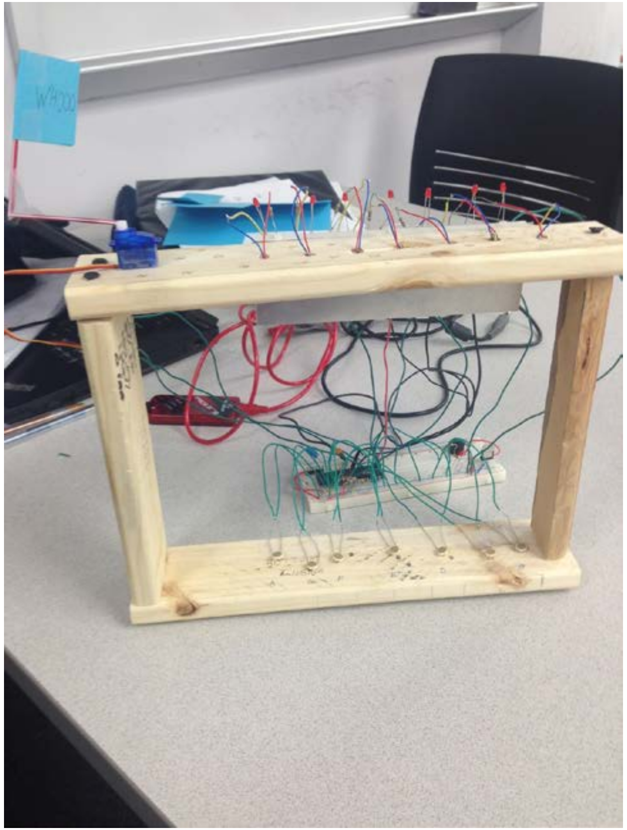
Laser Harp

Team 15

Jess Bai & Josh Keller

Project objective

When the laser beams are broken, buzzer plays the corresponding musical note, turn on the corresponding LED lights and the flag waves back and forth.



Inputs and outputs

Inputs:

7 Cds photocells - with 7 laser beams constanting shining directly on them. When the laser beam is blocked, the values that the photocells read change. Similar to the IR emitter and receiver.

Outputs:

Buzzer - makes 7 notes: A ,G, F, E, D, C, B.

7 LEDs - lights up correspondingly.

Servo motor-drived flag - waves back and forth when a note is being played.

PIC

- We used **RC2** for the **buzzer**, which is the only pin we can use because the buzzer requires PWM1.

- We set all the pins as analogs just in case, but we used **AN0 to AN6** for the **photocells**.

- We used **RD0 to RD6** as outputs for the **LEDs**.

- We used **RC3** for the **motor**/flag. Interrupts and timer0 were used to control the motor.

- 7 lasers were directly powered by the regulated voltage, so they did not go through the PIC

Results

It worked, but not perfectly.

With the waving flag, the LEDs twitch a little and the buzzer doesn’t sound very good. Without the flag, LEDs and buzzer work perfectly fine.

Advice

- Start early!
- Make sure the supplies you buy work for your power supply.
- Build the housing first.
- Check for syntax errors and typos first if something is not working.
- Try to think of every possible outcome if using if-statements.
- Use more PICs if necessary.

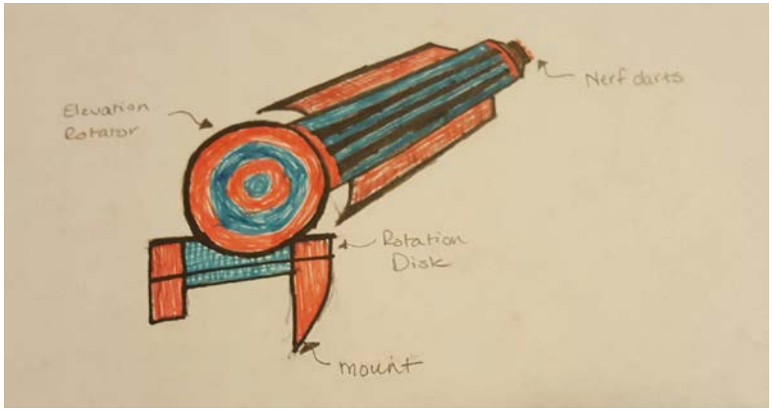
Mechatronics Final Project: Controlled Nerf Blaster Turret

By Tim Balz & Erin Scully

ROSE-HULMAN INSTITUTE OF TECHNOLOGY

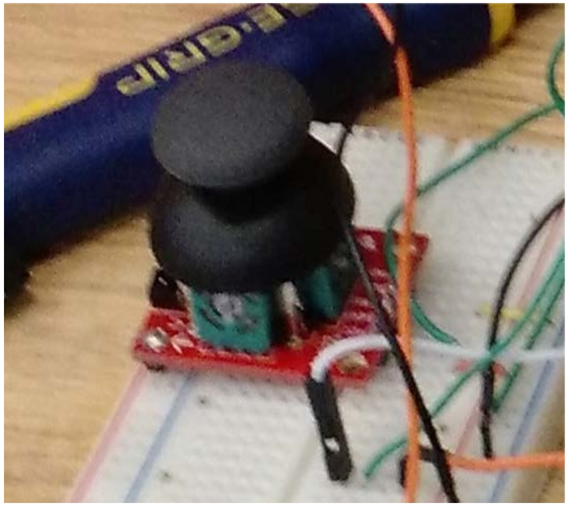
Objectives – Minimum Goals

- Turret will have full aiming capabilities
- Controlled in a smooth manner via a joystick
- Fire is controlled remotely and not with the mechanical trigger



Inputs to the System

- **Joystick -** ADC controls the positioning of the joystick with the intention of controlling the aim
- **Joystick button -** pressing the joystick acts as a trigger
- **DPDT Switch -** position switch which determines the direction of the lazy susan



Controller (PIC)

Interrupts:

- An RB0 interrupt triggers the continuous servo which loads the firing mechanisms
- Timer Interupt

PWM:

- The joystick controls whether the pwm duty cycle is increasing or decreasing on the tilt servo
- The firing servo is a continuous servo triggered on by the joystick button with a constant duty cycle

Outputs

Lazy Suzan:

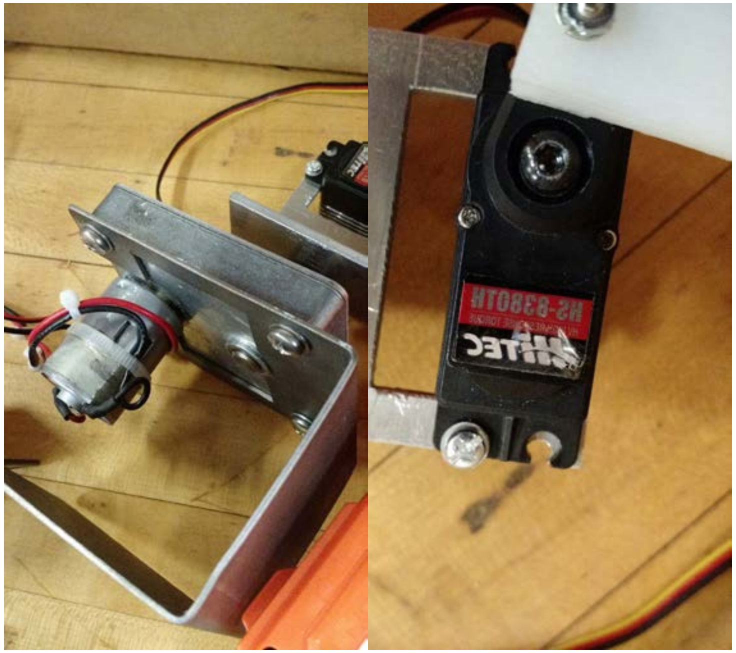
- DC motor which rotates the gun horizontally

HiTec HS-8380:

- High-powered servo motor tilts the gun vertically

HiTec HS-422:

- Continuous servo advances the next dart into the firing motors



Results

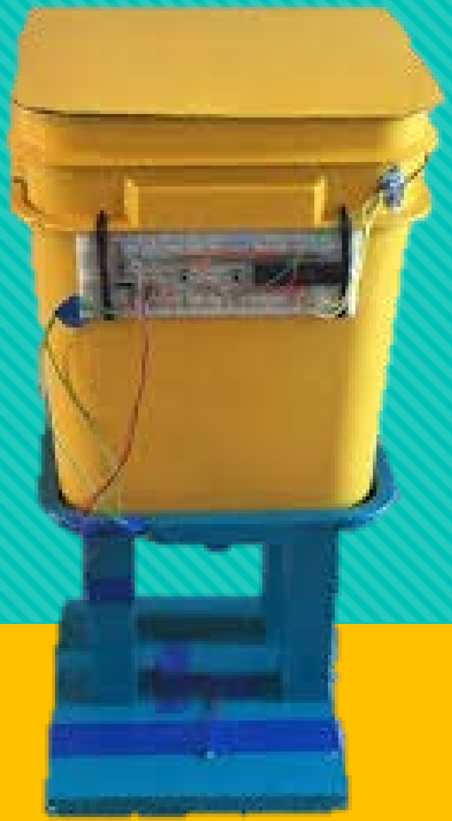
- The resulting turret can be controlled externally by the joystick and switch
- Aiming is accurate and firing the gun is smooth
- 25 dart capacity to pester people with

