REPORT

ON

ELECTRONIC DRUM GLOVE, AUTOMATED LIGHTING, MIDI THEREMIN AND INTERACTIVE TABLE

BY

Shreesh Prasanna Kulkarni	2017B5A70279G	MSc. Physics + B.E. Computer Science

AT

MAKER'S ASYLUM, MUMBAI

A PRACTICE SCHOOL-I STATION OF

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

MAY 2019 - JULY, 2019

REPORT

ON

ELECTRONIC DRUM GLOVE, AUTOMATED LIGHTING, MIDI THEREMIN AND INTERACTIVE TABLE

BY

Shreesh Prasanna Kulkarni 2017B5A70279G MSc. Physics + B.E. Computer Science

Prepared in partial fulfilment for the
Practice School - I Course No. BITS F221

AT

Maker’s Asylum, Mumbai

A Practice School-I station of

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI
MAY 2019 - JULY, 2019

Acknowledgements

I would like to sincerely thank everyone at Maker's Asylum and everyone involved in the Practice School Division.

I thank Dr Pavan Kumar Potdar, our instructor, for his cordial support, valuable information and guidance. I also thank Mr. Suraj Gupta, our Student Coordinator, for his support and guidance throughout the evaluation period.

I thank Mr. Vaibhav Chhabra, for founding Maker's Asylum in the first place, and hence giving us an opportunity to intern there.

I thank Ms. Sudha Shukla, our SPOC at Maker's Asylum, as well as the Operations head for helping us with our day to day activities and projects, and to get used to the environment. I thank Mr. Sameer Ambre, Mr. Nehal Sarangkar and Ms. Ipsita, our mentors, for helping us out with all our projects, and supporting us through all the challenges and issues that came up in the way.

I thank my teammates at the station for providing moral and technical support whenever challenges appeared. I also thank my parents for being there for me at all times.

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE
PILANI, RAJASTHAN
Practice School Division

Station: Mumbai **Center:** Maker's Asylum
Duration: From: 21-05-2019 To: 13-07-2019
Date of Submission: 12-07-2019

Project Title: Electronic Drum Glove, Automated Lighting, Midi Theremin and Interactive Table

ID Number: 2017B5A70279G **Name:** Shreesh Prasanna Kulkarni
Disciplines: Physics, Computer Science

Name of Expert: Mr. Sameer Ambre
Designation: SpaceMan (Space Manager)

Name of PS Faculty: Dr. Pavan Kumar Potdar

Key Words: Arduino, Raspberry Pi, Sensors
Project Area: Electronics

ABSTRACT:

Workshops regarding electronics need a lot of groundwork to be done before they can be conducted. As interns of the company, we were to prototype projects that were to be included in the company's workshops, focussed toward fun learning.

Using an Arduino and a Raspberry Pi, we prototyped projects that explore sensors, digital interfacing and interactive electronic instruments that can be used to teach these subjects in a fun and interactive way.

Signature of Student
Date

Signature of PS Faculty
Date

Table of Contents

1: Introduction

1.1: About Maker’s Asylum

Maker’s Asylum, Mumbai, is a space for enthusiasts, hobbyists and professionals to come together and to make their ideas happen, founded in 2014 by Mr. Vaibhav Chabbra.

It provides access to an ecosystem of stakeholders which includes Governments, Businesses, Incubators/Accelerators, Investors and subject matter experts. The space houses various labs that are co-located in order facilitate prototyping of ideas that are interdisciplinary in nature.

The main office in Andheri has 6,000 sq.ft. of space, wifi, work tables, a rooftop, a pantry and labs dedicated to different aspects of design, engineering and manufacturing: woodworking, metalworking, electronics, 3D printing, laser cutting and modelling, and community brainstorming.

Maker’s Asylum has its flagship spaces located in Mumbai and Delhi and in October 2018 it launched the third space in collaboration with the French Embassy in Jaipur.

It works on a membership model, and holds a large number and variety of workshops to reach out to people and to include them in the maker’s community. These workshops teach the people how to use the tools and how to think like a maker.

1.2: The Internship

During the first few days of the internship the team at Maker’s Asylum tried to get us into thinking like a maker, first and foremost, then learn the tools of the trade.

The Founder, Mr. Chhabra, wanted us interns to then use these skills to prototype a few projects he had in mind. These prototypes were to be used in a new set of industry-oriented certification workshops to be conducted at Maker’s Asylum: Digital Fabrication, Woodworking, Metalworking and Electronics workshops. Most of our initial projects were thought of and modelled along these lines.

Our SPOC, the Operations Head at the office, Ms. Sudha Shukla, wanted us to contribute to the space in our own way. Be it building our own projects in our free time, or remodelling furniture, or serving everyone tea from time to time. A few of our projects also came up due to her input.

2: Rapid Prototyping Workshop

In the first week, as part of the internship, we were required to learn to use the tools available in the maker’s space.

We started with the Rapid Prototyping (RP) workshop on the first day itself (May 21). We were told that along with teaching us how to use the tools, this workshop would also test how much of the skills we learned we are able to put into practice. The training included: power tools for woodworking (day 1), engineering drawing that is used to plan projects (day 2), CAD software and modelling (day 3), using 3D printers and laser cutters for precise manufacturing (day 4), and electronics involving Arduino boards (day 5).

To test if we have accurately learnt the skills, we were told to make a "Useless box". If a switch on the box is pressed, a hand comes out of the box and un-presses the switch, then goes back into the box. Building this box tests out all the aspects of the training, since it involves first planning and modelling the box using CAD, then actually building a box out of some material, building hinges, supports and an arm using 3D printers or laser cutters, and finally setting up, coding and executing the function of the box using an Arduino, a motor and a switch.

