2013-01-01

# 3D Printing For The Rapid Prototyping Of Structural Electronics

Rodolfo Salas
*University of Texas at El Paso*, rsalas4@miners.utep.edu

# 3D PRINTING FOR THE RAPID PROTOTYPING OF STRUCTURAL ELECTRONICS

RODOLFO SALAS

Department of Electrical and Computer Engineering

APPROVED:

Eric MacDonald, Ph.D., Chair

Ryan B. Wicker, Ph.D.

Berenice Verdin, Ph.D.

Benjamin C. Flores, Ph.D.
Dean of the Graduate School

# Dedication

This thesis is dedicated to my wife Carmen, without her unwavering support, completion of this work would have been impossible.

# 3D PRINTING FOR THE RAPID PROTOTYPING OF STRUCTURAL ELECTRONICS

by

RODOLFO SALAS, B.S.E.E.

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

Department of Electrical and Computer Engineering

THE UNIVERSITY OF TEXAS AT EL PASO

May 2013

## Acknowledgements

It is a pleasure to acknowledge the meaningful contributions from everyone who supported me to make this work possible. Heartfelt thanks to Dr. Ryan Wicker for his support and for allowing me the opportunity to work in the W. M. Keck Center where access to state of the art equipment, but more importantly to brilliant and supportive colleagues allowed this research to be accomplished. I am grateful to my advisor Dr. Eric MacDonald for his support and guidance from the start of this research project until completion. I would like to show my gratitude to my good friend and personal counselor Dan Muse, who encouraged me to pursue the worthy goal of a graduate education.

For their support and guidance I gratefully acknowledge Mahesh Tonde, Luis Ochoa, Armando Rivera, David Espalin, Frank Medina and the rest of his supportive team. To Efrain Aguilera, Fernando Cedillos, Richard Olivas and all current and previous colleagues I am indebted to you for the lessons I have learned from each of you about hard work, positive outlook and supportive attitude that are as valuable to me as any lesson I learned in a classroom.

# Abstract

In business, the time required to shepherd a new product through the phases of concept, design, prototypes, through manufacture and finally making the product available for sale (together known as: Time to Market) is critical to the success and profitability of the next generation of products. The use of relatively new technologies, known as 3D Printing, which allow the fabrication of complex tridimensional shapes within hours of a completed CAD design without the need for tooling or molds, has significantly reduced the time required for a designer to physically hold a part designed only hours before and to perform form and fit checks. However, as the complexity of these products increases, with the inclusion of sophisticated electronics encased in 3D structures with complex geometries and intricate detail Time to Market is compromised, resulting in lost opportunity. The use of advanced 3D Printing technology enhanced with component placement capabilities and Direct Write micro-dispensing to achieve electrical interconnect deposition has recently been pursued as a means to provide prototypes for functional electronic devices in similar time frame as traditional 2D bread board prototypes; however these 3D prototypes include the advantage of being embedded in more appropriate shapes in order to accurately represent finished products earlier in the development cycle.

The fabrication freedom offered by 3D Printing techniques such as Stereolithography (SL) and Fused Deposition Modeling (FDM) have recently been explored in the context of 3D electronics integration – referred to as Structural Electronics. Enhanced 3D Printing may eventually be used to produce end user parts, allowing for unit-level customization and local manufacturing; but until limitations on the types of materials that can be used and accuracies improved (an eventuality), these enhanced 3D printing technologies can be employed to reduce development times by providing advanced, geometrically accurate, functional structural electronic prototypes.

This application was demonstrated at the W.M. Keck Center by utilizing this technology to hasten the development process used to design a novelty six side gaming die. The die includes a micro-controller and accelerometer, which together detect motion and upon resting, identify its orientation through gravity to illuminate LEDs in the top surface for a fanciful effect. By applying this enhanced 3D printing, the prototype stage was expedited. For comparison, the device was ultimately fabricated

through the use of traditional manufacturing techniques, utilizing a flexible printed circuit board and a plastic injection molded case.

# Table of Contents

Acknowledgements ..... v

Abstract ..... vi

Table of Contents ..... viii

List of Tables ..... x

List of Figures ..... xi

Chapter 1: Introduction ..... 1

1.1 Use of Additive Manufacturing to produce prototypes ..... 1

1.2 Structural Electronics Process ..... 2

1.2.1 Stereolithography ..... 2

1.2.2 Fused Deposition Modeling ..... 3

1.2.3 Direct Print Micro-Dispensing ..... 4

1.3 Research Objectives ..... 4

1.4 The W. M. Keck Center ..... 5

1.5 Thesis Outline ..... 6

Chapter 2: Previous Work ..... 8

2.1 Introduction ..... 8

2.2 Literature Review ..... 8

Chapter 3: System Requirements ..... 29

3.1 Electronic Design ..... 29

3.1.1 Basic dice block diagram ..... 29

3.1.2 Lithium polymer cell, charge control circuit ..... 30

3.1.3 Lithium polymer cell safety circuit ..... 32

3.1.4 Full protection circuit ..... 33

3.1.5 Wireless charger system, battery module ..... 33

3.1.6 Wireless charger system, battery charger ..... 35

3.1.7 Full dice circuit versions 1-3 ..... 37

3.1.8 Manual Routing ..... 38

3.1.9 Automated Routing and Layout using Proteus ..... 40

3.2 Mechanical Design ..... 42

3.2.1 SolidWorks Challenges ..... 42
3.2.2 Design for Plastic Injection Molding ..... 44
3.3 Micro-Controller program ..... 46
3.3.1 Basic Algorithm ..... 46
3.3.2 Analog to Digital Conversion method ..... 47
3.3.3 Using Microchip Micro-Controller with MPLab IDE ..... 48
3.3.4 I2C Serial Communication method ..... 50
3.3.5 Using Atmel Micro-Controller with Studio 6 IDE ..... 51
3.4 Traditional Fabrication ..... 53
3.4.1 Flexible Circuit Fabrication ..... 53
3.4.2 Plastic Injection Molding Case Fabrication ..... 56
3.5 Cost Comparison of Fabrication Techniques ..... 57
Chapter 4: Applications ..... 63
4.1 Original design, powered by replaceable batteries ..... 63
4.2 Second iteration, recharged through wireless magnetic field ..... 64
4.3 Third version, recharged through a micro usb-b connector ..... 67
4.4 Prototype made through traditional manufacturing techniques ..... 68
4.5 Summary of smart dice prototype versions ..... 69
Chapter 5: Conclusions ..... 71
5.1 Comparison of prototype fabrication techniques ..... 71
5.2 Final conclusions ..... 72
References ..... 74
Appendix A: Micro-controller code for version 1 dice ..... 77
Appendix B: Micro-controller code for version 2 dice charger ..... 83
Appendix C: Micro-controller code for version 2 and 3 dice ..... 84
Appendix D: Micro-controller code for version 4 dice ..... 91
Appendix E: TWI function header for ATTiny to assume master function ..... 93
Appendix F: TWI function prototype for ATTiny to assume master function ..... 95
Curriculum Vita ..... 101

# List of Tables


Table 3.1 Bill of materials comparison between version 3 and 4 designs ..... 57
Table 3.2 Summary of costs for each fabrication methodology ..... 62
Table 4.1 Summary of prototype versions ..... 69
Table 5.1 Prototype fabrication methodologies ..... 71

# List of Figures


Figure 2.1 Steps for assembling two layers of PCBs [11] ..... 10
Figure 2.2 CNC milling machine with adapted polyurethane foam dispense system [12] ..... 11
Figure 2.3 Integration of SLA 250 with D.W. pump [16] ..... 13
Figure 2.4 Three stages of the build process. a) The SL build process is stopped b) The components are embedded c) The circuit is completed through Direct Write [17, 29] ..... 14
Figure 2.5 Toy figure being fabricated and tested [18] ..... 14
Figure 2.6 Printed battery and IPMC actuator are compatible, transistor is not [20] ..... 15
Figure 2.7 a) Circuit layout. b) Location of circuit within SL volume. c) Final appearance [21] ..... 16
Figure 2.8 a) SLA built part. b) With components. c) With silver ink interconnects [22] ..... 17
Figure 2.9 Three generations of magnetometer design. [23] ..... 18
Figure 2.10 Fourth generation magnetometer [23] ..... 19
Figure 2.11 Magnetic field is detected from the right side and has 2 of 3 degree of intensity [23] ..... 19
Figure 2.12 Standard arrangement for LIFT system. [24] ..... 21
Figure 2.13 Completed 555 flasher circuit achieved through the use of LIFT technique [24] ..... 22
Figure 2.14 Schematic of ultrasonic equipment [26] ..... 23
Figure 2.15 a) Machined cavity. b) Applied insulator. c) D.W. components. D) Completed. [26] ..... 24
Figure 2.16 Schematic of the AJP process [27] ..... 25
Figure 2.17 Aersol Jet deposition of nano-particle silver ink on curved surface. [27] ..... 26
Figure 2.18 FDM created wing with Aerosol Jet printed circuit and antenna [28] ..... 27
Figure 3.1 Block diagram of dice circuit ..... 30
Figure 3.2 Charging profile for lithium cells ..... 31
Figure 3.3 Simplified application diagram ..... 31
Figure 3.4 Connection example of safety circuit chip ..... 32
Figure 3.5 Safety and charging circuits implemented ..... 33
Figure 3.6 Wireless powered battery module circuit ..... 34
Figure 3.7 Two wireless powered battery modules partially and fully completed. ..... 35
Figure 3.8 Circuit for the resonant field generator ..... 36
Figure 3.9 Completed charger station with dice. ..... 37
Figure 3.10 Full dice circuit ..... 38
Figure 3.11 Routing of dice interconnects designed manually ..... 39
Figure 3.12 Full assembly of first generation dice ..... 40
Figure 3.13 A flat area is used to simulate an unwrapped cylinder ..... 41
Figure 3.14 The flat routing is designed into a cylinder ..... 42
Figure 3.15 Sketch is made with dimensions of part to embed ..... 43
Figure 3.16 Channels are cut to provide a path for the silver ink to be deposited ..... 44
Figure 3.17 Design for viable plastic mold channels ..... 45
Figure 3.18 Draft analysis tool indicates, in yellow, faces that require draft angle ..... 46
Figure 3.19 The 3 axis accelerometer ADXL330 and the 6 possible orientations it can assume ..... 47
Figure 3.20 MPLab IDE for programming PIC micro-controllers ..... 49
Figure 3.21 Connection diagram for using MPLab ICD 2 ..... 49
Figure 3.22 Samples of 6 bit result and interpretation ..... 50
Figure 3.23 IAR Embedded workbench IDE used to develop C code ..... 52
Figure 3.24 AVR Studio 6 screen shot, as was used to program the micro-controller ..... 53
Figure 3.25 Circuit design for 4th generation dice ..... 54
Figure 3.26 Automatic routing made using ARES software ..... 55
Figure 3.27 Visual approximation of both sides of flexible circuit ..... 55
Figure 3.28 Mold for cap and for case to form a dice set. ..... 57

Figure 3.29 Sample quote for plastic parts....60
Figure 3.30 Price drop based on ordered quantity....61
Figure 4.1 The designed and the implemented dice, version 1....63
Figure 4.2 Design of 2nd generation dice and implementation steps....65
Figure 4.3 Charger design and implementation....66
Figure 4.4 Design and Implementation of version 3 dice....67
Figure 4.5 Plastic case design and implementation....68
Figure 4.6 The flexible circuit design and implementation....68

# Chapter 1: Introduction

## 1.1 USE OF ADDITIVE MANUFACTURING TO PRODUCE PROTOTYPES

A new product must invariable undergo several transformations before becoming available for sale to the general public. An idea for a product is first made into a prototype or sample in order to evaluate the fit and finish of the built part and by doing the required steps to create the physical part evaluate also the ease of manufacture. In decades past, this step was both time-consuming and expensive, creating a significant hurdle for new product introductions especially for newcomers who lacked the appropriate and usually expensive, equipment already on hand. [1] Into this stage additive manufacturing made its entrance in the late 1980's, later applied to the prototyping task and known as "rapid prototyping" because it allowed future manufacturers to circumvent the lengthy process of typical prototyping by having a scaled-down or full-scale replica of the designed product quickly available.[2] These new products were only prototypes however, due to limitations of the additive manufacturing technologies, where compromises still have to be made between high-accuracy but exotic materials (SL), growing list of available materials but lower accuracy (FDM) or a relatively low-cost but low accuracy and lesser capabilities (Home FDM). While additive manufacturing technology continues to advance and to reduce these shortcomings, the technology remains best suited for prototype manufacture, for now. [3] Furthermore, these prototypes can normally, only satisfy the need to test for fit and form of the device housing with no option for a fully functional part. In order to have a fully functional electronic device, they had to make use of traditional printed circuit boards, which themselves, required typical lead times for manufacture. [1] This work describes how the W. M. Keck Center, one of the premier laboratories in the United States for additive manufacturing capabilities, undertook a project to showcase a full design cycle for an electronic device, a novelty six-sided gaming die. The process, from an idea through the prototype processes will be described, noting the significant advantages the availability of state-of-the-art additive manufacturing equipment played in its development; while showcasing the combination of additive manufacturing and direct print to create 3D structural electronics which more faithfully represent a completed electronic device.

## 1.2 STRUCTURAL ELECTRONICS PROCESS

In order to achieve a fully functional prototype of an electronic device, two technologies must be incorporated into a single technique to achieve it. An additive manufacturing technique is required to build the tri-dimensional design of the device housing, a means to include the electronic components that give it functional capabilities and a dispensing system that will allow the deposition of material to realize the electrical interconnects between the components.

### 1.2.1 Stereolithography

Stereolithography is one of the forerunners of the additive manufacturing revolution. It was patented in 1986 by Charles W. Hull who later founded 3D Systems, the first company to commercialize this process. [4, 5] Stereolithography employs a vat of photo-curable monomer resin which forms a polymer resin when exposed to UV light; and an ultra-violet laser to fabricate three-dimensional parts by building them-up layer by layer from the bottom up. [1] The process starts with a 3D design made on CAD software, this design is then prepared to be built by re-formatting the three-dimensional part into a file saved in .STL file format (STL is an acronym of Standard Tessellation Language). In this file format, solid three-dimensional parts are described by a grid of triangles formed across their surface. This file is further formatted using other proprietary software that allows the object to be converted into a series of layers, each of which will be created successively and cross-linked to make a solid object. During this step other activities must take place, such as the generating, and if needed, modifying sacrificial supports. These allow the part to be anchored to the base and provide physical support to any geometry that requires overhangs; they are made from the same resin but are generally 50% of the width of the build material to allow them to be broken off and discarded when the part is completed. Finally, unique parameters for the Stereolithography equipment are set, such as layer height, quantity of parts, orientation and number of units to fabricate, among several other, more specialized parameters.

The files describing these slices are transferred into the Stereolithography equipment and the process may begin. The ultraviolet laser draws each slice of the three-dimensional part on a moving platform, just below the surface of the resin in the vat. As the slice is drawn by the laser, the resin is

solidified to create a semi-hard slice of the three-dimensional part, as successive layers are stacked, a three-dimensional part is built. The process continues as the platform is lowered by a pre-defined height, allowing the next layer to be drawn by the laser while cross-linking it to the previous layer, giving it physical strength. When the top layer is completed, the platform rises to allow uncured resin to drain from any cavities and from the outside surface. The part is post-processed by submerging it in a container of Di-propylene Glycol (Normal) Propyl Ether to clean off the uncured resin from its surface. Then it is processed with a final dose of ultraviolet light in a sealed chamber to completely cure all surfaces and finish the hardening process. [1, 5]

### 1.2.2 Fused Deposition Modeling

Fused Deposition Modeling is included here for completeness, because its popularity has been increasing among those who seek to integrate additive manufacturing with technologies to deposit materials to realize electrical interconnects and thus achieve functional structural electronics; however, all the parts discussed here were made using the Stereolithography process. The Fused Deposition Modeling process begins, much like Stereolithography, with a CAD created tri-dimensional model. The model is then formatted into .STL format and further processed by the Fused Deposition Modeling software. A similar set of steps must be followed; the model is sliced into thin horizontal layers, supports are generated to brace overhanging features. In the Fused Deposition Modeling method, these supports are made from a water-soluble or brittle material, designed to be easily removed once the part is completed. Finally the design is ready to be built. The process begins inside of a heated chamber, where a moving nozzle dispenses material that has been heated just above its melting point; this is dispensed in a thin bead forming the design layer, as each slice is completed, the platform is lowered to allow the dispensing nozzle to form the next layer. The heated chamber helps maintain the dispensed layer at a high temperature to allow the currently dispensed layer to form a better bond between them. Whereas Stereolithography is preferred where high accuracy is required, Stereolithography resolution is due to laser beam width. FDM resolution is dictated by filament thickness and nozzle diameter. Fused Deposition Modeling has a wider range of materials available for manufacture, such as polycaprolactone, ABS, polycarbonate, and others, with properties to choose from, like different colors

or even a translucent material, various levels of strength, materials that are sterilize able, heat resistance, flame retardant, etc.[1]

### 1.2.3 Direct Print Micro-Dispensing

In order for any of the additive manufacturing technologies to be utilized for structural electronics manufacturing, a means of creating electrical interconnects between the electronic components that impart it with functional capabilities must be introduced. [6] The W. M. Keck Center at the University of Texas at El Paso has acquired a state of the art micro-dispensing system from nScrypt (nScrypt Inc., Orlando, FL.) The 3Dn-600HPx micro-dispense smart pump system can dispense fluids with a range of viscosities from 1 to 1,000,000 centipoise. This technology is used to dispense silver loaded inks in order to realize electrical interconnects between components. Once the interconnecting inks are deposited, if the ink is formed of silver ink particles suspended in a volatile binder, the parts need only be cured in order for the volatiles in the ink to evaporate to achieve electrical conductivity. The 3Dn-600 micro-dispensing system from nScrypt, boasts additional capabilities, from the 600X600mm of work area; a solid granite work area that acts as a counterweight to dampen vibrations as the fast moving dispensing head moves across the work surface; to unique grid mapping and path mapping technologies for conformal dispensing of irregular surfaces that allow it to maintain a fixed relative height from the work area irrespective of surface uniformity.

## 1.3 RESEARCH OBJECTIVES

The objective of this research is to chronicle a full design cycle of a novelty electronic device, in order to illustrate the advantages achieved by the utilization of the combination of Additive Manufacturing (AM) and Direct Write (DW) micro-dispensing to create a fully functional electronic device, where typical Rapid Prototyping (RP) can only achieve an outside physical representation of an electronic device. Furthermore, this research will include a typical prototype stage, where the electronic device will be manufactured using traditional materials and processes. A direct comparison between the two methodologies will be possible.

## 1.4 THE W. M. KECK CENTER

The W. M. Keck Center for 3D Innovation is a premier lab focusing on Additive manufacturing Technologies. Located in the engineering building at the University of Texas at El Paso, the Keck Center is home to a growing number of Additive Manufacturing equipment. The manufacturing labs contain equipment for functional rapid prototyping and tooling, micro fabrication, advanced materials, and 3D electronics. It has grown from one Additive Manufacturing machine in the year 2000 to 35 machines today, as it has expanded into 6,100 square foot facilities. It houses Additive Manufacturing capabilities that range from the table-top models to complex metal fabricating equipment. The lab includes:

2, Arcam Electron Beam Melting Machine (2, Arcam A2 EBM machine, Arcam S12 EBM Hot machine).

12, Stratasys Fused Deposition Modeling Machines (1, Fortus 900mc with the largest build envelope and highest throughput of any FDM machine available 36”x24”x36”; 1, Fortus 400mc; 1, FDM Titan; 7, FDM 2000/3000 – used as multi-axis deposition systems and FDM materials research machines; 2, UPrint tabletop models)

11, 3D Systems Stereolithography Machines (3, Viper si2 high resolution systems with 0.004” layer height; 1, SLA 500/5000; 7, SLA 250s – used primarily as research machines. Note: Of the 7 SLA 250s, 1 machine is the patented multi-material SL technology (US Patents 7,959,847 and 7,556,490), 1 machine is the patented SL and Direct Print technology used to fabricate 3D structural electronics (US Patents 7,658,603 and 7,419,630) and 1 proprietary sterile machine where an SL machine has been retrofitted inside a laminar flow hood for aseptic fabrication)

1, Custom Micro-Stereolithography Machine (with 2-micron resolution and multiple material capabilities). Dynamic mask projection μSL system including DMD, UV lamp, x-y and z stages, vat and platform, optical components and system hardware, and computer with control software as well as Radiometer for calibration of mercury lamp (EXPO, OmniCure R2000) and Large Area Imager for calibration of μSL optical system (Ophir-Spiricon, LBA-USB-L11058).

1, Objet PolyJet Machine (1, Objet Eden 333)

1, DTM Selective Laser Sintering Machine (1, DTM, now 3D Systems, Sinterstation 2000)

1, ZCorp 3D Color Printer (1, Z510) traditionally known as a 3D printer

1, EnvisionTec DLP Machine (1, Vanquish)

2, nScrypt Micro-Dispensing Systems (1, TableTop; 1, 3Dn 600 with integrated: 4 SmartPump print-heads, 2 laser systems for cutting/machining, etc., 1 DLP for light projection, 8 inkjet print-heads, and a component pick and place system)

3, Desktop 3D Printers (1, VFlash; 1, Fab@Home; 1, Solido SD300 Pro)

Since the early 2000's the W. M. Keck Center has focused on the processes required to build structural electronics, in which electronics can be fabricated in a 3D structure of arbitrary complexity able to accommodate human anatomy or simply to make more efficient use of a three-dimensional volume left available on a car, missile, satellite, equipment, airplane, etc. The applications of this technology are limited only by one's imagination and one day one may be able to download a file and print your next electronic device.

The facilities within the W. M. Keck Center for 3D Innovation (Keck Center) used here contain equipment purchased through Grant Number 11804 from the W. M. Keck Foundation, a faculty STARS Award from the University of Texas System, and two equipment grants from Sandia National Laboratories. This material is based in part upon work supported through the Mr. and Mrs. MacIntosh Murchison Chair I in Engineering and through research contract 504004 from Sandia National Laboratories in the Laboratory Directed Research and Development (LDRD) program. Sandia National Laboratories is a multi-program laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration contract DE-AC04-94AL85000.

## 1.5 THESIS OUTLINE

This thesis will contain, in addition to this chapter (Chapter 1 – Introduction) a review of the previous work performed in this field (Chapter 2 – Previous Work). The next chapter, (Chapter 3 – System Requirements) will describe the mechanical and electronic design required to achieve the work at hand. The following section will review the prototype phases achieved (Chapter 4 – Applications). The last chapter (Chapter 5 – Conclusions) will discuss the analyses reached through the performance of

this work. The section for References lists the research upon which this work attempts to build; and finally the Appendices section, includes a list of the programming code required to accomplish this work.

# Chapter 2: Previous Work

## 2.1 INTRODUCTION

Chapter 2 illustrates the chronology of other researcher's contributions to the field of freeform fabrication and to its utilization as a means for realizing Structural Electronics. Whereas some research described was performed at the University of Texas at El Paso, the author was not involved in the research described in this section, but would later benefit from the lessons learned. This author's contribution starts with the design and fabrication of the first generation in the smart dice project. This involvement is described starting in Chapter 3: System Requirements.

## 2.2 LITERATURE REVIEW

During the new product development cycle prototyping has an important purpose, for assessing fit, form and function of a new design before a significant investment is made in tooling. Until recently each prototype iteration could mean delays of weeks or even months, as such, only a few iterations could be made before tooling was created, resulting in parts that were, at best, not optimized or at worst, parts that would not work properly. [1] Decisions made during the prototype stage have a lasting and important effect on the manufacturing process of the final part; its impact varies depending on the product life cycle. [8]

Additive Manufacturing technologies have over 20 years of history; their major impact has been in the creation of prototypes allowing reduction in "Time to Market". [7] And while their focus has started to increase to include the fabrication of molds and dies, their main impact remains in that area. A major trend for Additive Manufacturing processes is the pursuit of new applications, as these technologies increase the availability of functional materials a goal of producing functional prototypes has emerged, whereby the part made through AM fulfills the functional requirements of the traditionally created part, not simply the prototype function. [5] This goal however remains elusive especially when the traditionally fabricated part includes more than mechanical features; if the part contains electronic functionality typical Additive Manufacturing techniques fall short. There are those who have seen this limitation and are working to overcome it.

The motivation to embed the electronics components required to accomplish a functional electronic or electromechanical device has been present for at least two decades. This research can be traced back to at least 1992 when Lee E. Weiss and Fritz B. Prinz of Carnegie Mellon University repackaged a simple electronic game in polyurethane foam to demonstrate that capability. They achieved this by depositing two-part polyurethane foam, after allowing it time to cure, machining it to allocate space for the electronic components and thermally spraying of metals using a mask to ensure the proper location of the interconnects. While their methodology still required several steps that would make it impractical as a direct substitute for traditional prototyping and was more directed towards the achievement of a functional electronic device regardless of time and laboriousness, it hinted at the capability of achieving functional electronics that could be packaged through non-traditional methodologies. [9] This methodology was patented by Lee E. Weiss and Fritz B. Prinz in 1994 and with funding from the Defense Advanced Research Projects Agency (DARPA) in 1996 demonstrated the repackaging of a small personal computer for divers into a housing that would make it conformal to a diver's leg and waterproof to 100 feet. Again, whereas this achievement can be considered a success in the accomplishment of the goal it sought, it was nonetheless a laborious technique that required long process times, a combination with subtractive methods, various manual tasks and the use of pre-made Printed Circuit Boards (PCBs) to achieve device functionality. These PCBs were already available to them, whereas for a new process, the fabrication of new PCBs would still have to be considered as a significant time delay. Figure 2.1 shows the steps required to assemble two layers of PCBs [10, 11]

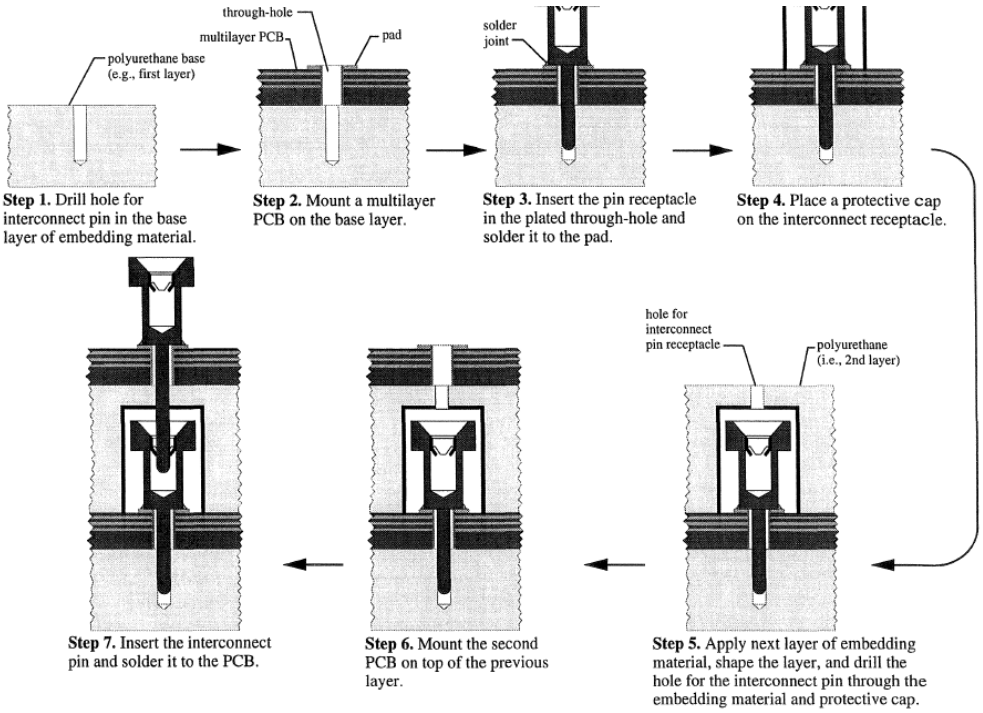



Figure 2.1 Steps for assembling two layers of PCBs [11]

This research was further expanded by Lee E. Weiss and Fritz B. Prinz when they integrated the two-part polyurethane foam dispenser into a 5-axis Computer Numerically Controlled (CNC) milling machine in order to take advantage of this equipment's precision capabilities to improve process repeatability for foam deposition as well. And the following year, Lee E. Weiss *et al*. improved upon the concept by developing algorithms to optimize these processes. Figure 2.2 shows the integration of the two-part polyurethane foam dispenser into the precision 5-axis CNC milling machine. [12, 13]

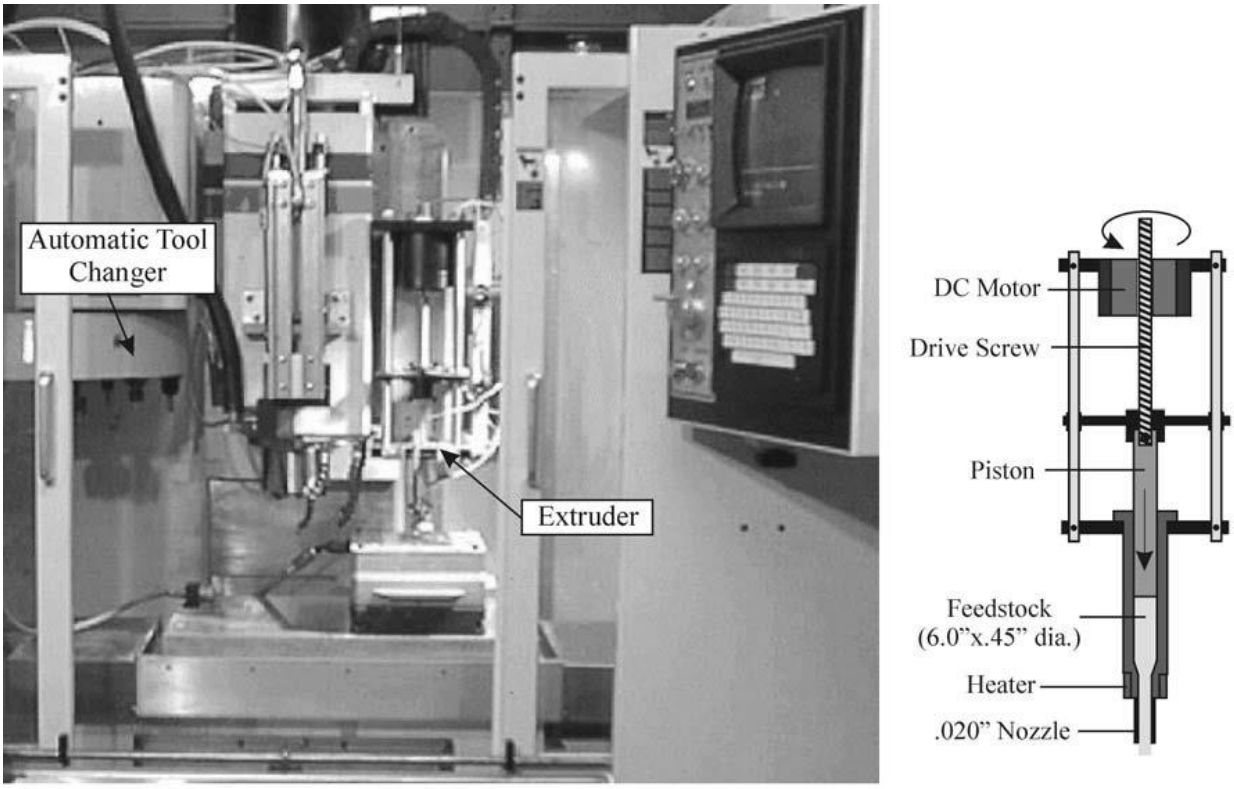



Figure 2.2 CNC milling machine with adapted polyurethane foam dispense system [12]

The previously described research accomplished the goals currently sought by many researchers, the integration of functional electronics into a newly built housing but at the cost of a significant time delay required for polyurethane foam to cure, a persistent need for subtractive processes required to form cavities for components and to remove excess materials as these were made as near-net shapes and with a methodology that reuses PCBs instead of creating an alternative to them.

The greater availability of Additive Manufacturing equipment with its capability for precision deposition of materials to create accurate, arbitrary, tridimensional shapes saw the interest in the previously described research fade. The project in this thesis, unlike the previously described research conducted by Lee E. Weiss and Fritz B. Prinz, takes advantage of the Additive Manufacturing technology known as Stereolithography to quickly build up a three-dimensional object and include the electronic components required to have a functional electronic circuit included; realizing a major improvement in time reduction to achieve a functional circuit within a custom three-dimensional volume.

