2012-09-18

Exploring New Directions In Multi-Modal Spatial Data Access

Hagedorn, Douglas


The FuSe Map System

ENEL589: Fourth Year Engineering Design Project

Department of Electrical and Computer Engineering
University of Calgary
Winter 2012

Report #2

Project Title:

Functionally Separated (FuSe) Multi-Modal Mapping System

Team Information

Team number	**9**
Team name (*optional*)	FuSe Team
Project manager	Colin Madsen
Member-2	Chris Dizon
Member-3	Yifan Xu
Member-4	Pravallika Ryali
Member-5	Elizabeth Hunter

Customer Information

Customer (*Full name of the organization if applicable*)	Douglass Hagedorn
Website (*if applicable*)	N/A
Has this team signed a legal nondisclosure agreement?	**No**

Section-1: Brief description of this project

The FuSe Map Project is a contraption designed to aid the visually impaired. Its purpose is to help these people decipher their surroundings and navigate through from place to place. The device aims to achieve this goal by utilizing the human senses of touch, hearing and even voice as a means of communicating and guiding the user through a given area.

The device will be composed of several elements such as a magnetic touch board, a tracking camera, a map projector and a mapping database. The touch board will be constructed as a grid of 8x8 pixels. These pixels will represent a given map by the means of using magnetic forces to represent "ON" and "OFF" states; a pixel in the "ON" state will emit a magnetic force that the user will be able to detect with his/her fingertip (via the use of a specialize glove outfitted with magnets at the fingertips), whereas a pixel in the "OFF" state, the user will not be able to detect the magnetic force. The other elements will work in conjunction with the touch board and will be discussed further in the report.

Overall, the FuSe Map Project aims to improving the quality of life for those who are dim-sighted or completely blind.

Section-2: Detailed description of low-level design

Section-2.1: Overview of the entire system

The FuSe Map System is composed of several elements that include a magnetic touch board (composed of a grid of solenoids and H-bridge circuits), relays, a micro-controller, a single 12 & 5 volt power supply and a PC database. Each of these elements makes up the device that we have designed to aid amaurotic (partial and totally blind) people to become more aware of their surroundings and be able to navigate through it.

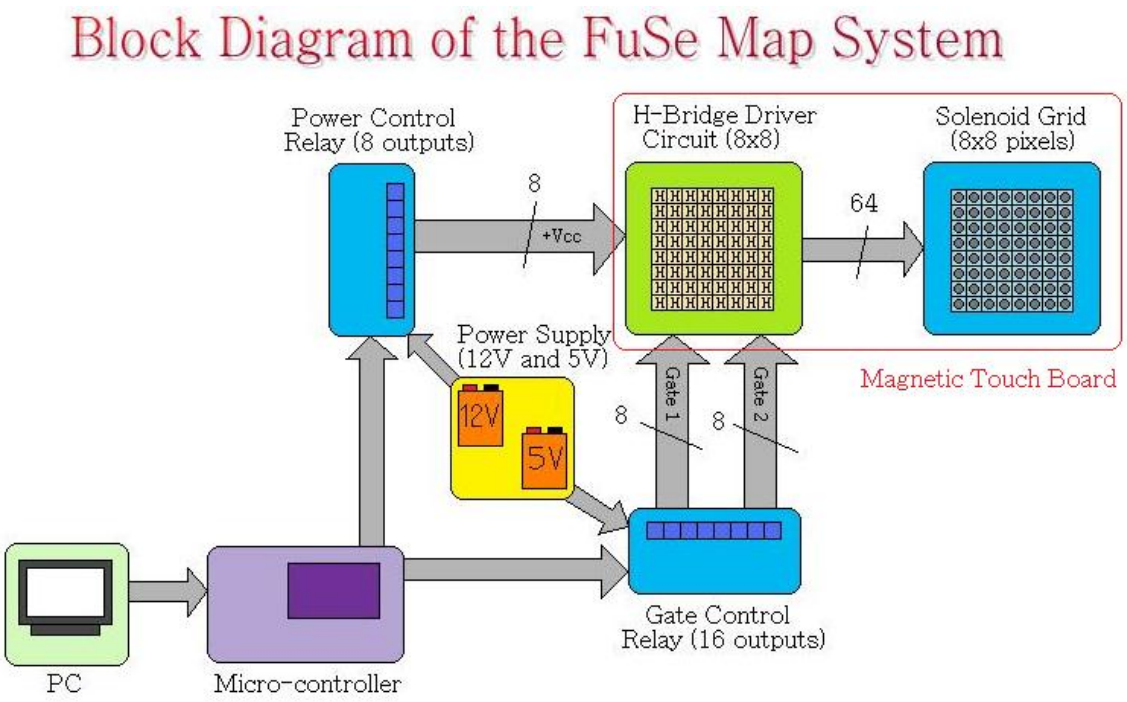


Figure 1: FuSe Map System

Section-2.2: Magnetic Touch Board (MTB)

Overall, the MTB is described as a grid of pixels (for our design, we chose to construct a grid of 8x8 pixels) that is used to represent a pixilated graphical map. Each pixel on the grid is able to send a magnetic force by the use of rare earth magnets planted beneath the touch board's surface (this function is done by using latching solenoids as it will be discussed in **Section-2.2.1**). The MTB would be controlled (via the micro-controller) to activate or deactivate certain pixels; the pixels that will be magnetize or demagnetize on the grid will depend on the map that is to be represented on the MTB.

The user would wear a specialize glove, outfitted with small permanent magnets at its fingertips, and by using this glove, he/she would smoothly slide his/her finger across the surface of the MTB; the user would then be able detect the magnetic pull of the rare earth magnets on certain pixels across the grid. Arranging a pattern of magnetized and demagnetized pixels on the MTB can create a graphical map in the user's mind, whether it may be simple pathway or a complex transit map **(NOTE: the complexity of the map that can be represented on to the MTB is dependent on the number of pixels the MTB is composed of; in our design, it is an 8x8 grid. The higher the number of pixels, the better the resolution of the map represented)**. By utilizing the sense of touch, the MTB is used to guide the user from a certain point to a specified destination.

As shown in **Figure 1** above, the magnetic touch board is composed of two sub-elements, the solenoid grid and the H-bridge driver circuit. Each of these elements works in conjunction with one another to achieve the intended functions, discussed in the previous paragraph.

Section-2.2.1: The Solenoid Grid

The Solenoid Grid is the mechanism that raises the rare earth magnets up to the bottom end of the pixel surface. The grid is basically 64 latching solenoids, arranged in an 8 by 8 square formation. Each solenoid behaves identically to one another, being able to latch and unlatch a metal shaft when a current flows through the solenoid coils; the direction of the current will determine whether the solenoid will latch or unlatch the shaft. Unlatching the solenoid will raise the shaft, thereby raising the rare earth magnet (which is placed on top of the shaft) against the pixel surface. And just as expected, latching the solenoid will clamp the shaft down in to the solenoid case, furthering the distance of the rare earth magnet from the pixel surface (refer to **Figure 2**).

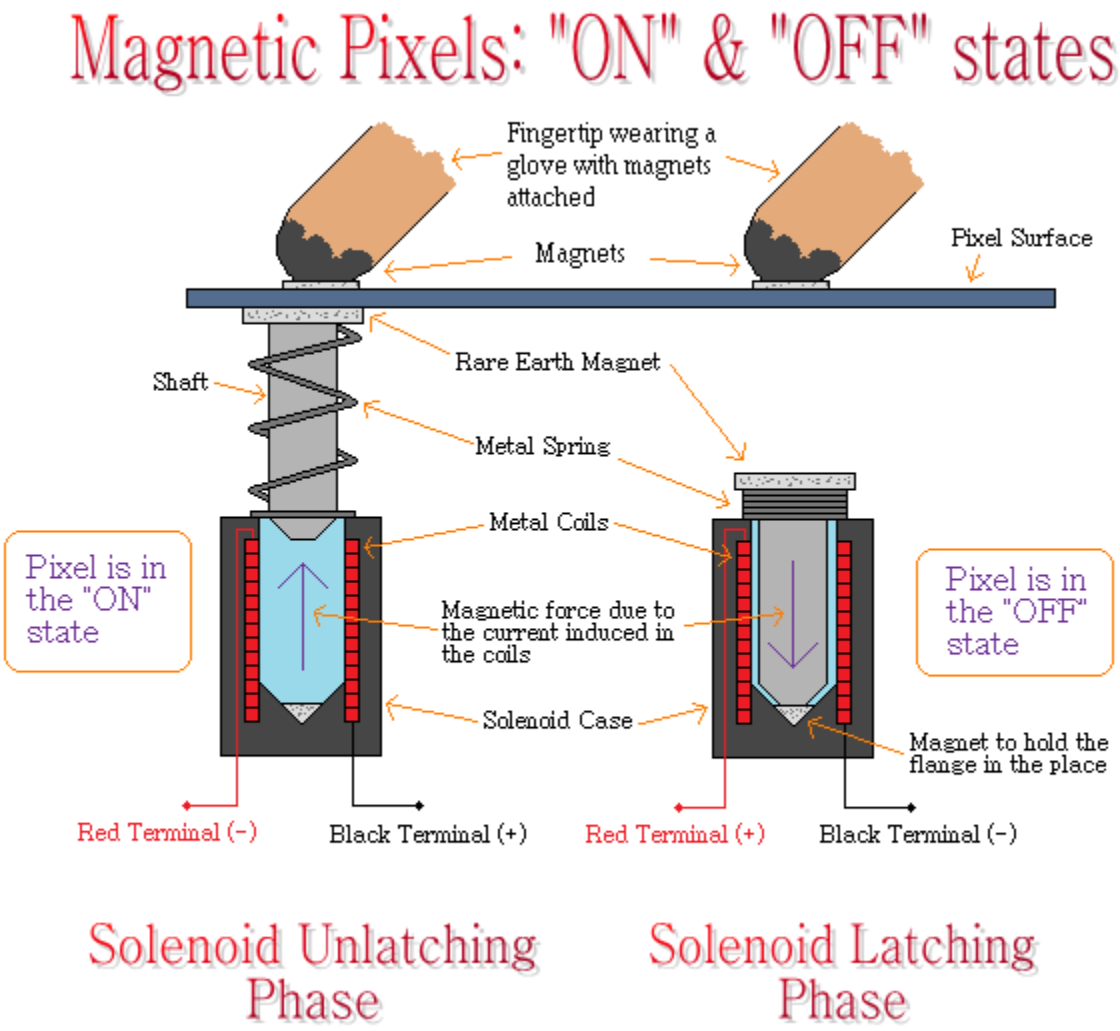


Figure 2: Pixel's "ON" & "OFF" States

From **Figure 2,** we can see that during the unlatching phase, the magnet is close enough to the pixel surface. The user's finger (wearing the iron filing glove) would be able to sense the magnetic pull between the magnet and the iron embedded in the glove, as he/she moves her finger across that specific pixel where its solenoid is unlatched; we

refer to this pixel to be in the “ON” state. During the latching phase, the solenoid distances the magnet from the pixel surface, allowing the pixel to be in the “OFF” state; at this distance (measured at 3cm), the user will not be able to detect any magnetic pull at this pixel.

