Black Jack Card Dealer

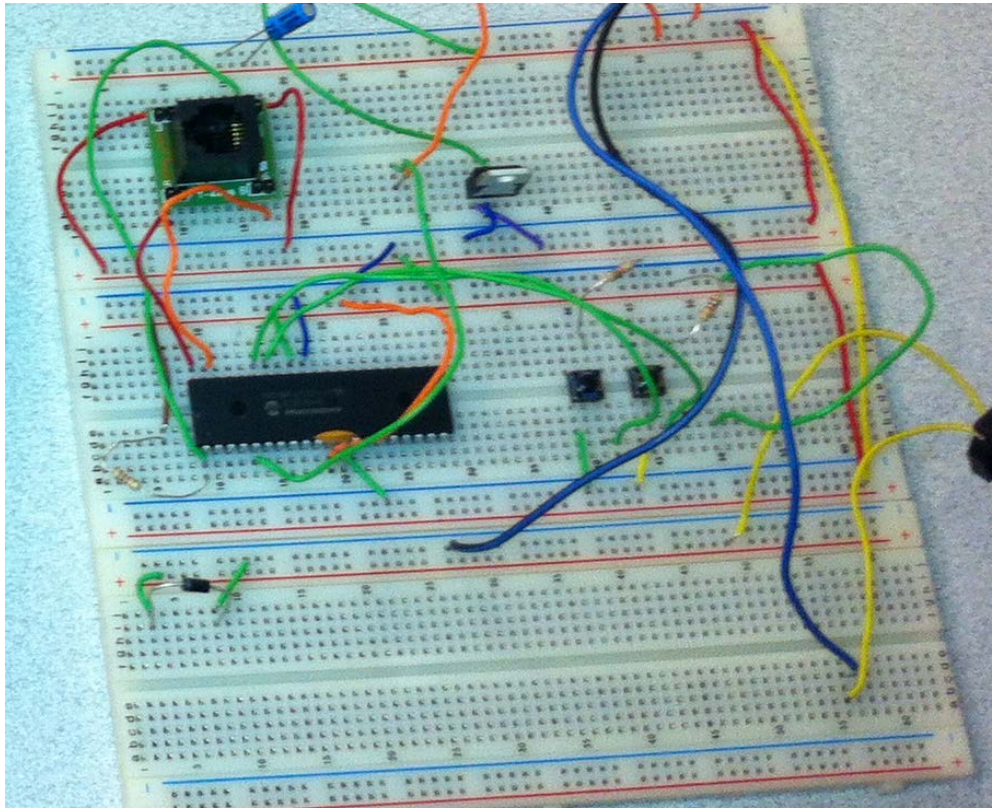
Cody Cheesman
Devin Wentz

Project Objective

- Input number of players via button press
- Activation button would start the game
- Motor attached to Dealer Shoe would deal one card to each player.
- Dealer Shoe would be sitting on a platform attached to a servo to assist in rotating from player to player.

Project Inputs

- Buttons to signal which players are present
- Activation button to start game



Using the PIC as the controller

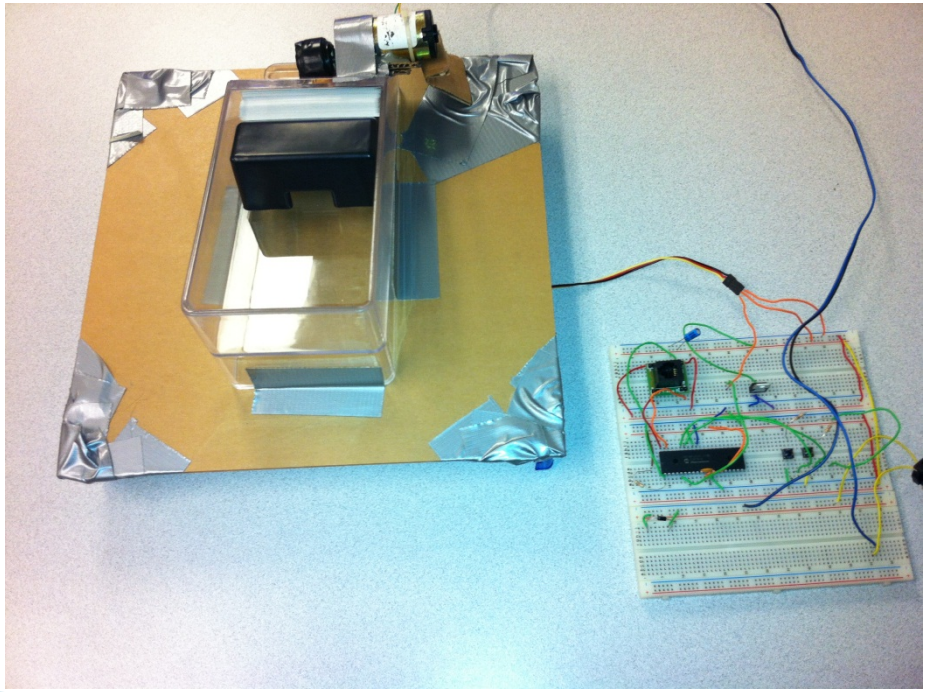
- Once a player had hit a button, a counter would be stored using interrupts.
- After the activation button was pressed, if statements within the main while loop would determine which players are present and deal to them.
- Once the servo had turned to a player, the motor on the dealer shoe would turn using PWM.