2.1: Power tools learnt

1) Jig saw

2) Power drill

3) Circular Saw

4) Power Sander

2.2: Engineering Drawing

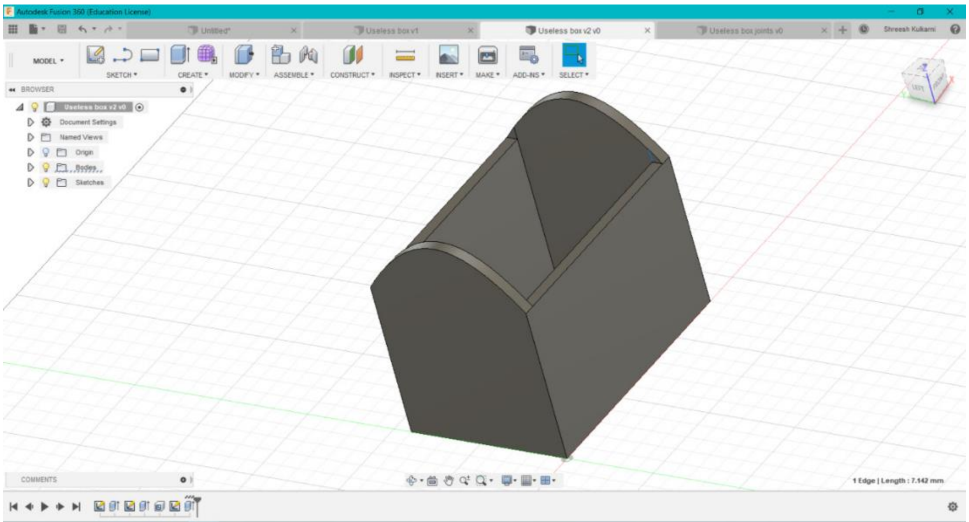
On the second day we learned the basics of engineering drawing: how to draw 1st and 3rd angle projections of objects. We were given Lego pieces to join together and create a small model. We drew the 3rd angle projection of this model.

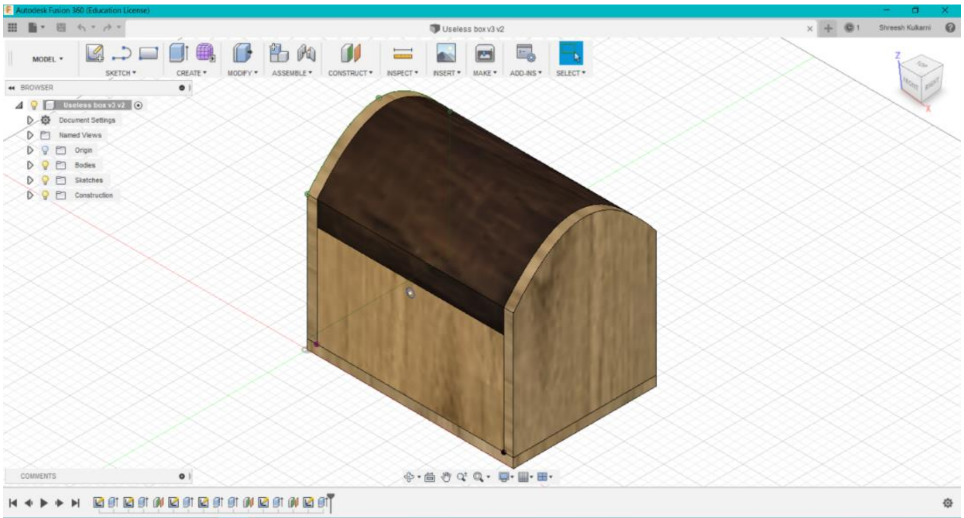
- Lego blocks

2.3: CAD Modelling

Next we learned 3D CAD modelling using Fusion360, a CAD software by Autodesk.

The model of the useless box I made:





2.4: 3D Printing and Laser Cutting

Next, we learned the basics of 3D printing and laser cutting. To learn 3D printing we were to make a small lithopane. We used an existing project on thingiverse to create a 3D model of the lithopane.[1]

We just had to feed the image that we wanted to convert to a lithopane, and the online customizer would let us download the files required to 3D print it. Next the file had to be sliced into layers (using online software) and uploaded into the 3D printer. We learned the methodology of operating the 3D printer, and the tricks involved.

The printer that we worked with is a Flashforge Creator Pro:

The lithopane I 3D printed:

After the lithopane was made, we shifted to laser cutting and laser etching. We learned the laser cutting software, RDWorks, and how to vectorize images using online software. Then we learned the methodology of working with the hardware, and the precautions involved.

We were able to etch logos on plywood. The logo I etched:

2.5: Building the Useless Box

Once we were done with learning the tools, we got to cutting the MDF sheets and putting together the box. The skills learned in the woodworking lab were put to use, as we used the jig saw, circular saw, the power sander and a hand drill. We (me and Aditya Kapoor) ended up with this:

The sides of the box were made using MDF (medium density fibreboard, an engineered wood product made by breaking down hardwood or softwood residuals into wood fibres, often in a defibrator, combining it with wax and a resin binder, and forming panels by applying high temperature and pressure.)

For the top of the box, my team mate (Aditya Kapoor) and I had planned to use a living hinge, inspired by some pieces kept on display in the maker’s space. A living hinge uses carefully engineered laser cut slits in a thin material (like MDF) to provide pliability. This makes it bendable to a large extent, and also provides a unique, aesthetic look.

After designing and executing the living hinge on and MDF sheet, we found out that our slits weren’t wide enough, and hence it wouldn’t bend. After a bit more experimentation and a lot of thought, we figured that changing the design itself would waste a lot of time, and gave up on the idea of a living hinge.

We resorted to using a traditional metal hinge to connect the box and the laser cut MDF sheet that we earlier had planned to use as a living hinge:

Once the box was made, we added supports for a servo motor, and chiselled out a slot in the MDF for a small two-way switch.

2.6: Electronics

Now, we started working on the electronics part: the Arduino.

AN Arduino Uno Board

We first learned how the microcontroller board works: it is simple in that we only need the Arduino IDE to code files and upload them onto the board, which the Arduino executes. We learnt the syntax of the Arduino IDE language, which is derivative of Java, a language most of knew beforehand.

Once we learned how to set up and operate switches and motors using the Arduino, we connected it to the useless box and tested it out. After tweaking some values (regarding rotation parameters, etc.) we were able to make the box work according to our expectations.

A Servo Motor

In the next chapter, we'll be looking at the projects that were assigned to us once we had finished the Rapid Prototyping workshop, and had become comfortable with the workspace.

