TIMSK2 &= ~_BV(OCIE2A); // interrupt on compare
    #else
        TIMSK &= ~_BV(OCIE2); // interrupt on compare
    #endif
    // T_TIMSK &= ~_BV(T_OCIE2); // interrupt on compare
}

```

```

void initRTC(void) {
    // setup RTC (interrupt 0) to run @ 1ms per interrupt
    #ifndef IS_644
        TCCR0A = _BV(WGM01); // CTC mode
        TCCR0B |= 0x3; // clock prescale by 64
        OCR0A = 250; // to make 1KHz clock
        TIMSK0 |= _BV(OCIE0A); // interrupt on compare
    #endif
}

```

```

#else
    TCCR0 = _BV(WGM01); // CTC mode
    TCCR0 |= 0x3; // clock prescale by 64
    OCR0 = 250; // to make 1KHz clock
    TIMSK |= _BV(OCIE0); // interrupt on compare
#endif
// T_OCRO = 250; // to make 1KHz clock
// T_TIMSK |= _BV(T_OCIE0); // interrupt on compare
}

void enableRTC(void) {
#ifdef IS_644
    TIMSK0 |= _BV(OCIE0A); // interrupt on compare
#else
    TIMSK |= _BV(OCIE0); // interrupt on compare
#endif
// T_TIMSK |= _BV(T_OCIE0); // interrupt on compare
}

void disableRTC(void) {
#ifdef IS_644
    TIMSK0 &= ~_BV(OCIE0A); // interrupt on compare
#else
    TIMSK &= ~_BV(OCIE0); // interrupt on compare
#endif
// T_TIMSK &= ~_BV(T_OCIE0); // interrupt on compare
}

void initHardware(void) {
    initializeUART(); // for debugging - gets turned on/off in main.h
    disableUART(); //
    enableUART();

    // setup PORT directions & pullups & stuff

    initializeServo();
    initializeADC();
    // etc...//

    LED_DDR |= _BV(RED_LED) | _BV(GREEN_LED);
    BUTTON_DDR &= ~_BV(BUTTON); // make button input
}

```

```

    BUTTON_PORT |= _BV(BUTTON); // turn on button pullup

    //alert user to startup
    RED_LED_ON;
    delay_ms(100);
    RED_LED_OFF;
    GREEN_LED_ON;
    delay_ms(100);
    RED_LED_ON;
    delay_ms(100);
    RED_LED_OFF;
    delay_ms(100);
    GREEN_LED_OFF;
}

#define BOOTLOADER_CHANNEL CHANNEL1 //the dongle has the 2 comm lines connected to
each other
#ifdef IS_644
    #define BOOT_START    0x7C00 //bootloader start address FOR MEGA32
#else
    #define BOOT_START    0x3E00 //bootloader start address FOR MEGA32
#endif

void testBootloader(void) {
    if ((pollClockLine(BOOTLOADER_CHANNEL)) && (pollDataLine(BOOTLOADER_
CHANNEL))) { //check for neighbor not handshaking
        setDataLine(BOOTLOADER_CHANNEL, 0);
        delay_ms(1);
        if (!pollClockLine(BOOTLOADER_CHANNEL)) { //could erroneously detect
start of handshaking here
            releaseDataLine(BOOTLOADER_CHANNEL);
            if (pollClockLine(BOOTLOADER_CHANNEL)) { //should happen
faster than handshaking
                blinkRed(1); //for user feedback
                goto_boot(); //start the bootloader
            }
        }
    }
}

