**Universal TV remote**

Milestone 8 Final Design Document

1. Latif Vaid, Miran Member, lvaid2@uic.edu, mmembe2@uic.edu
2. Universal TV remote
3. Abstract: Two arduino boards will act as a universal remote to communicate with any TV. What makes this project different is that we will have volume buttons, channel buttons, led screens.
4. Project Description:
   - 4.1. We will be utilizing the Infrared light sensors to control a TV. Some TVs use unique infrared which needs to be synced with the arduino for the arduino to communicate with the TV. In this project we will also utilize infrared receivers to see the unique infrared. Then transfer the data to the actual arduino that will emit the infrared to the tv.
   - 4.2. Subsystem:
     - 4.2.1. SubSystem 1: Name: Infrared Receiver Synchronization
       Work: Student 1 will take full responsibility to code and design the hardware so that a tv remote can send the IF signals and the arduino can catch the signals. The arduino board's job is to catch the signal and designate which button sent what signal. This will allow for the second arduino board to synchronize with the IF signals and work independently.
     - 4.2.2. Subsystem 2
       Name: TV Remote Arduino
       Work: Student 2 will have to design an arduino that will emit the Infrared Rays that will mimic that of the TV remote using the data provided from the first arduino. The first arduino will send the data using a serial connection.
   - 4.3. Input/Output:
     - 4.3.1. Subsystem 1:

| Device | Behavior | Input |
|---|---|---|
| Infrared Receiver | The sensor will detect the infrared rays of the remote since each remote uses different frequencies for their brand. We want the value of the Infared so subsystem 2 can use it. | Infrared waves |
| Buttons | There will be 2 buttons on the breadboard of the subsystem 1. If you hold any of the 2 buttons the receiver will be ready to receive the infrared waves and set it to that button. | Input Button |
| IR Remote | Any remote that has an infrared emitter will work such as TV | Infrared waves |

| | | |
|---|---|---|
| | remote, Led Remote, Fan Remote etc. | |

4.3.2. Subsystem 2:

| Device | Behavior | Input/Output |
|---|---|---|
| Infrared Emitter | After getting the values of the infrared waves from Subsystem 1 and saving them to their buttons in Subsystem 1. Subsystem 2 will copy the values of the infrared waves and assign them to 2 buttons on Subsystem 2. The buttons will perform the same actions that they received from the TV remote in Subsystem 1. | Infrared Waves (OUTPUT) |
| Buttons | There will be 2 buttons on the breadboard of the subsystem 2. If you hold any of the 2 buttons the Emitter will Emit the same waves fromSubsystem 1 Buttons. | Input Button (INPUT) |
| IR Device | Any device such as TV, LED, Fan etc that can receive the infrared waves from the arduino. | Infrared Waves (OUTPUT) |

4.4. Original Work:

Authors from DigiKey Electronics use radio frequency to communicate between two Arudionos while we will be using the serial communication to communicate with two subsystems.

Authors from Hackaday utilize Remoteduino, which is a special arduino designed to be used as an infrared remote control. While we will be using the Arduino Uno.

5.

| Weeks | Description |
|---|---|
| Week 12 (March 27 - April 2) | (Latif)For subsystem 1, make it so the arduino receives the infrared code and save it and parse it so it can be transferred to subsystem 2.<br>(Miran) Receive the infrared waves from serial communication from subsystem 1, and send the infrared rays. Make sure the Buttons work properly for the specific type of |

| | |
|---|---|
| | infrared rays and syncs for a specific TV remote function. |
| Week 13 (April 3 - April 9) | (Latif & Miran) Create the slides for Milestone 5 part 1.<br>Create a video presenting the slides for Milestone 5 part 1. |
| Week 14 (April 10 - April 16) | (Latif & Miran) Prepare mentally for the in-person presentation. Get the Arduino working. And finally present |
| Week 15 (April 17 - April 13) | Latif & Miran) Turn in Milestone 5 part 3. |
| Week 16 (April 24 - April 28) | Milestone 5 part 2(Expo)(4/22)<br>Milestone 5 part 3(Eval)(4/25)<br>Milestone 6 (4/27)<br>MIlestone 7 (4/28) |