3: Arduino Drum Glove

With this project, we're taking something an action as basic as tapping your fingers on the table and turning it into real-time sound. Users would be able to just slide on this glove like any other, and start tapping away as the rhythm of their fingers produces drum beats from a laptop. There are two levels to this project: an Arduino implementation, and a Raspberry Pi implementation. The Arduino implementation was jointly handled by Aditya, Nirmit, Sanjana (my teammates) and Shreesh (me).

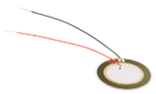
3.1: Basic aim of the Project

Five piezoelectric sensors represent five fingers on the glove. These are connected to an Arduino Microcontroller. Output is in the form of drum-kit sounds on the speaker of the laptop that the microcontroller is interfaced with. Each finger has a unique sound associated with it, and completely programmable.

3.2: Materials Needed

- An Arduino microcontroller

- A bread-board for all connections
- Connecting pins and wires
- Power source
- 5 piezoelectric sensors for each finger on the glove (later changed to 4 washers with jumper cables soldered onto them)



- A glove that fits over your hand

The source code for the input signals was edited and compiled on the Arduino IDE© (Windows x64). The source code and integration for the output was compiled on Processing© v3.5.3, and Python v3.6

3.3: Order of execution

***Step 1:** This mainly included brainstorming ideas and flows for our plan, choice of IDE and sensors. We did a brief comparison amongst piezoelectric sensors, pressure sensors, simple switches, and light sensors. Eventually, piezoelectric sensors were chosen because of their ease-of-use and sensitivity.

***Step 2:** In this next step, one individual sensor was used with the Arduino to test the functioning. We received a binary output on touching a finger to it. This confirmed its use in the tip of the glove, that is, it will trigger a response when tapped on the table. Only the Arduino IDE was used in this.

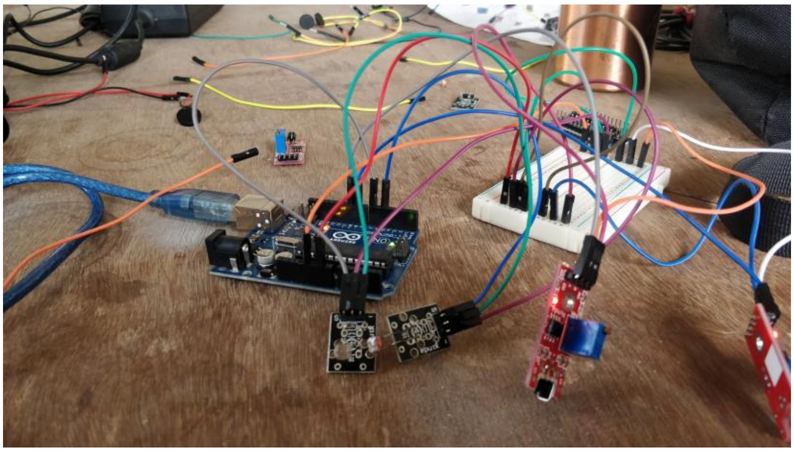
Step 3: On the second day, we programmed on the Processing© to use the input from the Arduino to produce drum-kit sounds on the laptop’s speaker. We imported .mp3/.wav files to the program to link our code to the output, however, we faced a NullPointerException during execution. This revealed that the program/GPU couldn’t locate the necessary drivers to utilise.

***Step 4:** Since the software mentioned above had failed to integrate the input and the output, we decided to try another approach. We have now used a Python script to input and decode the serial output of the Arduino. Using this decoded input, we find out which sensor was pressed. We played the corresponding file using winsound and playsound, pre-existing python library files. We faced a problem, namely the .wav/.mp3 files were unable to be played with this library due to some issues that we discovered only later. In the end, as a compromise, we decided to prototype the final response using beeps of different frequencies instead of different files, until we were able to finally play the sound files.

***Step 5:** On day 3, we fixed the problem with being able to play sound files; there was a permission access problem that was easily solved. Once we were able to play the files, we ran the code and got the expected response from the Arduino.

***Step 6:** Then we soldered the piezoelectric sensors and tried running the code again. When the sensors weren’t triggering the sound files, we realized that the specifications of the required piezo didn’t match with the one which we were using. Hence, we ordered new ones.

***Step 7:** As we discovered earlier, the piezo-electric sensors we were using were rated at 9V, not according to our specifications, and we needed sub-5 V sensors (the voltage at which an Arduino operates), and hence, they malfunctioned. We decided to try out touch sensors and light detecting sensors (LDRs). Unfortunately, we found that LDRs are susceptible to changes in the surrounding temperature, and the readings we recorded were inaccurate.



Step 8 : Finally, in order to simplify the working of the prototype, we decided to use switches. Only 4 fingers can act as triggers; we press the fingers to a grounded terminal that is attached to the thumb. To play sounds, we touch the fingers to the thumb. This idea made it to the final *working* model. The triggers, the Arduino and the wires were attached to a glove and we were able to wear it and play a few beats on the speaker.

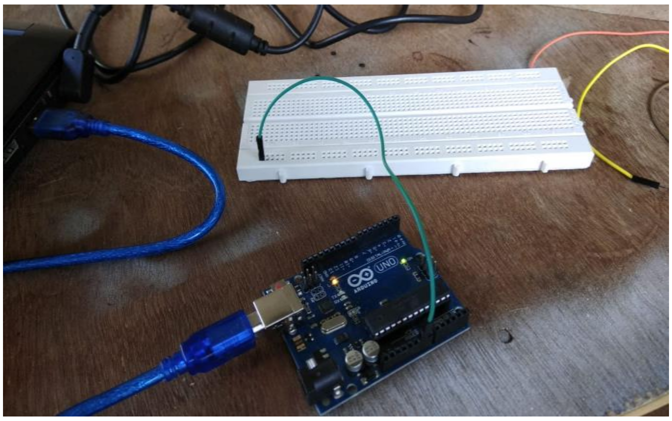
3.4: Teaching Methodology

The steps that are starred are the steps that need to be taught to the ones who are taking the course. Apart from this, basics about Arduino coding and python need to be taught on the first day itself. Interfacing between the Arduino and the PC is one of the major challenges, and the students have to navigate this challenge with the help of the instructors of the course.