Project Outputs

- Servo would turn to certain players depending on if someone had hit the button in each location.
- Motor would turn to deal a card after activation switch had been pressed.

Project Results

- Activation button was more difficult than we hoped.
- Servo/Motor signals being sent through PIC for 3 different players at once caused the servo and motor to overload with data thus, not work properly.
- Code became more simplistic with buttons to turn servo to each player location opposed to previously planned automation.



Future Advice

- Plan for adequate time to spend working on this project.
- Even the best teammates can't account for the hiccups and issues that can arise.
- Don't rely on the parts that are purchased, they don't always work as planned and sometimes are not the correct part that you need.

Couch Master Deluxe LTE 3000akb

Andrew Bowers
Kayla Edwards
Bobby Newman

Project Objectives

- Automatically turn TV on when someone sits down
- Automatically turn TV off when someone is not sitting on the chair for a pre-determined amount of time
- Change channels using tactile switches

Inputs

- The weight of a person sitting on a chair, using a force sensitive resistor.
- Analog values of the vertical and horizontal potentiometers on the joystick.

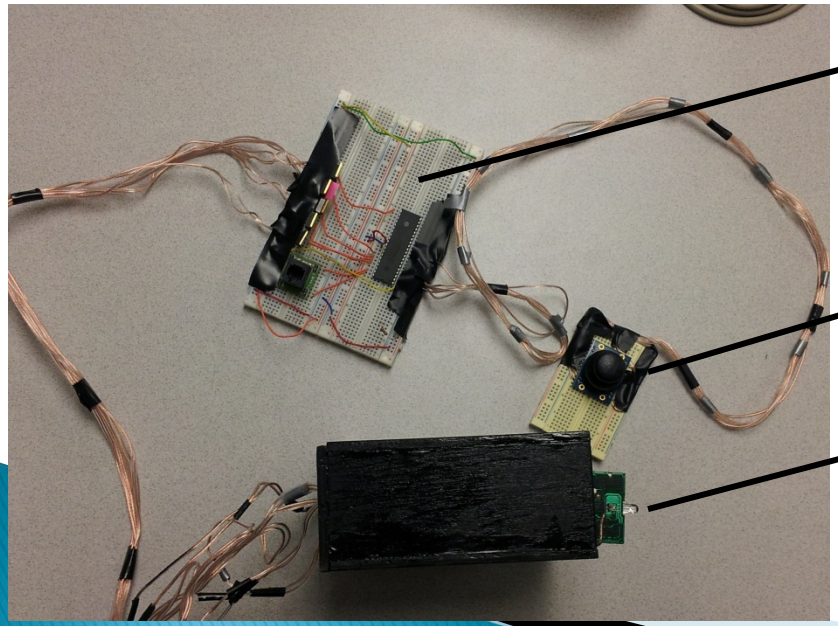
Using the PIC

- We took analog readings of both potentiometers on the joystick. Based on these values a specific MOSFET was turned on or off.

- We used timers to control the amount of time before the television would turn off after someone stood up.

Outputs

- Signals to the MOSFETs to turn the remote buttons on
- TV responds by turning on or off, changing channels up or down, or increasing or decreasing volume



Breadboard of MOSFETs and PIC

Joystick to control volume and channels

Black box concealing remotes

Results

- Everything worked as planned. We were also able to incorporate a volume up and volume down command, which was not in our original goals. The only thing that went wrong was our puck. We were unable to disconnect the puck from our breadboard and properly run the program. We believe it is due to a grounding problem that we were unable to fix because of the way the remote controllers were made. When the remotes were grounded together, random buttons were pressed instead of the ones we desired.

Advice to future classes

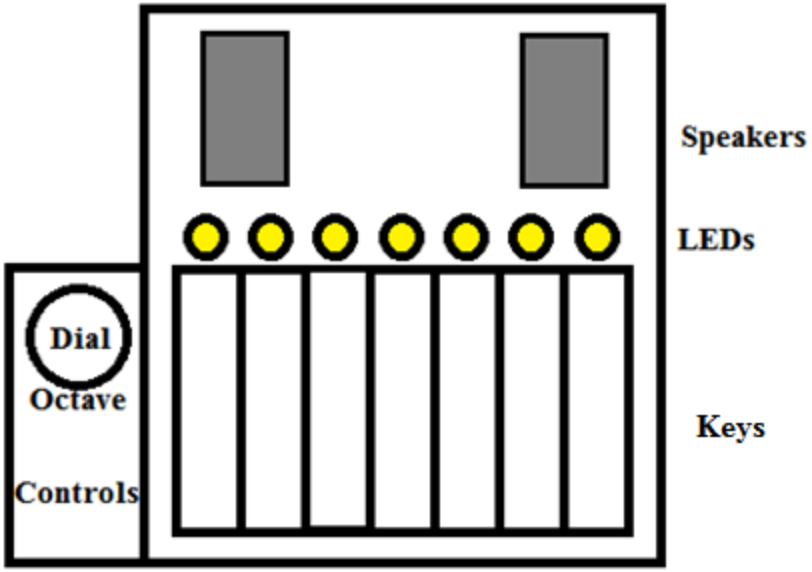
- Start your project early: don't leave it all for the last week or even two weeks
- Use electrical tape when you solder two wires together, it will help if you don't have a perfect connection
- Use all your resources: the professor's are extremely helpful

