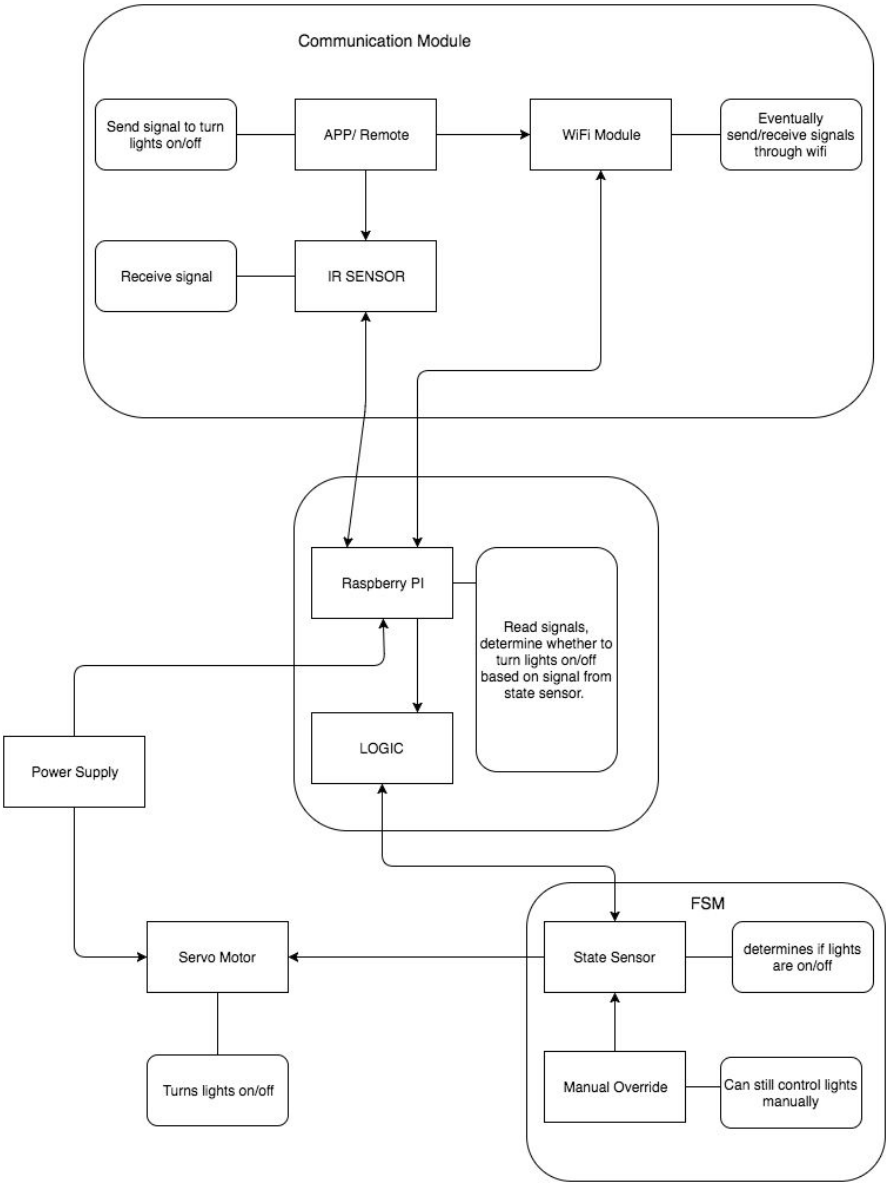
Wireless Communication and Lights
Div Nagpaul
Fares Takieddine
Jeff Ito