The need for integration of electronic components and electronic interconnects to achieve functional structural electronics was still present and in 2004 Jeremy Palmer from Sandia National Laboratories in conjunction with Ryan Wicker from the University of Texas at El Paso and their teams proposed a system that would combine the high build precision of the Additive Manufacturing technique known as Stereolithography with the material dispensing capabilities and precision of Direct Write technologies to deposit silver loaded inks to provide the electrical interconnects between components to provide true functionality [6]. This methodology was patented in 2008 [14]. The Direct Write capability described is a proprietary technology developed by nScrypt (nScrypt Inc., Orlando, FL.) and is described in a paper by Ken H. Church where this equipment is used to dispense conductive lines onto glass substrates to create sensor systems. Additionally it boasts capabilities to dispense materials with a viscosity range from 1 to 1,000,000 centipoise, whether conductive or insulating materials with picoliter volume control achieving line widths as small as 25microns and capable of depositing at 250 mm/second. Additionally the paper describes the capability of depositing material onto curved surfaces through 3D surface mapping. [15] The research was further expanded with the creation of a custom built machine that integrated a Stereolithography machine SLA 250 (3D systems, Valencia, CA.) with an nScrypt smartpump$^{TM}$ (nScrypt Inc., Orlando, FL.) to demonstrate the capability of printing conductive traces onto an intermediate layer of Stereolithography, cure the ink on site and then continue to build additional layers of Stereolithography. This research demonstrated the capability to attain conductivity between the inner conductive trace and a serpentine printed on the top layer [16]. Figure 2.3 shows the setup of how the Stereolithography equipment is integrated with a Direct Write micro-dispense pump to achieve these electrical interconnects.

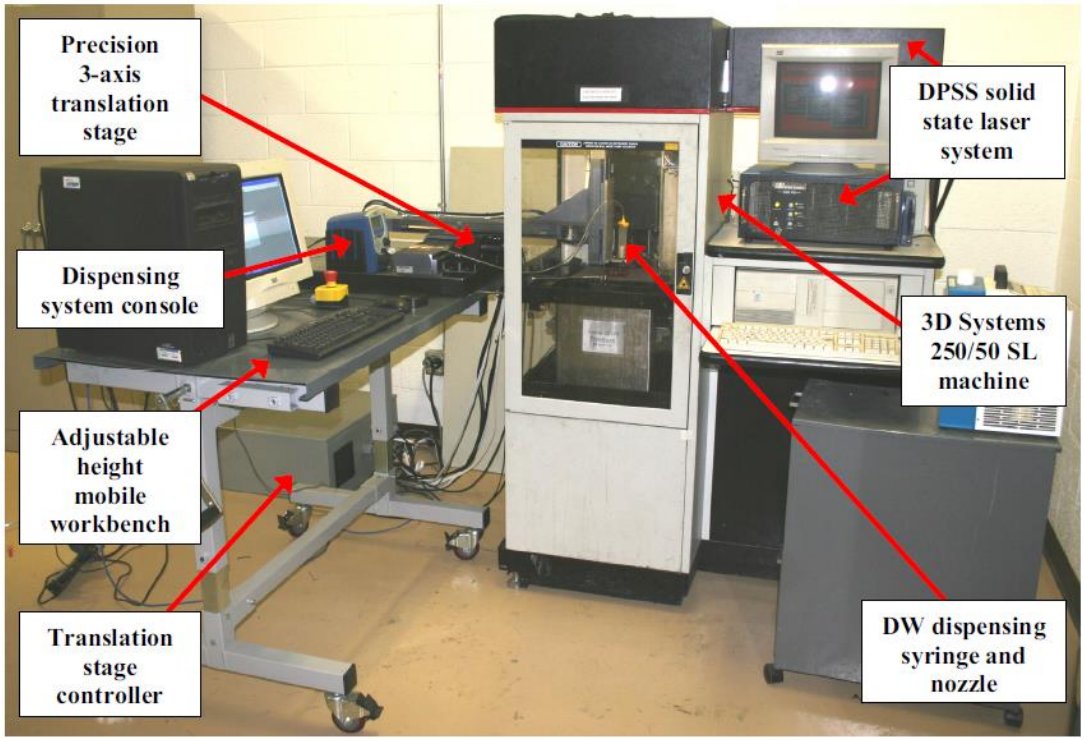



Figure 2.3 Integration of SLA 250 with D.W. pump [16]

The following year, that proof-of-concept design was applied to a functional circuit embedded within cured Stereolithography resin by halting the process mid-build, placing components into custom designed cavities, then continuing the Stereolithography process and finally through Direct Write, depositing electrical interconnects to complete the circuit. A 555 oscillator circuit was fabricated using a thermistor; the circuit would flash an LED at a rate proportional to the temperature perceived by the thermistor. Figure 2.4 shows the 3 stages of the process: a) Shows the point where the Stereolithography process is stopped, the cavities for the components are already made. b) Shows the stage after components were placed and embedded by successive layers of Stereolithography and c) Shows the completed circuit when Direct Write (DW) was used to deposit electrical interconnects to complete the circuit. [17, 29]

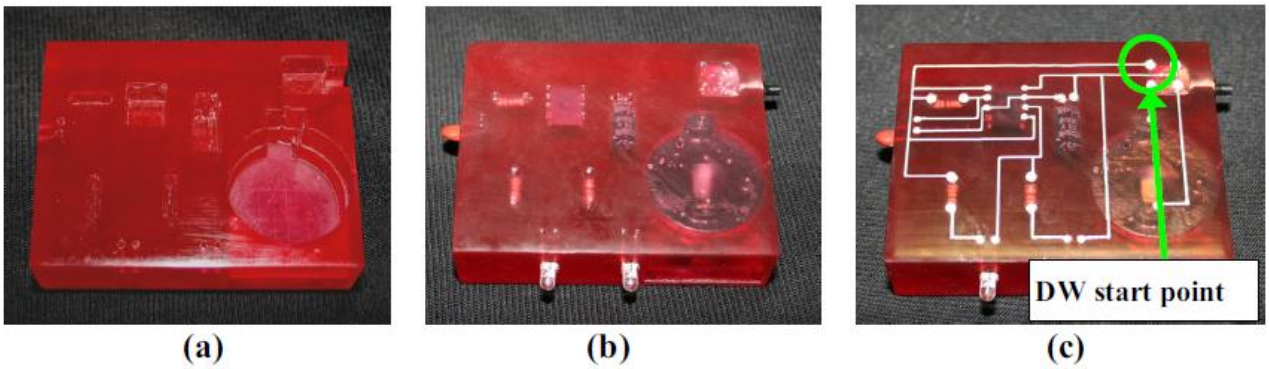



Figure 2.4 Three stages of the build process. a) The SL build process is stopped b) The components are embedded c) The circuit is completed through Direct Write [17, 29]

In 2007 Daniel Periard, Hod Lipson and Evan Malone from Cornell University utilized their table top Additive Manufacturing apparatus, the Fab@Home to create simple, yet functional circuits to demonstrate the capabilities of their system. Utilizing dispensed silver loaded conductive silicone to form electrical interconnects; they were able to create a 555 oscillator timer in 2 and 3 dimensions, a flashlight and a children's toy figurine with LEDs for eyes that would light up when a switch in its stomach was pressed. [18]

Figure 2.5 shows the toy figure with LEDs for eyes during the fabrication process and the completed part as it was tested.

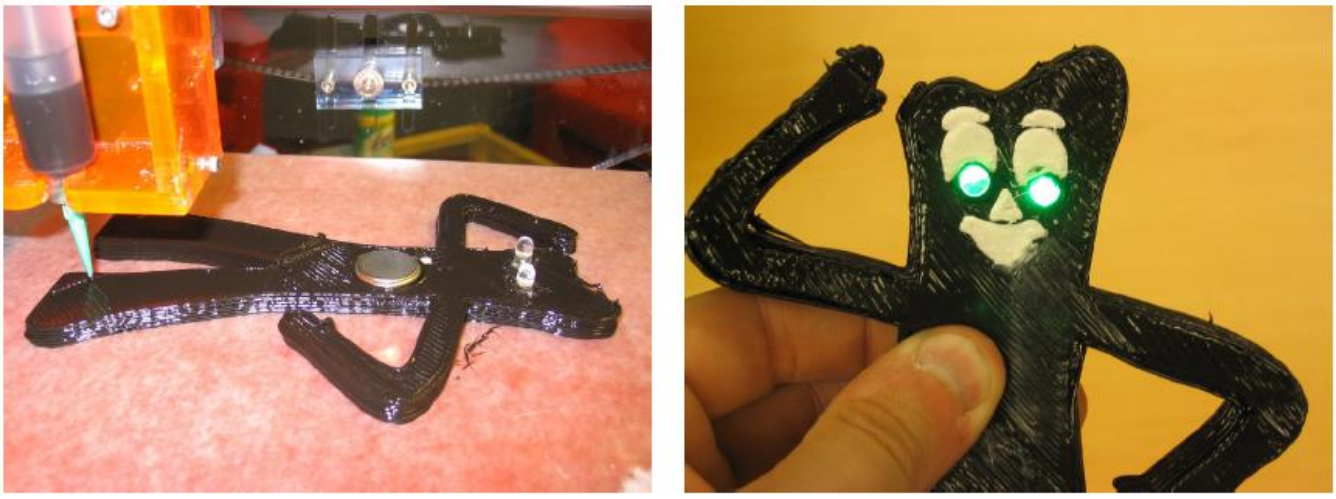


Figure 2.5 Toy figure being fabricated and tested [18]

This research was further expanded by the team of Evan Malone and Hod Lipson when they attempted the fabrication of a "Complete Electromechanical Relay". In their paper, they describe utilizing the same Fab@Home equipment to dispense "ionomeric polymer-metal composite actuators (IPMC)" and utilize their conductive surface to selectively open or close a load circuit. This research was successful in demonstrating the feasibility of creating a printed relay with limited capacity, however their overall goal of printing a complete circuit including: battery, transistor and actuator hit a roadblock when the current supplied by the printed battery [19] was appropriate for the electro-mechanical relay requirements, but three orders of magnitude larger than the current handling capabilities of organic printed transistors. [20] Figure 2.6 shows the summary of the issue with the attempt at a fully printed circuit.

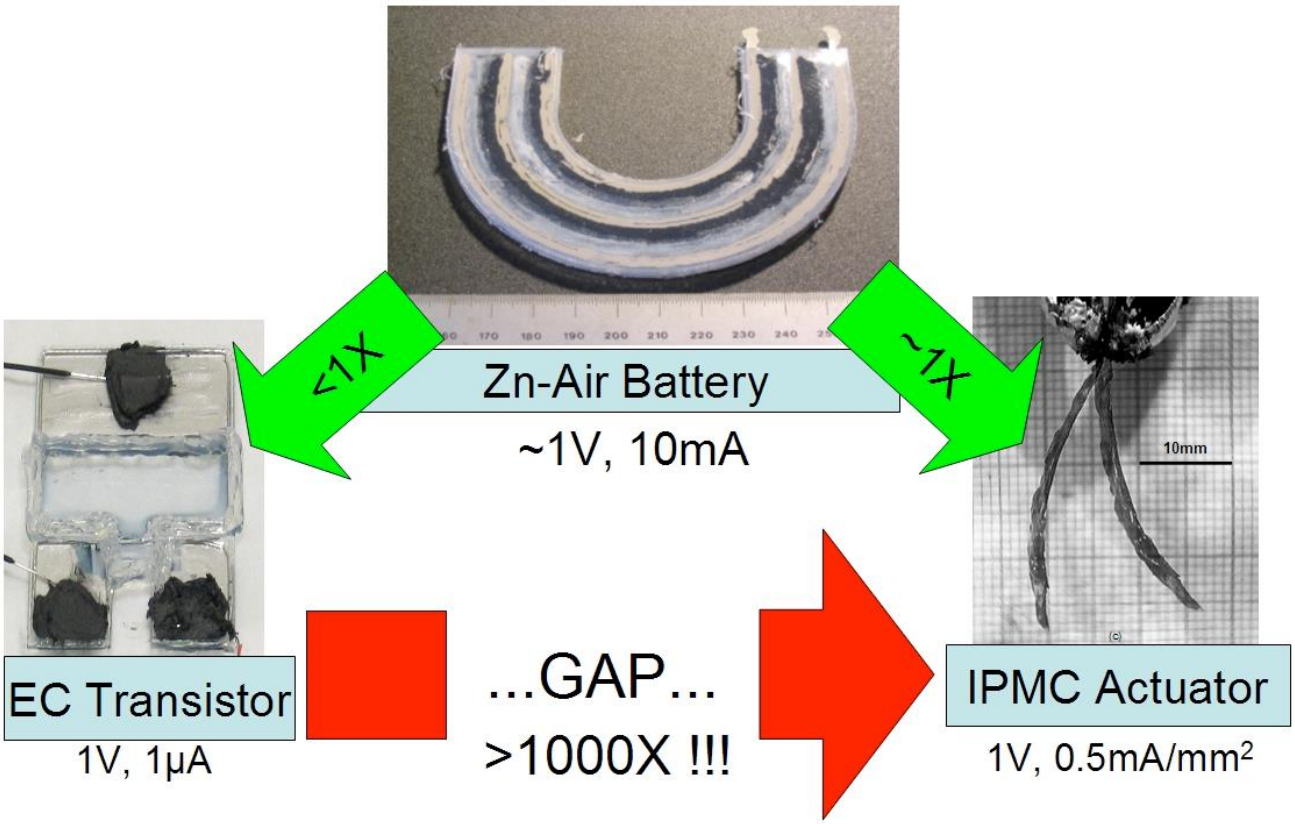



Figure 2.6 Printed battery and IPMC actuator are compatible, transistor is not [20]

The previously described limitation illustrates the complexity of the problem in trying to dispense and create every component of a circuit to achieve complete printed fabrication. On the other hand, the University of Texas at El Paso has taken a philosophy of: "Print what you can, embed what

you can't" to freeform fabrication. With this perspective, during the research on this thesis, the best components are used to achieve the stated goal, regardless of whether they are printed or otherwise sourced. The goal is to achieve functional and practical circuits and to utilize the advantages of available technologies as appropriate for the task at hand.

The research at the University of Texas at El Paso was further advanced by Misael Navarrete and Amit Lopes from University of Texas at El Paso and Jeremy Palmer from Sandia National Laboratories when a wireless motion sensor with GPS capabilities was built using off-the-shelf components with the previously described integrated SLA and DW equipment. This sensor was envisioned as an infrared motion sensor that could be embedded in a volume designed to appear innocuous. The Stereolithography part was designed to simulate a rock, designed to blend in with its environment but contain a motion sensor with the capability to detect movement and transmit its GPS coordinates to a central coordinating location. Figure 2.7 shows the design of the detection circuit, the location of the circuit within the SL volume and the final appearance of the part. [21]

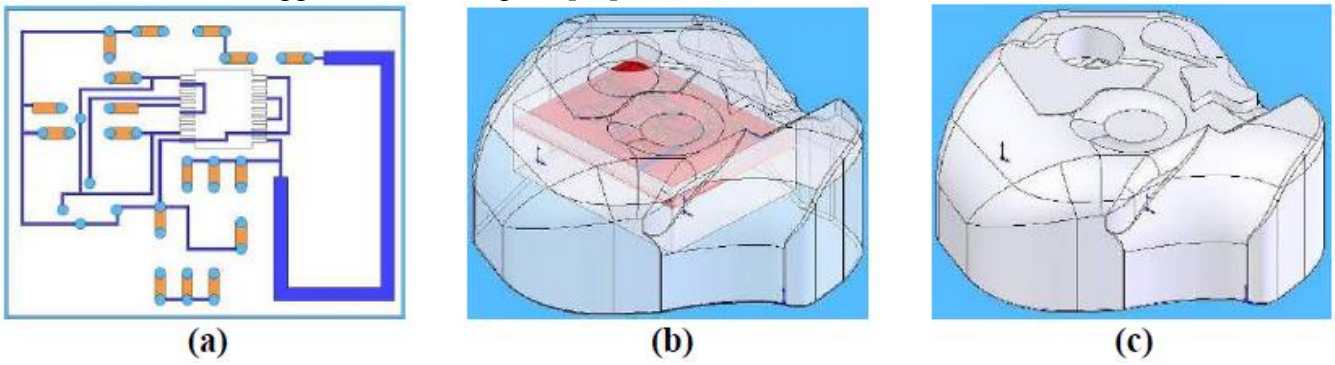



Figure 2.7 a) Circuit layout. b) Location of circuit within SL volume. c) Final appearance [21]

The capabilities of the Direct Write dispensing system were limited by its location on x-y capable stages; as such the circuits presented thus far have been limited to placement within the x-y plane. The next stage of research at the University of Texas at El Paso attempted to address this by creating a circuit that would require component placement in more than one plane. The paper presented by Eric DeNava, Misael Navarrete and Amit Lopes in 2008 demonstrated a 3-axis magnetic flux sensor (or magnetometer) that was only feasible by placing three hall-effect sensors, at least one of which must be place in a perpendicular plane to the other two.

The added advantage of component placement in more than one axis is the advancement towards the long sought-after goal of better volume utilization. In this search for better volume utilization a shortcoming of previous research was addressed, when the problem of trace width being a function of the silver ink viscosity and deposition parameters was instead controlled through a strategy that created the circuit traces before-hand during the Stereolithography process. The electronic circuit was designed and included in the Stereolithography process as channels that would contain the dispensed ink and would also provide a separating wall between closely spaced traces, eliminating the risk of bridging between them. This methodology allowed for a much tighter trace spacing because the channel dimensions were determined by the high accuracy capabilities of the laser in the Stereolithography apparatus and taken away from the lesser capabilities of the ink-dispensing system and the ink characteristics. Figure 2.8 shows the magnetometer in three stages of build. The first section shows the Stereolithography part where the circuit has already been built as cavities for the components and channels indicating the traces where silver ink will be deposited, the next section shows the components already placed in the cavities and the third shows the silver ink interconnections have been made. [22]

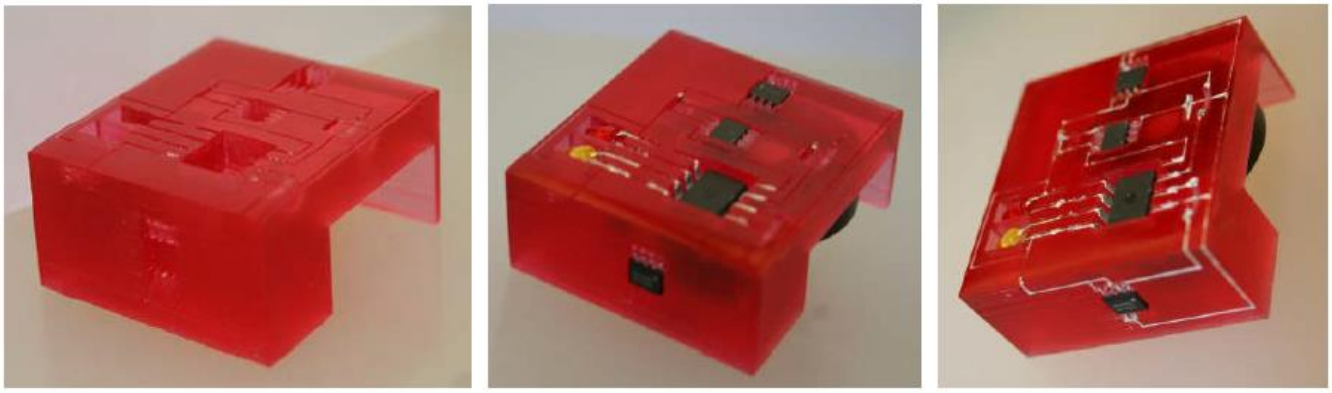


Figure 2.8 a) SLA built part. b) With components. c) With silver ink interconnects [22]

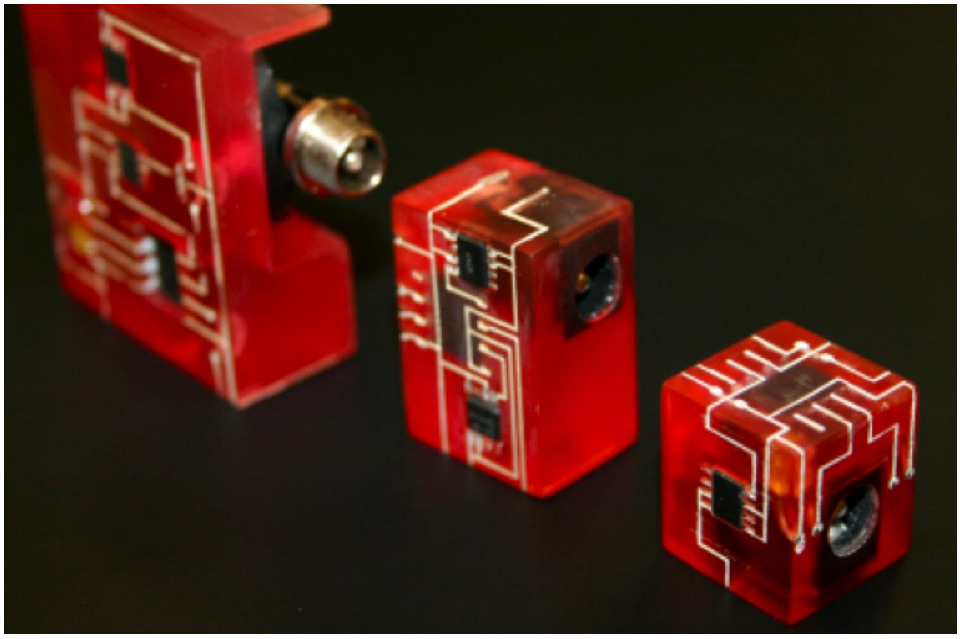


Figure 2.9 Three generations of magnetometer design. [23]

This design was later improved through superior strategies for component placement which improved volume utilization by the use of more faces for component placement. This achieved a smaller design that had the same capabilities as the original. Figure 2.9 shows two subsequent generations of the device where each achieved a significant volume decrease. Whereas this research succeeded in component placement and interconnection in more than one plane, these were still flat planes. The next research step at the University of Texas at El Paso was research that would allow the placement of components in curved planes that truly conform to irregular shapes. In the research presented by Richard Olivas in 2008, he describes improvements on the magnetometer design. One change in the design was the use of surface mount component packages for many of the circuit components. Additionally, tighter trace spacing was achieved with this change. The most significant departure from previous designs was that two planes are wrapped in a cylinder shape; this was achieved by the placement of components in a curved surface. Figure 2.10 shows the top of the fourth generation magnetometer, with a decidedly different form factor than the previous generations, achieved by component placement over a curved surface.

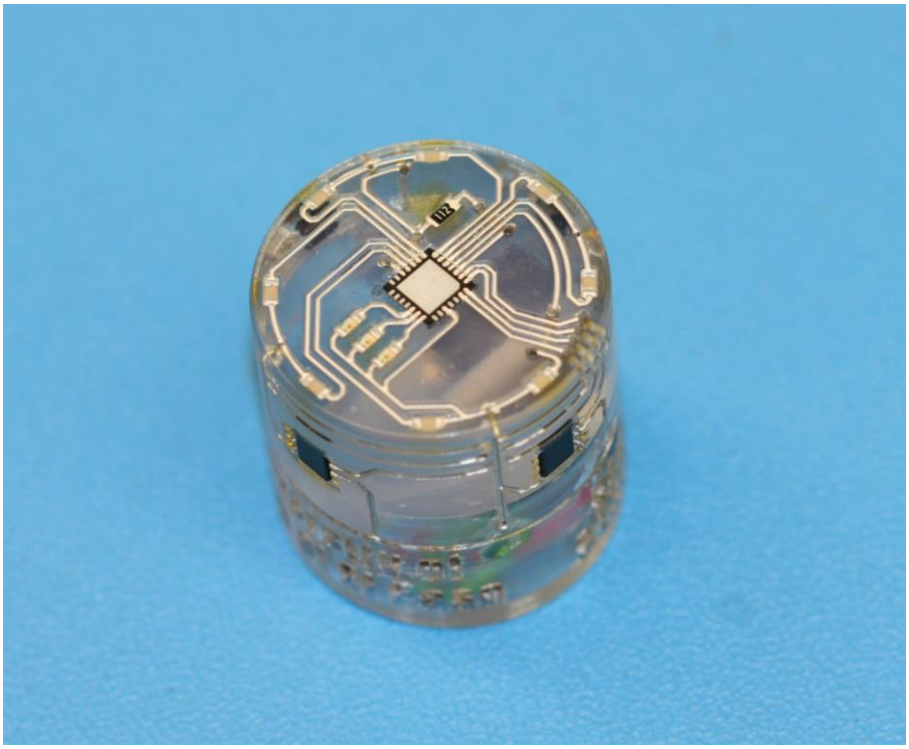


Figure 2.10 Fourth generation magnetometer [23]

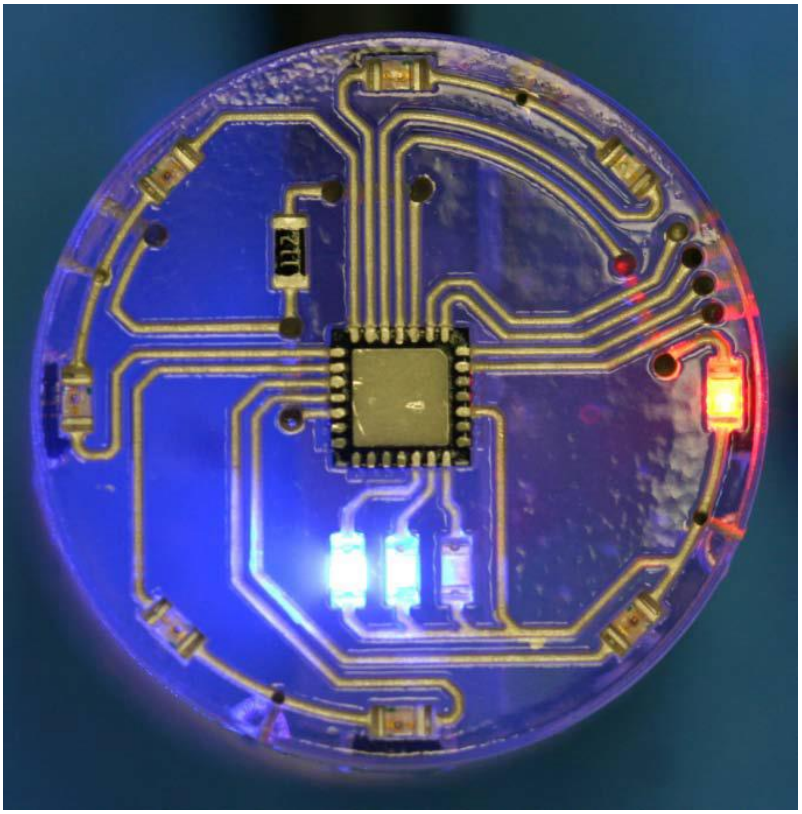


Figure 2.11 Magnetic field is detected from the right side and has 2 of 3 degree of intensity [23]

This change, along with clever micro-controller code changes allowed for a clearer indicator of magnetic field location, which was achieved by LED placement along the curved top; furthermore a pseudo “bar” light indicator, made up of three LEDs indicates intensity of magnetic field detected. Figure 2.11 shows the top of the magnetometer with closer trace spacing, surface mount components and better LED indicators of direction and intensity of magnetic field detected. [23] This newly developed methodology was utilized during my thesis research with the creation of a power module in a cylinder shape with a lithium polymer cell on the inside. The use of surface mount components was also exploited in order to reduce the area utilized. The techniques described, of channels built into the mechanical design and the use of surface mount components, were later exploited by this author’s research by allowing a small circuit design to be contained within the volume allowed by a regular size dice.

While this research was developed at the University of Texas at El Paso, there were other avenues being pursued by other investigators. In research presented in 2007 by Craig Arnold, Pere Serra and Alberto Pique they describe a methodology known as Laser Induced Forward Transfer or LIFT where a laser can be used to displace material from a ‘donor’ substrate into an ‘acceptor’ substrate as a means to build lithium based micro-batteries, micro ultra-capacitors, micro-solar cells and can also be used to create electrical interconnects between components with high resolution patterns. The technique requires the donor material to be a laser transparent substrate, to be coated with a thin layer of the material to deposit and to be placed in close proximity to the acceptor material. The laser is incident on the donor substrate and above a certain energy threshold; material is ejected from the source and propelled towards the receiving substrate. Figure 2.12 shows a standard arrangement.

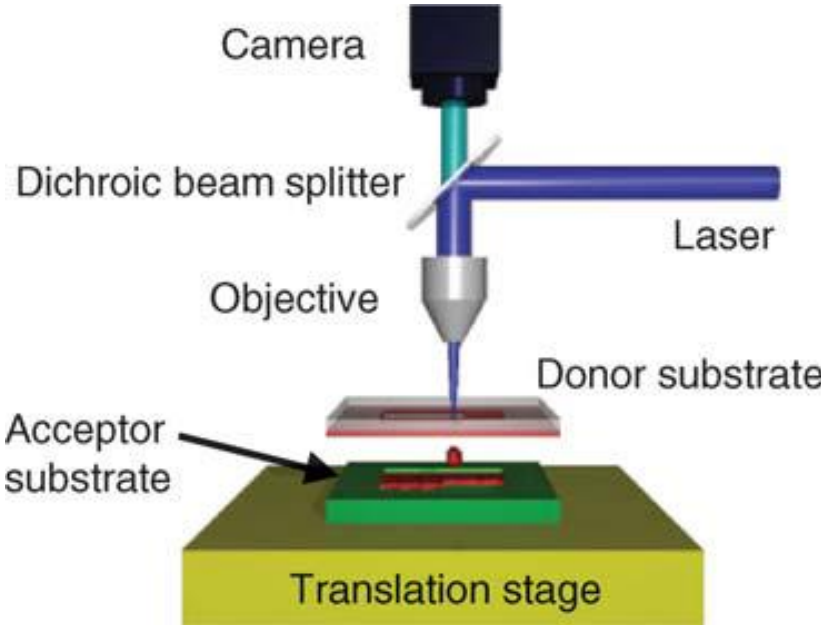



Figure 2.12 Standard arrangement for LIFT system. [24]

In their paper, Arnold, Serra and Pique demonstrate their technique by recreating a simple 555 timer flashing circuit by first using a laser to do micromachining of cavities to place passive components and an unpackaged 555 timer in bare die form, after placing the components; they were all encapsulated in a layer of polyimide. The interconnects first required the use of the laser to create vias in the polyimide layer to expose the pads on each device, through the LIFT technique, conductive silver ink is deposited creating the electrical interconnects between the components. The completed flasher circuit occupies a total area smaller than a packaged 555 timer chip. Their technique demonstrates the capability to fabricate functional circuits embedded on a substrate, where the substrate becomes both circuit board and packaging. Figure 2.13 shows the completed circuit. [24]

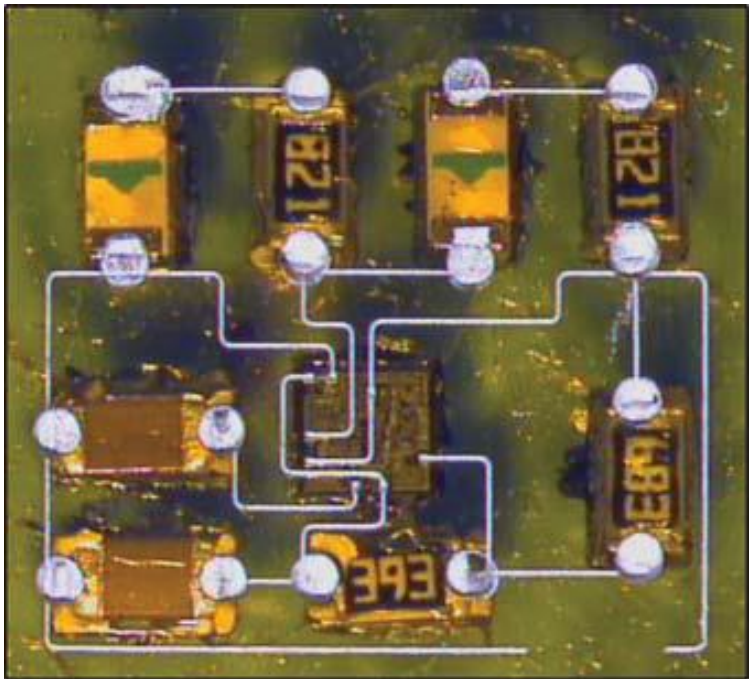



Figure 2.13 Completed 555 flasher circuit achieved through the use of LIFT technique [24]

Whereas this LIFT technique achieves essentially what the other research described, has been striving for: completely embedded circuits in a very small substrate on a flat surface. For the type of research described elsewhere in this work, the achievement of deposited thin traces is outweighed by the limitations due to the requirements to achieve them. For instance, for the power module design described in this thesis for the rechargeable lithium-polymer battery systems, the cylinder shape could not be realized using this methodology.

