Smart Electronic Component Organizer

Kaiwen Zhao
Canlin Zhang
Yihao Deng

ECE 445 Design Document
Team 53
Spring 2020
TA: Yichi Zhang

Table of Contents

1. Introduction

1.1 Objective

As EE students, most of us have stored many electronic components such as resistors, capacitors and MOSFETs. Traditionally, we would store these components in storage organizers, a huge cabinet with many transparent plastic drawers. A small organizer may only have as few as 20 drawers. However, a larger one can have up to a hundred organizers. A big problem is that people usually cannot immediately locate the components they want. They have to look into the transparent boxes or at the tags one by one, wasting a lot of time.

We propose a solution to this problem by creating a logger with indicators for people to better store and find components. It would also consist of mechanical designs to push the drawers out from the back. People could use an App on their Android phones to connect to the organizer to find components, log new components, clear drawers and open the drawers by sending commands to the organizer using their App.

1.2 Background

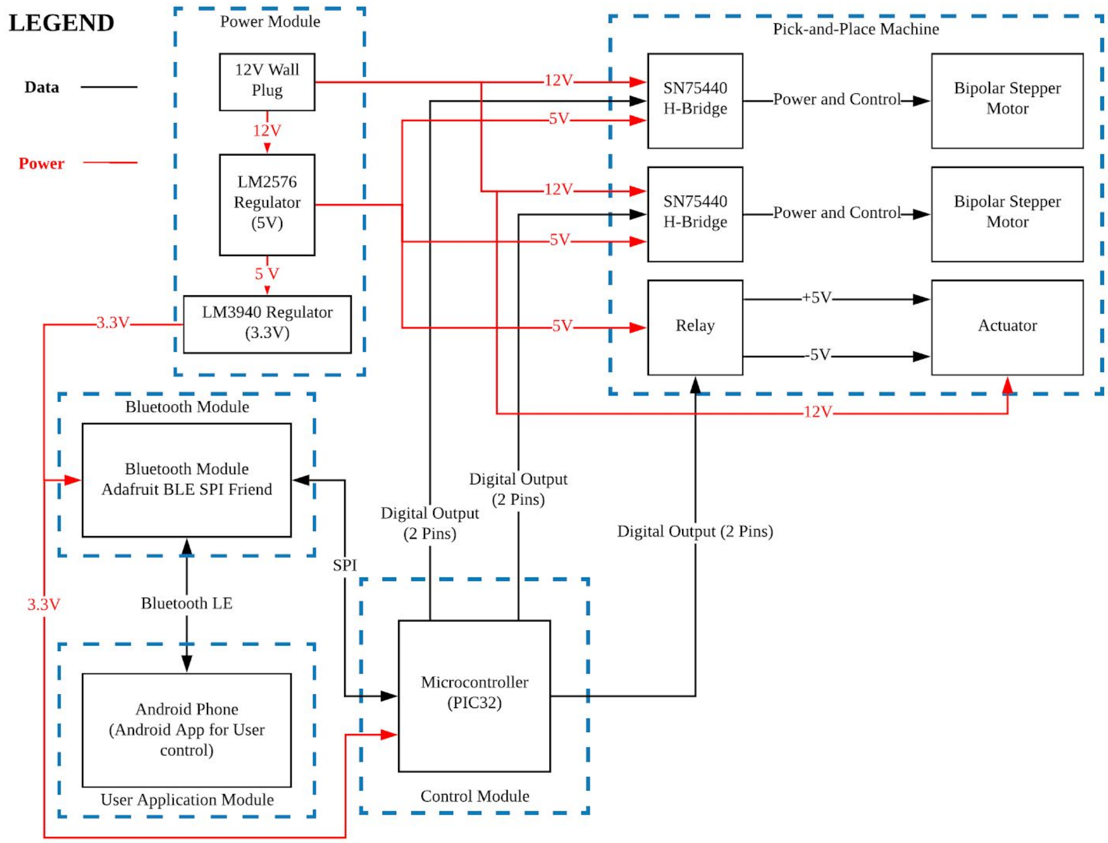
It is not a new idea to digitize boxes or lockers for better management. Though most of the lockers are accessed only by mechanical keys, many electronic lockers have been introduced to the market and deployed by mew department stores and apartments. However, hardly anyone considered bringing a digitized box management system to smaller instances, for example, an electronic components organizer. Although every experienced engineer would be used to find data in datasheets and lookup tables as well as to find a certain type of components out of hundreds of boxes, it would still waste much time. Because there are currently no widely-used and affordable digitized organizers on the market[1], the smart electronic components organizer is proposed to fill the white space.