The solenoid model that we chose was the **B14-L-X55-B-4** Latching Solenoid (**Appendix B**), manufactured by Ledex. We want all pixels to switch states (“ON” or “OFF”) almost instantly, magnetizing or demagnetizing each pixel the moment the micro-controller sends the command signals to the grid. From the solenoid datasheet, the nominal voltage value required for the solenoid to latch varies from 8.5V to 19V, depending on the speed of operation required; to induce the unlatching phase, a voltage of only 6V is required. We chose a 12V supply to be used for both the unlatch and latching states, as 12V circuitry is a standard in industry and by matching the solenoid power supply to the other components in our design (e.g. the relays), our power supply is significantly simplified. The 12V operation requires a maximum 25% duty cycle, however, due to the fact we will be individually energizing the solenoids, our maximum duty cycle will not exceed 1/64*100% = 1.6%. As well, the maximum rated pulse at 12V requires our latching period to remain less than 36 seconds; however we will be operating in the 0.1 second range which is well within all safety margins. The coil within the solenoid has two leads, representing the coil polarity - a red and black terminal. To activate the latching phase, the polarity requires the red terminal to be positive and the black terminal to be negative. As for the unlatching phase, the polarities would just be reversed - red is negative & black is positive (refer to **Appendix A** for further specifications and diagrams).

An advantage of using this specific solenoid is the fact that it is able to maintain its “states” without constant current flowing through the coils. That is, our solenoid grid will only draw power during the latching and unlatching transition period. The solenoid is equipped with a spring that encloses the shaft. During the unlatching phase, the spring is capable of supporting the shaft and the magnet (refer to **Figure 2 – Solenoid Unlatching Phase**) without the need to induce a magnetic force to unlatch the shaft. When the shaft has been latched, a magnet (placed inside the solenoid at the bottom) grasps the shaft firmly in place (refer to **Figure 2 – Solenoid Latching Phase**). Once the current flow through the coils is turned off, this magnet secures the shaft (even with the spring opposing it) until the solenoid goes through the unlatching phase.

Section-2.2.2: The H-Bridge Driver Circuit

As discussed, the solenoids are used in our design to elevate rare earth magnets up to the surface of every pixel on the grid. This function accomplishes the utilization of the sense of touch in the overall design of the FuSe Map System. However, in order to drive the solenoids, our design required a circuit to accomplish the task. Thus, our team designed the H-Bridge Driver Circuit (**Appendix C**). For each pixel, there is a solenoid (64 solenoids) and for each solenoid, there will be a single H-bridge circuit to drive it (64 individual circuits).

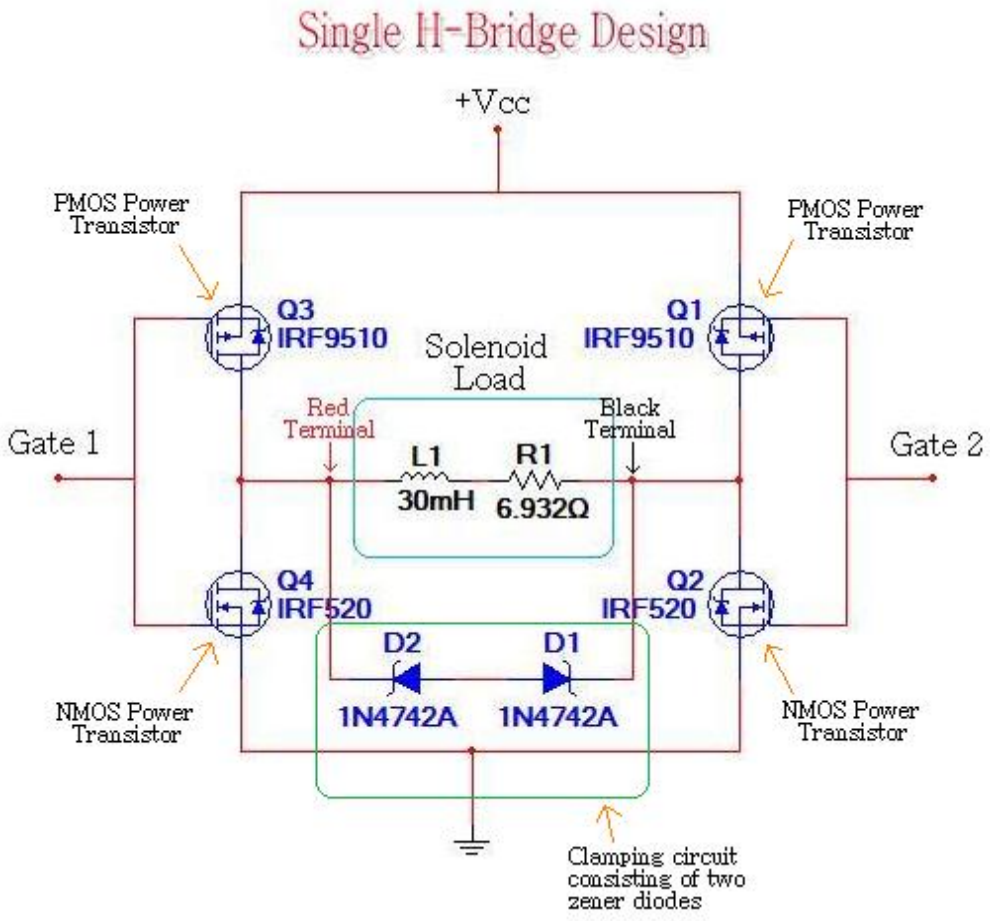


Figure 3: H-Bridge Design Schematics

The H-bridge circuit consists of a several components – 2 PMOS power transistors, 2 NMOS power transistors, 2 zener diodes and the solenoid load that the circuit is going to drive **(NOTE: the solenoid in our design has a resistance of 6.932 Ohms and an inductance of 5mH to 30mH, depending on whether or not the iron core is present in the solenoid. Therefore, it is modeled as a resistance and the worst-case inductance of 30mH in Figure 3)**. All of the MOSFET transistors in the design will function as an "ON & OFF" switch." It will be used to connect and break the circuit, from the source (+Vcc) through the load and to ground; the transistors that will be active or inactive will depend on whether we want to latch or unlatch the solenoid load.

As discussed in **Section-2.2.1**, inducing a positive voltage on the red terminal will cause the solenoid to latch; inducing a positive voltage on the black terminal will trigger the solenoid to unlatch. For a latching phase to take place, transistors Q3 and Q2 (refer to **Figure 3**) need to be active and rest would be inactive. This will allow current to flow from the voltage source (provided by an external power supply), across the load and exit towards ground (refer to **Figure 4**).

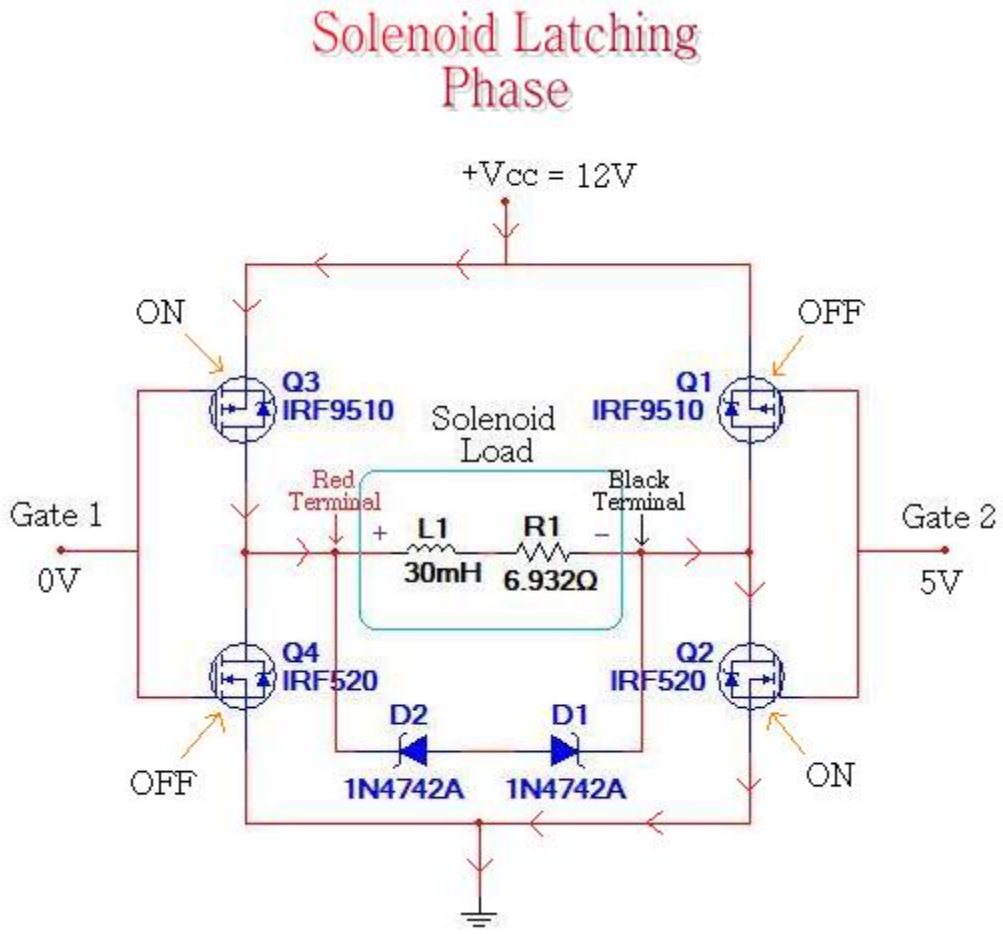


Figure 4: H-Bridge – Solenoid Latching Phase

From the above schematics, we can see that a voltage source of 12V is connected to the source terminal of the PMOS transistor Q3. During the latching phase, the transistor will pull the voltage at the solenoid load (connected to its drain terminal) up to the voltage connected to the transistor's source terminal; this means that a voltage drop of 12V will be induced on to the red terminal of the solenoid, thereby latching it. With the NMOS transistor Q2 active, the current flowing through the solenoid can exit to ground and thus, the circuit is complete. When transistors Q1 and Q4 off, it will behave just like an open circuit or infinitely high impedance.

