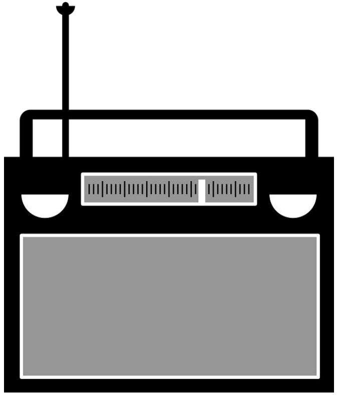
Lazy Man’s Radio

•••

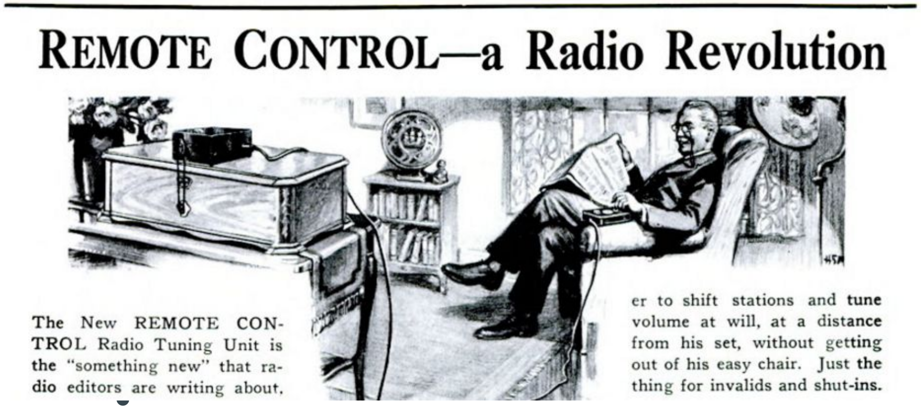
Chris Chivetta, Joseph D’amico, Stefan Olesnyckyj,
Sujay Tadwalkar, Anika Zaman

The Concept

MSP430 implementation of a lazy man's radio that integrates a soundboard, shift register, LCD display, and pir sensor to allow a user to lazily flip through his or her favorite tracks with a wave of the hand.



History and Demand



REMOTE CONTROL—a Radio Revolution

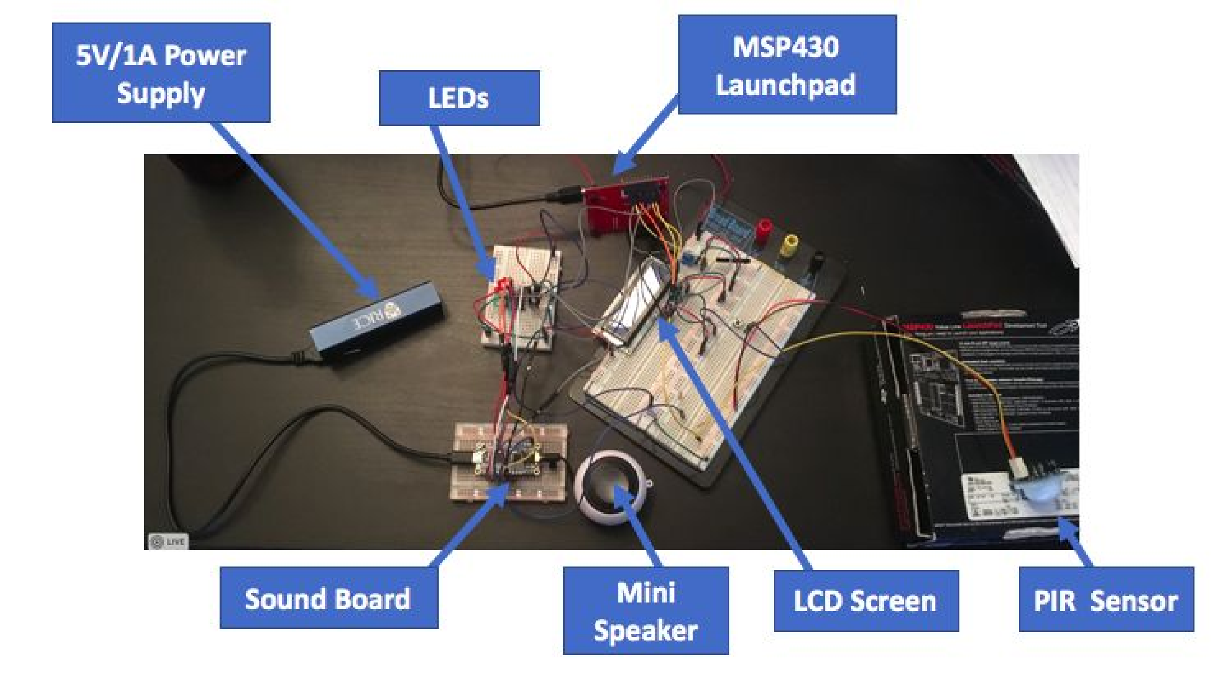
The New REMOTE CONTROL Radio Tuning Unit is the "something new" that radio editors are writing about.