1.3 High-level Requirement List

- The system should be able to keep track of different components. (names, values, quantities, and locations)
- The system should be about to register for new components as well as deleting unwanted components.
- After users have selected their electronic components using the application on their phones, the corresponding drawers should be opened.

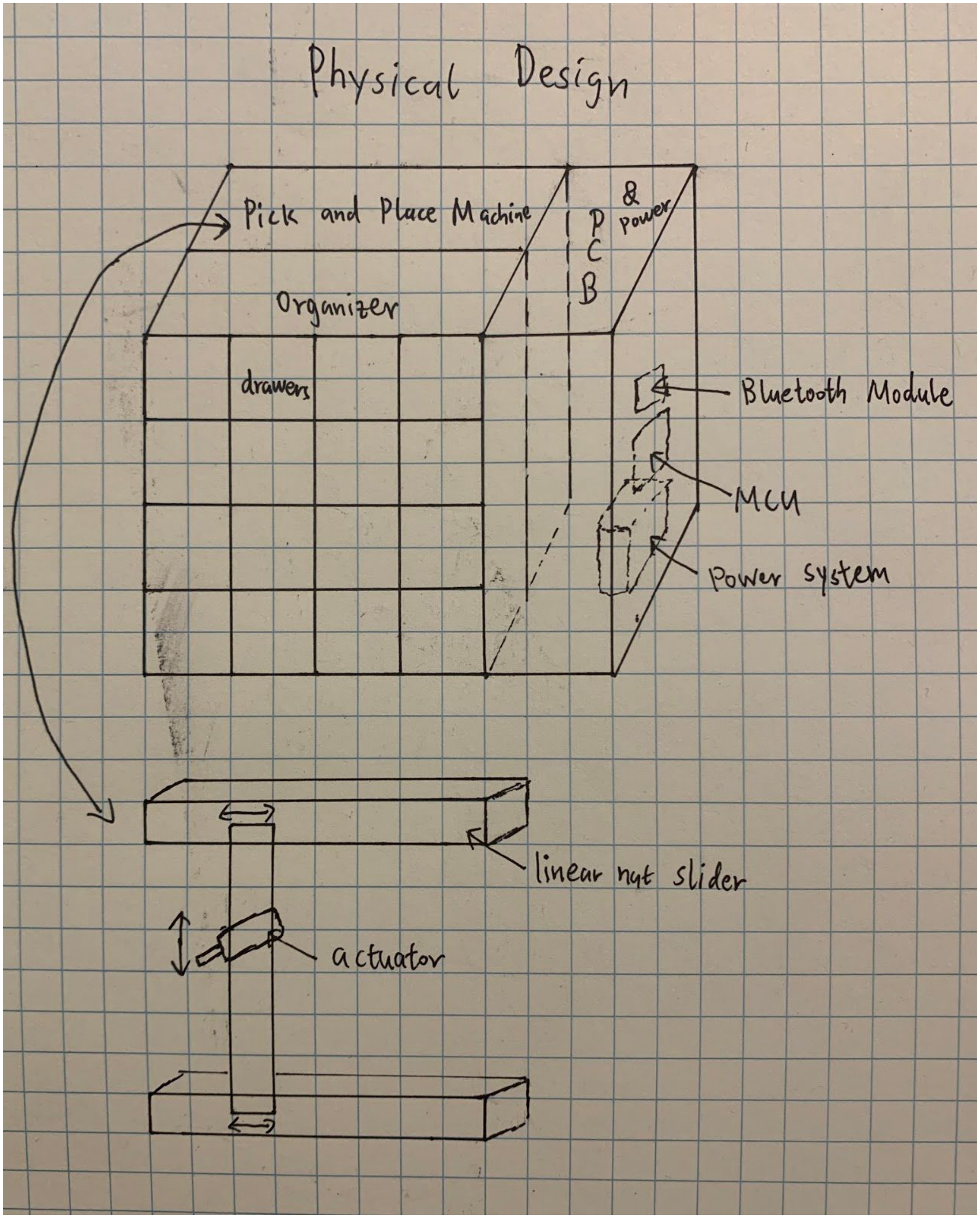
2. Design

2.1 Block Diagram



1. The user would enter the commands and information into their Android App, and the user inputs would be transmitted to the control module for further processing.
2. The user information is stored, processed and displayed in and on the user's Android phone.
3. All physical operations would be performed by the Pick-and-Place machine, the control signals will be sent to the machine by the control module.
4. All modules that require power will be powered by the power system.

