Initial Prototyping

ECE 388

Team 4:
Jacob Aubertine
Mathieu Bolduc-Clayton
Sal Fernandes

20 October 2018

Contents

1 Abstract

This report will cover the various methods materials and test plans used for this secure elevator control system. The description of each part of the project and the respective test plans for each part are shown.

2 Introduction

The secure elevator control system has the goal of of safely transporting authorized personnel from one floor to another floor. To accomplish the device is broken into several pieces. There are LED indicators, a buzzer, servo, and DC motor to serve as output to either show the user the status of the elevator or to move the elevator itself. There are IR sensors, a button matrix, and a magnetic card reader for user input as a way to direct how the elevator behaves. A micro controller ties all the parts together by driving output based on user input.

3 Methods

3.1 Equipment and materials

3.2 Problem statement

The goal of this project is to design and construct a secure private elevator control system. All aspects are to be designed to integrate together as an embedded system.

The concept block diagram (Figure 1), which encompasses both hardware and software, provides a general overview of the elevator's systems. The hardware block diagram (Figure 2) provides an overview of just the hardware aspects of the project and they interact with each other. The software block diagram (Figure 3) shows a flowchart of the elevator operations, providing the logic for the system.

3.3 Subsystem descriptions

3.3.1 Height/position

The elevator will be moved by a DC motor connected to a cable that winds and unwinds around a pulley above the shaft in order to bring the elevator up and down. The IR distance sensor measures the distance (using a range of voltages to specify the distance the object is) between the top of the shaft (near the pulley) and the elevator. This allows for knowledge of the elevators position at all times, ensuring it reaches the proper floor. Some sort of track would be ideal to allow the elevator to move smoothly without any unwanted rotation. Unwanted movement would likely disturb the IR sensor's measurements. A buzzer will be used to signify that a floor is reached and the doors are opening.

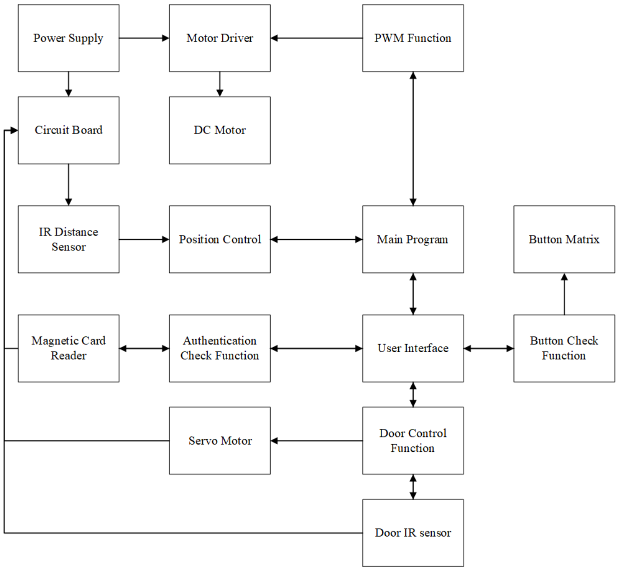


Figure 1: Concept block diagram

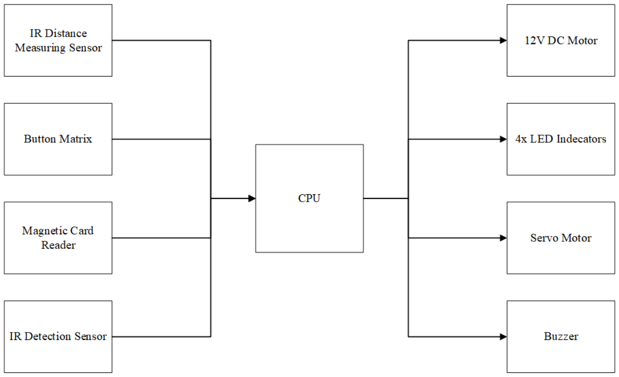


Figure 2: Hardware block diagram

3.3.2 Security

A magnetic stripe card reader will be used as the main form of security. It can allow access to individuals if they swipe a valid card. Once a valid card is swiped, they can call the elevator. The elevators doors should stay closed when not in use, so a swipe is required to call the elevator and have its doors open. LED indicators exist on the card reader itself which provide feedback to the user if the swiped card is valid or invalid. The current plan is to use UMass Dartmouth IDs, with a certain number specified as valid.