Advice to future newbies to Mechatronics

- Don’t fall behind
- Take extensive notes from the videos
- Study like heck for the tests - they go quick
- Do multiple simulated tests from previous quarters otherwise you wont understand how fast the test time goes by
- **Make it happen make it fun**

Automatic Trash Can

Trent Bauters
Daniel Truckai

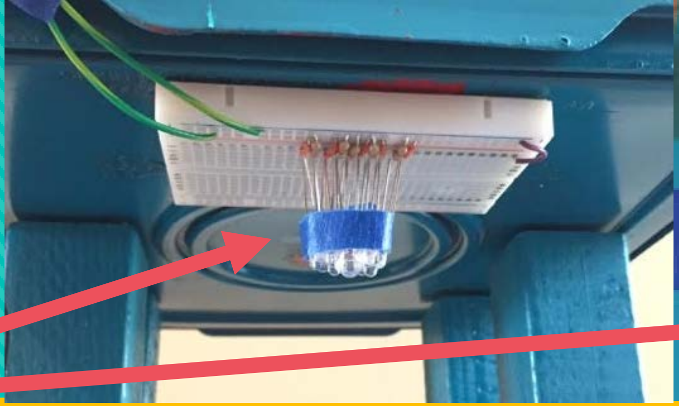


ME430 with Dr. Olson

The Objective

The objective of this project was create a trash can that could automatically open its lid by the user waving their foot under it.

Inputs sent to the PIC



An IR detector + emitters:

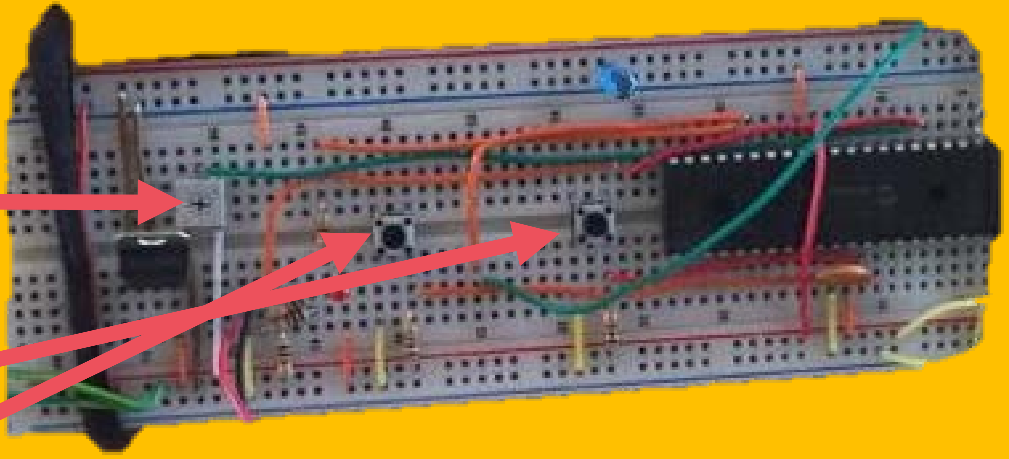
A specific voltage signal was sent through and controlled by an IR detector. An estimated 5 volts for when the infrared beam was broken by an object, and an estimated 3 volts when made.

A Potentiometer Reading:

The potentiometer was specifically set to 4 volts but could be adjusted for sensitivity..

Two Buttons:

The buttons send an input to open or close the Lid



Outputs sent from the PIC

A servo motor was told to move its arm to either 135º or 0°. Based on the specific input, the motor would stay open for 10 seconds, unless the pushbuttons were pressed.

How the PIC was used

The PIC was programmed to do the following:

Main while() loop:

Breaking the IR beam sends 5 volts to the PIC. The PIC sends a signal to the servo motor to turn 135º and sets a time for 10 seconds, then sends a signal to turn it back to 0º. The potentiometer was set to 4 volts, which adjusts the IR sensitivity reading correctly for indoor functionality. As soon as the IR beam is broke, then the loop also turns on an LED for 2 seconds to serve as an indication the PIC read the movement of the foot.

High Priority Interrupts:

An open button interrupts and breaks the circuit when pressed, telling the PIC to send a single command to turn the servo motor 135º. There is no code to tell it to close unless the close button is pressed.

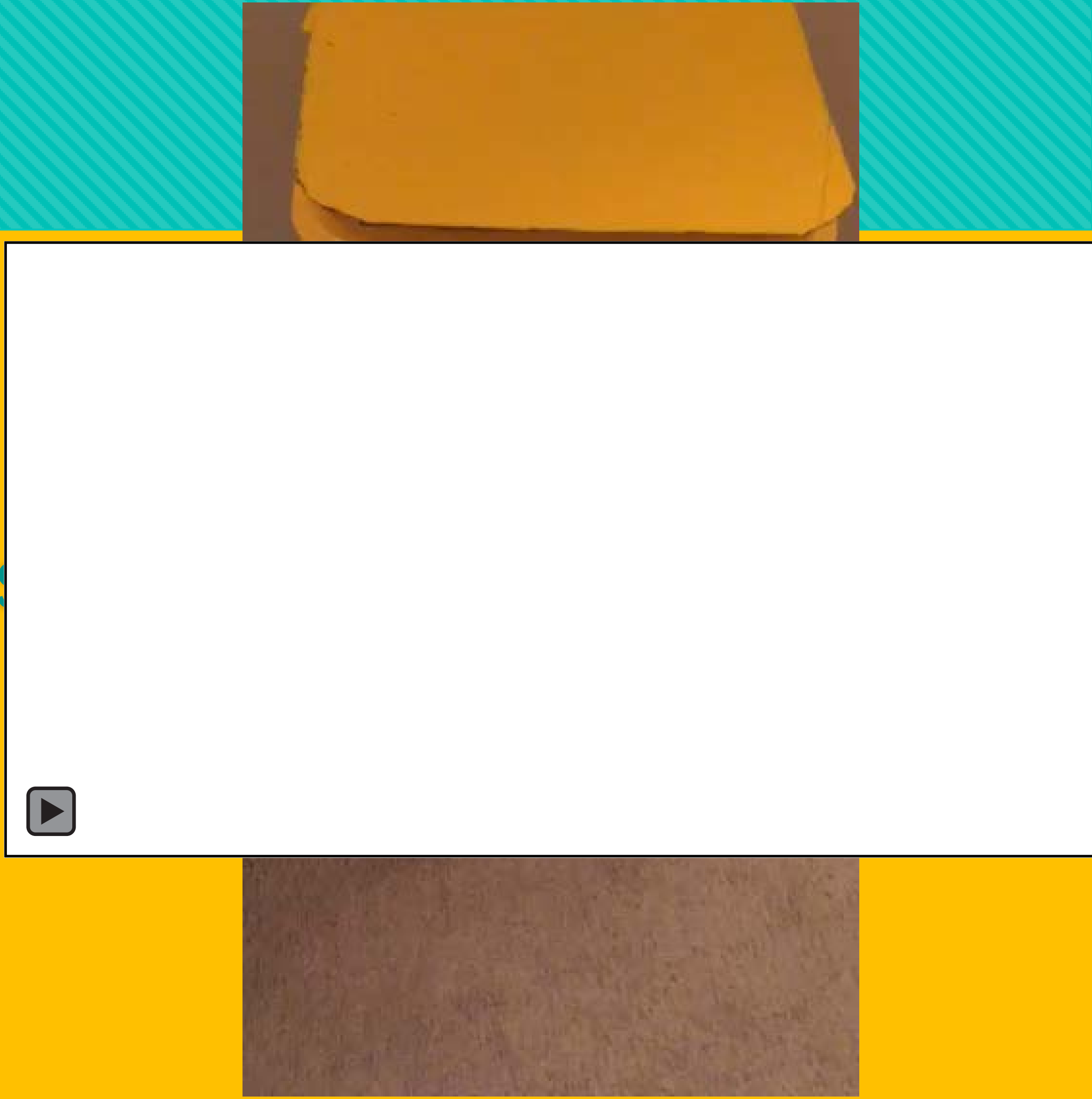
A close button interrupts and breaks the circuit when pressed, telling the PIC to turn back 0º, despite which way was used to open and turn on the servo motor to open the lid.

Results

The Automatic Trash Can worked!

The lid opened for ten seconds when the user swiped his/her foot.

The lid could also be open or closed with the buttons.