Electronic Piano

ME430 Mechatronics
Fall 2012

Ryan Oliver
Chris Nelson

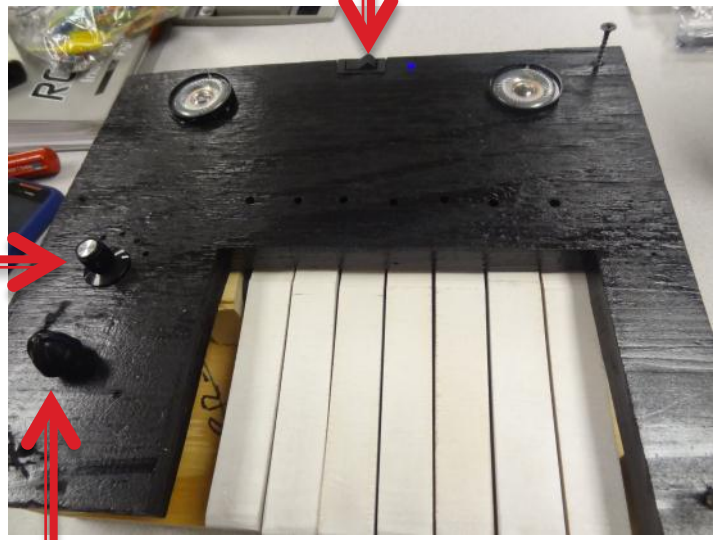
Objective

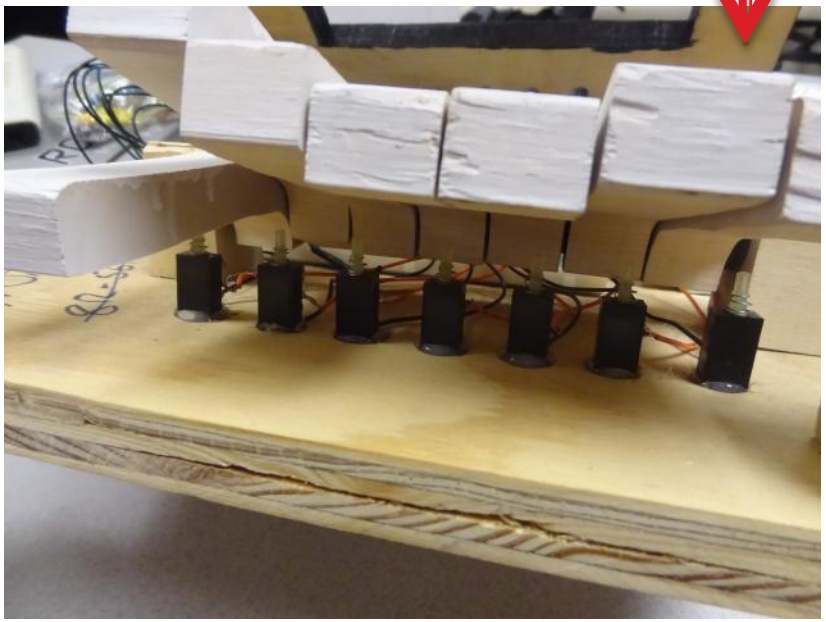


- A simple piano:
  - Easily used
  - Volume control
  - 5 Octave control
  - Can play 1 note at a time
  - On/off switch to save power
  - Battery powered (can use on the go)
  - PWM controlled speaker emits correct sound
  - Play note for length of time that key is pressed