er to shift stations and tune volume at will, at a distance from his set, without getting out of his easy chair. Just the thing for invalids and shut-ins.

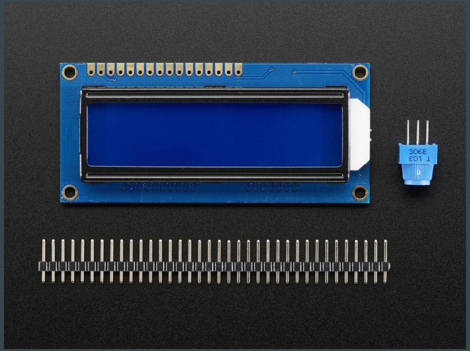
- Remote operation of music-playing devices has been desirable since the advent of radios
- Personal music devices such as mp3 players and cell phones with music apps
- Interfacing with personal music devices can be awkward and unwieldy in social settings, as you still have to touch the screen or buttons to change songs
- Being able to change song with a wave of your hand is extremely convenient

Features of Lazy Man's Radio

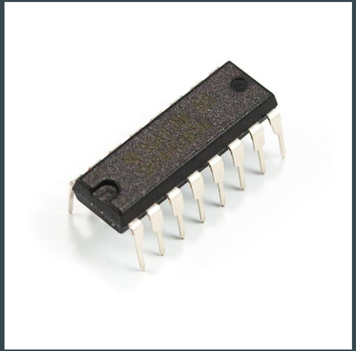
- LCD screen
- Motion Detector Track Selection
- Pushbutton Track Selection
- LED indicators
- Shift register
  - Modify and reprogram the system to skip one, two or more tracks per motion
  - Reorder the order in which to visit songs on the soundboard
- A (customizable) soundboard pre-loaded with eight of today's most popular hits



Parts

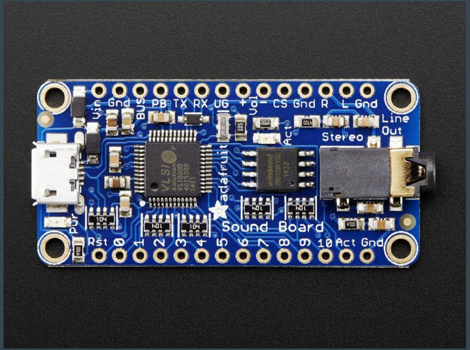


- 16x2 LCD Screen
    - Display track name for user
    - We are not using the SPI/I2C backpack for this model



- 74HC595 8-bit Shift Register
    - Reduce number of pins of the MSP430 that the Soundboard uses
    - Serves as 1-cold decoder

Parts


Sound Board
adafruit
Stereo
Line Out
Vin Gnd BUS PB TX RX UG + Vol - CS Gnd R L Gnd
Rst 0 1 2 3 4 5 6 7 8 9 10 Act Gnd