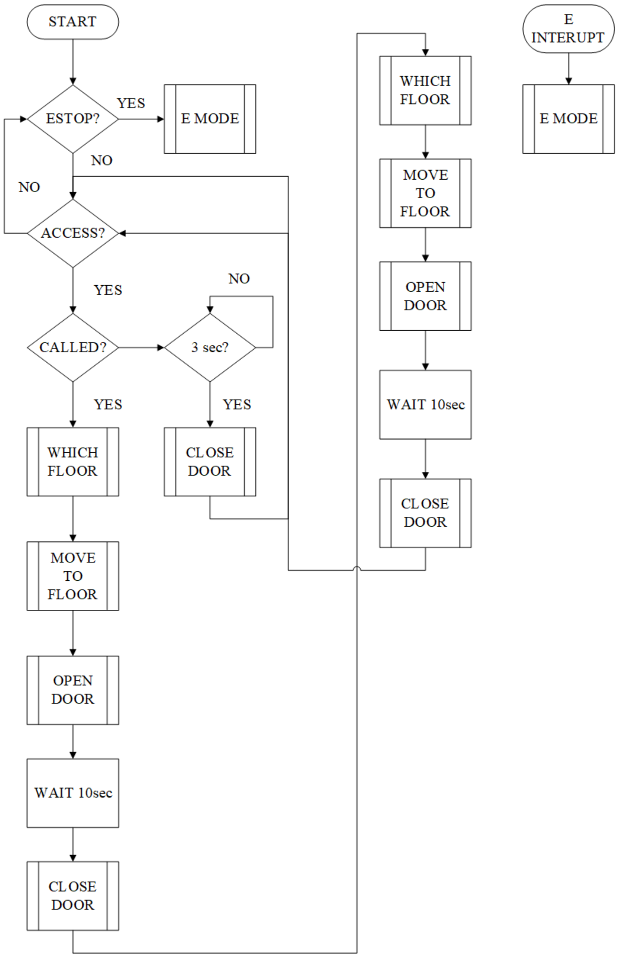


Figure 3: Software block diagram

3.3.3 Safety

An emergency stop button will exist that halts all function and stops the elevator sounding the buzzer until the button is pressed again. There will be an IR breakbeam sensor to detect if there is anything in the way of the open door. If there is something blocking the way, the door is to remain open until the obstruction is no longer there.

3.3.4 User Interface

Many buttons will be needed for this elevator system. Currently the elevator has four floors. There will need to be a call button on each of these floors (4 total), a button inside the elevator for each floor (4 total), plus an emergency stop button. Altogether, nine buttons are specified. A 4x4 button matrix would give more than enough buttons. If more than the nine are needed, it will be very easy to use the remaining buttons on the matrix. These buttons will be mounted outside of the elevator for testing and easy access.

3.3.5 Power

There are two main power regulators within the current design 5V and 12V. Both of these regulators will get the same DC power in from the power plug giving both sides up to 1.5A to current and there rated voltages as an output. On the 12V line only the DC motor and the respective motor controller will be connected to it, but the motor will use most of the available current from the regulator. The 5V line will power the rest of the device including the servo for moving the door, the buzzer, LED indicators,two IR sensors, and the microcontroller.

3.4 Test procedures

Thus far, we have tested some systems, components, and circuits. We have tested the LED circuit, the IR distance sensor, the servo motor, and the button matrix. The DC motor could not be tested without the drive for said motor; the ADC for the IR distance sensor was not tested do to time; the IR break beam sensor was not tested do to not having a sensor yet; the the magnetic tester was not tested do to not having the female ps/2 port; buzzer was not tested do to not having buzzer.

