ADAPTIVE H.I.D.

https://github.com/ctag/cpe495

http://cpe-design.uah.edu/2016/team1/index.html

CPE 496 Final Report

Christopher Bero
Bryant Johnson
Michael Baldwin
John Gould

SPRING 2016

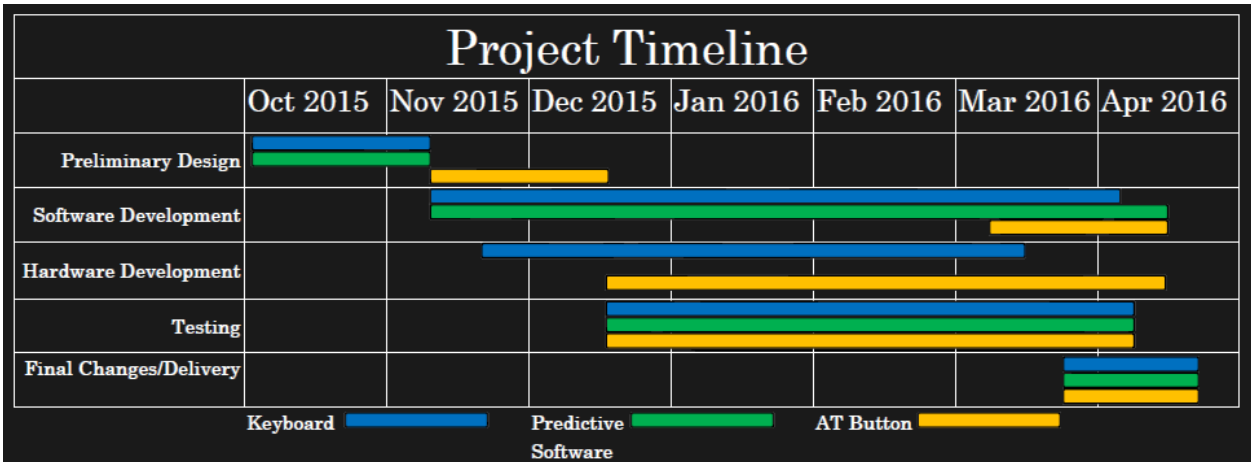
UNIVERSITY OF ALABAMA IN HUNTSVILLE

301 Sparkman Dr. NW, Huntsville, Alabama 35899

Table of Contents

Project Summary

Design a set of adaptive interface devices for modern computers with readily available consumer machinery and products. The product will have a focus on affordability, iterative design, and easy operation. Construct and test the devices for viability in general use; then generate a case study on the effectiveness of this approach to bringing adaptive technology to a larger audience.



Team Description

The Adaptive H.I.D. project was envisioned to develop affordable computer interface devices and bring adaptive technology to more people around the world. Each team member was selected based on how their personal expertise would benefit the project.

Chris Bero is the Hardware Lead for the project with the subtask of community outreach. He originally conceptualized the project after noticing the many shortfalls of adaptive technology in his free time and drafted multiple designs that could benefit the disabled community. He is very knowledgeable in microcontrollers, 3D printers, and collaborative software development.

Michael Baldwin is the Software Lead for the project. He leads the pursuit in either upgrading an existing package or creating our own custom keyboard interface which uses predictive text to lessen the number of characters required to complete a word. Michael has a very strong programming background and implements professional modern style when developing code.

Bryant Johnson is the Testing Lead with the subtask of technical writer. He is responsible for creating the unit tests and maintaining the documentation of the entire project. He excels in strategizing the project and rationalizing each process while paying attention to fine details. His interest includes cybersecurity and reverse engineering.

John Gould is the Systems Lead for this project. He is responsible for integration of the hardware and software that will be developed for this project and will ensure that the mechanical elements will work together in the finished product. His strengths include microcontroller based projects and software development.

Design Problem

Our team is researching and building Adaptive Technology human interface devices to help people living with disabilities interact with computers. Adaptive technology (AT) is a device or component which is specifically designed for persons with disabilities. Human Interface Devices (H.I.D) is a method by which a human interacts with an electronic information system either by input or output. Computers have dropped in price, grown in capability, and leapt into consumer markets in the past few years. Adaptive Technologies which help persons living with disability interface with a computer have been left behind in this regard. Solutions available are expensive and difficult to obtain. There is a distinct need for human interface devices which meet many AT requirements and are available to wider audiences for more reasonable cost. This project is going to be a student lead and noncommercial in nature. We wish to make clear that our design decision to present the project as open source incurs additional work for us to have a clean operating environment that others can approach and utilize in separate projects, not a sidestepping of features we do not intend to implement. We are developing two interfaces that include a Sliding Keyboard, and a AT Button. The sliding keyboard will require no finger movement to type characters, instead utilizing shoulder and elbow ranges. Our AT Button was created as a result from our customer requesting us to investigate the idea. We are interfacing with local disability service groups United Cerebral Palsy (UCP) of Huntsville Technology Assistance for Special Consumers (TASC)

We established our marketing and engineering requirements based upon UNICEF's World Report on Disability. UNICEF states, "Manufacturing or assembling [AT] products locally, using local materials, can reduce cost and ensure that devices are suitable for the context." Priority of the design was for cheap and readily available materials and techniques—allowing for the product to be easy to build and modify. Consumers should be able to with relative ease assemble the device themselves. Additionally, we wanted to build a product with equivalent or better functionality without increasing complexity of user interaction. The design should be durable and adhere to the standards of current AT marketable solutions. The Marketing Requirements are as follows:

- Open Source Design
  - Require only items available to common makerspaces
  - Easy to iterate and customize
  - Low cost requirements for parts and machinery
- Easy to Operate
  - Must provide as good or better functionality than existing solutions
- Reliable
  - Must accept improper input and have a robust mechanical design.