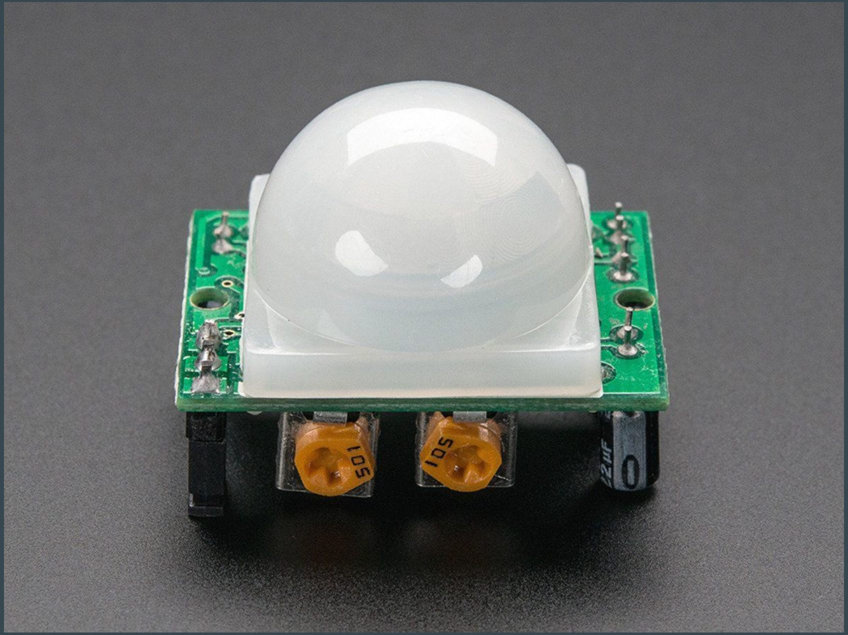
- Mini Pushbutton
  - Move to the next song

- Audio FX Sound Board
  - Audio sound effects and also has built in storage that holds up to 15 minutes of compressed audio

Parts

- MSP430 Launchpad
  - Using the MSP430G2553 model
  - Low cost
  - Low power consumption
  - Easy to interface with GPIO pins

Parts



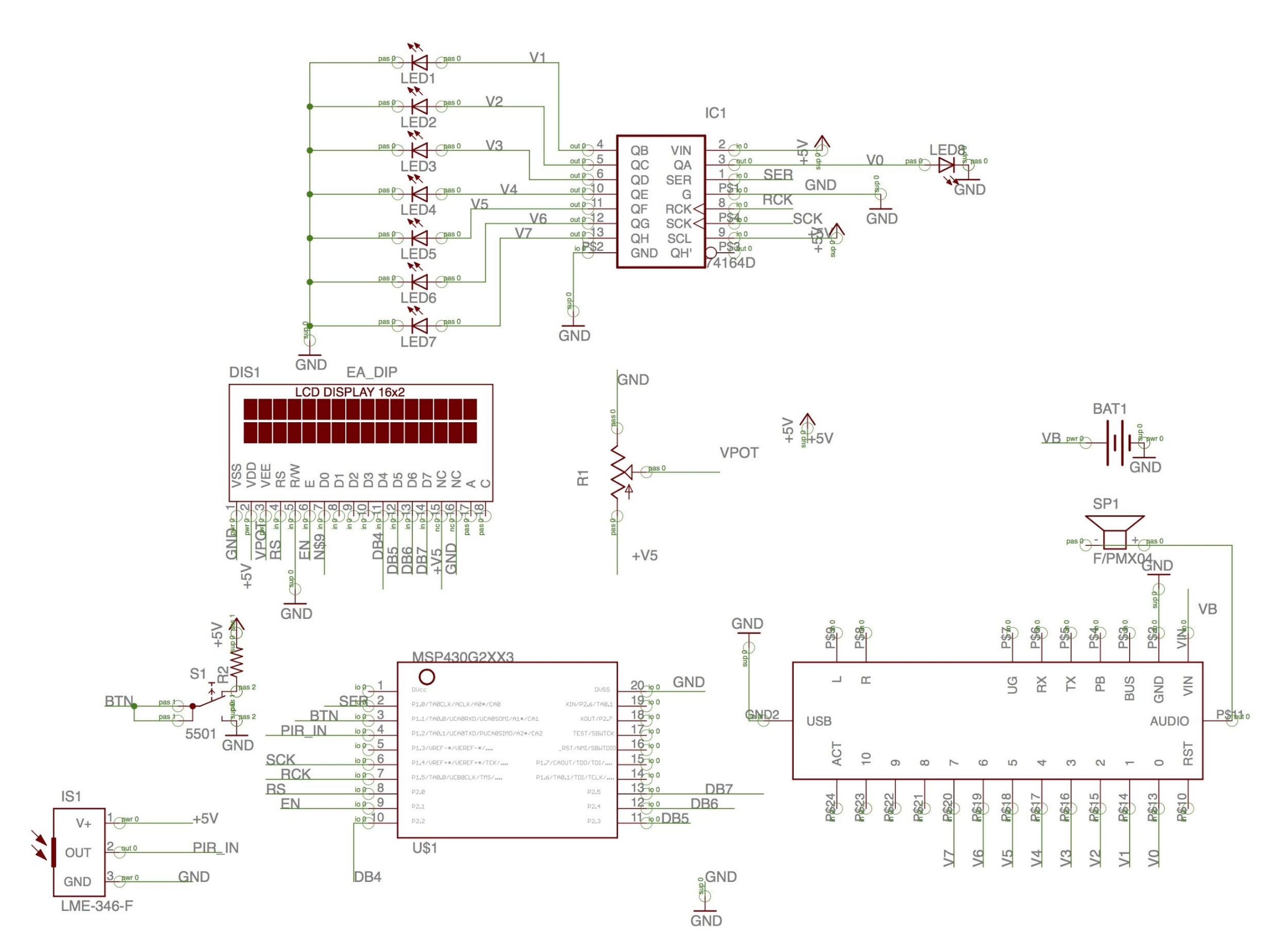
- PIR Motion Sensor
    - Pyroelectric (passive) InfraRed Sensor
    - Detect hand movements in order to move to the next song.
    - Fairly low power consumption