Inputs

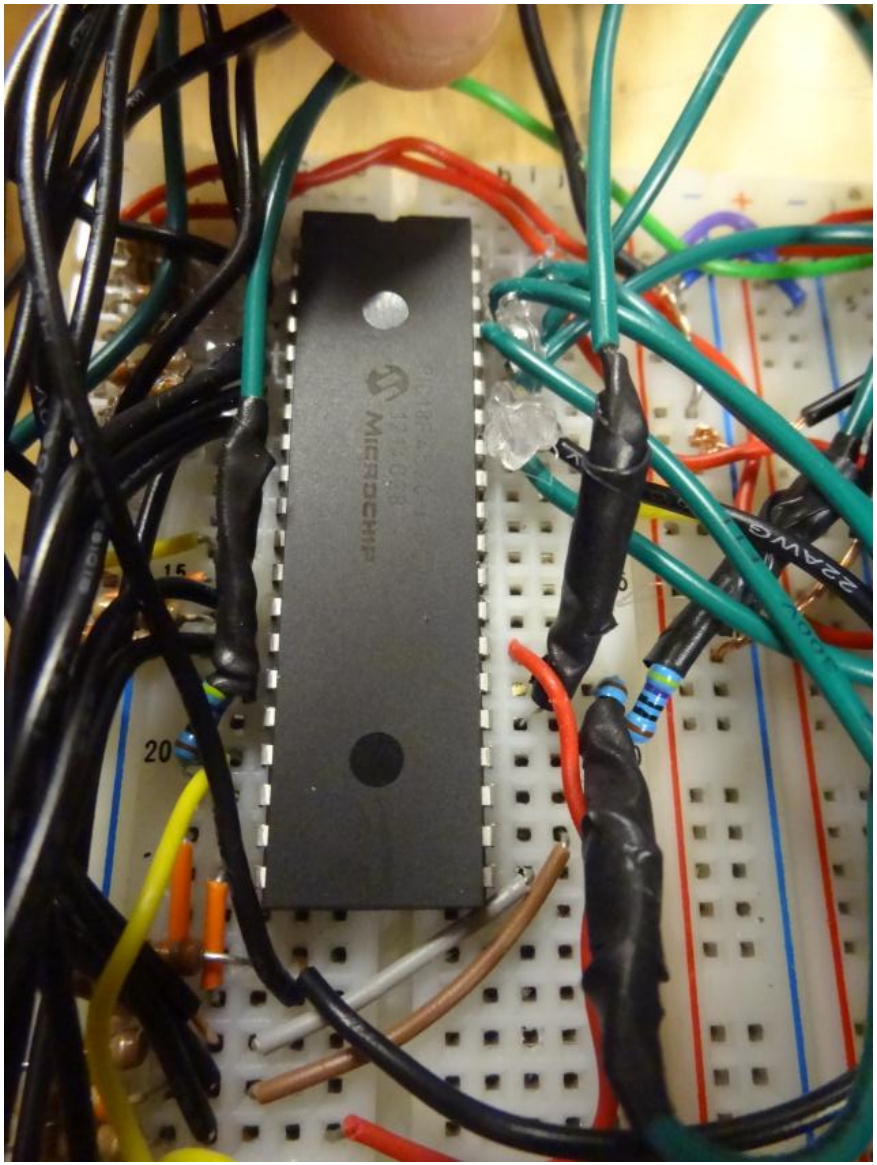
- Keys with spring-based touch sensors
- Multi-position switch to control octaves
- Volume potentiometer
- Power switch





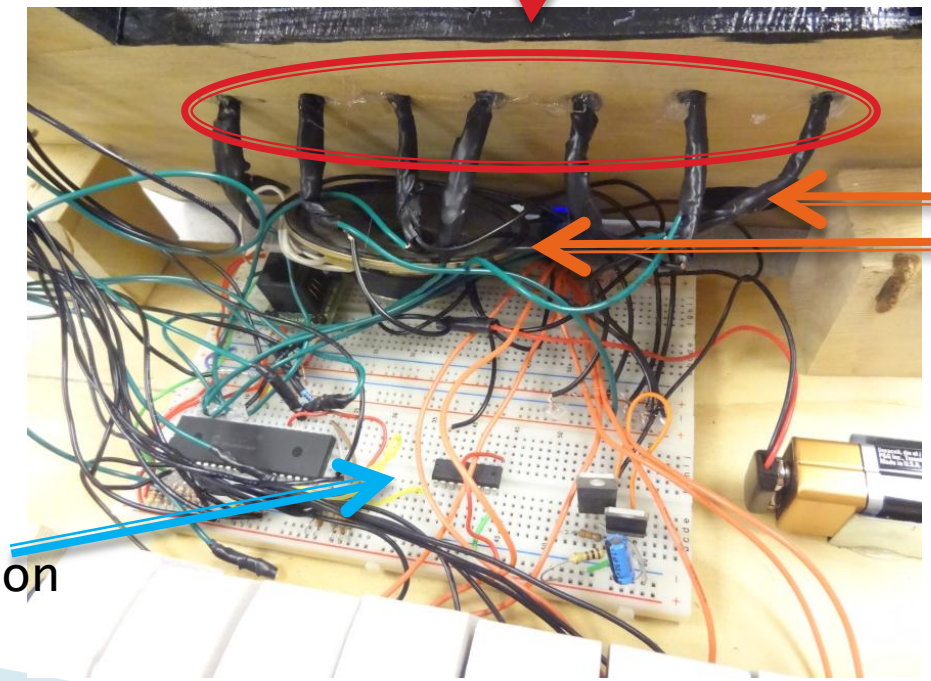
PIC Microcontroller

- RA0
  - analog input for volume dial
- RA1 – RA7
  - digital inputs for key touch sensors
- RB0 – RB6
  - digital output for key LEDs
- RC1 – RC2
  - PWM analog output to speakers (Darlington)
- RC3 – RC7
  - digital outputs for octave LEDs
- RD0 – RD4
  - digital inputs for octaves
- No interrupts and only PWM timer was needed