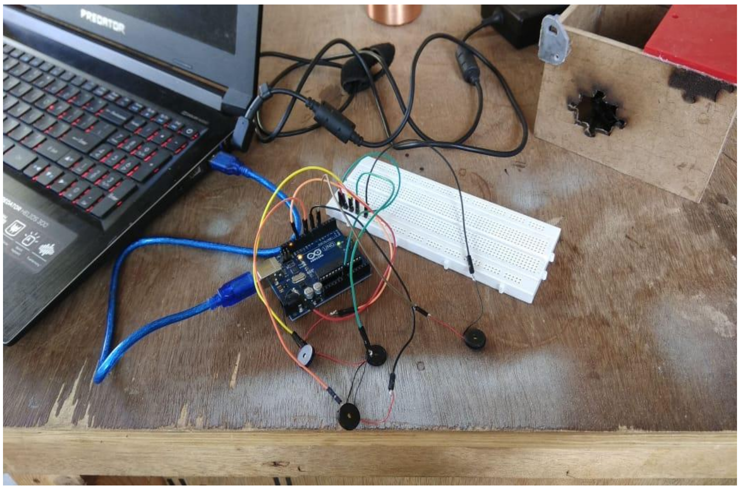
3.5: Learning Outcomes

When we started with the project we thought of the workflow as being very straightforward. The components were simple: an Arduino, a few sensors and a laptop. The glove was only required at the last stage, not to make the prototype. Once we actually started researching and working on the code, we realized that triggering actions on the laptop using an Arduino was our main challenge. We learnt how to interface between the Arduino and a laptop using python scripts. We learnt to work with and solved unexpected coding challenges that appear on the go. We also realized that the specifications of certain components are very important to the working of the project. Finally, now we can say and be sure that if any other project involves this kind of interfacing, we can make it happen quickly and without encountering similar problems.

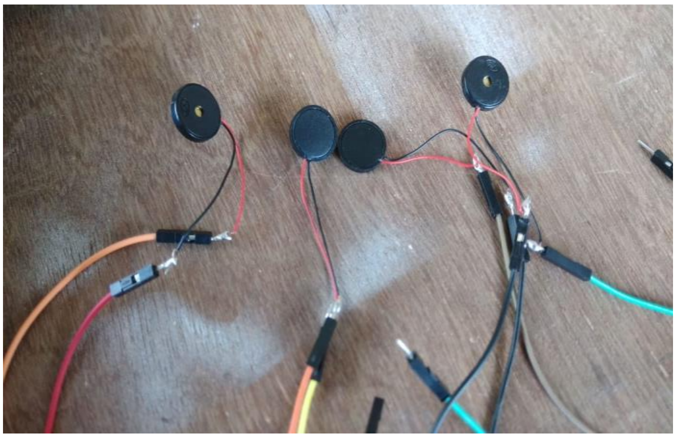
Arduino connected to the breadboard:



Sensors connected to the breadboard



Piezo sensors



4: Raspberry Pi Drum Glove

Once we were done with making the glove using an Arduino, we wanted the glove to be more of a standalone piece of equipment, only requiring a power source and a speaker. For this, we used a Raspberry Pi, a sort of computer on its own. This part was handled only by Shreesh (me).

4.1: Materials Required

- Raspberry Pi Board

- Keyboard, Mouse and Screen
- SD Card, SD card Reader
- Jumper Cables
- Solder wire, Solder Gun
- Washers
- Wearable Glove
- Tape

4.2: Order of Execution

Step 1 (Setup): On day 1, I received an RPi. I prepared an SD card for the storage requirements (formatting etc.) and downloaded the Raspbian OS in the form of NOOBS software. Once I transferred it to the SD card, I inserted it into the RPi and installed the OS. Downloading and installation took the most amount of time.

Step 2 (coding): Once the setup was done, on the second day I wrote a python script to use the GPIO pins as triggers. In the script, I take input from separate pins to figure out which music file to play. This script also uses the pygame module (pre-installed as part of the python library) to play sounds. For initial testing purposes I chose to print output rather than play music.

Step 3 (testing): I had to figure out how the GPIO pins are triggered. I wrote a separate script to figure out the behavior of each pin, as I was encountering garbage values. After testing, I found out that only 2 pins sensed a High input by default, while the rest sensed garbage values by default. With this began the quest to get rid of these garbage values and try to force the pins to sense only one value by default.

Step 4: (I/O) On the third day, I first tried a way to "pull down" the voltage value read by the input pin (so as to make it read 0 by default). When this didn't work, I tried to create a sort of switch mechanism to force high or low values as input. This would end up shorting the 3.3V and ground pins of the Pi, so the idea of switches was rejected.

Next was the idea of assigning 4 pins 4 different bit streams of 4 bits (0000, 0001, 0011, 0111) which the input pin could read and decode (based on number of 1's in a group of 4 bits). The 4 different pins could be assigned to the 4 different sounds. The pin that reads 1 as default could be used to differentiate the 4 inputs.

Implementing this idea would require using multiple threads to output the bit streams continuously on 4 pins, a highly impractical idea on an RPi. Hence, I went back to a simpler method.

Step 5: I went back to simply reading from 4 GPIO pins, but I used one pin as a permanent HIGH output. Almost miraculously, this solved the garbage value problem. I compiled a few drum samples that worked with the pygame module. Once the testing was done, I replaced the Arduino on the glove with the RPi, made the connections to the sensors, and the working model was complete.

Step 6: To pull it all together, I modified the .bashrc file of the RPi to run the python script whenever the RPi is switched on. Keyboard shortcuts can be used to close the python program, and to shut down the RPi. After doing this, only a speaker and a power cord is required to actually run the drum glove.

I modified the code a bit so that the delay between a finger touch and the sound playback is reduced, and one finger touch only plays the sound once. With this, the final working model was complete.

4.3: Challenges faced

A different platform meant a different set of problems. As the sensor issues were fixed, I didn't have to focus on that. The main issue was the garbage values being read by the RPi pins. The final issue was loose connections of the jumper cables, which was easily handled by getting sturdier cables.

4.4: Learning Outcomes

I learnt how to deal with binary output of the pins. A lot of brainstorming was involved with regards to getting around the GPIO pins' behavior. I became comfortable with the RPi environment (and the Linux environment), as well as python's pygame module.

