MECH 458 Project Report
Department of Electrical Engineering

Instructor: Kunwu Zhang

Date: April 22nd, 2020

Author:

1.	Connor Wiebe
Student Number	V00872959
2.	MINH DAI TRAN
Student Number	V00928014

Group 6

Executive Summary

The purpose of this project is to create a system for sorting objects on a conveyor belt into a sorting tray.

A circuit was designed to interconnect an at90usb1287 microcontroller with user input buttons, the system's sensors, and two motors.

The interface consists of two buttons, to start/stop the sorting process, and to initiate calibration. A DC motor drives the conveyor belt and a stepper motor drives the sorting tray. An analog reflectivity sensor determines the object material, and an optical sensor at the belt exit controls the conveyor belt.

A program was written in C to operate the sorting process.

The system successfully fulfills the design requirements: To sort between four objects types of different materials, in any order, queuing at least 4 items on the belt at a time. This was validated through our testing procedure. At the end of the report, we discuss the system limitations and features that we could have included or improved on.

Table of Contents

List of Figures

List of Figures

1. Project Description

The purpose of this project is to implement an object sorting system. The system consists of a conveyor belt, a sorting tray with four sections for four different materials, an exit sensor at the end of the conveyor belt and a reflective sensor.

Four different types of objects: *black, white, steel, and aluminium* are placed on the conveyor belt and sorted placed onto their respective segment of the sorting tray.

