CS/EE 120B Custom Laboratory Project Report

Iron Man's Mark I Arc Reactor

Nicole Garcia

03/17/2022

Introduction

The Mark I Arc Reactor consists of a 3D printed housing where 2 LED rings are embedded inside. These rings produce different light modes of different durations based on user input. The input can be a manual toggle via a pushbutton or any Android bluetooth capable device using google voice. The LED rings are controlled by three daisy chained shift registers, and voice input is done via a HC-05 bluetooth module. Along with each mode, there is a corresponding audio file that plays via a speaker in sync with the selected mode.

The audio complexity was the only feature I was not able to integrate with the system.

Complexities

- Shift Registers
    - Successfully implemented the controlling of each of the three shift registers outputs to toggle specific LEDs
- HC-05 Bluetooth module
    - Able to Serialize voice input to toggle through the modes
- Speaker/Audio
    - Unable to integrate at this time

User Guide

If not using the bluetooth feature, simply use the pushbutton to toggle between modes.

To use Bluetooth via an Android phone download the AMR_Voice app on the play store.

After downloading, make sure the arc reactor is on. Enter the app, use the menu button to open up a navigation tab then click on “connect robot” and select the bluetooth device titled “HC-05”. Then use any key word corresponding to the specific mode wanted.

Hardware Components Used

Quantity,Component

1, Arduino Uno R3

3, 8-Bit Shift Register

10,Red LED

9,White LED

16,220 Ω Resistor

1,HC-05 Bluetooth Module

1,1 kΩ Resistor

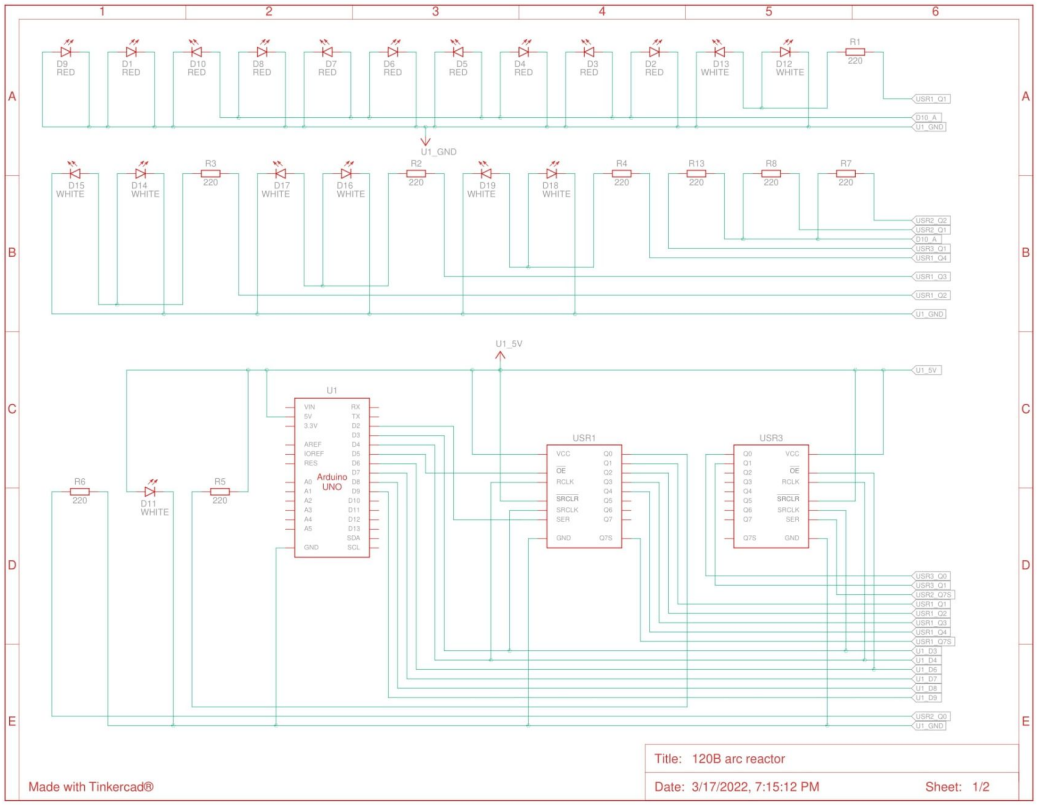
1,2 kΩ Resistor

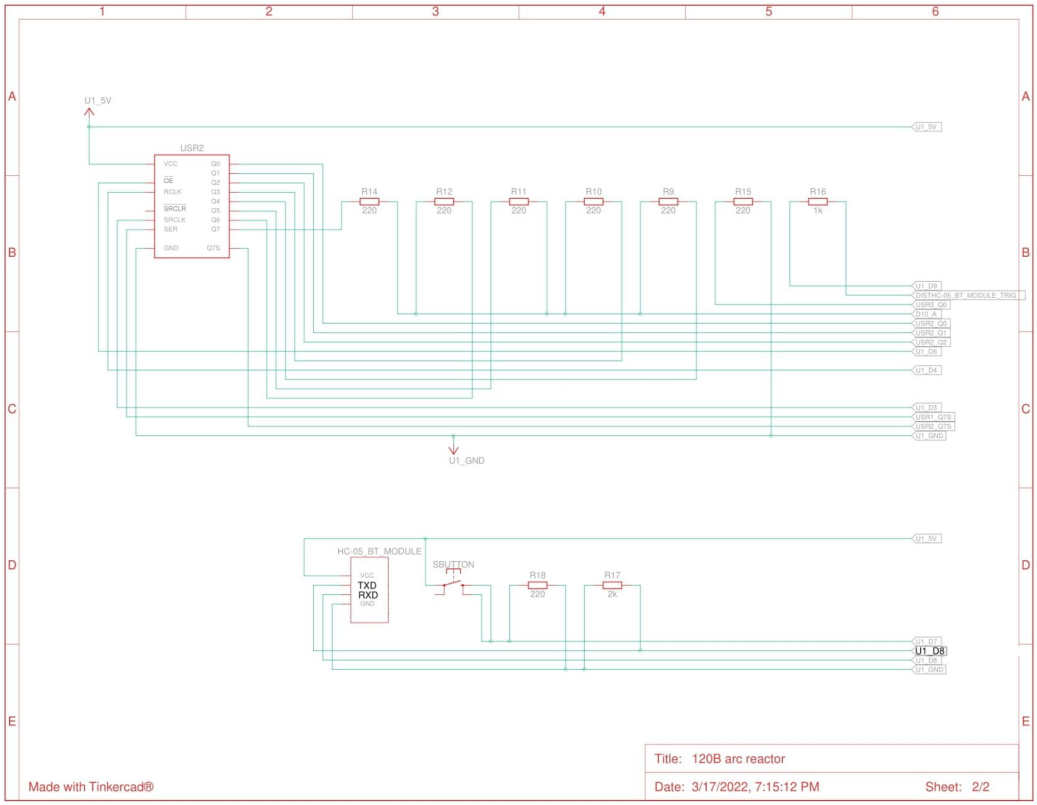
1, Pushbutton

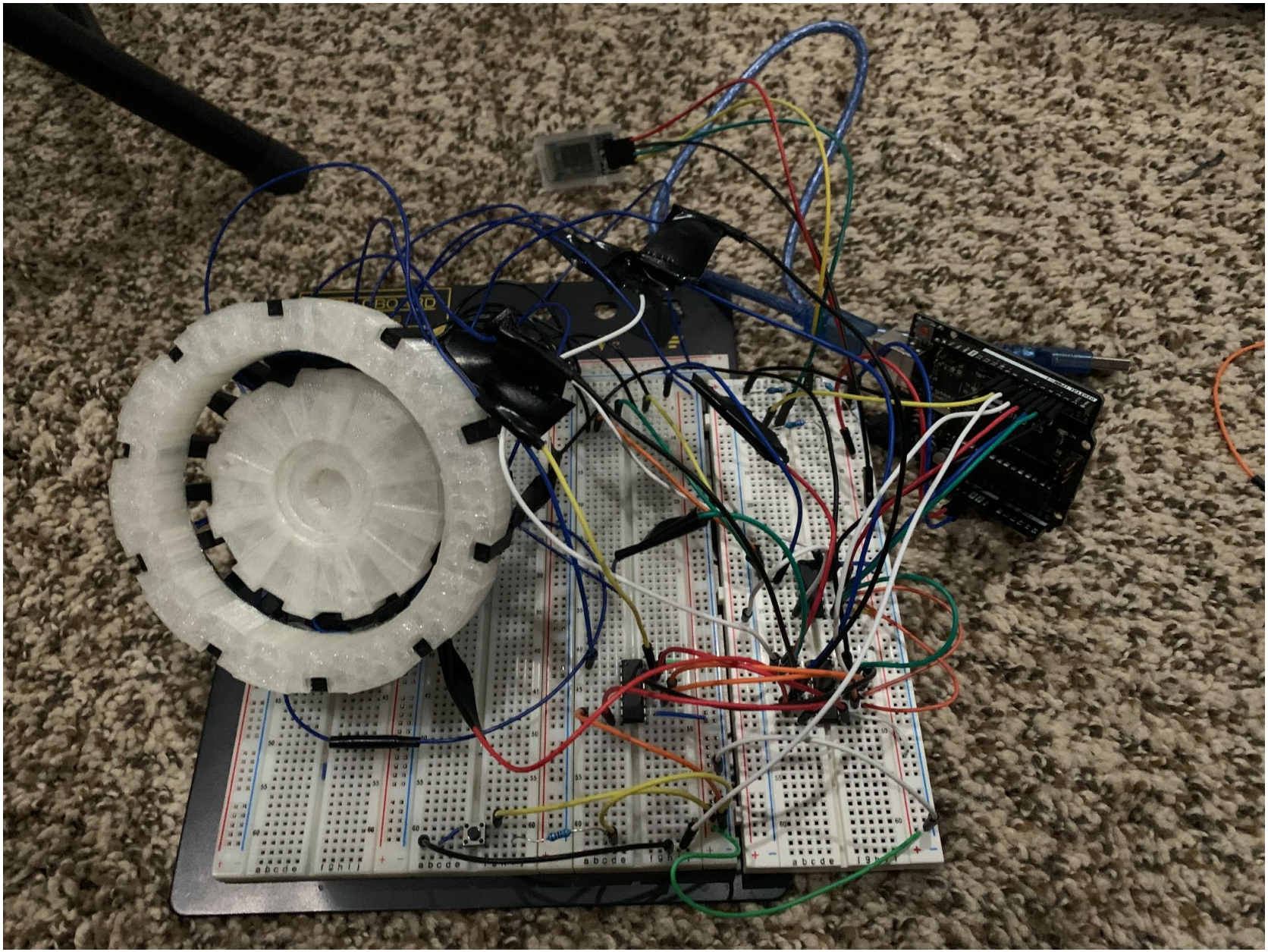
Software Libraries Used

- <ShiftRegister74HC595.h>
    - The library functions allow for a user-friendly interface to control each shift register’s output pins. I can select individual pins and check current pin states which makes programming each light mode infinitely easier.
- <SoftwareSerial.h>
    - Allows for the HC-05 Bluetooth Module’s serialized voice commands to be transmitted to the microcontroller.