4.5: Teaching Methodology

The RPi implementation serves as a logical step forward from the Arduino implementation. This time around, there is no interfacing between the glove and the PC; the glove itself is the PC. As a result, one of the major steps in the Arduino implementation is missing in the RPi implementation.

The students, by this time, must have become comfortable with dealing with python scripts and using the pins as input/output. Hence, this time we should focus on making them comfortable with the Linux environment, as the Raspbian OS is based on Linux.

Using the Linux terminal to navigate files and to run programs/scripts should be taught right after setting up the RPi. The python script used by me doesn't properly run on the IDLE terminal, so running it on bash terminal is essential (hence the focus on the bash terminal).

It is recommended for the students to come up with their own ideas to implement the logic for triggering sounds with the fingers. They may also encounter the garbage value problem with the GPIO pins, and if they get stuck on it for too long, they'll need to be told of the trick that I used. If they can come up with a better solution, so be it.

The 16 bit .wav sound files can be provided directly to the students, or they may search for them online and download them.

At the end of the day, the students will learn how to deal with input and output on an RPi and how to work on Linux terminal.

A picture of the final glove:

5: Automated Lighting

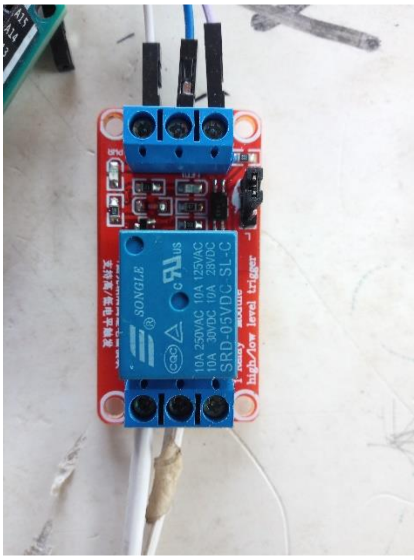
This project aims to use an Arduino board to switch a bulb on or off based on the entry or exit of a person into a room. The idea is to get rid of traditional switches and replace them with electronically controlled relays, whose action is controlled by the same circuit that handles the detection of human motion.

5.1: Materials Required

- Arduino Board
- Relay: to interface between the electronics (~5V DC) and the electrical circuit (~220V AC)
- PIR Sensor/Ultrasound Sensor: to detect human presence/motion
- Standard Electric Bulb
-

The first part involves the use of a relay, something that takes digital input (with a max voltage of 5V) and uses it to drive an electromagnetic switch. This switch is part of an AC circuit that consists of the bulb/tubelight/fan etc.

The relay:



The second part requires the use of a certain sensor that takes advantage of human characteristics, like our tendency to move, the heat given off by us, etc. The viable sensors are:

- Passive IR sensors (PIR sensors, also called motion sensors)
- Active IR sensors
- Ultrasonic sensors (radar based)

Out of the three, the first one is very cheap and easily available. The principle on which it works is that humans emit IR light, due to their average body temperature of 37 degrees Celsius. The PIR sensor uses a combination of Fresnel lenses to aggregate light from an effective angle of 30 degrees onto an IR detection module.

5.2: Execution

In my initial stages of testing the PIR sensor, I found out that this sensor is very prone to background noise; any IR light with wavelength in the range of 10-20 microns will set it off.

PIR sensor:

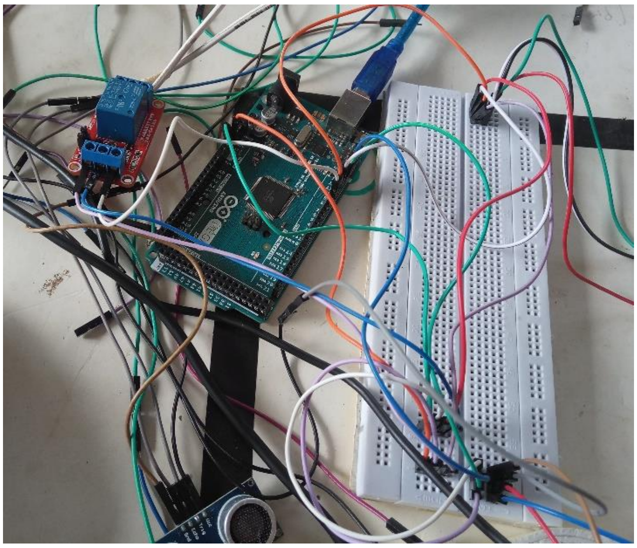
Once the PIR sensor was rejected as a viable sensor, I set out to work with two ultrasonic sensors. These sensors work on a basic principle: sonar. An ultrasound pulse is emitted by the ‘trigger’ speaker, and we read the reflected pulse with an ‘echo’ microphone. The time taken for the pulse to return to the sensor can be used to find the distance of an object/screen from the sensor.

The ultrasonic sensor:

The idea is to keep two sensors on two sides of a door. Normally, they would both read how far away the wall is from them. When a human enters the door, the distance read would reduce, for both the sensors. By carefully calibrating which distances trigger a response and the delay in taking consecutive readings, we can detect the motion of a human passing through the door.

Once the sensor side of the circuit was tested, I was able to hook up the relay. This relay works on output from the Arduino. This output can be triggered specifically when a human presence is detected by the ultrasound sensors. Whenever a human is detected, the switch is flipped, which means if a human enters the room, the switch is switched on, while if they then exit the room, the switch is switched off.

The circuitry:



5.3: Challenges Faced

As mentioned, the PIR sensors weren't able to give a reliable response, due to the noise sensitivity issue. These sensors are meant to be analog sensors, and hooking them up to digital circuitry is not the way to go.

The calibration of the ultrasonic sensors was also an issue, which had to be done separately for each room. Since it was detecting negative distances (which are actually distances too large for the unsigned integers to handle), I had to design a workaround to neglect such values.

Finally, attaching the sensors and the electronics to a room was the final challenge. I solved this one by using double sided tape, just for this prototype.

5.4: Learning Outcomes

I learnt a lot about sensors, especially the ones used to detect distances and motion. I learned how to interface between low DC voltages and high AC voltages, something very important for home automation.