2.2 Physical Design

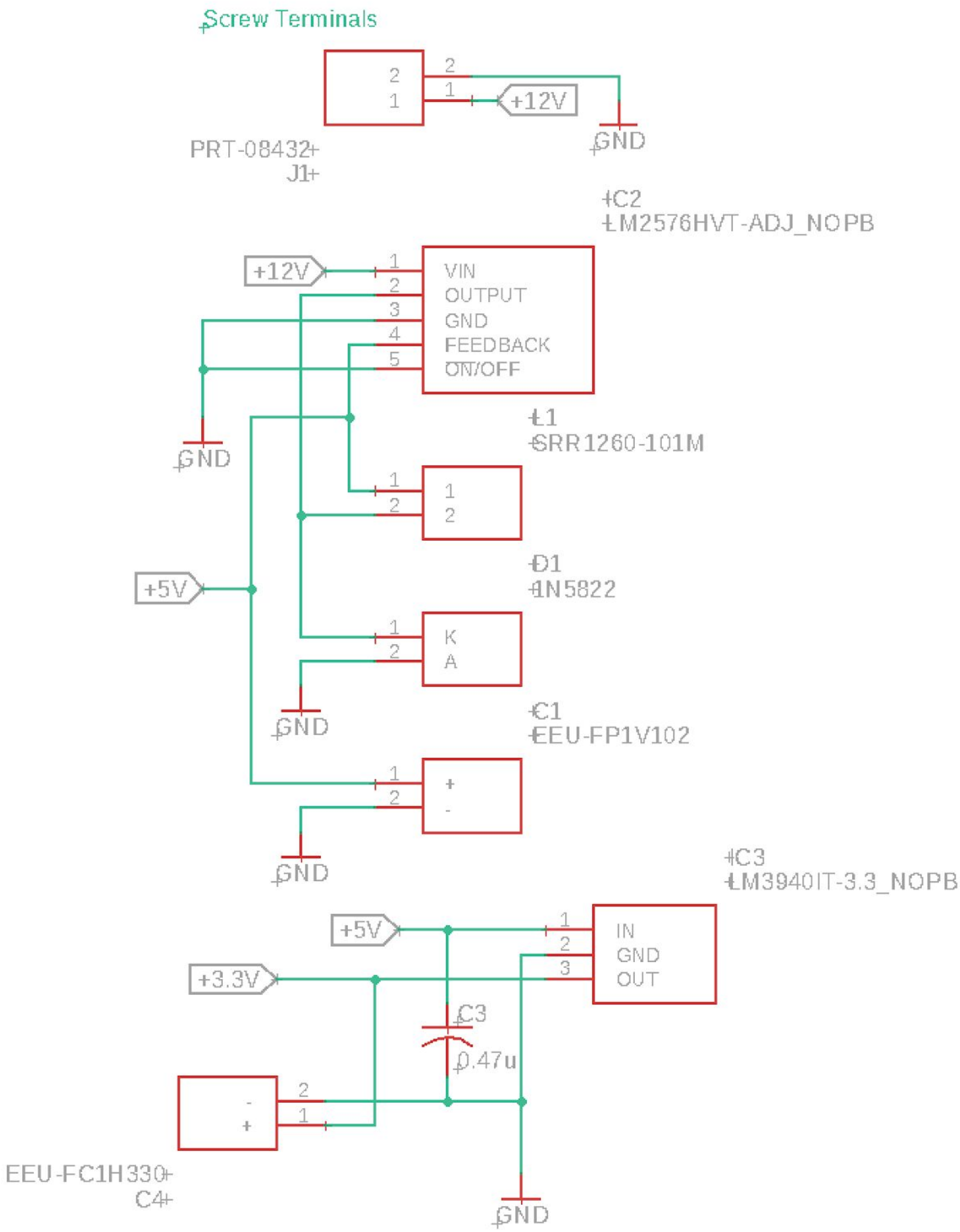


This picture above shows the arrangement of different submodules in our project. The mechanical module is placed behind the organizer, and the control, bluetooth, and power submodules are placed adjacent to our organizer to keep the overall design compacted and safe.

2.3 Functional Overview & Block Requirement

2.3.1 Power System

Schematic of the Power System



The power system is used to supply power for Control Module (MCU), Bluetooth LE module and Pick-and-Place Machine. The power system supplies 12 V to the stepper motors and actuators for the Pick-and-Place Machine, 5V for H-Bridge for step motors and 3.3V for MCU and Bluetooth module.
The power system consists of three parts: A 12 V DC power supply (Wall plug) to supply 12 V power, a LM2576 voltage regulator board to step down 12 V DC to 5V DC, a LM3940 linear regulator board to step down 5 V DC to 3.3V DC.

The power required by the modules are listed below:

Components	Quantity	Voltage	Power (W)	Powered By
PIC32 MCU [5]	1	3.3	1.5	Regulator
Adafruit BLE SPI module [6]	1	3.3	26.1m	Regulator
Actuator	1	12	24	Wall Plug
Stepper Motor [7]	2	12	24 * 2 = 48	Wall Plug

We estimate the total power consumption of our design will not exceed 100 watts.

Requirements	**Verification**
1. The wall plug must be able to deliver a stable 12 V DC voltage. 2. The regulator board must be able to step down the 12 V DC voltage and deliver a stable 5 V DC voltage. 3. The linear regulator board must be able to step down the 5 V DC voltage and deliver a stable 3.3 V DC voltage.	1. Use a digital multimeter to measure the output of wall plug and regulator board to check whether the correct voltage has been generated. 2. Connect a resistive load on both the wall plug and regulator board and measure the output current and voltage to check whether the power system could deliver stable and correct voltages under relatively heavy load. 3. Connect a resistive load on both the wall plug and regulator board to let the power system run for a prolonged period of time (e.g. 1 hour to 2 hours) to check whether the power system is robust under load.

2.3.2 Pick and Place Machine

Diagram for Motor Driver circuit [8]

