Smart Electronic Component Organizer

Kaiwen Zhao
Yihao Deng
Canlin Zhang

Group 53
ECE 445 Project Proposal
Spring 2020
Mentor TA: Yichi Zhang

Contents

Chapter 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 around 16 to 25.

However, a larger one can easily house hundreds of drawers, causing a significant problem: People usually cannot immediately locate specific components. They would have to traverse through many drawers to find the one with components they need.

Therefore, we propose a solution to this problem by creating a logger with indicators for people to better store and find components. Moreover, it would consist of mechanical designs to push the drawers out from the back.

People would use the logger to either assign a certain drawer to a component, or command the automated stick to push out the drawer with component they need. The logger would have a simple LCD screen and buttons (with labels of 0 to 9 and r, c, l, ic, value, number, enter, eject and clean).

Users would be able to log new components and find logged components using the screen and the button. To find a certain component, users would use the buttons to specify the component they want. For example, if user types r, 0603, value and 200 and presses enter, the indicator (a LED) of the drawer which user registered before for this component would be lit. If the user presses the eject button, the specified drawer would be pushed out from the back.

1.2 Background

It is not a new idea to digitize boxes or lockers for better management.

Even 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 for storing smaller objects, for example, an electronic components organizer.

Although every experienced engineer would be used to find data in the 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 is 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 Requirements

- Users are able to register new components into the system.
- After users input relevant information of the electronic components they request, they should be able to quickly locate these components with the help of a prompt message on LCD screen, a LED in the corresponding drawer, and the opened drawer (most obvious).
- The organizer should be able to open and close the corresponding drawers for users by a mechanical structure resembling a robot arm (with three straight sections, two rotating nodes and one small clipper at the tip) behind the organizer.

Chapter 2

Design

2.1 Block Diagram

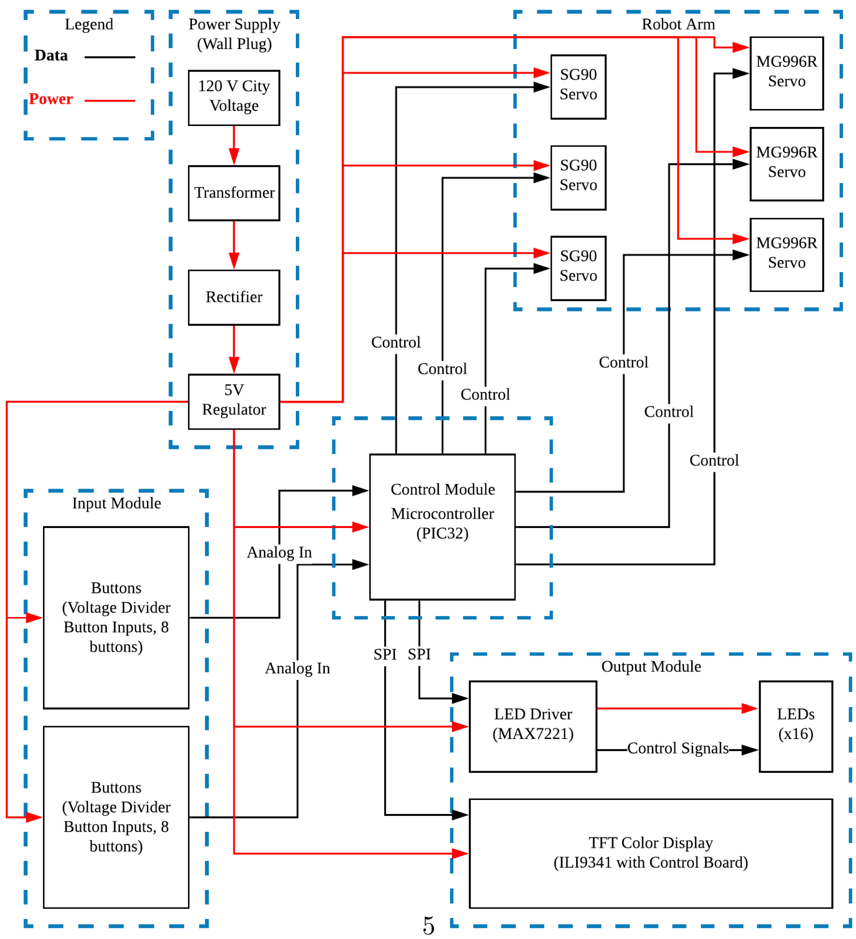

Figure 2.1: Block Diagram

- The user would enter the commands and information to the input module, and the user inputs would be transmitted to the control module for further processing.
- The user information would be displayed or indicated by the output module. The user information will be sent to the output module by the control module.
- All physical operations would be performed by the mechanical system, the control signals are sent from the control module to the mechanical system.
- All systems that need power to operate will be powered by the power system.