//jump to user's application

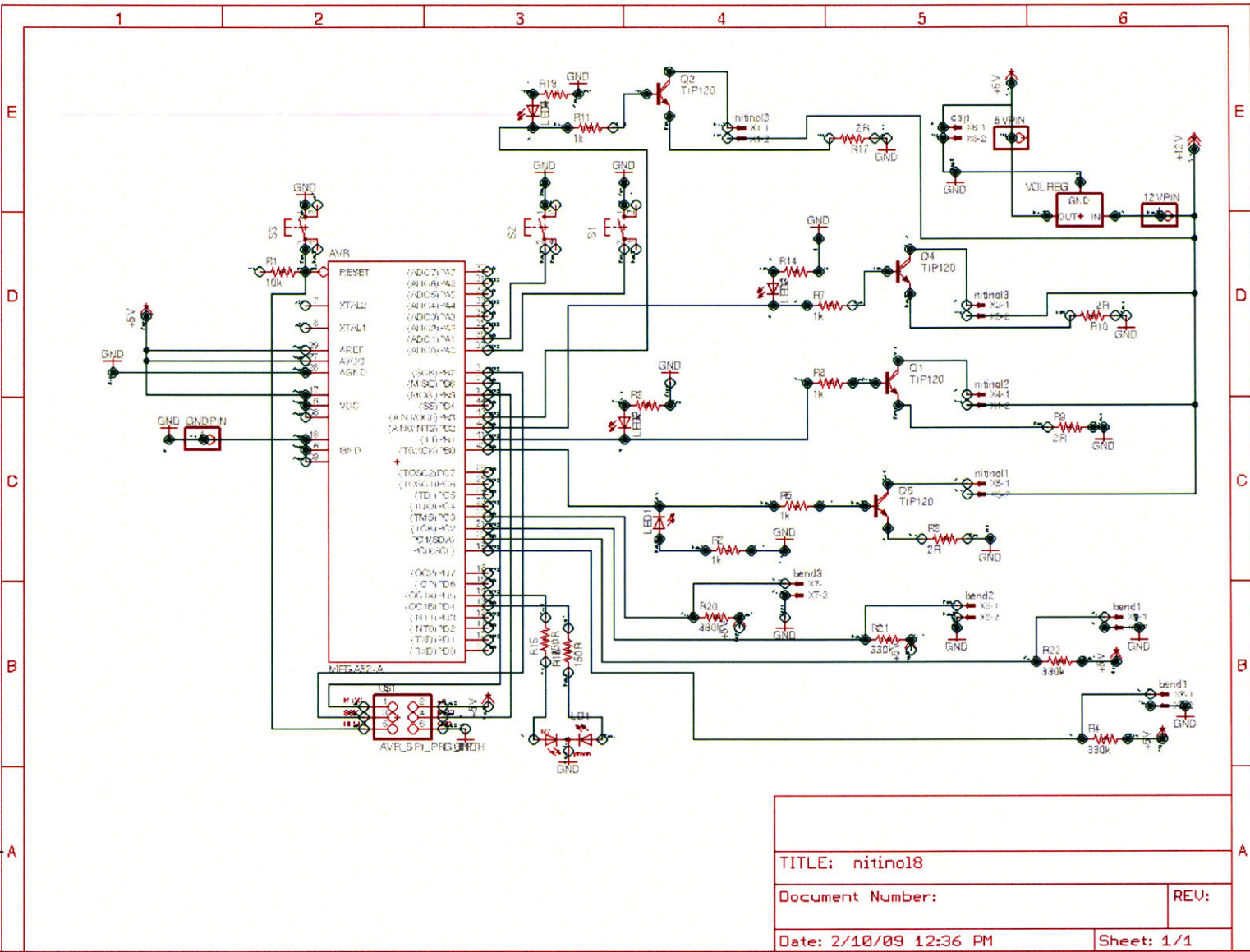
```

```
void goto_boot(void)  
{  
    ((void(*) (void))BOOT_START)(); //jump  
}
```



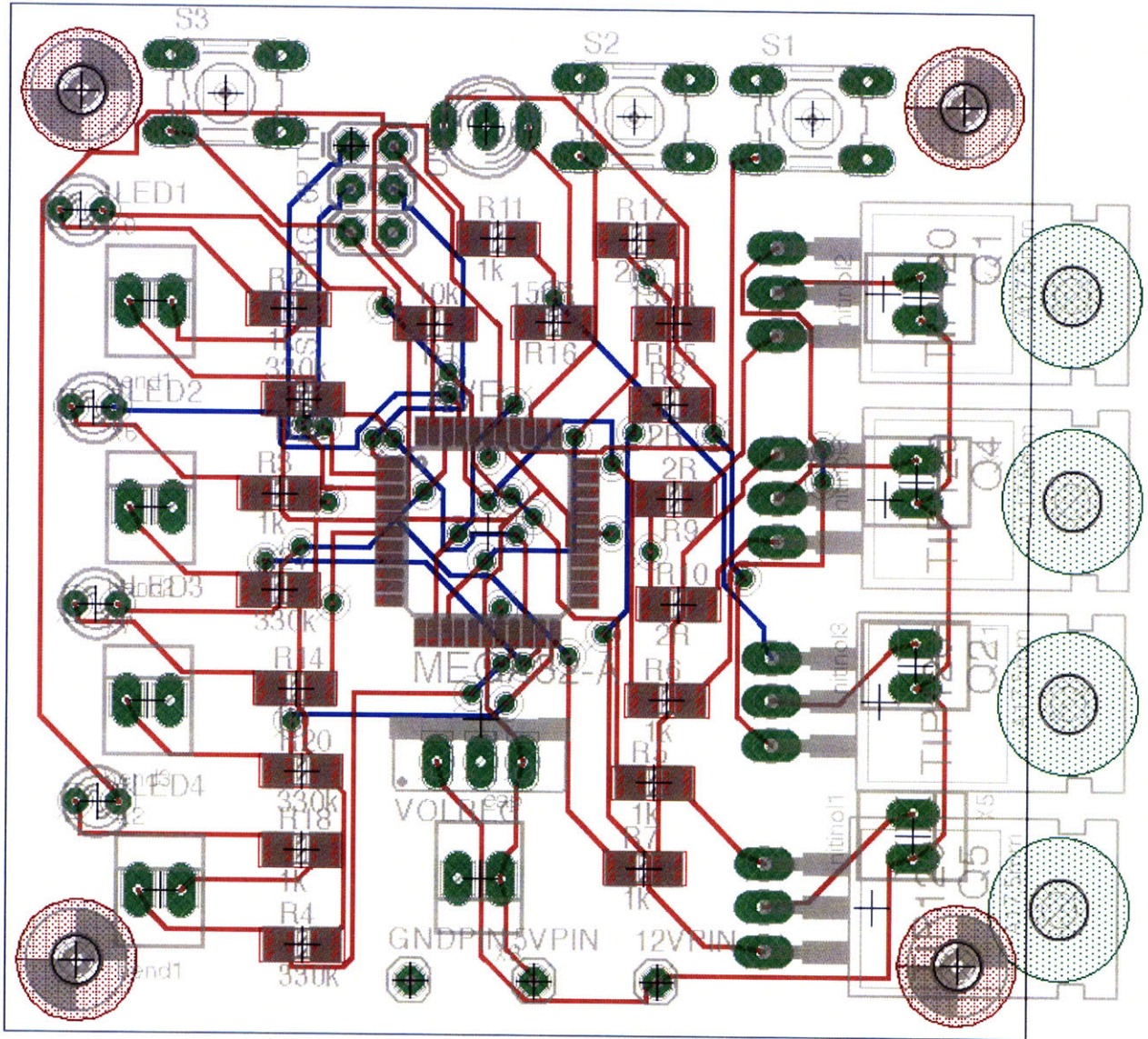


Bosuu PCB Schematic



TITLE: nitino18	
Document Number:	REV:
Date: 2/10/09 12:36 PM	Sheet: 1/1

# Bosu PCB Layout



## Bosu Firmware

```
//  
  
// AMANDA PARKES + ADAM KUMPF, OCTOBER 2007  
// NITINOL ACTUATOR DEMO SETUP FOR "ACTUATED MESH FABRIC" USING ATMEGA32  
// --> 12 BUTTON INPUTS & 12 NITINOL CONTROL OUTPUTS (OUT TO FET DRIVER).  
// THE PURPOSE OF THE MICROCONTROLLER IS TO LIMIT THE ON TIME TO A SET VALUE  
// AND THEN REQUIRE AN OFF TIME BEFORE THE NITINOL CAN BE TURNED ON AGAIN.  
// THIS WILL HOPEFULLY KEEP THINGS FROM BURNING, EVEN IF USERS ARE MALICIOUS!  
//  
  
//AMANDA PARKES + ANDREW GOESSLING, DECEMBER 2008  