6. References

https://github.com/Arduino-IRremote/Arduino-IRremote

Staff, M. (2019, April 22). How to wirelessly transmit data on Arduino. DigiKey Electronics - Electronic Components Distributor. Retrieved February 27, 2022, from https://www.digikey.com/en/maker/blogs/2019/how-to-wirelessly-transmit-data-on-arduino

Nardi, T. (2019, June 4). Simple arduino universal remote control. Hackaday. Retrieved February 27, 2022, from https://hackaday.com/2019/06/04/simple-arduino-universal-remote-control/
https://creately.com/lp/uml-diagram-tool/

7. List of Materials

SubSystem 1: Arduino Uno, Bread Board, Infrared Receiver, 2 Buttons, Breadboard Jumper wire, resisters, any infrared remote.

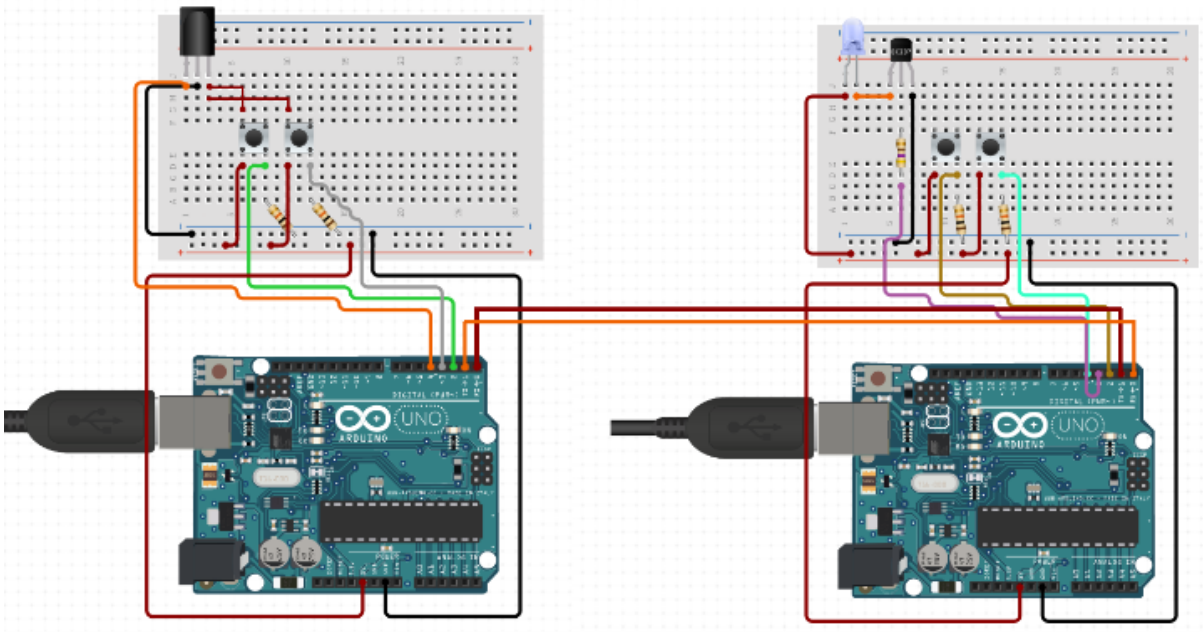
SubSystem 2: Arduino Uno, Bread Board, Infrared Emitter, 2 Buttons, Breadboard Jumper wire, resisters, any infrared device(TV, LED, FAN etc.).