Introduction

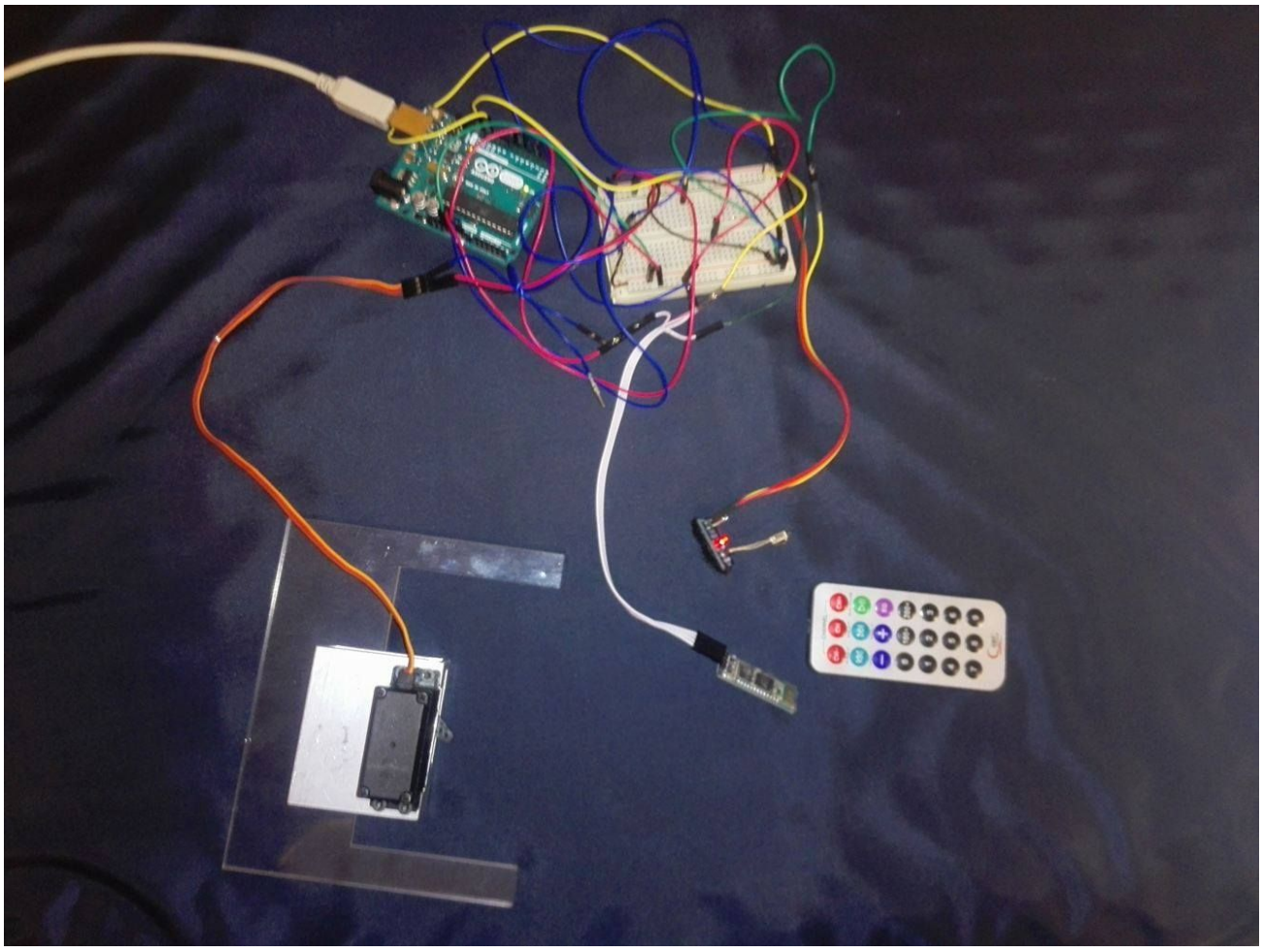
We wanted to create a system to satisfy the needs your average, lazy college student. The student has just got into bed and realized he did not lock the door or turn off the lights. He is already too comfortable to get back out of bed and do these things. Instead he could use our solution of a wireless communication system that would lock the doors and turn off the lights with the push of a button and without having to get out of bed.