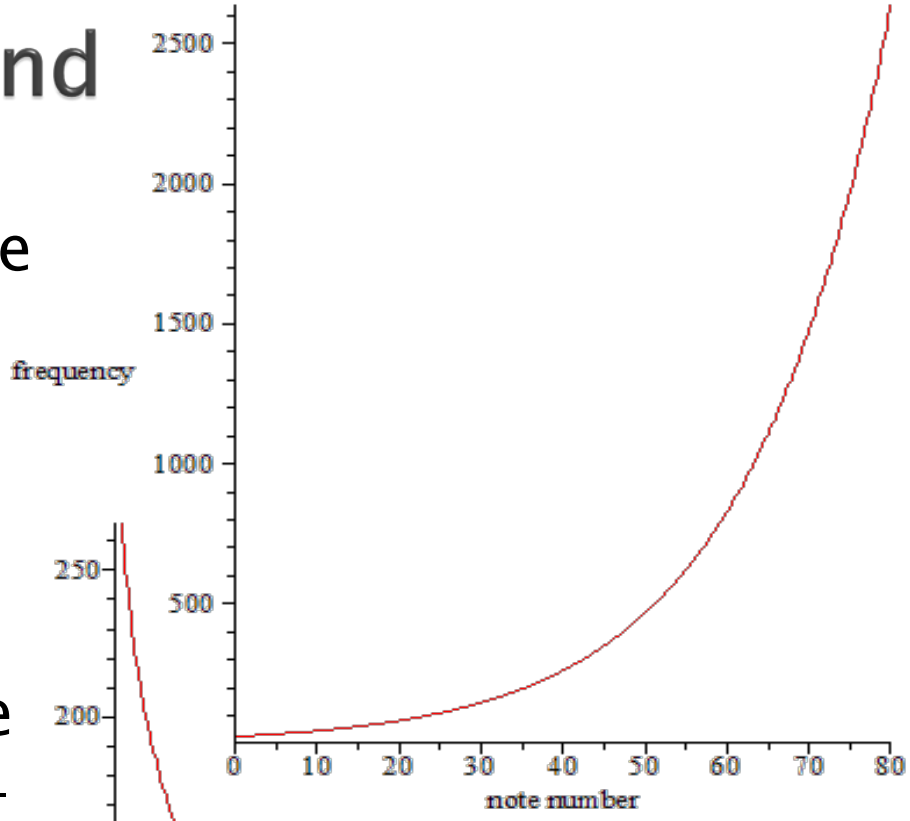
Outputs

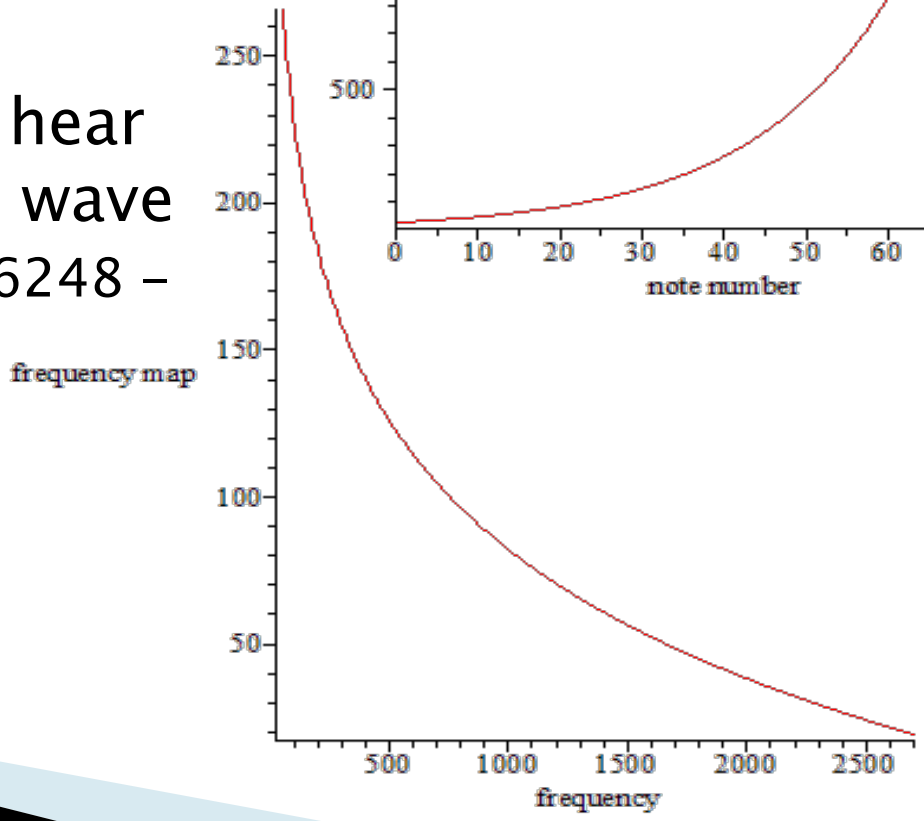
- Generated sound with speakers using PWM
  - Frequency based on octave and key number
  - Pulse width based on volume potentiometer
- Lights LEDS
  - Based on octave position
  - Based on key presses
  - Power On/Off