2.2 Physical Design Sketch

The design of the robot arm in following figure would be based on an online 3D robot arm design [2], while reducing it down to a simpler version to serve our purpose.

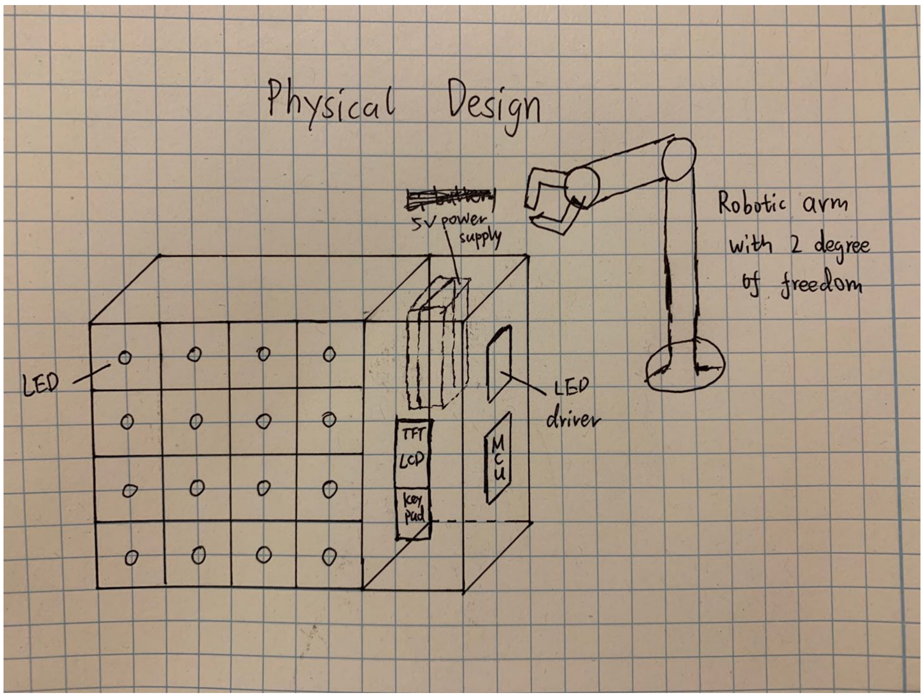

Figure 2.2: Physical Design Sketch

2.3 Functional Overview and Block Requirement

2.3.1 Power System

The power system is used to supply power for control module (MCU), output module (LEDs and the screen) and mechanical subsystem (robot arm).
The power required by the modules are listed below:

Components	Quantity	Voltage (Volts)	Power (Total, W)	Powered By
PIC32 MCU [3]	1	5	1.5	5V Power Supply
MAX7221[4] LEDs	1 (MAX7221) 16 (LEDs)	5 (MAX7221) 2 (LEDs)	2.3	5V Power Supply (Chip) MAX 7221 (LEDs)
ILI9341[5] [6]	1	5	0.22	5V Power Supply
SG90 [7]	3	5	7.5	5V Power Supply
MG996R [8]	3	5	15	5V Power Supply

Requirements

- The power system must be able to supply a stable 5V output to the control module, output modules and robot arm without any voltage spike or other fluctuations when the system is powered on.

Verification

- Use a digital multimeter to check whether the supplied voltage into the components is correct. (5V DC)
- Use a resistive load to make the power supply outputs 40 Watts for a prolonged period of time (1 Hour to 2 Hours) to verify that output voltage or current does not fluctuate.

2.3.2 Input Module

The input module consists of 16 buttons for user input. The user should be able to enter instructions into the control module using the buttons.
The buttons are connected to outputs of voltage dividers to the analog pin of the control module. [9]

Requirement

- The buttons must be able to generate their pre-mapped analog voltages correctly and stably upon button press. (Pre-determined by the voltage divider design)

Verification

- Use a digital multimeter to measure the output voltage level and stability upon each button press to check whether the buttons are generating the correct voltages.
- Verify the correctness of button - control module connection by:
  - Set up the ADC channels AN1 and AN2 in MCU.
  - Press the buttons and be able to observe different ADC results printed.