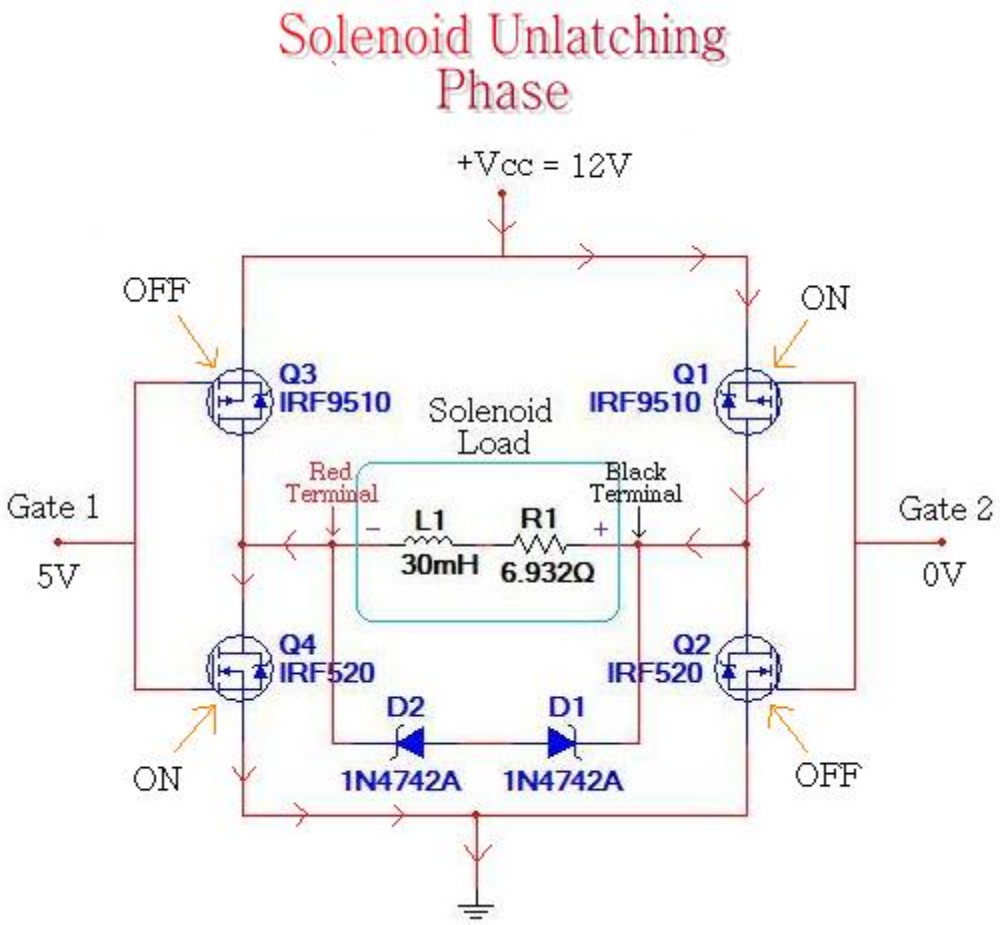


Figure 5: H-Bridge – Solenoid Unlatching Phase

During the unlatching phase, the states of all transistors will be reversed (Q1 & Q4 are on and Q3 & Q2 are off). Current will then travel through the active transistors, inducing a positive voltage on to the black terminal of the solenoid and exiting towards ground. Thereby, this procedure accomplishes the unlatching phase as we have expected during the design.

The transistors are able to switch on and off by manipulating the voltage induced at the transistors' gate terminals. From **Figure 4** and **Figure 5**, we have a voltage of 5V (supplied by a power supply) and 0V that is induced in "Gate 1" and "Gate 2." The voltages in these gates are controlled by a relay shown in **Figure 1** (the function of this relay will be further discussed in **Section-2.4**). A voltage of 5V at the gate will switch the PMOS transistor off and turn the NMOS transistor on. When the gate is connected to ground (0V), the NMOS turns off and the PMOS turns on.

The PMOS transistors that we incorporated in to our design are the model **IRF9510** and for the NMOS transistors, we used the model **IRF520**. Reasons we chose these specific models is because:

1. Both have a power dissipation that exceeds the amount that solenoid receives. From the testing results (**Section-4.3**), a voltage of 11V and a current of 1.5A go through the solenoid load. From the equation P = V*I, the power dissipated is 16.5W. From the transistor datasheets, the maximum power dissipation for the PMOS is 20W and for the NMOS is up to 60W.
2. The power transistors are equipped with a zener diode for built-in protection.
3. The specific models (IRF9510 & IRF520) are available in the database of the circuit modeling software that we used to create and simulate our circuit design (Multisim 11.0).
4. The transistors were affordable for our budget and were easy to obtain from the distributer (Digikey).

As mentioned earlier in **Section-2.2.1**, the solenoids in our design are able to maintain the "ON" and "OFF" states of each pixel on the grid (due to the spring and magnet built-in to the solenoid). This allows us to cut-off the power flowing through the solenoid, the instance that it latches or unlatches. However, during the early stages of designing the H-bridge circuit, we found (through the simulations) that shutting off the power causes voltage spikes across the solenoid load. This phenomenon occurred due to the inductance in the solenoid itself. These voltages spikes have the potential to damage the power transistors, so a clamping circuit was required to prevent this from happening. Therefore, we added two oppositely polarized zener diodes (D1 & D2 from **Figure 3, 4, and 5**) in series with each other and in parallel to the solenoid load to reduce the voltage spikes.

The zener diodes that we used were the model **IN4742**. We chose this model because they have a 12V zener breakdown voltage and 0.6V forward voltage drop, which allows the clamping circuit to cut-off any voltage spikes above 12.6V.

Overall, our H-bridge design accomplishes its intended function: to drive the solenoids with the appropriate voltage and current.

Section-2.2.3: The Rare Earth Magnets

The magnets that are attached to the solenoids are cylindrical neodymium rare earth magnets with a radius of 5mm and thickness of 1mm. Neodymium rare earth magnets are used because of their highly dense magnetic field (minimum 1T field strength) and this particular dimension was chosen due to the fact that the drop off in field strength was drastic enough after 1.3cm (the vertical travel of the solenoid plunger) to become imperceptible.

Section-2.3: The External Power Supply

As discussed in **Section-2.2.2**, the power transistors are being driven by ground and a 12V source at its source terminals and a 5V source at its gate terminals. These voltages are supplied by an external power supply, specifically the **BLP55** (**Appendix D**). The unit is also equipped with a built-in ground pin. The 12V and 5V pins are connected

respectively to two relays that are used to control the flow of current from the power supply to whichever H-bridge circuit on the grid the micro-controller chooses (refer to **Figure 1**). In general, the micro-controller sends input signals to these "control" relays (**Appendix E**) on which pixels on the grid it wants to activate or deactivate; the choice of pixels will solely depend on the map the user wants to be represented on the MTB.

Section-2.4: The Gate and Power Control Relays

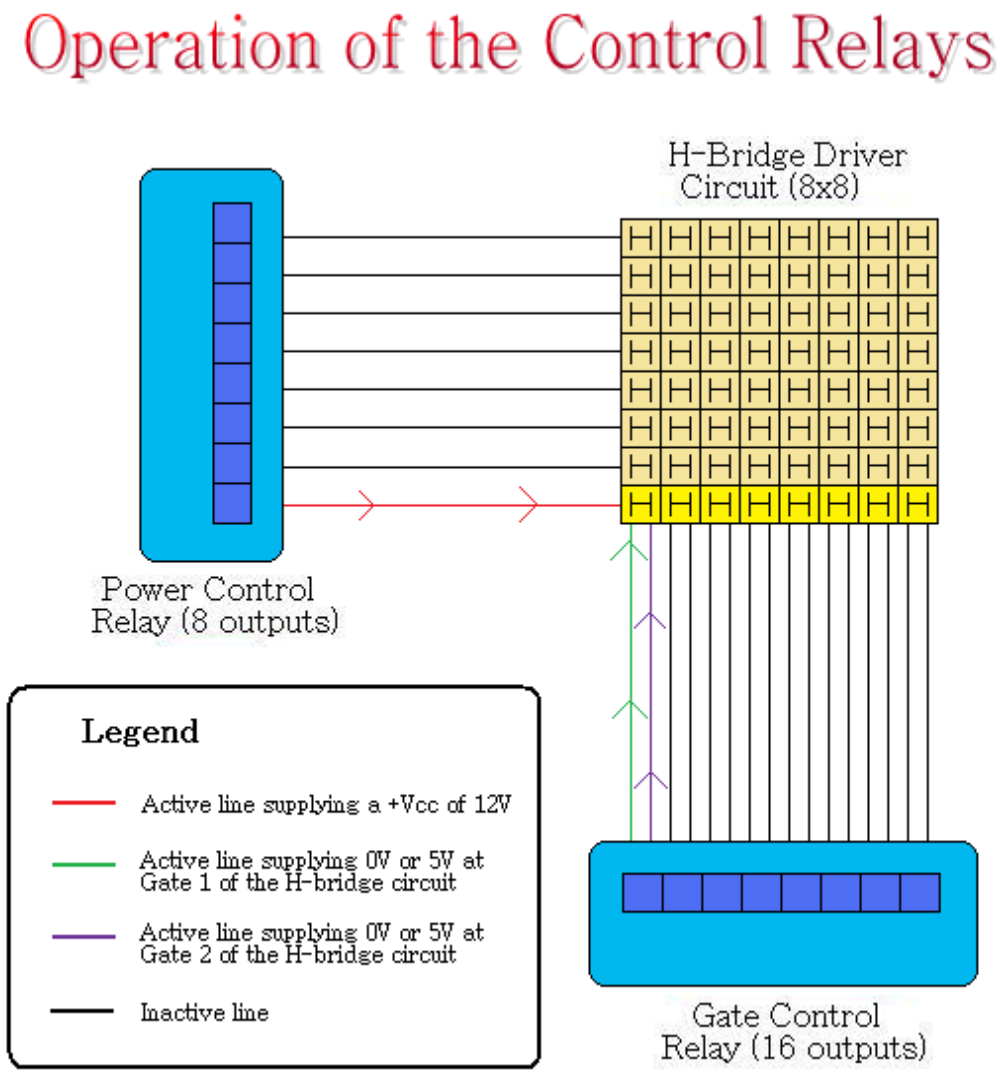


Figure 6: Relay Operations

In our design, we programmed our micro-controller to change the states of each pixel on the grid, one by one. The cycle would begin at the top row of the grid and at the left-most column. It would move across each pixel horizontally and change its states (magnetize or demagnetize) if it's necessary to do so. After cycling through the first row, it would then shift to down to the row below and being the cycling again, starting

at the left-most column. These steps are repeated until all 64 pixels have been examined.