//AMMENDED AND EXPANDED TO INCLUDE RECORD AND PLAYBACK  
  
#DEFINE F_CPU 8000000UL // CPU FREQUENCY IN HZ  
  
#INCLUDE <AVR/IO.H>  
#INCLUDE <UTIL/DELAY.H>  
#INCLUDE <AVR/INTERRUPT.H>  
#INCLUDE <AVR/EEPROM.H>  
  
#DEFINE IDLE          0x23  
#DEFINE PLAY         0x24  
#DEFINE RECORD       0x25  
#DEFINE TRUE         0x26  
#DEFINE FALSE        0x27  
#DEFINE BENDPINS     PINC  
#DEFINE EEPROMMAX    120  
#DEFINE RED_LED_ON   PORTD=PORTD|(1<<5)  
#DEFINE RED_LED_OFF  PORTD=PORTD&~(1<<5)  
#DEFINE GREEN_LED_ON PORTD=PORTD|(1<<4)  
#DEFINE GREEN_LED_OFF PORTD=PORTD&~(1<<4)  
// ----- FUNCTION PROTOTYPES -----  
VOID INIT_UART(VOID);  
VOID DELAY_MS(UINT16_T MILLIS);  
VOID INIT_TIMER0(VOID);  
VOID FET_ON1(VOID);  
VOID FET_OFF1(VOID);  
VOID FET_ON2(VOID);  
VOID FET_OFF2(VOID);  
VOID FET_ON3(VOID);  
VOID FET_OFF3(VOID);  
VOID FET_ON4(VOID);  
VOID FET_OFF4(VOID);
```