Advice

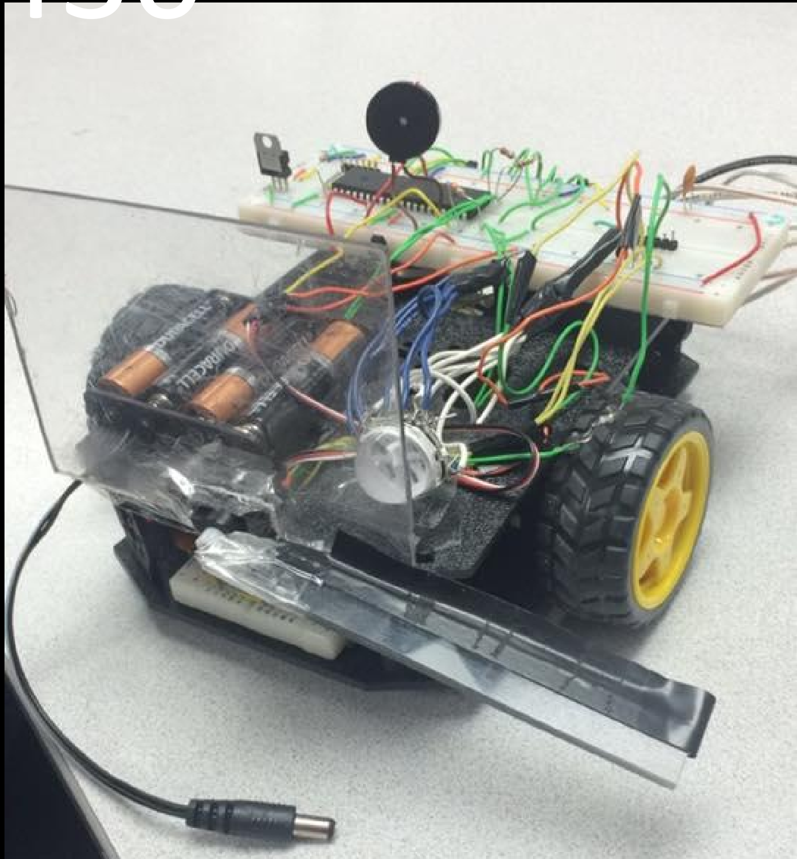
Take my advice....

Try to make something fun and unique, but don’t go overboard.

Even the simplest things can go wrong, so work carefully and thoroughly; understand everything. It will probably save you a few hours of work.

When using IR sensors for longer distances (over 6 inches), make sure to increase the resistance in the IR receiver circuit or else the voltage will not vary enough.

Mechatronics ME430
RC Cart Diana



Amaryllis Biduaka
Quinn Cunningham
Marcus Washington

Overall Objective

- Create a joystick controlled car that can go left, right, forward, and backward
- Create a car with the following features:
  - Blinkers
  - Headlights
  - Windshield Wiper
  - Horn

Inputs

Switches

- Left Blinker (RB1)
- Right Blinker(RB2)
- Headlight(RB3)
- Windshield wiper(RA5)

Horn Button(RB0)

ADC reading for Joystick

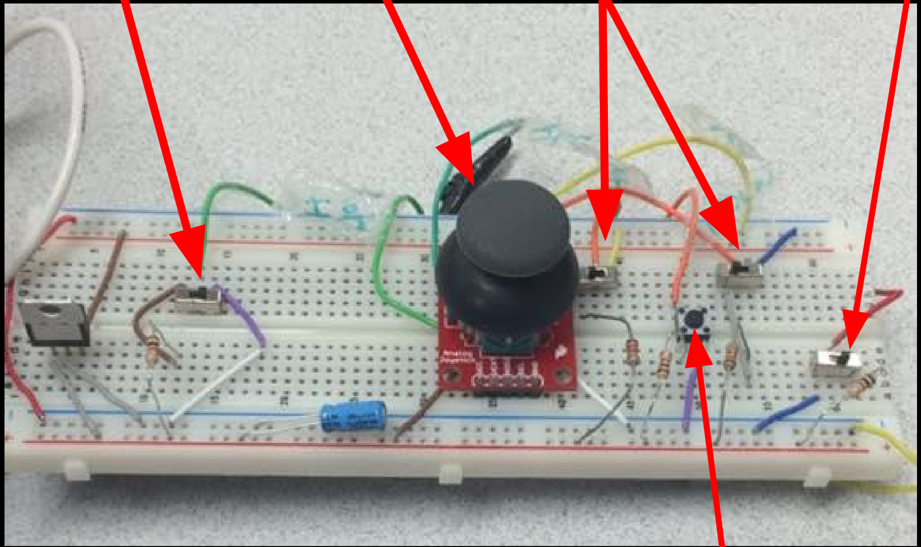
- CH_2 (RA2)
- CH_3 (RA3)

Headlight Switch

Joystick Steering

Blinker Switches

Windshield Wiper Switch



Controler Breadboard

Horn Button

PIC(s) as a Controller

- Interrupts

  -Servo Motor for windshield wiper (high)

  -Blinkers (high)

- PWMs

  -PIC 1: PWM1 and PWM2 for left and right wheel motor control

  -PIC 2: PWM1 for buzzer horn

- Timers

  -Timer3 for the blinkers interrupt to flash at moderate speed

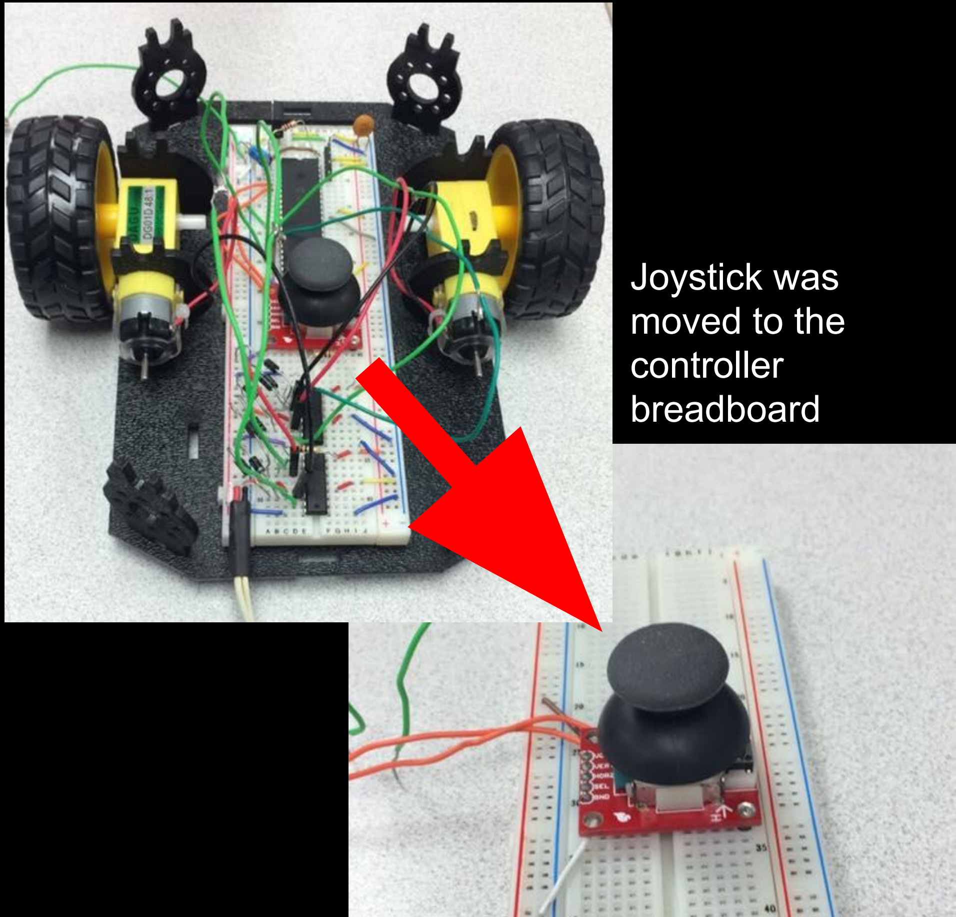
  -Timer3 for the windshield wiper to operate realistically

- Delays

  -Windshield wiper

  -ADC for motor control

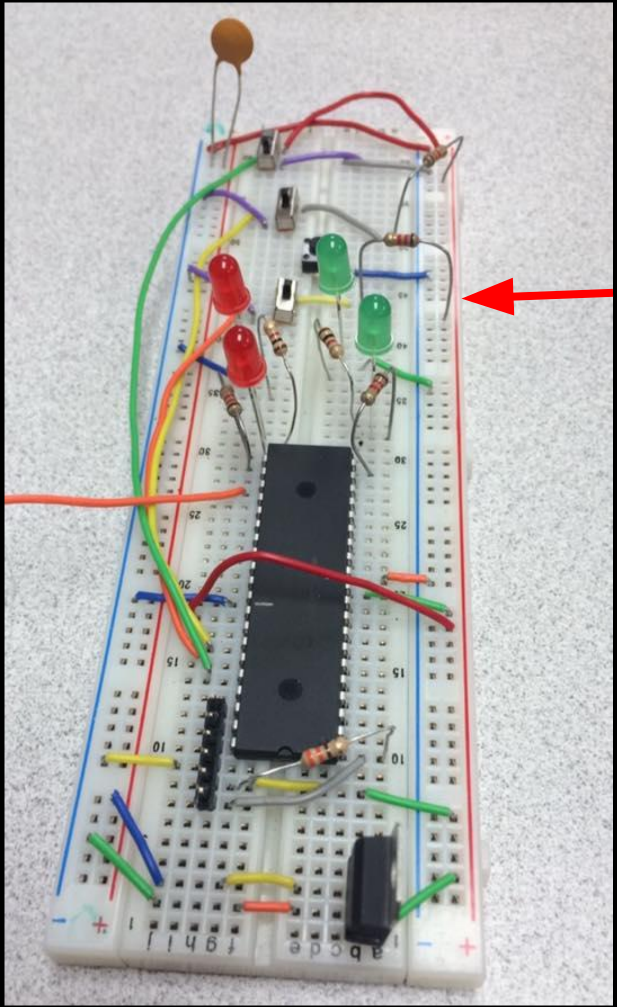
  - Buzzer horn



Joystick was moved to the controller breadboard

Outputs

- Blinkers(RD1 + RD2)
- Headlight(RD3)
- Windshield wiper(RD0)
- Motors(2 H-bridges)
- Buzzer(RC2)



The testing of the LED blinkers

Results

- Diana worked!! :)
- The threshold for the joystick had to be adjusted for every additional component (car accessories, battery pack, divine intervention, etc).
- Having two breadboards and a power source mounted onto the cart adds a tad bit more weight than the cart is capable of holding. This causes the wheels to rub against the sides of cart which affects the steering.
- Having the servo move to two locations with one switch was difficult. We needed to add counters in the code to make the servo move from one position to the next.
- The wipers originally had their own timer interrupt but it altered the blinkers' timing. Both wipers and blinkers were placed within the same interrupt.
- The volume of the buzzer was lower than we expected. We also had to use delays to make the buzzer produce a sound.
- Wiring between the cart and controller is difficult and limits the user. The wires had to be stored so as not to impede the motion of the car. Wireless UART would have been a better option if it was learned earlier in the quarter.