The Engineering Requirements were based upon current mechanical keyboard design standards. We needed the hardware to react to changes under 6 milliseconds while in use. Each switch will need to last at least 10 Million cycles while supporting at least 30 words per minute operation at 90% accuracy. The device should be operational from -10 Degrees Celsius up to 70 Degrees

Celsius. These requirements are typically found on the product speciation section of modern mechanical keyboards by various brands on the marketplace. Research was conducted by browsing and comparing specifications of listed items in retail stores.

Background Information

There are many shortfalls when it comes to AT. The main issue is that there is high development costs to a low volume market. Existing products include a Large Switch Keyboard which is aimed towards people with motor disabilities and lack of precision. It is less costly but does not necessarily solve the problem just lessens the possibility of error. The Orbitouch by Blue Orb Inc. is the main system that we based our design from. The keyboard retails for $450 with eight sections for character selection and the ability to simulate a mouse.

Figure 1 Http://orbitouch.org/

In lieu of any legal advice available to us, we read all of Blue Orb Inc’s listed patents based on the design. Our research found that the patents covered the mechanical design of the internal mechanisms of the keyboard. We moved our design approach to have our AT keyboard use the key map of Orbitouch.

Our team began to pursue creating the AT Button after our first keyboard design review with UCP TASC. TASC specifically asked us to investigate the creation of AT Buttons because of their current costs and need. Currently, AT Buttons are sold in the range of $60 - $160 and are out of the budget of many families coping with disabilities.

Figure 2 http://www.boundlessat.com/Switches/Wired-Switches/Big-Buddy-Buttons

We wanted to create a solution that was able to be made easily by home consumers by using a 3D printer.

The most significant differences between competing products versus our product is the low price and iterative design. We strive to create and publish an open source product that will allow people develop custom, low cost AT wherever a makerspace is available.

Trade-off Analysis of Design Alternatives

Various alternative products were considered in the design of this project. We have reviewed the possibility of creating four different systems to be part of our suite of low cost AT devices that we will deliver as the outcome of our project. With our trade-off analysis we investigate the options available for our project and the design decisions which lead to our chosen systems.

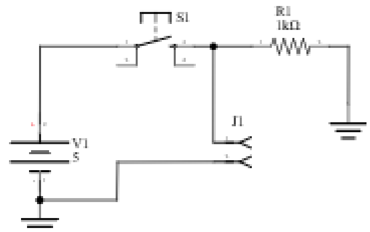
Products

Within the project of creating AT HID, our team generates several product ideas and needs to decide which are worth pursuing and how to go about designing the results.

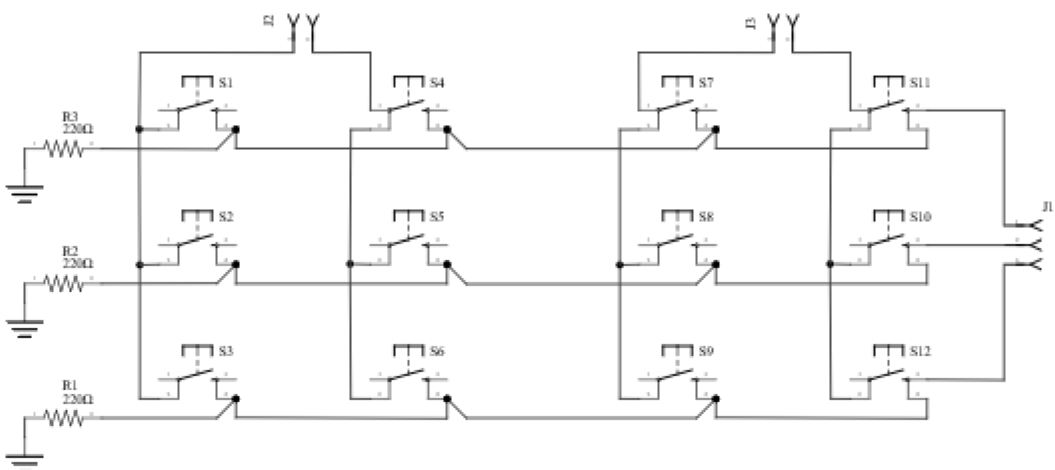
Sliding keyboard. Our first product is a keyboard to limit the required dexterity of typing. Targeted at users with conditions commonly associated with amyotrophic lateral sclerosis or cerebral palsy, this keyboard removes keys from the user's path and instead places two horizontal "pucks" down as control surfaces. Pucks can be a custom design from a handle to a joystick, anything that the user can operate to slide the base of the puck around an inset octagon that it resides on. At each corner of the octagon are mechanical switches, totaling sixteen between the two pucks. If each puck is moved to contact a switch, the full range of the QWERTY keyboard layout can be represented.

AT Button. While our customer expressed general interest in the above solutions, we also learned a lot about the current state of AT in use at UCP and received enough direction to build out our case study plan for more AT. One of the more common systems for actuation in use at UCP are SPST buttons -- roughly four-inch diameter pads -- that terminate to a 3.5mm mono audio pole. These buttons are then plugged into a variety of host items in order to provide a greater ease of access to the item. At UCP a conflict has occurred where the market price for these buttons is high -- in the $60 to $80 range -- and their use level is low -- an example is sub $20 toy adapting. We suspect this is caused by the quality of the buttons, if the button is associated with a device to request food or water, then it *must* work each and every time. Toy adapting isn't quite as mission critical though, and a solution which provides "button to 3.5mm jack" at a lower price would be ideal. Thus we intend to investigate using mechanical keyboard switches with thermoplastic or acrylic cases to create more affordable actuation devices and test them as part of our case study.