3.4.1 IR distance sensor

The IR distance sensor output a voltage that varies based on how far the infrared beam sensed it traveled. A test was conducted with the sensor supplied 5 V, and the output voltage being measured on the digital multimeter. A meter stick was used to measure the distance, and it was laid flat on the table with the sensor at the 0 cm mark. The datasheet[1] stated that a white surface provided the most accurate readings because it was more reflective than darker shades. A notebook was moved at set intervals from the sensor and the voltages were recorded. The results of the test are shown in Table 3.

3.4.2 LED circuit

The LED circuit shown in Figure 4 was tested independently to verify its correct function. The 1 kΩ resistors were present to limit the current through the diodes. The polarity of the LEDs were designed so that the 328PB would sink current instead of source current. The microcontroller can only output so much current, so by sinking instead of sourcing the current this should not be an issue. It does mean that a low signal (0 V) would be output to light an LED, and a high signal (5 V) would turn it off. Each LED was connected to a different pin of the microcontroller, in this test the pins were PORTD0 through PORTD3. The code used the ”_delay_ms()” function to turn all LEDs on for 1000 ms (1 second) and then off for 1000 ms. The output pins should produce a 0.5 Hz square wave, provided the correct CPU frequency was defined in the code.

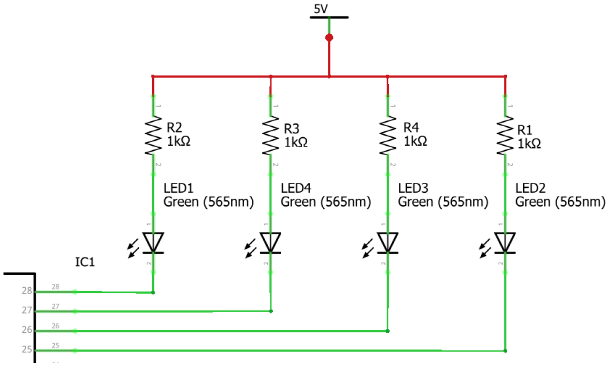


Figure 4: LED schematic

3.4.3 Servo motor

To test the servo motor with the 328PB, a timer had to be set up in PWM mode. The datasheet for the SG90 servo motor specifies a 50 Hz signal, with duty cycles between 5% (1 ms) and 10% (2 ms). Initially, timer 0 was to be used, but even with the largest prescaler a frequency of 50 Hz would not be possible. This is shown in the calculations below. The maximum prescaler is 1024, so a 1250 prescaler would not be possible.

$$\begin{aligned} F_{OCO} &= \frac{f_{clk}}{N(256)} \\ 50Hz &= \frac{16MHz}{N(256)} \\ N &= \frac{16MHz}{12,800} \\ N &= 1250 \end{aligned} \tag{1}$$

Calculations were found online [2] that used timer ticks to calculate the frequencies, shown below. Timer 1 was used as it is a 16 bit instead of an 8 bit timer. In Fast PWM mode, timer 1 has 65,536 total ticks.

$$\begin{aligned}
freq &= \frac{f_{clk}}{N * ticks)} \\
N &= \frac{16MHz}{50Hz * 65,536)} \\
N &= 4.9 \\
\text{rounded up to the next highest prescaler:} \\
N &= 8 \\
\text{plug N back in to find frequency} \\
freq &= \frac{16MHz}{8 * 65,536)} \\
freq &= 30.5Hz \\
\text{30.5 Hz is too slow, should be 50 Hz} ticks &= \frac{16MHz}{8 * 50Hz} \\
ticks &= 40,000
\end{aligned} \tag{2}$$

So The ICR1 register should equal 40,000 with a prescaler of 8 to produce a 50 Hz square wave. The duty cycles were just percentages of that value, so the OCR1A register should equal 2000 for a 5% duty cycle, and 4000 for a 10% duty cycle.