```

VOID FET_On5(VOID);
VOID FET_Off5(VOID);
VOID FET_On6(VOID);
VOID FET_Off6(VOID);
VOID FET_On7(VOID);
VOID FET_Off7(VOID);
VOID FET_On8(VOID);
VOID FET_Off8(VOID);
VOID FET_On9(VOID);
VOID FET_Off9(VOID);
VOID FET_On10(VOID);
VOID FET_Off10(VOID);
VOID FET_On11(VOID);
VOID FET_Off11(VOID);
VOID FET_On12(VOID);
VOID FET_Off12(VOID);
INT BUTTON1_IsDown(VOID);
INT BUTTON2_IsDown(VOID);
INT BUTTON3_IsDown(VOID);
INT BUTTON4_IsDown(VOID);
INT BUTTON5_IsDown(VOID);
INT BUTTON6_IsDown(VOID);
INT BUTTON7_IsDown(VOID);
INT BUTTON8_IsDown(VOID);
INT BUTTON9_IsDown(VOID);
INT BUTTON10_IsDown(VOID);
INT BUTTON11_IsDown(VOID);
INT BUTTON12_IsDown(VOID);
// -----

INT COUNT_THRESH = 200; // PERIOD OF THE ON/OFF,
INT COUNT_STEP_ON = 13; // COUNT UP TO THRESHOLD BY THIS VALUE,
INT COUNT_STEP_OFF = 10; // COUNT DOWN FROM THRESHOLD BY THIS VALUE, BIGGER TO TURN OFF
SOONER

INT FETCOUNTER1 = 0; // COUNTER FOR FET ON/OFF TIME
INT FETCANTURNON1 = 1; // BOOLEAN TO DETERMINE IF IT'S OKAY TO TURN ON THE FET
INT FETCOUNTER2 = 0; // COUNTER FOR FET ON/OFF TIME
INT FETCANTURNON2 = 1; // BOOLEAN TO DETERMINE IF IT'S OKAY TO TURN ON THE FET
INT FETCOUNTER3 = 0; // COUNTER FOR FET ON/OFF TIME
INT FETCANTURNON3 = 1; // BOOLEAN TO DETERMINE IF IT'S OKAY TO TURN ON THE FET
INT FETCOUNTER4 = 0; // COUNTER FOR FET ON/OFF TIME
INT FETCANTURNON4 = 1; // BOOLEAN TO DETERMINE IF IT'S OKAY TO TURN ON THE FET
INT FETCOUNTER5 = 0; // COUNTER FOR FET ON/OFF TIME
INT FETCANTURNON5 = 1; // BOOLEAN TO DETERMINE IF IT'S OKAY TO TURN ON THE FET

```

```

INT FETCOUNTER6 = 0; // COUNTER FOR FET ON/OFF TIME
INT FETCANTURNON6 = 1; // BOOLEAN TO DETERMINE IF IT'S OKAY TO TURN ON THE FET
INT FETCOUNTER7 = 0; // COUNTER FOR FET ON/OFF TIME
INT FETCANTURNON7 = 1; // BOOLEAN TO DETERMINE IF IT'S OKAY TO TURN ON THE FET
INT FETCOUNTER8 = 0; // COUNTER FOR FET ON/OFF TIME
INT FETCANTURNON8 = 1; // BOOLEAN TO DETERMINE IF IT'S OKAY TO TURN ON THE FET
INT FETCOUNTER9 = 0; // COUNTER FOR FET ON/OFF TIME
INT FETCANTURNON9 = 1; // BOOLEAN TO DETERMINE IF IT'S OKAY TO TURN ON THE FET
INT FETCOUNTER10 = 0; // COUNTER FOR FET ON/OFF TIME
INT FETCANTURNON10 = 1; // BOOLEAN TO DETERMINE IF IT'S OKAY TO TURN ON THE FET
INT FETCOUNTER11 = 0; // COUNTER FOR FET ON/OFF TIME
INT FETCANTURNON11 = 1; // BOOLEAN TO DETERMINE IF IT'S OKAY TO TURN ON THE FET
INT FETCOUNTER12 = 0; // COUNTER FOR FET ON/OFF TIME
INT FETCANTURNON12 = 1; // BOOLEAN TO DETERMINE IF IT'S OKAY TO TURN ON THE FET

UNSIGNED CHAR PLAYBACKTIMER = 0, RECORDTIMER = 0; //USED TO COUNT FAST TIMER TO .5 S
VOLATILE UNSIGNED CHAR MODE = IDLE; //CURRENT MODE IE RECORD PLAY IDLE
UNSIGNED CHAR BUTCOUNTER1 = 0, BUTCOUNTER2 = 0; //DEBOUNCES BUTTON
VOLATILE UNSIGNED CHAR BUT1 = FALSE, BUT2 = FALSE; //BUTTON STATUS
VOLATILE UNSIGNED CHAR CURRENTRECORDPOS = 0; //CURRENT EEPROM POS
VOLATILE UNSIGNED CHAR CURRENTPLAYPOS = 0; //
VOLATILE UNSIGNED CHAR CURRENTPLAYBYTE = 0; //CURRENT BYTE TO PLAY

```