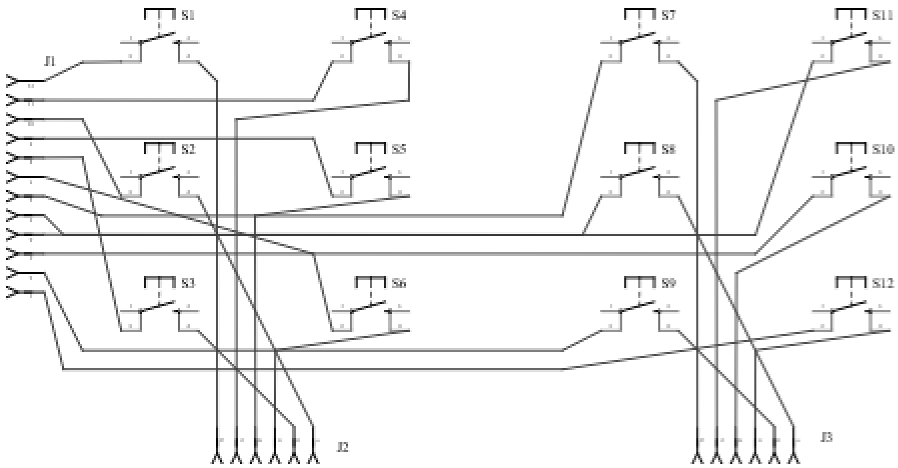
Switch Circuits. For switches, we will use port inputs on either the MSP430 or ATmega and arrange switches with pulldown resistors.



Our plan is to implement a scanning grid for the switches as used in CPE322 for FPGAs.



For testing our firmware and hardware approach, we constructed a set of twelve isolated switches which can be wired into breadboards. The tradeoff here is ease of configuration in exchange for a lot of tidiness.



Material breakdown for sliding keyboards

We're following Design Analysis from the University of Delaware's *Nonprofit Management Certificate Course*.

For our design decisions, we hold the sliding keyboard to have similar enough hardware qualities as to occupy the same space of design approaches. Hardware approaches were based on components that we have had previous experience with, could easily source, and knew to have high availability in common makerspaces (PLA printing, 30W laser CNC bed).

Design Scenario A

- MSP430F5529 processor
- Cherry Corp. MX switches
- Thermoplastic case

Design Scenario B

- MSP430F5529 processor
- Gatereon switches
- Laser cut case

Design Scenario C

- ATmega328p processor
- Cherry Corp. MX switches
- Thermoplastic case

Design Scenario D

- ATmega328p processor
- Gatereon switches
- Laser cut case

Each approach's set of decision criteria is given an importance magnitude 1-5 and instance rating 1-5. The resulting grade (= importance * rating) is assigned and reviewed alongside functional or behavioral decompositions.

Scenario A - Decision Criteria	**Importance**	**Rating**	**Grade**
Production Cost	5	3	15
Component Availability	4	2	8
Assembly Skill/Time	4	3	12
System Durability	3	4	12
Design Complexity	2	3	6
Grade total			53

Scenario B - Decision Criteria	**Importance**	**Rating**	**Grade**
Production Cost	5	5	25
Component Availability	4	4	16
Assembly Skill/Time	4	4	16
System Durability	3	2	6
Design Complexity	2	4	8
Grade total			71

Scenario C - Decision Criteria	**Importance**	**Rating**	**Grade**
Production Cost	5	2	10
Component Availability	4	3	12
Assembly Skill/Time	4	4	16
System Durability	3	4	12
Design Complexity	2	4	8
Grade total			58

Scenario D - Decision Criteria	**Importance**	**Rating**	**Grade**
Production Cost	5	4	20
Component Availability	4	5	20
Assembly Skill/Time	4	5	20
System Durability	3	2	6
Design Complexity	2	4	8
Grade total			74

Analysis factors

- Cherry Corp. products are often difficult to source, are more expensive than Gatereon compliments.
- Grading thermoplastic and laser laminate cases is difficult due to the vast plethora of available systems and operator knowledge. We settled on as unbiased a format as possible, with plastic cases offering better system durability and longevity with a higher complexity and design cost.
- The two microcontrollers compared here were chosen either for their popularity in crowd-designed keyboards (such as the Kiibohd Infinity) or their use in previous UAH courses (CPE323).
- We tried to favor availability and assembly factors heavily in this stage, as a durable and cheap system is unhelpful if nobody is willing or capable to generate them for people in need.

Although the Atmel/Gatereon solution provided the best score by our criteria, our team did not have a “will fetch a good grade” section. This unfortunately ended up being the deciding factor, and we chose to investigate the msp430/cherry/thermoplastic option as it used microcontrollers presented in a previous UAH class and promised a higher level of complexity. We wonder how often this happens in the real world.

Design information & Test Scenarios

Keyboard - Base materials

The team wanted to ensure that the keyboard provided a proper balance between being comprised of easy to source materials, designed to be easily assembled, and still highly function to the end user. We chose sheet-based materials in order to provide a wider array of assembly options. A 3-

5mm thick sheet of material can be laser cut, routed on a CNC mill, or manually machined with a drill press, jigsaw, bandsaw, or coping saw. 3D printed components were utilized in such a way that they could be replaced by additional segments of sheet material glued into a 14mm square mount.

We tried a small grouping of woods and plywoods with our preferred method of prototyping: laser cut panels. Sanded plywood (3mm) available at local home improvement stores provided excellent results, especially when sealed with clear waterproofing. Warping did prove to have some detrimental effects, such that we suggest inspecting sheets before purchase, and then storing them in a cool, humidity controlled location. For a more industrial end product, we additionally chose 3-5mm acrylic sheets. We suspect these acrylic sheets entail more difficulty for CNC and hand crafted approaches, but they produce fantastic results on a laser cutter. While being subjected to the laser, acrylic will sublimate into a toxic [1] fume, which necessitated proper ventilation and operator procedures. During our use of the laser cutter, a two-stage external fan and vent system was kept running, and completed cuts were allowed to vent for several seconds before the laser enclosure was opened.