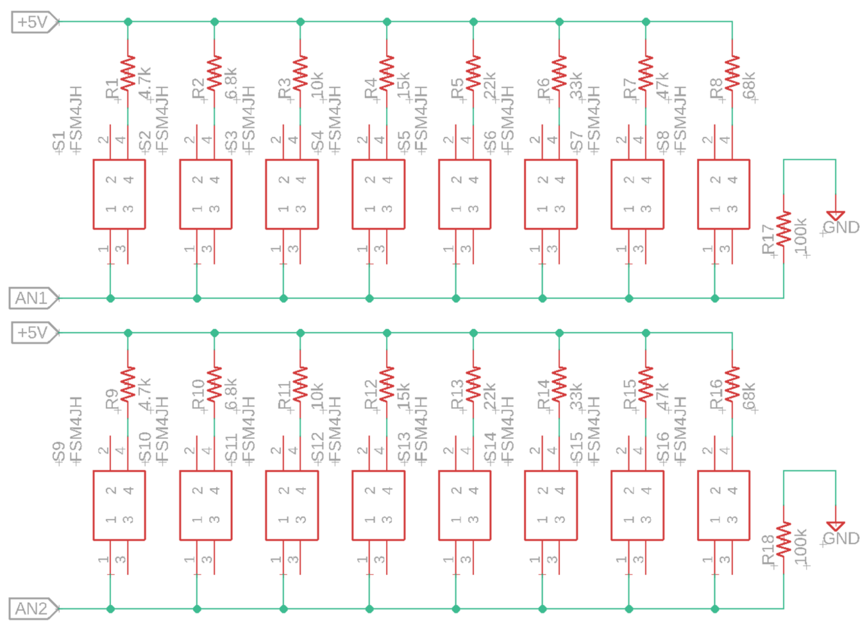

Figure 2.3: Schematic for Buttons

2.3.3 Output Module

The output module consists of 16 LEDs, one for each drawer and a small TFT display. The information, including current user operation, overall drawer availability and drawer content information, should be displayed. The LEDs are driven by a MAX7221 LED Driver chip, which is connected to the control module through SPI interface. The screen [5] is controlled by the control module through SPI interface.

Requirements

- The LEDs must be able to indicate whether a drawer is empty or occupied.
- The Display must be able to display the following information:
  - The command the user is currently inputting.
  - The content of the drawer selected.
  - Results for searching specific contents.

Verification

- Test the functionality of LEDs by testing control module's ability to successfully drive any combination of 16 LEDs specified by data sent by control module over SPI protocol. More specifically:
  - Set up SPI communication between MCU and MAX7221.
  - Send 0xX900 to set Decode Mode at No Decode.
  - Send 0xX101, and 0xX280 and ensure that LED1 (connected between SEGG and DIG0 pins) and LED16 (connected between SEGDP and DIG1 pins) are lit.
- Test the functionality of screen by testing control module's ability to successfully display basic texts and shapes on the display.
- Test the functionality of display function of control module to successfully display correct information on the display. This should be tested before mechanical assembly, and the screen should initially be working only with power and control modules.