```

// -----
// ----- MAIN -----
// -----
INT MAIN(VOID) {
    DDRB = 0xFF; // SET PORTB ALL OUTPUT.
    DDRD = 0xFF; // SET PORTD ALL OUTPUT.
    DDRC = 0x00; // SET PORTC ALL INPUT.
    DDRA = 0x00; // SET PORTA ALL INPUT.
    PORTA = 0x03; // INTERNAL PULL UP PA0 AND PA1

    INIT_TIMER0(); // INITIALIZE TIMER 0
    SEI(); // ENABLES INTERRUPTS

    WHILE(1) {

        IF(BUT1 == TRUE){
            BUT1 = FALSE;

            SWITCH(MODE){
                CASE IDLE:

```



```

        RED_LED_ON;
        CURRENTRECORDPos = 0;           //RESET AND
RECORD FROM ZERO

        MODE = RECORD;
        BREAK;

    CASE RECORD:
        RED_LED_OFF;
        GREEN_LED_ON;
        CURRENTPLAYPos = 0;           //RESET
AND PLAY FROM ZERO

        MODE = PLAY;
        BREAK;

    CASE PLAY:
        GREEN_LED_OFF;
        CURRENTPLAYByte = 0;         //TURN OFF OUTPUT
TO FET

        MODE = IDLE;
        BREAK;

    DEFAULT:
        BREAK;
    }
}

    IF((BUT2 == TRUE)&&(MODE==IDLE)){ //SECONDARY BUT TO PLAY
FROM LAST RECORD W/O NEW RECORD
        BUT2 = FALSE;
        GREEN_LED_ON;
        CURRENTPLAYPos = 0;
        MODE = PLAY;
    }

    ELSE
        BUT2 = FALSE;           //MAKE
SURE BUTTON DOESN'T STICK AROUND IN DIFFERENT MODES

    }
    RETURN 0;
}

//-----
//-----

```

```

VOID INIT_TIMER0(VOID){
    //TIMER0 SETUP AS AN INTERRUPT TO HANDLE BIT-TWIDDLING PWM (2 TRI-COLOR LEDs)
    //TCCR0
    //WGM01:0 --> 10 = CLEAR ON COMPARE MATCH TO OCR0
    //COM01:0 --> 00 = DON'T CHANGE OUTPUT PINS ON COMPARE MATCH
    //CS02:0 --> 011 = CLOCK SOURCE = Osc/64 (100 = Osc/256) (101 = Osc/1024)
    TCCR0 = (1<<CS02) | (1<<WGM01);
    //OCR0 = TIMER0 OUTPUT COMPARE VALUE
    OCR0 = 0x80;
    //TIMSK = TIMER/COUNTER INTERRUPT REGISTER
    //OCIE0:TIMER/COUNTER0 OUTPUT COMPARE MATCH INTERRUPT ENABLE
    TIMSK |= (1<<OCIE0);
}

//TIMER0 COMPARE MATCH VECTOR
//THIS INTERRUPT GETS TRIGGERED EVERY TIME TIMER0 REACHES OCR0.
//TIMER0 IS SETUP TO RESET ITSELF WHEN IT REACHES OCR0 ALSO.
SIGNAL (TIMER0_COMP_VECT){

    IF((RECORDTIMER >= 122)&&(MODE == RECORD)){
//EVERY .5 s
        RECORDTIMER = 0;

        IF(CURRENTRECORDPos < EEPROMMAX){
//IF NOT AT END
            EEPROM_WRITE_BYTE((UNSIGNED CHAR*)CURRENTRECORDPos, BENDPINS);
//SAVE BENDPINS BYTES ONE AFTER ANOTHER
            CURRENTRECORDPos++;
        }

        ELSE{
            CURRENTPLAYPos = 0;
//AT END THEN PLAY
            MODE = PLAY;
        }

    }
    RECORDTIMER++;

    IF((PLAYBACKTIMER >= 122)&&(MODE == PLAY)){
//EVERY .5s
        PLAYBACKTIMER = 0;

        IF((CURRENTPLAYPos < CURRENTRECORDPos)&&(CURRENTPLAYPos < EEPROMMAX))

```

```

{ //IF NOT AT END OF RECORD OR EEPROM
    CURRENTPLAYBYTE = EEPROM_READ_BYTE((UNSIGNED CHAR*)CURRENTPLAYPos);
    //GET THE BYTE
    CURRENTPLAYPos++;
}