Another avenue of research has been pursued by Brent Stucker of Utah State University and Ryan Wicker of the University of Texas at El Paso. Through Office of Naval Research funded project, they pursued a project to integrate Ultrasonic Consolidation and Direct Write material deposition to achieve embedding of circuits encased in metal for the purpose of fabricating devices that would be of interest to the Navy. [25] This research was later expanded by Ludwing Hernandez of Utah State with Brent Stucker as his advisor in which he further pursued the project of integrating Ultrasonic Consolidation and Direct Write Micro dispensing to embed an electrical system within a metallic enclosure. Ultrasonic Consolidation is a methodology for Additive Manufacturing in which solid parts are created by joining metal foils through localized high frequency vibration and compression.

Ultrasonic equipment takes 60Hz electrical energy, converts it to 15-40 KHz energy which is supplied to a converter, which transforms it into mechanical energy at ultrasonic frequencies. This energy is transmitted to a booster which elevates the frequency and transmits it to the horn. The horn is placed in contact with the base material and thus transfers this vibration energy to them. Figure 2.14 shows a basic schematic of the ultrasonic equipment.

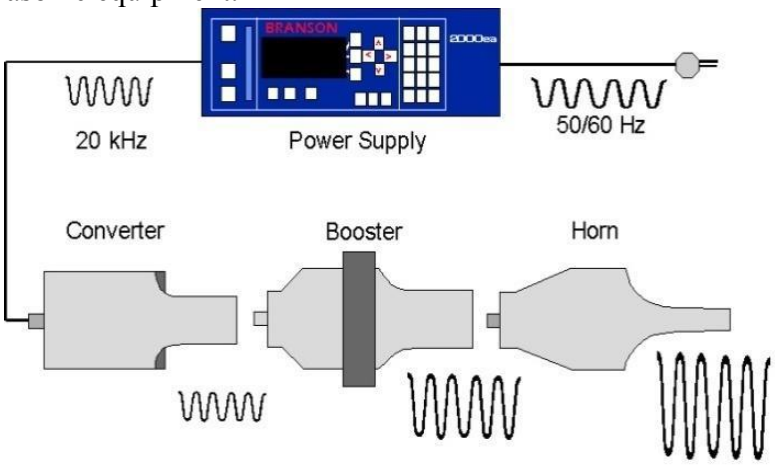



Figure 2.14 Schematic of ultrasonic equipment [26]

Ultrasonic Consolidation does not require a special atmosphere to operate, it does not require high temperature, and its operating range varies from room temperature to over 200° C. Given that the process does not involve material changing phase, dimensional errors due to shrinkage, residual stresses and other metallurgical incompatibilities are not an issue.

In their research Hernandez and Stucker describe the fabrication of a touch sensor and LED as proof of concept of their capability to embed a functional circuit within a metal enclosure. The process starts by machining a channel into a base metal, the inner surface of the channel and the surrounding area is coated by hand with a thermoset insulator to prevent short circuits; a combination of printed and off-the-shelf electronic components was used to create a touch sensor and J/K flip-flop to detect human touch and to indicate by lighting up an LED. The circuit was then tested for functionality and later covered with additional layers of metal to completely encase it, making sure the sonotrode is unable to contact the circuit parts, as it would destroy them. Figure 2.15 shows the steps to achieve this: a) The metal is machined to create a cavity for the circuit; b) The inner surface of the channel is coated by hand

with a thermoset insulator; c) The passive circuit components that can be printed with the Direct Write system are printed; d) Additional components are added to complete the circuit. [26]

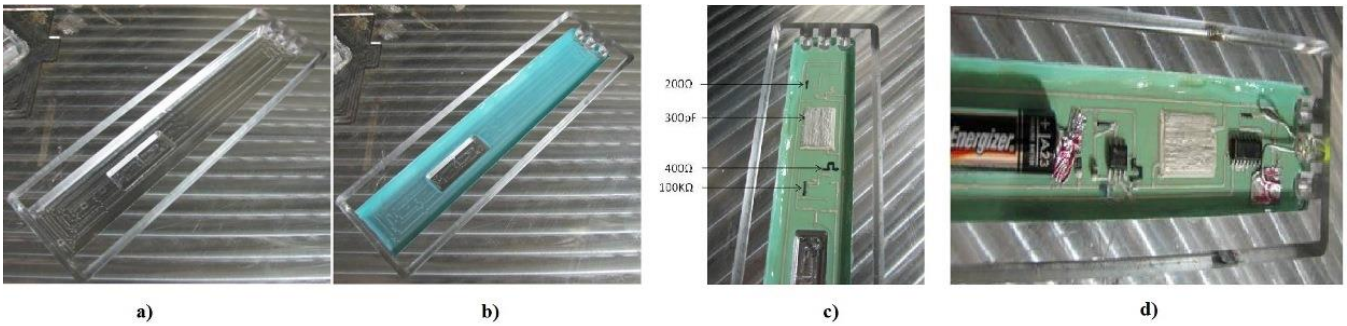



Figure 2.15 a) Machined cavity. b) Applied insulator. c) D.W. components. D) Completed. [26]

This avenue of research has distinct advantages and problems that must be discussed. While the ability to embed a circuit in metal provides it with strong mechanical protection, the added difficulty of having to create a cavity and applying a thermoset dielectric to create an insulated surface for it makes the use of metal not worth the effort. Likewise the ability to print some passive components through the use of Direct Write micro dispensing is undercut by its inability to print all components, is the extra effort necessary to print some components worth it if the additional step of physically embedding other components will not be eliminated? Additive Manufacturing techniques that employ a material that provides an insulated substrate for a circuit appear better suited for the integration of electronic circuitry. For the smart dice project, the utter number of interconnects required would make the subtractive process required so time consuming (if possible at all), that it would not be practical as a rapid prototyping technology.

To complete this literature review, an additional method for conductor deposition must be discussed. In 2012 a research paper by Wesley Verheecke of Lessius Mechelen University College campus De Nayer along with Matthias Van Dyck, Frederick Vogeler, Andre Voet and Hans Valkenaers describes their efforts to optimize the operating parameters for an Aerosol Jet Printing (AJP) process. The AJP process was developed for the manufacture of customized micro-electronics and is becoming an alternative for thick-film and printing processes, like screen-, stencil- or inkjet printing. The AJP process is very similar to an inkjet process, but is more tolerant in the distance from the nozzle to the

work area. As such, it is gaining notoriety and acceptance for the fabrication of 3D conformal electronics. The basic AJP system consists of two important components: the atomizer and a focusing module; inks with a viscosity range from 1 to 1,000 centipoise and particle size of up to 500 nanometers can be atomized. The atomizer uses a high velocity gas stream to generate the aerosol. These particles are transported through the virtual impactor. After the aerosol is generated in the atomizer (1) and passes through the virtual impactor (2) the print head (3) focuses the aerosol into a concentrated beam by adding a concentric mantle of sheath gas (4) around the aerosol. The focused aerosol beam is directed by the sheath gas to the substrate. The sheath gas maintains the droplets tightly focused over a distance of 3mm to 5mm giving the AJP system the ability to print over uneven surfaces. Figure 2.16 shows a schematic diagram. In their research Wesley Verheecke and team worked with the AJP equipment to develop appropriate operating parameters to create a robust dispensing process. [27]

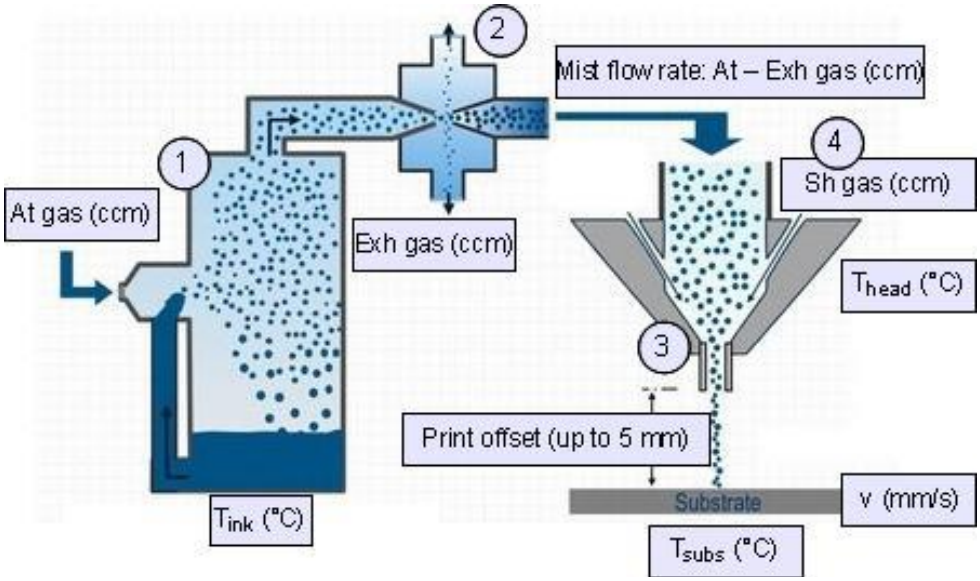



Figure 2.16 Schematic of the AJP process [27]

This research was further developed by Jason A. Paulsen, Michael Renn, Kurt Christenson and Richard Plourde from the Aerosol Jet Advanced Applications Lab at Optomec Inc. As manufacturers of the Aerosol Jet equipment, Optomec had an interest in demonstrating their equipment's capability and in their 2012 paper demonstrated an application of this non-contact technique for printing conformal circuits. In this paper, they describe printing of a conformal antenna array, embedded circuitry and

sensors and electronic packaging. The Optomec paper describes the ability to print over surfaces that are substantially planar but with some curvature. This is applicable for the capacity to include embedded sensors or antenna onto non-planar surfaces, such as aircraft fuselages or body armor on military personnel. Figure 2.17 shows an application where a silver nano-particle ink was printed onto a rigid, but curved surface, achieving a mechanically robust solution.

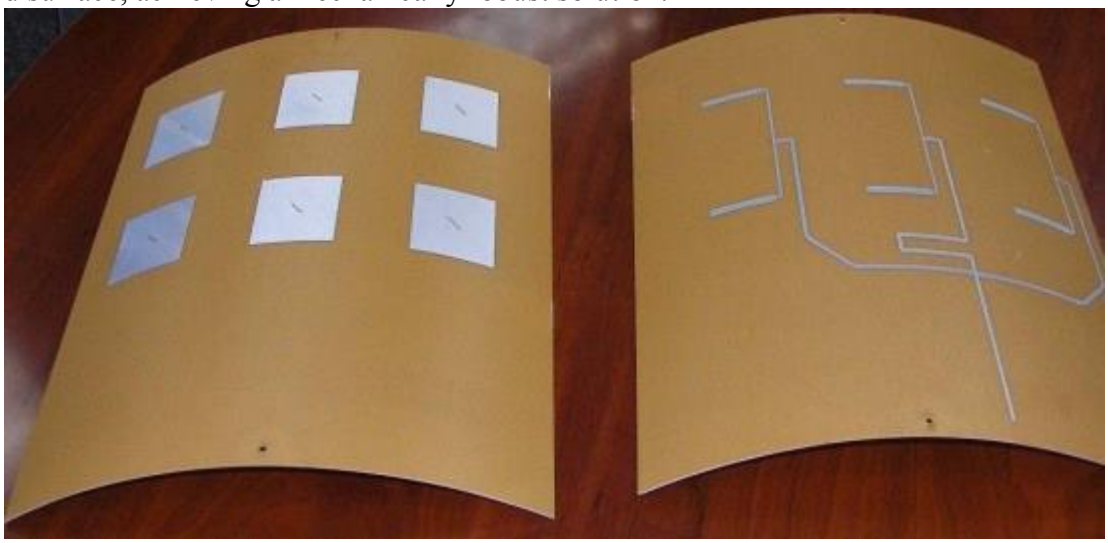


Figure 2.17 Aersol Jet deposition of nano-particle silver ink on curved surface. [27]

The Jason A. Paulsen paper goes on to describe another demonstration of capabilities where a conformal sensor, antenna and power distribution circuit were printed onto a mock UAV structure that itself was printed using Fused Deposition Modeling. The wing surface is grit blasted to improve ink wetting characteristics and a dielectric undercoat was applied to smooth the surface and fill fissures. Circuits were then printed on the wing to power the LEDs and propellers. A 5.8GHz T-slot antenna and strain gage sensor were also printed directly onto the wing. Figure 2.18 shows the wing with printed circuits. [28]

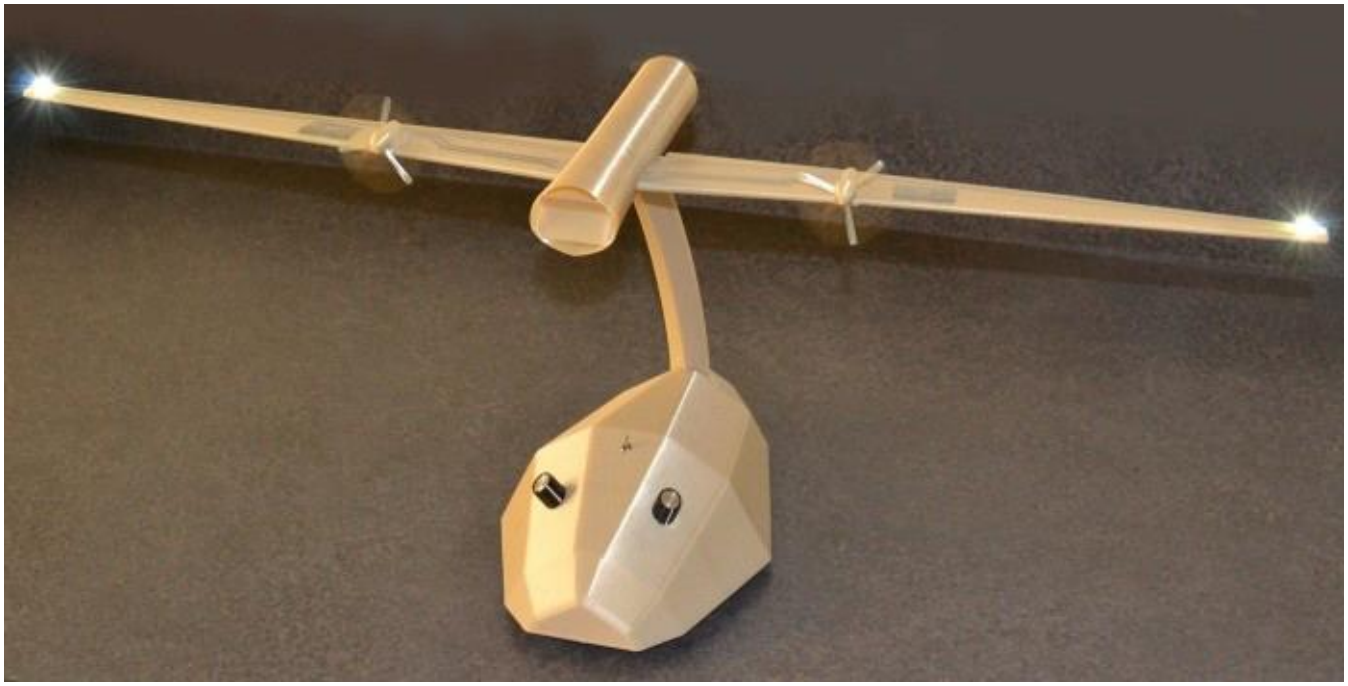


Figure 2.18 FDM created wing with Aerosol Jet printed circuit and antenna [28]

The Aerosol Jet technology has great potential to be utilized for depositing electrical interconnects in conformal circuits. It does also have limitations that must be explored before it can be considered a better alternative to Direct Write micro-dispensing. Given the ink is "tossed" from the nozzle instead of deposited; it is much more tolerant to separation variations between the nozzle and the work surface. On the other hand, it is a process designed for nano-particle inks, perhaps too small for the typical structural electronics application. The viscosity range of the DW system is 1,000 times larger, able to handle materials 1,000 times more viscous than the highest capacity of AJ; therefore corresponding limitations are likely to be found. The Jason A. Paulsen paper describes the need for a treated surface in order to achieve proper deposition. Likewise, a system that deposits aerosol droplets is likely to end with an exceedingly thin layer of ink and given the formula for actual resistance in a conductor $R=\rho L/A$ for an equal width and length, the cross-sectional area is directly proportional to the thickness resulting in high resistance, or a need for several passes with the dispensing nozzle.

All of these questions must be answered and are surely being researched, before Aerosol Jet can be considered as the logical successor for DW technologies. In the meantime, the University of Texas at

El Paso continues its research using DW for 3D structural electronics integration and looking for better alternatives as well.

# Chapter 3: System Requirements

## 3.1 ELECTRONIC DESIGN

As previously described, the University of Texas at El Paso has developed a methodology for electronics integration with Additive Manufacturing where the electronic circuit is laid out and included in the mechanical design of the structural device before the process for Additive Manufacturing takes place. As a result, the electronic design of these devices is done before the mechanical part is built.

### 3.1.1 Basic dice block diagram

The basic operation of the electronic dice utilizes an accelerometer to determine its orientation and this data is transmitted to a micro-controller. The micro-controller utilizes the orientation information to determine when the shaking or throwing motion has stopped, at that point it determines the side of the dice that is oriented towards the top and energizes LEDs on that surface to indicate it. Given the available computing power of modern micro-controllers, the LEDs are energized in an eye-catching sequence, and then the circuit is placed into a power-saving sleep mode to save battery power. The circuit is nevertheless allowed the capability to detect movement, such that when it is picked up and shaken again, the sequence can be repeated. Figure 3.1 shows a simplified block diagram of the dice circuit, this diagram illustrates the simplicity of the concept. The implementation of this simple concept into a fully functional prototype would require integration of several important steps.

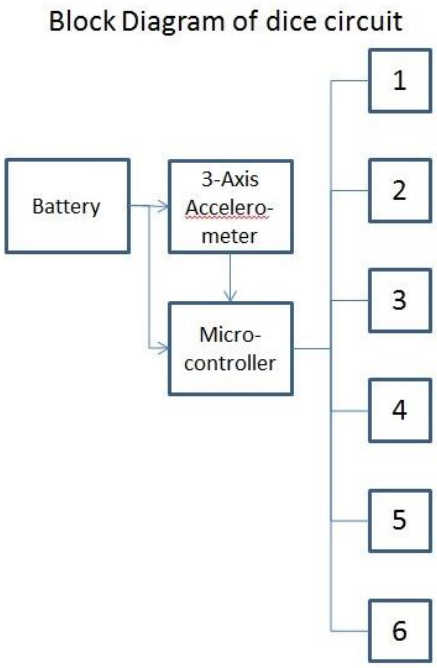



Figure 3.1 Block diagram of dice circuit

### 3.1.2 Lithium polymer cell, charge control circuit

Whereas the first iteration of the dice design utilized two 1.5V zinc-air or alkaline LR-44 size coin cell batteries, subsequent versions would utilize a single-cell 3.6V lithium polymer rechargeable battery with 45mAh energy storage capacity. As a result of this, a charge control circuit was required for the safe operation of this type of battery. Lithium polymer batteries have high energy density, which means they have a large amount of energy as a proportion of their overall mass, when compared to the other two options for batteries used in portable equipment: nickel-cadmium and nickel-metal hydride. In addition, lithium polymer batteries don't suffer from the "memory effect" that causes nickel-cadmium and nickel-metal hydride to lose energy storage capacity when they are not allowed to charge to full capacity during their charge cycle. On the other hand, this high energy storage capacity comes with an increased danger should the cell fail. Lithium chemistry cells cannot accept overcharging; anything above the stated capacity causes stresses. Figure 3.2 shows the proper charge sequence required to safely charge a lithium polymer cell.

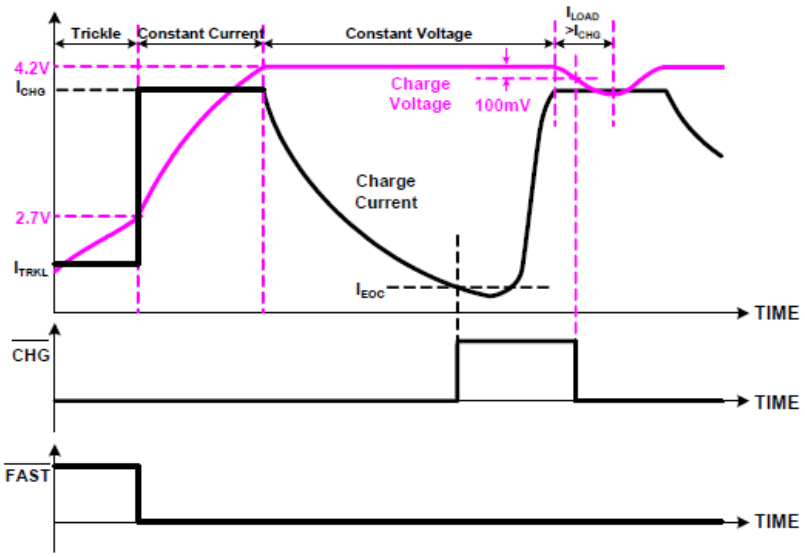



Figure 3.2 Charging profile for lithium cells

It is not uncommon for lithium polymer cells to enter runaway thermal conditions when overcharged or maintained at voltages above 4.2V and release internal pressures by venting gasses and possibly burst into flames. In consideration of this danger, it was determined that if lithium polymer cells would be used to power the dice, a decision which was made for the second generation onwards, an appropriate charge control circuit had to be added to the basic circuit design. The High Input Voltage 1.2A charger for single-cell lithium-ion batteries part number MC34673AEPR2 (Freescale Semiconductor, Austin, TX.) was selected for the charge control of this cell. Figure 3.3 shows a simplified application diagram from the datasheet.

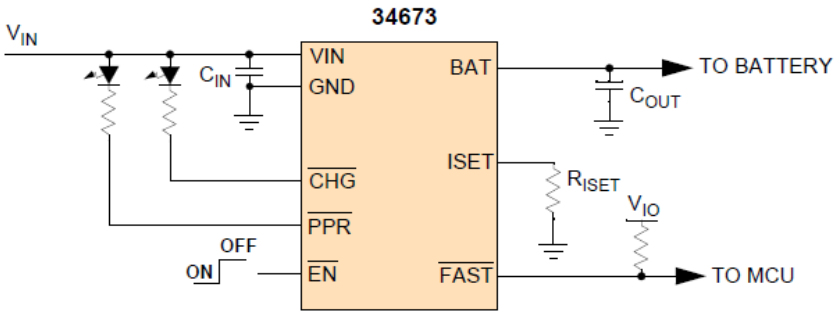



Figure 3.3 Simplified application diagram

The circuit was modified to eliminate the use of charging and “power present” indicator LEDs since the normal operation of the design would require 21 LEDs it was determined that additional LEDs

would cause confusion as they would be taking attention away from the device's design purpose. Likewise, pin 6 for fast charge indicator was eliminated from the design. Finally, the EN pin can be utilized to disable the charger by pulling to high voltage. Since this feature was not planned to be used, it was instead connected to ground, to make sure the charger would not be disabled. As per the datasheet the resistor $R_{ISET}$ which is used to set the charging current during the constant current mode of charging was determined using the listed formula of: $R_{ISET}= (4000/I_{CHG})-96$

### 3.1.3 Lithium polymer cell safety circuit

For similar considerations as discussed in the previous section, the use of a lithium polymer cell requires the use of a safety circuit to prevent the cell from being damaged and from causing damage to others due to conditions that would cause it to fail. For this function we selected the S-8241 series of lithium-ion/lithium-polymer rechargeable battery protection IC part number 628-8241-ABPM-G (Seiko Instruments Inc. Chiba, Chiba, Japan). These ICs are suitable for protection of 1-cell lithium-ion/lithium-polymer battery packs from overcharge, over-discharge and overcurrent. Figure 3.4 shows a connection example for the safety circuit. The basic operation of this IC is that it controls both the charge and discharge flows of current to and from the cell through FETs and is capable of interrupting either path if it detects any condition that could cause the cell to fail.

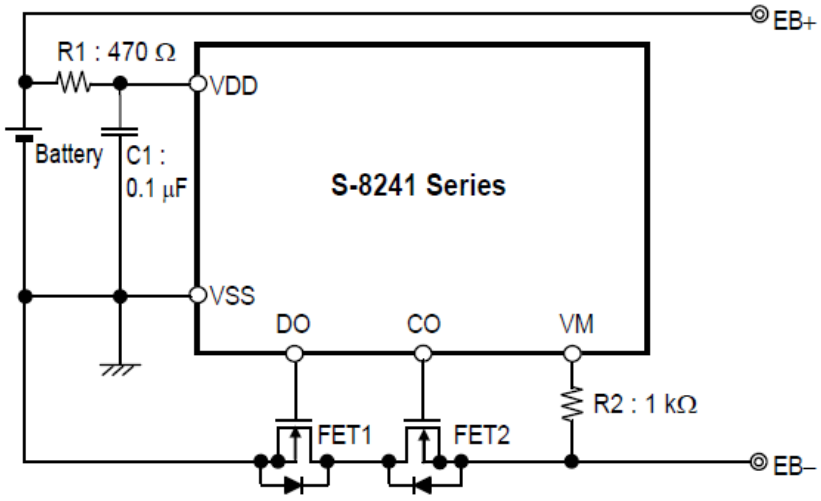



Figure 3.4 Connection example of safety circuit chip

### 3.1.4 Full protection circuit

The full protection circuitry for the lithium polymer cell utilized for the dice circuit must include both the charging and safety circuits. Figure 3.5 shows the final circuit as implemented.

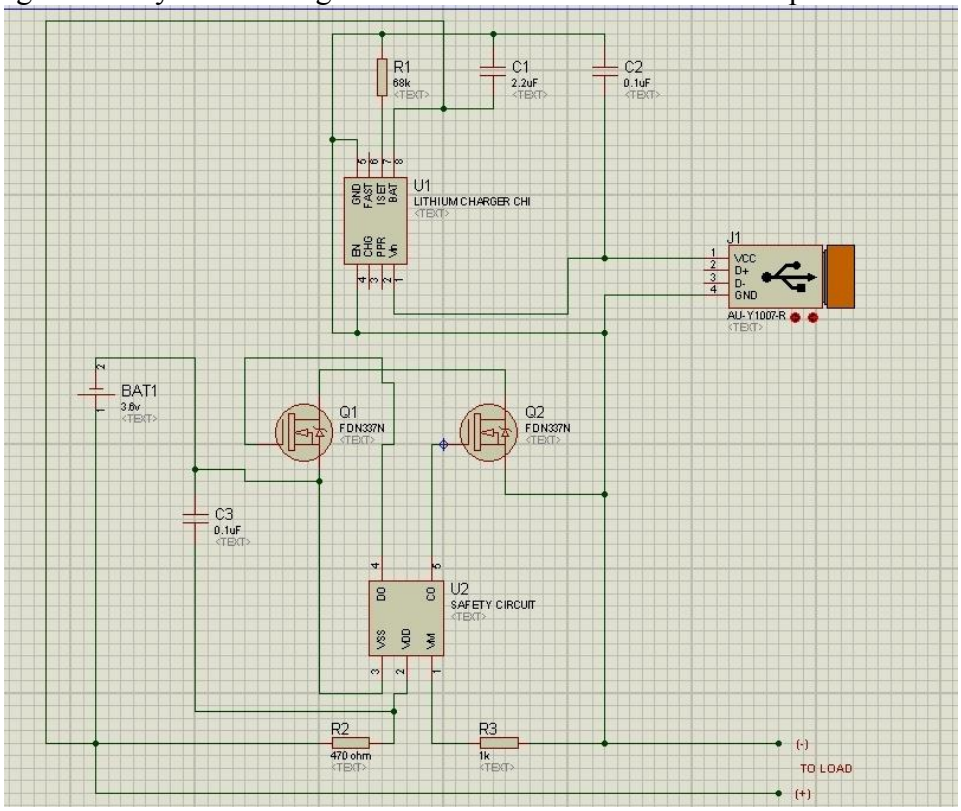



Figure 3.5 Safety and charging circuits implemented

### 3.1.5 Wireless charger system, battery module

For the second iteration of the dice it was determined the disposable battery power supply system would be replaced with a rechargeable power system. A single cell lithium polymer battery was selected for its size and adequate energy storage capacity. The cell part number GM041215 (Powerstream, Orem, UT.) has dimensions of 4mm X 12mm X 15mm and a capacity of 45mAh; these dimensions allowed us the capability to create a power module that would fit within the inner dimensions of the dice. The power module was designed as a 13mm diameter and 16mm tall cylinder with a cavity along its length where the lithium polymer cell would be housed. On the cylinder face, the required circuitry would be embedded.

In order to have a wireless charging system a resonant magnetic field would be used between an energy supply tank circuit located on a “battery charger” and an energy receiving tank circuit, housed on the outside surface of the cylinder. The circuit itself consists of an inductor and capacitor in parallel, implemented as a coil of magnet wire wrapped around the additive manufactured cylinder and a capacitor on its surface. The received power would be high frequency alternating voltage; this would be rectified using a rectifying diode bridge and a large filtering capacitor. The current would then be directed through the charging and safety circuits discussed previously into the lithium polymer battery.

Figure 3.6 shows the circuit designed and embedded into the lithium polymer module.

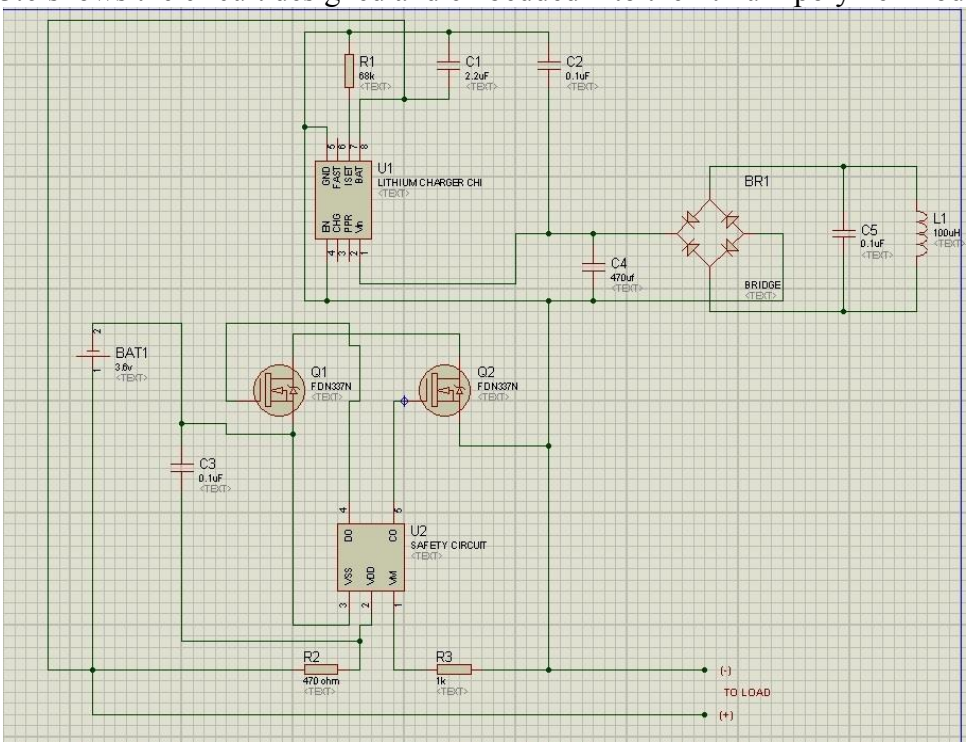



Figure 3.6 Wireless powered battery module circuit

Figure 3.7 shows two battery modules in different states of completion, on the left side a module that has the cavities and channels designed into the surface, where the components have been placed but no interconnects have been made; on the right, a fully completed battery module including the coil that is wrapped around the cylinder.