2. System Summary

2a. High level block diagram

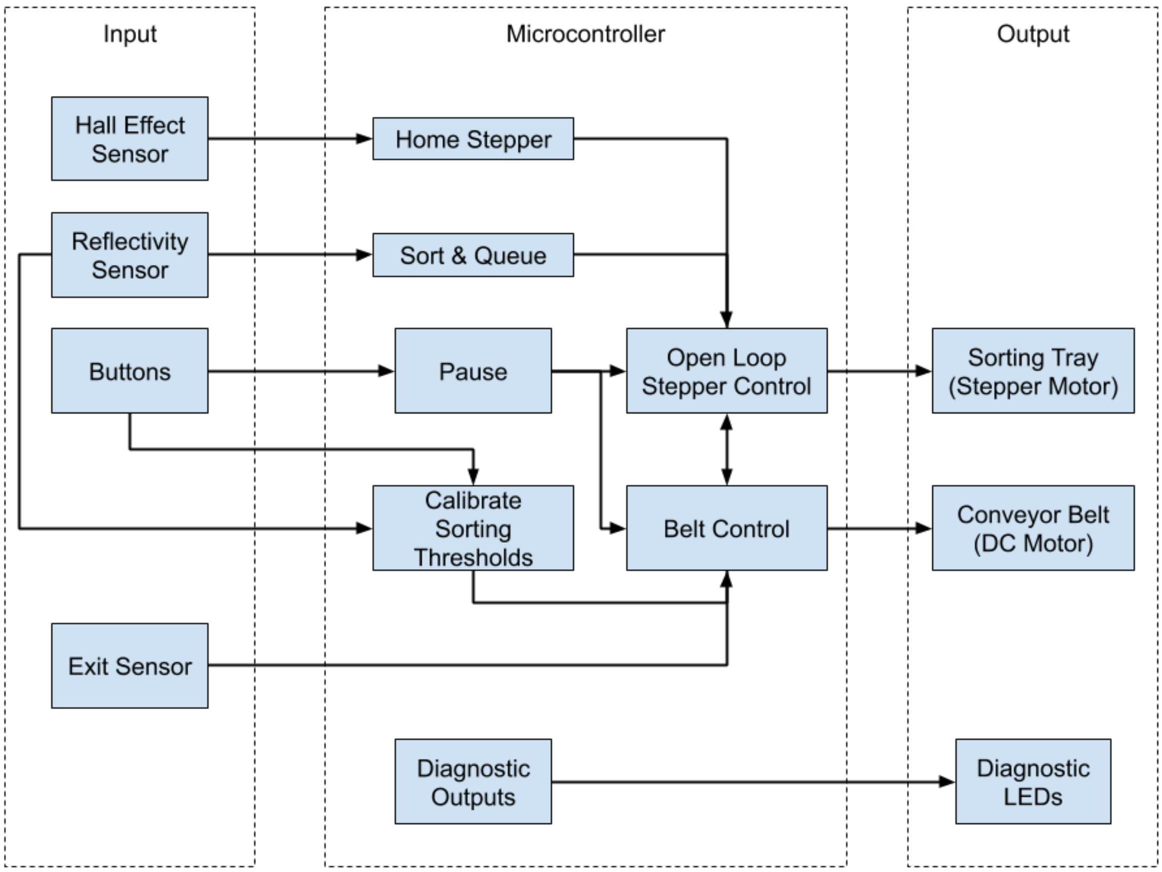

Figure 1: High Level Block Diagram

2b. Circuit Diagram

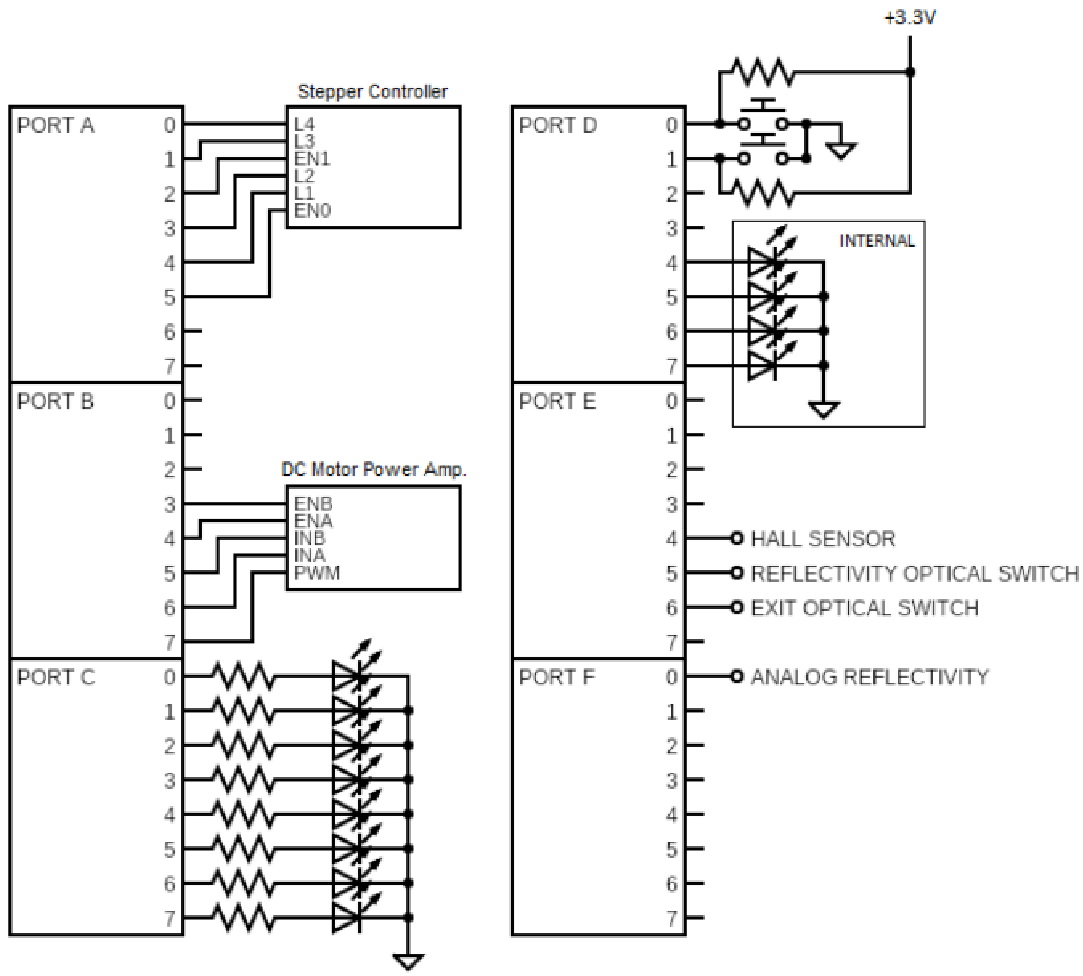

Figure 2: System Circuit Diagram

2c. System Algorithm

- Pause:
  - Button Pressed:
    - Stops Conveyor Belt
    - Moves tray to its destination
    - Wait for the pause button to be pressed again
- Ramp Down:
  - Button Pressed:
    - Set RAMP_DOWN flag
  - When the queue is empties:
    - Stops motors
- Reflectivity Sensor:
  - Laser tripped:

        - Start conversions
        - Continuously store lowest conversion value
    - Laser Closes
        - Stop Conversions
        - Classify based on minimum value
        - Add to main queue
- Exit Sensor:
    - Laser tripped:
        - If tray is not in position, stop conveyor belt
    - Laser closes
        - Read from the queue
        - Run conveyor belt
- Hall Sensor:
    - Homing
        - Turns clockwise until the sensor fires, then sets the home position, and offset to the center of the nearest bin
        - Run-Time adjustment:
            - Resets the step position whenever the sensor fires, if turning clockwise
- Conveyor Belt:
    - Only stops if an item is preset, and the tray is not in position for it to be dropped

2d. Flow Chart

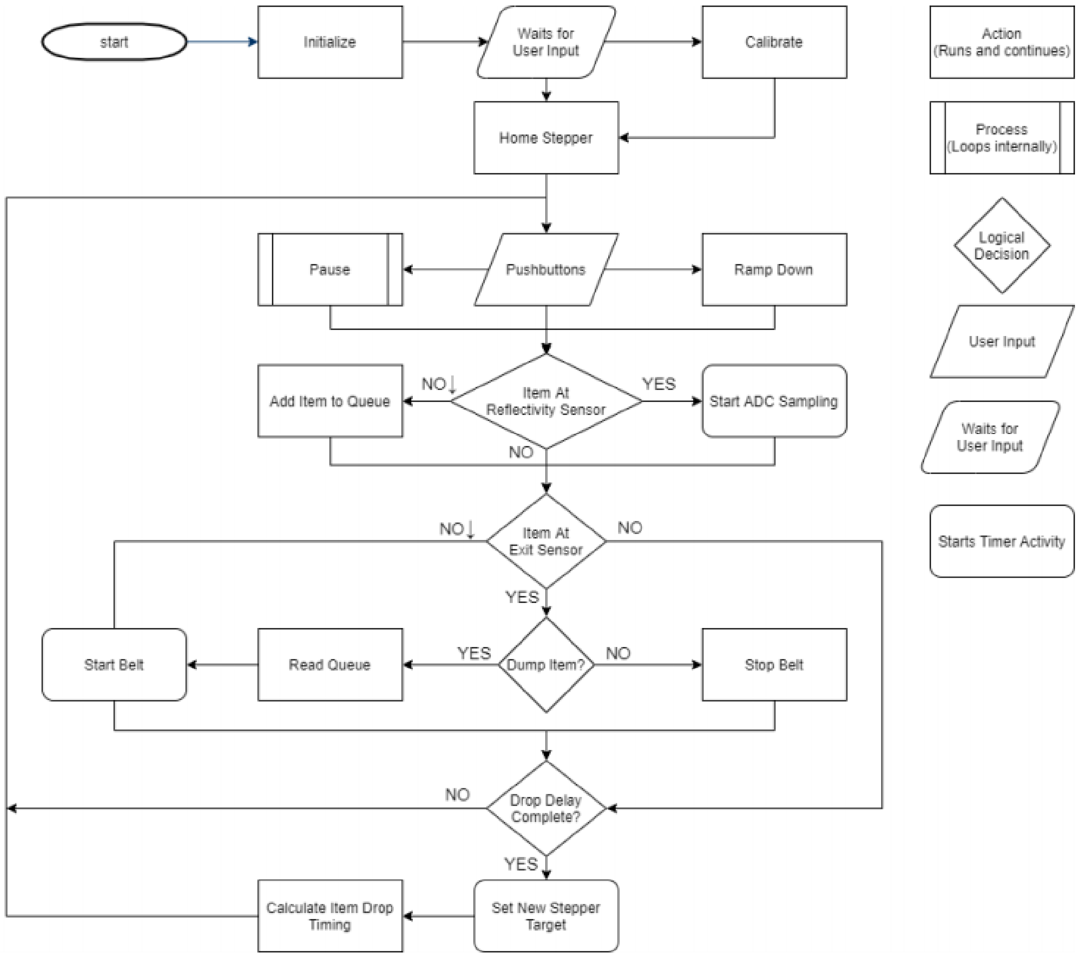

Figure 3: System Flow Chart

3. Performance Specification

The stepper motor position control uses a trapezoidal acceleration curve, so the time to move between two points is directly proportional to the acceleration constant, until the peak speed in the curve reaches maximum speed.

This stepper driver is double stepping, so there are 400 steps per revolution.

After a dispense command dispenses an item from the belt, the item must travel from the belt exit to the tray; a delay which we call the item drop time. During the drop time the stepper is delayed from starting its next movement.

A dispense command starts the belt when the tray has come within a certain number of steps to its destination - this is the angle buffer. At zero, the item is dispensed only after the tray has stopped, and increasing the buffer allows the item to be dispensed while it is still in motion, to prevent the belt from having to wait for the item to fall from the belt.