Advice to Future “Generations”

- Get started early!!
- Come up with an exciting but feasible idea
- Be ready to spend the money necessary for your idea or find ways to use household items in your project
- You will be working on your project in multiple locations, so make your project portable
- Look at previous projects and make a better version of a previous idea
- Be ready to make constant adjustments to functionality, in relation to weight and in relation to electric loads
- Better is the enemy of good.

Temperature Controlled Fan

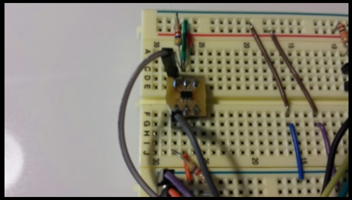
Sam Reinhart

Trey Datweiler

Objectives

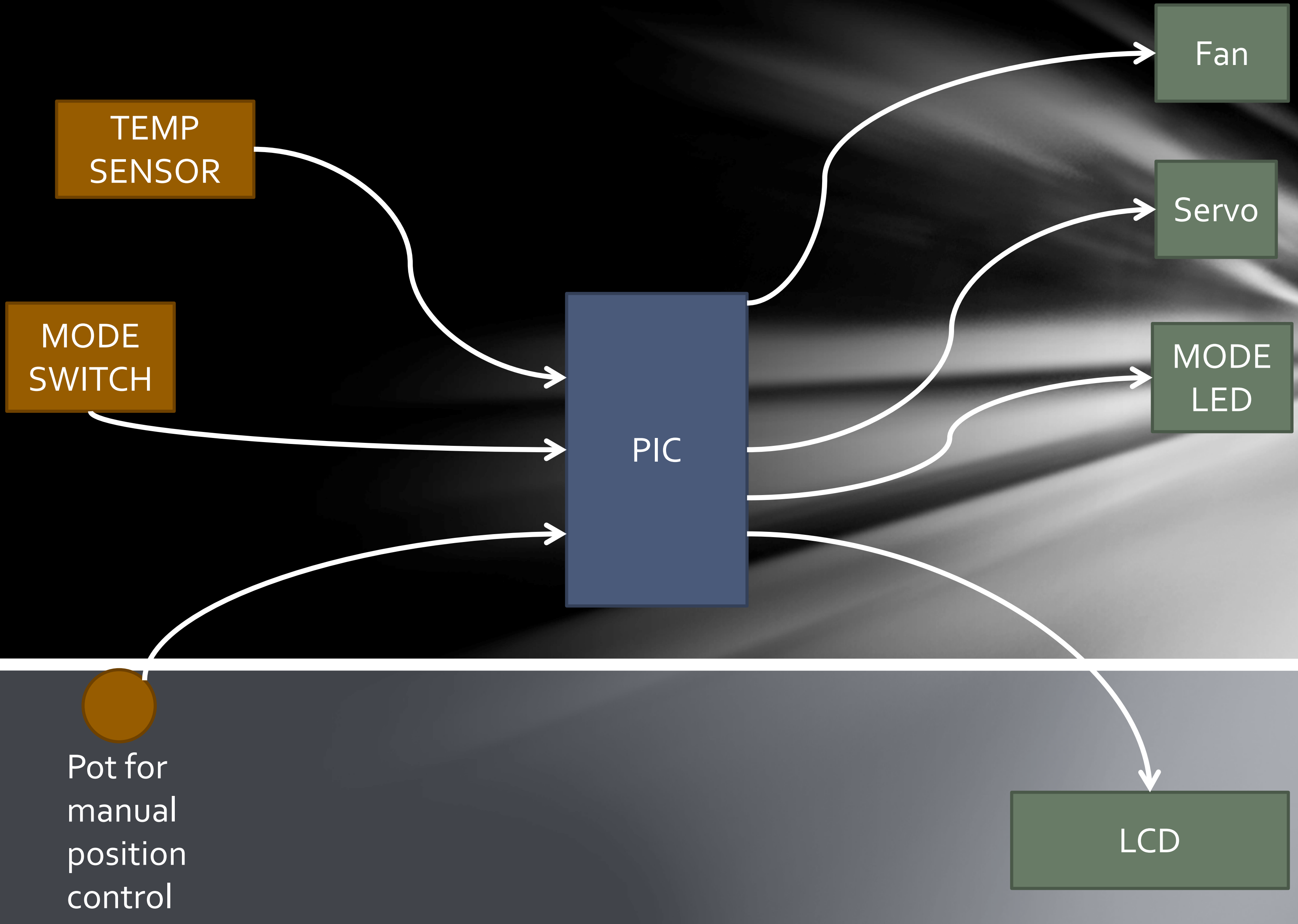
- Temperature controlled fan turns off/on at a set point
- Positioning of the fan is either automatic or manual based on mode selected
- Temperature and set point are output to LCD

INPUTS

Input		*How is it sensed?*
Temperature		Temperature Sensor
Mode Selection Setting		Simple Switch connected to PIC
Manual Position Setting		ADC from Potentiometer

Outputs

Output		*How is it created?*
Mode Light		Digital Output
Fan On/Off		Digital Output->MOSFET
Servo Position		Digital Output Using Delays
LCD		Digital Outputs


TEMP SENSOR
MODE SWITCH
PIC
Fan
Servo
MODE LED
LCD
Pot for manual position control

How was the PIC Used?

1 Analog Input for ADC from Potentiometer

2 Communication Pins (I2C) for Temp Sensor

1 Digital Input (mode switch)

9 Digital Outputs (LCD, LED, Servo)

Results

- Worked perfectly except LCD programs infrequently

Advice

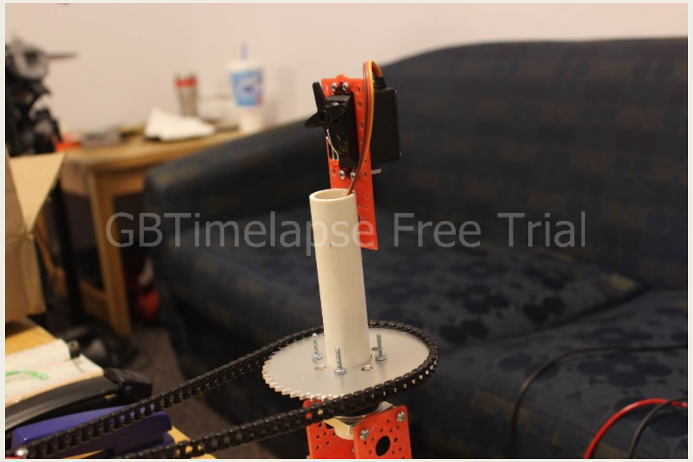
- Avoid using compilers that you are unfamiliar with
- Wire neatly for easier troubleshooting
- Start early

INTERNATIONAL SPACE STATION TRACKER

Jackson Fairfield

Nathan Blank

ISS Tracker



- Calculates the position of several celestial bodies and satellites and tracks the approximate position compared to the current location. The buttons switch and reset the celestial object being tracked and updates are handled on the computer end that then transmits the information over UART.

Inputs

- Computer takes initial heading for target that is calibrated manually.
- PIC takes step counts and servo setting.

Controller

- The PIC is used to control the servo motor and the stepper motor.
- The servo is operated on a low priority interrupt and the heading and stepper motor are updated in a high priority interrupt.

Outputs

- The computer outputs a heading for the steps that the stepper motor needs to be set at and the setting for the servo motor.
- The PIC outputs a signal to the servo in order to operate and set its position. It also operates the Darlington/H-bridge controller for the stepper motor.