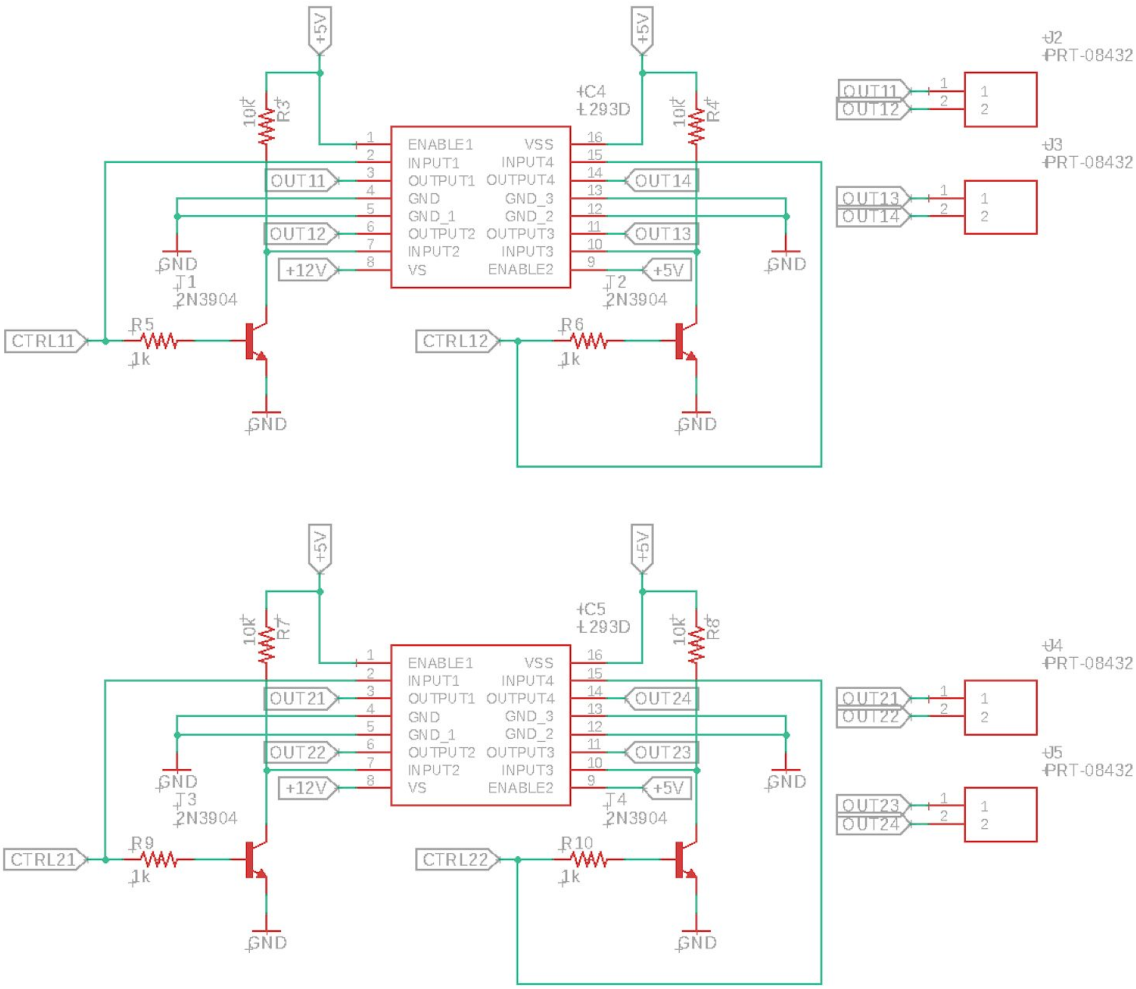
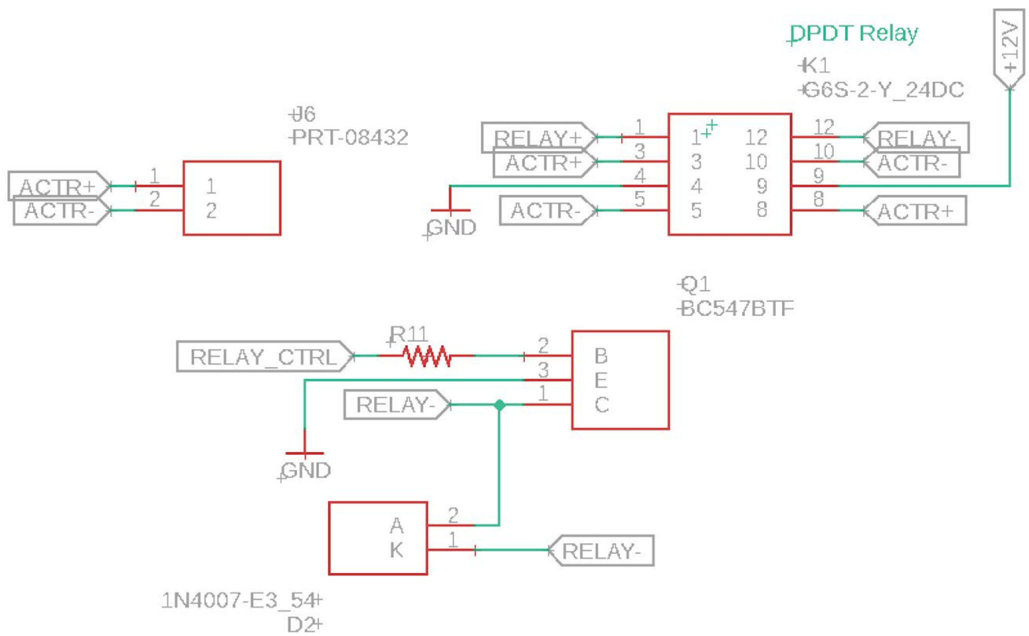


Diagram for Actuator Driver Circuit [9][10]



The pick and place machine consists of two screw sliders, an actuator, and two stepper motors. One stepper motor is used for the linear motion of the actuator in x direction and the other one is for y direction. The actuator, with a stroke length of 4 inches, is used to push open the drawer from its back. Each drawer is 2-⅛ inch wide and 1-½ inch tall.