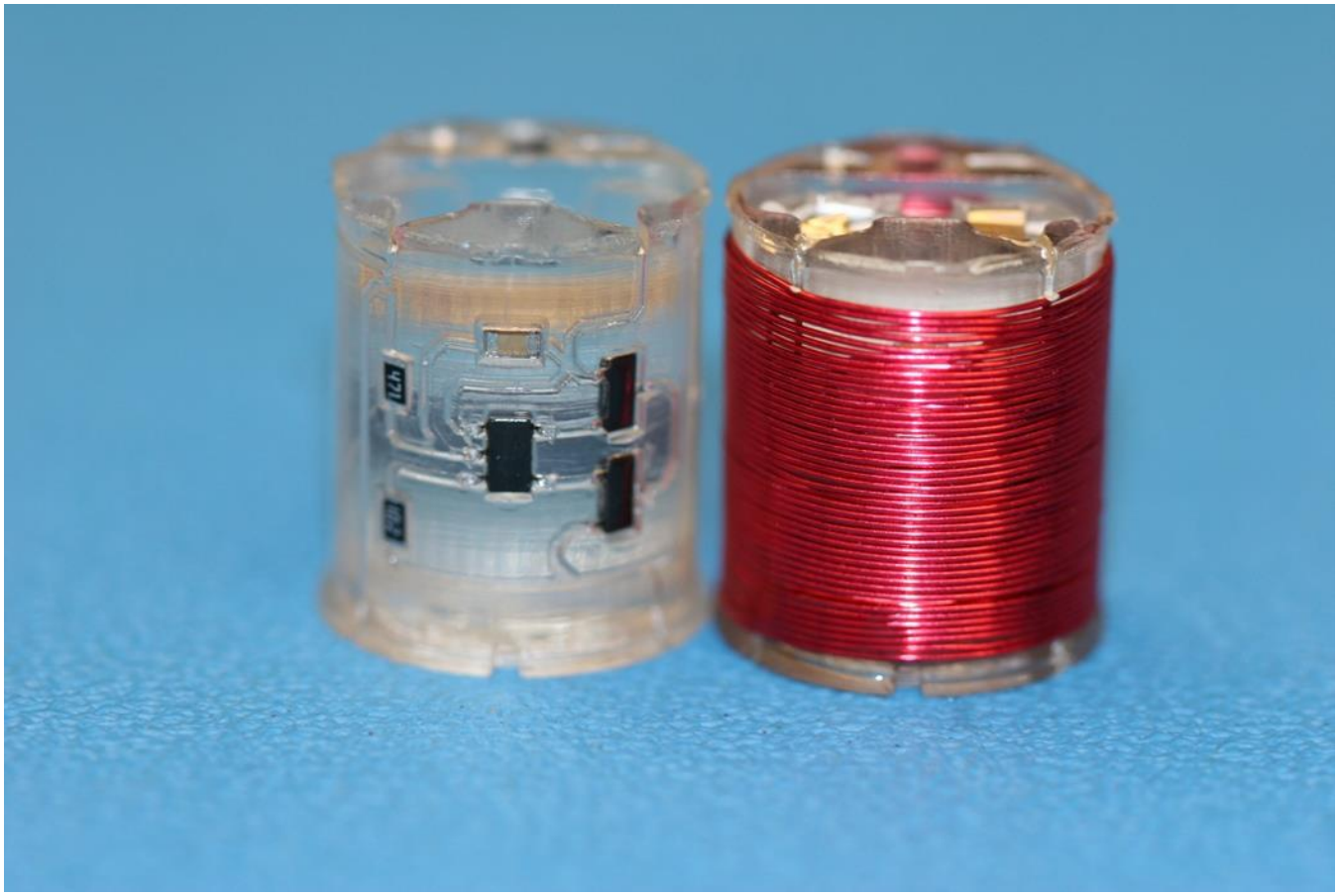


Figure 3.7 Two wireless powered battery modules partially and fully completed.

### 3.1.6 Wireless charger system, battery charger

In order to supply power to the battery module described above, a resonant magnetic field had to be generated at a frequency that matched the battery module tank circuit frequency. The easiest manner to accomplish this is to match the component values of the tank circuit located in the receiving circuit. Otherwise, the component values may be determined based on the available components and the resonant frequency that we need to match. The resonant frequency is obtained using the formula for resonant radian frequency:

$$\omega_0 = 1/\sqrt{(LC)}$$

And this can be solved for either component value that is required as:

$$L = \omega_0^2 C \text{ or } C = \omega_0^2 L \text{ as required}$$

The “charger” circuit is to be designed using the parameters that will cause it to generate a magnetic field that will resonate at the battery module frequency. For a magnetic field to be generated at that frequency an inductor and capacitor in series were used and power was provided in an alternating frequency that matched the requirements. To do this an H bridge was created using two P type FETs part number FDN340P (Fairchild Semiconductor, San Jose, CA) and two N type FETs part number FDN337N (Fairchild Semiconductor, San Jose, CA) to direct the flow of current in opposite directions through the inductor coil; using a PIC18F2520 micro-controller (Microchip Technology Inc., Chandler, Arizona) to turn on and off the pair of FETs that would accomplish this in at the appropriate frequency. Figure 3.8 shows the completed circuit.

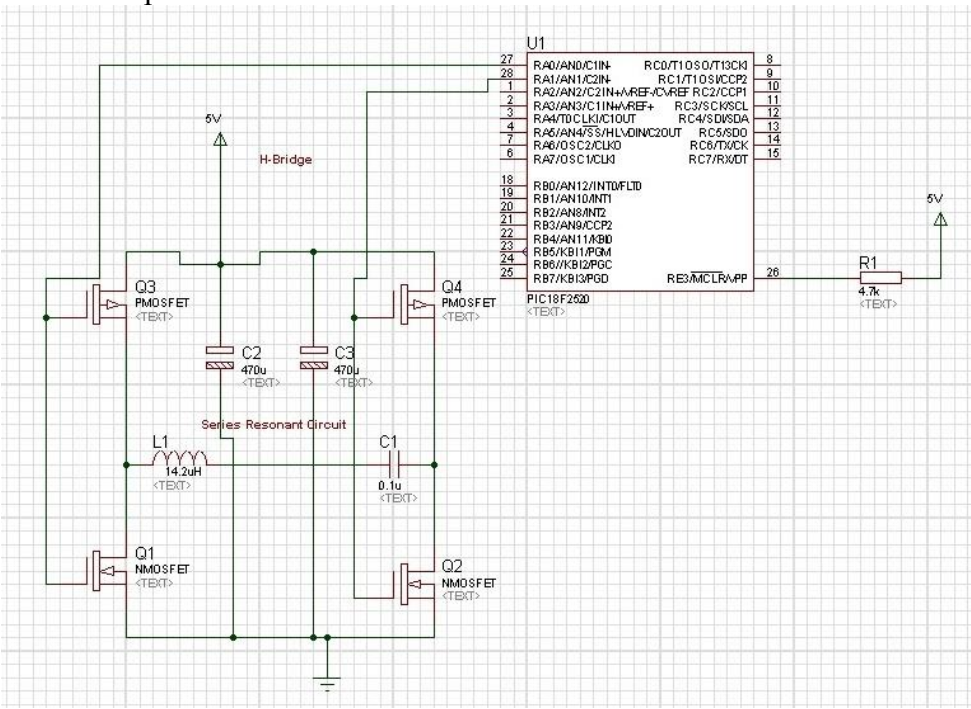



Figure 3.8 Circuit for the resonant field generator

The signals from the micro-controller alternate between port RA0=0 and RA1=1 or RA0=1 and RA1=0 to change the flow of current. Since we have a P and an N type FET on each side, a 0 turns one on and one off; while on the opposite side, a 1 turns one off and one on directing the flow of current in one direction. When these switch, the direction of current flow switches in the opposite direction generating an alternating current at the appropriate frequency. Figure 3.9 shows a completed charger

station with two, version 2 dice. The coil shown in the center generates the resonant magnetic field that wirelessly powers the dice batteries.

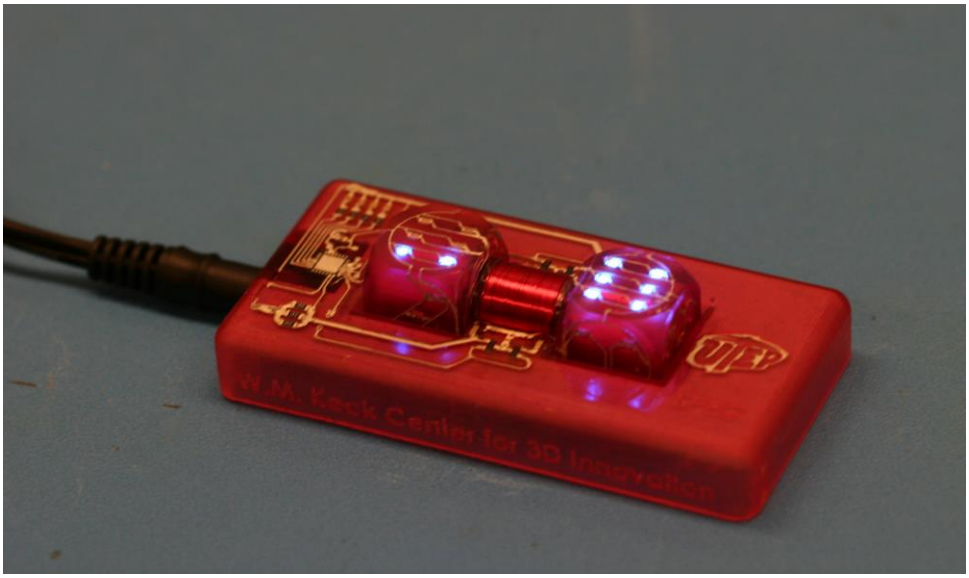


Figure 3.9 Completed charger station with dice.

### 3.1.7 Full dice circuit versions 1-3

The dice circuit minus the power supply is very similar to the schematic shown in section 3.1.1 except each component is shown with proper interconnects. The only major change this circuit has undergone in the first three iterations is that version 1 utilized a PIC18F2520 and versions 2 and 3 used a PIC18LF14K22; both are 8 bit micro-controllers from Microchip (Microchip Technology Inc., Chandler, Arizona). Additionally, version 1 circuits used a port to control a single LED; this allowed the freedom to make an involved light-up sequence each time a group of LEDs ended in the top face. The additional work, given the creation of these prototypes was a largely manual process, was deemed superfluous and was scaled down on subsequent generations. By version 3, each port of the micro-controller had the load of a maximum of 3 LEDs, which was still sufficient to create a back-and-forth effect when the numbers 4, 5 or 6 ended on top. Figure 3.10 shows the full circuit of the dice.

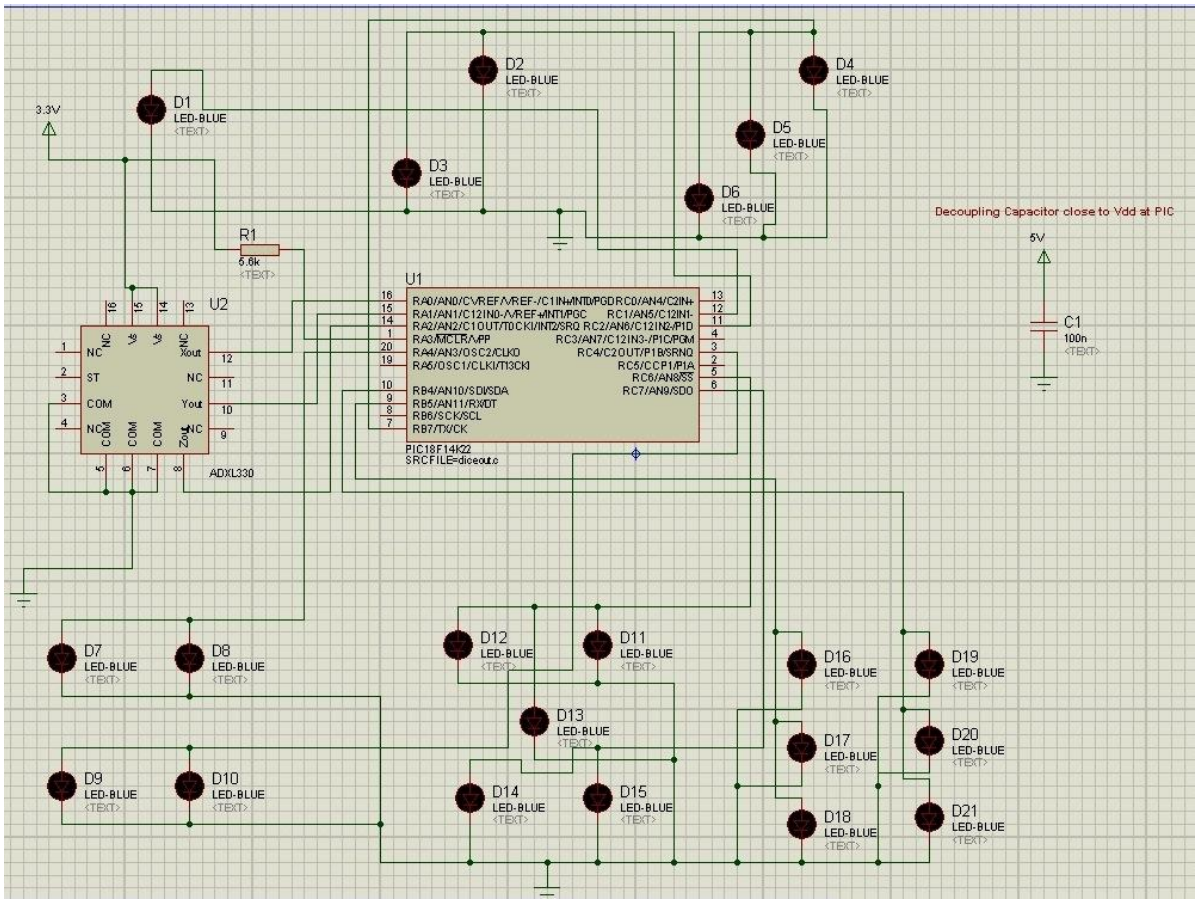



Figure 3.10 Full dice circuit

### 3.1.8 Manual Routing

Whereas software packages exist to perform automated routing and layout functions, these only work with the implicit understanding of a flat surface where the circuit will be placed. Even if the software allows for the possibility of two layers of components, assuming the circuit will have components on both sides of a Printed Circuit Board (PCB); the system does not take into account the possibility of fully three-dimensional component layout. For the dice design, we had a cube that would have components and interconnecting traces on each of its six outside and six inside surfaces. While each of its surfaces could be treated individually and routed that way, the best way to perform the routing was to consider the possible interconnects that could be accomplished between each of the flat surfaces, since these would simplify the final circuit by providing alternate paths for interconnections.

For this reason, the layout and routing of interconnects for the dice prototypes made through additive manufacturing was done manually, without software support and subsequently transferred this routing design to SolidWorks and mechanically built utilizing the Additive Manufacturing process. Figure 3.11 shows the layout and routing done for the first version of the dice, with the top row showing the outside surface of the dice half and the bottom row showing the inside surface.

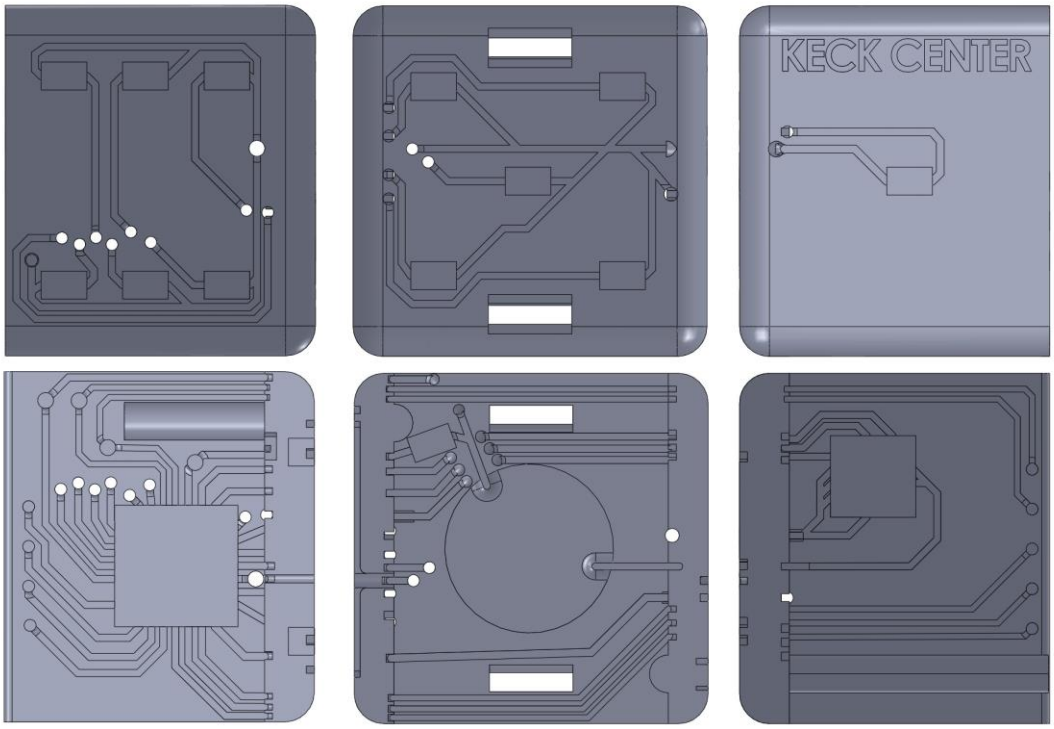



Figure 3.11 Routing of dice interconnects designed manually

The part shown in figure 3.11 is only the top half of the dice, the assembly of the two parts had to be laid out and routing had to include both pieces, since the same power source, micro-controller and accelerometer had to be interconnected to both parts. Figure 3.12 shows the full assembly of the first version of the dice design with a center cylindrical cavity for two LR-44 zinc-air or alkaline non-rechargeable batteries.

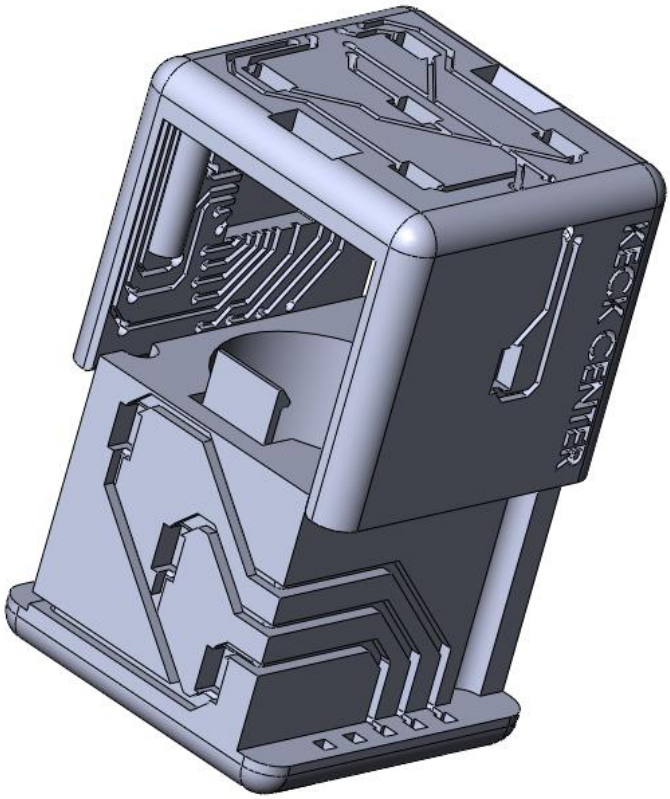



Figure 3.12 Full assembly of first generation dice

### 3.1.9 Automated Routing and Layout using Proteus

When the three dimensional surface to be utilized consists of an area that can be represented by a flat area wrapped around a single axis creating a cylinder or curved surface, the software package Proteus may be utilized. The dice design for the third generation dice calls for a cylinder shaped battery pack inside; the cylinder will have a cavity in the center for the lithium polymer cell and with both safety and charging circuits wrapped around the curve of the cylinder; for this case automatic routing software maybe used. Figure 3.13 shows the layout of components on a flat rectangular area where the software can perform automated routing and the result can be used to create a circuit around the curve of a cylinder: the battery pack for the dice. The yellow border is designated as the “board edge” for the layout and routing to take place within that area. This board edge area represents the outside surface of the cylinder we wish to create; the design need only be wrapped around that cylinder.

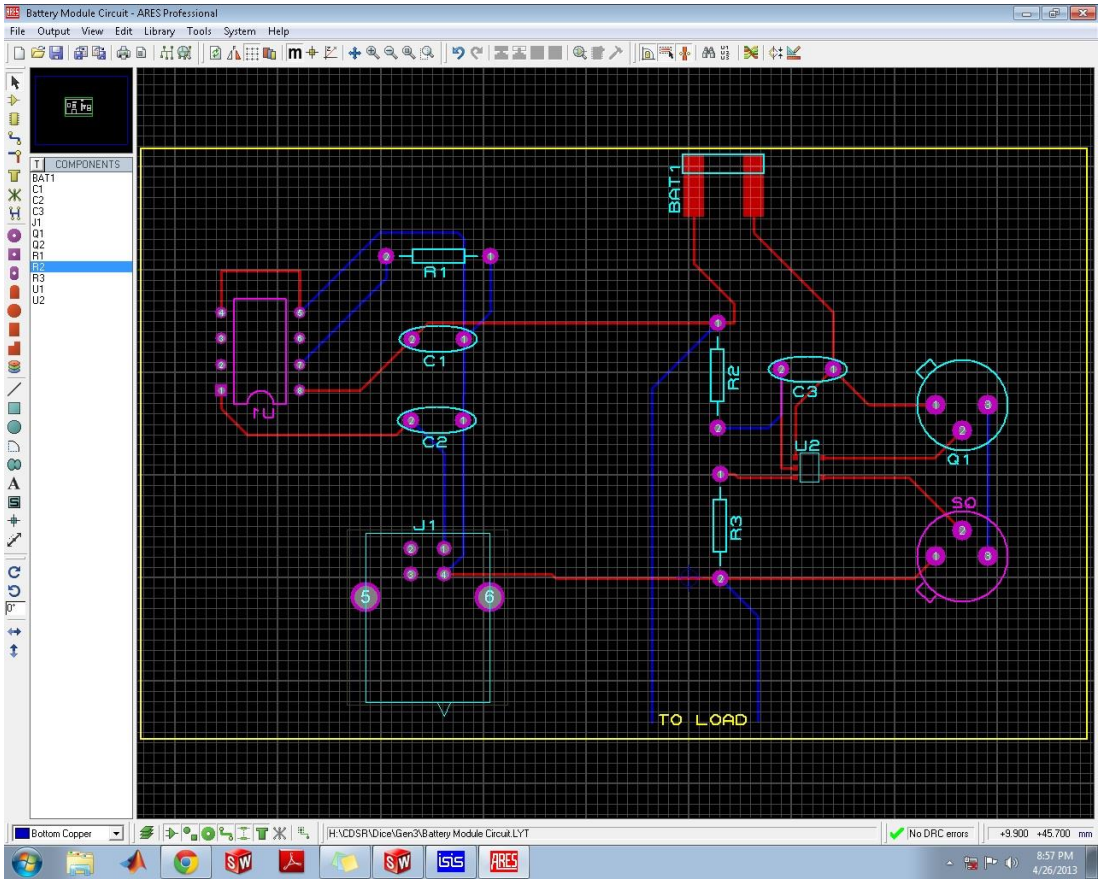



Figure 3.13 A flat area is used to simulate an unwrapped cylinder

The routing can be used to design the layout of the circuit into the mechanical design of the battery module. The figure number 3.14 shows the layout when applied to a cylinder, the location of each component and the routing of the channels to create interconnects, is nearly identical to the automated routing obtained from the software. The only required change is due to the ability to draw a trace from the right side and go in the right direction to connect on the left side of the flat area or from the left side going in the left direction.

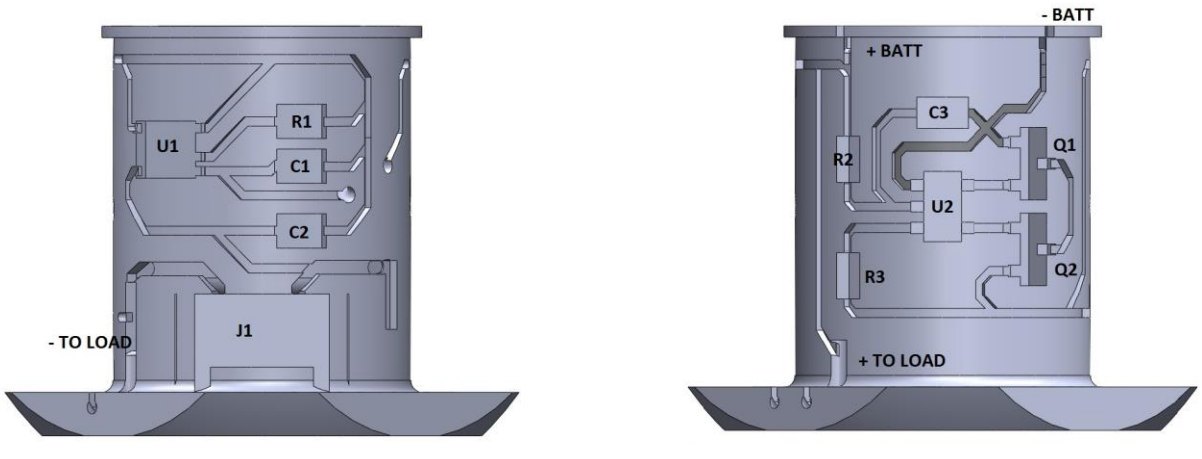



Figure 3.14 The flat routing is designed into a cylinder

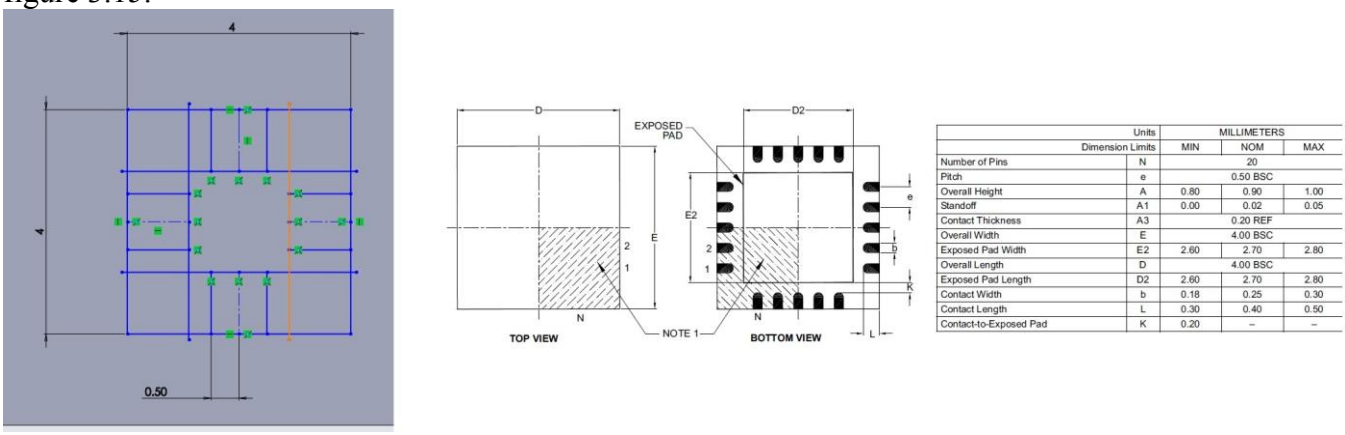
## 3.2 MECHANICAL DESIGN

The mechanical design of structural electronics is done using the SolidWorks CAD design software (3D Systems, Waltham, MA.) The design includes the design of the structure, the object where the electronics are to be included and the electronic circuit itself. After the electronic circuit is designed, it must be laid out over the available surfaces and integrated into the design of the part itself. Through this methodology, the structure of the part becomes the dielectric substrate for the circuit itself and there's no need for a Printed Circuit Board, as the circuit is printed directly onto the part. This process allows better utilization of the available volume and a much better integration of the electronics into the device itself. For this project, the ability to print electronic circuits onto additive manufactured parts, allows us the opportunity to develop accurate, three-dimensional prototypes that include functional electronic circuits.

### 3.2.1 SolidWorks Challenges

As previously discussed, the electronic circuit that is to be embedded into a structure is designed first, then it is laid out with or without the aid of a computer program. Then the circuit must be mechanically designed into the part itself. The first step is to perform the layout of the components, to assign the location where each electronic component must be located. The first consideration must be the circuit's own requirements. For instance, for the dice design, we had to place an LED in the position

occupied normally by a dot or ‘pip’ of a normal die. These LEDs are therefore laid out on each of the six faces in their pre-determined positions. The next consideration is the ease with which the electronic interconnects can be made to the components, just like any other electronic circuit. Once the layout is made, the dimensions of each component is obtained from its datasheet and a cavity is designed in the surface of the part to allow the component to be embedded, leaving the part flush with the part’s surface. In order to do this, the surface where the component is to be embedded is selected as the base plane for the sketch and the sketch is then drawn, based on the dimensions of the part to embed. As shown in figure 3.15.




| | Units | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Number of Pins | N | | 20 | |
| Pitch | e | | 0.50 BSC | |
| Overall Height | A | 0.80 | 0.90 | 1.00 |
| Standoff | A1 | 0.00 | 0.02 | 0.05 |
| Contact Thickness | A3 | | 0.20 REF | |
| Overall Width | E | | 4.00 BSC | |
| Exposed Pad Width | E2 | 2.60 | 2.70 | 2.80 |
| Overall Length | D | | 4.00 BSC | |
| Exposed Pad Length | D2 | 2.60 | 2.70 | 2.80 |
| Contact Width | b | 0.18 | 0.25 | 0.30 |
| Contact Length | L | 0.30 | 0.40 | 0.50 |
| Contact-to-Exposed Pad | K | 0.20 | – | – |

Figure 3.15 Sketch is made with dimensions of part to embed

The next step is for the circuit interconnects to be designed into the part as well. Using the sketch of each component as a guide, a line is drawn indicating the interconnect path as dictated by the circuit design. The offset command is then used to create two lines on either side of each interconnect path. The option to “make base construction” is selected to make sure the center line is no longer active as it would interfere with the function of the next command to use. The offset distance to use is half the required channel width. Next the Feature command “Extrude cut” is used to cut a channel of the drawn channel profile into the part surface. The depth is typically the same as the width selected. Figure 3.16 shows how the sketch of the pad location is used as a guideline to create the channels where interconnects will be made using silver loaded ink.

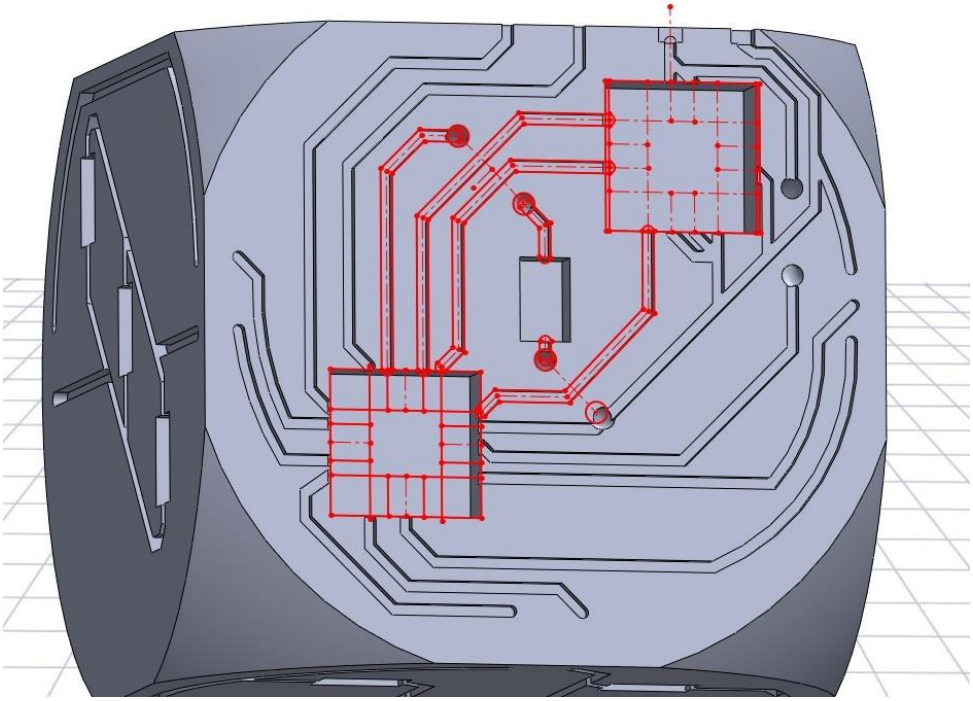


Figure 3.16 Channels are cut to provide a path for the silver ink to be deposited

In this manner, all interconnects are designed into the part as channels and then electrical interconnects are made utilizing silver loaded ink. In previous research at the University of Texas at El Paso, it was determined that Ercon 1660 (Ercon Inc., Wareham, MA) was the silver loaded ink that provided the best characteristics for conductivity, low temperature requirement for cure and low average resistivity. Because of this, this is the material that is dispensed into the channels to make electrical interconnects. [17, 29]

### 3.2.2 Design for Plastic Injection Molding

The 4$^{th}$ generation of the dice design was intended to be a commercially viable product; as a result more traditional manufacturing methods and materials were selected to create it. The substrate itself is not to be fabricated using additive manufacturing; instead plastic injection molding was selected to keep costs down. Plastic injection molding fabrication has specific requirements that limit the design freedom and these had to be addressed during the redesign process of the case in order to be able to create it using plastic injection molding process.