Figure 6 shows a demonstration of how the process occurs. In the above example, the cycle will move through the bottom row of the grid, from left to right. Our design is able to carry-out this function by using a pair of relays to direct the voltage drop and current required to activate the H-bridge circuits. The relays that we chose was the 16-Channel Relay Interface board manufactured by Sainsmart due to its high current switching capacity (10A at rated voltage), high number of outputs, and ability to directly interface with our **Arduino Mega 2560** microcontroller (**Appendix F**).

Typically, when we require a specific pixel to be active or inactive, the appropriate transistors on the H-bridge circuit will be turn on and off; this is done by sending either a voltage of 0V or 5V to Gate 1 and Gate 2 (**Figure 3, 4 and 5**). The Gate Control Relay in **Figure 6** controls two elements: the voltage drop that will be induced at Gate 1 & 2 and the pixel along the active row (highlighted in yellow in **Figure 6**) that will receive the gate voltages. The voltages at Gate 1 & 2 will always be opposites of one another (eg. When Gate 1 = 5V, Gate 2 = 0V and vice-versa). This statement can be justified by **Figure 4 & 5**.

Once the gate voltages of all 8 H-bridge circuits have been set, the Power Control Relay will activate the source voltage (+Vcc = 12V) for that specific row. The PMOS transistors will receive a 12V drop on its source terminal, thereby closing the circuit and H-bridge will either latch or unlatch the solenoid (depending on the gate voltage settings). Until the Power Control Relay turns on, +Vcc will be 0V and solenoid load will not receive any current.

Section-2.5: The Micro-controller

The micro-controller that is being used in our design is the Mega 2560, manufactured by Arduino. The program in the micro-controller can be simply divided into two parts: serial communication handling and digital I/O handling.

Serial communication is an advanced feature that allows the Arduino Mega 2560 to send and receive serial data. It means that during the time the processor is running the program, the micro-controller is still able to receive messages (in bytes). The messages will be received from a serial port which is connected to the PC and the micro-controller will transmit messages through the same port and display it on the PC screen. By taking advantage of this feature, user interaction on the magnetic touch board can be implemented (through the micro-controller).

In our design, the user will be able to latch or unlatch one more solenoids by simply input the coordinates of the particular solenoids on to the PC screen. In the meantime, some useful information about the status of the program will be displayed on to the PC screen as well. Digital output pins are the ones actually generating the output signal (5V, 30mA). The program takes command messages from the PC through the serial port

and switches the output pins (on and off) to generate different signals. The actual functions implemented in the program are listed on the following table:

Function Name	**Parameter Type**	**Return Type**	**Description**
Serial.available	none	int	Monitors incoming messages from the serial port in bytes. As long as buffer has something to ready to read, this function returns the number of bytes in buffer. Hence, the return value is always >=0.
Serial.begin	int	void	Sets up serial communication by the rate as the number which is passed as parameter.
Serial.read	none	char	Reads one byte from the buffer as one char type value.
Serial.print	string	void	Sends the message which is received from the parameter byte by byte to the user end by serial port. It is often used when user needs to print information on the screen.
Serial.write	any type	void	Similar to print function except it does not transfer the messages by ASCII code. It is used to send messages without strings.
pinMode	int, Mode	void	Sets up the pin mode (input or output) for a particular pin which is indicated by the first parameter.
digitalWrtie	int, int	void	Writes logic 0 or 1 to a particular pin which is indicated by the first parameter.
serialHandler	none	void	Custom function. It involves all the functions listed above. This function always takes a pair of numbers from the buffer and checks if they match any solenoid number we have on board. If it is true, send logic 0 or 1 (depending

			on the previous status) through that particular pin to latch or unlatch a solenoid. If it is not true, simply ignore.
setup	none	void	System function. Always running at the first when program is being executed. Everything that needs to be initialized such as pinMode, Serial.begin needs to be done here.
loop	none	void	System function. It involves serialHandler. All the code written in this function is being executed infinite times till the micro-controller is powered off.

Table 1: Micro-controller Features

Section-2.6: PC Database

The PC serves as a database for the graphical maps that is to be translated through our software code (designed in C++ and discussed in **Section-2.9**) and sent to the micro-controller as signals to activate specific pixels on the MTB.

Section-2.7: Overhead Projector

The Projector is included to project a visual image of the map on to the MTB. The resolution of the map will be consistence with the number of pixels on the MTB (in our design, it will be an 8x8 pixel resolution). The idea of projecting a visual of the map on to the MTB is to provide further assistance to users who are partial-sighted or still maintains a small degree of their vision. We chose the Optoma PK201 Pocket Projector for its compact size and short focal range, allowing it to be mounted unobtrusively on our physical design.

Section-2.8: Tracking System

Along with the projector, a camera is place over the MTB. This component will be used to track the user's fingertip as he/she moves it across the pixel surface. This function will allow the PC to follow the user's movements across the map and send a feedback of audio signals to inform or alert the user on where he/she is. (e.g. If the map being represented on the MTB was the Calgary C-train map, the PC can inform the user on what station his/her finger is moving on top on). Testing (**Section-4.10**) was successfully completed using a Nintendo Wii remote; however we are attempting to implement a Microsoft Kinect camera for its upgraded hand-tracking capabilities.

Section-2.9: The Software Design Interface

The purpose of our software design is to program the micro-controller, allowing it to change the pixel states of the MTB accordingly to represent a desired map. For testing and demonstration purposes, we chose to model the Calgary C-train map on to the magnetic touch board (**Figure 7**). **(NOTE: the representation of the C-train map shown in Figure 7 is by a grid of 12x16 pixels. Our final design will represent the same map, using an 8x8 grid)**

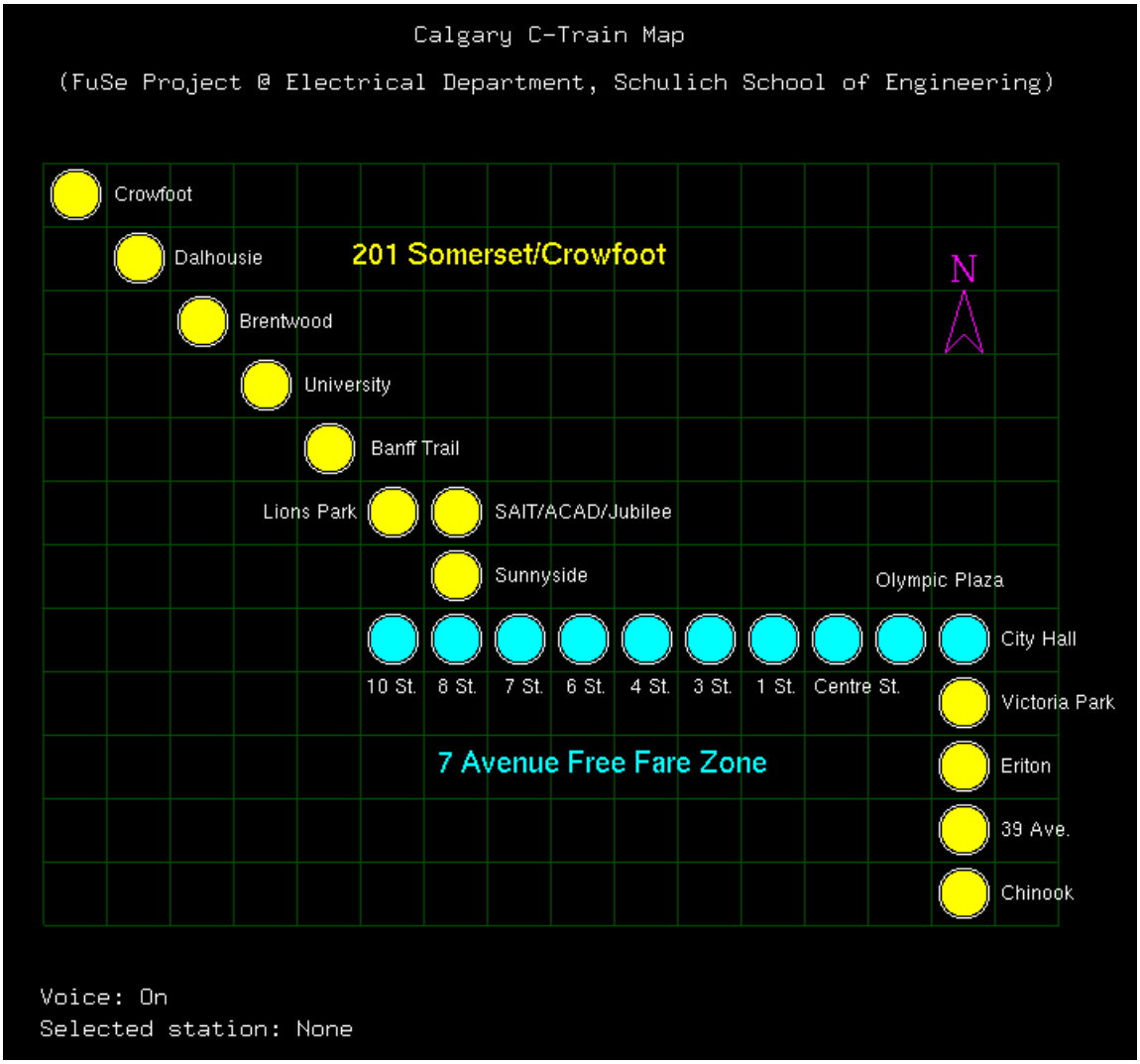


Figure 7: Calgary C-Train Map generated using C++

The display of the Calgary C-Train map consists of six critical classes that work altogether to form the entire map interface. These six classes are the:

- **Point Class**: this class is the basic data structure of constructing a single point. A point is usually one pixel large when it is shown on screen. The point is a basic element to that is used to construct the complex shapes of graphics such as lines, circles, polygons, etc. A point class includes:

Data Member	Data Type	Description
x_	Float	Represent x value of a 3D coordinate system in space.
y_	Float	Represent y value of a 3D coordinate system in space.
z_	Float	Represent z value of a 3D coordinate system in space.