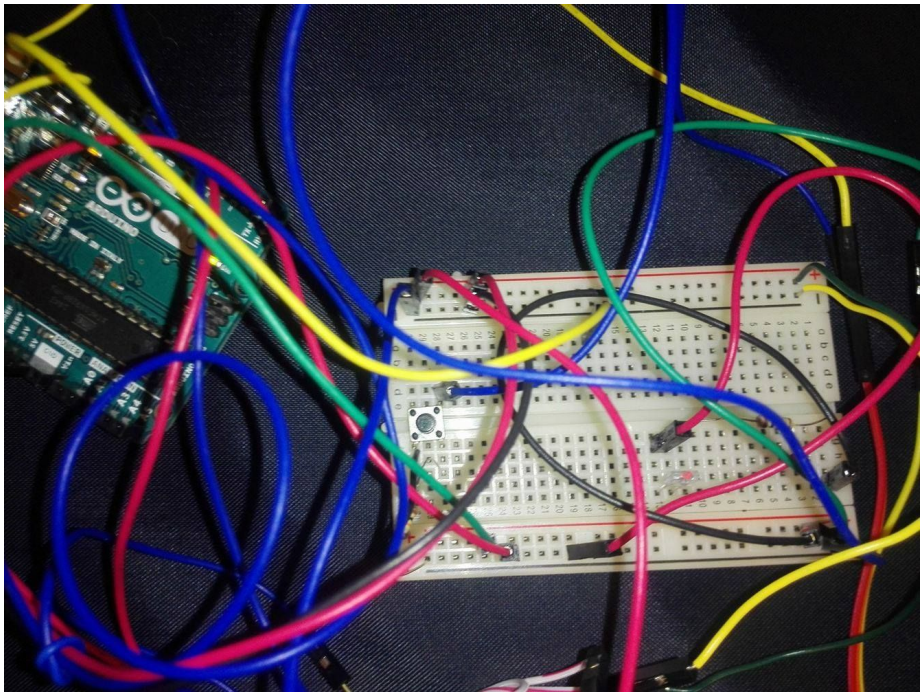
Design

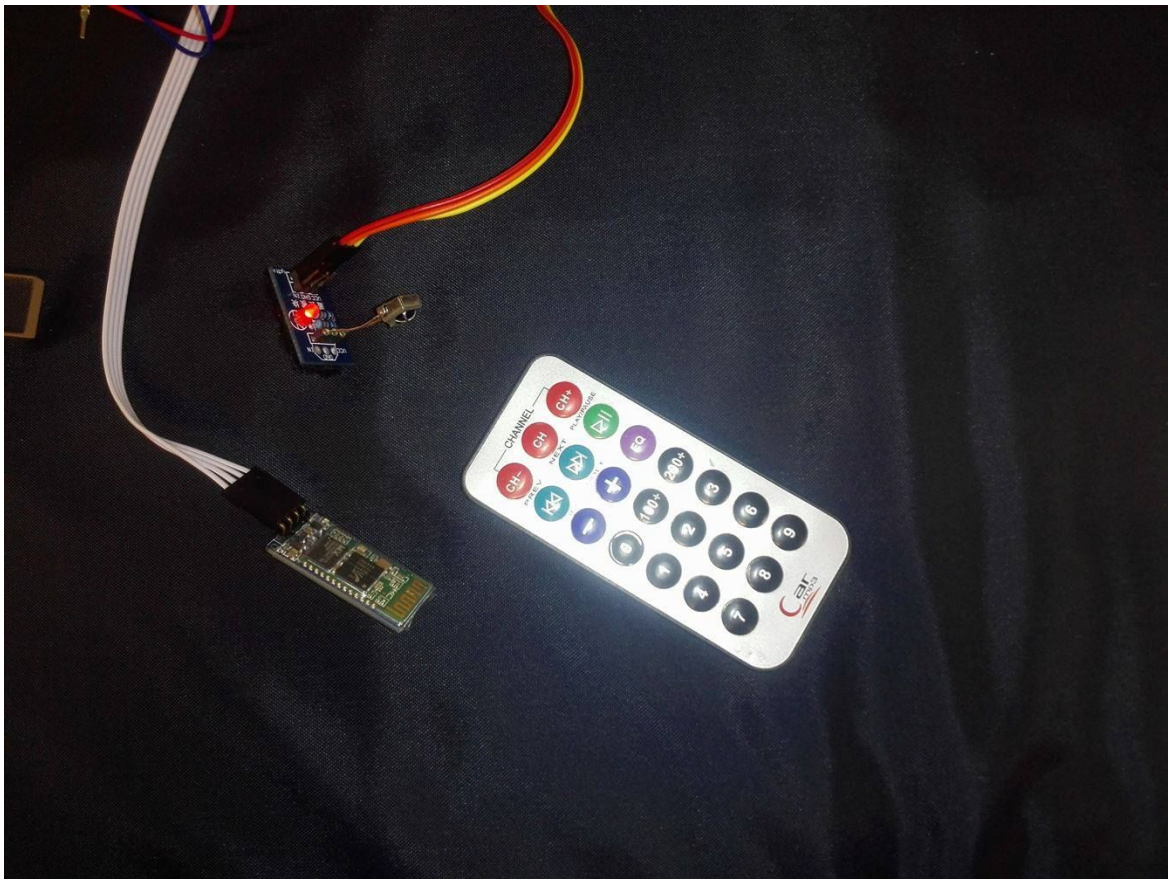


Our initial design utilized a remote that would send a signal to an infrared sensor. The sensor would be connected to a Raspberry Pi so that when it received a signal it would tell the Raspberry Pi to activate. The Raspberry Pi would be connected to a logic circuit and a state sensor to determine what state the servo motor was in. Based on the state, the Raspberry Pi would then tell the servo motor to move accordingly. We would also have a manual override button attached to the servo motor so that you would be able to turn the lights on and off without have to use the remote. Eventually we would integrate a Wi-Fi module to the Raspberry Pi and communicate through an app.

Our final design was very close to the original. The only minor changes we made were utilizing an Arduino-Uno rather than a Raspberry Pi. One other minor change we made was to use a bluetooth module attached to the Arduino to communicate with terminal commands rather than a Wi-Fi module communicating with an app.