Other materials, such as the PVC pipe cap and springs, were selected as a "basic approach" in which we challenged ourselves to find local material sources. These materials served functional purposes well, though we support the notion that more tuned and ergonomic solutions exist. We believe that these improvements, such as 3D printed paddles, can be adjusted and implemented in the assembly process with minimal deviation from the overall design. The 4 inch PVC pipe caps are durable and can be easily machined and then used as sliders on both plywood and acrylic. Also, the PVC cap diameter falls close to the target maximum 3D printing footprint, so printing custom and ergonomic paddles is supported without modification to the keyboard's structure.

One exception made was for an online order of M3 bolts. The keyboard can, with relative ease, be adjusted to use lasercut clips for joining the panels, but we desired more rigidity in the construction, as well as the ability to tear down the keyboard quickly without fear of breaking the pins.

Keyboard - Microcontroller Hardware

We began the class by taking a look at microcontroller based USB clients. On either brand's side we found solutions for emulating USB protocols in firmware or hardware level systems provided by the manufacturer. ATmel provides USB testing boards based on the AT90USB1287, which has full speed USB one-the-go (OTG) support [2]. Originally, the team intended to fabricate a custom PCB to mount either an MSP430F5529 LQFP IC or ATmega328P DIP package. After looking into online PCB manufacturing, we decided that the involved complexity, wait time, and cost were great enough that just mounting the default experimenter hardware into the keyboard produced more favorable results. To this end, the keyboard baseplate includes mounting points for a Ti MSP430F5529 Launchpad PCB. We wanted to investigate using V-USB [3] with an ATmega328 based Arduino, but eschewed it due to time constraints.

Keyboard - Microcontroller Firmware

The discussion about which microcontroller package to use was occasionally a heated one. ATmel chips are easily the most well-known and implemented controllers for hobby level projects all over the world. But we had just spent a semester learning about the MSP430 architecture in CPE323, and wanted to implement something from it. The decision eventually came down to the results obtained from the section on microcontroller hardware, where the firmware level implementation on AVR, and hardware level versions of the MSP430 were the best candidates. Hardware USB support on the MSP430F5529 Experimenter Board was more favorable -- and we hoped more robust as well -- for delivery to an end-user than a firmware approach with ATmegas.

We were shocked by the loss of familiarity in developing MSP430 firmware. Texas Instruments now pushes a package called 'driverlib' heavily, in what appears to be an attempt to compete with the popularity of Arduino's IDE. What results is less of a departure from formal C than the Processing-like code available in Arduino's IDE, but it's even more estranged and has a huge lack of clean programming interface left behind. In CPE323 I (writing as Chris Bero) enjoyed working with the control registeres and bitmasks. With Ti's driverlib, that has all been replaced by a highly opaque suite of functions, which all have horrendously distended naming schemes such as:

GPIO_setAsInputPinWithPullDownResistor(LEFT_BUTTONS_PORT,

LEFT_BUTTONS_PINS, GPIO_LOW_TO_HIGH_TRANSITION)

To properly interact with the hardware level USB support on the MSP430F5529, driverlib is required. This had us double-guessing the design decision, and by the end of the CPE496 semester, we had constructed a test-circuit for implementing HID USB from the V-USB package with an ATmega328 [4].

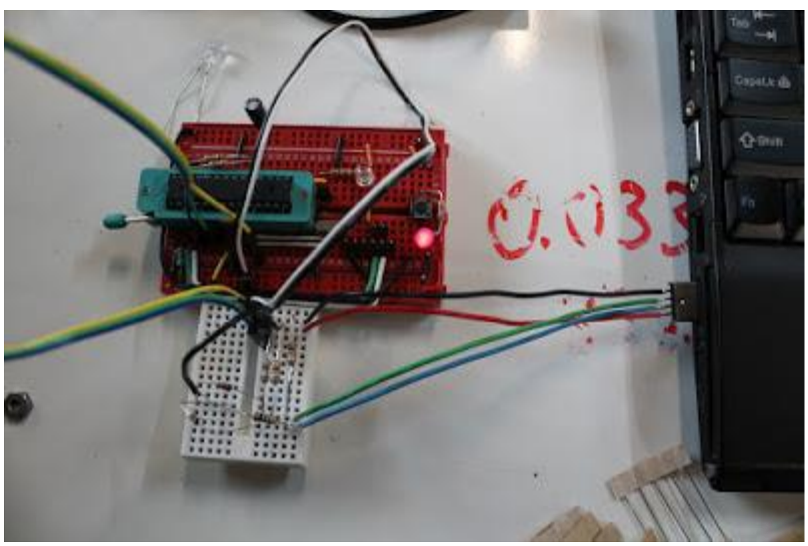


Figure 3 Working ATmel USB support

3D Printer - Base System

As a prospective freshman, I (Chris Bero) was walked through the highbay machine shop in Tech Hall. During the tour, we were told that as engineering students we would have access to this facility and others like it around campus. Previously I had rejected internship offers at UAH, because I didn't know if I could commit to enrolling. After that day touring campus I decided to become a student here. By the time I found myself signing up for CPE495 I had interacted with the machine shop manager several times, and fully understood that it was off limits except for a single semester if I took the appropriate machining class. Even then only projects for that class were allowed. Hurt feelings and mistrust aside, we entered CPE495 with the knowledge that we'd have to source the manufacturing equipment ourselves.

The team chose a Printrbot Simple Metal due to its price point, mechanical construction, and the background of Printrbot as a company that works with open source projects and contributes to the RepRap community [6]. We purchased the printer and assembled it at Makers Local 256, where we had access to staging space for printing close to campus.