The first process specific issue to address was that independent cavities for each of the components in contact with the mold could not be built without the use of inserts, which would increase the cost of the mold to create the molded part. An insert is an additional, moving part of a mold that requires moving in and out of position during the plastic injection process. Instead a single cavity would be designed for all LEDs in the mold direction, such a single opening allows for the mold feature that creates it to move out of contact with the part through the channel. Figure 3.17 shows the channels cut on the inside of the case to accommodate the LEDs that were designed to be achievable through the mold process.

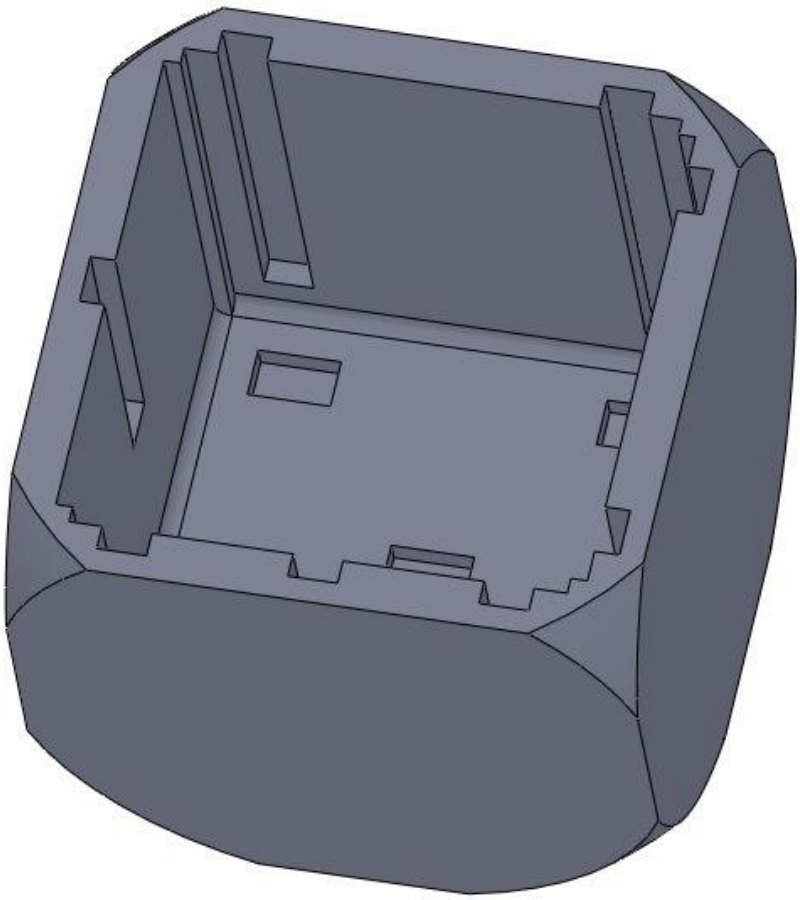


Figure 3.17 Design for viable plastic mold channels

Another mechanical design requirement unique to plastic injection molding processes is the need for a draft angle on any feature that is designed to be parallel to the mold movement. A perfect cube

could be designed and a mold can be fabricated to create it; however, when the plastic solidifies and the mold halves move to separate, friction will make this action more difficult and contact with the mold as it moves will scrape the part's surface. When performing mechanical design for plastic injection molding, SolidWorks provides specific "mold" tools that allow analysis methods like "draft analysis" and features to correct them like "draft" tools. Figure 3.18 shows the use of the "draft analysis" tool where the faces that require the addition of a draft angle are marked in yellow.

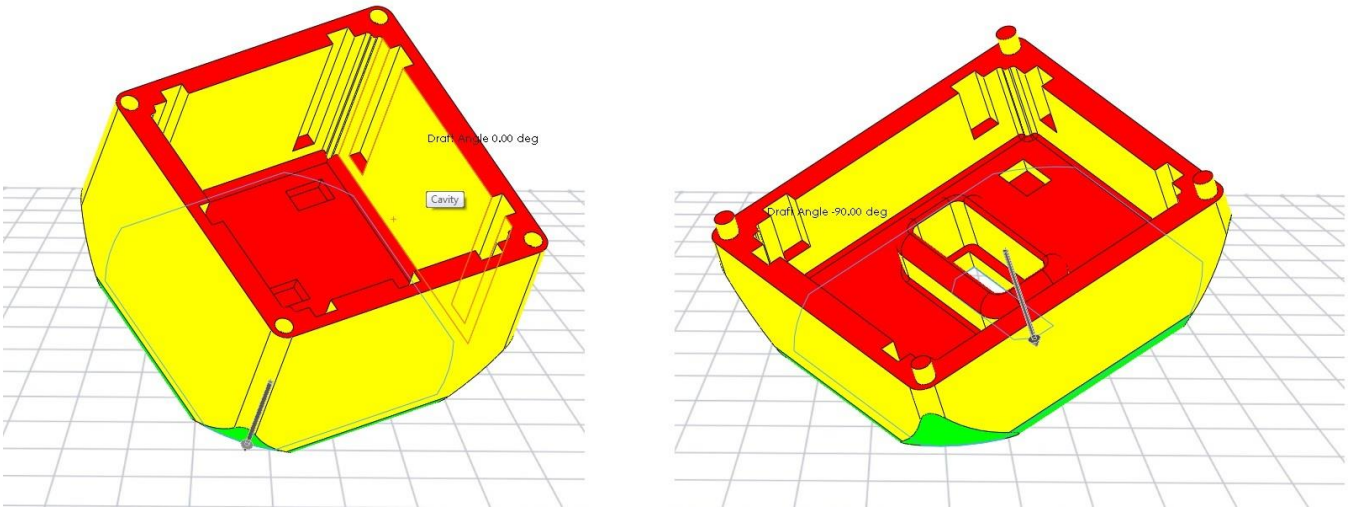



Figure 3.18 Draft analysis tool indicates, in yellow, faces that require draft angle

### 3.3 MICRO-CONTROLLER PROGRAM

The brain of this smart dice device is the micro-controller, which receives the acceleration information from the 3-axis accelerometer and uses it to determine its orientation. The micro-controller then uses its orientation to determine which side is pointed up in order to flash the LEDs on that side.

#### 3.3.1 Basic Algorithm

The micro-controller continuously receives signals from the 3-axis accelerometer; these signals are received either through 3 independent channels with positive or negative orientation to cover each of the six possible sides of the dice; or through serial communication where a six bit message indicates the inclination of the dice in a particular axis. Either of these two methods corresponds to either of the two accelerometers used for this project. The micro-controller used receives the signal from the accelerometer and interprets it to determine when the dice is moving. When the micro-controller detects

the dice has stopped moving, it uses the information received from the accelerometer to determine orientation and sends a signal to flash the LEDs on that face. After the LED light up sequence is completed, the micro-controller enters a power saving sleep mode. During sleep mode the circuit retains the capability to detect movement in order to detect when a shake sequence has started in order to perform its main function once more.

### 3.3.2 Analog to Digital Conversion method

The ADXL330KCPZ (Analog Devices, Norwood, Massachusetts) is the 3-axis accelerometer used for the first three iterations of the smart dice project. This accelerometer has three output ports, each of which has a neutral voltage value of approximately 1.5V when there is no acceleration detected in that axis. When acceleration is detected the accelerometer outputs a 1.2V or lower or a 1.9V or higher signal. When the dice is in a rest position in any of its six sides, two ports will have the neutral 1.5V value and the port that corresponds to the axis upon which the dice is resting will have either a 1.2 or lower or 1.9V or higher signal. Figure 3.19 shows the accelerometer indicating the 6 possible orientations it can detect when placed on the dice substrate on a flat plane orthogonal to the other 2 possible orientations.

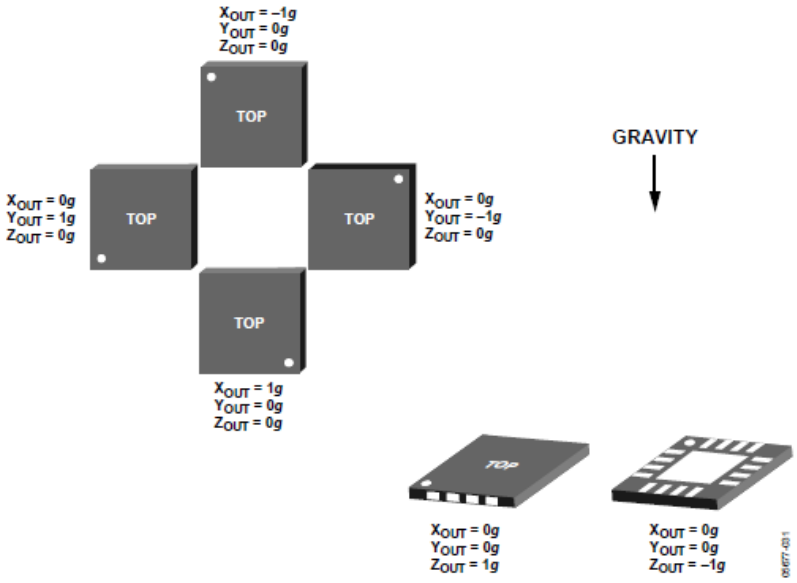



Figure 3.19 The 3 axis accelerometer ADXL330 and the 6 possible orientations it can assume

The micro-controller receives the voltage value, uses the internal Analog to Digital Conversion capability to convert those voltage values into 10 bit numbers which are then manipulated to determine when motion has stopped and which side is on top and flashes the LEDs on that side. The micro-controller used in conjunction with ADXL330KCPZ (Analog Devices, Norwood, Massachusetts) accelerometer is the PIC18LF14K22 (Microchip Technology Inc., Chandler, Arizona). This 8 bit micro-controller has 12 ADC capable channels; 12 of the I/O pins can be used to receive and convert an analog voltage signal into a 10 bit value, which can be mathematically manipulated to determine the voltage value detected at the I/O due to the precision obtained from having 2^10 (1024) segments between the pre-set voltage references. As previously described, the accelerometer will provide a 1.5V voltage signal on two of the ports and a voltage lower than 1.2 or higher than 1.9 on the axis where the die is resting. Using this information the micro-controller determines which LEDs it must flash to indicate the top.

### 3.3.3 Using Microchip Micro-Controller with MPLab IDE

The PIC18LF14K22 micro-controller is programmed using proprietary software and hardware system consisting of the MPLab ICD2 programmer (Microchip Technology Inc., Chandler, Arizona) which interfaces with MPLab Integrated Development Environment (IDE). The MPLab IDE has C language support capability which allows the micro-controller program to be written in C language; MPLab IDE then converts the program into .hex code and through the MPLab ICD2 programmer is loaded into the micro-controller. Figure 3.20 shows the MPLab IDE screen, showing where the program is written in C language and the menus and buttons which allow the programming of the micro-controller when connected to MPLab ICD2 programmer and through it, to the micro-controller.

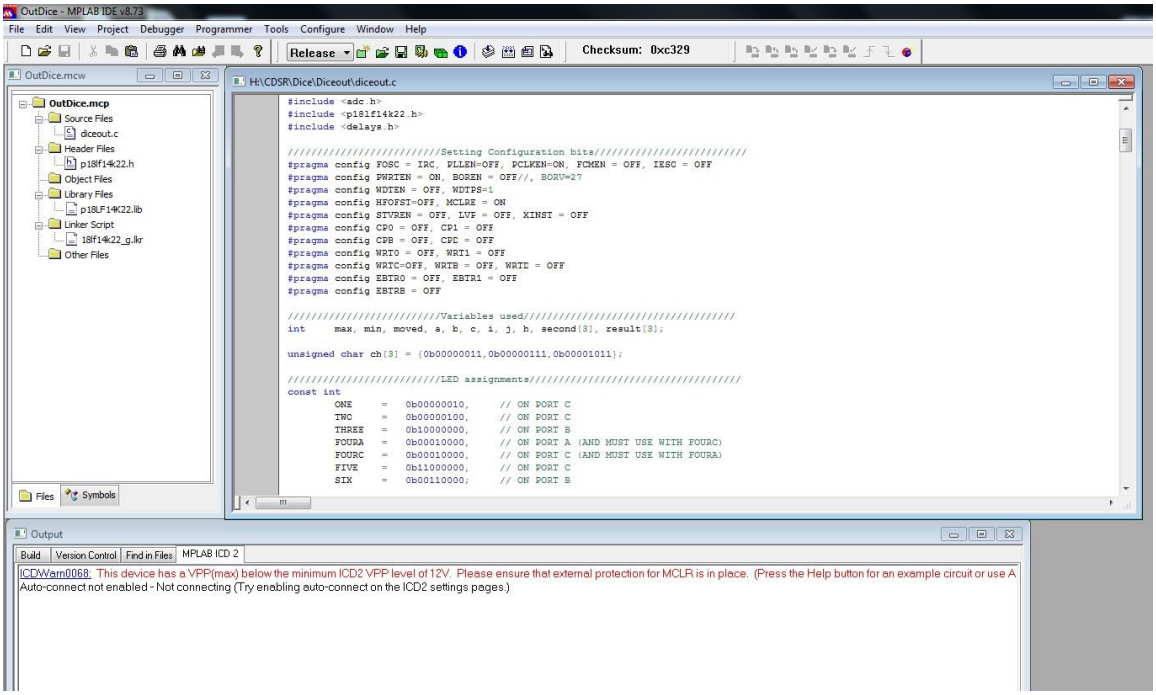



Figure 3.20 MPLab IDE for programming PIC micro-controllers

The MPLab ICD2 programmer has 5 connections that must be made to the micro-controller and the programmer being utilized must be selected in the IDE for proper operation. Figure 3.21 shows the datasheet diagram indicating the proper connection method for connecting the IDE to the micro-controller pins.

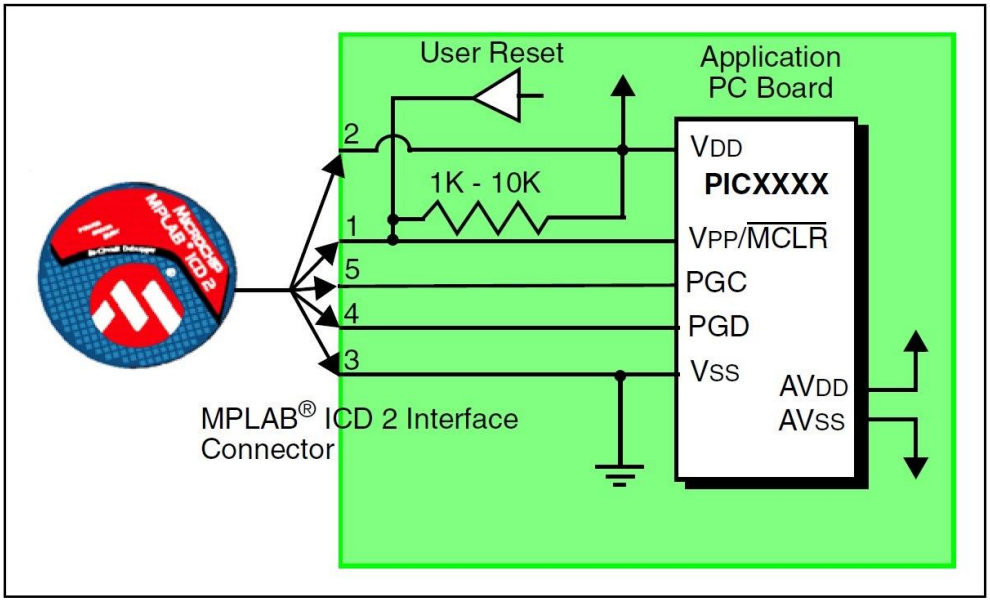



Figure 3.21 Connection diagram for using MPLab ICD 2

### 3.3.4 I2C Serial Communication method

The accelerometer used for the 4th generation dice is the 3-axis orientation/motion detection sensor part number MMA7660FC (Freescale Semiconductor, Austin, TX.) which in addition to the ability to detect orientation in 3 axis, has the ability to communicate this information through serial communication. This capability makes the work for the micro-controller easier, since no Analog to Digital conversion has to take place and no mathematical manipulation is required to determine orientation. The MMA7660FC accelerometer transmits a 6 bit value to indicate the inclination degree in each of the axes and a range of acceleration that it's detecting. A simple "window" condition is required to make the decision on which LEDs to flash. Figure 3.22 shows the type of serial data the accelerometer transmits, showing how a simple "window" condition in the micro-controller code will be used to determine orientation.

| 6-bit result | Binary | 2's Comp | g value | Angle X or Y | | Angle Z |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0.000g | 0.00° | | 90.00° |
| 1 | 1 | 1 | 0.047g | 2.69° | | 87.31° |
| 2 | 10 | 2 | 0.094g | 5.38° | | 84.62° |
| 3 | 11 | 3 | 0.141g | 8.08° | | 81.92° |
| 4 | 100 | 4 | 0.188g | 10.81° | | 79.19° |
| 5 | 101 | 5 | 0.234g | 13.55° | Z-axis must be in the range | 76.45° |
| 6 | 110 | 6 | 0.281g | 16.33° | | 73.67° |
| 7 | 111 | 7 | 0.328g | 19.16° | | 70.84° |
| 8 | 1000 | 8 | 0.375g | 22.02° | | 67.98° |
| 9 | 1001 | 9 | 0.422g | 24.95° | | 65.05° |
| 10 | 1010 | 10 | 0.469g | 27.95° | | 62.05° |
| 11 | 1011 | 11 | 0.516g | 31.04° | | 58.96° |
| 12 | 1100 | 12 | 0.563g | 34.23° | | 55.77° |
| 13 | 1101 | 13 | 0.609g | 37.54° | | 52.46° |
| 14 | 1110 | 14 | 0.656g | 41.01° | | 48.99° |
| 15 | 1111 | 15 | 0.703g | 44.68° | | 45.32° |
| 16 | 10000 | 16 | 0.750g | 48.59° | | 41.41° |
| 17 | 10001 | 17 | 0.797g | 52.83° | | 37.17° |
| 18 | 10010 | 18 | 0.844g | 57.54° | | 32.46° |
| 19 | 10011 | 19 | 0.891g | 62.95° | | 27.05° |
| 20 | 10100 | 20 | 0.938g | 69.64° | | 20.36° |
| 21 | 10101 | 21 | 0.984g | 79.86° | | 10.14° |

Figure 3.22 Samples of 6 bit result and interpretation

The micro-controller used with the MMA7660FC (Freescale Semiconductor, Austin, TX.) accelerometer is the ATTINY20-XU (Atmel Corporation, San Jose, CA.) which was selected in large

part due to its reduced price and adequate capability. Unfortunately while this micro-controller does have I2C capability, it only has the capability to act as a "slave" to receive serial messages, but not to take control of the serial bus to perform the functions of "master". Given this limitation a written function was necessary for this micro-controller to simulate I2C master capabilities, known as "bit-bang". Through the use of this function, the micro-controller is capable of communicating through serial communication with the 3-axis accelerometer as a "slave" in order to configure it and to draw orientation information from it. The micro-controller configures the accelerometer to take sample measurements at a selected interval and to make multiple readings to determine when the dice has stopped. When the dice has stopped, the accelerometer transmits a 6 bit number to indicate inclination. The micro-controller uses a simple "if" statement with a value window to determine the orientation and flash the LEDs on the top surface. The code that determines orientation based on a window condition is shown below:

```
if (X>17 && X<28)   //X positive
if (X>37 && X<47)   //X negative
if (Y>17 && Y<28)   //Y positive
if (Y>37 && Y<47)   //Y negative
if (Z>17 && Z<28)   //Z positive
if (Z>37 && Z<47)   //Z negative
```

Next, the micro-controller places the accelerometer and itself in power saving sleep mode. The accelerometer is configured to, upon detection of movement, return to active mode and send an interrupt signal to the micro-controller, rousing it from sleep mode as well, making them both ready to perform their functions.

### 3.3.5 Using Atmel Micro-Controller with Studio 6 IDE

The ATtiny20 micro-controller used for the 4$^{th}$ generation prototype of the dice requires the use of an exclusive programmer, the AVR ISP mkII and Integrated Development Environment (IDE) AVR Studio 6 for Atmel devices (Atmel, San Jose, CA.) Additionally since the ATtiny20 micro-controller does not have the ability to perform the functions of master in I2C serial communications protocol, a function to allow this added capability was obtained; unfortunately this function was written within a separate IDE, the IAR Embedded workbench, Kickstarter edition (IAR systems, Uppsala, Sweden). As a result, the dice program for the 4$^{th}$ generation of the dice was written within the IAR Embedded workbench IDE to take advantage of the bit-bang function for the ATtiny 20 micro-controller, including

the generation of the .hex file and the Studio 6 IDE used to program this “makefile” file into the ATtiny 20 micro-controller by utilizing the “Use External MakeFile” option within the Studio 6 IDE. Figure 3.23 shows the IAR Embedded workbench IDE where the code was created.

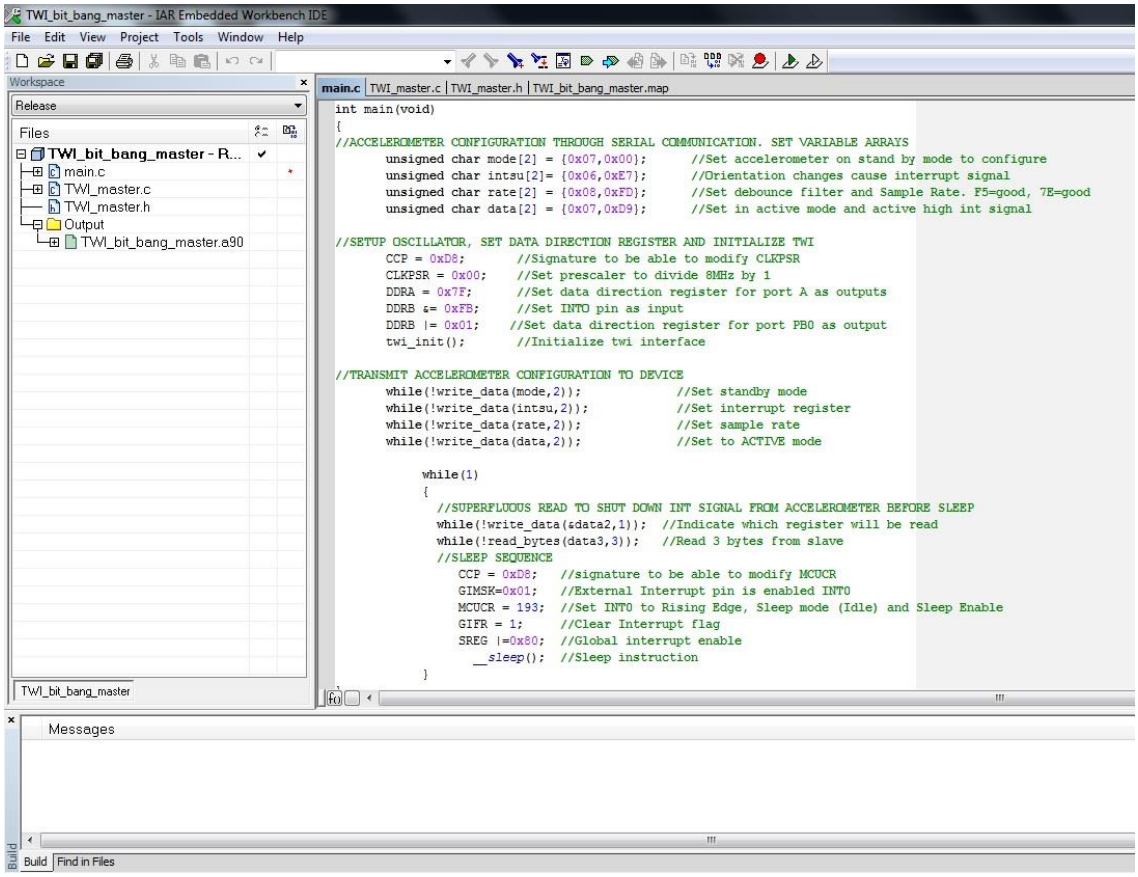



Figure 3.23 IAR Embedded workbench IDE used to develop C code

Figure 3.24 shows the Studio 6 IDE where the “Use External Makefile” was used to program the ATtiny 20 micro-controller using the AVR ISP mkII programmer.

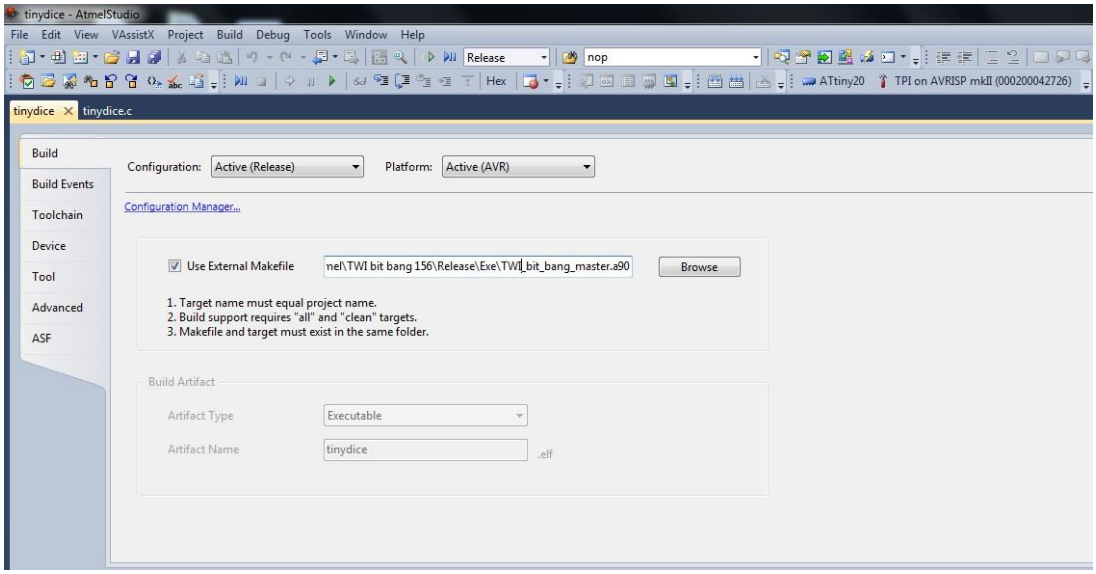



Figure 3.24 AVR Studio 6 screen shot, as was used to program the micro-controller

## 3.4 TRADITIONAL FABRICATION

In order to create a device that could be commercially viable traditional fabrication methodologies were selected to create the 4th generation of the dice. These procedures required adjustments to the build strategy pursued up to that point. The prototype build process had to be revisited to ensure choices made would be appropriate for these new methodologies. The first change was to revisit the selection of electronic components for the dice by selecting components of lower cost, higher availability and adequate functionality.

### 3.4.1 Flexible Circuit Fabrication

The circuit itself was for the first 3 generations made through Direct Write micro-dispensing of silver loaded ink onto a substrate made through Additive Manufacturing. For a commercially viable product, the reliability of a flexible circuit inserted into a hollow cube was more appropriate. Though the circuit design for this version is very similar the one shown in Figure 3.10 some changes were evident: A new set of components was used, a methodology to program the micro-controller after it was built was required and the fact it was planned for a flat surface allowed the use of software to perform automated layout and routing. Figure 3.25 shows the circuit design for the 4th generation dice.

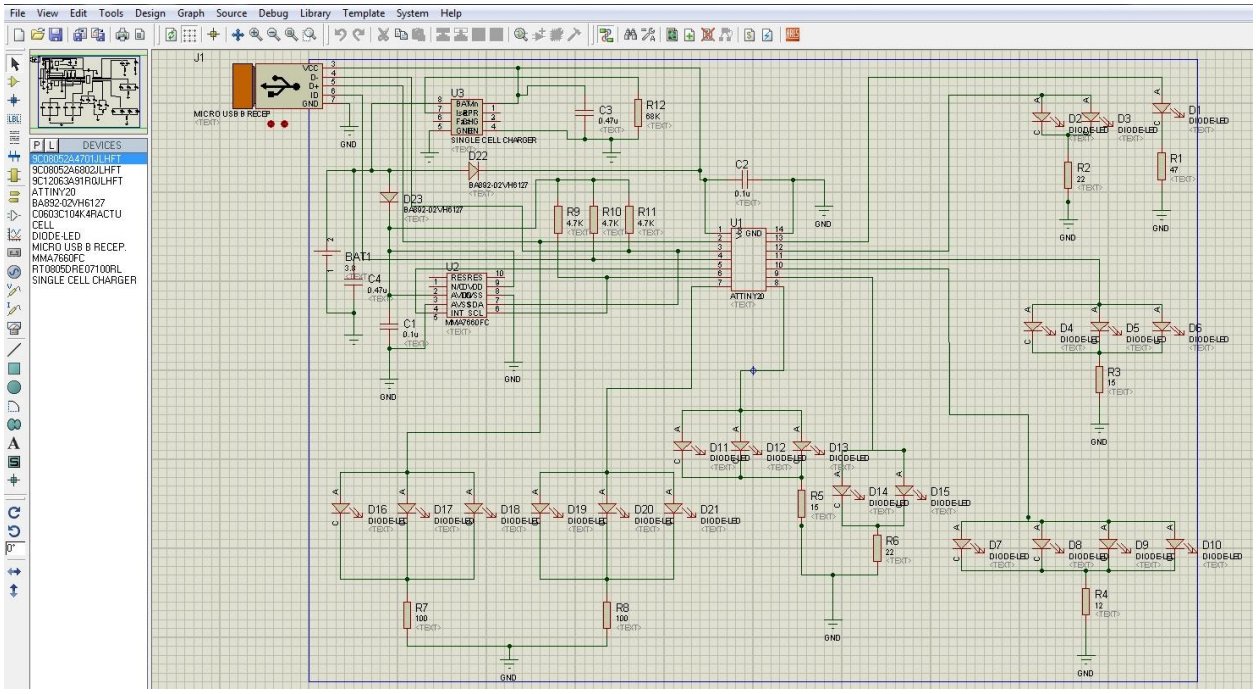



Figure 3.25 Circuit design for 4th generation dice

The design for this dice described the need for a hollow plastic cube where a flexible circuit could be folded to have the LEDs show through each face. To achieve this, the flexible circuit was designed to be an expanded cube, a “cross” shape where the LEDs would be on one side and the remaining components would be on the opposite side. Using the ARES portion of the Proteus software, this was accomplished and is shown in Figure 3.26.