Requirements	Verifications
1. The actuator is able to move to the center of the back of each drawer. 2. The actuator is able to extend to push out the drawer and retract back to the original location.	1. Use a ruler to make a measurement of the distance between the actuator and the edge of the drawer after the actuator stops its movement. 2. Use a ruler to measure the stroke length after the actuator fully extends.

2.3.3 Bluetooth Module

The Bluetooth Module consists of an Adafruit Bluetooth LE SPI board. The Bluetooth Module connects to the control module through SPI interface and connects to the user's Android phone through Bluetooth 4.0 protocol.

The Bluetooth Module receives the user command sent from the Android phone using an Android application, then it sends the command data to the control module through SPI interface. It could also receive data packets from the control module, then send them to the user's mobile phone to be read and processed by the Android application.

Requirement	Verification
1. The Bluetooth module must be able to connect to the control module through SPI. a. The control module could send data packets to the Bluetooth module b. The control module could receive data packets correctly to the Bluetooth module 2. The Bluetooth module must be able to connect to the user's Android phone through Bluetooth 4.0 LE. a. The bluetooth module could send data packets previously received from the control module to the user's phone. b. The bluetooth module could send	1. Verify that bluetooth module is correctly setup, initialized and connected to the control module by checking the contents in the control registers of the bluetooth module. 2. Verify that bluetooth module is able to receive or send data packets from or to the control module by sending test data packets from control module and route back to the control module again. 3. Verify that the bluetooth module is able to send data packets to the Android phone by sending test data packets from the control module and check whether the phone can receive the test data packets. 4. Verify that the bluetooth module is able to receive the data packets from the Android phone by sending test data packets to the bluetooth module and check whether the control module can receive the test data packets.

2.3.4 Control Module

Diagram for Control Module Circuit (MCU)

The control module consists of a PIC32MX270 MCU. The control module connects to the mechanical components (Pick-and-place machine) and the bluetooth module.
The control module receives the data packets from the Android phone by reading from the Bluetooth module. Then, it processes these data into control signals to the mechanical components. It could also report its internal status, such as the status of current task, values of status registers, etc. to the android phone by sending data packets to the Bluetooth module.

Requirement	Verification
1. The control module must be able to correctly control the mechanical components. 2. The control module must be able to receive commands from an Android phone and process them properly. 3. The control module must be able to report its own status, such as task status.	1. Write unit test codes to test the control module's ability to control mechanical components, including stepper motor and actuator. 2. Send test data packets to the control module and see whether it receives the data and handles them properly. 3. Write unit test codes to test the self-report functions of the control module.

2.3.5 Android Application

The Android Application is a software inside the user's Android phone. The application takes the user inputs and either sends commands to the control module or updates its internal data respectively.

The application also displays the current status of the storage drawers and mechanical components, including the contents in each drawer, the location of empty drawers and current location of the actuator in the pick-and-place machine.

The application communicates with the control module through a bluetooth module using Bluetooth 4.0 protocol, it could either send the user commands to the control module or receive the status feedback from the control module for further processing.

Requirement	Verification
1. The application must be able to correctly store user data into their phone and retrieve the data upon application start. User data includes: a. The contents of each drawer. b. The locations of empty drawers. c. User's previous search history for components. d. Username e. The bluetooth device ID and name of the bluetooth module 2. The application must be able to accept user commands, including: a. Finding a specific component. If the component exists, the corresponding drawer will be shown to the user. b. Opening a drawer either by location, component name or previous search result. c. Store one or more components into one or several drawers. If there are no empty drawers, the application would notify the user and cancel. If the component already exists in one of the drawers, the application will notify the user of the location of such drawer.	1. Test the user data storage capability by inputting random user commands and restarting the application. 2. Test the communication between the phone and the Bluetooth and control modules by sending random commands to the control module or requesting status from the control module. 3. Test the correct execution of user commands by observing the behavior after executing each user command and match them with the expected behavior. 4. Test the robustness of the application by running for a prolonged period of time and observe whether the application runs properly or has crashed. (e.g. 4 hours - 8 hours)

d. Remove one or more components from one of several drawers. 3. The application must be able to communicate with the Bluetooth module and control module properly. The possible data in the communication are: a. Commands that should be interpreted by the control module to control mechanical components. b. Status messages generated by the control module to be read and processed by the application.	

