Michael Chen - SID: 861108671
Brandon Carlos - SID: 860888301
EE - 128 - 021

Laboratory

Mini-project

R E P O R T

Elevator

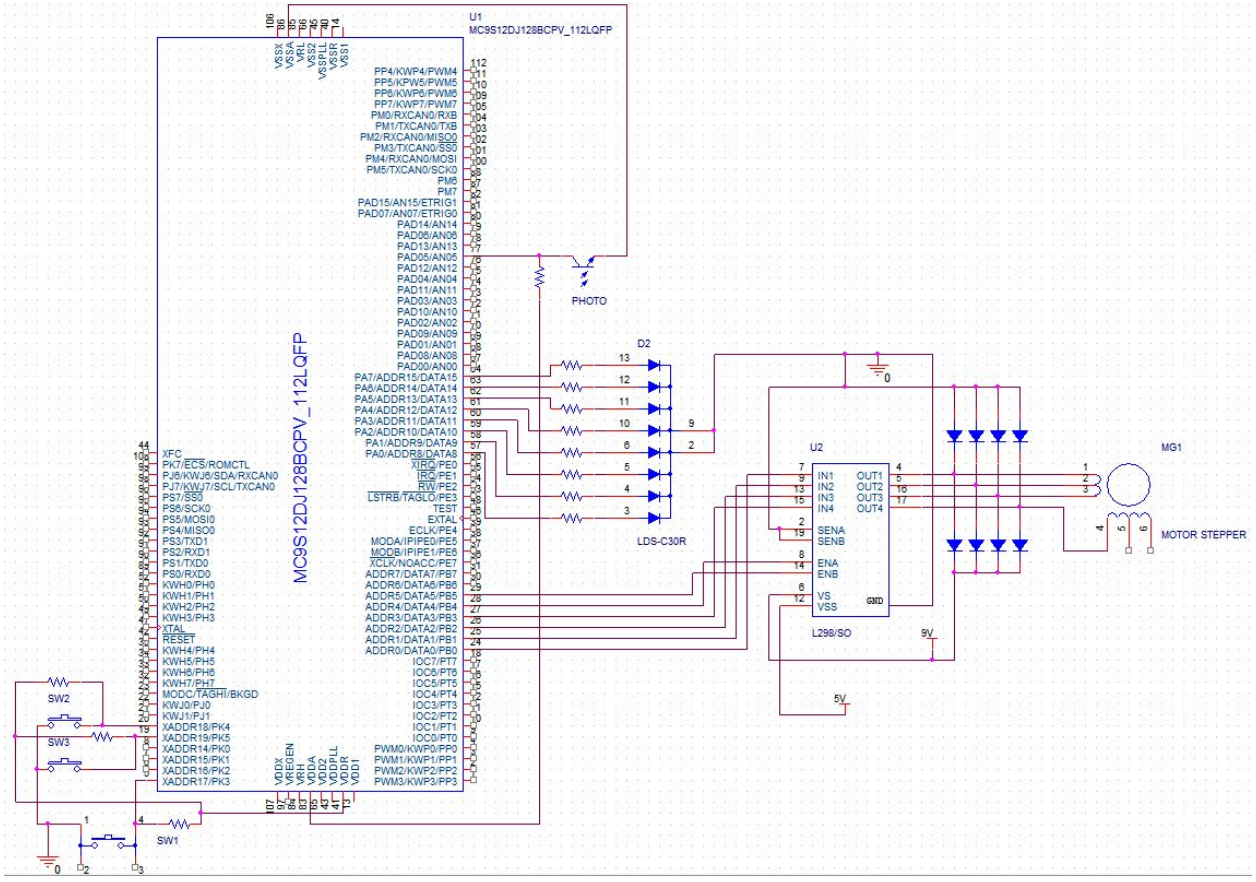
PROJECT DESCRIPTION

This project is based on an elevator system in small scale application. By using lego blocks to make the frame of the 3 level elevator, a platform is raised to the desired level by the press of a button and stopped by use of a photoresistor. A string with a stepper motor is used to pull the platform to each level. Once the platform arrives to the appropriate level, a 7-seg BCD display shows the current level of the platform.