3D Printer - Modifications

We quickly set about adjusting and building on the printer to create a machine capable of handling the proposed manufacturing and prototyping. The printer was paired with a Raspberry Pi which ran a website and camera for remote operation, along with an ATmega328 to control various elements such as lighting, fans, environment sensors, and heated build plates. We drafted our own filament guide design [7] and later pivoted to a PVC scaffolding which held filament reels above the printer to provide less chance of the filament breaking. The PVC later served as a structure to hold LED lighting and an area fan for the printer as well. While performing build plate material tests, we added a heated platform to the build plate, which was controlled in gcode.

The hotend was exchanged for a j-head MK-V [8]. The j-head is almost an entire inch shorter than the default Ubis hotend, and required modifications to the printer to be compatible. We initially tried printing an adapter socket which would mount in the extruder and then hold the j-head as low as the Ubis, but this broke twice while printing. We abandoned the adapter in favor of modifying the build plate, which was lifted on springs to meet the shorter hotend.

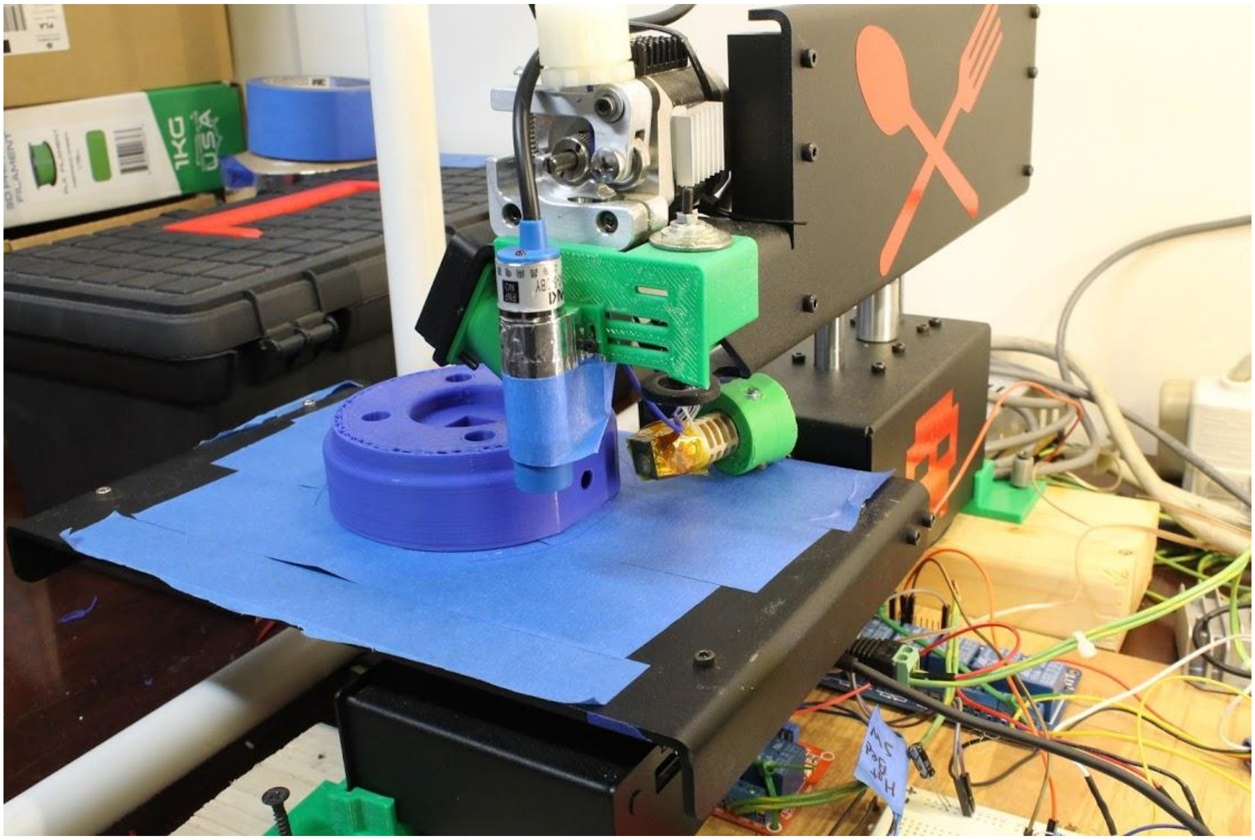


Figure 4 Replacement Inductive Sensor

During build plate material testing, we encountered a problem where the inductive Z-probe could not “see” the aluminum plate with our borosilicate glass plate mounted above it. To address this we purchased a more powerful inductive probe and designed an external harness to mount it on the extruder assembly.

The fan which came with the Simple Metal kit wore out partway through the CPE496 semester, and we replaced it with a more powerful blower fan, incidentally improving the quality of surface finishes on printed parts.