5.5: Teaching Methodology

We should start with the basics of Arduino, which should take around a day. Next, input/output specific to Arduino can be taught, which flows well into what kind of sensors the students want to use. The experimentation must be left up to them; all the sensors must be made available to them. It is up to the student to figure out which one works best, and how to optimize the input. Next some basics of AC circuits can be given, then the relay introduced. Finally, it can all be wrapped up. This project can teach students the interfacing between AC and DC circuits and basics of the Arduino.

6: MIDI-Arduino Theremin

The idea of this project is to create a Theremin with materials available in the space, namely an Arduino, a laptop and sensors.

A theremin is a musical instrument that detects the presence of a hand and measures its distance from two perpendicular axes using radio sensors. One axis represents pitch and the other represents volume. The continuous detection of position in the 2D plane allows for a continuous range of frequency and volume, giving it a characteristic sound.

The key attribute of a theremin, a sort of marker, is its use of radio sensors. This makes it a rare touchless instrument. With its implementation with an Arduino, I wanted to duplicate such behaviour. Without the availability of radio sensors, I had to find alternatives. One reliable alternative is an ultrasound sensor. This theremin was made primarily with ultrasound sensors as the main focus.

A Conventional Theremin

6.1: Materials Required

- An Arduino
- Two ultrasound sensors
- A small board of Plywood/MDF
- Jumper cables
- Solder wire + solder gun

6.2: Execution

With previous experience with ultrasound sensors (from the Automated Lighting project), I was able to seamlessly integrate them into the programming for data input. I had to now figure out the output part of the project. Every Arduino comes with a provision of outputting serial data. Initially I thought of using Serial to communicate with a Python script on the laptop. Since python (or most programming languages) are not designed to handle sound very efficiently, pitch bending was not possible with python. Hence, I discarded the idea.

Ultrasound Sensor

Next I looked to my music production software, FL Studio. Pitch bending is frequently done on synths and samplers; they're best suited to dealing with changing parameters on the fly. But the challenge of communicating with music software is that we have to use MIDI, a serial mode of communication. Since Arduino can communicate serially, MIDI is the best option for interfacing between the electronics and the software.

FL Studio interface with an oscillator synth loaded

The next step was to understand and implement MIDI serial communication. There are two pieces of software necessary to read

MIDI data from an Arduino: a serial to MIDI (SM) converter and a MIDI port. The SM converter packages the binary data into a stream of bytes and passes it on to the MIDI port, which acts like a simulated 5-pin hardware MIDI port. This MIDI port can be connected to a Digital Audio Workstation (DAW), for example, FL studio. Through the DAW, the MIDI data can be assigned to different knobs on various synthesizers. These knobs are used to play notes, change volume and bend the pitch of the played notes.

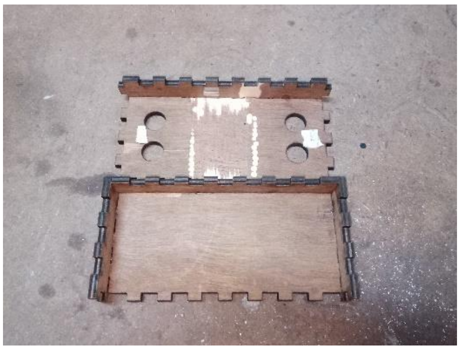
Once I understood the interfacing, I had to test my understanding by passing some basic MIDI data from the Arduino. I wrote a test file using a few MIDI libraries. When this did not work, I decided to learn the MIDI language and talk to the software on the bit level. I found a very good resource online.

Once I had learned the MIDI language, I ran the test file again (with modifications) and found out that this is the most efficient method of transmitting MIDI data (directly outputting binary data).

After writing the code for different actions (switching notes on and off, volume bending, pitch bending) and figuring out the logic (how to get a seamless playback), I tested the pitch response and the volume response separately. Both were seamless after some changes in parameters.

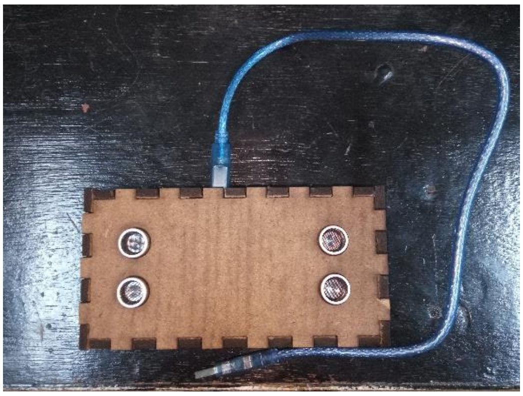
The problem arose when I tried to merge the pitch and volume response. I had to make some major changes in the code. Mainly, I had to adjust the range in which the ultrasound sensors were to sense both the distances. I also had to assign the MIDI channels properly to the knobs on the synth, something which took a lot of trial and error due to the large number of channels (700 or so).

The working prototype was made, and next was to finalize the design. To make it presentable as an instrument, I started work on a casing for all the electronics and sensors. I used the online tool “maker case” to create a blueprint for a laser cuttable box. After cutting the sides of the box, I realized that the dimensions were completely wrong; the Arduino wasn’t fitting inside.



The defective case that I tried fixing, to no avail

After trying to fix it by sanding the impeding material, I realized there was too much material to take away for the Arduino to fit. I then laser cut another box, making sure there was enough space for the Arduino as well as the wiring to fit. Once this box was made, I finalised the code and the casing and gave final touches to the whole thing.



The final, working model

6.3: Challenges faced

The main challenge in this project was learning MIDI interfacing. Next was implementing synths in FL studio. I learned a new facet of music production, namely digital interfacing. As this was my own project, I also learned how to apply the skills I learned on the way to full effect, and how to manage time and set deadlines.

6.4: Learning outcomes

MIDI interfacing and FL Studio DAW environment. It was also fun learning to play various songs using the instrument!

7: Interactive Table

An interactive table is a table or a surface which is able to display some content and allows for user input by means of some interaction paradigm. As our final project, we were given the task of designing an interactive table using LED strips, which could be controlled using a keyboard. It was a fun project where we had to make a functional table on which the snake game could be played.

7.1: Aim

The goal for this project is to create a functional interactive table, which can be programmed to play arcade games. The aesthetics of the table are almost as important as the functionality. Our personal goals are to improve our skills in mechatronics and to have fun while doing it.

