Low Cost Communication Between Microcontrollers Using Infrared

ABSTRACT

Wireless communication between devices happens frequently in our daily lives. With my project, I investigated where infrared (IR) communication could be used to control microcontrollers wirelessly. The project required two types of microcontrollers: 1) the base station to send the commands, and 2) the receiver units that process the IR signal. A base station that can be controlled by a device with a user interface (e.g., computer, tablet) can send IR information to multiple receiver units that interpret the IR signal and perform some action (e.g., change the output color of a light) to what the user requested. Citing evidence from PixMob, I hypothesized this kind of communication setup would work, but my main research question was whether it would work on a smaller, cheaper scale. In my testing, I found that it was entirely possible to submit commands from a base station to manipulate lighting coming from multiple microcontrollers. While my solution proved to be a cheaper solution for wireless communication, it did have several shortcomings compared to other wireless communication methods. The range of IR communication in my testing was limited to under 20 feet, while options like Bluetooth and WIFI can stretch distances greater than 100 feet.

BACKGROUND

IR communication is a technology commonly associated with wirelessly controlling devices from across the room. When a button is pressed, the IR LED uses pulse width modulation to rapidly flash on and off to transmit a signal that is interpreted and decoded by the receiving device. Once the device has decoded the signal the desired action is executed. In a case study developed in my project, the LED ring lights up with the desired color or pattern. This type of technology can be used to control a range of low-cost devices. For example, this technology can be used to easily keep runners on pace during long distances races, or help get the crowd involved in sporting events and concerts.

GOAL

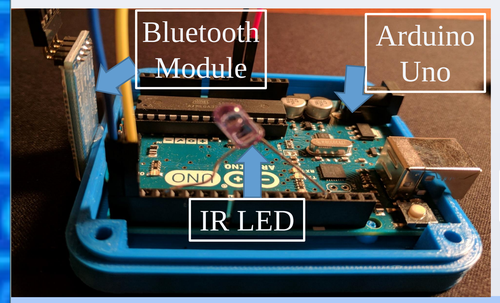
Create a low-cost communication system between microcontrollers using the existing IR protocol. Such a system will be controlled via an Android tablet that is connected over Bluetooth to the IR base station. The IR base station will receive instructions from the user and send commands using IR signals to the receiver units that will execute the desired effect.

In addition, I wanted to experiment with other technologies like 3d printing and app creation. The enclosures for the microcontrollers are all 3d printed using PLA filament. Control of the system is done through an app created using MIT's App Inventor software.

Screenshot of the app used to control the system: The app is divided into four sections: Solid color, Breathing effect, patterns, and a setup panel. The color selector's open a draw down where the user can pick from 8 colors and the arrows at the bottom of each section sends the selected command to the base station.

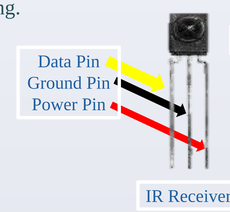
MATERIALS

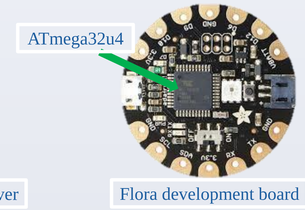
- Arduino Uno with IR LED
- Flora Microcontroller with IR receiver
- RGB LED ring with custom 3d printed housing
- Android tablet with Bluetooth
- Small coin cell or LI-PO battery to power receiver unit



METHODS AND PROCEDURES

1. Interpret IR signals coming from trusted sources, like IR-based remotes, and decode them using a microcontroller and the Arduino IR library.
2. Write code to execute certain commands (e.g., Change lighting patterns) when the correct IR signal is received by the receiving microcontroller.
3. Create new IR HEX codes that the base station can transmit to the receiver units.
4. Using Adafruit NeoPixels, write code that allows for multiple different lighting effects for the same receiving unit.
5. Shrink the receiving unit down to something that can easily be worn on the body or clothing.

