CS 362 - Computer Design Milestone 7

Player 1: William Ocampo | wocampo2 | wocampo2@uic.edu
Player 2: Amy Ngo | ango8 | ango8@uic.edu

Title: Internet Radio

Abstract:

This project implements an Internet Radio utilizing the Raspberry Pi and Arduino Uno. The Pi runs the music player and outputs its sound to speakers. The Arduino operates the radio via buttons pressed. An external Arduino interacts with the radio through either a port interface or through sound waves generated by the radio. The Pi communicates with the main Arduino via a specific firmware that will make the Arduino be operated by the Pi through Python code running on the Pi itself. Almost every component is connected via wires when the physical radio gets assembled.

Detailed Project Ideas:

Overall Description of Project Idea:

Our project idea is to construct a physical Internet Radio using the Raspberry Pi 3 and the Arduino Uno. This will be implemented in three main steps. The first main step is to setup the Raspberry Pi, if one hasn't done so already, with the Raspbian OS and the required packages as well as implement the radio onto the Pi. The second main step is to write the Python code that will be uploaded to the Arduino Uno with the LCD screen already mounted on. The third and final main step is to assemble all the hardware (Pi, Uno, speakers, etc.) into a radio cage to construct the physical radio and build and implement the creative LED design circuit.

Final Project Design stating how Multiple Arduinos will be used:

The Pi will run the music player, which will receive its input from an online radio station. The Arduino Uno will be used as a controller to operate the radio (pause, play, next, etc.), as well as display the current track being played on the LCD monitor. The creative aspect is the radio turning off whenever the photoresistor detects shade. The first button will play the next track stored in the Pi. The second button is used to pause and play the current track.

Final Plan for Use and Communication between the multiple Arduinos:

In order to get the Pi to communicate with the Arduino, we are going to need to download a specific library on Python, as well as specific firmware for the

Arduino. This will allow the Pi, with a USB connection to the Arduino, to operate the Arduino via a program running on the Pi. The Pi communicates with the external Arduino by direct connection.

Final Project Design stating Expected Inputs/Outputs:

For starters, we are going to need an Internet connection for the Pi to be able to receive the online radio transmission. Thankfully, the Pi 3 already has built-in Wi-Fi; however we need to use Secure Shell to the Pi via a computer or laptop to make the project a bit more portable. As for sound, we are going to output sound from speakers or earphones. Also, to be able to print out the current track playing, we’re going to need to use a 16x2 LCD display. Lastly, we are going to need two buttons to operate the online radio to go to the next track and play or pause the song.

Final Description of the original work being attempted by your project:

The main references we’re using for this project are outdated. A lot of the parts we have to implement aren't going to be the same. As such, we're going to have to do a lot of research and tweaking to get the main part functioning correctly. We're also implementing an external Arduino Uno to interact with the radio. The feature will have the radio turn off automatically if it gets too dark in the room. In addition of a button pausing the song and playing the song. Another button is used to go to the next track stored in the Pi.

Discussion on how to build the project:

We plan to use Liquid Crystal on the LCD display to show the radio station and song title. We choose one button to be an interrupt state. The interrupt button changes the text of the next track and plays that track. The second button behaves like an on and off state. When pressed to pause, the song’s time is stored in the music player client. The stored time is used to keep track of where it last left off and start playing from that time. As the photoresistor reads a value and determines if lack of light is detected. When the read value is under a certain range, the Pi will stop playing the song and shut off the program.

CS 362 - Computer Design Milestone 7
Player 1: William Ocampo | wocampo2 | wocampo2@uic.edu
Player 2: Amy Ngo | ango8 | ango8@uic.edu
Title: Internet Radio

Discussion on how the project is used:

The user inputs commands to open the song files and add to the Pi. Certain commands are used to start up the Pi and play music. One of the buttons is used as a controller to play and pause the radio. The second button is to go to the next track. The user can control the volume from their speakers or headphones. The user can manually turn off the Pi by typing commands in Secure Shell or hover over the photoresistor until it’s dark enough to turn off the program.

Supporting Materials:

Proposed Timeline:

Preliminary Installation and Setup by Friday 10/25/19 (Completed)
Add Radio to Raspberry Pi by Friday 11/1/19 (Completed)
Write Python Sketch for Arduino Uno by Friday 11/8/19 (Completed)
Assemble Physical Radio by Friday 11/15/19 (Completed)
Add Photoresistor and Interrupt Buttons to External Arduino by Sunday 11/24/19 (Completed)
Design Presentation on Monday 11/25/19 (Completed)
Project Demonstration on Monday 12/2/19 (Completed)

List of Materials:

x1 Raspberry Pi 3 Model B
x1 Arduino Uno
x1 Monitor
x1 Keyboard
X1 Mouse
x1 Speakers or Headphones
x1 Radio Cage
x1 USB Cable
x1 16x2 LCD Display
x4 Resistors
x2 Buttons
x1 Photoresistor
x25 Jumper Wires