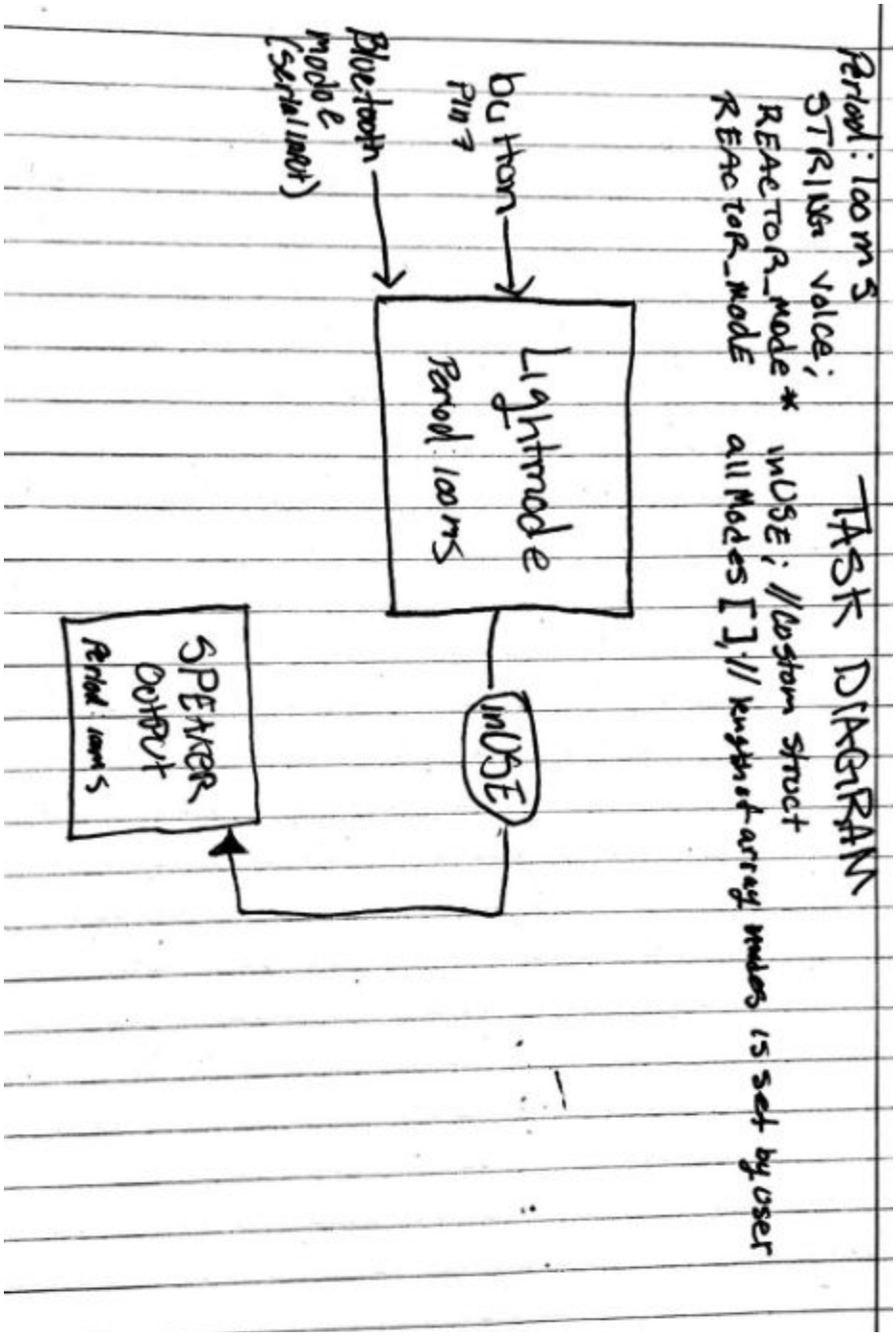
Wiring Diagram


D9 RED
D1 RED
D10 RED
D8 RED
D7 RED
D6 RED
D5 RED
D4 RED
D3 RED
D2 RED
D13 WHITE
D12 WHITE
R1 220
USR1_Q1
D16_A
U1_GND
D15 WHITE
D14 WHITE
R3 220
D17 WHITE
D16 WHITE
R2 220
D19 WHITE
D18 WHITE
R4 220
R13 220
R8 220
R7 220
USR2_Q2
USR2_Q1
D10_A
USR3_Q1
USR1_Q4
USR1_Q3
USR1_Q2
U1_GND
U1_5V
U1_5V
U1
Arduino UNO
USR1
USR3
R6 220
D11 WHITE
R5 220
USR3_Q0
USR3_Q1
USR3_Q7S
USR1_Q1
USR1_Q2
USR1_Q3
USR1_Q4
USR1_Q7S
U1_D3
U1_D4
U1_D6
U1_D7
U1_D8
U1_D9
USR2_Q0
U1_GND
Made with Tinkercad®
Title: 120B arc reactor
Date: 3/17/2022, 7:15:12 PM