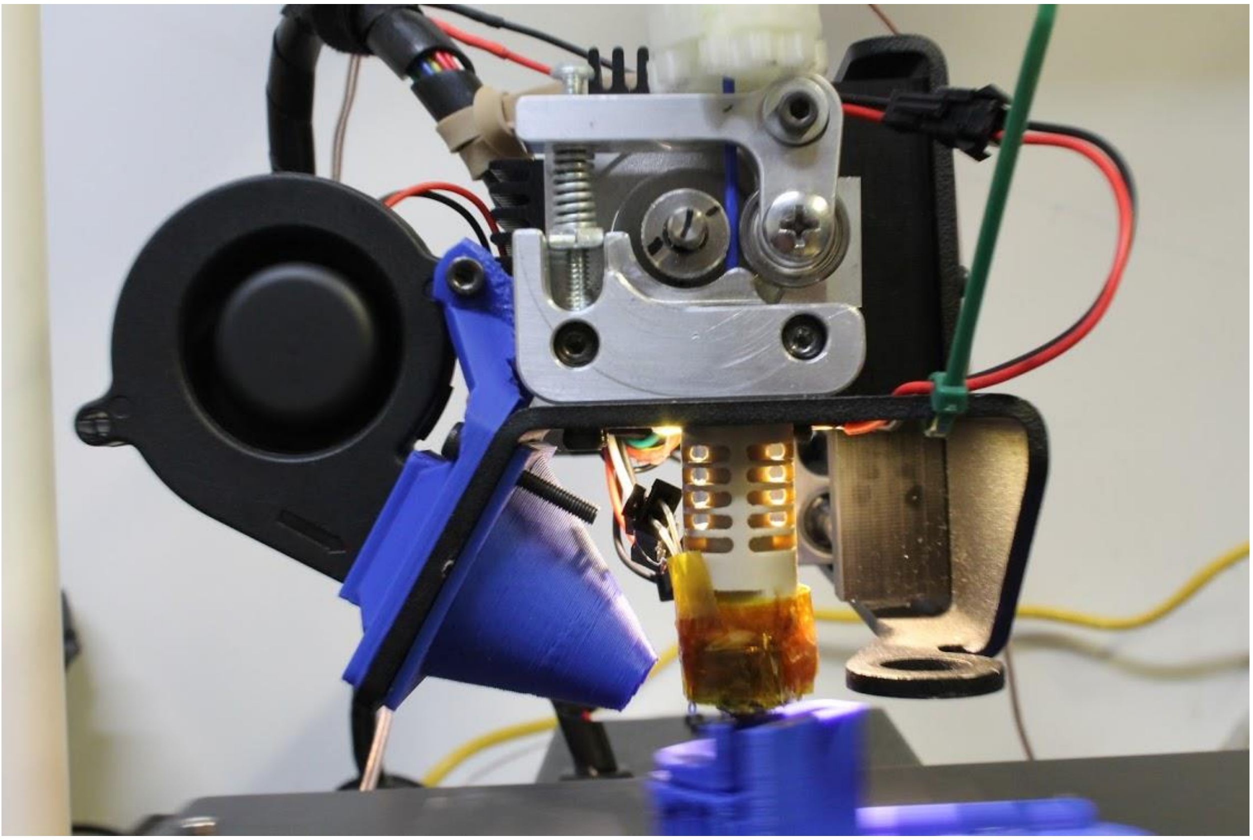


Figure 5 Blower fan addition

3D Printer - Build Plate Materials

We tested an assortment of materials based on discussions and posts available from RepRap's wiki and forums.

Aluminum

We tried printing directly to an aluminum plate. No plastic adhesion for either cool or heated build plate.

Painters Tape

"Blue Tape" worked quite well on a cool print surface, but would not stick to heated aluminum. Parts adhered very well to the tape, but the stress of curling would usually break the tape from the build plate, causing a warped print and more time for cleanup and re-application of the tape.

Kapton Tape

We had varied results with kapton tape. For the most part our results were similar to that of the plain aluminum, with the first layer of a part not adhering. Additionally, applying kapton tape to the build plate was by far the most intensive effort, as special care had to be given to avoid air bubbles trapped underneath.

Polyetherimide (PEI) with 3M 468MP

PEI also delivered mixed results. We had some fantastic prints, where the part would adhere to a heated build plate with this material, and then almost pop free on its own after cooling. Unfortunately we couldn't tie down consistent prints and had to move on.

Polyetherimide/Fiberglass Laminate

We purchased this in the form of "Buildtak", a commercial product. This material was very tricky to properly tune, with the initial layer of a part printed at precisely 0.2mm at 210 degrees C and 150% over extrusion. Once set up though, the Buildtak began delivering excellent results with minimal overhead. At one point we had a prototype AT button stick to the print plate so well that it could only be removed by cutting the Buildtak free. We chose this material as our best option.

Borosilicate Glass

We tried many, many different forms of this approach. Glass with blue tape, glass with PEI, glass with Polyvinylpyrrolidone (PVP) glue painted onto it, glass with PVA glue painted onto it, glass with sandpaper glued to it, glass with kapton tape, and plain glass. Some worked better than others, and a few of these we would consider re-investigating with a manually leveled print bed (no inductive sensor). Ultimately none delivered consistent results.

3D Printer - Serialization

During the course of this class, we investigated the design of a serialized 3D printer. Current home-use printers require fairly involved human interaction for upkeep and operation overhead. In order to maximize the utility of our printer we wished for it to print consecutively both day and night. Conservative estimations would put the printer at 3 printed items per day as opposed to the 1 printed part per day we achieved with full operator involvement.

Our first approach was to design a custom Aluminum build plate which would include an actuated lever to detach finished parts. This design worked well initially, but a design flaw where the heated aluminum would warp due to the channel of missing material caused us to abandon it.

Next we constructed a print enclosure by which printed parts would be removed through the application of thermal expansion. We settled on Borosilicate glass as a fair candidate due to the glass's tendency to stretch when heated. The heated build plate was modified to run over-spec at 480 Watts, and we affixed the glass above it. The process worked correctly: after printing a part, the heater would be cycled on and off several times over the course of two hours, which repeated stretching and contracting of the glass build material would cause printed parts to delaminate from it and they could then be easily knocked to the side, and another print could start. Unfortunately, we discovered toward the end of our investigation that unfavorable results (inconsistency of the first layers to a print) was likely due to a slight convex warp to the glass plate, where it curved 0.3mm from the edges to the center, which ruined the calibration of prints requiring a first layer of 0.2mm.