3.4.4 Button matrix

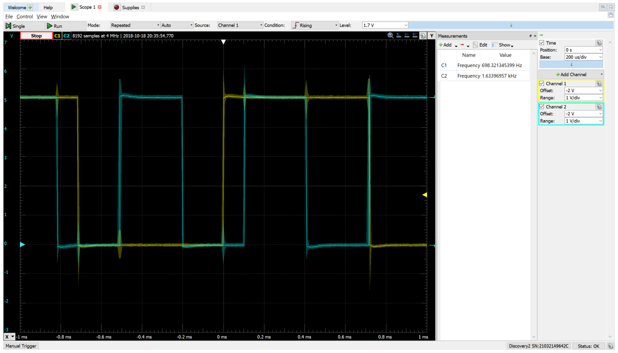
To test the 4x4 button matrix the 8 pins (pins [1:4] for rows; pins [5:8] for columns) were connected to PortD [0:3] (rows) and PortC [0:3] (columns). Using timer0 (PD6) and timer2 (PB3) on the 328PB to send a signal with varying frequencies based on what row and column a button was pressed on. Table 1 and 2 shows the different frequencies for each row and column on the 4x4 button matrix. Using the built in oscilloscope on the analog discovery 2 we were able to measure the frequencies of the signals being produced on timer0 and timer2. Figure 5 shows the signals being produced after a button press.

Table 1: IR Distance Sensor Voltages

Row	**Frequencies (Hz)**
1	697
2	770
3	852
4	941

Table 2: IR Distance Sensor Voltages

Columns	Frequencies (Hz)
1	1209
2	1336
3	1477
4	1633



Mode: Repeated | Auto | Source: Channel 1 | Condition: Rising | Level: 1.7 V

8192 samples at 4 MHz | 2018-10-18 20:35:54.770

Measurements

	Name	Value
C1	Frequency	698.321345399 Hz
C2	Frequency	1.63396957 kHz

☑ Time
Position: 0 s
Base: 200 us/div

☑ Channel 1
Offset: -2 V
Range: 1 V/div

☑ Channel 2
Offset: -2 V
Range: 1 V/div

Discovery2 SN:210321A9642C | Status: OK

Figure 5: Signal from 4x4 matrix

4 Experimental Results

4.1 IR distance sensor

A stepper motor was initially considered for controlling the elevator's height, but since it might not produce smooth movement a DC motor was chosen instead. This has not been tested yet, but it will be once a motor controller is received.

A servo motor[3] was chosen to control the door. Depending on the positive pulse width it can be in many different positions. The servo replaced the previous solenoid selection because of the seduced current requirement and the lack of extra hardware requirements. Although the it is slightly more difficult to program compared to the trivial driving of the solenoid but, the servo takes time to move making it easier for safety checks and responses.

5 Discussion

5.1 Future test plans

Most of the tests specified involve hardware and software working together. Very few tests, at this point, involve just hardware or just software. The subsections are divided roughly by subsystem or component. Since tests often involve multiple subsystems, they are grouped with what was considered most relevant.

Table 3: IR Distance Sensor Voltages

Distance (cm)	**Voltage (V)**
5	3.11
10	2.2
15	1.6
20	1.2
25	1.0
30	0.890
35	0.780
40	0.680
45	0.620
50	0.530
55	0.480
60	0.440
65	0.400
70	0.320
75	0.050
80	0.030

5.1.1 DC motor

- When connected to the motor controller, try making the motor rotate in both directions and verify that it does so.

- Measure the time it takes for the elevator to travel from the bottom floor to the top, and top floor to the bottom. These times should be the same. Repeat the measurements with different levels of PWM to determine which speed looks most reasonable.

- Using PWN to slow down the speed of the elevator as it approaches the desired floor as to not stop abruptly.

5.1.2 IR distance sensor

- The sensor by itself had been tested by using a voltmeter to measure its voltage at specific distances.