2.4 Tolerance Analysis

One important component we have to examine is the stepper motor, which is being used to transport the actuator. Let say there is a larger gear compared to our stepper motor, and assume the gear ratio of our stepper motor to this gear is 1 to X. If we want our actuator to move faster than just using the stepper motor, X must be greater than 1. With the consideration of 5% of error that the stepper motor has, every rotation of the stepper motor takes, the gear is able rotate 1.72X to 1.89X more rotations (using the diameter of the stepper motor). The coordinates of centers of the drawers are integers multiply with 1-1/16" for x direction and integers multiply with ¾" for y direction. As long as the actuator is moving certain integer steps of a chosen gear or the stepper motor that is equal to these coordinates, the location of the actuator would match the center of each drawer.

3. Cost and Schedule

3.1 Cost Analysis

The average hourly wage for an Electrical Engineer in the United States is $34/hr [4]. There are 16 weeks in a semester. We have 3 people working on the development of this project, and each of us works about 15 hours per week.
Estimate hours = 3 x 16 x 15 hours = 720 hours
Estimate cost of labor = $34/hour x 720 hours x 2.5 = $61,200

Component	Cost
PIC32MX270F256B MCU	$4.32
Akro-Mils 10116 16 Drawer Plastic Parts Storage Hardware and Craft Cabinet, 10.5-Inch x 8.5-Inch x 6.5-Inch, Black	$13.66
Low Noise 100mm Electric Micro Linear Actuator 12VDC 4.4 lbs Force IP65	$60.55
Adafruit Bluefruit LE SPI Friend - Bluetooth Low Energy (BLE)	$17.50
STEPPERONLINE Stepper Motor Nema 17 Bipolar 40mm 64oz.in(45Ncm) 2A 4 Lead 3D Printer Hobby CNC	$12.99 (x2)
Screw Terminals 5mm Pitch (2-Pin)	$0.95(x6)
LM2576HVT-15/NOPB	$6.09
LM3940IT-3.3/NOPB	$1.82
SRR1260-101M	$1.25
1N5822	$0.41
EEU-FP1V102	$1.61
L293D	$3.91(x2)
G6S-2F-DC24	$4.12
1N4007-T	$0.19
BC547BTA	$0.21

SMD Ceramic Capacitors and SMD Resistors	$20.00

Estimate cost of parts = $171.23

Total cost = $61,200 + $171.23= $61,371.23

3.2 Schedule

Week	Kaiwen	Canlin	Yihao
3/2	Test power circuit submodule and relay polarity inverter circuit with actuator on breadboard.	Write codes for BLE module initialization, MCU - BLE module communication. Learn basics of Android Development.	Communicate with Machine shop about the mechanical design, make adjustments if necessary
3/9	Design first version of PCB including power, MCU and motor controlling submodules	Write the skeleton of Android App for Phone - BLE module communication. Write code in MCU to control the BLE module to send/receive data packets from phone.	Write commands to test actuator and motors to find correct parameters for signal input and to make sure there is enough input power
3/16	Finish and order first version of PCB from Oshpark	Continue developing the Android App to allow user inputs. Write code to enable MCU to receive and process data packets sent from the phone.	Complete the mechanical submodule and start testing it with previously written commands associated with potential user input