ELSE
    CURRENTPLAYPos = 0;
//RESET AND PLAY AGAIN
}
PLAYBACKTIMER++;

//BUTTON1 DEBOUNCE
IF(!(PINA&(1<<0))){
    IF(BUTCOUNTER1 <= 22)
        BUTCOUNTER1++;
}

IF((BUTCOUNTER1 >= 22)&&(PINA&(1<<0))){
    BUT1 = TRUE;
    BUTCOUNTER1 = 0;
}

//BUTTON2
IF(!(PINA&(1<<1))){
    IF(BUTCOUNTER2 <= 22)
        BUTCOUNTER2++;
}

IF((BUTCOUNTER2 >= 22)&&(PINA&(1<<1))){
    BUT2 = TRUE;
    BUTCOUNTER2 = 0;
}

/* TEMPLATE FOR EACH BUTTON/OUTPUT PAIR! :)
//
// REPLACE NUM WITH CHANNEL NUMBER (I.E., 1,2,3,...)
//

// [NUM]
IF((CURRENTPLAYBYTE & (1<<(NUM-1))) && FETCANTURNONNUM == 1){
    FET_OnNUM();
    IF(FETCOUNTERNUM >= COUNT_THRESH){
        FETCOUNTERNUM = COUNT_THRESH;
    }
}

```

```

        FETCANTURNONNUM = 0;
    }
    FETCOUNTERNUM += COUNT_STEP_ON;
}ELSE{
    FET_OffNUM();
    IF(FETCOUNTERNUM > 0){
        FETCOUNTERNUM -= COUNT_STEP_OFF;
    }ELSE{
        FETCOUNTERNUM = 0;
        FETCANTURNONNUM = 1;
    }
}
*/

// [1]
IF((CURRENTPLAYBYTE & (1<<0)) && FETCANTURNON1 == 1){
//IF CORRESPONDING BIT IN CURRENTPLAYBYTE IS SET
    FET_On1();
    //TURN ON CORRESPONDING FET
    IF(FETCOUNTER1 >= COUNT_THRESH){
        FETCOUNTER1 = COUNT_THRESH;
        FETCANTURNON1 = 0;
    }
    FETCOUNTER1 += COUNT_STEP_ON;
}ELSE{
    FET_Off1();
    IF(FETCOUNTER1 > 0){
        FETCOUNTER1 -= COUNT_STEP_OFF;
    }ELSE{
        FETCOUNTER1 = 0;
        FETCANTURNON1 = 1;
    }
}

// [2]
IF((CURRENTPLAYBYTE & (1<<1)) && FETCANTURNON2 == 1){
    FET_On2();
    IF(FETCOUNTER2 >= COUNT_THRESH){
        FETCOUNTER2 = COUNT_THRESH;
        FETCANTURNON2 = 0;
    }
    FETCOUNTER2 += COUNT_STEP_ON;
}ELSE{
    FET_Off2();
    IF(FETCOUNTER2 > 0){

```

```

        FETCOUNTER2 -= COUNT_STEP_OFF;
    }ELSE{
        FETCOUNTER2 = 0;
        FETCANTURNON2 = 1;
    }
}

// [3]
IF((CURRENTPLAYBYTE & (1<<2)) && FETCANTURNON3 == 1){
    FET_ON3();
    IF(FETCOUNTER3 >= COUNT_THRESH){
        FETCOUNTER3 = COUNT_THRESH;
        FETCANTURNON3 = 0;
    }
    FETCOUNTER3 += COUNT_STEP_ON;
}ELSE{
    FET_OFF3();
    IF(FETCOUNTER3 > 0){
        FETCOUNTER3 -= COUNT_STEP_OFF;
    }ELSE{
        FETCOUNTER3 = 0;
        FETCANTURNON3 = 1;
    }
}

// [4]
IF((CURRENTPLAYBYTE & (1<<3)) && FETCANTURNON4 == 1){
    FET_ON4();
    IF(FETCOUNTER4 >= COUNT_THRESH){
        FETCOUNTER4 = COUNT_THRESH;
        FETCANTURNON4 = 0;
    }
    FETCOUNTER4 += COUNT_STEP_ON;
}ELSE{
    FET_OFF4();
    IF(FETCOUNTER4 > 0){
        FETCOUNTER4 -= COUNT_STEP_OFF;
    }ELSE{
        FETCOUNTER4 = 0;
        FETCANTURNON4 = 1;
    }
}

