Build Instructions

Parts Crib Database

Group Name: The Walking Programmers

By: Masoud Rahguzar

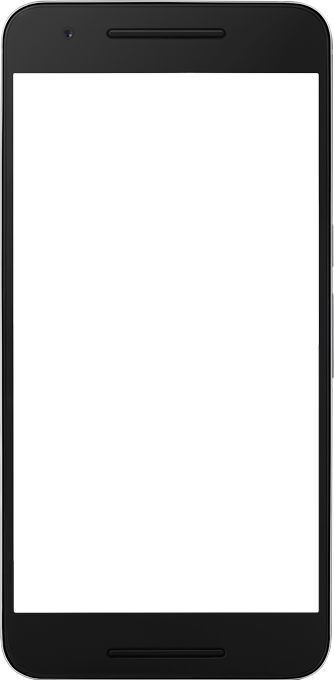
Date: December 4 2016

**Introduction**

The project I decide to work on is the Parts crib database. The reason why I choose this project is because I felt like it met all my capabilities. The purpose of this project is to create a quick and efficient way to take out items from the parts and store them in a database. The group members I decided to work with on this project are Rafel Yashooa, Divesh Oree and Gurpreet Jhita.

**System Diagram**

The humber parts crib database project functionality is to be able to scan barcodes on student id cards and the associated part items from the parts crib and then be placed in a database. Now understanding what the concept of the humber part crib database project is, the system diagram can be easily interpreted.







Process

Output

Input

**Bill of Materials/Budget**

The main materials/components required for my project are a raspberry pi, webcam and barcodes. Added features/materials are PCB for light indicator, sound bar and acrylic box. The PCB and acrylic box were both provided by the school. In addition, the raspberry pi requires a monitor, keyboard and mouse to be setup. My budget was fairly simple and not expensive except for the raspberry pi kit. My full excel version of my budget is provided in my GitHub page.



**Time Commitment**

The time commitment for my project took about 15 weeks to complete in its entirety. Firstly, the project began with ordering parts through websites like amazon, eBay and all sorts of technology base websites. Once ordered, the delivery took about week and a half (week 2 & 3) to come in. As everything arrived, I then began to setup my raspberry pi and all of its components which took my about 1 hour to setup. As week 5 approached I printed out my PCB and started soldering everything. The soldering approximately took me about 2 hours to complete at school. Afterward, at week 6 and week 7 I started to test my raspberry pi and its components which took me about 6 hours in its entirety. Firstly, I tested if the PCB by implementing the code given by the teacher that can test if the lights work and the sensors. Then I started implementing the code in which will scan barcodes with my webcam which then again took me about 5 hours to complete. Next, during week 9 I created I remote desktop with my raspberry pi and laptop so I could connect it remotely without using an external monitor, keyboard and mouse. Lastly, during week 12 and 13 I began and created my acrylic box, the box design took me about 1 hour to complete and the lazar cut took about 20 minutes. Here at the bottom is all the task I completed and will be easier to understand.

Time Commitment:

Ordering Parts and Delivery 🡪 2 weeks

Raspberry Pi Setup 🡪 1 hour

PCB Soldering 🡪 2 hours

Testing PCB 🡪 1 hour

Coding 🡪 5 hours

Remote Desktop 🡪 1 hour

Mechanical Assembly 🡪 15 minutes

Acrylic box assembly 🡪 10 minutes

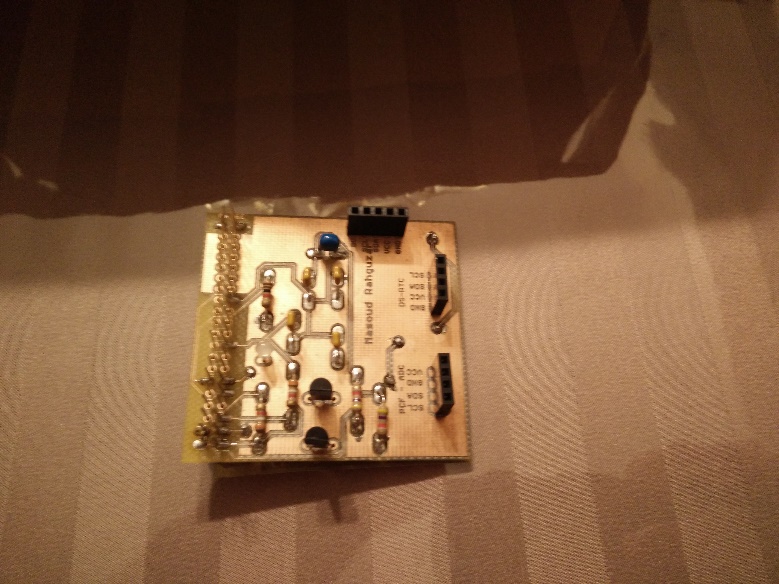
All these up top tasks are completed in order.

**Mechanical Assembly**

The assembly of my project is very simple first connect the PCB on top of the raspberry pi which contains the light indicator. Secondly, proceed and connect the webcam to the USB port on the raspberry pi and then lastly connect the raspberry pi to a power outlet. And that’s how to Assembly all the parts of my project.

**PCB / Soldering**

Humber college provided us with a PCB which is called the Modular Sensor Hat. The PCB contains 20 pin GPIO header, two 4 pin header, 5 pin header, a couple of resistors, transistors and a light. Before soldering I took care of my safety by wearing my safety glasses and making sure my work place was clear. The soldering was fairly simple to complete because of the schematic which outlined where everything goes.



**Power Up**

The power begins with the setup of the raspberry pi. First connect the raspberry pi to a external monitor, keyboard and mouse and then to a power outlet. Once powered on the setup of the OS will commence which will take a few minutes. When complete open up the terminal and start updating the raspberry pi using the command “sudo apt-get update” which will update the raspberry pi to the latest version. Afterwards, PCB must be tested because it is to be used as a light indicator for scanning barcodes. The code for the light on the PCB was given by Humber College and could be run by running this command in the terminal “sudo ./traffic2B” which will make the light change red to green every few seconds. The code is also provided on my GitHub page.

Easy Steps to Follow:

* The raspberry pi must be updated by using "sudo apt-get update" and "sudo apt-get upgrade" command.
* Install python program and the the python/zbar library using "sudo apt-get install python-dev" and "sudo apt-get install python-pip".
* Install pillow using "sudo apt-get install python-pip" and "sudo apt-get install python-httplib2" command.
* Install FSWEBCAM by using the "sudoapt-get install fswebcam" command to test webcam.
* Download the zbar library from the github main page.
* Unzip the folder using the command, zip = ZipFile('file.zip'); zip.extractall()
* Next, insert the zip files into the correct folder.
* Compile the files using the command "gcc –Wall –o executable name c\_file.c – lwiringPi".
* Run the program with the command "python program\_name.py".

**Unit Testing**

The uniting testing begins the PCB as mentioned in the power up. It is recommended to use the code that is given by Humber College or at my GitHub page to test if the light indicator works. The light indictor can work with the command “sudo./traffic2B” if the code is there. The next step is to check if the webcam work. This can be done by installing FSWEBCAM by putting “sudo apt-get install fswebcam” in the command line of the terminal which will install an easy way to check if the camera works. Once installed type “fswebcam image.jpg” in the command line and if it takes a picture with good quality the webcam works. After this, the code created for this project can be used which is in my GitHub page. The program can run by typing “python bar\_code.py” in the command line.

**Production Testing**

The production testing is fairly simple. Run the barcode program bar\_code.py and place a barcode in front of the webcam. In addition, focus the webcam (manually if needed) so that it can quickly scan the barcode. Once scanned the barcode will be place in a txt file and in the terminal it can display the txt file by typing for example “cat nameoftxt.txt”.

**Reproducible**

My project can be reproducible by following the instruction I have laid out.