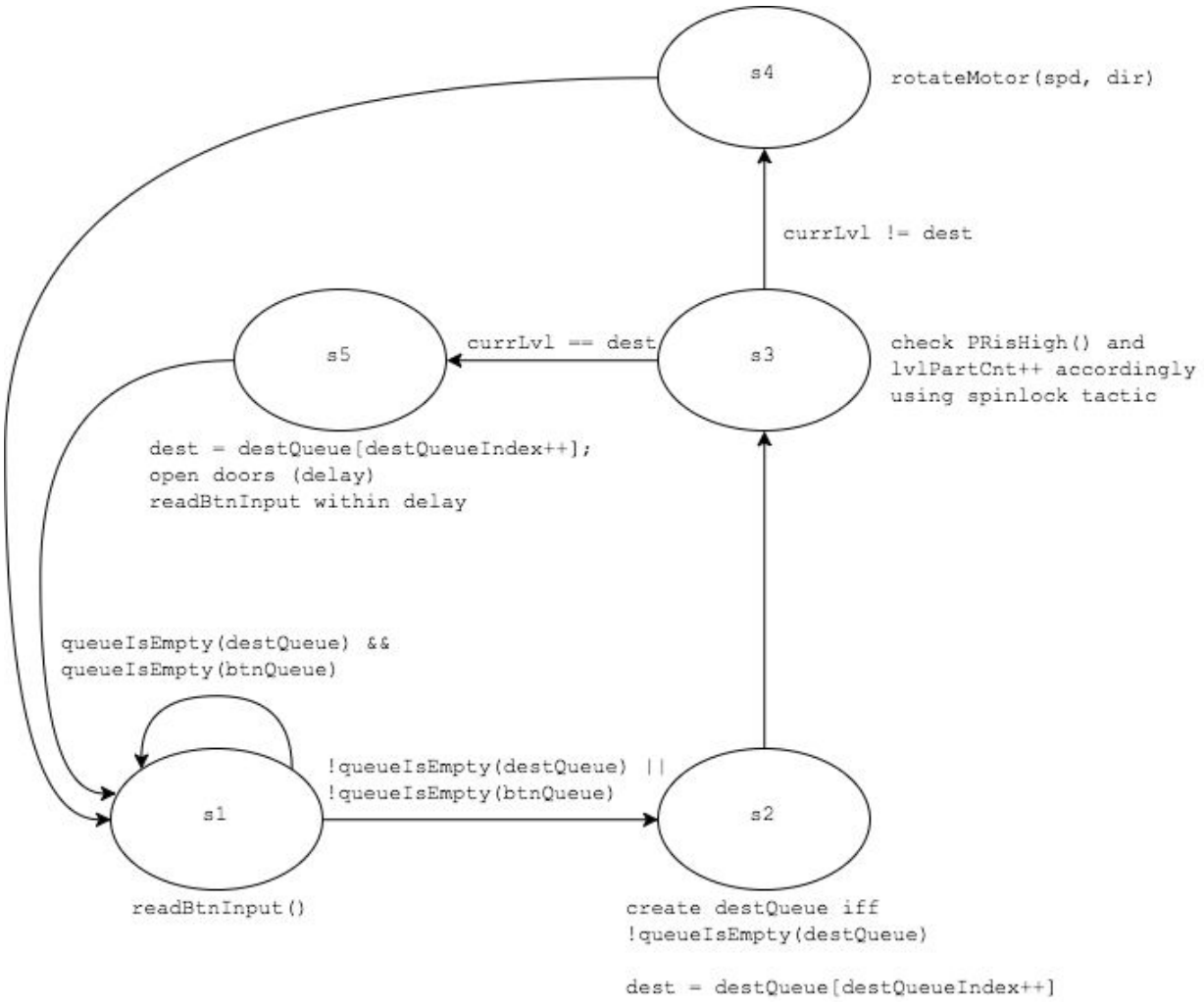
The goal of this project was to create a structure with multiple levels and a platform that can be raised to each level mimicking the function of an elevator. The automated motion of raising the platform needs to stop at the correct position and the correct level of the structure. The use of buttons activates the platform and multiple button inputs can be taken.

SYSTEM DESIGN

Youtube demonstration: https://goo.gl/JfNxLp



IMPLEMENTATION DETAILS



Priority queue implementation:

- `btnQueue`
  - Store button presses in order of priority (FIFO)
  - enqueue w/ post incrementation of btnQueueIndex
    - `btnQueue[btnQueueIndex++] = 1; // 1 | 2 | 3`
- `destQueue`
  - Elevator listens to this queue of destination levels
  - Created from btnQueue
    - Level 1
      - `destQueue = bubbleSort(btnQueue, ascending);`
    - Level 2
      - `destQueue = btnQueue;`

        - Level 3
            - `destQueue = bubbleSort(btnQueue, descending);`
    - pop w/ post incrementation of destQueueIndex
        - `dest = destQueue[destQueueIndex++];`