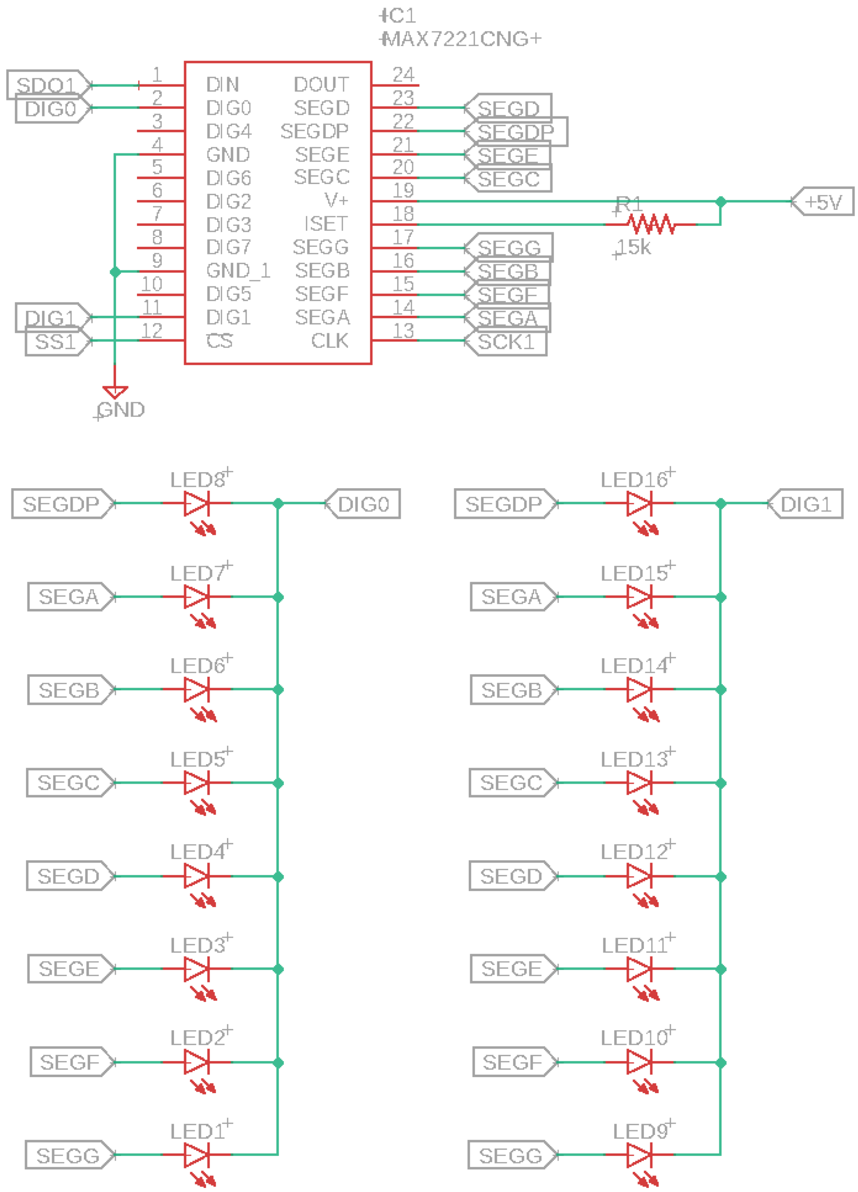

Figure 2.4: Schematic for LED Driver and LEDs

2.3.4 Mechanical System (Robot Arm)

The mechanical module consists of a robot arm which physically interacts with the drawers, including pulling (to close the drawers) and pushing (to open the drawers). The robot arm could open or close one drawer at a time, and the control system will command the robot arm to reach and either open or close the drawer. The robot arm is based on a current design.[2] However, less servos will be used to simplify the complexity of mechanical design and reduce the number of electric components.

Requirements

- Robot arm must be able to be controlled by the control module.
- Robot arm must be able to reach all of the drawers with good precision. (Max. Error around one centimeter)
- Robot arm must be able to push and pull all of the drawers with good precision. (Max. Error around one centimeter)

Verification

- Test the correctness of mechanical and electrical assembly of robot arm by manually stimulating each of the servo and see whether the robot arm rotates, extends or tilts correctly.
- Test the ability of controlling the exact movement of robot arm through control module. Connect the robot arm with the control module, and use a simple program to control the robot arm to move to a pre-determined location.
- Test the ability of moving to a specific drawer precisely by controlling the robot arm to move to a specified drawer, then use a vernier scale to measure the error.

Figure 2.5: Mechanical Overall Model of Robot Arm

2.3.5 Control Module

The control module uses a PIC32 MCU to control the output module and robot arm and process user inputs from input module.

Requirements

- The control module must be able to recognize and process this list of user inputs.
  - Open/Close one of the drawers.
  - Assign contents (content name) to one of the empty drawers.
  - Find any non-empty drawers by specific content name.
  - Clear contents (make drawer empty) of one or more drawers.
- The control module must be able to display the list of user information.
  - Availability of each drawer. (Indicated by each LED on every drawer)
  - Search results when the user tries to search for specific contents.
  - Which drawer is currently under mechanical control (Opening or closing)
- The control module must be able to store user information into its storage device. (FLASH memory) The list of user information is:
  - Contents of each drawer.
  - The list of drawers that are currently empty.
- The control system must be able to control the mechanical system to interact with the drawers physically. In this case, this would be that the robot arm could reach the drawers accurately.