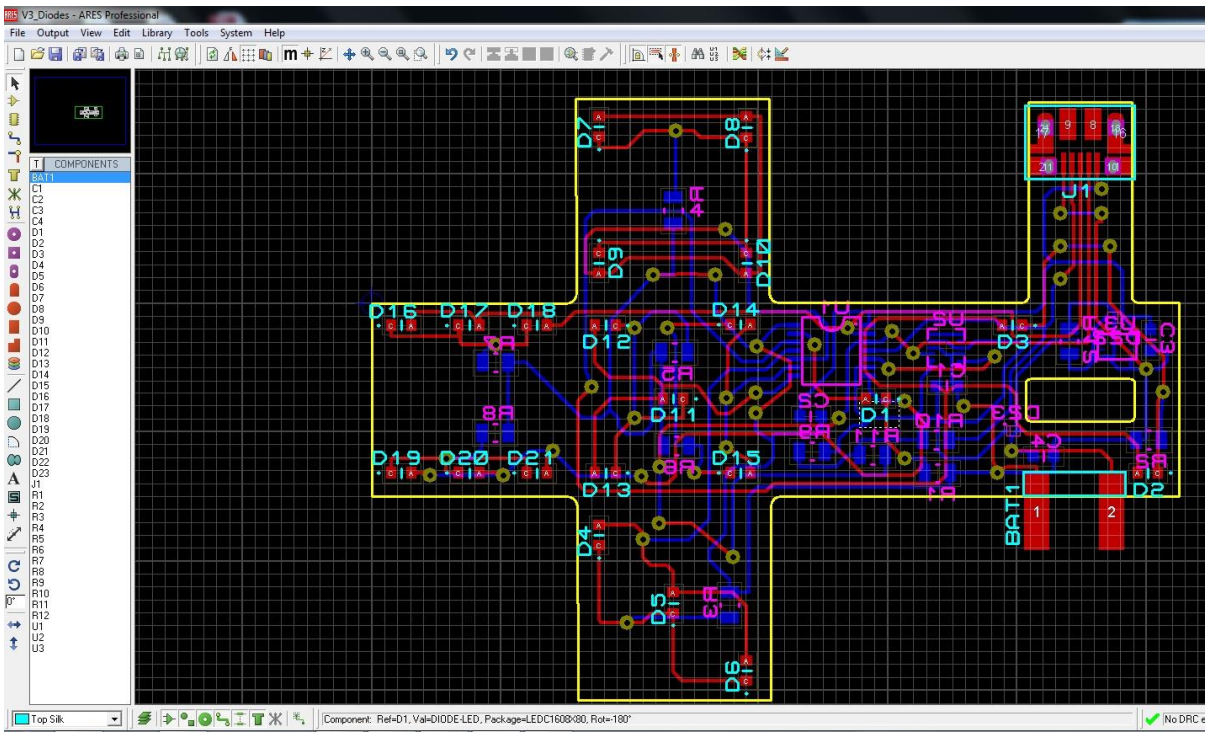



Figure 3.26 Automatic routing made using ARES software

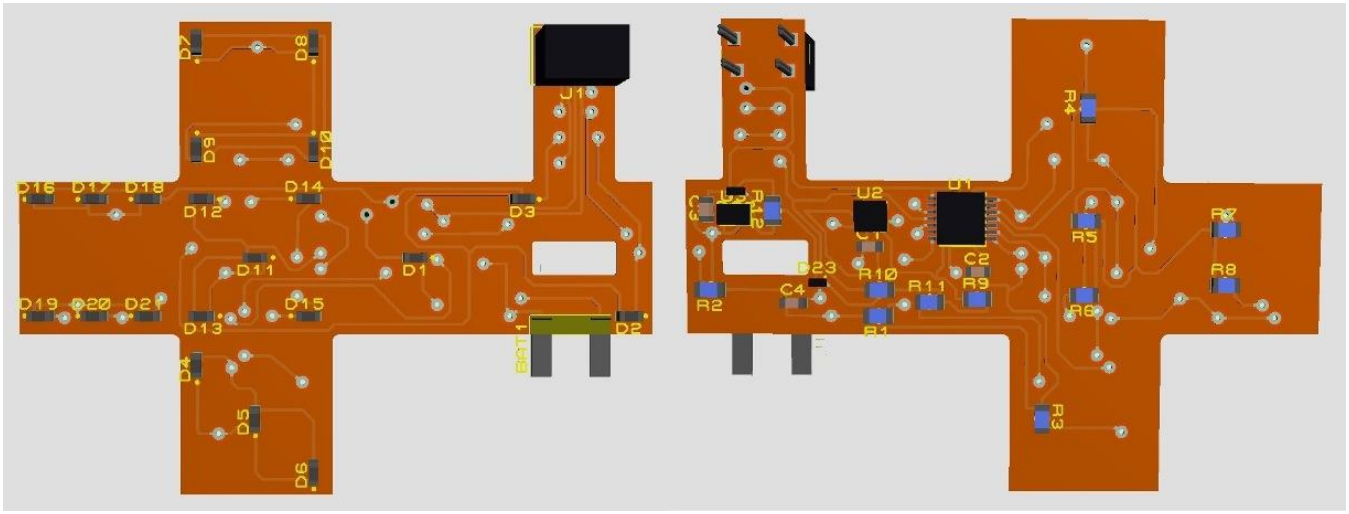



Figure 3.27 Visual approximation of both sides of flexible circuit

The layout shows the components were manually laid out on the “component side” in order to have the LEDs showing their proper position, simulating a regular dice and automated routing provided the interconnects for all components. Figure 3.27 shows a more realistic simulated view. In figure 3.27 two more features are evident: two nickel tabs are left outside the flexible circuit, where the lithium

polymer battery will be attached to the circuit and the poorly drawn USB-b connector is shown on a protruding tab above a cutout hole. The design calls for the flexible circuit to be folded into the hollow plastic cube and the protruding tab to be folded inwards for the USB-b connector to be accessible to a connecting cable through the cutout hole on the face where the 2 LEDs are.

### 3.4.2 Plastic Injection Molding Case Fabrication

For the reasons stated in the previous section, the dice case would be built separately from the circuit. The housing is designed to be a translucent cube of plastic with all flat surfaces and a hollow cavity where the flexible circuit could be folded into place such that the LEDs for each face would be lined up with each of the six surfaces and visible from the outside. The LEDs would not have individual cavities for each LED as the molding process required to accomplish this would require several inserts and these would increase the price dramatically. Instead a channel is to be created where the LEDs that are lined up in each face could be inserted and the metal feature on the mold could still be pulled out without the requirement of inserts on the mold. The process for plastic injection molding starts with two metal parts to create a mold such that when they join they form a cavity and the negative space of that cavity will form the part to be fabricated. The typical implementation requires the part to be made to be split along a line where the two mold halves will meet, this “parting line” must be considered in such a way that after it is completed, the physical features of the part still allow the mold to separate. This requirement means the whole of the features cannot be placed on one half of the mold pair, since the dice has curves on both ends, if that was attempted the part would not be able to exit the mold once formed. Using the SolidWorks tools known as “mold” tools, a design was completed that, when evaluated using “draft analysis” tool was deemed acceptable and completed. Figure 3.28 shows a realistic representation of the mold features that would form each of the mold halves to fabricate each of the “housing” and “cap” required to complete a dice.

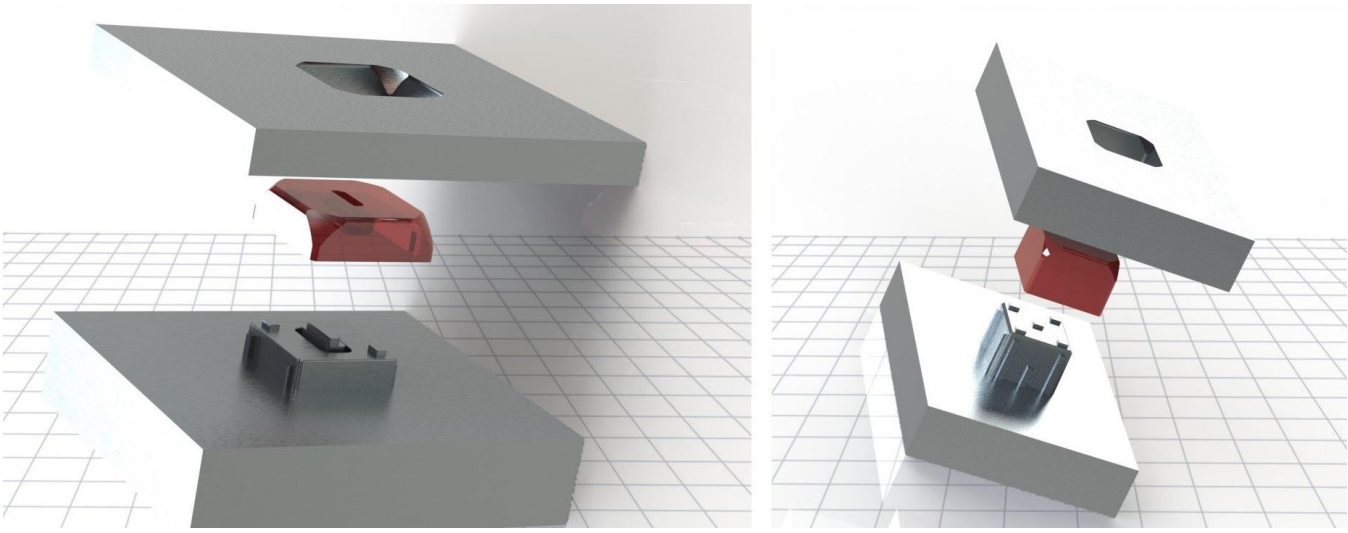


Figure 3.28 Mold for cap and for case to form a dice set.

The part on the left side can be correlated to the flexible circuit design, the top surface has a hole that corresponds to the cutout hole in the flexible circuit and is the point where the USB-b connector will be accessible from the outside in order to program the micro-controller and to charge the lithium polymer battery

### 3.5 Cost Comparison of Fabrication Techniques

Given the 4th generation dice was designed to be a commercially viable product; in this section a direct comparison of the associated costs of building parts through the previously seen methodology are shown side by side with the improved cost structure achievable through the methodologies selected for this 4th generation design. The first set of items to evaluate is the cost of components, which were re-evaluated with this specific purpose in mind. Table 3.1 shows the bill of materials for both the 3rd and 4th generations of dice, after this re-evaluation was made.

Table 3.1 Bill of materials comparison between version 3 and 4 designs

BILL OF MATERIALS FOR 3RD GENERATION DICE COMPONENTS

| Vendor P/N | Description | Reference Designator (s) | Unit Cost | Qty. | Extended Cost |
|---|---|---|---|---|---|
| 609-4050-1-ND | CONN RCPT STD MICRO USB TYPE B | J1 | 0.267 | 1 | 0.267 |
| MC34673AEPR2CT-ND | IC SGL CELL BATTERY CHRGR 8-UDFN | U1 | 0.51364 | 1 | 0.51364 |
| 628-8241-ABPM-G | BATTERY PROTECTION IC SAFETY | U2 | 0.30305 | 1 | 0.30305 |

| | | | | | |
|---|---|---|---|---|---|
| | CIRCUIT | | | | |
| GM041215 | Lithium Polymer Battery 45mAh | BAT1: 1,2 | 1.37 | 1 | 1.37 |
| FDN337NTR-ND | MOSFET N-CHANNEL SSOT3 | Q1,Q2 | 0.10208 | 2 | 0.20416 |
| 311-68KARCT-ND | 1/8 watt resistor, 68K ohm resistor, 5% 0805 | R1 | 0.00182 | 1 | 0.00182 |
| 311-470KARCT-ND | 1/8 watt resistor, 470 ohm resistor, 5% 0805 | R2 | 0.00182 | 2 | 0.00364 |
| 311-1.0KARCT-ND | 1/8 watt resistor, 1K ohm resistor, 5% 0805 | R3 | 0.00182 | 2 | 0.00364 |
| C0402C229C4GACTU-ND | Ceramic Capacitor 2.2uf, 16V 0603 | C1 | 0.05425 | 1 | 0.05425 |
| 399-1096-1-ND | Ceramic Capacitor 0.1uf, 16V 0603 | C2,C3 | 0.0036 | 2 | 0.0072 |
| PIC18LF14K22T-I/ML-ND | IC PIC MCU FLASH 256K, 20 PIN VQFN | U1 | 1.51 | 1 | 1.51 |
| ADXL330KCPZ-RLTR-ND | IC ACCELEROMETER 3 AXIS | U2 | 8.748 | 1 | 8.748 |
| 399-1096-1-ND | Ceramic Capacitor 0.1uf, 16V 0603 | C1 | 0.05425 | 1 | 0.05425 |
| 311-5.6KARCT-ND | 1/8 watt resistor, 5.6K ohm, 5%, 0805 | R1 | 0.00182 | 1 | 0.00182 |
| 516-1445-1-ND | LED 469nm BLUE DIFF 0805 | D1-D21 | 0.378 | 21 | 7.938 |
| | | | | Total Cost | 20.98047 |

## BILL OF MATERIALS FOR 4TH GENERATION DICE COMPONENTS

| Vendor P/N | Description | Reference Designator (s) | Unit Cost | Qty. | Extended Cost |
|---|---|---|---|---|---|
| 609-4050-1-ND | CONN RCPT STD MICRO USB TYPE B | J1 | 0.267 | 1 | 0.267 |
| ATTINY20-MMHR-ND | MCU AVR 2KB FLASH 12MHZ 20QFN | U1 | 0.612 | 1 | 0.612 |
| MMA7660FCT-ND | ACCELEROMETER 3X3 DGTL 10-DFN | U2 | 0.6629 | 1 | 0.6629 |
| MC34673AEPR2CT-ND | IC SGL CELL BATTERY CHRGR 8-UDFN | U3 | 0.51364 | 1 | 0.51364 |
| GM041215-PCB | Lithium Polymer Battery 45mAh | BAT1: 1,2 | 1.87 | 1 | 1.87 |
| 604-APT1608QBC/D | Standard LED - SMD Blue 470nm Water Clear 100mcd | D1-D21 | 0.073 | 21 | 1.533 |
| 311-47ARTR-ND | 1/8 watt resistor, 47 ohm resistor, 5% 0805 | R1 | 0.00182 | 1 | 0.00182 |
| 311-22ARCT-ND | 1/8 watt resistor, 22 ohm resistor, 5% 0805 | R2, R6 | 0.00182 | 2 | 0.00364 |
| 311-15ARCT-ND | 1/8 watt resistor, 15 ohm resistor, 5% 0805 | R3, R5 | 0.00182 | 2 | 0.00364 |
| 311-12ARCT-ND | 1/8 watt resistor, 12ohm resistor, 5% 0805 | R4 | 0.00182 | 1 | 0.00182 |
| 311-100ARCT-ND | 1/8 watt resistor, 100 ohm resistor, 5% 0805 | R7, R8 | 0.00182 | 2 | 0.00364 |
| 311-4.7KARCT-ND | 1/8 watt resistor, 4.7K ohm resistor, 5% 0805 | R9,R10,R11 | 0.00182 | 3 | 0.00546 |
| 311-68KARCT-ND | 1/8 watt resistor, 68K ohm resistor, 5% 0805 | R12 | 0.00182 | 1 | 0.00182 |
| 399-1096-1-ND | Ceramic Capacitor 0.1uf, 16V 0603 | C1,C2 | 0.0036 | 2 | 0.0072 |

| | | | | | |
|---|---|---|---|---|---|
| 399-4922-1-ND | Ceramic Capacitor 0.47uf, 16V 0805 | C3,C4 | 0.019 | 2 | 0.038 |
| BA 892-02V H6127CT-ND | Diode RF 35V, 100mA SC79 | D22,D23 | 0.05904 | 2 | 0.11808 |
| | | | | Total Cost | 5.64366 |

The total cost reduction from $20.98 to $5.64 shows the significant cost savings per unit that were realized through judicious component selection based on cost. The next evaluation is to compare actual fabrication cost as each methodology requires specific processes. The structural electronics approach to prototype fabrication requires the use of Stereolithography equipment and silver ink deposition whereas the traditional methodology requires a flexible circuit and a plastic molded part. The cost to manufacture the substrate using Additive Manufacturing was obtained based on the volume of the part and the number of parts that can be made using the Stereolithography methodology with the Viper SI2 SLA system (3D systems, Valencia, CA.), this cost is valid if the machine is used to fabricate 32 sets at one time; for smaller quantities the price is adjusted upwards up to $20.00 ea. if one set is made. The cost to fabricate each case using plastic injection molding process was obtained from the company Protomold (http://www.protomold.com) this company specializes in fabrication of small quantities of plastic molded parts; based on material selection, lead time requirement and quantity of parts desired, a sample quote was obtained that showed an approximate price.

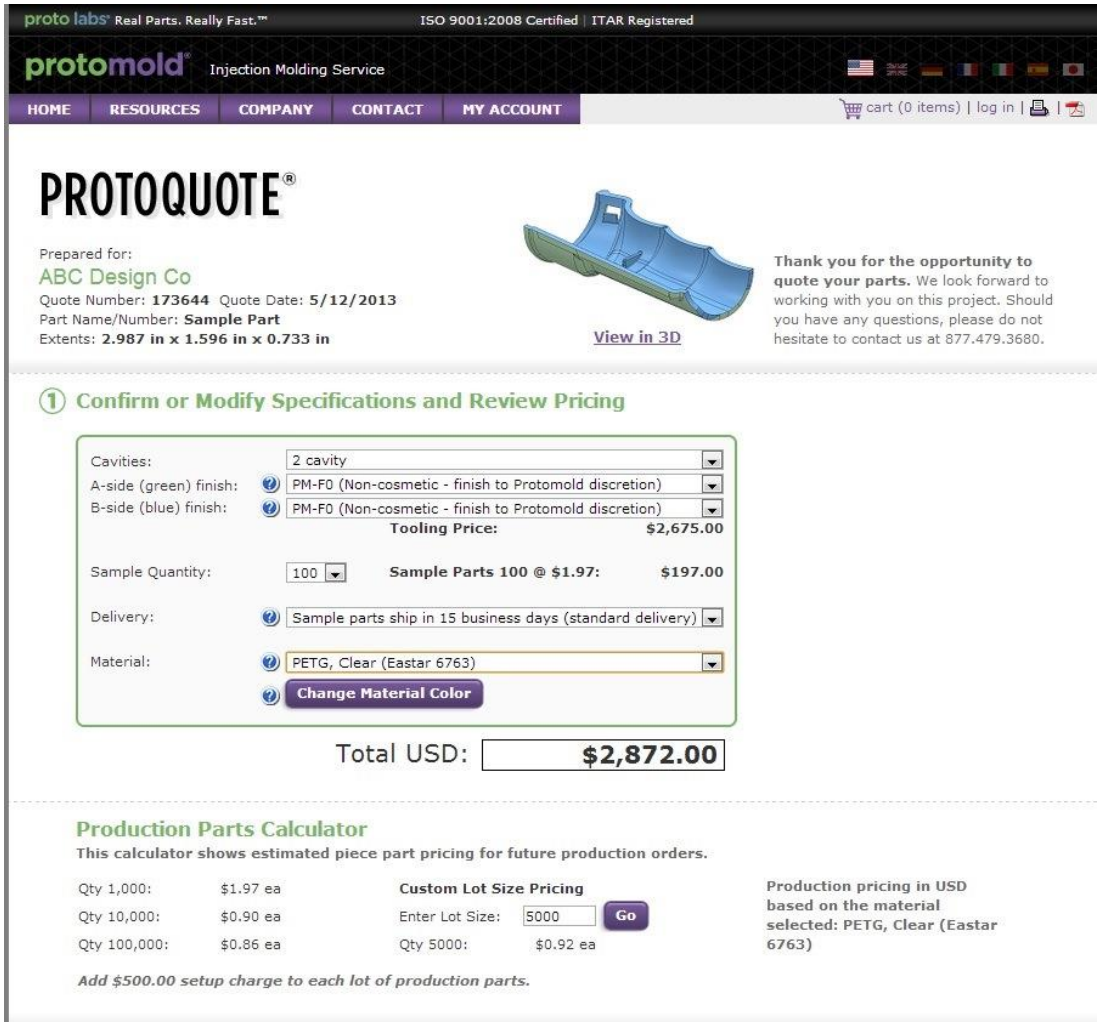



Figure 3.29 Sample quote for plastic parts

Additionally, using the fixed tooling price of $2,675.00 and the listed unit price of $1.97 (for a quantity of 100 to 1,000 from quote above), a true total price can be obtained and dividing by the number of units, an all-inclusive unit price can be obtained. It is evident that as few as 210 pcs. ordered is enough to have a lower price than a fully utilized SL machine; by 1,000 pcs. the unit price is already ~1/3 than the price using SL; as can be seen in Figure 3.30

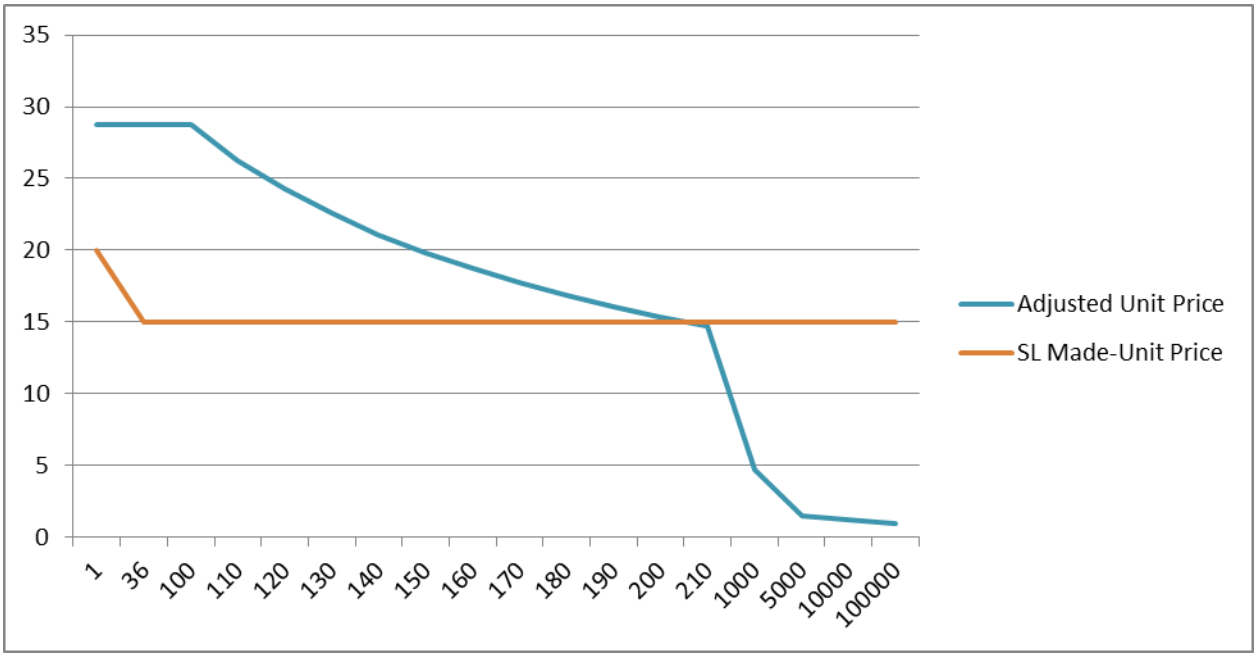



Figure 3.30 Price drop based on ordered quantity

The estimated per-piece pricing for production orders, specifically for a 10,000 part order is the price used, because 10,000 parts is the lot quantity used for all materials cost estimates. The cost to fabricate the electronic circuit is shown only for the 4$^{th}$ generation approach to indicate the per piece cost of fabricating the flexible circuits. A quote for 10,000 provided by an external supplier is used to justify this price. The cost to fabricate using structural electronics methodology is illustrated in the labor category since the circuit itself is not pre-fabricated and is instead built onto the substrate itself. The estimated cost was based on 8 hrs. of labor, since that is the average time to build one part our previous experience suggests. The total cost of $32.00 was obtained using a $4.00/hr. labor rate because $4.00 is a typical value for a fully burdened hourly labor rate that can be obtained by outsourcing the assembly to a twin plant company in Juarez, Mexico. In order to have a fair comparison, we used the same labor rate for both versions being compared. The labor cost for the traditional manufacturing was obtained by assuming a one minute assembly time for folding a flexible circuit into the plastic case and ultrasonically sealing it; further an 80% efficiency was also considered, giving a production rate of 40 pcs./hr. therefore the $4.00 labor rate is divided by 40 pcs. to obtain a labor cost per piece of $0.10. Table 3.2 shows the summary of fabrication costs for each of the two methodologies discussed.

Table 3.2 Summary of costs for each fabrication methodology

| | Structural Electronics | Traditional Mfg. |
|---|---|---|
| Cost to manufacture the substrate/case | $15.00 | $0.90 |
| Cost to fabricate electronic circuit | $0.00 | $3.47 |
| Labor Cost | $32.00 | $0.10 |
| **Mfg. Cost per piece** | $47.00 | $4.47 |
| **Materials Cost** | $20.98 | $5.65 |
| **Total Cost per piece** | $67.98 | $10.12 |

# Chapter 4: Applications

The capability to create fully functional prototypes through the use of 3D printing has allowed the ability to move a new design quickly through the design and improvement phases and to reach a level where a commercially viable product can be created. This is made possible through the use of Additive Manufacturing techniques coupled with Direct Print micro-dispensing. This section illustrates the progressive advances that were achievable thanks to the rapid implementation of improvements to each version of this dice design, made possible by the use of state of the art equipment available at the W. M. Keck Center.

## 4.1 ORIGINAL DESIGN, POWERED BY REPLACEABLE BATTERIES

The original dice was made as a proof of concept to demonstrate whether this complicated design could be implemented within the available volume.

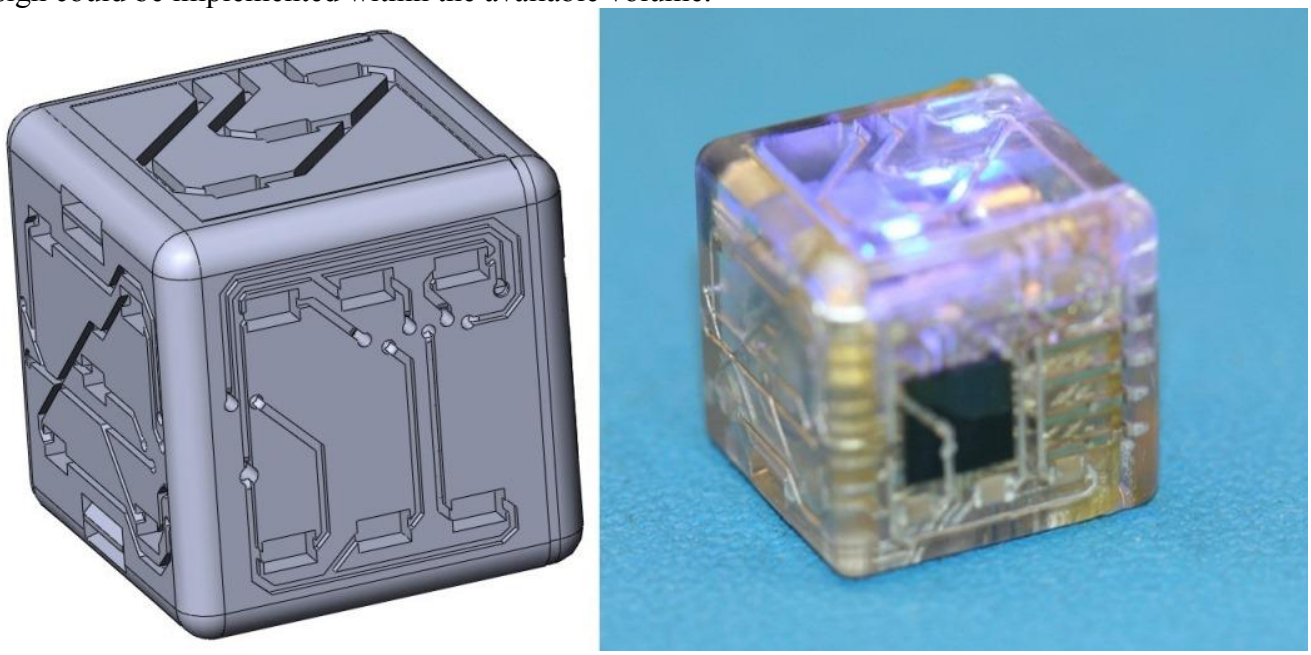


Figure 4.1 The designed and the implemented dice, version 1

The implementation was successful in demonstrating the capability to use 3D printing to achieve a prototype that would also include a functional circuit within. The items that were seen as requiring improvement were: the use of disposable batteries, the need of a natural look, absent by lacking rounded corners; as seen in figure 4.1

## 4.2 SECOND ITERATION, RECHARGED THROUGH WIRELESS MAGNETIC FIELD

The second iteration was implemented to improve on the previous design; it included a rechargeable lithium-polymer battery with safety and charging circuits, a better design with rounded corners and a wireless charging system. Figure 4.2 shows the designed dice for the 2nd generation and the necessary steps to achieve it. The battery module, which includes the safety and charging circuits must be fabricated first and inserted within the dice cavity and connected to power it; the cap is then assembled to complete the assembly.

Figure 4.2 Design of 2nd generation dice and implementation steps.

Figure 4.3 shows the charger designed to produce the resonant magnetic field to induce power in the battery module within each dice, to charge its battery.

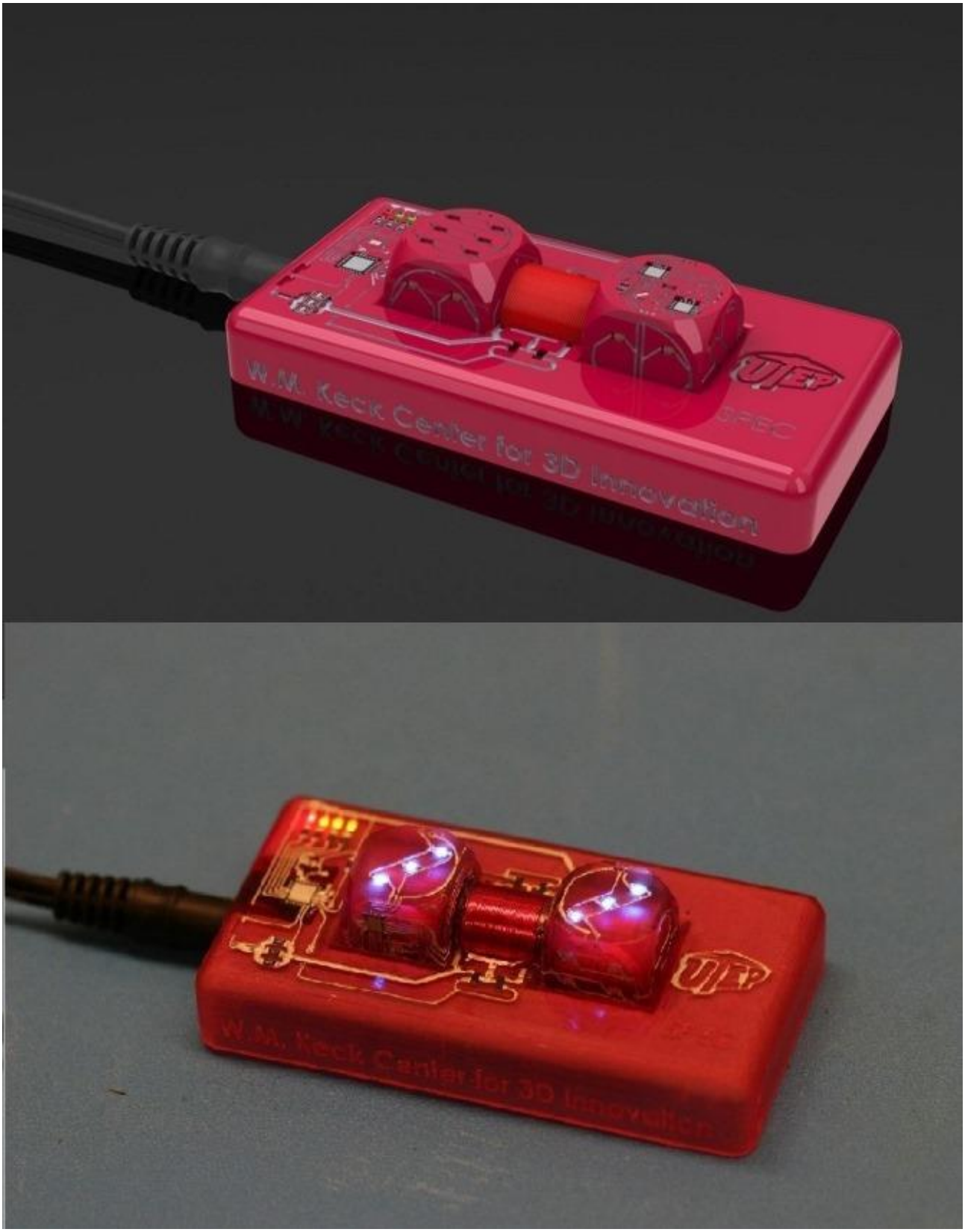



Figure 4.3 Charger design and implementation

The second generation of the dice design addressed the concerns found in the first version, however new shortcomings were identified, such as the increased complexity of having to design and build a charger device to recharge the batteries. Thanks to the capacity to build functional prototypes within a short time span, these would be addressed in the following generation.

### 4.3 THIRD VERSION, RECHARGED THROUGH A MICRO USB-B CONNECTOR

The 3rd generation of the dice changed the method for recharging the battery and the cap was eliminated by merging it with the power module into a single part to improve the ease of fabrication.

The charger was eliminated and in its place a commercially available USB to USB-b cable would be used to charge the dice from a USB port on any appropriate device.

Figure 4.4 shows the design of the cap and battery module part that includes a cavity for the micro USB-b connector to use for charging the battery.