Parts

- 5V/1A USB Power Supply
  - Power the speaker and soundboard

- Mini Speaker
  - Play the sounds from the soundboard

Schematic


LED1
LED2
LED3
LED4
LED5
LED6
LED7
LED8
V0
V1
V2
V3
V4
V5
V6
V7
IC1
QB
QC
QD
QE
QF
QG
QH
GND
VIN
QA
SER
G
RCK
SCK
SCL
QH'
74164D
+5V
SER
GND
RCK
SCK
GND
DIS1
EA_DIP
LCD DISPLAY 16x2
VSS
VDD
VEE
RS
R/W
E
D0
D1
D2
D3
D4
D5
D6
D7
NC
NC
A
C
GND
+5V
VPOT
RS
EN
N$9
DB4
DB5
DB6
DB7
+V5
GND
R1
VPOT
+V5
BAT1
VB
SP1
F/PMX04
S1
R2
5501
BTN
MSP430G2XX3
U$1
DVcc
DVSS
SER
BTN
PIR_IN
SCK
RCK
RS
EN
DB4
DB5
DB6
DB7
IS1
V+
OUT
GND
LME-346-F
+5V
PIR_IN
GND
GND2
USB
AUDIO
L
R
UG
RX
TX
PB
BUS
GND
VIN
ACT
RST
VB

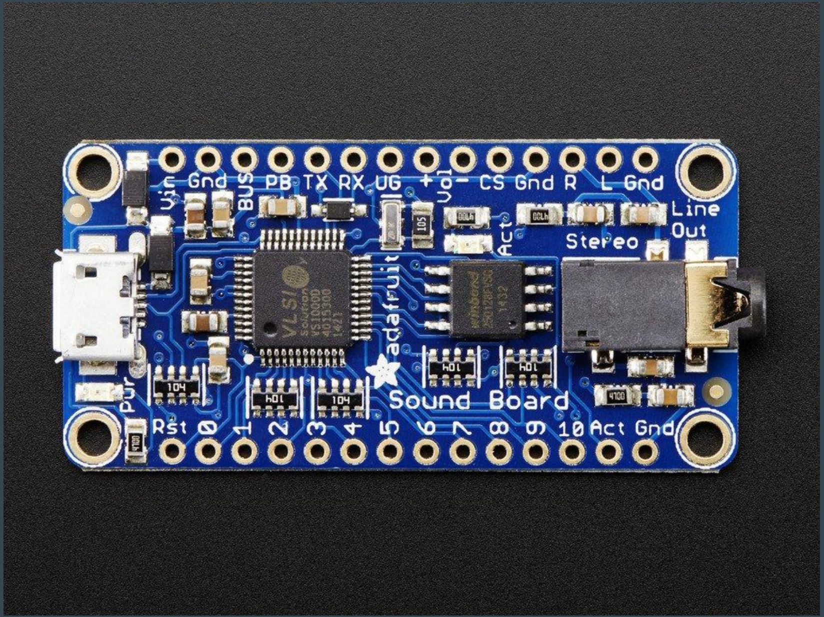
Hardware Architecture

- Interface with soundboard
- Interface with the LCD
- Interface with a motion sensor