Function Name	Parameter Type	Return Type	Description
Cross product	Point, Point	Point	Does cross product of two vectors which are represented in Point structure.
Dot product	Point	double	Does dot product of two vectors which are represented in Point structure.
Unit vector	Point	Point	Transfers one vector which is represented in Point structure, to a unit vector.

Table 2: Point Class

- **Grid Class**: this class is the basic element of forming the map. A map is composed of multiple grids, where it contains several bits of information such as the content within the grid, color, size, etc. In our map, there are 64 pixels, each represented by a "software grid." In total, these "software grids" represent the total number solenoids on the MTB design. A grid class includes:

Data Member	Data Type	Description
Id	Int	The id number assigned to this grid.
hasEdges	Bool	A Boolean valuable indicating if this grid has edges.
Upleft	Point	Upper-left point of a grid
Upright	Point	Upper-right point of a grid
Downleft	Point	Down-left point of a grid
Downright	Point	Down-right point of a grid
hasCircle	Bool	A Boolean valuable indicating if this grid has a circle (station point).
Centre	Point	Centre point of this grid
Radius	Float	Radius value of the circle
color	std::string	Color of the circle
isHighlighted	Bool	A Boolean valuable indicating if this grid is highlighted.
radius_big	Float	A number indicating how big the circle becomes when it is highlighted.
name	std::string	Name of the station point if this grid contains a circle.

Function Name	Parameter Type	Return Type	Description
addCircle	None	void	Set “hasCircle” data member to true.
withdrawCircle	None	void	Set “hasCircle” data member to false.
highlightOn	None	void	Set “isHighlighted” data member to false.
highlightOff	None	void	Set “isHighlighted” data member to false.
edgesOn	None	void	Set “hasEdges” data member to true.
edgesOff	None	void	Set “hasEdges” data member to true.
setColor	std::string	void	Set “color” data member to the value passed by parameter.
setName	std::string	void	Set “name” data member to the value passed by parameter.
drawGrid	None	void	Most important function in this class. This function draws every grid according to its content on the map.

Table 3: Grid Class

- **Map Class**: this class uses the grid class to compose the whole map. It generates multiple grids on the screen and sets up the content for each grid. For example, the map class can assign a circle to some of the grids and labels those circles. This class includes:

Function Name	Parameter Type	Return Type	Description
addtoLineDalhousie	Grid, std::string	void	Call “”addCircle” function in Grid. Set color to yellow. Set circle name same as the parameter.
addtoLineDowntown	Grid, std::string	void	Call “”addCircle” function in Grid. Set color to blue. Set circle name same as the parameter.
buildMap	std::vector< std::vector<Grid> >	void	Most important function in this class. It builds the whole map with multiple grids where some of the grids have circles (station point).

Table 4: Map Class

- **Text Class**: this class writes the station names and places it besides the corresponding stations (represented by circles on a single pixel).

Function Name	Parameter Type	Return Type	Description
drawString	void*, const std::string&	void	Draw one character by one character according to the parameter string.
drawText	bool, std::string	void	Use "drawString" function to draw all text for each grid which has a circle in and place them at the proper place. This function also draws the title above the map and the status line below the map.

Table 5: Text Class

- **Speak Class**: this class is responsible for providing audio feedback when a circle is highlighted. The class uses MS speech API.

Function Name	Parameter Type	Return Type	Description
speakStationName	std::string	int	This function initializes the speech machine first and then plays audio feedback according to the string that is passed into the function.

Table 6: Speak Class

- **Main Class**: this is the control class of the program. Also, user interactions are managed by this class.

Function Name	Parameter Type	Return Type	Description
speakFunc	void*	void	Call "speakStationName" function to speak out the station name which is highlighted.
display	None	void	Call "drawGrid" function to draw every grid with its content. Call "drawText" function to put on all the text.
resizeWindow	Int, int	void	When the window is resized, keep the same scale of the map.
keyboard	Unsigned char, int, int	void	Allow user interaction through keyboard. Pressing 'v' to switch on/off voice; pressing 't' to switch on/off text; pressing 'g' to switch on/off grids.

mouseMove	int, int	void	Determine the current mouse position. Check if the mouse is in a grid; in a circle or out of a grid; out of a circle. Call "speakFunc" to play audio feedback when the mouse is in a circle.
main	int, char**	int	Initialize global valuables such as Map, Text, etc. Set up glut and pass the control to glut at the end.

Table 7: Main Class

Section-3: Detailed test procedure

Our testing procedures are categorized in to 2 parts: software and hardware. The software side will only test the micro-controller and the programming code (designed in C++), which will be used send command signals to the micro-controller. On the hardware side, we have listed procedures that will test the main elements in our design, such as the magnetic touch board (the solenoid grid and the H-bridge driver circuits), the external power supply and the control relays (refer to **Figure 1** for a block diagram overview).

Section-3.1.1: Test Plan for the Software Interface

Features to be Tested	**Test Suite**	**Testing Procedures**	**Expected Results**
Stations and Grids display	Test if the map shows all the grids and station points correctly.	1. Open the file "FuSe.exe" 2. View map	Map shows 64 green grids where some of the grids contain station points with a white circle and blue content.
Text display	Test if the map shows a text message beside each station point.	1. Open the file "FuSe.exe" 2. Watch map	Every station point has a white colored text beside it.
Audio display	Test if the map is able to play audio feedback.	1. Open the file "FuSe.exe" 2. Move the mouse over a single station point. 3. Repeat step 2 for all other station points.	Once the mouse moves across a station point, the name of the station will be played as audio feedback. The name spoken should correspond to the text written beside the point. All the station points should behave in the same manner.
Station points highlighting	Test if all the station points can	1. Open the file "FuSe.exe" 2. Move the mouse over a	Once the mouse moves across a station point, the circle point will

	be highlighted when the mouse is moved over it.	single station point. 3. Repeat step 2 for all other station points.	be enlarged and turn red. All the station points should behave in the same manner.
Station points de-highlighting	Test if all the station points can be de-highlighted when the mouse is moved away from the point.	1. Open the file "FuSe.exe" 2. Move the mouse over a single station point. 3. Move the mouse away from the selected station point. 4. Repeat step 2 and 3 for all other points.	Once a station point is highlighted, moving the mouse away of the circle point should de-highlight this point. It means the point circle will return back to its normal size and turn blue. All the station points should behave the same manner.
Grids: on/off switching	Test if the grids can be switched on and off.	1. Open the file "FuSe.exe" 2. Press 'g' on the keyboard (lower-case only). 3. Press 'g' again.	The first time 'g' is pressed, all 64 grids on the display will become hidden. Pressing 'g' again will restore the grids back. The on/off status is correctly shown at the bottom of map.
Voice: on/off switching	Test if the audio feedback can be switched on and off.	1. Open the file "FuSe.exe" 2. Press 'v' on the keyboard (lower-case only). 3. Press 'v' again.	The first time 'v' is pressed, the audio feedback will be disabled. Repeating the test procedures for the "**Audio display**" will result in no audio feedback. Pressing 'v' again will restore the audio feedback feature. The on/off status is correctly shown at the bottom of **Figure 7**.
Text: on/off switching	Test if the text displays can be switched on and off.	1. Open the file "FuSe.exe" 2. Press 't' on the keyboard (lower-case only). 3. Press 't' again.	The first time 't' is pressed, all text on the display will become hidden. Pressing 't' once again will restore the text back on to the display. The on/off status is shown at the bottom of **Figure 7**.

Table 8: Software Test Plan

Section-3.1.2: Test Plan for the Micro-controller

Features to be Tested	Test Suite	Testing Procedures	Expected Results
Digital output	Test if digital output pin 22 - 45 functions.	1. Connect pin 22 - 45 to 24 LED lights. 2. Connect each LED to the GND pins on the board. 3. Connect the board with PC through USB cable. 4. In the program, set pin 22 -	When logic 1 is set for those pins, uploading program causes all the LED lights activated. When logic 0 is set for the pins, uploading program causes all the LED lights de-activated.

		45 as output using "pinMode" in "setup" function. 5. In the program, set pin 22 - 45 to generate logic 1 using "digitalWrtie" in "loop" function. 6. Upload program to Arduino Mega. 7. Repeat step 5 and 6 by setting output as logic 0.	
Serial data transmitting	Test if data can be transmitted by serial port to the user PC through USB cable.	1. Connect the board with PC through USB cable. 2. In the program, serial communication is set up, using "Serial.begin" in "setup" function. 3. In the program, write "Hello World" using "Serial.print" in "setup" function. 4. Upload program to Arduino Mega.	After "serial monitor" window is open in PC and waiting for 10 seconds, "Hello World" is printed in the console.
Serial data receiving	Test if data can be received by serial port from user to micro-controller through USB cable.	1. Connect pin 22 - 45 to 24 LED lights. 2. Connect each LED to the GND pins on the board. 3. Connect the board with PC through USB cable. 4. In the program, set pin 22 - 45 as output using "pinMode" in "setup" function. 5. In the program, set pin 22 - 45 to generate logic 1 using "digitalWrtie" in "serialHandle" function and put it in the "loop" function. 6. In the "serialHandle" function, check the data in the buffer by pairs. If this pair of numbers matches the number of one LED, switch on/off this LED. Otherwise, ignore.	**Case 1**: numbers which are less than 22 or greater than 45 do not cause any changes on LEDs. **Case 2**: numbers which are between 22 and 45 are entered pair by pair (e.g.: 22, 34, 41, etc). The corresponding LEDs are switched on/off as the order of the pairs being entered. **Case 3**: numbers which are between 22 and 45 are entered more than one pair at once (e.g.: 223441). All corresponding LEDs are switched on/off at the same time. **Case 4**: numbers which are between 22 and 45 are entered digit by digit (e.g.: 2, 2, 3, 4, 4, 1). The corresponding LED is not changed until a pair of numbers are formed. In this case, LED 22, 34, 41 are switched on/off once