Sheet: 1/2


1
2
3
4
5
6
A
B
C
D
E
U1_5V
USR2
VCC
OE
RCLK
SRCLR
SRCLK
SER
GND
Q0
Q1
Q2
Q3
Q4
Q5
Q6
Q7
Q7S
R14
220
R12
220
R11
220
R10
220
R9
220
R15
220
R16
1K
U1_D9
DISTHC-05_BT_MODULE_TRIG
USR3_Q6
D10_A
USR2_Q6
USR2_Q1
USR2_Q2
U1_D6
U1_D4
U1_D3
USR1_Q7S
USR2_Q7S
U1_GND
HC-05_BT_MODULE
VCC
TXD
RXD
GND
SBUTTON
R18
220
R17
2k
U1_D7
U1_D8
U1_D5
Made with Tinkercad®
Title: 120B arc reactor
Date: 3/17/2022, 7:15:12 PM
Sheet: 2/2

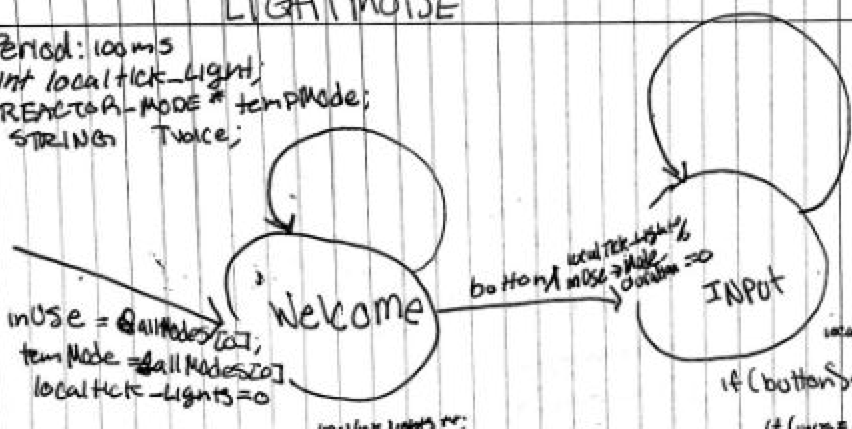


Task Diagram



LIGHTMODE

Period: 100ms
int localtick_Light;
REACTOR_MODE * tempMode;
STRING TVoice;