- After connecting the sensor to the micro controller ADC, test the readings at different distances.

- Record the readings again, this time with the sensor at the top of the elevator shaft. Get positions for each floor based on the distance from the sensor.

- Once the positions are known, have the elevator travel to each floor from above and below (if possible) to measure how far the elevator is physically from the desired position. The elevator should line up from both directions.

- Set a distance for the motor to stop at, and see how long it takes for it to stop, and record how far it stopped from the desired location.

5.1.3 Buzzer

- Observe the timer output from the micro-controller and verify that the frequency of the waveform matches what was specified.
- Send the elevator to a floor and verify that the buzzer buzzes when it reaches the floor.
- Press the emergency stop button, verify that the buzzer buzzes until the button is pressed again.

5.1.4 Emergency stop

- Press emergency stop to stop elevator. Try to press buttons to call elevator and send it to another floor. The elevator should not move until the emergency stop is disengaged.

5.1.5 Security: magnetic stripe reader

- With the reader connected to the micro-controller, record the data received from swiping a card. This card and its data will be considered valid. Try to get a match by swiping other cards. Ideally they should not match and should be rejected.
- The elevator should not be called until a valid card is swiped. Try different input combinations to move elevator or open a door before a valid card is swiped. Also try this when an invalid card is swiped. Repeat the same tests by swiping a valid card and ensure that the doors open and the elevator does move (if not on the floor).

5.1.6 Door

- The door should open shortly after a floor is reached. This time will be measured and adjusted as necessary, and must be after the elevator stops moving.
- The door should then close some time after a floor is reached and remained closed until a valid card is swiped. This time will also be measured and adjusted to ensure passengers have enough time to safely exit the elevator.
- The door should remain closed while moving, and the elevator should continue to its destination before moving to another floor. The call buttons should be pressed for different floors while the elevator is moving to make sure it continues and the doors remain closed until stopped.

- If at a floor with the door open and something is detecting in the path of the door, the door should remain open and the elevator should not move until the path is clear for a time. The call buttons should be pressed while the path is blocked to test if the elevator does actually move. The time it takes the door to close after the path has been cleared will be measured to ensure it can safely be cleared and remain clear.

5.1.7 Buttons

- The button matrix should be tested independently to ensure every button can be properly detected.
- The emergency stop button should halt all function and continue to until it is pressed again. This should be tested when the elevator is in different positions, such as when it is moving between floors and stopped at a floor.

6 Conclusions

We still have a few more components and systems to test at later dates. The tests we have conducted so far have been successful and we are happy with the results we have gotten.

7 Reflection

A few things were changed from our initial plans for what will be used in the design. While thinking about how the elevator would operator it was recommended that we use a DC motor instead of a stepper motor for a smoother ”ride” for the elevator. We felt that was a valid point and switched our future testing to the DC motor. Looking into the solenoid we realized too much current would've been used to power the solenoid, so instead we opted to use a servo motor; making for easier programming and a simpler circuit.

8 Appendix

8.1 Team member contributions

The individual contributions of team members were not measured by time. Instead, a general outline of what each member has done is provided in Table 4. The project so far has mostly been broadly designing subsystems, making decisions on how to implement the subsystems, and testing individual components, which all group members have contributed to.

Table 4: Team member contributions

Member	Contributions
Jacob Aubertine	Wrote entirety of servo motor report and this report. Will focus on both hardware and software.
Mathieu Bolduc-Clayton	Created hardware and software block diagrams. Will focus on hardware.
Sal Fernandes	Created concept block diagram. Will focus on software.

References

[1] "GP2Y0A21YK0F Distance Measuring Sensor Unit data sheet," Sharp.

[2] "(TUT) ATMEGA328PB, 16-bit timer, servo, FastPWN, HXT900," avrfreak.com.

[3] "Servo Motor SG90 data sheet," Tower Pro.

[4] "ULN2003 High-Voltage, High-Current Darlington Transistor Array data sheet," Texas Instruments.