Results

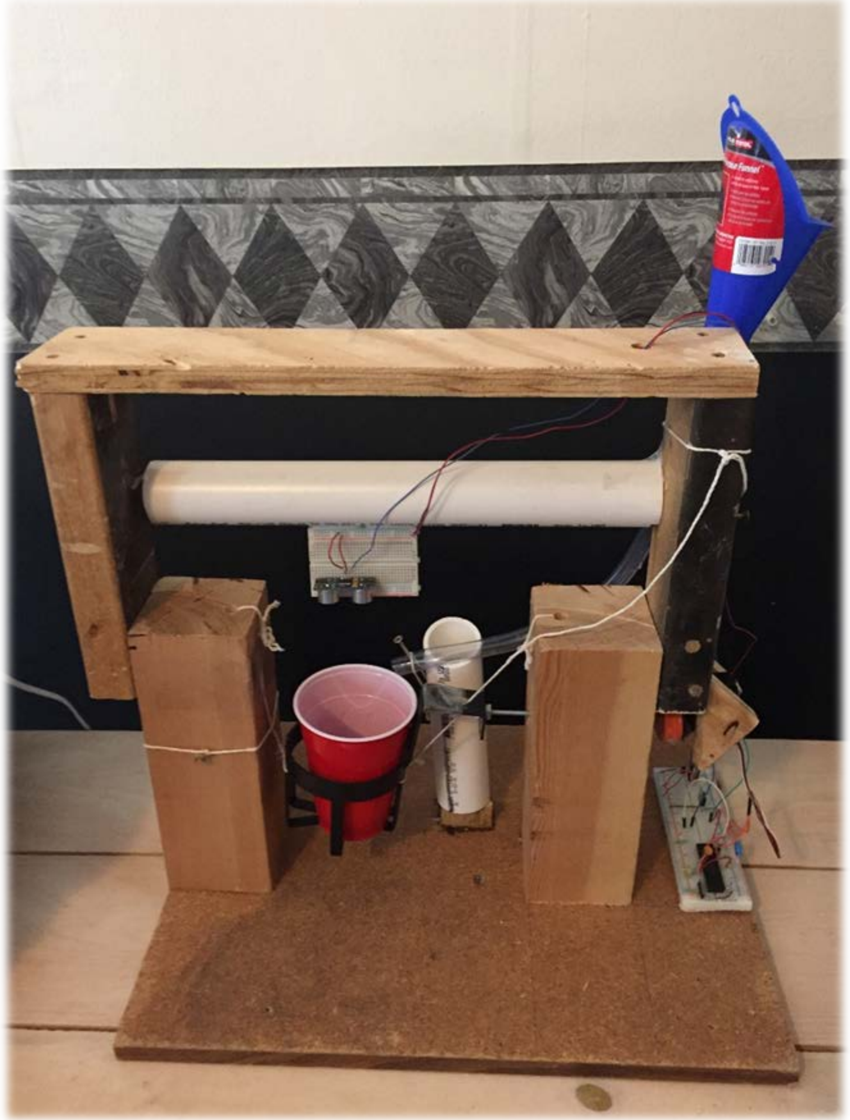
- The tracker worked perfectly! There was a slight tremor to the servo motor due to the lack of threading on a PIC forcing the low interrupt to cancel, but the tracking was relatively accurate and also switched between targets flawlessly.

Advice

- If your project uses UART, do the associated lab early so that you can get started quickly on the project.
- Get parts early so that kinks and snafus can be corrected before they become detrimental.
- Choose something you’re willing to spend money on. It will make it more expensive, but you’ll get to keep it afterwards.
- Have fun!

Tilt

Chris Grable



Craig Larew

Overall Objectives

1. Cup upright & empty

2. Tilt cup ≈ 30 degrees

3. Tilt cup upright when approaching full

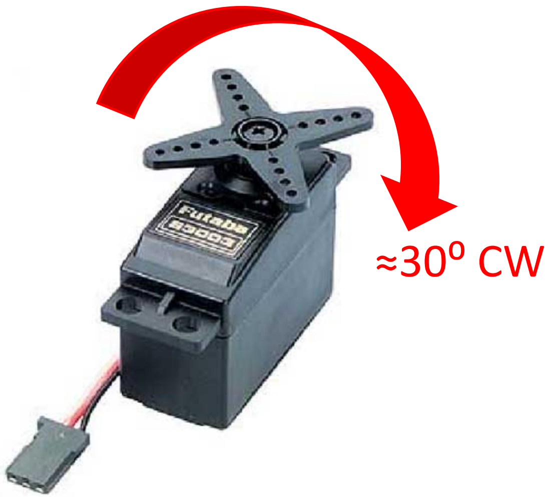
Inputs

Button push to tilt the cup about 30 degrees

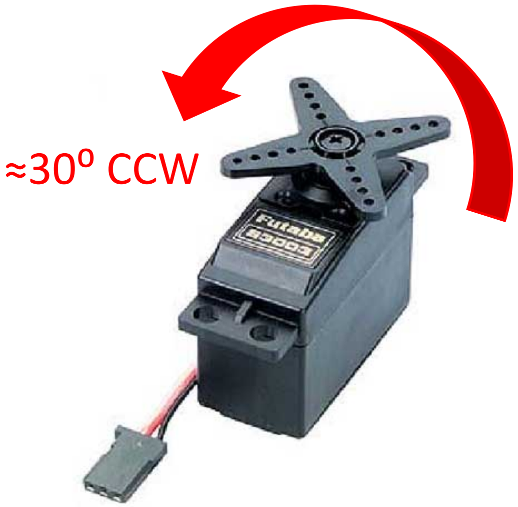
Water Level - Ultrasonic senor relayed distance to the PIC

Outputs

- If RB0 pushed once
- Servo rotates 30 deg
  - Cup rotates 30 deg

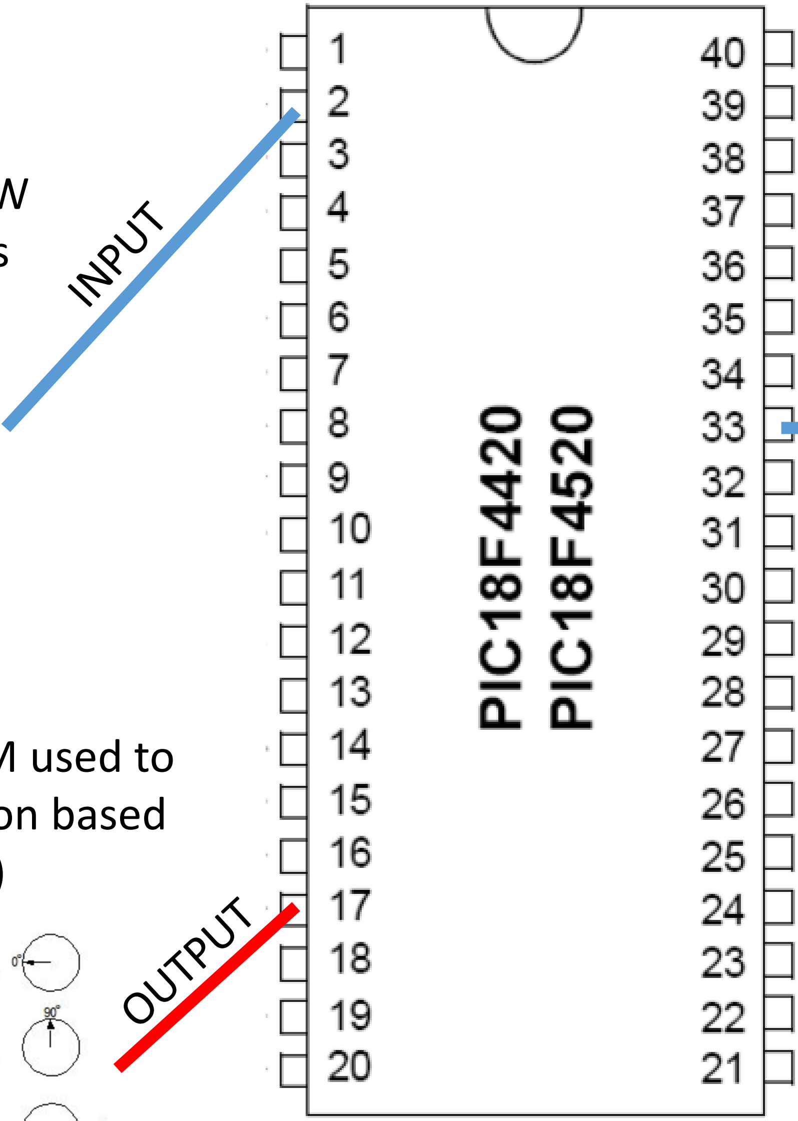


- If distance threshold hit from Ultrasonic Sensor
- Servo rotates CCW 30 deg
  - Cup rotates 30 deg

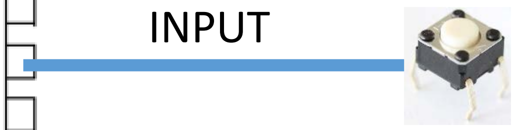


PIC Usage

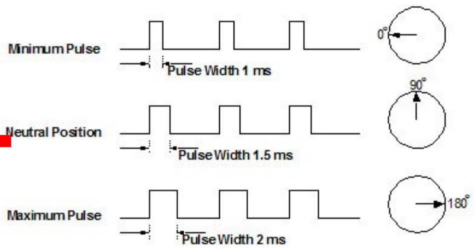
RA0 = Input (water level in cup translated as distance by ultrasonic sensor). Cause motor to rotate CCW from 30 deg to 0 deg if the motor is in the 30 deg state.