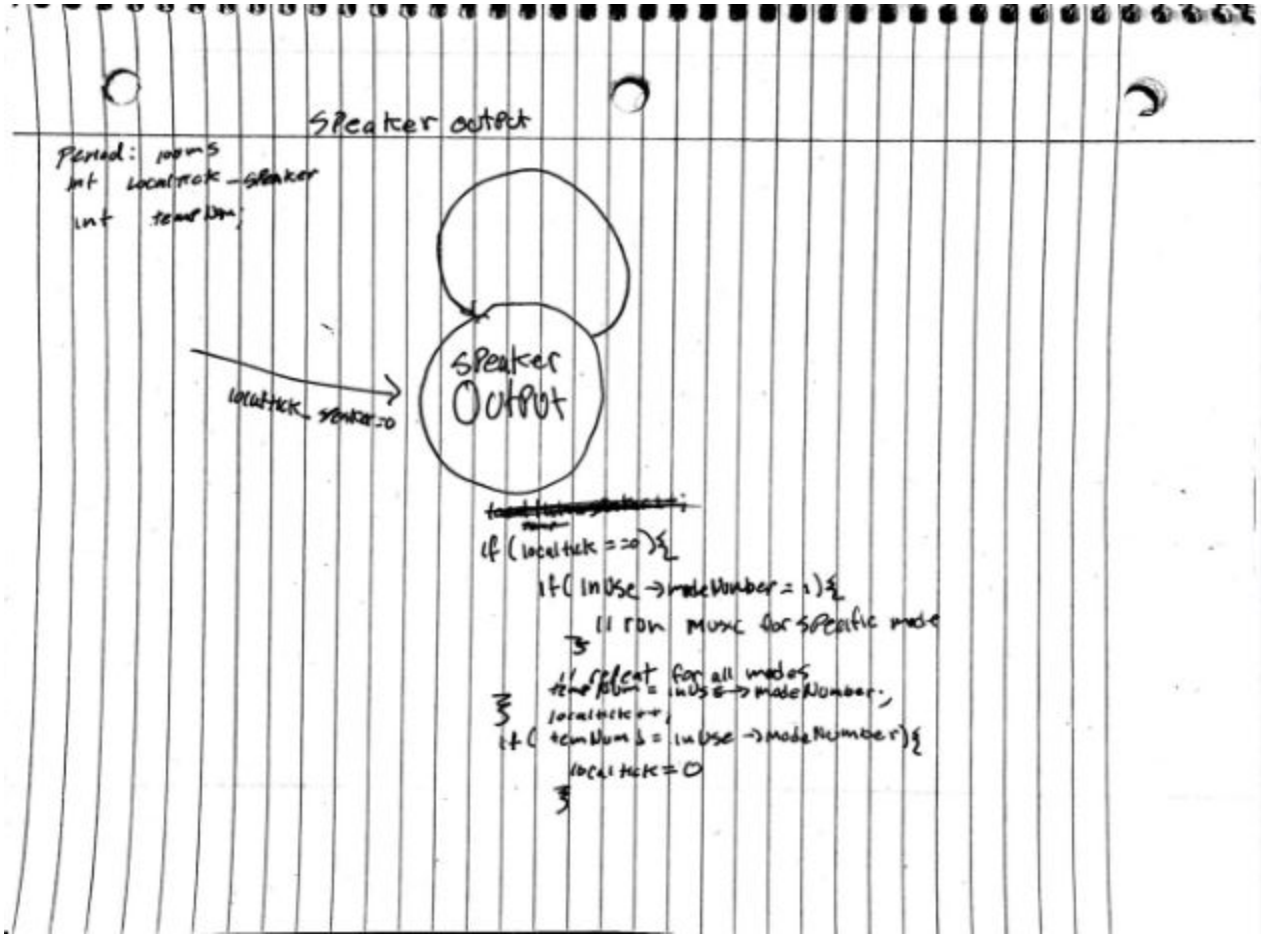


inUse = &allModes[0];
tempMode = &allModes[0]
localtick_Lights=0

localtick_Lights ++;

local tick_Lights ++;

```
if (button){ tempMode = &allModes[inUse→mode Number +1]
    if (inUSE→modeNumber > totalModes)
        { tempMode = &allModes[0]; }
    else { inUSE = tempMode; }
}
While ( BT. avalible() ){
    // Read in Chars into voice variable
    TVoice = voice;
}
int num = set BT Mode(TVoice); // returns mode num
                                    or -1 if No match
if ( num != -1){ temp mode = &allModes[num];
inUSE = tempMode }
if (inUse→modeNumber == 1) { // call light mode
                                   fun for 1 }
if ( ... == 2 ....
```


Speaker output
Period: 100ms
int localTick_Speaker
int temNum;
localTick_Speaker=0
Speaker
Output
if (localTick ==0){
if(inUse->modeNumber = 1){
// rdm Music for specific mode
}
// repeat for all modes
temNum = inUse->modeNumber;
localTick++;
if(temNum = inUse->modeNumber){
localTick=0
}