The system overall is limited by the speed at which the stepper can move - the conveyor belt is capable of delivering very many items in a short amount of time. Therefore, to maximize performance, the acceleration curve should be optimized for point-to-point speed, and the timing should be calibrated to minimize stepper downtime - i.e. the stepper should always be moving.

3a. Maximum Specification

Acceleration of the stepper was tested to function up to 1700 steps/s^2, or 4.25 r/s^2 beyond which stuttering occurred. This acceleration is a constant value, for simple linear acceleration curves.

The DC motor is capable of running at maximum speed, only being switched on and off. What speed and acceleration this represented was not measured.

The drop time should be set to an accurate measured drop time for a dispense command. This prevents any unnecessary waiting after a dispensation is completed. If the angle buffer is exactly calibrated, this delay will have no effect.

The angle buffer should be set to an angle such that a deceleration over that angle will finish in exactly *drop time*. Therefore the bin will reach its destination at the same time the item lands in the bin, given the trapezoidal acceleration curves.

Instead of being measured, this value can be calculated by summing the elements in the acceleration curve’s frequency array - the index where the sum is larger than drop time is the optimal angle buffer.

3b. Recommend Specification (For remote testing)

For stable condition acceleration is set to 1000 steps/s^2, to allow for differences in the stations’ stepper motors, with a large margin of error.

The DC motor is set at 62% of its maximum speed, which we subjectively considered to be ‘reasonably fast’ and which did not cause overshoot problems when braking at the exit sensor.

In the remote test, the drop time is set to 400ms, a deliberate overestimate, to ensure correct sorting. From watching video of the machine, the true value is near ¼ of a second; the difference between these times would be directly subtracted from the sorting time of each item - approximately 2 seconds over 16 items.

To ensure the objects land in the center of their bin, both the angle buffer and drop time should be run with a safety margin - the drop time 100ms above the measured value, and the angle buffer set to 50% of its maximum accurate value.

4. Testing Procedures

Procedure:

1) Calibration:
    a) Start the calibration process by pressing button 2.
    b) Feed several black items through the reflectivity sensor then press button 2. Repeat this step for white then steel items.(Ignore aluminium)
    c) Read the reflectivity value from the LED display. In order, they are black, white and steel. Make sure that these values make sense.
    d) If the reading on the LED display matches with our expected value that means the sensor is working properly. Black and white items are very to each other in values and we need to make sure the system can differentiate between these two items.
2) Press button 1 to start the program:
    a) Watch the stepper motor home, note whether or not it spins clockwise and stops at the center of the black bin.
    b) If the stepper home to the correct position, that means the system has the correct position for each bin for the sorting process later.
3) When both the reflectivity value make sense and the stepper homing works correctly, start the sorting process. Load items on the conveyor belt, 4 at a time to a total of 48 items.
4) After the system finishes sorting, check the tray to make sure that all 48 items are sorted correctly. Check the count variable and make sure that it manages to keep track of all 48 items.
    a) If the system keeps count of all 48 items as well as sort those items correctly. This means that the system fulfills its sorting functionality with no error.

5) Press button 1 to start the system again, place a few items on the sensor. Press 1 again and observe if the system pauses. Press 1 again and make sure the system starts from where it left off.
    a) If the system pauses and continues where it left off with button 1 press, this means that the pause functionality works as intended. The user can pause and continue as they want by pressing a button.

5. System Limitation

The ramping functionality has been removed so that the system can successfully sort the objects in the limited amount of testing now that we have limited testing time due to remote work.

The optimal drop time has been neglected so that the stepper motor can operate in conjunction with the DC motor. We also had to alter our testing procedure to find the optimal speed for both motors. This optimal speed test procedure would involve finding the maximum speed the system could operate while yielding an accurate sorting result. These limitations result in the system unable to operate any faster than it already is.

The system may not also be able to calibrate correctly under extreme humidity and dusty conditions.

6. Reflection

There are many things we could have done to further improve our design, espeicallying in the speed department. Due to testing limitations we were unable to implement our ramping functionality which can improve the speed of the system. Furthermore, we could further tweak our speed value on the conveyor belt and sorting tray to improve the overall speed of the system. This includes finding the optimal drop time of the items. Another feature that can potentially improve our design would be a touch screen interface that has all the values and buttons labeled on the screen.

This project was a valuable experience in both introducing us to mechatronic design and optimization. We may not have been able to do as much with the project as we first expected to but we have learned how to optimize our work load, when having to do remote work with limited testing time.