In conclusion to our serialization exercise, we find it to be a feasible end-goal at relatively low cost and simple assembly.

AT Button - Design Aspects

We began working on the AT Button late in the class, and tried to account for the lack of time by designing to a useable first prototype. Our basis for form factor came from looking at images of available products and guessing at their size and structure. We then paired that with the information that the typical small print envelope size for a reprap is 120mm, and settled on a 100mm diameter button base. This was stretching the limits of our printing capabilities, so we paired additional work on the design of the part’s first few layers against the previous investigation into print plate materials.

Predictive Text Application

The application takes input from the user using a textbox with a listener attached. This serves the purpose of listening for keystrokes. When a key is pressed, the application looks for the text the user is currently typing and presumes it is a partially completed word. The program then takes this word and asks a SQLite database for the 5 most likely completions to this word. The selected candidates are chosen based on frequency from the user, as well as alphabetical proximity to the letters already input so far.

The program consists of two primary objects: the listener is in the main object, while sql queries are handled by another object. This object is run on a child thread, allowing the user to continue using the application even while the sql object is waiting for a reply from the database. Within the main object, once the word to be completed is obtained, it is forwarded to the sql object and waits for it to reply with the five results. Once the potential candidates are calculated, each answer is posted to a button for the user to choose from. The sql object receives the partially completed word and stores it in a variable 'arg' and runs the following query: "SELECT WORD FROM dictionary

WHERE WORD LIKE '" + arg + "%' ORDER BY COUNT DESC LIMIT 5;". The result is then parsed into an array of 5 strings and is sent to the main object.

The database consists of two columns: WORD and COUNT. The WORD column contains each word that can be recommended to the user, while the COUNT column determines which word is used more often than another. More specifically COUNT contains how many times that word has been chosen as the intended word and the word with the highest COUNT is considered to be the most likely to be used. Using this algorithm, the application will begin to learn what words really are more common for each user and will become more and more useful. The database is populated with the English dictionary and COUNT is initiated using various online articles so even on initial launch the application has a general idea of what words a random sampling of users prefers as their most used words.

Cost Data

October 2015		
Item	Quantity	Sum Cost
Ti USB Developer Board	2	$38.90
ATmega328	4	$6.99
Screw Terminal Blocks	1	$5.95
PCB proto board	1	$7.99
PCB proto board	1	$3.57
Breadboard	1	$3.58
Cherry Mx switches	24	$26.74

Subtotal $93.72

November 2015		
Item	Quantity	Sum Cost
Female Mono Sockets	10	$4.66
3ft Mono Cables	11	$14.82

Subtotal $19.48

February 2016		
Item	Quantity	Sum Cost
3" PVC Cap	2	$7.96
4" PVC Cap	2	$15.90
Spring Doorstop	6	$11.82
Cherry Mx switches	50	$30.00
Assorted Bolts	100+	$44.44
JST Connectors	200	$13.98
4x8' 1/8" Acrylic Sheet	1	$132.32
2x4' 3mm Plywood	2	$12.00

Subtotal $268.42

March 2016		
Item	Quantity	Sum Cost
Assorted HomeDepot Hardware	X	$49.55

Subtotal $49.55

April 2016		
Item	Quantity	Sum Cost
Printed Poster	1	$39.24
Poster Frames	2	$20.78

Subtotal $39.24

Product Cost Estimate			
AT Keyboard—$41.50	**Cost**	**AT Button—$9.00**	**Cost**
16 x Cherry MX Switches	$16.00	1 x Cherry MX Switch	$1.00
2’ x 4’ Plywood	$6.00	1 x 3.5mm Mono	$0.50
2 x Handles	$10.00	0.3kg PLA	$7.50
1 x MSP430F5529	$12.00		

Global and Societal Impact of Project

We have reviewed the environmental impact of this project to be negligible. Components of the system are all readily available for purchase in the United States and have passed all pertinent consumer regulations. Hardware, when discarded, should be disposed of based on the relevant regulations.

Security vulnerabilities can be introduced when a person decides to build our design with modifications to the microcontroller firmware. Since many existing keyboard kits available on the market use similar microcontrollers and firmware approaches, this keyboard is adequately secure in respect to the scope of modern keyboards. Our project is open source, and if another person was to use our design and implement a key logger into the source code for the keyboard, they could potentially steal information inputted by another user. We plan to counteract this possibility by providing source and binary checksums while requesting all potential developers build the project directly from source or from a trusted authority.

We are designing our hardware to be interoperable to conform to existing standards (USB 1.1, 2.1 CDC and HID). Legacy and emerging computer systems should allow for the keyboard to be operational via the USB interface.

With respect to personal privacy, our prospective desktop software design plans be a learning program that will remember most commonly used phrases in order to help predict the user's word input and sentences. We will have to take ethical measures to ensure the raw or generated data taken is with consent and not distributed to unauthorized parties. We will also review later designs to implement fuzzing techniques which prevent the extraction of valid writings and communique of the user.

Health and safety concerns are minimal; the keyboard is constructed of nontoxic material. As far as the development of our product, we are prototyping with a chemical laser bed and we have to ensure proper ventilation and eye protection is used during operation. The 3D printers are printing with polylactic acid (PLA) thermoplastics that are food grade, thus posing limited risk and no noxious fumes during extrusion. Furthermore, we are soldering with leadless solder and are ensuring that proper safety measures are taken in the lab.

In regard to regulatory and legal issues, we are using MSP430 TI USB API Stack which includes the driverlib package. Both are clearly licensed under BSD compliant licenses via the following [9]. MSP430 hardware is subject to export restrictions for certain packages, but we are not crossing state or country lines and are not mailing the hardware systems.