RB0 = Input (button press causes motor to rotate 30 deg if the motor is at 0 deg)



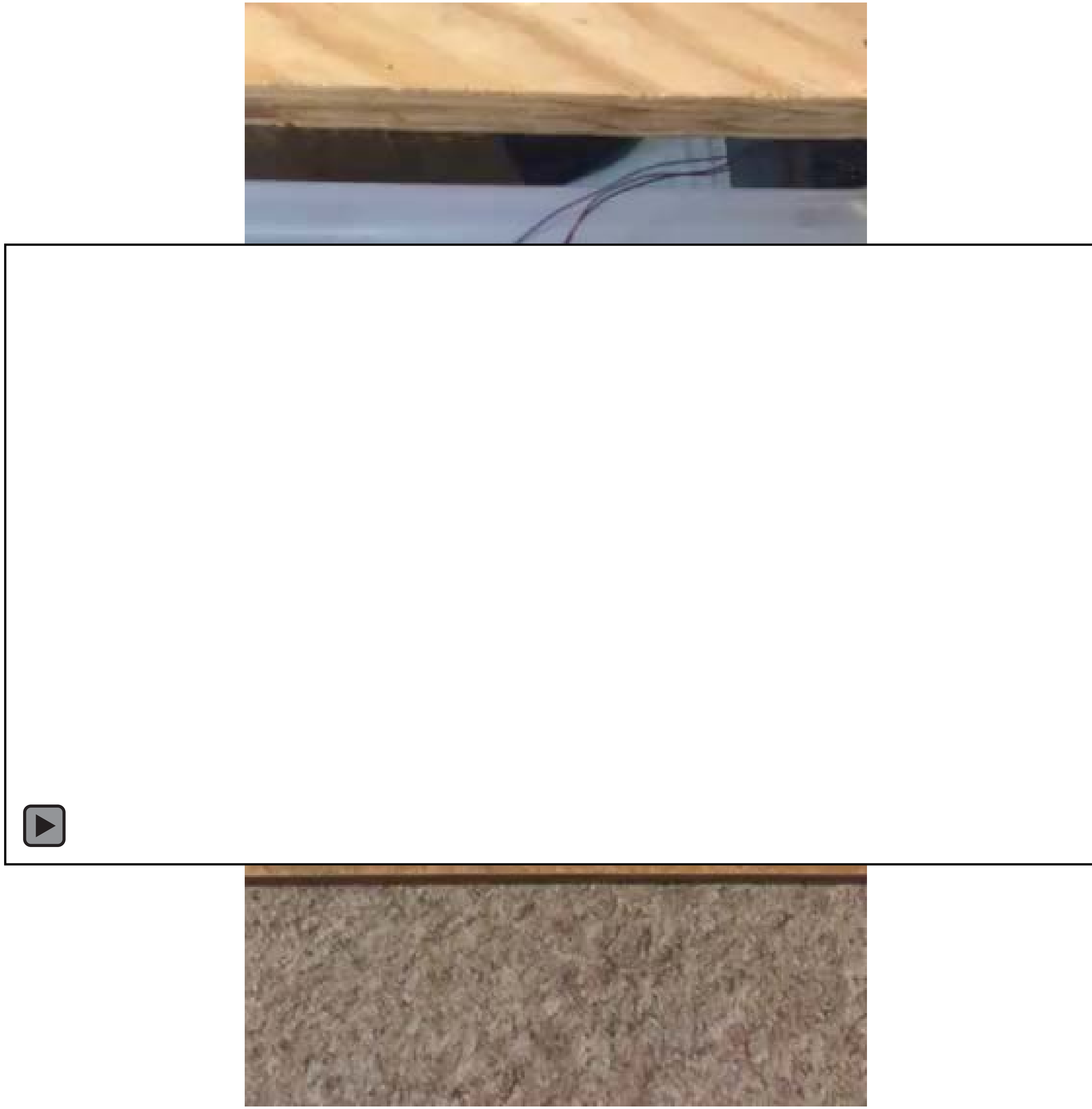
RC2 (CCP1) = Output (PWM used to control servo motor position based on inputs from RA0 & RB0)



*** A voltage regulator was used and all necessary power connections were made

Results

- Worked perfectly on our own (see video → ☺)
- Less desirable results on presentation day
  - Sensor bumped during transportation



Advice to Future Classes

- Ultrasonic Sensors are extremely accurate, and therefore VERY sensitive
  - A millimeter difference in sensor position can give you very different results if sensing a small area
- Get a CAD model put together for your design early in the quarter to help ensure you have all your material needs covered
- Think about little stuff!
  - Power supply, glue, string, long wire, etc.

Coin Counter

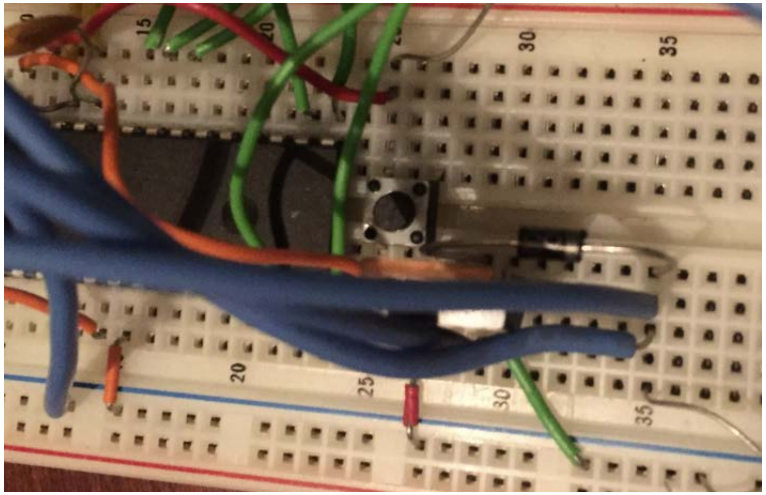
Matt Kelsay
Nate Kinsella

Project Objectives

- The goal of our project was to accurately sort and count coins.

Inputs

- Push Button—turned on/off the DC motor
- 4 IR sensor pairs—one pair for each type of coin

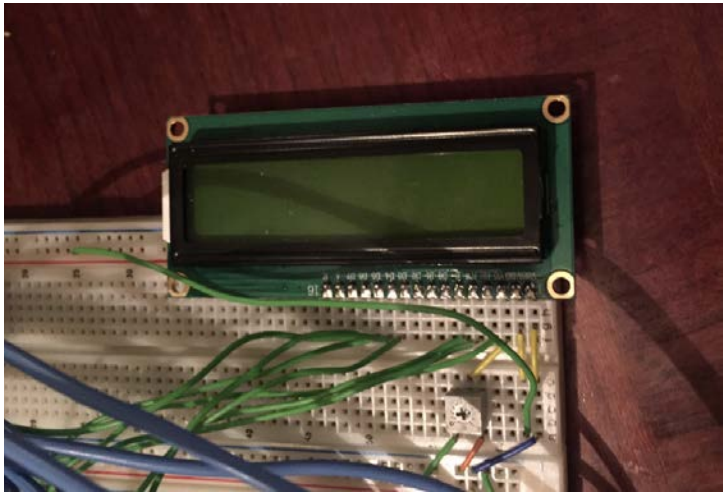


Using the PIC

- Push button interrupt was used to turn the motor on and off.
- Coin totals and total value of the coins were counted using the PIC.
- When a certain number of each coin was counted, the PIC would turn off the motor.

Outputs

- The total value of the coins was continuously updated on an LCD screen.
- When one of the coin containers was full, the LCD would tell the user which one needed emptying.



Project Results

- We were able to get all of the functions of the coin counter to work. The most difficult part of the project was getting the IR sensors to detect when the coins passed through them. On occasion, the sensors would miss a coin, but, overall, the system worked well.

Advice

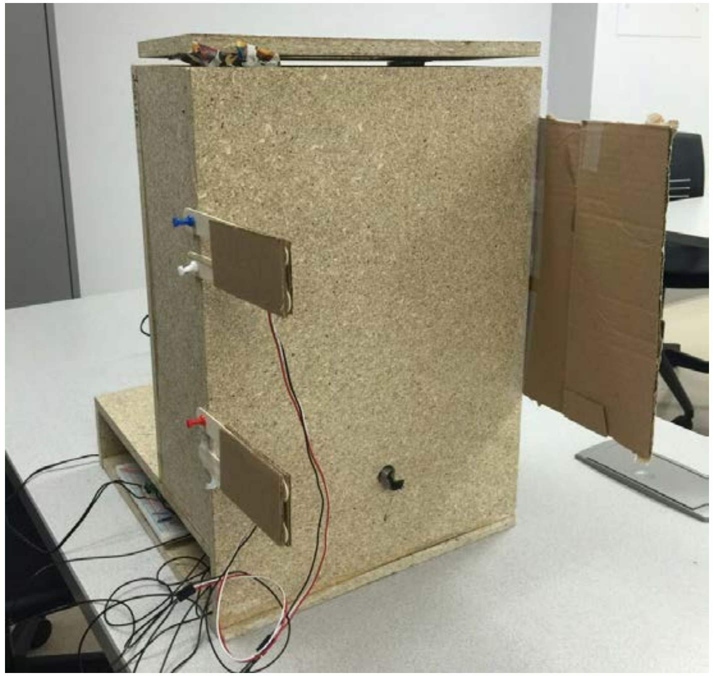
- Start early—leave plenty of time to troubleshoot
- Buy as much supplies from the ECE window as possible. Buying random components on the internet is risky.