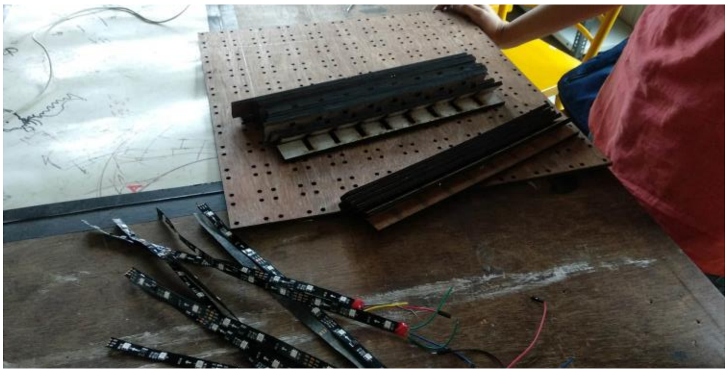
7.2: Materials and tools required

- Plywood
- Arduino UNO
- A-4 sheets of paper
- WS2811B 5050 RGB LED Strips (9 x 9)
- AC to DC converter
- Soldering machine and solder wire
- Jumper cables
- Fevicol glue
- Jigsaw
- Laser machine
- Glue gun

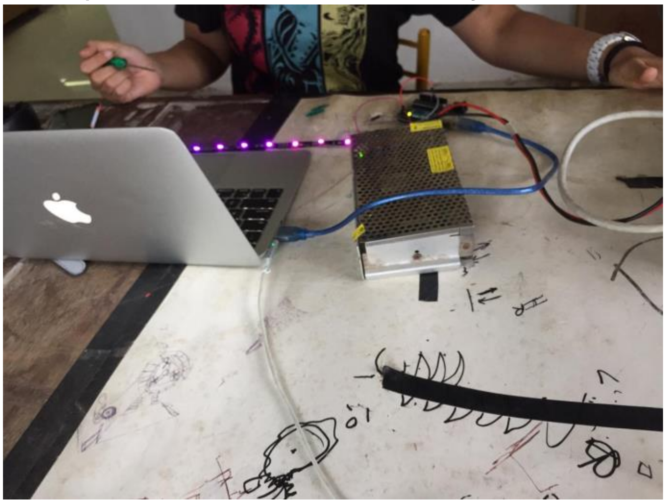
7.3: Order of Execution

Step 1: We started off by deciding the dimensions of the table and the number of LEDs we would require to do so. We agreed on constructing 9 x 9 compartments for the table, wherein, each compartment would comprise of one LED each (a total 81 LEDs). Therefore, we had to use 9 LED strips of 9 LEDs each. The dimensions of our table were 33 x 33 (in cm).

Step 2: After deciding upon the dimensions, we started with the woodwork. The plywood was cut using the jigsaw. The walls of the compartments and the borders of the table were cut using the laser machine.

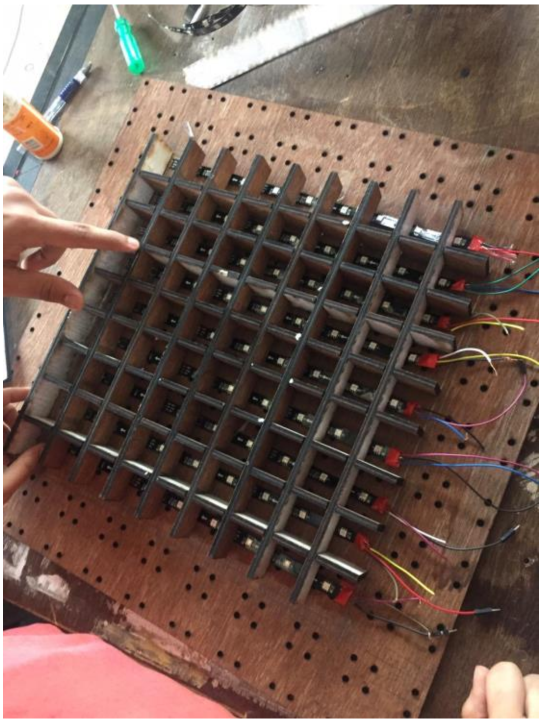


Step 3: We started cutting the LEDs and soldered 4 wires to each LED and hot glued each connection. But after soldering around 15-20 LEDs, we realized that the soldered connections were not strong enough and kept coming out easily. Also, soldering the wires was very time consuming. So instead, we chose to use the LEDs in premanufactured strips and not individually.



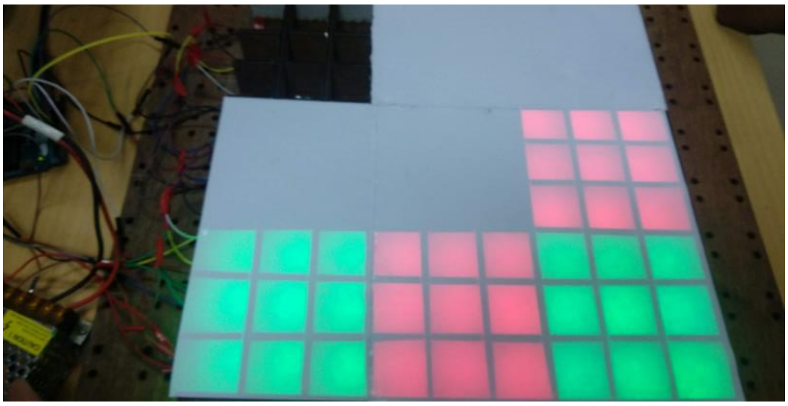
Step 4: We had a long strip of LEDs, which after testing, had to be cut into groups of 9 each. We did so (cut 9 strips) and soldered jumper cables to the connections of each individual strip, applying hot glue and tape to secure the solder. After testing each strip, we stuck them onto the wooden base, on top of which we assembled the compartments and the borders that we had laser cut.

Step 5: We shorted the power and ground connections of each strip and connected it to a common power supply (10V DC). Once all the connections were secured, we started discussing the logic of the snake game.