8. Steps to build your project

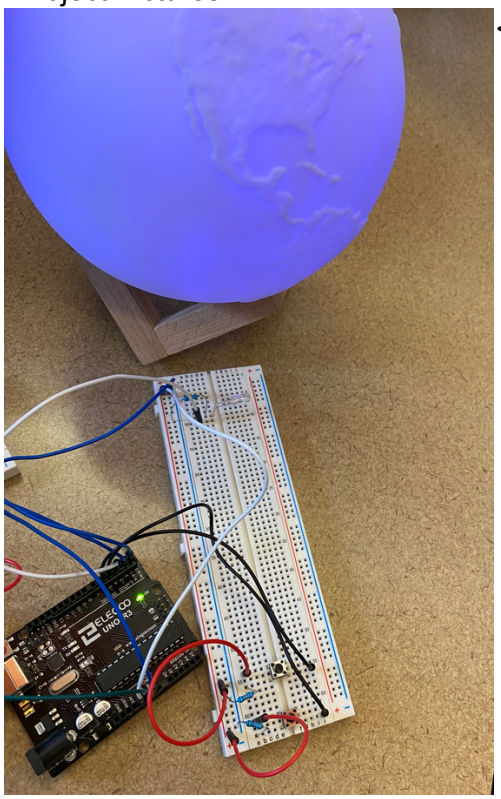
Using the Hardware Diagram set up the arduino and make sure all the wiring is correctly done, and the buttons are placed correctly. Subsystem one is for the infrared receiver which receives the signal and catches it. Subsystem two is an infrared emitter which sends the signal out, while making sure that both communicate correctly. For the code we used the provided links and IRremote.hpp library to receive or transmit infrared remote control codes. With that we obtain the codes for each button we are trying to

connect it to. Once the code is received it is then sent to set the frequency for the emitter to control the object. For example TV IR signals are decoded and signals are then used in the final application to emmit the corresponding infrared signals using IR Emitter

9. There will be 2 buttons on the board of the subsystem 2 which are to be used as any button selected on the remote. To do that you have to hold any of the 2 buttons the Emitter which will Emit the same waves from Subsystem 1 Buttons and can be used as it. The boards are both synced with each other.
10. Hardware Diagram:

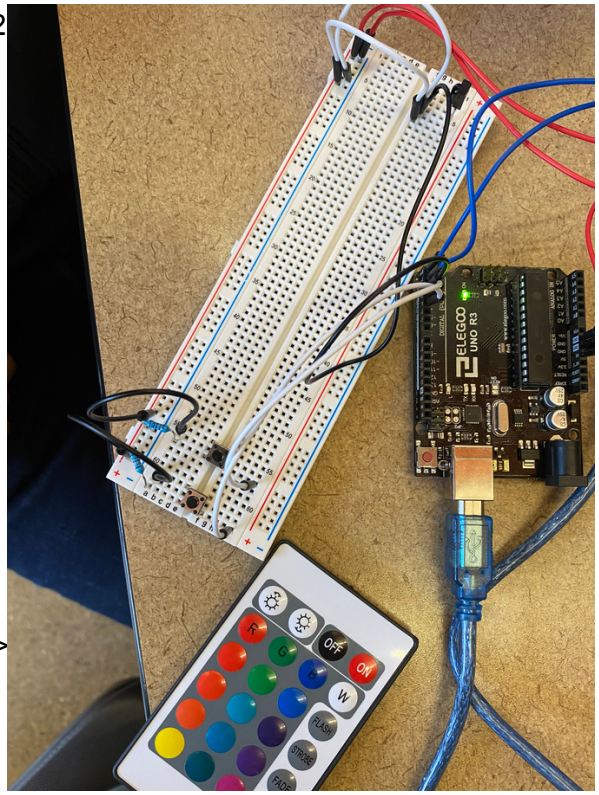


11. Project Pictures:



<- SubSystem 2

Subsystem 1 ->



**Universal TV remote**

12. Youtube Video Demonstration: https://youtu.be/1BPOnU7M7gU
13. Source Code:

---

SubSystem 1:

---

```
#include <IRremote.hpp>
uint8_t rawCode[RAW_BUFFER_LENGTH]; // The durations if raw
uint8_t rawCodeLength; // The length of the code

uint8_t rawCode1[RAW_BUFFER_LENGTH]; // The durations if raw
uint8_t rawCodeLength1; // The length of the code

int buttonState = 0;

void setup()
{
  IrReceiver.begin(2, ENABLE_LED_FEEDBACK); // Start the receiver
  Serial.begin(9600);
  pinMode(3, INPUT);
  pinMode(4, INPUT);
}
void loop() {
  if (digitalRead(4) == HIGH) {
    if (IrReceiver.decode()) {
      IrReceiver.printIRResultRawFormatted(&Serial, true); // Output the results in RAW
format
      rawCodeLength = IrReceiver.decodedIRData.rawDataPtr->rawlen - 1;
      IrReceiver.compensateAndStoreIRResultInArray(rawCode);
      IrReceiver.resume();
      for (int i = 0; i < rawCodeLength; i++) {
        Serial.println(rawCode[i]);
      }
      delay(500);
      Serial.write(1);
      Serial.write(rawCodeLength);
      delay(500);
      Serial.write(rawCode, rawCodeLength);
    }
  } else if (digitalRead(3) == HIGH) {
    if (IrReceiver.decode()) {
```

**Universal TV remote**

```
    IrReceiver.printIRResultRawFormatted(&Serial, true); // Output the results in RAW
format
    rawCodeLength1 = IrReceiver.decodedIRData.rawDataPtr->rawlen - 1;
    IrReceiver.compensateAndStoreIRResultInArray(rawCode1);
    IrReceiver.resume();
    for (int i = 0; i < rawCodeLength1; i++) {
      Serial.println(rawCode1[i]);
    }
    delay(500);
    Serial.write(2);
    Serial.write(rawCodeLength1);
    delay(500);
    Serial.write(rawCode1, rawCodeLength1);
   }
  }
}
```

---

SubSystem 2:

---

```
#include <IRremote.hpp>
uint8_t rawCode[RAW_BUFFER_LENGTH]; // The durations if raw
uint8_t rawCodeLength; // The length of the code

uint8_t rawCode1[RAW_BUFFER_LENGTH]; // The durations if raw
uint8_t rawCodeLength1; // The length of the code

//uint8_t buf[100] = {179, 87, 12, 32, 12, 10, 12, 10, 12, 11, 12, 10, 12, 10, 12, 10, 12, 10,
13, 9, 12, 32, 12, 32, 12, 32, 12, 33, 12, 32, 12, 32, 12, 32, 12, 32, 12, 32, 12, 10, 12, 11,
12, 10, 12, 10, 12, 10, 12, 10, 12, 11, 12, 10, 12, 32, 12, 32, 12, 32, 12, 33, 12, 32, 12,
32, 12, 32, 12, 33, 12, 10, 12, 10, 12, 10, 12, 11, 12, 10, 12, 10, 12, 10, 13, 10, 12, 32,
12, 32, 12, 33, 12, 32,12,32,12,33,12};

int incomingByte = 0;
void setup()
{
  IrSender.begin(3, ENABLE_LED_FEEDBACK);
  Serial.begin(9600);
  pinMode(4, INPUT);
  pinMode(5, INPUT);
}
void loop() {
   if (Serial.available() > 0) {
```

```
    incomingByte = Serial.read();
    Serial.print(incomingByte);
    Serial.println(" Button");
    if (incomingByte == 1) {
      incomingByte = Serial.read();
      rawCodeLength = incomingByte;
      Serial.print(rawCodeLength);
      Serial.println(" Length");
      Serial.readBytes(rawCode, rawCodeLength);
      for (int i = 0; i < rawCodeLength; i++) {
        Serial.println(rawCode[i]);
      }
      Serial.println(rawCodeLength);
    } else if (incomingByte == 2) {
      incomingByte = Serial.read();
      rawCodeLength1 = incomingByte;
      Serial.print(rawCodeLength1);
      Serial.println(" Length");
      Serial.readBytes(rawCode1, rawCodeLength1);
      for (int i = 0; i < rawCodeLength1; i++) {
        Serial.println(rawCode1[i]);
      }
      Serial.println(rawCodeLength1);
    }
  }
  if (digitalRead(4) == HIGH) {
    IrSender.sendRaw(rawCode, rawCodeLength, 38);
  } else if (digitalRead(5) == HIGH) {
    IrSender.sendRaw(rawCode1, rawCodeLength1, 38);
  }
}
```