Verification

- Verify the basic functionalities of the control module (MCU):
  - Verify the ability of connecting the control board with the USB for programming. Connect the control module to a JTAG debugger to a computer.
  - Verify the correctness of output of each output pin of the control board. Use a simple program to assert output of each output pin, then use a digital analyzer or a digital multimeter to measure the correctness of the output signal.
  - Verify the ability of utilizing onboard FLASH memory. Use a simple program to store certain data into the flash memory, power off, then restart to retrieve the data from the flash memory.

- Verify the correctness of input handling by connecting the input module to the control module, then use a random stream of user input to test control module's ability to process user inputs.
- Verify the correctness of output to LEDs and the screen upon user inputs by connecting both input and output modules to the control module, then use a random stream of user input to test control module's ability to generate correct outputs to LEDs and the screen.
- Verify the ability of controlling the robot arm correctly by connecting the robot arm to the control module and use a simple program to control the movement of robot arm.

2.3.6 Tolerance Analysis

The calculations mainly focuses on two parts, the powered used by the entire system and the precision of robot arm.
For the power used by the system, we first calculated the estimated power usage of each components as indicated in section 2.3.1, then we multiply the power usage by 1.5 times, giving us a larger margin of error for power consumption estimation. In the absolute worst case, we calculated that the power consumption would be around 40 Watts.
Another problem is the precision of robot arm. As servos will generate small errors during their movement, either by mechanical manufacturing error or the small errors in timing, using several servos in robot arm would cause these errors to accumulate, resulting in location deviation of robot arm, making it unable to reach the correct drawers. The method here would to be use an encoder to read the current position of servo, then use a PID control loop to gradually reduce the error to an acceptable margin.
Also, the dimension of the organizer drawers and robot arm dimensions are important. Measurement must be made to ensure that the robot arm would reach all of the 16 drawers with a good extra margin of error.

Chapter 3

Ethics and Safety

One safety issue might occur in our project is handling the pack of lithium batteries 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 robot arm pushes one of the drawers with a large momentum 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 robot arm when opening or closing a drawer, and making adjustments to the position of robot arm 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 [10]. 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].

Bibliography

[1] 3DFacture, *Chipseasy-Smart Electronic Component Organizer*. 2017 (Accessed Feb 24, 2020). Retrieved from https://www.kickstarter.com/projects/3dfacture/chipseasy-smart-electronic-component-organizer?ref=BetaPage.

[2] Dejan, *DIY Arduino Robot Arm with Smartphone Control*. 2018 (Accessed Feb 24, 2020). Retrieved from https://howtomechatronics.com/tutorials/arduino/diy-arduino-robot-arm-with-smartphone-control/.

[3] *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 (Accessed Feb 24, 2020). Retrieved from https://www.mouser.com/datasheet/2/268/60001168J-967542.pdf.

[4] *MAX7219/MAX7221, Serially Interfaced, 8-Digit LED Display Drivers*. Maxim Integrated, Inc., 2003 (Accessed Feb 24, 2020). Retrieved from https://datasheets.maximintegrated.com/en/ds/MAX7219-MAX7221.pdf.

[5] *LCD MODULE SPECIFICATION, Model: MI0283QT-11*. Multi-Inno Technology Co., Ltd., 2012 (Accessed Feb 24, 2020). Retrieved from https://cdn-shop.adafruit.com/datasheets/MI0283QT-11+V1.1.PDF.

[6] *ILI9341, a-Si TFT LCD Single Chip Driver 240RGBx320 Resolution and 262K color*. ILI Technology Corp., 2010 (Accessed Feb 24, 2020). Retrieved from https://cdn-shop.adafruit.com/datasheets/ILI9341.pdf.

[7] *SERVO MOTOR SG90 DATASHEET*. Tower Pro Pte. Ltd., 2014 (Accessed Feb 24, 2020). Retrieved from https://components101.com/sites/default/files/component_datasheet/SG90

[8] *MG996R Servo*. Tower Pro Pte. Ltd., 2014 (Accessed Feb 24, 2020). Retrieved from https://www.towerpro.com.tw/product/mg996r/.

[9] codebender_cc, *How To: Multiple Buttons on 1 Analog Pin - Arduino Tutorial*. 2016 (Accessed Feb 24, 2020). Retrieved from

https://www.instructables.com/id/How-to-Multiple-Buttons-on-1-Analog-Pin-Arduino-Tu/.

[10] IEEE, *IEEE Code of Ethics*. 2016 (Accessed Feb 12, 2020). Retrieved from http://www.ieee.org/about/corporate/governance/p7-8.html.