`lvlPartCnt`

- Each level is defined as three lvlPartCnt(s)
    - 1. Photoresistor is initially covered by the lego blocks - low analog signal
    - 2. Photoresistor receives light - high analog signal
    - 3. Photoresistor is covered by lego blocks - low analog signal

`char PRIsHigh();`

- A form of spinlocking
- Increments lvlPartCnt
- Only allows a low/high analog signal to be received once on the rising edge.
    - `If a high analog signal was received`
        - `lvlPartCnt++`
        - `Lock all future high signals`
        - `Unlock all future low signals`
    - `If a low analog signal was received`
        - `lvlPartCnt++`
        - `Lock all future low signals`
        - `Unlock all future high signals`
    - This pseudo code describes two flags. In our implementation, a single flag is sufficient to achieve the desired effect.

Example of Scheduling Algorithm:

If a person is on level 1, the only direction the elevator can go is up. If level 3 is pressed, the elevator begins moving. On the way from level 1 to level 2, if level is two is pressed, the elevator stops and picks up the person on level 2. The elevator then resumes ascending to level 3.

```
btnQueue: 1 -> 3 -> 2
destQueue: 1 -> 2 -> 3
```

This also applies to the following similar queue scenario:

```
btnQueue: 3 -> 1 -> 2
destQueue: 3 -> 2 -> 1
```

TESTING/EVALUATION

We tested the project incrementally to ensure that the final build would come together smoothly.

We first tested the motor function which required the use of the DC power supply. Receiving the signals from the buttons were tested next which only required the buttons and dragon board input pins and ground/5V connections. The same circuit was used for checking the photoresistor but the analog input pin was used instead.

After the construction was complete and pieces were in place, the motor was tested again for the direction in which it was pulling in accordance with the button and its level. The photoresistor was ignored for this part until the button and motor we cooperating correctly. Finally, the photoresistor was checked and fixed as best as we could.

Without taking into consideration the cheap quality of the photoresistor, our requirements for this project were met. After testing each part of this project individually and putting it together to test every piece as one whole, the system did function as intended.

DISCUSSIONS

The very first problem we ran into was getting the motor to function properly. Rather than altering the code this time, we decided to use the code from a previous lab without altering any of it. This solved our initial problem but more were to come.

The next of which was creating the code for the logic of button inputs and creating an efficient way of moving the platform in complicated situations. Arrays were used to arrange the inputs in closest level if multiple inputs were pressed and to keep track of the current and next levels that need to be reached.

Our next problem was a hardware mistake. We mistakenly assumed that the input and output of a double pole single throw push button were on opposite sides of each other. This caused the system to constantly receive a high signal from the button. The problem was fixed when we connected the pins on the same side.

Our final problem that persistently arose was receiving the proper signal from the photoresistor. This component works by allowing leakage current to flow when light hits the surface of the resistor and prevents that current to flow when there is no light. Despite the various alterations we made, the photoresistor would not cooperate and seemed to function properly at random times. We checked the wiring, changed the analog signal level, increased the light by use of flashlights, decreased the light, and changed the structure of the elevator with no lasting success. The best results we had were from the use of flashlights and hoping for the best when reloading the program onto the board.

Limitations of the design include the photoresistor. We could have used a different component, such as an optocoupler, which would have saved time. An improvement in the build would have been to create a more stable platform as uneven weight distribution shifted the platform's position during the upward motion.

ROLES AND RESPONSIBILITIES OF GROUP MEMBERS

Michael's responsibilities:

- Create elegant code and contribute to the design and construction of the elevator
- Collaborate with partner on how structure should be made and placement of components
- Build and test code progressively to check for bugs or problems with design
- Contribute to the creation of the presentation slides and split presentation evenly
- Cooperate with partner on the completion of the project report

Brandon's responsibilities:

- Focus on hardware and provide extra materials as well as add to code
- Checking and fixing building design and providing suggestions on component placement
- Verifying code correctness by testing individual features of project and adjusting when necessary
- Working with partner on presentation slides as well as the project report

BRIEF CONCLUSION

We can fully appreciate the learning process of assembling the small working parts of a project into a finished product. The project has provided us with some insight of planning a project, building that idea, collaborating with a colleague, and adjusting the project as necessary to ensure deadlines are met. With the presentation, we not only acquired public speaking skills but it also allowed us to present our hard work to our fellow peers. This report allowed us to document and record the process for future reference or demonstration. Beyond the classroom, we can see the possibilities open to us with devices such as the Dragon and Arduino boards along with the knowledge gained here. With this project comes appreciation and a learning process from weeks of hard work and also valuable skills that we can apply in the real world.