Math Behind the Sound

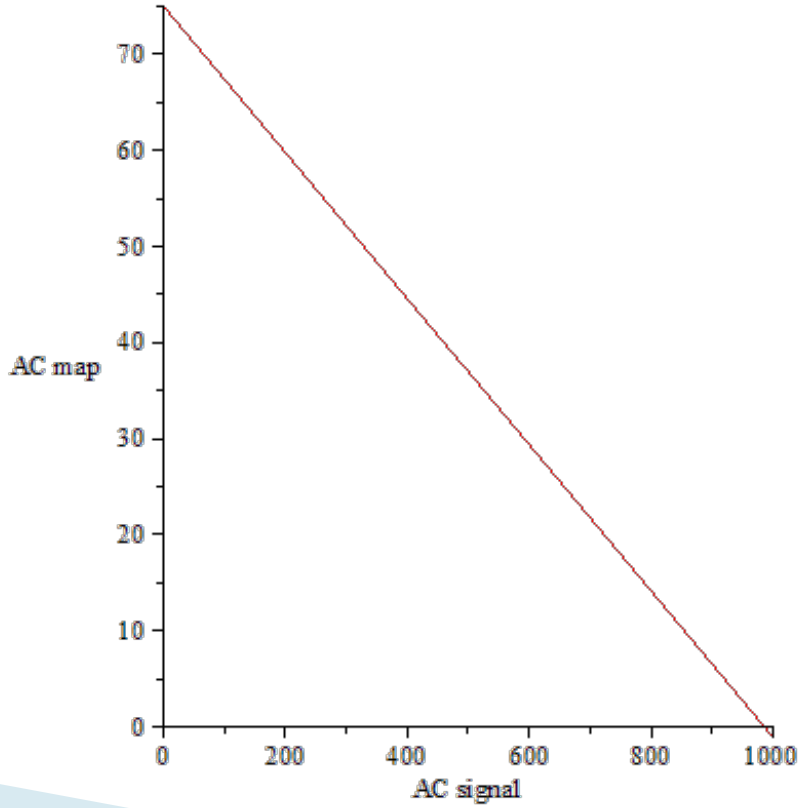
- Calculated real piano note value & found frequency
  - Freq = pow(2.0, ((note-49.0)/12.0))*440.0
- Scaled frequency to hear difference in square wave
  - FreqMap = (int) (519.6248 - 63.334*log(Freq)))





Math Behind the Sound

- Scaled volume – changing duty cycle on PIC also changes frequency (out of playable range)
  - Volume = (int) (75.0 – (25.95/341.0)*volumeOrig));



Results

- The Electronic Piano functioned as planned
- PWM wasn't not very consistent
  - Square wave with speakers made annoying sounds
- Easily user controlled with a 5 octave range

Advice

- PWM isn't very consistent
  - Duty cycle doesn't stay constant
  - Changing duty cycle can change frequency
  - Making the clock slower to have a large range of PWM frequencies can cause noticeable delays in outputs
- Plan ahead
  - Don't make the project too complex
  - Split up project into smaller parts to make milestones more achievable
  - Test code and hardware in parallel rather than doing all of one first
  - Buy parts well in advance to make sure everything works

Proof Mixology

ME 430
Final Project

Abram Summerfield
Brian Cherbak

Objective

Automate mixing of drinks

Minimum Viable Product:

Two bottles with two preprogramed drink combinations. Drinks will be make by using programed push buttons.

Inputs

Momentary push buttons

Toggle switch (on/off)

Outputs

Solenoid valves

The Program

We programmed digital pushbutton inputs on RA.0 - RA.2 and digital outputs on RB.0 - RB.2.

Three functions controlled how long each valve would be open using timer delays. The valves were connected to the H-bridge to get 500 mA of current.

Each function was called by each input (RA.0 – RA.2).

Results

The project worked out really well.

We exceeded the minimum viable product by having three drinks with three bottles.

We also used perfboard to create a more permanent and finished product.

Our Advice

The class allows you to choose the difficulty of the project – choose something you'll be proud of and use when you're done. Cheers!

Start early on programming because it can take a long time to debug.

LASER LID

Garrett Broadnax

Katy Dimon

OBJECTIVE & OVERVIEW

Ever tried to open the trash with hands full of eggshell?

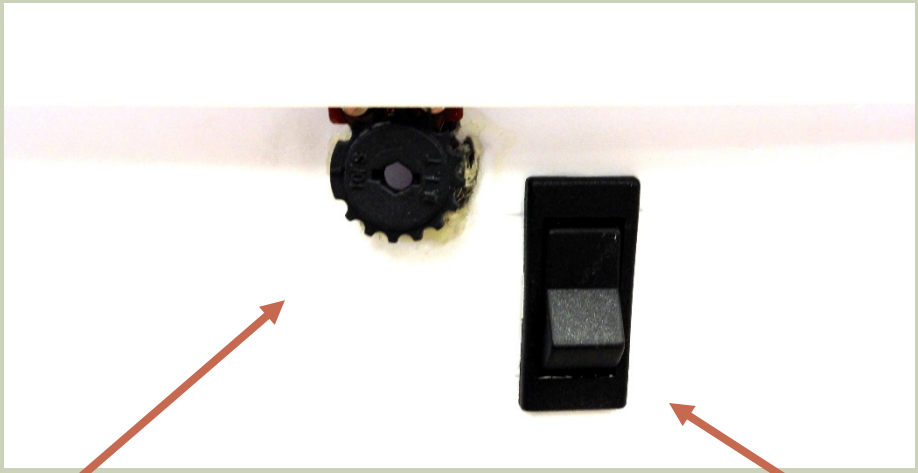
With a garbage lid activated by your foot, you'll never need to worry about the trash being inaccessible again. Laser Lid is a trash can that opens/closes with the presence/absence of your foot in a laser break-beam sensor. Featuring an on/off switch and a light adjustment sensor for day or night situations, your trash is accessible whenever you need it and safe from unwanted guests like small children or pets.