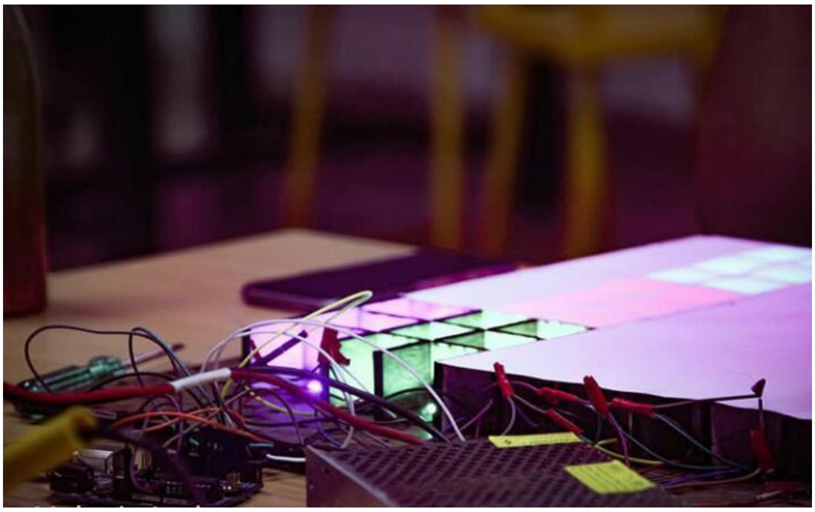


Step 6: The code was then finalised and uploaded. However, once we ran the code, we realised that the LED strips could be programmed only in the groups of three. This made the snake game impractical on this board and hence we decided to make a tic-tac-toe game instead.

Step 7: For the tic-tac-toe game, the basic circuit remained the same but the code had to be changed completely. Once the necessary changes were made, we tested the code. The code was working properly, but we noticed that one of the LED strips was not glowing. We realised that the wires soldered to it had come out. We soldered it again, but it still didn't work as required. We concluded that its contact was burned and hence we had to replace it.



Step 8: After changing the strip, we ran the code and it worked perfectly. Our mentor wanted us to also include an option of getting input from a keypad, but due to time constraints we could not finish this part of the project. Hence, we concluded the project as it was.



7.4: Learning Outcomes

This project was our mega project where we could use all the skills that we had learnt at Maker's Asylum. We used the laser machine, the woodworking lab, the soldering iron, Arduino boards and LEDs. We acquired troubleshooting abilities. Since all seven of us worked on this together, it helped build team spirit among us.

8: Miscellaneous contribution to the MakerSpace:

8.1: Remodelling and painting a cupboard

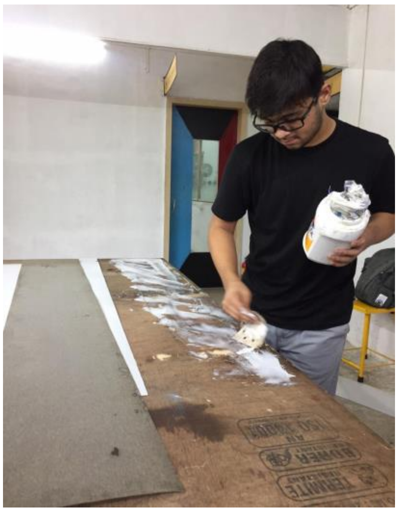
An old, broken cupboard in the space was assigned to us to be fixed. First, we turned the cupboard upside down (it was being stored upside down in the first place). Next, we measured the dimensions of the base and shelves and cut plywood accordingly. The shelves and base were glued inside the cupboard in their positions. Then, we applied two coats of paint to give it a new look!

8.2: Badges for SDG School

Before the founder left to Paris for Maker's Asylum SDG School, preparations for the trip were in full flow. We were asked to paint the badges which would be given to the participants at the school. These were then laser cut and pins were hot glued to each of them after they had been separated.

8.3: Covering a table with Sunmica (veneer)

The table that had been assigned to us to use as a workspace had to be finished with Sunmica. First, we removed everything that had been stuck on the table and sanded the entire table so that it would have an even surface. Next, we stuck white and red Sunmica in an abstract pattern after cutting the strips. We then placed heavy objects over the table and left it overnight to dry completely.



8.4: Inventory for the office:

We were given 6-7 crates of stationary and other items that are used in the space for workshops and programs conducted by them. We had to count the number of each item in all the crates and organise similar items in the same crate. We then had to write a list of all items present in the crate along with the count and stick it on the crate. Along with the crates, we were given an assortment of building materials (sheets of acrylic, plywood and MDF). We measured the dimensions of these sheets and noted them down on paper.

9: Conclusion

All the projects that we made helped us to learn a lot about electronics, basic coding with Arduino IDE and python, and various methods of crafting. The use of LED strips in the circuit proved to be a very important aspect in many of the projects. The importance of designing a device and planning it before implementation is the biggest takeaway from all these projects.

We also designed a teaching algorithm for an electronics project, that the science enthusiasts who come here to make, break and create can do so with the help of the same. The Projects we made are to be used to teach the processes involved in the workshops to be held by the company. The games and devices made have been displayed in the space and are used for entertainment purposes for the people who visit and the spacemen at work!

We got the opportunity to use machines like the 3-D printer and laser cutter which are not only difficult to find but also very expensive to learn. From making our own games to cutting our wood, these projects gave us a hands-on experience.

Personally, I learned a lot about digital communication, design as well as learning and teaching methodology. As a dual degree Computer Science student, the things I learned here will be of great value, and will help me to relate to my academics in the coming year. I’m very thankful to the station for inculcating values of an engineer and a designer in me. Terminating my period of internship is a little disheartening but some good memories to look back at.

10: References

1: https://www.thingiverse.com/thing:308109
2: https://www.instructables.com/id/SNF-Drumming-MIDI-Glove-using-Arduino-and-light-se/
(Arduino drum glove)
3: https://makezine.com/projects/make-33/simple-soundboard/
(raspberry pi reference)
4: https://create.arduino.cc/projecthub/jake/how-to-use-an-ultrasonic-sensor-with-an-arduino-63527b
5: https://ym
/www.cs.cmu.edu/~music/cmsip/readings/MIDI%20tutorial%20for%20programmers.html
6: https://en.makercase.com/
7: https://github.com/adafruit/Adafruit_NeoPixel