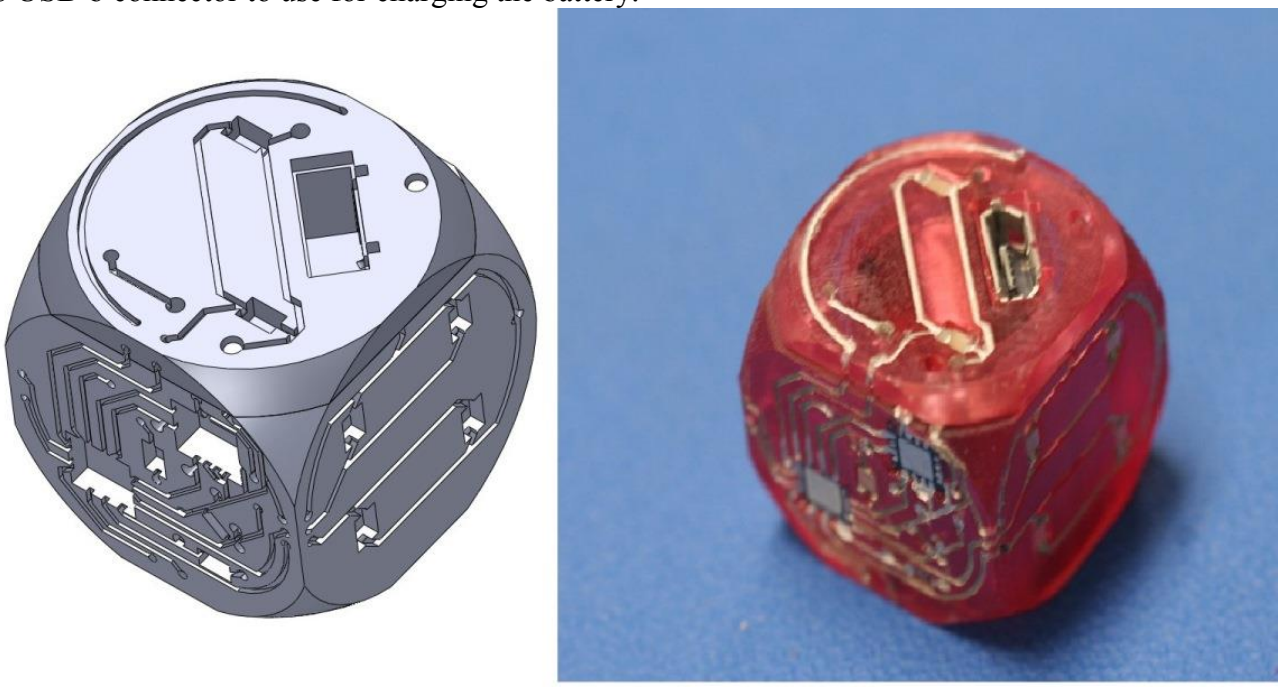


Figure 4.4 Design and Implementation of version 3 dice

This version of the dice was seen as a good design; however given the next step in its evolution would be a truly commercial product, a significant redesign was made to make it compatible for flexible circuit and plastic injection molding implementation. Additionally the components were re-evaluated with a cost saving mindset.

### 4.4 PROTOTYPE MADE THROUGH TRADITIONAL MANUFACTURING TECHNIQUES

The 4th generation of the dice design includes several changes, though related to cost improvements and improvements to increase ease of manufacture. Figure 4.5 shows an image of the case design and implementation using Somos Watershed Stereolithography resin, due to time constraints, the plastic part was not built in time.

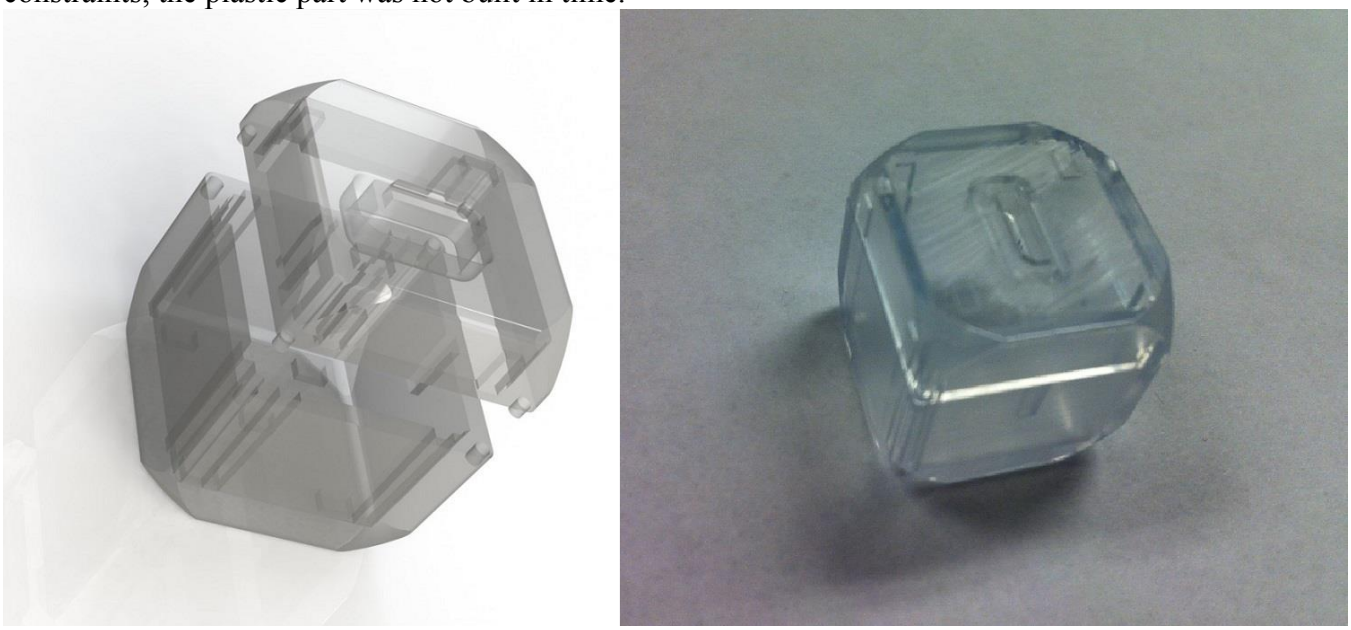


Figure 4.5 Plastic case design and implementation

Figure 4.6 shows the design of the flexible circuit and the improved rendered image of the implemented circuit.

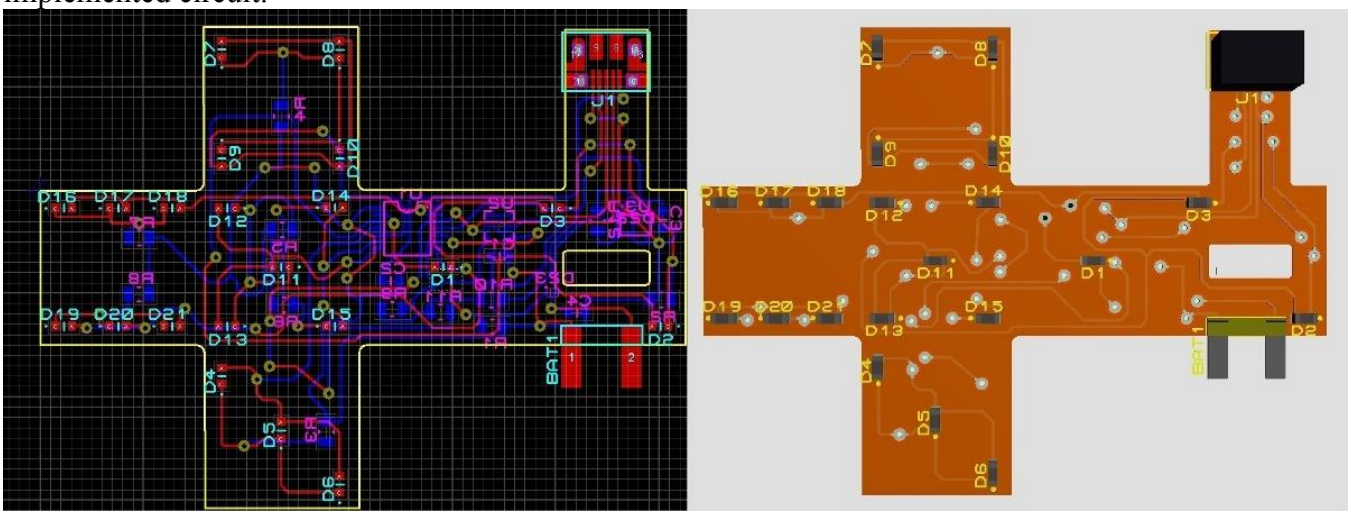
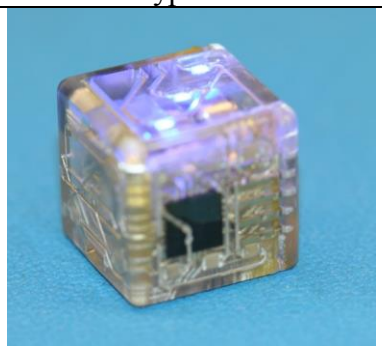



Figure 4.6 The flexible circuit design and implementation

### 4.5 SUMMARY OF SMART DICE PROTOTYPE VERSIONS

Table 4.1 below summarizes the four smart dice versions fabricated to: demonstrate feasibility, optimize the design and lastly to create a commercially viable version. Additionally each version is accompanied by a fabrication schedule, which illustrates the interval each design required to be implemented using Structural Electronics methodology for the first three versions and through traditional manufacturing techniques for the fourth iteration.

Table 4.1 Summary of prototype versions

| Prototype Version | Schedule | Purpose | Improvement Opportunities |
|---|---|---|---|
|  Version 1 | May 18, 2010- August 6, 2010 | - To demonstrate feasibility of packaging components required to create a functional smart dice design in the available volume. | - Use of rechargeable batteries.<br>- Round-off corners for a more natural feel.<br>- Change design to hollow cube and cap. |
|  Version 2 | August 9, 2010- November 22, 2010 | - To improve upon previous design.<br>- To demonstrate capability of wireless recharge system. | - Eliminate the complexity of additional charger system.<br>- Find practical alternative for charging |

| | | | |
|---|---|---|---|
|  Version 3 | November 2011- January 23, 2012 | - To improve upon previous design.<br>- To approach a commercially viable design. | - To create a truly commercially viable design.<br>- To find best methodology for a commercial product. |
|  Version 4 | August 24, 2012- June 2013 | - To re-evaluate materials and methods for commercial viability.<br>- To evaluate materials and methods selected for manufacturability. | - To be determined based on hands on evaluation. |

# Chapter 5: Conclusions

## 5.1 COMPARISON OF PROTOTYPE FABRICATION TECHNIQUES

Table 5.1 Prototype fabrication methodologies

| | Breadboard Prototyping | Structural Electronics | Flexible circuit with SL case | Flex. circuit with Molded Case |
|---|---|---|---|---|
| Process Step | Time Required | Time Required | Time Required | Time Required |
| Electronic Circuit Design | 10 hrs. | 10 hrs. | 10 hrs. | 10 hrs. |
| Sourcing Components | 7 hrs. | 6 hrs. | 6 hrs. | 6 hrs. |
| Breadboard Prototype | 4 hrs. | | | |
| Sourcing Flexible Circuit | | | 24 hrs. | 24 hrs. |
| Populating Flexible circuit | | | 8 hrs. | 8 hrs. |
| Mechanical Design | | 10 hrs. | 10 hrs. | 10 hrs. |
| Build or procure SL Case | | 4 hrs. | 24 hrs. | |
| Procure molded part | | | | 24 hrs. |
| Interconnect Deposition | | 8 hrs. | | |
| Assemble/Test | | | 4 hrs. | 4 hrs. |
| **Total** | **21** hrs. | **38** hrs. | **62** hrs. | **62** hrs. |

The main two fabrication methodologies have been thoroughly discussed where applicable. The Structural Electronics technique used for versions 1, 2 and 3 for prototype optimization and the traditional fabrication technique of flexible circuit fabrication and plastic injection molded case used in version 4. The alternatives discussed and hinted at in the abstract have been compiled to evaluate advantages and disadvantages for each as well. Table 5.1 illustrates the schedule required to fabricate a functional prototype of version 3 dice. As expected, a simple breadboard prototype has a time frame comparable to Structural Electronics methodology, but is still faster than any other alternative. However, this option lacks the ability to showcase the final appearance, fit or feel of the intended product. As such, it is no more a viable prototyping alternative than a valuable precursor to any of the three realistic methodologies.

Among the three remaining techniques, the total time for the two options that require a flexible circuit, does not equal the sum of the time required for each of the listed tasks, this is considered valid if both the sourcing of a flexible circuit and the case (regardless of type of fabrication) is done concurrently. Also noteworthy is that for the Structural Electronics method, employed at the W. M. Keck Center the time frame to build an SL substrate is only 4 hours, thanks to equipment availability, while for those who source these components from a vendor, the time required is 24 hours at best. Consequently, both of the methodologies that require major components sourced from outside vendors are still nearly twice as time-consuming as the Structural Electronics prototyping methodology. Additionally the 24 hour time frame to source these components from outside suppliers are based on a best case scenario of a "next day delivery" option, given that such a short turn-around time is not a typical time frame for obtaining these components rather it's an available, albeit costly option.

## 5.2 FINAL CONCLUSIONS

From table 4.1 we can see that, whereas the first three versions, fabricated using Structural Electronics, each required approximately 3 months to go from concept to fully functional electronics prototype; the fourth version, implemented through traditional manufacturing techniques required 10 months to complete. This schedule analysis demonstrates how Structural Electronics methodology for prototype fabrication affords a ~70% reduction in implementation time.

A more detailed analysis, shown in table 5.1 shows that a reduction from 62 hrs. to 38 hrs. can be achieved. This 38% reduction in effective time is still achieved even if the best case scenario of quick turn-around times is realized for the more traditional methodologies at a significant cost penalty.

However, whereas 3D printing has proven itself a useful tool to realize practical prototypes with functional circuits, the cost per piece for mass production is not competitive with traditional manufacturing techniques, as illustrated in figure 3.30. This research has shown Structural Electronics is a valuable tool to advance a design through the necessary improvement steps which result from hands-on analysis borne of accurate prototypes. The last prototype stage was nonetheless undertaken using traditional manufacturing methodologies in order to more faithfully represent the final product as available to a consumer. This is necessary to truly evaluate the cost of manufacture. However, as additive manufacturing technologies increase their selection of available materials and as interconnection methodologies improve their reliability, the use of 3D printing may eventually provide sufficient justification to move to the production phase. In conclusion, the value of 3D printing as a tool to quickly implement improvements throughout the prototype stage has been demonstrated; as the capabilities of this methodology improve through the automation of process steps and expanding of its capabilities, this technology is only going to see its use increase.

# References


[1] Pham, D. T., and R. S. Gault. 1998. "A comparison of rapid prototyping technologies." International Journal of Machine Tools and Manufacture 38, no. 10-11: 1257-1287.

[2] Paulo Jorge Bartolo, Ian Gibson. 2011. “Stereolithography: Materials, Processes and Applications” ISBN: 0387929037. Chapter 2

[3] Krause, F-L., M. Ciesla, Ch. Stiel, and A. Ulbrich. 1997. "Enhanced rapid prototyping for faster product development processes." CIRP Annals-Manufacturing Technology 46, no. 1. 93-96.

[4] C.W. Hull. Apparatus for production of three-dimensional objects by Stereolithography, US Patent 4575330, 1986

[5] Kruth, J-P., M. C. Leu, and T. Nakagawa. "Progress in additive manufacturing and rapid prototyping." CIRP Annals-Manufacturing Technology 47, no. 2 (1998): 525-540.

[6] J.A. Palmer, P. Yang, D.W. Davis, B.D. Chavez, P.L. Gallegos, R.B. Wicker, and F. Medina. 2004. “Rapid Prototyping of High Density Circuitry,” Proceedings of the Rapid Prototyping & Manufacturing 2004 Conference, Rapid Prototyping Association of the Society of Manufacturing Engineers, May 10-13, 2004, Hyatt Regency Dearborn, Dearborn, Michigan. Also, SME Technical Paper TP04PUB221 (Dearborn, Mich.: Society of Manufacturing Engineers, 2004).

[7] Levy, Gideon N., Ralf Schindel, and Jean-Pierre Kruth. "Rapid manufacturing and rapid tooling with layer manufacturing (LM) technologies, state of the art and future perspectives." CIRP Annals-Manufacturing Technology 52, no. 2 (2003): 589-609.

[8] Altan, T., Lilly, B., & Yen, Y. C. (2001). “Manufacturing of dies and molds.” CIRP Annals-Manufacturing Technology, 50(2), 404-422.

[9] Beck, James E., Fritz B. Prinz, DanielP Siewiorek, and L. E. Weiss.(1992) "Manufacturing mechatronics using thermal spray shape deposition." In Proceedings of Solid Freeform Fabrication Symposium, pp. 272-9. ASCE, 1992.

[10] Prinz, Fritz B., Lee E. Weiss, and Daniel P. Siewiorek. (1994)"Electronic packages and smart structures formed by thermal spray deposition." U.S. Patent 5,278,442, issued January 11, 1994.

[11] Weiss, Lee, F. Prinz, G. Neplotnik, K. Padmanabhan, L. Schultz, and R. Merz. (1996) "Shape deposition manufacturing of wearable computers." In Proceedings of the 1996 Solid Freeform Fabrication Symposium, pp. 31-38. The University of Texas at Austin, 1996.

[12] Prinz, Fritz B., and Lee E. Weiss. (1998) "Novel applications and implementations of shape deposition manufacturing." Naval research reviews 50: 19-26.

[13] Cham, Jorge G., Beth L. Pruitt, Mark R. Cutkosky, Mike Binnard, Lee E. Weiss, and Gennady Neplotnik. (1999)."Layered manufacturing with embedded components: process planning considerations." In Proceedings of DETC99: 1999 ASME Design Engineering Technical Conference, Las Vegas, NV, September 12-15.

[14] Palmer, J. A., Davis, D. W., Chavez, B. D., Gallegos, P. L., Wicker, R. B., & Medina, F. R. (2008). U.S. Patent No. 7,419,630. Washington, DC: U.S. Patent and Trademark Office.

[15] K.H. Church, C. Fore, and T. Feeley. (2000) “Commercial Applications and Review for Direct Write Technologies,” Materials Development for Direct Write Technologies, San Francisco, CA, April 24-26, Volume 624, pp. 3-8.

[16] Medina, Frank, A. J. Lopes, A. V. Inamdar, Robert Hennessey, J. A. Palmer, B. D. Chavez, Don Davis, Phil Gallegos, and R. B. Wicker. (2005) "Hybrid Manufacturing: Integrating Direct-Write and Stereolithography." Proceedings of the 2005 Solid Freeform Fabrication.

[17] Lopes, A. J., Misael Navarrete, Francisco Medina, J. A. Palmer, Eric MacDonald, and R. B. Wicker. (2006) "Expanding rapid prototyping for electronic systems integration of arbitrary form." In Proceedings of the 17th Annual Solid Freeform Fabrication Symposium, University of Texas at Austin, Austin, TX. Society of Manufacturing Engineers.

[18] Periard, Daniel, Evan Malone, and Hod Lipson. (2007) "Printing embedded circuits." In Proceedings of the 18th Solid Freeform Fabrication Symposium, Austin TX, pp. 503-12. Society of Manufacturing Engineers.

[19] Berry, Megan E., Evan Malone, and Hod Lipson. (2005) "FREEFORM FABRICATION OF ZINC AIR BATTERIES WITH TAILORED GEOMETRY AND PERFORMANCE." Proceedings of the 16th Solid Freeform Fabrication Simposium, Austin, TX.

[20] Malone, Evan, and Hod Lipson. (2007) "Freeform fabrication of a complete electromechanical relay." In Proceedings of the 18th Solid Freeform Fabrication Symposium, pp. 513-526.

[21] Navarrete, Misael, Amit Lopes, Jacqueline Acuna, Raul Estrada, Eric MacDonald, Jeremy Palmer, and Ryan Wicker. (2007) "Integrated layered manufacturing of a novel wireless motion sensor system with GPS". TEXAS UNIV AT EL PASO WM KECK CENTER FOR 3D INNOVATION.

[22] DeNava, Erick, Misael Navarrete, Amit Lopes, Mohammed Alawneh, Marlene Contreras, Dan Muse, Silvia Castillo, Eric MacDonald, and Ryan Wicker.(2008). "Three-Dimensional Off-Axis Component Placement and Routing for Electronics Integration using Solid Freeform Fabrication." In Proceedings of Solid Freeform Fabrication Symposium, the University of Texas at Austin, Austin TX, pp. 362-369.

[23] Olivas, Richard I. (2011) "Conformal electronics packaging through additive manufacturing and micro-dispensing."

[24] Arnold, Craig B., Pere Serra, and Alberto Piqué. (2007) "Laser direct-write techniques for printing of complex materials." Mrs Bulletin 32, no. 01: 23-31.

[25] Stucker, Brent E., and Ryan Wicker. (2012) "Direct Digital Manufacturing of Integrated Naval Systems Using Ultrasonic Consolidation, Support Material Deposition and Direct Write Technologies". UTAH STATE UNIV LOGAN.

[26] Hernandez, Ludwing A. (2010) "Integration of Ultrasonic Consolidation and Direct-Write to Fabricate an Embedded Electrical System within a Metallic Enclosure".

[27] Verheecke, Wesley, Matthias Van Dyck, Frederik Vogeler, André Voet, and Hans Valkenaers. (2012) "Optimizing aerosol jet® printing of silver interconnects on polyimide film for embedded electronics applications." Status: published.

[28] Paulsen, Jason A., Michael Renn, Kurt Christenson, and Richard Plourde. (2012) "Printing conformal electronics on 3D structures with Aerosol Jet technology." In Future of Instrumentation International Workshop (FIIW), IEEE, pp. 1-4.

[29] Amit Joe Lopes, Eric MacDonald, Ryan B. Wicker, (2012) "Integrating stereolithography and direct print technologies for 3D structural electronics fabrication", Rapid Prototyping Journal, Vol. 18 Iss: 2, pp.129 - 143

# Appendix A: Micro-controller code for version 1 dice

```
////////////////////////////////////////////////////////////////////////////
//      File Name:   bbdice.c                                             //
//                                                                        //
//            Version:      2.1                                           //
//                                                                        //
//            Description:  LED dice program. The breadboard version with //
//            sleep mode and an interrupt service routine to wake         //
//            it up when motion is detected. With light up                //
//            sequences.    Modified to light up top side only by         //
//            testing for biggest value change from ADC                   //
//                                                                        //
//            Author:             Rudy Salas                              //
//                                                                        //
//            Last updated: July 26,2010                                  //
////////////////////////////////////////////////////////////////////////////

#include <adc.h>
#include <p18f2520.h>
#include <delays.h>
#include <portb.h>

//////////////////////////Setting Configuration bits//////////////////////////
#pragma config OSC = INTIO67, FCMEN = OFF, IESO = OFF
#pragma config PWRT = ON, BOREN = OFF
#pragma config WDT = OFF
#pragma config PBADEN = OFF
#pragma config MCLRE = ON, LPT1OSC = ON
#pragma config STVREN = OFF, LVP = OFF
#pragma config XINST = OFF, DEBUG = OFF
#pragma config CP0 = OFF, CP1 = OFF, CP2 = OFF, CP3 = OFF
#pragma config CPB = OFF, CPD = OFF
#pragma config WRT0 = OFF, WRT1 = OFF, WRT2 = OFF, WRT3 = OFF
#pragma config WRTB = OFF, WRTD = OFF
#pragma config EBTR0 = OFF, EBTR1 = OFF, EBTR2 = OFF, EBTR3 = OFF
#pragma config EBTRB = OFF
//////////////////////////Variables used/////////////////////////////////////
int           max, min, value, moved, a, b, c, i, j, k, h, result[3], second[3];
unsigned char ch[3] = {ADC_CH12,ADC_CH1,ADC_CH2};

//////////////////////////LED assignments////////////////////////////////////
const int
              ONE =  0b00000001,          // ON PORT C
              TWO =  0b11000000,          // ON PORT B
              THREE = 0b00001110,         // ON PORT C
              FOUR =        0b11110000,         // ON PORT C
              FIVE =        0b11111000,         // ON PORT A
              SIXB =        0b00111110,         // ON PORT B AND MUST INCLUDE PORT A
              SIXA =        0b00000001;         // ON PORT A AND MUST INCLUDE PORT B

//////////////////Function prototypes for light up sequences////////////////
void ONESQ    (void);
void TWOSQ    (void);
void THREESQ(void);
void FOURSQ   (void);
void FIVESQ   (void);
void SIXSQ    (void);
```

```
//////////////////LED  light up sequence functions//////////////////////
void ONESQ ()
       {
       PORTA=0;
       PORTB=0;
       PORTC=ONE;
       Delay10KTCYx(10);
       PORTC=0;
              for (h=0;h<3;h++)
                     {
                     Delay10KTCYx(3),
                     PORTC=ONE,
                     Delay10KTCYx(3),
                     PORTC=0,
                     Delay10KTCYx(3);
                     }
       PORTC=ONE;
       Delay10KTCYx(10);
       }

void TWOSQ ()
       {
       PORTA=0;
       PORTC=0;
       PORTB=TWO;
       Delay10KTCYx(10);
              for (h=0;h<3;h++)
                     {
                     PORTB=0b01000000,
                     Delay10KTCYx(3),
                     PORTB=0b10000000,
                     Delay10KTCYx(3),
                     PORTB=0,
                     Delay10KTCYx(3);
                     }
       PORTB=TWO;
       Delay10KTCYx(10);
       }

void THREESQ ()
       {
       PORTA=0;
       PORTB=0;
       PORTC=THREE;
       Delay10KTCYx(10);
              for (h=0;h<3;h++)
                     {
                     PORTC=0b00000010,
                     Delay10KTCYx(3),
                     PORTC=0b00000100,
                     Delay10KTCYx(3),
                     PORTC=0b00001000,
                     Delay10KTCYx(3),
                     PORTC=0,
                     Delay10KTCYx(3);
                     }
       PORTC=THREE;
       Delay10KTCYx(10);
       }
```

```
void FOURSQ ()
	{
	PORTA=0;
	PORTB=0;
	PORTC=FOUR;
	Delay10KTCYx(10);
		for (h=0;h<3;h++)
			{
			PORTC=0b10000000,
			Delay10KTCYx(3),
			PORTC=0b01000000,
			Delay10KTCYx(3),
			PORTC=0b00100000,
			Delay10KTCYx(3),
			PORTC=0b00010000,
			Delay10KTCYx(3),
			PORTC=0,
			Delay10KTCYx(3);
			}
	PORTC=FOUR;
	Delay10KTCYx(10);
	}

void FIVESQ ()
	{
	PORTB=0;
	PORTC=0;
	PORTA=FIVE;
	Delay10KTCYx(10);
		for (h=0;h<3;h++)
			{
			PORTA=0b01000000,
			Delay10KTCYx(3),
			PORTA=0b10000000,
			Delay10KTCYx(3),
			PORTA=0b00100000,
			Delay10KTCYx(3),
			PORTA=0b00010000,
			Delay10KTCYx(3),
			PORTA=0b00001000,
			Delay10KTCYx(3),
			PORTA=0,
			Delay10KTCYx(3);
			}
	PORTA=FIVE;
	Delay10KTCYx(10);
	}

void SIXSQ ()
	{
	PORTC=0;
	PORTA=SIXA;
	PORTB=SIXB;
	Delay10KTCYx(15);
		for (h=0;h<3;h++)
		{
		PORTA=0,
		PORTB= 0b00100000,
		Delay10KTCYx(3),
		PORTB= 0b00010000,
```

```
                Delay10KTCYx(3),
                PORTB= 0b00001000,
                Delay10KTCYx(3),
                PORTB= 0b00000100,
                Delay10KTCYx(3),
                PORTB= 0b00000010,
                Delay10KTCYx(3),
                PORTB= 0x00,
                PORTA= 0b00000001,
                Delay10KTCYx(3),
                PORTA=0,
                Delay10KTCYx(3);
                }
        PORTA=SIXA;
        PORTB=SIXB;
        Delay10KTCYx(10);
        }
////////////////////////////////ABSOLUTE VALUE FUNCTION//////////////////////
int abs(int val);

#pragma code
int abs(int val)
        {
                if (val<1)
                        {
                return val*(-1);
                        }
                else
                        return val;
        }

/////////////////////////////Interrupt Service Routine declaration///////////
void isr (void);
#pragma interrupt isr

#pragma code high_vector = 0x08
void high_vector ()
                {
                _asm GOTO isr _endasm
                }

//////Interrupt Service Routine, all it does is clear the flag and wake up//
#pragma code
void isr (void)
                {
                INTCONbits.INT0IF=0;
                }

////////////////////////////Program start/////////////////////////////////////
#pragma code
void main ()
{
/* SETTING 3 ANALOG PORTS AN1(RA1), AN2(RA2) AND AN12(RB0) AS INPUTS (1)
 THE REST OF THE A PORT,(RA0 AND RA3 THROUGH RA7)
 THE B PORT FROM B1 TO B7 AND ALL THE C PORTS AS OUTPUTS (0) FOR THE LED'S*/
                TRISA = 0b00000110;
                TRISB = 0b00000001;
                TRISC = 0b00000000;

///////////////// INFINITE LOOP TO PERFORM THE TASK CONTINUOUSLY/////////////
```

```
while (1)
       {
////////////////DO Loop to do while dice is still moving////////////////////
       do
              {
              moved=0;
/////////////// A LOOP TO TAKE ANALOG SAMPLES FROM ALL 3 CHANNELS////////////
                     for (i=0;i<3;i++)
                            {
                            OpenADC (     ADC_FOSC_64  &
                            ADC_RIGHT_JUST &
                            ADC_12_TAD,
                            ch[i]         &       //Using unsigned char ch[3] to get the 3
channels
                            ADC_REF_VDD_VSS &
                            ADC_INT_OFF,12);
                            Delay10TCYx(5);
                            ConvertADC();
                            while(BusyADC());
                            result[i]=ReadADC();
                            CloseADC();
                            }
//////////A 0.2 SECOND DELAY TO TEST FOR STOPPAGE OF MOTION ON DICE////////
                            Delay10KTCYx(10);
////////////// TAKING A SECOND READING FROM ALL 3 CHANNELS//////////////////
                     for (j=0;j<3;j++)
                            {
                            OpenADC (     ADC_FOSC_64  &
                            ADC_RIGHT_JUST &
                            ADC_12_TAD,
                            ch[j]         &
                            ADC_REF_VDD_VSS &
                            ADC_INT_OFF,12);
                            Delay10TCYx(5);
                            ConvertADC();
                            while(BusyADC());
                            second[j]=ReadADC();
                            CloseADC();
                            }
//////////////To determine if dice is still moving////////////////////////////////////
                    if( (result[0]-second[0])*(result[0]-second[0])>60)
                           moved=moved+1;
                    if( (result[1]-second[1])*(result[1]-second[1])>60)
                           moved=moved+1;
                                  if( (result[2]-second[2])*(result[2]-second[2])>60)
                           moved=moved+1;

              } while(moved>0);

////////////// IF MOTION STOPPED, LIGHT UP THE LEDS (ON TOP ONLY)//////////////////////

                            max=544;      //Upper value limit for no acceleration detected.
                            min=430;      //Lower value limit for no acceleration detected.

                     for (h=0;h<3;h++)
                            {
                            OpenADC (     ADC_FOSC_64  &
                            ADC_RIGHT_JUST &
                            ADC_12_TAD,
                            ch[h]         &
```

```
                        ADC_REF_VDD_VSS &
                        ADC_INT_OFF,12);
                        Delay10TCYx(5);
                        ConvertADC();
                        while(BusyADC());
                        result[h]=ReadADC();
                        CloseADC();
                        }

//////////////////////To determine which side is up////////////////////////////////
                                                a=abs(result[1]-512);
                                                b=abs(result[2]-512);
                                                c=abs(result[3]-512);

                                                if (a>b)
                                                       {if (a>c)
                                                              {
                                                              if (result[1]<min)
                                                                     FIVESQ();
                                                              if (result[1]>max)
                                                                  TWOSQ();
                                                              }
                                                       else
                                                              {
                                                              if (result[3]<min)
                                                                     THREESQ();
                                                              if (result[3]>max)
                                                                  FOURSQ();

                                                              }
                                                       }
                                                else
                                                       {if (b>c)
                                                              {
                                                              if (result[2]<min)
                                                                     SIXSQ();
                                                              if (result[2]>max)
                                                                  ONESQ();
                                                              }
                                                       else
                                                              {
                                                              if (result[3]<min)
                                                                     THREESQ();
                                                              if (result[3]>max)
                                                                  FOURSQ();
                                                              }
                                                       }

//Sleep sequence. Configure interrupt, clear flag, enable global interrupt, turn off all LEDs.
Go to sleep///
       OpenRB0INT(PORTB_CHANGE_INT_ON & RISING_EDGE_INT & PORTB_PULLUPS_OFF);
       INTCONbits.INT0IF=0;  //Clear Interrupt Flag if it occurred while awake
       INTCONbits.GIE=1;      //Enable global interrupts for wake-up during sleep mode
       PORTA=0;              //Turn off PORT A LEDs
       PORTB=0;              //Turn off PORT B LEDs
       PORTC=0;              //Turn off PORT C LEDs
       Sleep ();                     //Go to sleep. Power save mode.
        }
}
```