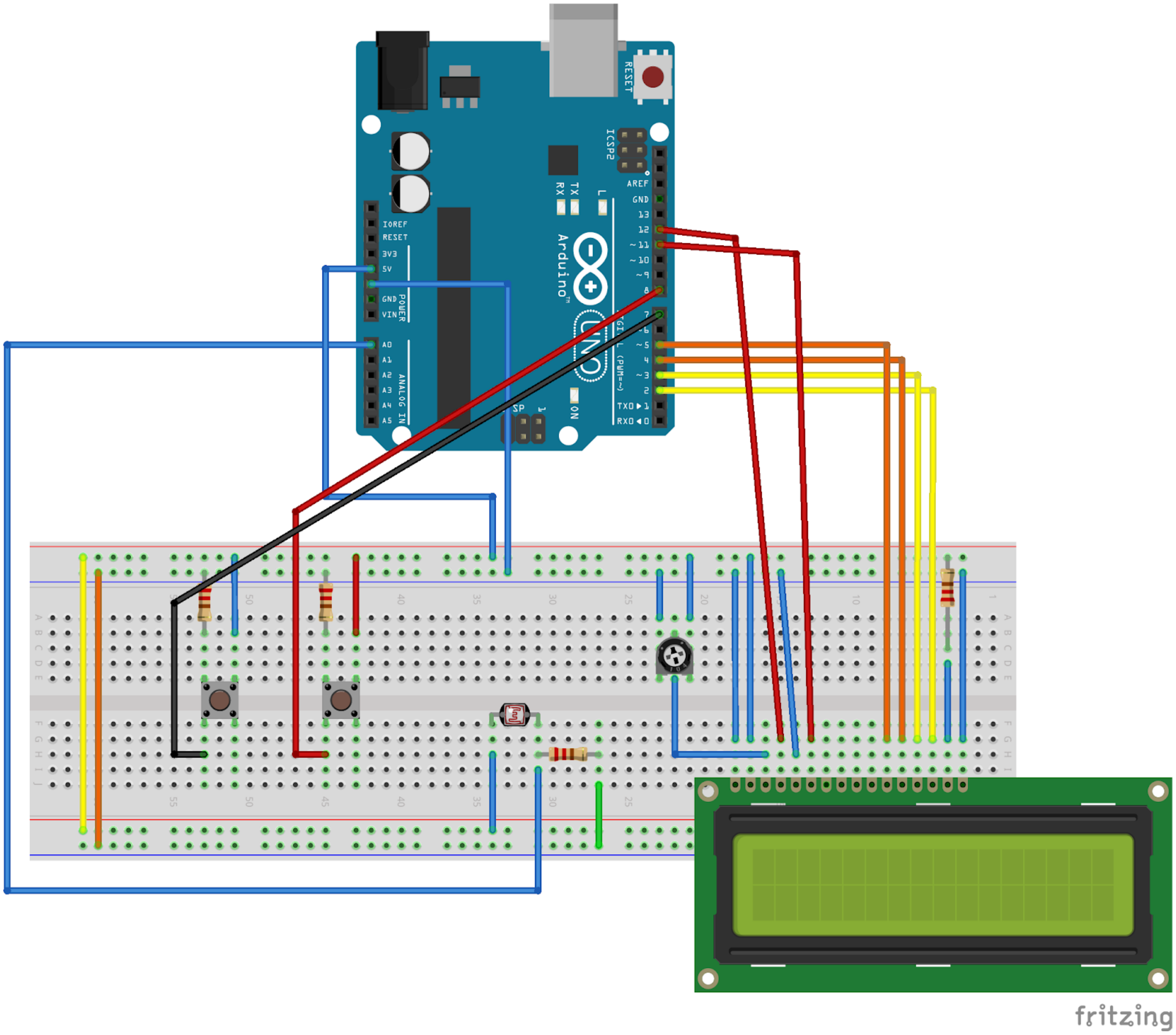
CS 362 - Computer Design Milestone 7
Player 1: William Ocampo | wocampo2 | wocampo2@uic.edu
Player 2: Amy Ngo | ango8 | ango8@uic.edu
Title: Internet Radio

List of References:

https://www.instructables.com/id/Arduino-Raspberry-Pi-Internet-Radio/
https://www.instructables.com/id/How-to-use-a-photoresistor-or-photocell-Arduino-Tu
https://www.arduino.cc/en/Tutorial/HelloWorld
https://www.arduino.cc/en/Tutorial/Button
http://arduino.cc/en/Reference/attachInterrupt
https://www.youtube.com/watch?v=QumIhvYtRKQ&t=341s
https://pypi.org/project/nanpy/
https://www.radionomy.com/en/style

CS 362 - Computer Design Milestone 7
Player 1: William Ocampo | wocampo2 | wocampo2@uic.edu
Player 2: Amy Ngo | ango8 | ango8@uic.edu
Title: Internet Radio

Final Diagram (Fritzing)



CS 362 - Computer Design Milestone 7
Player 1: William Ocampo | wocampo2 | wocampo2@uic.edu
Player 2: Amy Ngo | ango8 | ango8@uic.edu
Title: Internet Radio

Final Code Sketches (Python):