Soundboard

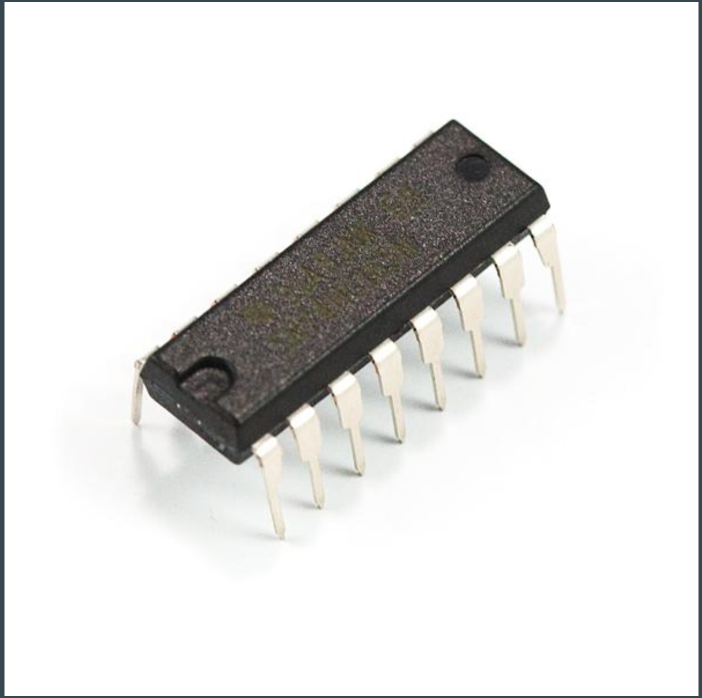
- Adafruit AudioFX soundboard
  - Low cost
  - Simplicity

- Soundboard has 11 pins
  - Just have to drive a corresponding pin low to play a corresponding song
  - MSP430 has limited number of GPIO pins
    - Had to find solution


Vin Gnd BUS PB TX RX UG +Vol- CS Gnd R L Gnd
Line Out
Stereo
Act
adafruit
Sound Board
Pwr
Rst 0 1 2 3 4 5 6 7 8 9 10 Act Gnd

The Solution

- 1-Cold Decoder (Initially)
    - Control 16 pins with only 4 GPIO pins of the MSP430
- Fear of timing issues
    - Switched to sequential equivalent
    - 74HC595 shift register
        - Control 8 pins with only 3 GPIO pins
        - Could scale with more shift registers if necessary



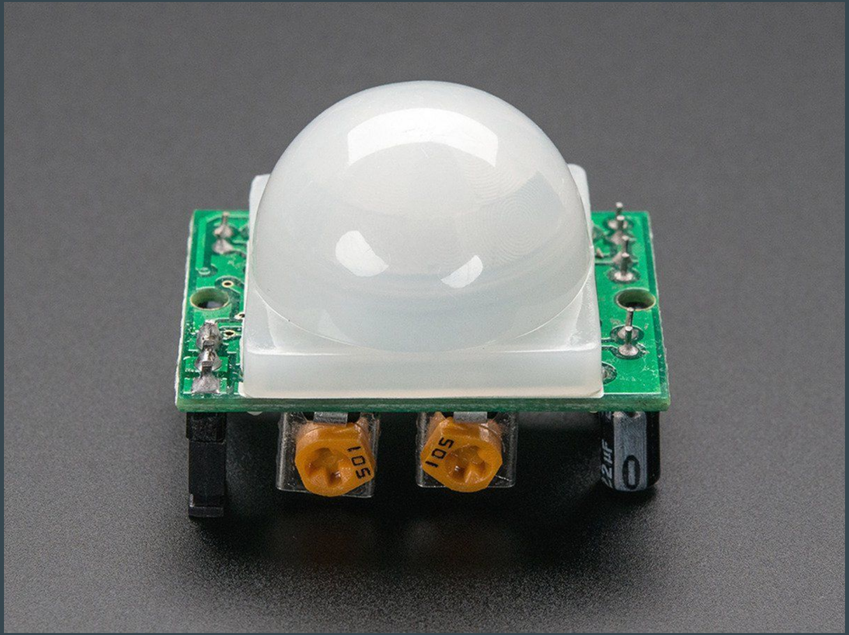
The LCD Display

- Initial considerations
  - Wanted to use fewer GPIO pins
  - Tried using SPI/I2C backpack
    - Only needed 3 to 4 pins instead of 6 or 7 pins
    - Support for the MSP430 is limited
    - Arduino drivers were written in C++
      - Difficult to port
  - Decided to use traditional interface with LCD display

- LCD Hardware
  - RS Pin
  - RW pin
  - E pin
  - DB4 to DB7
  - DB0 to DB3
- Initialization commands
  - Can configure LCD in different bit modes
  - We used 4 bit mode
  - Incorporated Kevin Lin’s LCD device driver