		7. Type numbers from 0 to 50 in the PC monitor.	numbers 2, 4 and 1 are received.

Table 9: Micro-controller Test Plan

Section-3.2: Test Plan for the Hardware Components

Features to be Tested	Test Suite	Testing Procedures	Expected Results
Magnetic Touch Board – consists of the latching solenoids and H-bridge circuit	Test to determine if the solenoids will latch and unlatch, being driven by the H-bridge circuit described in **Section 2.2**	1. We will use a circuit simulation program (Multisim 11.0) to build and test the H-bridge driver circuit. 2. Record the results of the simulated circuit – the current and voltage drop across the solenoid). 3. Build a single magnetic pixel along with the driver circuit to latch and unlatch the solenoid. 4. Test the actual circuit and record the current and voltage drop across the solenoid. 5. Compare the results from the experiment with the simulation and calculate the percentage error. If the percentage error is less than 10%, construct a row of magnetic pixels (8 pixels). 6. Test the design to determine if the power supply can drive the magnetic pixels adequately.	Given that a 12V source will be power the solenoids, we expect the unit to latch and unlatch. 12V will be induced during the latching and unlatching phases, which is more than capable enough to activate the solenoid (from **Appendix A**, voltage required to latch is 8.5V and to unlatch is 6V). We are using 12V to ensure that any discrepancies in voltages will not hinder the solenoid's operation. Multisim results were reasonable (refer to **Figure 8** from **Section-4.3**) and expect our actual tests to produce similar results.
Power Supply (equipped with 12V, 5V and ground pins)	To test and ensure that the power supply is operating properly.	1. The black lead from the main power cable was soldered to the switch. A Multi-tester was used to test for continuity. Each end was attached to the Multi-tester lead and the resistance was measured.	We are expecting that the power supply is capable of providing the +Vcc (12V) and gate voltages (5V) of the H-bridge circuit we designed. We are expecting the output pins to register 0V after turning the device off.

		a. < 1Ω indicates a connection b. "OL" indicates no connection 2. The line from the main cable was attached to the line pin in the molex connector. Continuity was tested with the Multi-tester. 3. When the switch was placed in the "OFF" position, the main cable was plugged in to an electrical outlet and the molex connector was connected to the power supply. Voltage drop was measured across a 1.2kΩ resistor connected between the power supply's output and ground. a. There are 4 voltage outputs. +12V, -12V, and two +5V. Each one was tested with the Multi-tester. 4. When the switch is turned off, pins were tested to ensure they turned off immediately as well. 5. A 3A load was connected to the 12V and later the 5V pins and was allowed to draw current for a 2 hour period. 6. The power supply was unplugged after the continuous loading test and any unusual temperatures noted	We expect the power supply to supply a 3A continuous load for two hours without overheating.
Relays	To test the functionality of	1. We connected the appropriate input pins to the	The relays should switch appropriately according to the

	the relays in switching a 12V signal under no-load and loaded conditions.	logic input (for the microcontroller) and the 12V power supply, then energized the power supply. 2. We individually tested each of the 16 relays under a no-load condition by placing a 1 mega ohm resistor across each of the relay outputs and sequentially drove each relay with the appropriate microcontroller input. 3. We repeated the previous testing procedure, replacing the 1 mega ohm resistor with a solenoid.	microcontroller input under both no-load and loaded conditions.

Table 10: Hardware Test Plan

Section-4: Testing and refinement

Section-4.1: Expected performance(s)

We require the magnetic touch board to be able to switch states (ON & OFF); when a pixel is in the "ON" state, the user will be able to easily detect the magnetic pull of the magnet beneath the pixel surface (refer to **Figure 2**). The pixels, being driven by latching solenoids, require a reasonable amount of power to switch states (almost) instantly – roughly 0.1s. Overall, we expect the magnetic touch board to display a pattern of "ON" & "OFF" pixels that will represent a graphical map with which the user can interpret through the sense of touching the pixel surface.

Section-4.2: Expected list of specifications related to performance(s)

1. We require that the power drawn by each solenoid will not exceed 30W.
2. We require our software code and micro-controller to function properly as described in the **Section-2.9** & **Section-2.5**.
3. While changing the pixel states, we require the device to activate/deactivate one row of solenoids at a time (due to power consumptions).
4. For the magnetic touch board, we are aiming to constructing a grid of 20x30 pixels.
5. For each solenoid, we require it to have a magnetic flux density (***B***) of 0.1T in the "OFF" state and 1.5T in the "ON" state.
6. We require the entire contraption to not exceed the maximum weight of 50kg.

7. We are aiming to using a projector that has a resolution that is compatible with the resolution of the magnetic touch board.
8. We require the overhead camera to be able to track the user's fingertip up to a single magnetic pixel.

Section-4.3: Test result of specification #1

We require that the power drawn by each solenoid will not exceed 30W.

4.3.1-Further description of test procedure

Refer to the "MTB" test procedures in **Table 10** from **Section-3.2**.

4.3.2-Test result

After following the testing procedures described in **Table 10**, we measure a current value of 1.73A, following through the solenoid load. Using an RLC meter (supplied to our team by the lab techs in ENA305), we measured the resistance of the solenoid to be 6.9 ohms and the inductance to be 5-30mH. Using the power equation P = I*V, we calculated the power drawn by the solenoid:

P = I*V = R*I^2 = (6.9ohms)*(1.73A) ^2 = 20.65W

From the above result, we have met our design specification, for the power drawn does not exceed 30W.

We simulated our H-bridge design to drive solenoid in Multisim and got the following results shown in **Figure 8**.

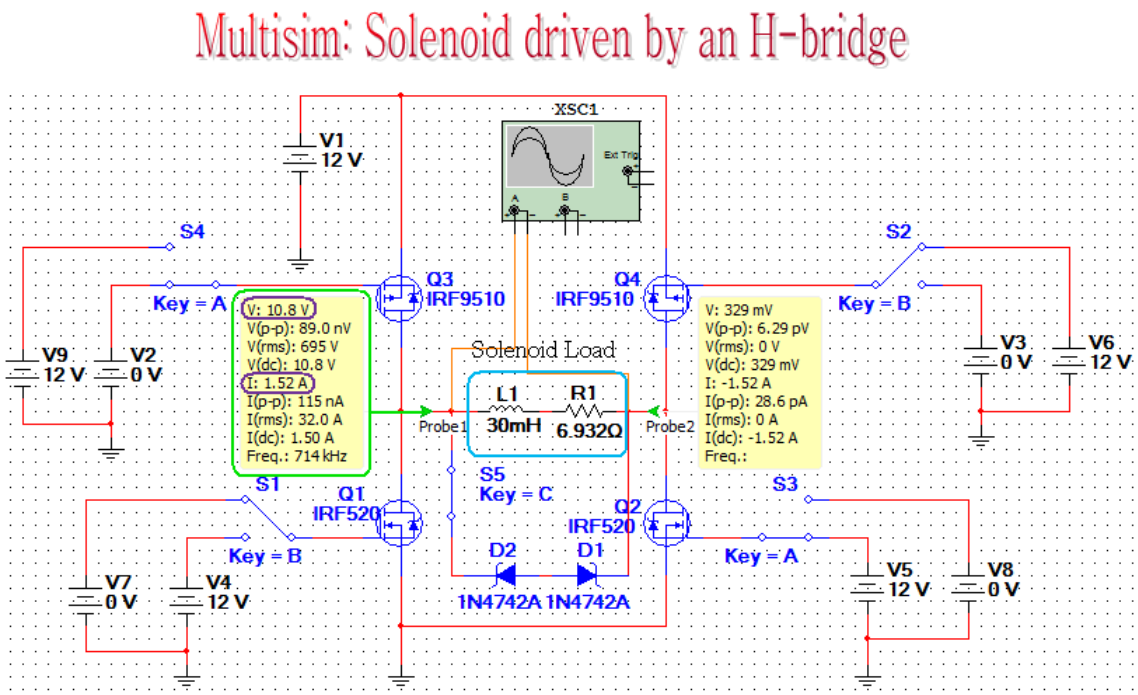


Figure 8 – Solenoid & H-bridge Simulation

"Probe 1" from the above picture shows the voltage drop and current flowing through the load. Using the power equation, we get:

P = I*V = (1.73A)*(10.8V) = 16.42W

Comparing this with our measured results, we calculated the percent error

%error = ((20.65 – 16.42) / 20.65) * 100 = 20.5%

Though the difference in values between the experiment and simulation is 20.5%, we concluded that the discrepancy was due to the circuit components used in the simulation. MOSFET transistors and zener diodes modeled in Multisim different slightly from the actual parts that we purchased due to different manufacturers. Nevertheless, our test result satisfies our design conditions for the power consumption of the solenoid does not exceed 30W.

4.3.3-Refinement

Given the success of our test results, we did not require any further improvement in this section of the design.

4.3.4-Witnesses

We declare the test results recorded above were valid and accurate.

Name: _____ Signature: _____

Name: _____ Signature: _____

Section-4.4: Test result of specification #2

We require our software code and micro-controller to function properly as described in the **Section-2.9** & **Section-2.5**.

4.4.1-Further description of test procedure (*if applicable*)

Refer to **Table 8** & **9** from **Section-3.1.1** for the testing procedures.

4.4.2-Test result

When we followed the testing procedures describe in **Section-3.1.1** & **Section-3.1.2**, we were able to obtain the expected results; as stated the **Expected Results** column in **Table 8** & **9**, our micro-controller operated correctly and the software code perform its intended features, generating the program displayed in **Figure 7** from **Section-2.9**.

4.4.3-Refinement

Since the programming code has functioned as intended, no further refinement has been required.

4.4.4-Witnesses

We declare the test results recorded above were valid and accurate.

Name: _____ Signature: _____

Name: _____ Signature: _____

Section-4.5: Test result of specification #3

While changing the pixel states, we require the device to activate/deactivate one row of solenoids at a time (due to power consumptions).

4.5.1-Further description of test procedure

Refer to the "Relay" test procedures in **Table 10** from **Section-3.2**.

4.5.2-Test result

Due to the relays that we have chosen and to the power limitations, we are unable to activate one row of solenoids at a time. The relays are pre-designed and therefore have no control over the power supplied by it. During testing, we came to realize that the relays can only draw 1 or 2 solenoids at a time.

4.5.3-Refinement

Due to our test results, we have decided to draw 1 or 2 solenoids at a time. Thereby, our design will only be capable of activating a maximum of two solenoids at once.

4.5.4-Witnesses

We declare the test results recorded above were valid and accurate.