# Appendix B: Micro-controller code for version 2 dice charger

```
#include <adc.h>
#include <p18f2520.h>
#include <timers.h>
#include <delays.h>
#include <portb.h>


// Config bit settings
#pragma config OSC = INTIO7, FCMEN = OFF, IESO = OFF
#pragma config PWRT = ON, BOREN = OFF
#pragma config WDT = OFF
#pragma config PBADEN = OFF
#pragma config MCLRE = ON, LPT1OSC = ON
#pragma config STVREN = OFF, LVP = OFF
#pragma config XINST = OFF, DEBUG = OFF
#pragma config CP0 = OFF, CP1 = OFF, CP2 = OFF, CP3 = OFF
#pragma config CPB = OFF, CPD = OFF
#pragma config WRT0 = OFF, WRT1 = OFF, WRT2 = OFF, WRT3 = OFF
#pragma config WRTB = OFF, WRTD = OFF
#pragma config EBTR0 = OFF, EBTR1 = OFF, EBTR2 = OFF, EBTR3 = OFF
#pragma config EBTRB = OFF, CCP2MX = PORTBE

//Function Prototypes
void tmr2_isr(void);

#pragma interrupt tmr2_isr

#pragma code high_vector=0x08
void high_vector ()
{
 //DETERMINE WHICH INTERRUPT OCURRED
 if(PIR1bits.TMR2IF) //If Timer 2 int occured
   {
    _asm GOTO tmr2_isr _endasm
   }
}

///Interrupt Service Routine that is invoked when Timer2 overflows
#pragma code
void tmr2_isr (void)
{

     PORTA=PORTA^0b00000011;

     PIR1bits.TMR2IF = 0;
}

void main()
{
ADCON1=0b00001111;
INTCONbits.GIEH=0;
PIE1 = 0b00000010;
RCON = RCON|0b10000000;  //Enable priority interrupts
PR2 = 0b00000010;

TRISC=0x00; //Make port C an output
TRISA=0x00;
```

```
PORTC=0b00000000;
PORTA=0b00000001;

//Set up internal Oscillator for 8MHz
OSCCONbits.IRCF2=1;               //
OSCCONbits.IRCF1=1;               // OSCON<6:4> 111
OSCCONbits.IRCF0=1;               //

OSCTUNEbits.PLLEN=1;   //Enable PLL (For Multiplier X4) 8X4=32MHz

OpenTimer2( TIMER_INT_ON &
            T2_PS_1_1     &
            T2_POST_1_10   );


while(1)
{

INTCONbits.GIEH=0;
PORTA=0b00000000;
Delay10KTCYx(128);
PORTA=PORTA^0b00000001;
INTCONbits.GIEH=1;
Delay1KTCYx(128);

}

}
```

## Appendix C: Micro-controller code for version 2 and 3 dice

```
////////////////////////////////////////////////////////////////////////////
//      File Name:   diceout.c                                            //
//                                                                        //
//            Version:      2.0                                           //
//                                                                        //
//            Description:  LED dice program. The newer version where the //
//            accelerometer and PIC are outside along with all            //
//            traces. The micro used is PIC18LF14K22                      //
//                                                                        //
//            Author:              Rudy Salas                             //
//                                                                        //
//            Last updated: February 25, 2011                             //
////////////////////////////////////////////////////////////////////////////

#include <adc.h>
#include <p18lf14k22.h>
#include <delays.h>

//////////////////////////Setting Configuration bits//////////////////////////
#pragma config FOSC = IRC, PLLEN=OFF, PCLKEN=ON, FCMEN = OFF, IESO = OFF
#pragma config PWRTEN = ON, BOREN = OFF
#pragma config WDTEN = OFF, WDTPS=1
#pragma config HFOFST=OFF, MCLRE = ON
#pragma config STVREN = OFF, LVP = OFF, XINST = OFF
#pragma config CP0 = OFF, CP1 = OFF
```

```
#pragma config CPB = OFF, CPD = OFF
#pragma config WRT0 = OFF, WRT1 = OFF
#pragma config WRTC=OFF, WRTB = OFF, WRTD = OFF
#pragma config EBTR0 = OFF, EBTR1 = OFF
#pragma config EBTRB = OFF

//////////////////////////Variables used////////////////////////////////////
int           max, min, moved, a, b, c, i, j, h, second[3], result[3];

unsigned char ch[3] = {0b00000011,0b00000111,0b00001011};

//////////////////////////LED assignments////////////////////////////////////
const int
              ONE    =      0b00000010,          // ON PORT C
              TWO    =      0b00000100,          // ON PORT C
              THREE   =   0b10000000,           // ON PORT B
              FOURA  =      0b00010000,          // ON PORT A (AND MUST USE WITH FOURC)
              FOURC         =      0b00010000,          // ON PORT C (AND MUST USE WITH FOURA)
              FIVE   =      0b11000000,          // ON PORT C
              SIX    =      0b00110000;          // ON PORT B

//////////////////Function prototypes for light up sequences///////////////
void ONESQ    (void);
void TWOSQ    (void);
void THREESQ(void);
void FOURSQ   (void);
void FIVESQ   (void);
void SIXSQ    (void);

//////////////////LED  light up sequence functions//////////////////////////
void ONESQ ()
       {
       PORTA=0;
       PORTB=0;
       PORTC=ONE;
       Delay10KTCYx(10);
       PORTC=0;
              for (h=0;h<3;h++)
                     {
                     Delay10KTCYx(3),
                     PORTC=ONE,
                     Delay10KTCYx(3),
                     PORTC=0,
                     Delay10KTCYx(3);
                     }
       PORTC=ONE;
       Delay10KTCYx(10);
       }

void TWOSQ ()
       {
       PORTA=0;
       PORTB=0;
       PORTC=TWO;
       Delay10KTCYx(10);
       PORTC=0;
              for (h=0;h<3;h++)
                     {
                     Delay10KTCYx(3),
                     PORTC=TWO,
```

```
                        Delay10KTCYx(3),
                        PORTC=0,
                        Delay10KTCYx(3);
                        }
        PORTC=TWO;
        Delay10KTCYx(10);
        }

void THREESQ ()
        {
        PORTA=0;
        PORTB=THREE;
        PORTC=0;
        Delay10KTCYx(10);
        PORTB=0;
                for (h=0;h<3;h++)
                        {
                        Delay10KTCYx(3),
                        PORTB=THREE,
                        Delay10KTCYx(3),
                        PORTB=0,
                        Delay10KTCYx(3);
                        }
        PORTB=THREE;
        Delay10KTCYx(10);
        }

void FOURSQ ()
        {
        PORTA=FOURA;
        PORTB=0;
        PORTC=FOURC;
        Delay10KTCYx(15);
                for (h=0;h<3;h++)
                {
                PORTA=FOURA;
                PORTC=0;
                Delay10KTCYx(3);
                PORTC=FOURC;
                PORTA=0;
                Delay10KTCYx(3);
                }
        PORTA=FOURA;
        PORTC=FOURC;
        Delay10KTCYx(10);
        }

void FIVESQ ()
        {
        PORTA=0;
        PORTB=0;
        PORTC=FIVE;
        Delay10KTCYx(10);
                for (h=0;h<3;h++)
                        {
                        PORTC=0b10000000,
                        Delay10KTCYx(3),
                        PORTC=0b01000000,
                        Delay10KTCYx(3),
                        PORTC=0,
```

```
                        Delay10KTCYx(3);
                        }
        PORTC=FIVE;
        Delay10KTCYx(10);
        }

void SIXSQ ()
        {
        PORTA=0;
        PORTB=SIX;
        PORTC=0;
        Delay10KTCYx(10);
                for (h=0;h<3;h++)
                        {
                        PORTB=0b00100000,
                        Delay10KTCYx(3),
                        PORTB=0b00010000,
                        Delay10KTCYx(3),
                        PORTB=0,
                        Delay10KTCYx(3);
                        }
        PORTB=SIX;
        Delay10KTCYx(10);
        }
////////////////////////////////ABSOLUTE VALUE FUNCTION//////////////////////
int abs(int val);

#pragma code
int abs(int val)
        {
                if (val<1)
                        {
                return val*(-1);
                        }
                else
                        return val;
        }

////////////////////////////Interrupt Service Routine declaration///////////
void isr (void);
#pragma interrupt isr

#pragma code high_vector = 0x08
void high_vector ()
                {
                _asm GOTO isr _endasm
                }

//////Interrupt Service Routine, it clears the flags, wakes up and reset ADC bits//
#pragma code
void isr (void)
                {
                PORTA=0b11111111;
                PORTB=0b11111111;
                PORTC=0b11111111;
                Delay10KTCYx(2);
                PORTA=0;
                PORTB=0;
                PORTC=0;
                INTCONbits.INT0IF=0;
```

```
        INTCON3bits.INT1IF=0;
              INTCON3bits.INT2IF=0;
              ANSELbits.ANS0=1;
              ANSELbits.ANS1=1;
              ANSELbits.ANS2=1;
        }

//////////////////////Program start////////////////////////////////////
#pragma code
void main ()
{

       // SETTING OSCILLATOR SPEED
              OSCCONbits.IRCF0 = 1;
              OSCCONbits.IRCF1 = 1;
              OSCCONbits.IRCF2 = 0;

/* SETTING 3 ANALOG PORTS AN0(RA0), AN1(RA1) AND AN2(RA2) AS INPUTS (1)
 THE REST OF THE A PORT,(RA3 THROUGH RA7) ALL THE B AND C PORTS AS OUTPUTS (0)*/
              TRISA = 0b00000111;
        ANSEL |=0b00000111;
              TRISB = 0b00000000;
              TRISC = 0b00000000;

//////////////////////POWER UP SEQUENCE////////////////////////////////

                    PORTC=0b00000001;    //LIGHT UP THE ONE
                    Delay10KTCYx(5);
                    PORTC=0;
                    PORTB=0b00010000;    //LIGHT ONE PORT FROM SIX
                    Delay10KTCYx(5);
                    PORTB=0b00100000;    //LIGHT ONE PORT FROM SIX
                    Delay10KTCYx(5);
                    PORTB=0b10000000;    //LIGHT ONE PORT FROM THE THREE
                    Delay10KTCYx(5);
                    PORTB=0;
                    PORTA=0b00010000;    //LIGHT ONE PORT FROM THE FOUR
                    Delay10KTCYx(5);
                    PORTA=0;
                    PORTC=0b00010000;    //LIGHT ONE PORT FROM THE FOUR
                    Delay10KTCYx(5);
                    PORTC=0b01000000;    //LIGHT ONE PORT FROM THE FIVE
                    Delay10KTCYx(5);
                    PORTC=0b10000000;    //LIGHT ONE PORT FROM THE FIVE
                    Delay10KTCYx(5);
                    PORTC=0b00000100;    //LIGHT ONE PORT FROM THE TWO
                    Delay10KTCYx(5);
                    PORTC=0;


//////////////// INFINITE LOOP TO PERFORM THE TASK CONTINUOUSLY/////////////
while (1)
       {
////////////////DO Loop to do while dice is still moving////////////////////
       do
              {
              moved=0;

////////////// A LOOP TO TAKE ANALOG SAMPLES FROM ALL 3 CHANNELS///////////
                    for (i=0;i<3;i++)
```

```
                        {
                        OpenADC (     ADC_FOSC_64  &
                        ADC_RIGHT_JUST &
                        ADC_12_TAD,
                        ch[i],        //Using unsigned char ch[3] to get the 3 channels
                        ADC_REF_VDD_VSS &
                        ADC_INT_OFF, 7);
              ADCON0=ch[i];
                        ADCON1=ADCON1 & 0b00000000;

                        Delay10TCYx(5);
                        ConvertADC();
                        while(BusyADC());
                        result[i]=ReadADC();
                        CloseADC();
                        }
//////////A 0.2 SECOND DELAY TO TEST FOR MOTION ON DICE////////////////////
                        Delay10KTCYx(5);
////////////// TAKING A SECOND READING FROM ALL 3 CHANNELS//////////////////
                  for (j=0;j<3;j++)
                        {
                        OpenADC (     ADC_FOSC_64  &
                        ADC_RIGHT_JUST &
                        ADC_12_TAD,
                        ch[j],
                        ADC_REF_VDD_VSS &
                        ADC_INT_OFF, 7);
              ADCON0=ch[j];
                        ADCON1=ADCON1 & 0b00000000;

                        Delay10TCYx(5);
                        ConvertADC();
                        while(BusyADC());
                        second[j]=ReadADC();
                        CloseADC();
                        }
//////////////To determine if dice is still moving//////////////////////////////////
                  if( (result[0]-second[0])*(result[0]-second[0])>50)
                        moved=moved+1;
                  if( (result[1]-second[1])*(result[1]-second[1])>50)
                        moved=moved+1;
                              if( (result[2]-second[2])*(result[2]-second[2])>50)
                        moved=moved+1;
            }
                  while(moved>0);

////////////// IF MOTION STOPPED, LIGHT UP THE LED'S (ON TOP ONLY)///////////////////////

                  max=544;          //Upper value limit for no acceleration detected.
                  min=430;          //Lower value limit for no acceleration detected.

//////////A 0.2 SECOND DELAY TO DETERMINE WHICH SIDE OF DICE IS UP////////////////////
                        Delay10KTCYx(5);

                  for (h=0;h<3;h++)
                        {
                        OpenADC (     ADC_FOSC_64  &
                        ADC_RIGHT_JUST &
                        ADC_12_TAD,
                        ch[h],
```

```
                        ADC_REF_VDD_VSS &
                        ADC_INT_OFF,7);
                ADCON0=ch[h];
                        ADCON1=ADCON1 & 0b00000000;

                        Delay10TCYx(5);
                        ConvertADC();
                        while(BusyADC());
                        result[h]=ReadADC();
                        CloseADC();
                        }

//////////////////////To determine which side is up//////////////////////////////////
                        a=abs(result[0]-512);           //  Z-axis
                        b=abs(result[1]-512);           //  Y-axis
                        c=abs(result[2]-512);           //  X-axis

                                                if (a>b)
                                                        {if (a>c)
                                                                {
                                                                if (result[0]<min)
                                                                    ONESQ();
                                                                if (result[0]>max)
                                                                    SIXSQ();
                                                                }
                                                        else
                                                                {
                                                                if (result[2]<min)
                                                                    TWOSQ();
                                                                if (result[2]>max)
                                                                    FIVESQ();


                                                                }
                                                        }
                                                else
                                                        {if (b>c)
                                                                {
                                                                if (result[1]<min)
                                                                    FOURSQ();
                                                                if (result[1]>max)
                                                                    THREESQ();
                                                                }
                                                        else
                                                                {
                                                                if (result[2]<min)
                                                                    TWOSQ();
                                                                if (result[2]>max)
                                                                    FIVESQ();

                                                                }
                                                        }

                        PORTA=0;                      // Turn off PORT A LED's
                        PORTB=0;                      // Turn off PORT B LED's
                        PORTC=0;                      // Turn off PORT C LED's

//SLEEP SEQUENCE. CONFIGURE INTERRUPTS, CLEAR FLAGS, ENABLE GLOBAL INTERRUPTS, DISABLE ANALOG
PORTS, GO TO SLEEP.//
                        INTCONbits.INT0IF=0;    // Clear Interrupt flag on INT0
```

```
                        INTCON3bits.INT1IF=0;       // Clear Interrupt flag on INT1
                        INTCON3bits.INT2IF=0;       // Clear Interrupt flag on INT2

                        INTCONbits.GIE=1;           // Global Interrupt Enable

                        INTCONbits.INT0IE=1;    // Enable external interrupt INT0
                        INTCON3bits.INT1IE=1;       // Enable external interrupt INT1
                        INTCON3bits.INT2IE=1;       // Enable external interrupt INT2

                        INTCON2bits.INTEDG0=0;      // Set up INT0 for falling edge trigger
                        INTCON2bits.INTEDG1=0;      // Set up INT1 for falling edge trigger
                        INTCON2bits.INTEDG2=0;      // Set up INT2 for falling  edge trigger

ANSELbits.ANS0=0;           // Disable AN0 analog port, to use as external interrupt trigger
ANSELbits.ANS1=0;           // Disable AN1 analog port, to use as external interrupt trigger
ANSELbits.ANS2=0;           // Disable AN2 analog port, to use as external interrupt trigger

                        Sleep();
      }
}
```

## Appendix D: Micro-controller code for version 4 dice

```
/*********************************************************************
/       Filename: main.c on TWI bit bang 156 folder
/       Description: LED dice with correct LED assignments
/                    This version is ready to be programmed into completed
/                    flex circuit and work correctly.
/
/       Author: Rudy Salas
/
/       Last modified: March 29, 2013
/*********************************************************************/

#include "TWI_Master.h"

//CREATE GLOBAL VARIABLES
       unsigned char data2 = 0x00;  //Set which register to read first
       unsigned char data3[3];      //Because we'll store 3 bytes read
       char X,Y,Z;                  //Set variables to store each axis data
       int A,B,C;                   //Create variables to use for flashing LED's

//TOGGLING FUNCTIONS
  void seq(int A,int B)
{
      PORTA=A;
      PORTB=B;
      __delay_cycles(1000000);
      PORTA=0;
      PORTB=0;
      __delay_cycles(1000000);
       for (C=0;C<5;C++)
       {
            PORTA=A;
            PORTB=B;
            __delay_cycles(1000000);
```

```
            PORTA=0;
            PORTB=0;
            __delay_cycles(1000000);
        }
}


// INTERRUPT SERVICE ROUTINE
  #pragma vector=INT0_vect

 __interrupt void ISR_INT0 (void)
{
//READ ORIENTATION FROM ACCELEROMETER
         while(!write_data(&data2,1));  //Indicate which register will be read
         while(!read_bytes(data3,3));   //Read 3 bytes from slave
              X = data3[0];             //Store each byte on its -
              Y = data3[1];             //- corresponding variable -
              Z = data3[2];             //- x,y or z.

              // CONDITIONS TO DETERMINE ORIENTATION
               if (X>17 && X<28)   //X positive. The 5 on top.
               { A = 48;
                 B = 0;
                 seq(A,B);
               }
               if (X>37 && X<47)   //X negative. The 2 on top.
                {A = 2;
                 B = 0;
                 seq(A,B);}
               if (Y>17 && Y<28)   //Y positive. The 3 on top.
                {A = 4;
                 B = 0;
                 seq(A,B);}
               if (Y>37 && Y<47)   //Y negative. The 4 on top.
                {A = 8;
                 B = 0;
                 seq(A,B);}
               if (Z>17 && Z<28)   //Z positive. The 6 on top.
                { A = 64;
                  B = 1;
                 seq(A,B);}
               if (Z>37 && Z<47)   //Z negative. The 1 on top.
                {A = 1;
                 B = 0;
                 seq(A,B);}
 //DISABLE INTERRUPTS
          GIFR=1;        //CLEAR INTERRUPT FLAG
          GIMSK =0x00;   //TURN OFF INTERRUPTS MASK
          SREG &=0xEF;   //DISABLE GLOBAL INTERRUPTS
}

int main(void)
{
//ACCELEROMETER CONFIGURATION THROUGH SERIAL COMMUNICATION. SET VARIABLE ARRAYS
    unsigned char mode[2] = {0x07,0x00};      //Set accelerometer on stand by mode to configure
    unsigned char intsu[2]= {0x06,0xE7};      //Orientation changes cause interrupt signal
    unsigned char rate[2] = {0x08,0xFD};      //Set debounce filter and Sample Rate.
    unsigned char data[2] = {0x07,0xD9};      //Set in active mode and active high int signal

//SETUP OSCILLATOR, SET DATA DIRECTION REGISTER AND INITIALIZE TWI
```

```
        CCP = 0xD8;        //Signature to be able to modify CLKPSR
        CLKPSR = 0x00;     //Set prescaler to divide 8MHz by 1
        DDRA = 0x7F;       //Set data direction register for port A as outputs
        DDRB &= 0xFB;      //Set INTO pin as input
        DDRB |= 0x01;     //Set data direction register for port PB0 as output
        twi_init();        //Initialize twi interface

//TRANSMIT ACCELEROMETER CONFIGURATION TO DEVICE
        while(!write_data(mode,2));             //Set standby mode
        while(!write_data(intsu,2));            //Set interrupt register
        while(!write_data(rate,2));             //Set sample rate
        while(!write_data(data,2));             //Set to ACTIVE mode

             while(1)
             {
               //SUPERFLUOUS READ TO SHUT DOWN INT SIGNAL FROM ACCELEROMETER BEFORE SLEEP
               while(!write_data(&data2,1));  //Indicate which register will be read
               while(!read_bytes(data3,3));   //Read 3 bytes from slave
               //SLEEP SEQUENCE
                  CCP = 0xD8;   //signature to be able to modify MCUCR
                  GIMSK=0x01;   //External Interrupt pin is enabled INT0
                  MCUCR = 193;  //Set INT0 to Rising Edge, Sleep mode (Idle) and Sleep Enable
                  GIFR = 1;     //Clear Interrupt flag
                  SREG |=0x80;  //Global interrupt enable
                    __sleep();  //Sleep instruction
             }
}
```

## Appendix E: TWI function header for ATTiny to assume master function

```
/* This file has been prepared for Doxygen automatic documentation generation.*/
/*! \file *********************************************************************
 *
 * \brief  Bit bang TWI master driver.
 *
 *      This file contains the function prototypes and enumerator definitions
 *      for various configuration parameters for the AVR TWI master driver.
 *
 *      The driver is not intended for size and/or speed critical code, since
 *      most functions are just a few lines of code, and the function call
 *      overhead would decrease code performance. The driver is intended for
 *      rapid prototyping and documentation purposes for getting started with
 *      the AVR TWI master.
 *
 *      For size and/or speed critical code, it is recommended to copy the
 *      function contents directly into your application instead of making
 *      a function call.
 *
 *
 * $Date: 2012-06-01 13:03:43 $  \n
 *
 * Copyright (c) 2012, Atmel Corporation All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions are met:
 *
 * 1. Redistributions of source code must retain the above copyright notice,
 * this list of conditions and the following disclaimer.
```

```
 *
 * 2. Redistributions in binary form must reproduce the above copyright notice,
 * this list of conditions and the following disclaimer in the documentation
 * and/or other materials provided with the distribution.
 *
 * 3. The name of ATMEL may not be used to endorse or promote products derived
 * from this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY ATMEL "AS IS" AND ANY EXPRESS OR IMPLIED
 * WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
 * MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE EXPRESSLY AND
 * SPECIFICALLY DISCLAIMED. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT,
 * INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
 * (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
 * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
 * ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
 * THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 *
 *  Author: gary.grewal
 ************************************************************************/

#ifndef TWI_MASTER_H_
#define TWI_MASTER_H_

#include "ioavr.h"
#include "inavr.h"

/*! \brief Definition of pin used as SCL. */
#define SCL PA7

/*! \brief Definition of pin used as SDA. */
#define SDA PB1

/*! \brief Definition of PORT used as SCL. */
#define PORT_SCL PORTA
/*! \brief Definition of DDR used as SCL. */
#define DDR_SCL DDRA
/*! \brief Definition of PIN used as SCL. */
#define PIN_SCL PINA

/*! \brief Definition of PORT used as SDA. */
#define PORT_SDA PORTB
/*! \brief Definition of DDR used as SDA. */
#define DDR_SDA DDRB
/*! \brief Definition of PIN used as SDA. */
#define PIN_SDA PINB

/*! \brief Slave 8 bit address (shifted). */
#define SLAVE_ADDRESS 0x98           //MMA7660 Accelerometer

#define READ_SDA() (PIN_SDA & (1 << SDA))
#define SET_SDA_OUT() DDR_SDA |= (1 << SDA)
#define SET_SDA_IN() DDR_SDA &= ~(1 << SDA)
#define READ_SCL() (PIN_SCL & (1 << SCL))?1:0

#define WRITE 0x0
#define READ 0x1

/*! \brief Delay used to generate clock */
```

```
#define DELAY 2

/*! \brief Delay used for STOP condition */
#define SCL_SDA_DELAY 1

void twi_disable();
void twi_init();
void toggle_scl();
void write_scl(char x);
char twi_start_cond(void);
char send_slave_address(unsigned char read);
char write_data(unsigned char* data, char bytes);
char i2c_write_byte(unsigned char byte);
char read_bytes(unsigned char* data, char bytes);
char i2c_read_byte(unsigned char* data, unsigned char bytes, unsigned char index);
void write_sda( char x);



#endif /* TWI_MASTER_H_ */
```

## Appendix F: TWI function prototype for ATTiny to assume master function

```
/* This file has been prepared for Doxygen automatic documentation generation.*/
/*! \file *********************************************************************
 *
 * \brief  Bit bang TWI master driver.
 *  TWI_master.c
 *
 *      This file contains the function prototypes and enumerator definitions
 *      for various configuration parameters for the AVR TWI master driver.
 *
 *      The driver is not intended for size and/or speed critical code, since
 *      most functions are just a few lines of code, and the function call
 *      overhead would decrease code performance. The driver is intended for
 *      rapid prototyping and documentation purposes for getting started with
 *      the AVR TWI master.
 *
 *      For size and/or speed critical code, it is recommended to copy the
 *      function contents directly into your application instead of making
 *      a function call.
 *
 *
 * $Date: 2012-06-01 13:03:43 $  \n
 *
 * Copyright (c) 2012, Atmel Corporation All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions are met:
 *
 * 1. Redistributions of source code must retain the above copyright notice,
 * this list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above copyright notice,
 * this list of conditions and the following disclaimer in the documentation
 * and/or other materials provided with the distribution.
 *
```

```
 * 3. The name of ATMEL may not be used to endorse or promote products derived
 * from this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY ATMEL "AS IS" AND ANY EXPRESS OR IMPLIED
 * WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
 * MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE EXPRESSLY AND
 * SPECIFICALLY DISCLAIMED. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT,
 * INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
 * (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
 * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
 * ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
 * THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 *
 *  Author: gary.grewal
 ****************************************************************************/
#include "TWI_master.h"

/*! \brief initialize twi master mode
 */
void twi_init()
{
	DDR_SCL |= (1 << SCL);
	DDR_SDA |= (1 << SDA);

	write_sda(1);
	write_scl(1);

}

/*! \brief disables twi master mode
 */
void twi_disable()
{
	DDR_SCL &= ~(1 << SCL);
	DDR_SDA &= ~(1 << SDA);

}

/*! \brief Sends start condition */
char twi_start_cond(void)
{
	 write_sda(0);
	__delay_cycles(DELAY);

	write_scl(0);
	__delay_cycles(DELAY);
	return 1;

}

/*! \brief Sends slave address */
char send_slave_address(unsigned char read)
{
	return i2c_write_byte(SLAVE_ADDRESS | read );
}

/*! \brief Writes data from buffer.
    \param indata Pointer to data buffer
    \param bytes  Number of bytes to transfer
```

```
    \return 1 if successful, otherwise 0
 */

char write_data(unsigned char* indata, char bytes)
{
       unsigned char index, ack = 0;

       if(!twi_start_cond())
              return 0;
       if(!send_slave_address(WRITE))
              return 0;

       for(index = 0; index < bytes; index++)
       {
              ack = i2c_write_byte(indata[index]);
              if(!ack)
                    break;
       }
       //put stop here
       write_scl(1);
       __delay_cycles(SCL_SDA_DELAY);
       write_sda(1);
       return ack;

}


/*! \brief Writes a byte on TWI.
    \param byte Data
    \return 1 if successful, otherwise 0
 */
char i2c_write_byte(unsigned char byte)
{
        char bit;
       for (bit = 0; bit < 8; bit++)
       {
            write_sda((byte & 0x80) != 0);
            toggle_scl();//goes high
            __delay_cycles(DELAY);
            toggle_scl();//goes low
            byte <<= 1;
            __delay_cycles(DELAY);
        }
       //release SDA
       SET_SDA_IN();
       toggle_scl(); //goes high for the 9th clock
       //Check for acknowledgment
       if(READ_SDA())
       {
              return 0;
       }
       __delay_cycles(DELAY);
       //Pull SCL low
       toggle_scl(); //end of byte with acknowledgment.
       //take SDA
       SET_SDA_OUT();
       __delay_cycles(DELAY);
       return 1;

}
```

```
/*! \brief Reads data into buffer.
    \param data Pointer to data buffer
    \param bytes  Number of bytes to read
    \return 1 if successful, otherwise 0
 */
char read_bytes(unsigned char* data, char bytes)
{
       unsigned char index,success = 0;
       if(!twi_start_cond())
              return 0;
       if(!send_slave_address(READ))
              return 0;
       for(index = 0; index < bytes; index++)
       {
              success = i2c_read_byte(data, bytes, index);
              if(!success)
                     break;
       }
       //put stop here
       write_scl(1);
       __delay_cycles(SCL_SDA_DELAY);
       write_sda(1);
       return success;
       }

/*! \brief Reads one byte into buffer.
    \param rcvdata Pointer to data buffer
    \param bytes  Number of bytes to read
    \param index Position of the incoming byte in hte receive buffer
    \return 1 if successful, otherwise 0
 */
char i2c_read_byte(unsigned char* rcvdata, unsigned char bytes, unsigned char index)
{
        unsigned char byte = 0;
       unsigned char bit = 0;
       //release SDA
       SET_SDA_IN();
       for (bit = 0; bit < 8; bit++)
       {
              toggle_scl();//goes high
              if(READ_SDA())
                     byte|= (1 << (7- bit));
              __delay_cycles(DELAY);
              toggle_scl();//goes low
              __delay_cycles(DELAY);
        }
       rcvdata[index] = byte;
       //take SDA
       SET_SDA_OUT();
       if(index < (bytes-1))
       {
              write_sda(0);
              toggle_scl(); //goes high for the 9th clock
              __delay_cycles(DELAY);
              //Pull SCL low
              toggle_scl(); //end of byte with acknowledgment.
              //release SDA
              write_sda(1);
              __delay_cycles(DELAY);
       }
```

```
       else //send NACK on the last byte
       {
              write_sda(1);
              toggle_scl(); //goes high for the 9th clock
              __delay_cycles(DELAY);
              //Pull SCL low
              toggle_scl(); //end of byte with acknowledgment.
              //release SDA
              __delay_cycles(DELAY);
       }
       return 1;

}
/*! \brief Writes SCL.
    \param x tristates SCL when x = 1, other wise 0
 */
void write_scl (char x)
{
      if(x)
      {
            DDR_SCL &= ~(1 << SCL); //tristate it
           //check clock stretching

           while(!READ_SCL());

      }
      else
      {
            DDR_SCL |= (1 << SCL); //output
            PORT_SCL &= ~(1 << SCL); //set it low

      }
}

/*! \brief Writes SDA.
    \param x tristates SDA when x = 1, other wise 0
 */
void write_sda (char x)
{
      if(x)
      {
            DDR_SDA &= ~(1 << SDA); //tristate it

      }
      else
      {
            DDR_SDA |= (1 << SDA); //output
            PORT_SDA &= ~(1 << SDA); //set it low

      }

}
/*! \brief Toggles SCL.
 */
void toggle_scl()
{
       if(PIN_SCL & (1<<SCL))
       {
              DDR_SCL |= (1 << SCL); //output
              PORT_SCL &= ~(1 << SCL); //set it low
```

```
        }
        else
        {
                DDR_SCL &= ~(1 << SCL); //tristate it
                while(!READ_SCL());
        }
}
```

# Curriculum Vita

Rodolfo Salas was born on December of 1970 in Juarez, Mexico. The 7th of 8 children for Alfonso and Angela Salas who, while lacking education beyond elementary school understood its transformative value and instilled it in their children. Rodolfo attended high school at CBTis#114 a combination high school and vocational school being awarded an Associate's Degree. Rodolfo attended college at the Technological Institute in Juarez (ITCJ) graduating in 1994 with a Bachelor's in Electrical Engineering. After a brief stay with Osram Sylvania as Product Engineer, Rodolfo acquired legal resident status and moved to El Paso where he worked at small local businesses in Quality, Supervisory and Technical roles with increasing levels of responsibility. In 2008 Rodolfo was working for Federal Mogul Inc. but due to the economic downturn the Automotive Industry downsized significantly and Rodolfo was one of the many who were left unemployed. Having this been his second time experiencing a layoff, Rodolfo took this as an opportunity to improve his career prospects, returned to school seeking a graduate degree in Electrical Engineering.

Permanent address: 11773 Corona Crest Ave.
El Paso, TX. 79936

This thesis was typed by Rodolfo Salas.