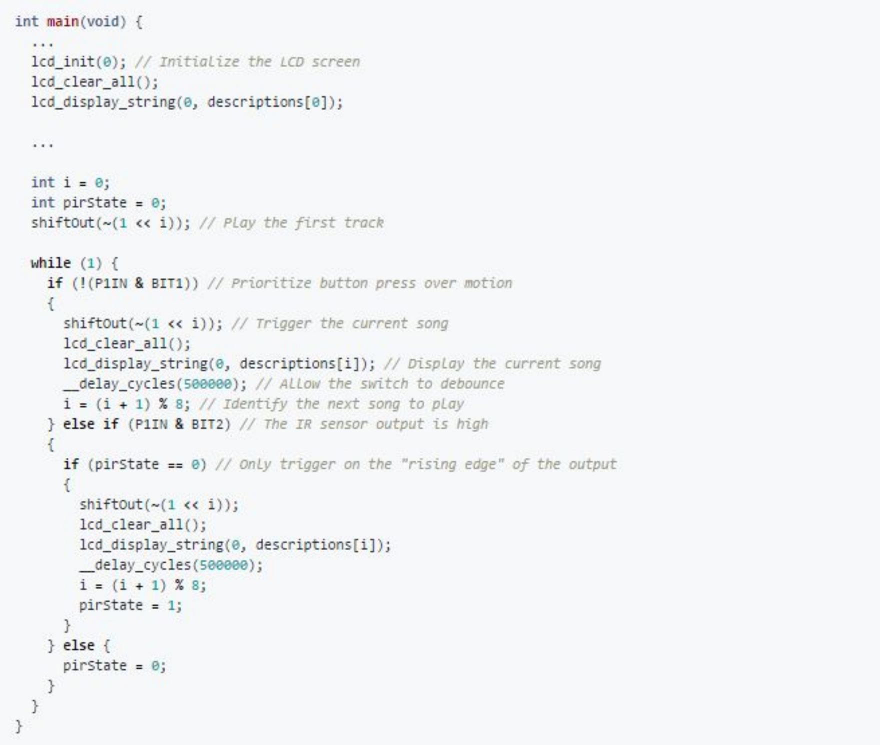
Motion Sensor

- Flight-Sensor
    - Get accurate readings of detected objects in the vicinity
    - I2C comatability
        - I2C was very difficult to implement
- PIR Motion sensor
    - Less accurate
    - Only 1 bit of information
    - Much easier to use with the rest of the system
    - Requires only 1 GPIO bit
        - Needs to interface with the ADC
    - Tuned sensitivity



Software Architecture

- Drivers
  - Open Source
  - Kevin Lin
  - Used these to interface with shift register and MSP430 from a high level
- Polling system
  - Poll switchbutton and PIR
  - Planning on using interrupts i the future
  - Not enough time to organize an interrupt system



```
int main(void) {
  ...
  lcd_init(0); // Initialize the LCD screen
  lcd_clear_all();
  lcd_display_string(0, descriptions[0]);

  ...

  int i = 0;
  int pirState = 0;
  shiftOut(~(1 << i)); // Play the first track

  while (1) {
    if (!(P1IN & BIT1)) // Prioritize button press over motion
    {
      shiftOut(~(1 << i)); // Trigger the current song
      lcd_clear_all();
      lcd_display_string(0, descriptions[i]); // Display the current song
      __delay_cycles(500000); // Allow the switch to debounce
      i = (i + 1) % 8; // Identify the next song to play
    } else if (P1IN & BIT2) // The IR sensor output is high
    {
      if (pirState == 0) // Only trigger on the "rising edge" of the output
      {
        shiftOut(~(1 << i));
        lcd_clear_all();
        lcd_display_string(0, descriptions[i]);
        __delay_cycles(500000);
        i = (i + 1) % 8;
        pirState = 1;
      }
    } else {
      pirState = 0;
    }
  }
}
```

Class Concepts

- Soldering
- Breadboarding
- PCB design
- Sensor inputs
- ADC Utilization

Next Steps

- Implementing our system design on our PCB
- Reorganizing software to use an interrupt-enabled system instead of a polling system
- Using a more sophisticated motion sensor
  - Flight Sensor
- Using I2C/SPI to interface with the LCD screen
- Upgrading sound board