3/23	Communicate with other group member over software and mechanical module development	Starting to test the integration of the control module and mechanical components. Test basic functions to command the actuator and stepper motor.	Assist teammate on the integration of mechanical submodule and control module and test function on software
3/30	Solder components on PCB and test	Continue to develop the App. Starting to test integration of phone - BLE - control module - mechanical component.	Help teammate mount components onto PCB and start testing
4/6	Integrate the PCB with bluetooth module and mechanical submodule	Continue developing the App to allow the user to send commands to control the mechanical components.	Assist teammates with final integration of all submodules
4/13	Debug and optimize design	Debugging	Debugging with teammates
4/20	Mock Demo & debugging	Mock Demo & Debugging	Mock Demo & debugging
4/27	Demonstration & Mock Presentation	Demonstration & Mock Presentation	Demonstration & Mock Presentation
5/4	Presentation	Presentation	Presentation

4. Ethics and Safety

One safety issue might occur in our project is handling the wall supply that supplies for all of the other modules. We need to carefully determine the power consumption of each module, dedicate to safely distribute power in our power management design, and decide the working conditions of our organizer as a reference to our users.

Another potential safety issue is there is a possibility that the actuator pushes one of the drawers too far so that the whole organizer might fall on the user. This could be prevented by setting a maximum current or voltage applied on the mechanical structure, conducting a series of testing about the force exerted by the actuator when opening the drawer, and making adjustments to the position of the actuator correspondingly.

Coming up with solutions for both of the statements above are to uphold the IEEE Code of Ethics #1, by disclosing any safety issues and also trying to prevent them during the development stage[3]. As a team of three people, we aim to collaborate with each other, support each other's idea, and make the best use of each one's strength as well as learn from each other. This would reflect the IEEE Code of Ethics #10[3].

Along the way to the completion of this project and to the future, staying active and positive to respond to any supportive criticism is essential to the development of our product. Making immediate modifications based on feedback and correcting mistakes without any delay is necessary for any project development. This is suggested by IEEE Code of Ethics #7[3].

5. Citations

[1] "Chipseasy-Smart Electronic Component Organizer," *Kickstarter*. [Online]. Available: https://www.kickstarter.com/projects/3dfacture/chipseasy-smart-electronic-component-organizer?ref=BetaPage . [Accessed: 25-Feb-2020].

[2] "DIY Arduino Robot Arm with Smartphone Control," *HowToMechatronics*, 15-Dec-2019. [Online]. Available: https://howtomechatronics.com/tutorials/arduino/diy-arduino-robot-arm-with-smartphone-control/ . [Accessed: 25-Feb-2020].

[3] Ieee.org, "IEEE IEEE Code of Ethics", 2016. [Online]. Available: http://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 12- Feb- 2020].

[4] Salary.com, "Hourly wage for Electrical Engineer I," *Salary.com*. [Online]. Available: https://www.salary.com/research/salary/benchmark/electrical-engineer-i-hourly-wages. [Accessed: 26-Feb-2020].

[5] PIC32MX1XX/2XX 28/36/44-PIN, 32-bit Microcontrollers (up to 256 KB Flash and 64 KB SRAM) with Audio and Graphics Interfaces, USB, and Advanced Analog. Microchip Technology, Inc., 2016. [Online]. Available: https://www.mouser.com/datasheet/2/268/60001168J-967542.pdf. [Accessed 24-Feb-2020]

[6] MDBT-40 Series, RAYTAC, Inc. 2015 [Online]. Available: https://cdn-learn.adafruit.com/downloads/pdf/introducing-the-adafruit-bluefruit-le-uart-friend.pdf [Accessed: 26-Feb-2020]

[7] 17HS49-2004D Stepper Motor, Stepperonline. 2018 [Online]. Available: https://www.omc-stepperonline.com/download/17HS19-2004D.pdf [Accessed: 26-Feb-2020]

[8] Circuit for Bipolar Stepper Motor, Arduino Community. 2020 [Online]. Available: https://www.arduino.cc/en/Reference/StepperBipolarCircuit [Accessed: 26-Feb-2020]

[9] How To Use Relays To Control Linear Actuators, Actuonix Motion Device. 2017 [Online]. Available http://blog.actuonix.com/2017/12/how-to-use-relays-to-control-linear.html [Accessed: 26-Feb-2020]

[10] Arduino Relay Control Tutorial, Saddam. 2017. [Online]. Available https://circuitdigest.com/microcontroller-projects/arduino-relay-control [Accessed: 26-Feb-2020]