Pet Food Feeder

By:
Eric Lai
Zhiyi Yuan
Mingmin Liu

Overall Project objective

- Design a Pet Food Feeder which would dispense food once the pet trigger the sensor



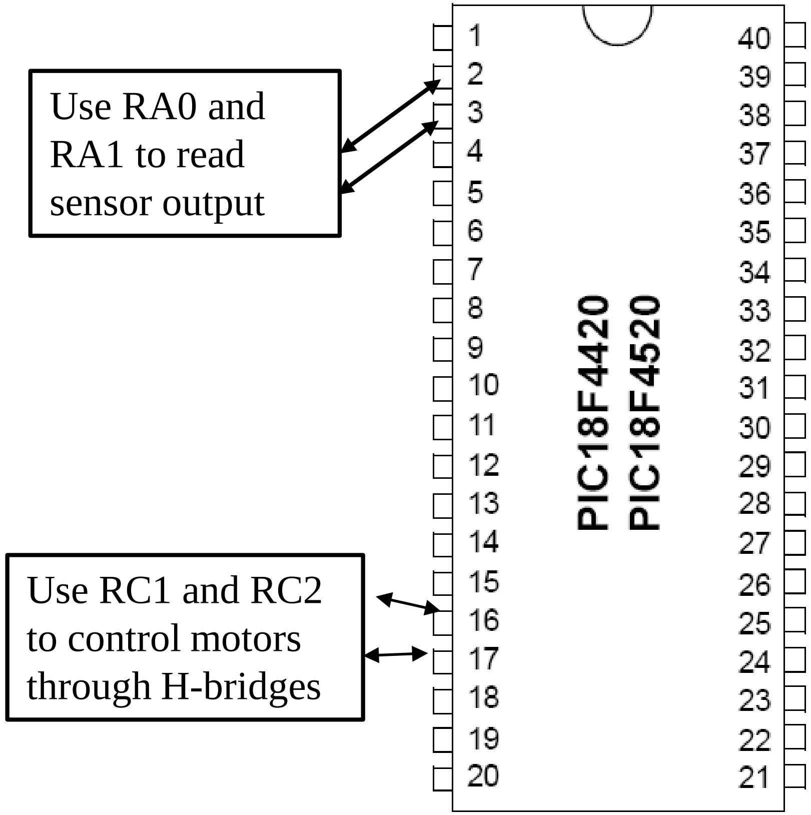
What Inputs? How it is sensed?

Inputs

- ADC reading from light sensor (Sharp GP2Y0A21YK0F IR Range Sensor)
- Reset Push Button
- When the pet comes close to the feeder, the sensor gives a higher reading. If the reset button is pushed, the feeder will dispense different amount of food depending on the size of pet.

How did you use PIC as the controller?

- Delays were used in order to let the motor run.
- Used RA0 and RA1 pins to hook up with the sensors to get reading.
- Although the motor was supposed to run one direction, we used H-Bridge to control the motor.
- Used RA2 to hook up with the push button for reset the feeder.



Use RA0 and RA1 to read sensor output

Use RC1 and RC2 to control motors through H-bridges

What outputs? How it is created?

Outputs

- Rotating Motor
- LCD Display

Created

- Turned auger to dispense food
- Display the amount of food dispensed

Results

- While building the Pet food feeder, we realized that pet food was too big for the auger that we bought. Therefore we had to improvise and convert from pet food to rice.
- The project worked as planned, with the different level sensors activating the motor for different periods of time to dispense various amount of food.
- For future improvements, we should have moved the top sensor further out so that it would be triggered first by a larger pet and dispense more food.

Advice to future “generations”

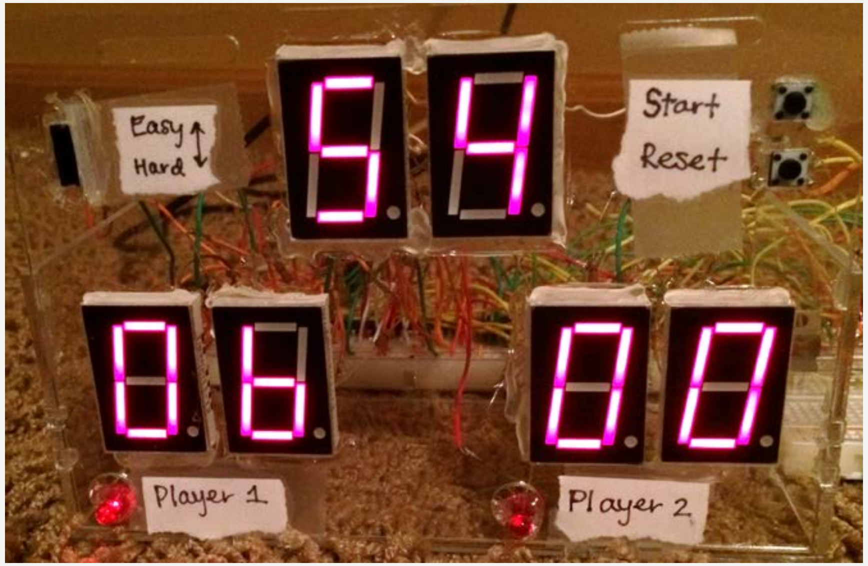
- Begin the project earlier in the quarter to allow for more time to troubleshoot any issue.
- Always keep the project as simple as possible.
- You can always add more stuff after you finish what you have planned out.

BASKETBALL ARCADE GAME

CLAIRE RAYCRAFT AND WHITNEY STRAIN

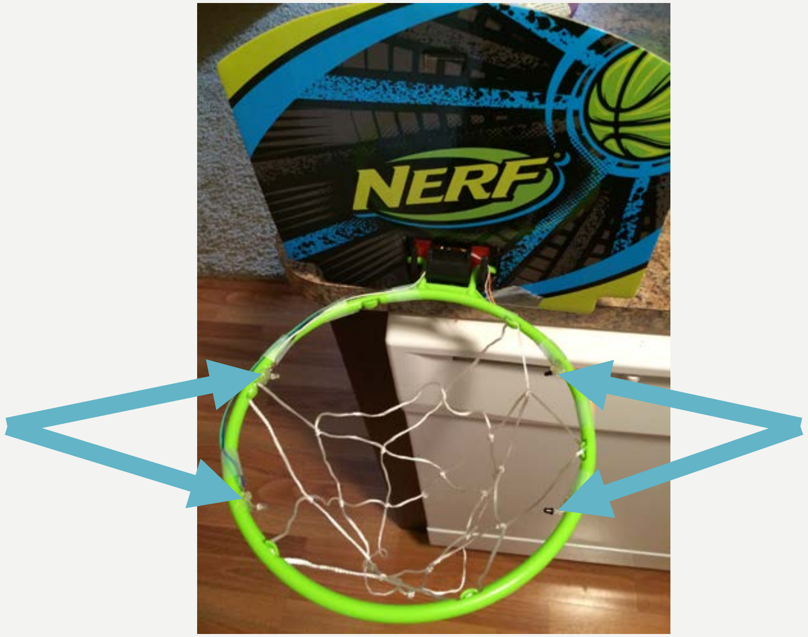
OVERALL PROJECT OBJECTIVE

- Accurately count each made basket and indicate appropriate score for each player
- Display time left in player’s turn
- Make basket move in hard mode
- Indicate winner at the end of game
- Ability to start and reset the game
- 5 second countdown before each round



INPUTS

- number of baskets of player 1- detected with 2 sets of IR emitters and detectors
- number of baskets of player 2- also detected with 2 sets of IR emitters and detectors
- start button
- reset button
- switch position (easy or hard mode)