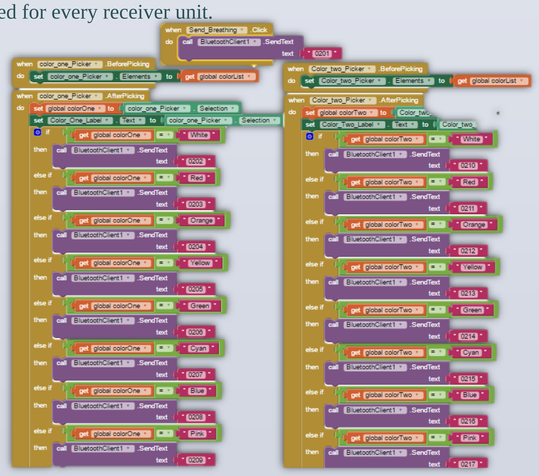




ANALYSIS

The project proved that it is possible to communicate between microcontrollers using IR light! Because IR does not make a client to client connection, with enough base stations a nearly infinite amount of receiver units could be controlled using this system. The limiting factors of this project is the short range, high cost per unit and limited additional features due to IR communication. Right now the range is now the operational range is no more than a few feet in any direction of the base station. The range is limited by the small current output from the Arduino. This could be solved with the addition of a transistor to boost the current of the IR LED. Each one of these prototype units cost around $25 dollars in components. Before something like this could be given to every attendee at a sporting event or concert, the cost would likely need to be well below $10. Future systems would likely need a custom printed circuit board to cut down on cost and time of manufacturing to be affordable enough to roll out in larger quantities. In the configuration that I have built for this project, the properties of IR communication limit the functionality of the system. One shortcoming is that the IR communication is only one way. This means that the base unit has no way to confirm that the command was received and executed by the receiver unit. Another limitation of IR communication is that peer to peer communication between the base station and an individual receiver unit is almost not possible due to its complexity. In order to communicate with a specific device a unique 32 bit IR code would have to be created for every receiver unit.

```
oraReciever §
void loop() {
  if (irrecv.decode(&results)) {
    Serial.println(results.value, HEX);
    irrecv.resume();

    ////////////////////////////////////Solid Colors


    if (results.value == 0xE0E020DF) {
      setAll(255, 255, 255);}//White

    else if (results.value == 0xE0E0A05F) {
      setAll(255, 0, 0);}//Red

    else if (results.value == 0xE0E0609F) {
      setAll(255, 128, 0);}//Orange

    else if (results.value == 0xE0E010EF) {
      setAll(255, 255, 0);}//Yellow

    else if (results.value == 0xE0E09067) {
      setAll(0, 255, 0);}//Green

    else if (results.value == 0xE0E050AF) {
      setAll(0, 255, 255);}//Cyan

    else if (results.value == 0xE0E030CF) {
      setAll(0, 0, 255);}//Blue

    else if (results.value == 0xE0E0B04F) {
      setAll(255, 0, 255);}//Pink

    else if (results.value == 0xE0E07689) {
      setAll(0, 0, 0);}//OFF
```



Left: Section of the code that controls the solid lighting on the receiver units. Each color has its specific IR HEX code to trigger it. For example, when "E0E020DF" is received, all of the pixels are turned on to make the ring shine white.

Middle: Section of code that controls the breathing lighting effect of the receiver units. After a specific color is chosen from the app, its corresponding Bluetooth code is sent over to the base unit where the code is interpreted and executed. For example, if the user selects Pink for their first color the app sends the text "0209" over to the base unit using Bluetooth.

Right: Prototype receiver unit with 3d printed housing. The front side (Top) contains the LED ring and IR receiver. The back (Bottom) holds the battery and Flora.

SAMPLE CODE

This section of code is running in a loop on the base station. The code is constantly checking for commands from the Bluetooth module. Once a command is received, it matches up with the If statement that has the correct Bluetooth code. It then sends out the IR code for that specific command. If the user wanted the lighting to be red the App would send the Bluetooth code "0102" to the base station. The base station would then send out the IR HEX code "E0E0A05F" using its IR LED.

```
noSender
#include <IRremote.h>

int light = 2;

String readString;
IRsend irsend;

void setup()
{
  Serial.begin(9600);
  pinMode(light,OUTPUT);
}

void loop() {
  while (Serial.available()) {
    delay(10);
    char c = Serial.read();
    readString += c;
  }

  //Solid Colors
  if (readString == "0101") { // Solid White
    irsend.sendNEC(0xE0E020DF, 32);
    }
  else if (readString == "0102") { // Solid Red
    irsend.sendNEC(0xE0E0A05F, 32);
    }
  else if (readString == "0103") { // Solid Orange
    irsend.sendNEC(0xE0E0609F, 32);
    }
  else if (readString == "0104") { // Solid Yellow
    irsend.sendNEC(0xE0E010EF, 32);
    }
  else if (readString == "0105") { // Solid Green
    irsend.sendNEC(0xE0E0906F, 32);
```

CONCLUSION

This research has demonstrated that using IR communication to command multiple microcontrollers is entirely possible with respect to certain limitations. Big companies like Disney and PixMob are investing lots of resources in designing systems that address most of the shortfalls mentioned in this project. Given its low cost and ease of use for users, IR based wireless communication systems are here to stay.

FUTURE RESEARCH

- Increasing range of IR transmission
- Creating a custom PCB receiver unit to dramatically reduce costs
- Change from a Bluetooth solution to a WIFI solution to allow for multiple base stations

REFERENCES

Ada, Lady. "IR Sensor." Using an IR Sensor | IR Sensor | Adafruit Learning System, 29 July 2012, learn.adafruit.com/ir-sensor/using-an-ir-sensor.

Griffiths, Dawn. Head First Android Development: a brain-friendly guide. S.l.: O'REILLY MEDIA, 2017. Print.

Koenig, Andrew, and Barbara E. Moo. Accelerated C : Practical Programming by Example. Boston, MA, Addison-Wesley, 2000.