Name: _____ Signature: _____

Name: _____ Signature: _____

Section-4.6: Test result of specification #4

For the magnetic touch board, we are aiming to constructing a grid of 20x30 pixels.

4.6.1-Further description of test procedure (*if applicable*)

Determine whether the scale of building 20x30 pixels will accommodate our maximum budget of $2800 (finances were provided to us by our client). We will determine this by the parts that our team is able to obtain and afford from manufacturers (Digikey, Ledex, Arduino, etc.)

4.6.2-Test result

During the early stages of designing, we have initially designed the MTB to be built from a grid of 20x30 pixels. However, further along the design stage, we have added the H-bridge circuits and relays to the overall design, driving up the cost of supplies that we would have to purchase (**Appendix A**). Due to the budget constraints of our project, we have downsized our MTB design to an 8x8 grid.

4.6.3-Refinement

As stated in the previous section, we have altered our initial MTB design from a 20x30 to an 8x8 pixel grid.

4.6.4-Witnesses

We declare the test results recorded above were valid and accurate.

Name: _____ Signature: _____

Name: _____ Signature: _____

Section-4.7: Test result of specification #5

For each solenoid, we require it to have a magnetic flux density (***B***) of 0.1T in the "OFF" state and 1.5T in the "ON" state.

4.7.1-Further description of test procedure

We will use neodymium magnets with a standardized filed strength of 1.4T and slowly pull the magnets away from the pixel surface.

4.7.2-Test result

At a distance of 13mm away from the surface, the 1,4T field was reduced to imperceptibility.

4.7.3-Refinement

We conclude that no further refinement is required for this specification.

4.7.4-Witnesses

We declare the test results recorded above were valid and accurate.

Name: _____ Signature: _____

Name: _____ Signature: _____

Section-4.8: Test result of specification #6

We require the entire contraption to not exceed the maximum weight of 50kg.

4.8.1-Further description of test procedure

A constraint proposed to our team by Doug (our client) was to ensure that the prototype design will not exceed 50kg. This constraint was given to us for the sake of transporting the prototype with ease in order for our client to demonstrate its features

and capabilities to Innovate Calgary. This requirement is meet by ensuring that we select and purchase the minimum amount of light-weighted parts.

4.8.2-Test result

As of our current design, our prototype does not exceed the 50kg limit, weighting at about 4kg only.

4.8.3-Refinement

Since we are satisfied with the results of this design requirement, refinements will not be necessary,

4.8.4-Witnesses

We declare the test results recorded above were valid and accurate.

Name: _____ Signature: _____

Name: _____ Signature: _____

Section-4.9: Test result of specification #7

We are aiming to using a projector that has a resolution that is compatible with the resolution of the magnetic touch board.

4.9.1-Further description of test procedure

Determine whether the projector that we purchased (refer to **Section-2.7**) is adequate enough to accomplish its intended purpose.

4.9.2-Test result

Indeed, as expected, our pocket project is able to meet our design requirement. Having a short focal range, it is compatible with the resolution of the MTB.

The pixels are scaled down therefore the projector will display the same resolution.

4.9.3-Refinement

The projector purchased was of excellent quality, thereby, we require no further improvement whatsoever.

4.9.4-Witnesses

We declare the test results recorded above were valid and accurate.

Name: _____ Signature: _____

Name: _____ Signature: _____

Section-4.10: Test result of specification #8

We require the overhead camera to track the user's fingertip up to a single magnetic pixel.

4.10.1-Further description of test procedure

We require the overhead camera (described in **Section-2.8**) to be able to track the user's fingertip up to a single magnetic pixel.

4.10.2-Test result

Testing was successfully completed using a Nintendo Wii remote. With a PC, we projected an 8x8 grid of pixels on to a wooden desk (simply any surface just to test the tracking system itself). We then shot a laser pointer at a random pixel on the projected image. The remote was able to track the laser and the pixel responded to the laser by changing its states; the pixel was highlighted red when we pointed the laser at it and returned to its normal shade of green when the laser was turned off. Overall, our tracking system fulfilled its purpose successfully.

4.10.3-Refinement

We are currently attempting to implement a Microsoft Kinect camera to upgrade the hand-tracking capabilities of our current design. If unsuccessful, we will stay with our current tracking system.

4.10.4-Witnesses

We declare the test results recorded above were valid and accurate.

Name: _____ Signature: _____

Name: _____ Signature: _____

Section-5: Suggestions for further improvements

Further improvements for our current design include:

- The usage of custom solenoids that are smaller in size will allow a greater pixel resolution and lowered power consumption.
- The use of a custom PCB with surface mounted components instead of attached circuit components to a perf board. The PCB will prove to be more reliable and durable in comparison.
- Upgraded software will allow multiple solenoids to be powered at once; this will allow a proportionately faster refresh rate when switching the maps that will be represented on the MTB. For example: energizing four solenoids at once (the current maximum capability of the hardware), will allow the refresh rate to decrease by a factor of four.

Section-6: Conclusion

According to the World Health Organization (reference below), there are over 285 million people who are visually impaired: 39 million are blind and 246 million have low vision. Among that number, about 90% of the world's visually impaired live in developing countries. These statistics are as of October, 2011.

Knowing that over 4% of the world's population suffers from visual handicaps, we believe that the FuSe Map System can make a difference in these people's lives. We have worked passionately to design and construct an operating prototype that will meet the essential design requirements of the FuSe Map System. We have full confidence in what we have designed, built and tested, knowing that our product will be safe to use, reliable and capable of performing its intended functions: to provide guidance and assistance to the visual impaired, granting them more opportunities and options to pursue in the growing society today.

Reference:

World Health Organization, (October 2011), Visual impairment and blindness, Retrieved from http://www.who.int/mediacentre/factsheets/fs282/en/

Appendices:

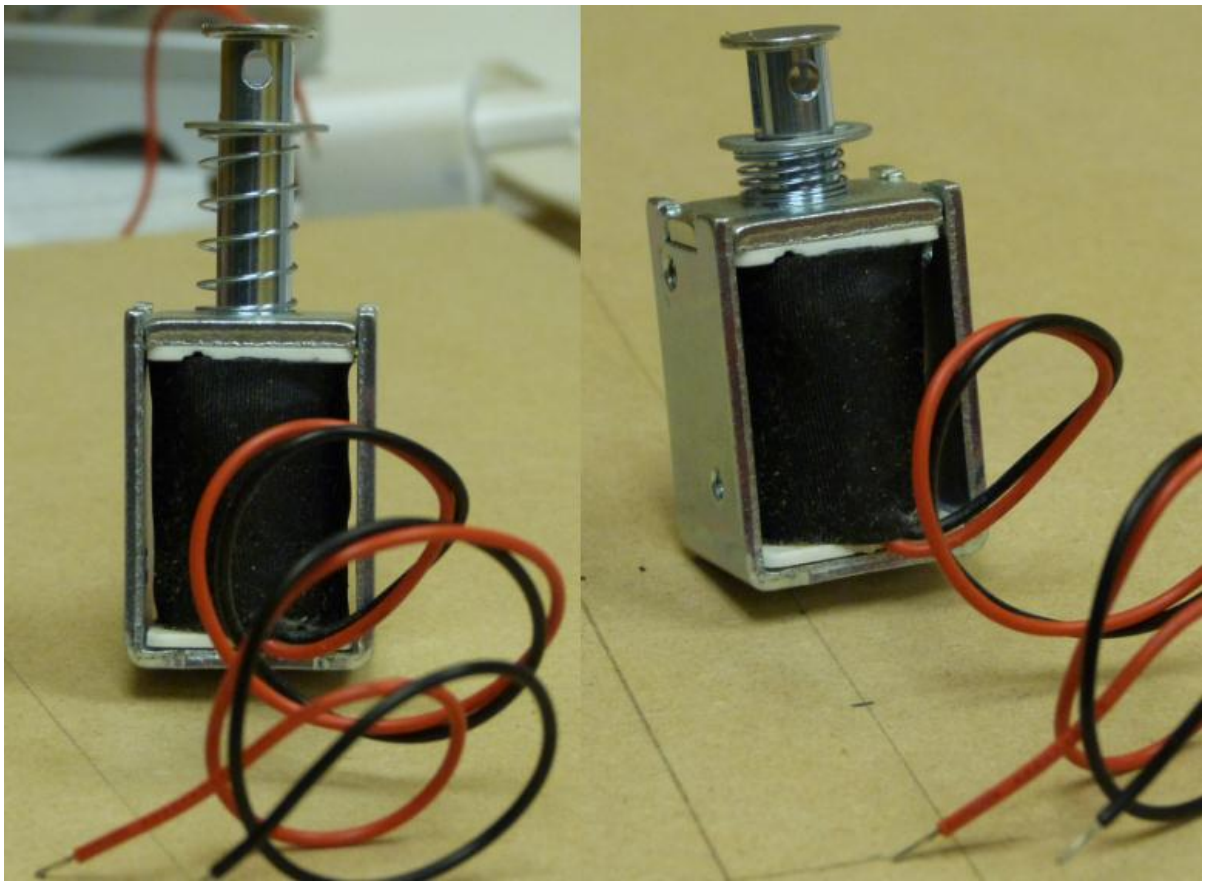
Appendix A: Overall Cost of all Parts

Vendor	Item	MFG #	Unit Cost	qty	Total ($)
Active Tech Electronics	2 Part Epoxy	8332-25ml	6.99	1	6.99
Digikey	IC OP Amp	LM741CNNS-ND	0.76	17	12.92
Digikey	IC MUX/DEMUX 4x16	HEF4067BP-652	2.6	3	7.8
Digikey	IC MUX/DEMUX 8x1	74HCT4051N-112	0.68	5	3.4
Digikey	Arduino Mega 2560	1050-1018-ND	63.67	1	63.67
Digikey	Power Supply 55W 5V-12V	179-2269-ND	44.64	1	44.64
Home Depot	3/4 steel tube	7718787369070	17.79	1	17.79
Newark	Ledex Latching Solenoid	20m0939	16.79	64	1074.56
Digikey	Rocker Switch	450-1040-ND	1.45	1	1.45
Digikey	Mosfet N-Ch	IRF520PBF-ND	0.7976	140	111.664
Digikey	Mosfet P-CH	IRF9510PBF-ND	0.7544	140	105.616
Digikey	PC Board 10X10	438-1020-ND	25.09	2	50.18
Digikey	PC Board 10X4	438-1036-ND	10.71	2	21.42
SainSmart	16 Channel Relay	20-018-103	28.99	2	57.98
Amazon.ca	Optoma PK201 Pocket Projector	B0039XRJ68	275.38	1	275.38
Deal Extreme	ND RE Magnets (100Pack)	13517	16.5	3	49.5
Newark	IC H-Bridge	15r0526	3.03	0	0
Newark	8 CH Wire Board Header	56h5848	0.79	16	12.64
Newark	8 CH Wire Board Receptacle	54h6822	0.67	16	10.72
Newark	Molex Wire Contacts	54h5523	0.216	128	27.648
Newark	HookUp Wire 20AWG Single core		4.62	5	23.1
Digikey	Diode Zener 12V 1W	1N474AFSCT-ND	0.1433	150	21.495
Digikey	6 CH Connector housing	wm2126-nd	0.81	1	0.81
Digikey	3 Ch connector housing	wm2123-nd	0.43	1	0.43
Digikey	Molex Wire Contacts	wm2313-nd	0.176	9	1.584
The Source	4 Line Rainbow Wire	278-0757	7.99	2	15.98
Home Depot	Aluminum Flat Bar	773204121030	9.98	1	9.98
Home Depot	1/2" Wooden Dowel	773294925126	1.42	1	1.42
Home Depot	1/2"x4x8 MDF	094561902996	26.25	1	26.25
Home Depot	Wood Glue	061083002510	3.99	1	3.99
Future Shop	Stereo Digital Camera		143	1	143
The Source	4-cond, 24-gauge solid wire (25')	2780757	7.99	1	7.99
The Source	18-gauge 2 cond, lamp wire (25')	2781250	11.99	1	11.99
The Source	22-gauge stranded speaker wire (100')	2781385	13.99	1	13.99
B&E Electronics	Chiselled Soldering Tip, 1.60mm	T18D16	12.95	1	12.95

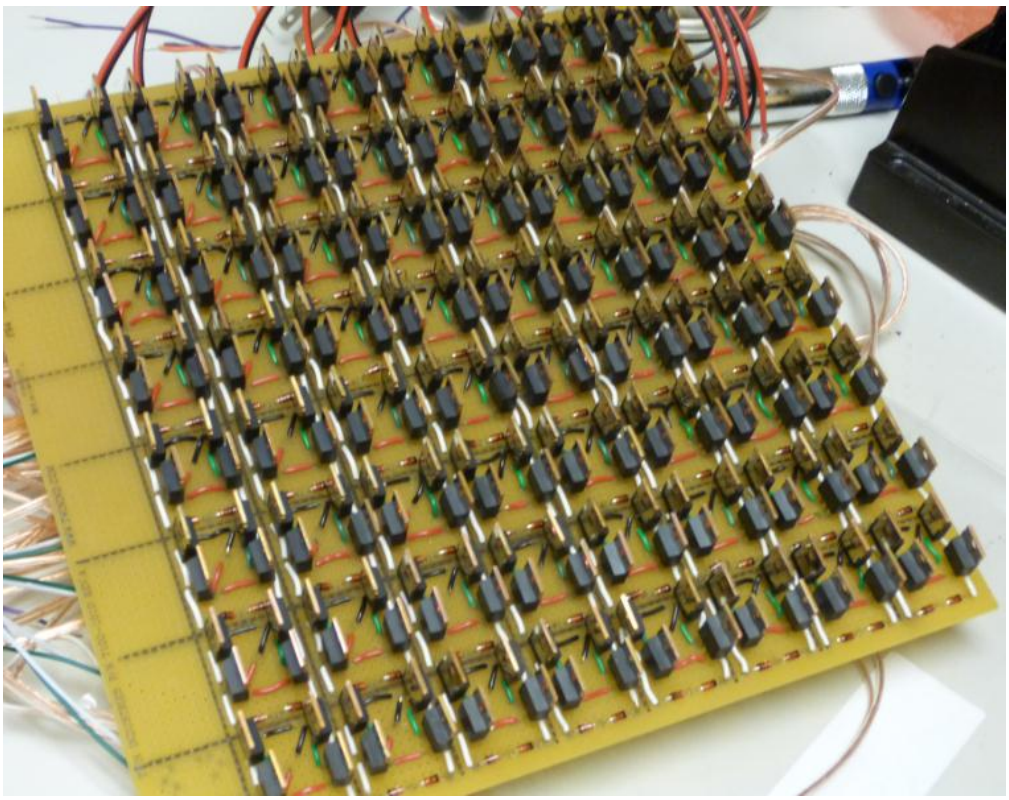
Stereo Digital Camera	143	1	143
Enclosure	50	1	50
Mounting Board	8	2	16
12V fan	25	1	25
USB Cable	5	2	10
PWR Cable 120V	2	1	2
Mounting hardware	25	1	25

Total Cost ($)	**2521.927**
W/5% GST ($)	**2648.023**

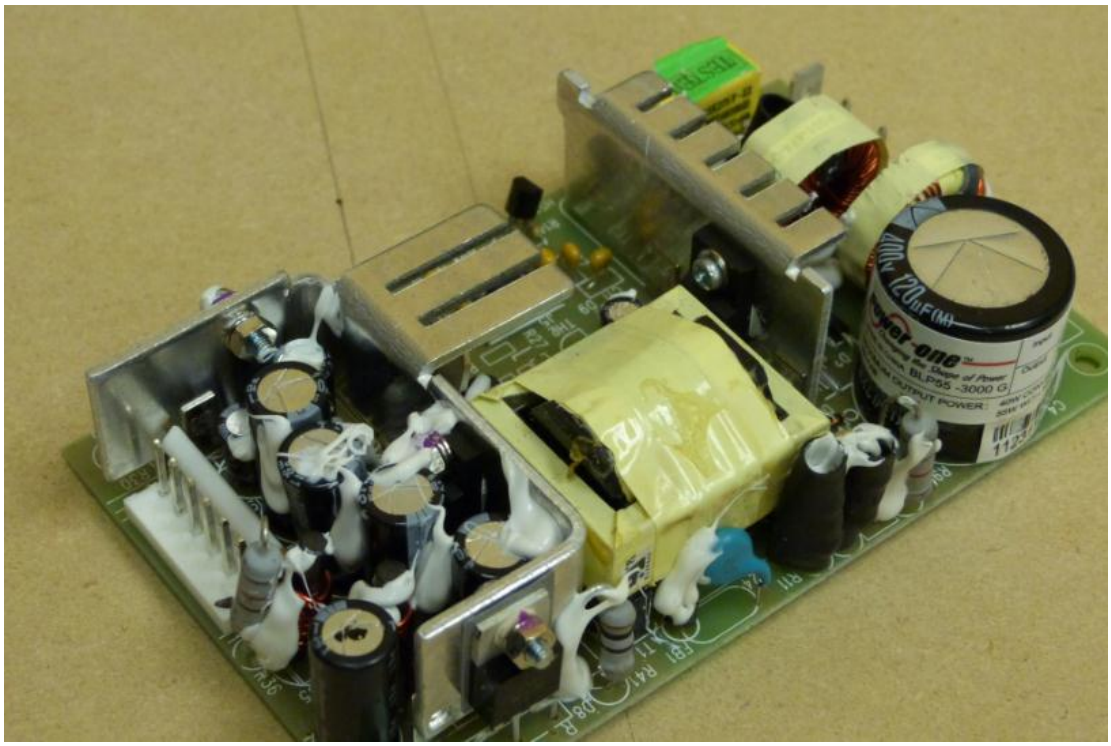
Appendix B: Solenoid (Unlatch & latch states) - B14-L-X55-B-4



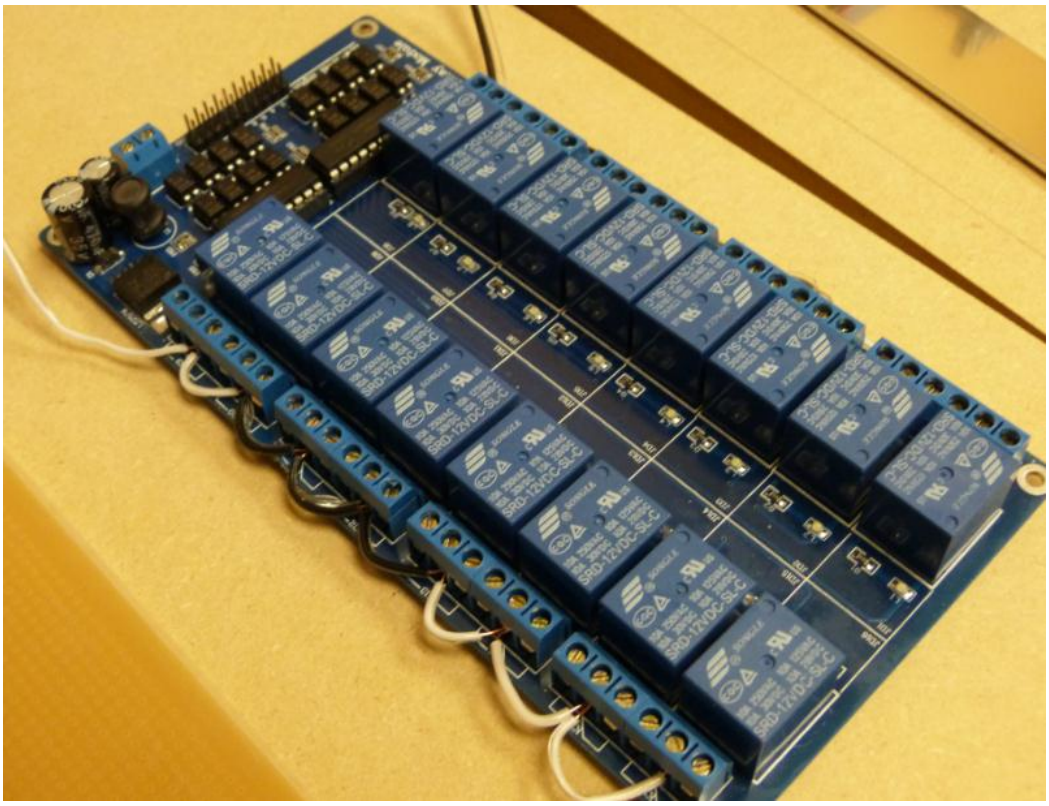
Appendix C: H-bridge circuit (8x8 grid)



Appendix D: External Power Supply



Appendix E: Relay



Appendix F: Arduino Mega 2560 Microcontroller