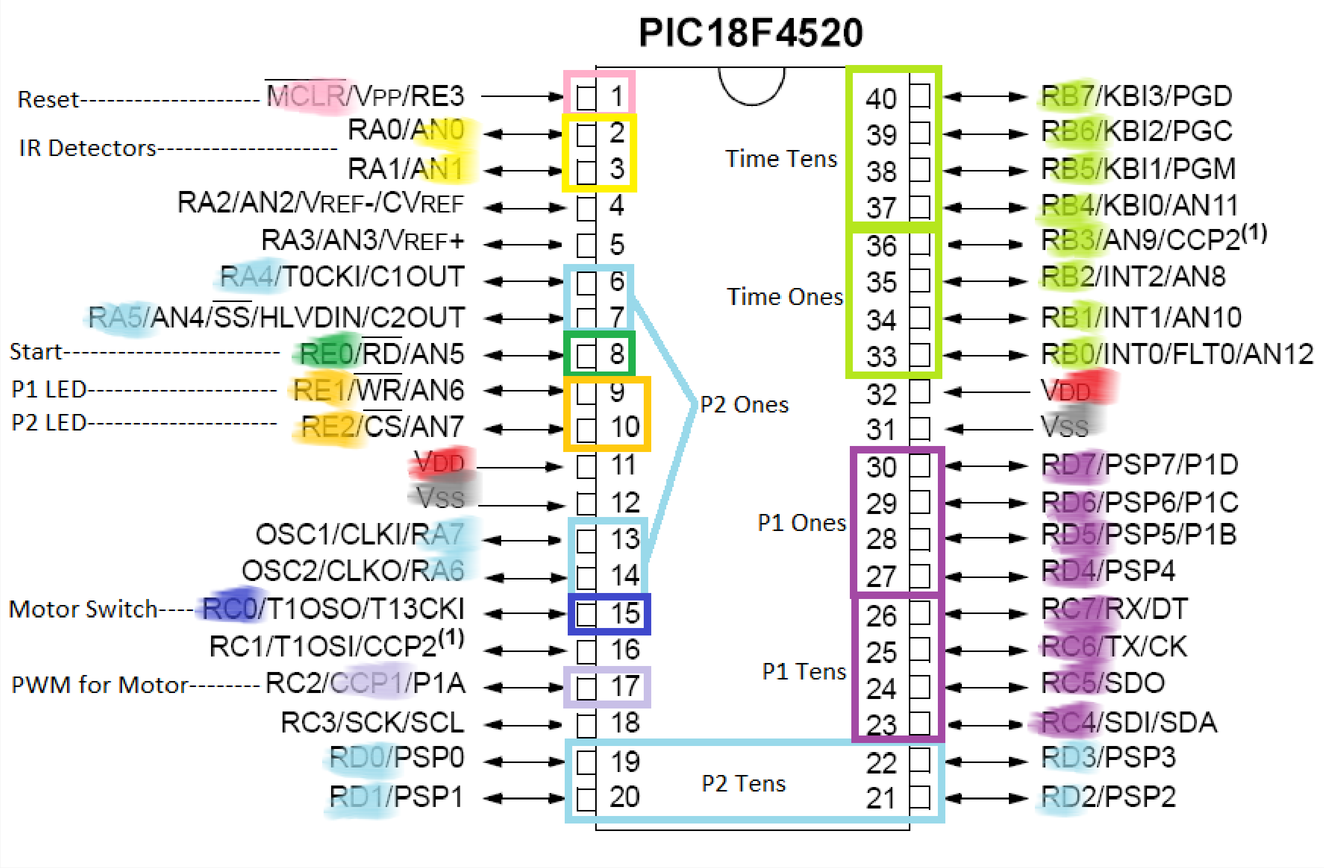


IR emitters

IR detectors

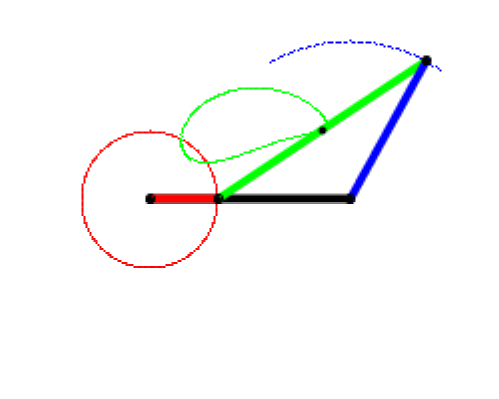
USE OF PIC AS CONTROLLER

- Timers
- Interrupts for timing
- PWM to control the motor speed
- Both analog and digital pins



OUTPUTS

- Timer- shown on two seven segment displays
- player 1 and 2 score- shown on two seven segment displays per player
- player 1 and 2 LEDs (to identify which player is up and winner at end of game)
- Motor to move basket- motor drives 3D printed four bar linkage attached to basket



RESULTS

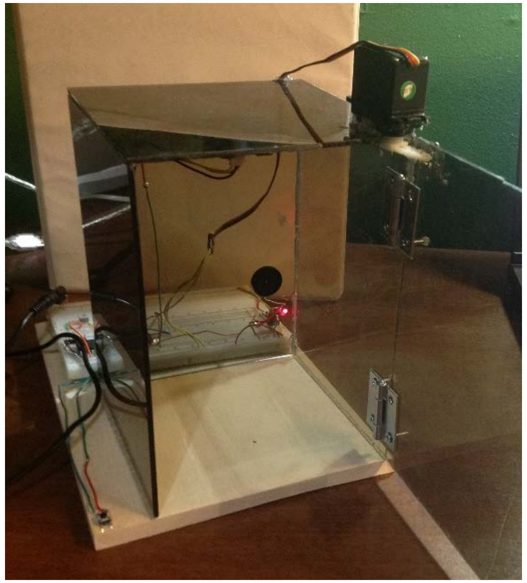
- It works!
  - Accurately counts each made basket and indicates appropriate score for each player
  - Displays time left in player's turn
  - Makes basket move in hard mode and keeps it stationary in easy mode
  - Indicates winner at the end of game by flashing the LED of the player that won
  - Can start and reset the game
  - 5 second countdown before each round
- An IR detector was bent once when hit with the basketball, so more hot glue would have helped. However, the game continued to work properly once the detector was bent back.

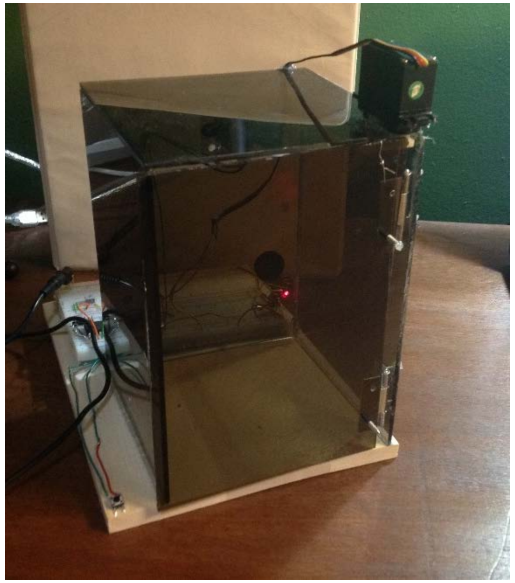
ADVICE

- Test out the motor early in the project, we had to borrow a higher voltage motor and power supply the day before the project demonstration.
- Don’t use seven segments unless you want to solder for hours. Additionally, using the driver chips help, but they still take up lots of places on the PIC. We had to creatively use bit shifts and clearing the bits to make the counting work properly.
- Do get parts either laser cut out of acrylic or 3D printed. It worked well for us.
- If you need help, ask!

Self-Closing “Refrigerator”

Tucker Nelson and Alex Tester





The objective of the project was to make a smart refrigerator door that…

- could automatically close

- could sense when somebody was in front of it and stay open

- could teach users to be better about closing the door

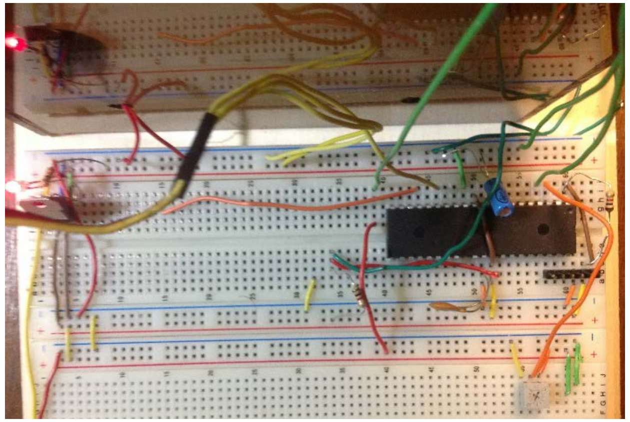
- could be functioned by a responsible user normally

Three primary inputs were used in the system

<table>
<tr>
<td>Button Switch
• Digital input</td>
<td rowspan="2">Infrared Proximity Sensor
• SparkFun SEN-08958
• 150 cm range
• Analog input
• Reads distance object in path is from sensor
 </td>
</tr>
<tr>
<td>Potentiometer
• Analog input
• Sensor sensitivity</td>
</tr>
</table>

PIC Programming Control

- No code in main while loop- all control by interrupts
- Low priority interrupt looks for button control, which opens and closes servo motor. Also reads analog values of both potentiometer and proximity sensor. Flagged at 50 Hz.
- High priority interrupt tracks how long proximity sensor has read below the potentiometer threshold, and controls buzzer and overriding door close. Flagged at 1 Hz.
- PWM is used to control the buzzer output



The system had two outputs

Buzzer	Standard Servo Motor
• PWM Output • Sensor controlled • Arbitrary value used for tone	• Hitec HS-322HD • Button controlled • Sensor controlled • Position controlled by delays

Did it work as planned? YES!

Every scenario worked as expected

- Door could be opened and closed by button
- Door could be opened by button, turn on buzzer, closed automatically
- Door could be opened by button, turn on buzzer, closed by button
- Door could be held open by obstructing proximity sensor, resulting in an analog value higher than potentiometer threshold
- Potentiometer could adjust the sensitivity of proximity sensor

Only issue – At "closed stated" door was ajar by about 10°

Advice to Future Generations

- This project was simple- and that is okay. Do not overcomplicate.
- Align axes of rotation between rotating parts and motors (reason behind door issue)
- Prototype the code on greenboard if possible. We did this, and were confident we had a working code before building anything requiring only mechanical / electrical troubleshooting
- Hot glue is a necessity for building
- Inherit parts if possible- we were lucky to find most components from people who took the class previously