INPUTS & SENSORS

The Break-Beam Sensor

is a Cadmium Sulfide piezo-electric cell whose resistance varies linearly with light level. To ensure that the sensing of light was not impeded by common changes in the room, we chose a laser as the source for its light intensity.

INPUTS & SENSORS



The Light Adjustment Wheel varies the sensitivity of the CdS cell, giving the can the freedom to work in any environment, night or day.

The On/Off Switch makes it simple to disengage the break-beam sensor, turning it back into a manual trashcan, and saving energy.

PIC

The operation of the trash can was controlled by timers within an interrupt.

- The system continuously checked that the power was on and compared the sensor level to the light level before enabling stepper motor operation.
- Stepper motor operation was controlled by case states and the speed of the motor was set by writing values to the timer.
- ADC was used to convert the analog readings from the CdS sensor and the potentiometer's reading of the light level to digital values.

OUTPUTS

Shying away from readouts and lights that would only serve to indicate what you're already seeing, our goal was simple:

When the light beam is broken, the lid raises; when not, it remains in stand-by, after calmly returning to the closed state.

*EXTRAS

Our goal was to make the lid raise and lower without effort.

We added features as if we were taking the product to market. The on/off switch and light adjustment sensor were in an effort to alleviate eventual customer issues before they arose.

The next steps before production would be to

- PCB mount everything
- Make the laser run off the board
- Have it run on battery

RESULTS

Despite efficient programming, clean wiring, quality mentoring, and our best efforts, our results were inconsistent.

Sometimes our program ran the system exactly as planned, sometimes the lid responded after a varying amount of time, sometimes the motor changed speeds, sometimes nothing moved. MPLab was extremely buggy for us, sporting around a 10% success rate for programming (without any changes); also, our puck opened (split in half) without any effort.

However, when it decided to work, the lid responded very well; all sensors did exactly what they were supposed to and the system showed its high level of robustness.

5 PIECES OF ADVICE

1. Pick a feasible project; add features if it seems too simple
2. Start building around week 5
3. If you insist on doing a mechanically intensive project, get motors that are rated much higher than your predicted needs
4. Consult with class members & mentors early, we recommend the EE Shop if it's available to you
5. Do a project you'll want to use afterwards

Light Undimmer

A Practical Use of the Most Useless Machine

Alex Andrews
Chris Hopwood

Objective

- The lights in the new apartments automatically turn off
- Students often have to get down from their decks or desks to reactivate the lights
- Our goal was to reactivate the lights automatically
- Have a clock on the device, and be able to set "dark hours" during which the lights which the device will not turn back on lights

Input and Output

- Inputs
  - CDS cell to detect light
  - Potentiometer to adjust light sensitivity
  - 5 push buttons for menu navigation
  - Switch for enabling/disabling ligh reactivation
  - Power switch
- Outputs
  - LCD display
  - "light detected" led indicator
  - "light reactivation enabled" led indicator
  - Power led indicator
  - Servo with a flag attached for light reactivation

PIC Features Used

- Timers and Interrupts
  - Updates the clock
  - Handles the PWM signal for the servo
- PWM
  - PWM was not used for the servos as it is not accurate enough.

Results

- The device can correctly detect when the lights turn off
- The clock was fairly accurate, losing only a few seconds a day
- All user interface elements worked as planned including
  - Setting dark hours
  - Enabling/disabling dark hours
  - Setting the time

Advice

- Start early so you can:
  - Have a project box laser cut, not built by hand
  - Actually enjoy the project and not rush it together
  - Not have it held together by combination of epoxy, wood glue, gorilla glue, duct tape, staples, strings, thread, and nails
- Get your project box laser cut!
  - Talk to Ron
  - Start communication with him early
- Don't clear the LCD screen, just write text on top of the old text

Pinball Machine

Franklin Totten
Kevin Dickson

Objectives

- Create playable pinball game
- Base Score on time and hitting objectives
- Automatically sense balls launched and lost
- Use mechanical launching and flipper systems
- Display score one 4-digit LED

Inputs

- IR Sensors:
  - 1 Launching: senses when ball is launched
  - 1 Lost: senses when ball passes between flippers
- Pressure:
  - 2 Objectives: sense when ball passes over objective point and increases score
- Power Switch:
  - Hardwired to cut power; serves as reset

Outputs

- 4-Digit LED
  - Uses 4 output pins to vary digits
    - Digits display 1 at a time quickly
  - Uses 8 output pins for A-G and decimal
  - Displays score live for the player

PIC Controller Uses

- Timers
  - Used to switch between LED digits and increase score
  - Used with a simple if statement, interrupts were not necessary
- On/Off Switch
  - Cuts Power to the PIC
  - Results in a manual reset by restarting program

Results

- Made to be expanded upon beyond scope of class
  - As a result, it needs some finishing still
- All guaranteed aspects were completed, so project was a success
- Mechanical parts proved trickier than anticipated
  - ADVICE: make mechanical parts quickly so they do not hold up implementation.