In the end, systems are often victim to aging technology because the rapid innovations in technology for able bodied people—for example, systems developed a decade ago were not modified for high speed USB connections or flat panel monitors. Many of these old technologies are still in use today because they have no replacement or are prohibitively expensive to exchange. While the US does have programs that allow people who need AT, not everyone around the world has access to proper government programs and health insurance that will offset the costs. The project will aim to prioritize the safety and affordability of the users.

Future Work

The AT Keyboard came a long way over the past two semesters, but there were several tasks and aspects we did not investigate as fully as we wished to. The keyboard has recently been redesigned to have eight switches per paddle with a different character mapping. This new design should undergo more stringent user testing and review. Additionally, we are interested in supporting both MSP430 and Atmel microcontrollers as the driver for the keyboard; currently only the MSP430 has working firmware.

Our AT Button design reached a fairly stable point before the end of the semester. There is still much room for expansion on the idea (such as into wireless capabilities) and manufacturing methods (laser cut, CNC, etc).

Conclusion

Solutions available are expensive and difficult to obtain. There is a distinct need for human interface devices which meet many AT requirements and are available to wider audiences for more reasonable cost. The most significant differences between competing products versus our product is the low price. We strive to encourage others to create and publish an open source product that will allow people develop custom, low cost AT wherever a makerspace is available. Ultimately, the goal was to lay the groundwork for better availability of AT HID devices for the future.

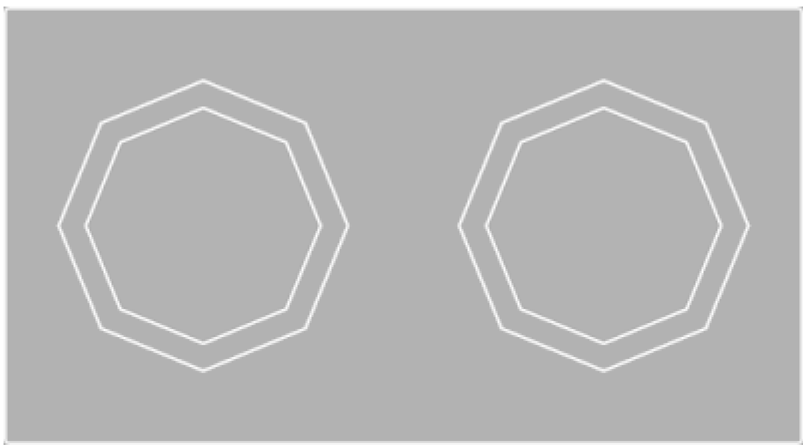
Appendix A

User's manual - AT Keyboard

In the most recent design, the AT Keyboard consists of a few major components: the mounting backplane, two 8-key paddles, and an MSP430 microcontroller launchpad. The end design of the AT Keyboard will include an ATmega microcontroller, but should be functionally identical.

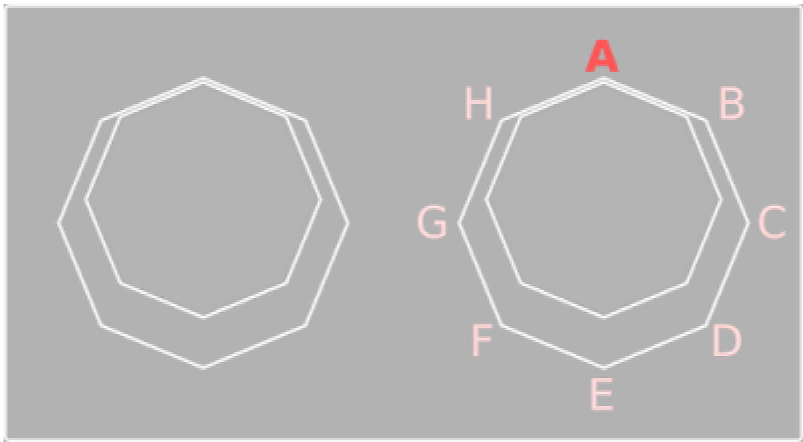
To use the AT Keyboard, simply plug in a USB-micro cable to both the keyboard and a host computer. The keyboard will automatically enumerate as a HID keyboard to the host computer, and should now be active for typing.

Begin with both paddles centered in their default state.



Move the left paddle to select a grouping of characters.

Keeping the left paddle pressed against the corner, move the right paddle to select and type a character.



To type the next character, first release or guide both paddles back to the center of their octagon, then repeat the group and character selection.

User's manual - AT Button

The AT Button in earlier releases came with a separate 3.5mm male to male mono cable. Newer versions include the cable as an integrated component of the button. All versions should be otherwise interoperable.

To use the AT Button, first locate and configure a device capable of accepting a 3.5mm momentary switch input. The button is designed to accept multiple base materials, and can have a rubber base to be set on a flat desk-like surface, a velcro base to be mounted to a velcro surface, or a bolt-hole pattern to be bolted to a receiving plate. Once attached and oriented for the user, plug the button into its receiving device and use as you would any commercially available assistive touch button.

User's manual - Predictive Software

The predictive text plugin is available for Chromium browsers. To use, install from the Chrome App Store (note: plugin is not yet available, on the Chrome App Store). While typing, the plugin will monitor text input and provide suggested auto-completions.

Appendix B - External Resources

[1]

http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1475175/?page=1

[2]

http://www.atmel.com/tools/AT90USBKEY.aspx

[3]

https://www.obdev.at/products/vusb/index.html

[4]

https://www.obdev.at/products/vusb/index.html

[6]

http://reprap.org/wiki/Printrbot

[7]

http://www.thingiverse.com/thing:591674

[8]

https://www.hotends.com/index.php?route=product/product&path=68&product_id=88

[9]

http://software-dl.ti.com/msp430/msp430_public_sw/mcu/msp430/MSP430_USB_Developers_Package/latest/index_FDS.html