// [5]
IF((CURRENTPLAYBYTE & (1<<4)) && FETCANTURNON5 == 1){

```

```

    FET_On5();
    IF(FETCOUNTER5 >= COUNT_THRESH){
        FETCOUNTER5 = COUNT_THRESH;
        FETCANTURNON5 = 0;
    }
    FETCOUNTER5 += COUNT_STEP_ON;
}ELSE{
    FET_Off5();
    IF(FETCOUNTER5 > 0){
        FETCOUNTER5 -= COUNT_STEP_OFF;
    }ELSE{
        FETCOUNTER5 = 0;
        FETCANTURNON5 = 1;
    }
}

// [6]
IF((CURRENTPLAYBYTE & (1<<5)) && FETCANTURNON6 == 1){
    FET_On6();
    IF(FETCOUNTER6 >= COUNT_THRESH){
        FETCOUNTER6 = COUNT_THRESH;
        FETCANTURNON6 = 0;
    }
    FETCOUNTER6 += COUNT_STEP_ON;
}ELSE{
    FET_Off6();
    IF(FETCOUNTER6 > 0){
        FETCOUNTER6 -= COUNT_STEP_OFF;
    }ELSE{
        FETCOUNTER6 = 0;
        FETCANTURNON6 = 1;
    }
}

// [7]
IF((CURRENTPLAYBYTE & (1<<6)) && FETCANTURNON7 == 1){
    FET_On7();
    IF(FETCOUNTER7 >= COUNT_THRESH){
        FETCOUNTER7 = COUNT_THRESH;
        FETCANTURNON7 = 0;
    }
    FETCOUNTER7 += COUNT_STEP_ON;
}ELSE{
    FET_Off7();
    IF(FETCOUNTER7 > 0){

```



```

        FETCOUNTER7 -= COUNT_STEP_OFF;
    }ELSE{
        FETCOUNTER7 = 0;
        FETCANTURNON7 = 1;
    }
}

// [8]
IF((CURRENTPLAYBYTE & (1<<7)) && FETCANTURNON8 == 1){
    FET_ON8();
    IF(FETCOUNTER8 >= COUNT_THRESH){
        FETCOUNTER8 = COUNT_THRESH;
        FETCANTURNON8 = 0;
    }
    FETCOUNTER8 += COUNT_STEP_ON;
}ELSE{
    FET_OFF8();
    IF(FETCOUNTER8 > 0){
        FETCOUNTER8 -= COUNT_STEP_OFF;
    }ELSE{
        FETCOUNTER8 = 0;
        FETCANTURNON8 = 1;
    }
}

/*// [9]
IF(BUTTON9_IsDOWN() == 1 && FETCANTURNON9 == 1){
    FET_ON9();
    IF(FETCOUNTER9 >= COUNT_THRESH){
        FETCOUNTER9 = COUNT_THRESH;
        FETCANTURNON9 = 0;
    }
    FETCOUNTER9 += COUNT_STEP_ON;
}ELSE{
    FET_OFF9();
    IF(FETCOUNTER9 > 0){
        FETCOUNTER9 -= COUNT_STEP_OFF;
    }ELSE{
        FETCOUNTER9 = 0;
        FETCANTURNON9 = 1;
    }
}

// [10]
IF(BUTTON10_IsDOWN() == 1 && FETCANTURNON10 == 1){

```

```

    FET_On10());
    IF(FETCOUNTER10 >= COUNT_THRESH){
        FETCOUNTER10 = COUNT_THRESH;
        FETCANTURNON10 = 0;
    }
    FETCOUNTER10 += COUNT_STEP_ON;
}ELSE{
    FET_Off10());
    IF(FETCOUNTER10 > 0){
        FETCOUNTER10 -= COUNT_STEP_OFF;
    }ELSE{
        FETCOUNTER10 = 0;
        FETCANTURNON10 = 1;
    }
}

// [11]
IF(BUTTON11_IsDown() == 1 && FETCANTURNON11 == 1){
    FET_On11());
    IF(FETCOUNTER11 >= COUNT_THRESH){
        FETCOUNTER11 = COUNT_THRESH;
        FETCANTURNON11 = 0;
    }
    FETCOUNTER11 += COUNT_STEP_ON;
}ELSE{
    FET_Off11());
    IF(FETCOUNTER11 > 0){
        FETCOUNTER11 -= COUNT_STEP_OFF;
    }ELSE{
        FETCOUNTER11 = 0;
        FETCANTURNON11 = 1;
    }
}

// [12]
IF(BUTTON12_IsDown() == 1 && FETCANTURNON12 == 1){
    FET_On12());
    IF(FETCOUNTER12 >= COUNT_THRESH){
        FETCOUNTER12 = COUNT_THRESH;
        FETCANTURNON12 = 0;
    }
    FETCOUNTER12 += COUNT_STEP_ON;
}ELSE{
    FET_Off12());
    IF(FETCOUNTER12 > 0){

```