ROBO-COP

Andrew Gorman

Nick Wanstrath

OBJECTIVE

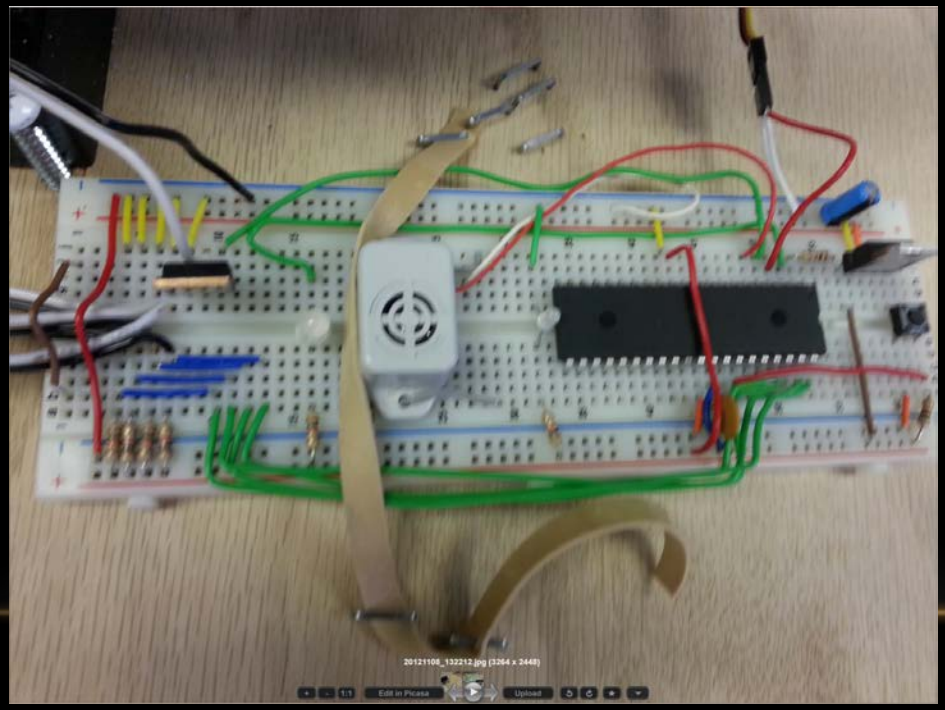
- Build a home defense automated torrent
- Have a fully automated airsoft gun that reacts to open doors or windows with a furry of bullets and report

INPUTS

- Four position switches located at doors and windows of the room for triggering the system
- Pushbutton was used in order to arm/disarm the system and to disable the alarm and firing

CONTROLLER

- We used PORTB interrupts for the door sensors
- We used a timer interrupt for the servo motor position
- All other aspects were run form the main code

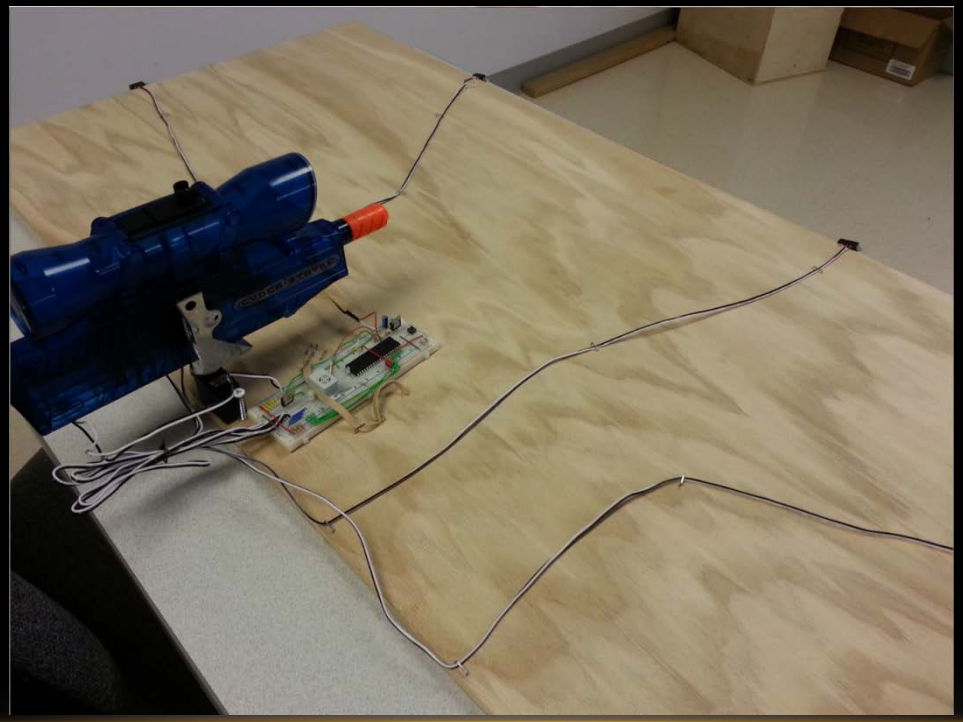


OUTPUTS

- Servo motor will turn the gun towards the location of the breach
- Mosfet would fire the gun
- PIC would activate an alarm
- One LED for power
- One LED for system status (arm/disarm)

RESULTS

- Robo-Cop performed valiantly in duty and the system worked as originally planned
- Robo-Cop was a first run prototype



ADVICE

- Pick something your interested in and will be excited about the final product
- FINISH ON TIME