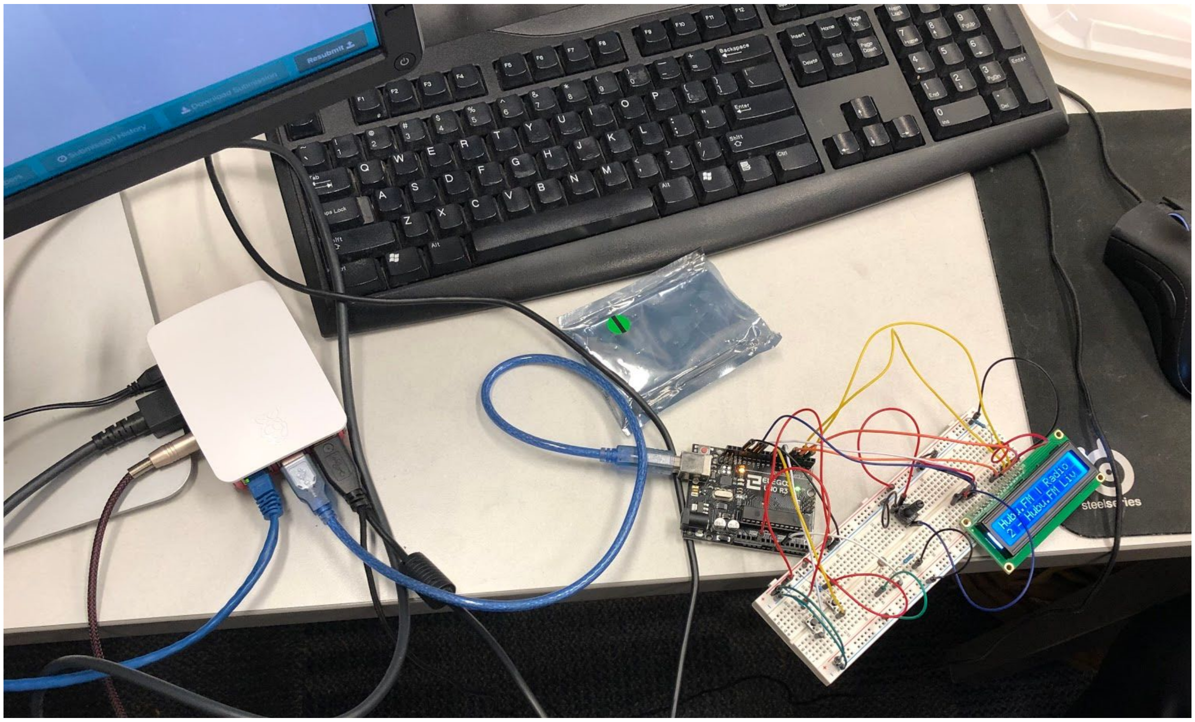
```
import os
from nanpy import Arduino, Lcd

# Initialize buttons and states
buttonPin = 7
buttonPin2 = 8
photoR = 14
buttonState = 0
buttonState2 = 0
val = 0

# Activate buttons
Arduino.pinMode(buttonPin, input)
Arduino.pinMode(buttonPin2, input)
Arduino.pinMode(photoR, input)

# Initialize LCD
lcd = Lcd([12, 11, 5, 4, 3, 2], [16, 2])

# Maximum number of stations held in the mpc list
max_trax = 5

# Print station info to LCD
def getTrack():
   L= [S.strip('\n') for S in os.popen('mpc').readlines()]
   station = L[0][0:15]
   track = L[0][-16:-1]
   lcd.printString(16*" ", 0, 0)
   lcd.printString(station, 0, 0)
   lcd.printString(16*" ", 0, 1)
   lcd.printString(track, 0, 1)
   print (L)
   print (station)
   print (track)

# Play first station and print out station info to the LCD
track_num = 1
os.system("mpc play "+str(track_num))
getTrack()
```

CS 362 - Computer Design Milestone 7
Player 1: William Ocampo | wocampo2 | wocampo2@uic.edu
Player 2: Amy Ngo | ango8 | ango8@uic.edu
Title: Internet Radio

```
# Event loop
while True:
   # Get button inputs
   buttonState = Arduino.digitalRead(buttonPin)
   buttonState2 = Arduino.digitalRead(buttonPin2)
   val = Arduino.analogRead(photoR)
   # Change to next station
   if buttonState:
     track_num += 1
     # After track 4, go back to track 1
     if track_num > max_trax:
          track_num = 1
     os.system("mpc play " + str(track_num))
     getTrack()
   # Toggle station
   elif buttonState2:
     os.system("mpc toggle")
     getTrack()
   # Turn off radio
   elif val < 200:
     os.system("mpc pause")
     break
```

CS 362 - Computer Design Milestone 7
Player 1: William Ocampo | wocampo2 | wocampo2@uic.edu
Player 2: Amy Ngo | ango8 | ango8@uic.edu
Title: Internet Radio

Images



CS 362 - Computer Design Milestone 7
Player 1: William Ocampo | wocampo2 | wocampo2@uic.edu
Player 2: Amy Ngo | ango8 | ango8@uic.edu
Title: Internet Radio