CHANNEL
CH+
CH
CH-
PLAY/PAUSE
VOL-
VOL+
EQ
100+
200+
0
1
2
3
4
5
6
7
8
9

Final Code:

```
#include "Servo.h"
#include "stdio.h"
#include <IRremote.h>
#include <SoftwareSerial.h>
int bluetoothTx = 2;  // TX-O pin of bluetooth mate, Arduino D2
int bluetoothRx = 3;  // RX-I pin of bluetooth mate, Arduino D3
int dataFromBt;
SoftwareSerial bluetooth(bluetoothTx, bluetoothRx);
Servo myservo;
const int buttonPin = 5;     // the number of the pushbutton pin
int currentPosition=40;
int buttonState = 0;         // variable for reading the pushbutton status
int x =0;
int z=0;
int receiver = 11; // pin 1 of IR receiver to Arduino digital pin 11
IRrecv irrecv(receiver);           // create instance of 'irrecv'
decode_results results;            // create instance of 'decode_results'
//int analogdecode=A5;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);      // open the serial port at 9600 bps:
  bluetooth.begin(115200);  // The Bluetooth Mate defaults to 115200bps
  bluetooth.print("$");  // Print three times individually
  bluetooth.print("$");
  bluetooth.print("$");  // Enter command mode
  delay(100);  // Short delay, wait for the Mate to send back CMD
  bluetooth.println("U,9600,N");  // Temporarily Change the baudrate to 9600, no parity
  // 115200 can be too fast at times for NewSoftSerial to relay the data reliably
  bluetooth.begin(9600);  // Start bluetooth serial at 9600
  pinMode(buttonPin, INPUT);
  myservo.attach(7);
  Serial.println("IR Receiver Raw Data + Button Decode Test");
  irrecv.enableIRIn(); // Start the receiver
}
void loop() {
///// bluetooth begins
int bluetoothcalled=0;
if (bluetooth.available()) // If the bluetooth sent any characters
  {Serial.println("something was sent");
    // Send any characters the bluetooth prints to the serial monitor
    bluetoothcalled=1;
```

```
    Serial.println((char)bluetooth.read());
    dataFromBt = bluetooth.read();
  }
/////bluetooth ends
    // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);
 //Serial.print(buttonState);
  int y=1000;
  // check if the pushbutton is pressed.
  // if it is, the buttonState is HIGH:
  if (irrecv.decode(&results) ){ // have we received an IR signal?
     //irrecv.decode(&results);
    Serial.println(results.value, HEX); // UN Comment to see raw values
    y=translateIR();//(buttonState);
     //delay(1000);
    irrecv.resume(); // receive the next value
}
  //Serial.println(y);
  if(y==0||buttonState==1||bluetoothcalled==1 ){
     if(z==0){
      myservo.write(179);
      delay(1500);
      z=1;
    }
    else{
      myservo.write(0);
       delay(1500);
      z=0;
    }
  }
  //irrecv.resume();
}
int translateIR() // takes action based on IR code received
// describing Car MP3 IR codes
{
  switch(results.value)
  {
  //int x;
  case 0xFFA25D:
    Serial.println(" CH-            ");
    x=20;
    break;
  case 0xFF629D:
```

```
  Serial.println(" CH             ");
  x=20;
  break;
case 0xFFE21D:
  Serial.println(" CH+            ");
  x=20;
  break;
case 0xFF22DD:
  Serial.println(" PREV           ");
 x=20;
  break;
case 0xFF02FD:
  Serial.println(" NEXT           ");
  return 20;
  break;
case 0xFFC23D:
  Serial.println(" PLAY/PAUSE     ");
  x=20;
  break;
case 0xFFE01F:
  Serial.println(" VOL-           ");
  x=20;
  break;
case 0xFFA857:
  Serial.println(" VOL+           ");
  x=20;
  break;
case 0xFF906F:
  Serial.println(" EQ             ");
  x=20;
  break;
case 0xFF6897:
  Serial.println(" 0              ");
  x=0;
  break;
case 0xFF9867:
  Serial.println(" 100+           ");
  x=100;
  break;
case 0xFFB04F:
  Serial.println(" 200+           ");
  x=200;
  break;
```

```
  case 0xFF30CF:
    Serial.println(" 1             ");
    x=1;
    break;
  case 0xFF18E7:
    Serial.println(" 2             ");
    x=2;
    break;
  case 0xFF7A85:
    Serial.println(" 3             ");
    x=3;
    break;
  case 0xFF10EF:
    Serial.println(" 4             ");
    x=4;
    break;
  case 0xFF38C7:
    Serial.println(" 5             ");
    x=5;
    break;
  case 0xFF5AA5:
    x=6;
    Serial.println(" 6             ");
    break;
  case 0xFF42BD:
    x=7;
    Serial.println(" 7             ");
    break;
  case 0xFF4AB5:
    x=8;
    Serial.println(" 8             ");
    break;
  case 0xFF52AD:
    x=9;
    Serial.println(" 9             ");
    break;
  default:
    Serial.println(" other button   ");
    x=25;
  }
  delay(500);
  return x;
}
```

```
/*
void translateIR(int button){
Serial.print(results.value);Serial.print("\n");
  if(results.value == 0xFF6897 || button==1){
    // turn LED on:
    if(x==0){
      myservo.write(179);
      delay(1500);
      x=1;
    }
    else{
      myservo.write(0);
       delay(1500);
      x=0;
    }

  }
  }
  */
```

Results

Our final results were very positive. We were able to demonstrate our system accomplishing the appropriate motions based on signals from an IR remote and commands from a terminal. We were also able to demonstrate a working override button.

Future Work

One thing we were not able to accomplish on our project during the semester was collaborating with the other group who were working on the second part of the system (lock). We would integrate both systems for controlling the lights and controlling the lock into one microcontroller so that you would be able to control the lights and lock from one place.

Another thing we could work on for the future would be to fine tune the mechanical portions of the project. This means figuring out a mechanical attachment for the servo motor to make our system universal so that it can be mounted to any light and be able to turn the light on and off.

Conclusion

We were able to get the IR remote and sensor communicating properly with each other with a good working range. We were able to get the Bluetooth module working properly to communicate with terminal commands. We were able to get the servo motor to move to appropriate positions to turn lights on and off.

We were not able to get a proper mechanical system attached to the servo motor to control all lights. We did get a certain mechanism attached to the servo motor to control a specific light switch. However, when we tried the mechanism with other light switches, it would not work properly.

We learned a great deal working through this project. We mainly learned about how infrared communication works. We learned about how infrared signals are sent by the remote and picked up by the sensor. We learned how these signals are then decoded by a system which can then send out other signals based on the original IR signal. We learned that the IR sensor can be very voltage sensitive. If the sensor is not receiving a fairly constant voltage it can act like it is receiving wrong signals when it is receiving no signals at all. We also learned some things about Arduino code? We learned a number of things about bluetooth communication.