```

        FETCOUNTER12 -= COUNT_STEP_OFF;
    }ELSE{
        FETCOUNTER12 = 0;
        FETCANTURNON12 = 1;
    }
}*/
//PORTB |= (1<<PB2)|(1<<PB3)|(1<<PB4)|(1<<PB5);

}

//-----
//-----

/*// --- USER INTERFACE (BUTTONS) ---

INT BUTTON1_IsDown(VOID){
    IF((PINC & 1<<PC0) != 0x00){
        RETURN 1;
    }
    RETURN 0;
}

INT BUTTON2_IsDown(VOID){
    IF((PINC & 1<<PC1) != 0x00){
        RETURN 1;
    }
    RETURN 0;
}

INT BUTTON3_IsDown(VOID){
    IF((PINC & 1<<PC2) != 0x00){
        RETURN 1;
    }
    RETURN 0;
}

INT BUTTON4_IsDown(VOID){
    IF((PINC & 1<<PC3) != 0x00){
        RETURN 1;
    }
    RETURN 0;
}

INT BUTTON5_IsDown(VOID){
    IF((PINC & 1<<PC4) != 0x00){

```

```

        RETURN 1;
    }
    RETURN 0;
}

INT BUTTON6_IsDown(void){
    IF((PINC & 1<<PC5) != 0x00){
        RETURN 1;
    }
    RETURN 0;
}

INT BUTTON7_IsDown(void){
    IF((PINC & 1<<PC6) != 0x00){
        RETURN 1;
    }
    RETURN 0;
}

INT BUTTON8_IsDown(void){
    IF((PINC & 1<<PC7) != 0x00){
        RETURN 1;
    }
    RETURN 0;
}

INT BUTTON9_IsDown(void){
    IF((PINA & 1<<PA0) != 0x00){
        RETURN 1;
    }
    RETURN 0;
}

INT BUTTON10_IsDown(void){
    IF((PINA & 1<<PA1) != 0x00){
        RETURN 1;
    }
    RETURN 0;
}

INT BUTTON11_IsDown(void){
    IF((PINA & 1<<PA2) != 0x00){
        RETURN 1;
    }
    RETURN 0;
}

```

```

}

INT BUTTON12_IsDown(VOID){
    IF((PINA & 1<<PA3) != 0x00){
        RETURN 1;
    }
    RETURN 0;
}*/

// -- OUTPUTS (to FETs!) -----
// [1]
VOID FET_On1(VOID){
    PORTB |= 1<<PB0;
}
VOID FET_Off1(VOID){
    PORTB &= ~(1<<PB0);
}

// [2]
VOID FET_On2(VOID){
    PORTB |= 1<<PB1;
}
VOID FET_Off2(VOID){
    PORTB &= ~(1<<PB1);
}

// [3]
VOID FET_On3(VOID){
    PORTB |= 1<<PB2;
}
VOID FET_Off3(VOID){
    PORTB &= ~(1<<PB2);
}

// [4]
VOID FET_On4(VOID){
    PORTB |= 1<<PB3;
}
VOID FET_Off4(VOID){
    PORTB &= ~(1<<PB3);
}

// [5]

```

```

VOID FET_On5(VOID){
    PORTB |= 1<<PB4;
}
VOID FET_Off5(VOID){
    PORTB &= ~(1<<PB4);
}

// [6]
VOID FET_On6(VOID){
    PORTB |= 1<<PB5;
}
VOID FET_Off6(VOID){
    PORTB &= ~(1<<PB5);
}

// [7]
VOID FET_On7(VOID){
    PORTB |= 1<<PB6;
}
VOID FET_Off7(VOID){
    PORTB &= ~(1<<PB6);
}

// [8]
VOID FET_On8(VOID){
    PORTB |= 1<<PB7;
}
VOID FET_Off8(VOID){
    PORTB &= ~(1<<PB7);
}

// [9]
VOID FET_On9(VOID){
    PORTD |= 1<<PD0;
}
VOID FET_Off9(VOID){
    PORTD &= ~(1<<PD0);
}

// [10]
VOID FET_On10(VOID){
    PORTD |= 1<<PD1;
}
VOID FET_Off10(VOID){

```

```
        PORTD &= ~(1<<PD1);
    }

    // [11]
    VOID FET_On11(VOID){
        PORTD |= 1<<PD2;
    }
    VOID FET_Off11(VOID){
        PORTD &= ~(1<<PD2);
    }

    // [12]
    VOID FET_On12(VOID){
        PORTD |= 1<<PD3;
    }
    VOID FET_Off12(VOID){
        PORTD &= ~(1<<PD3);
    }
}
```