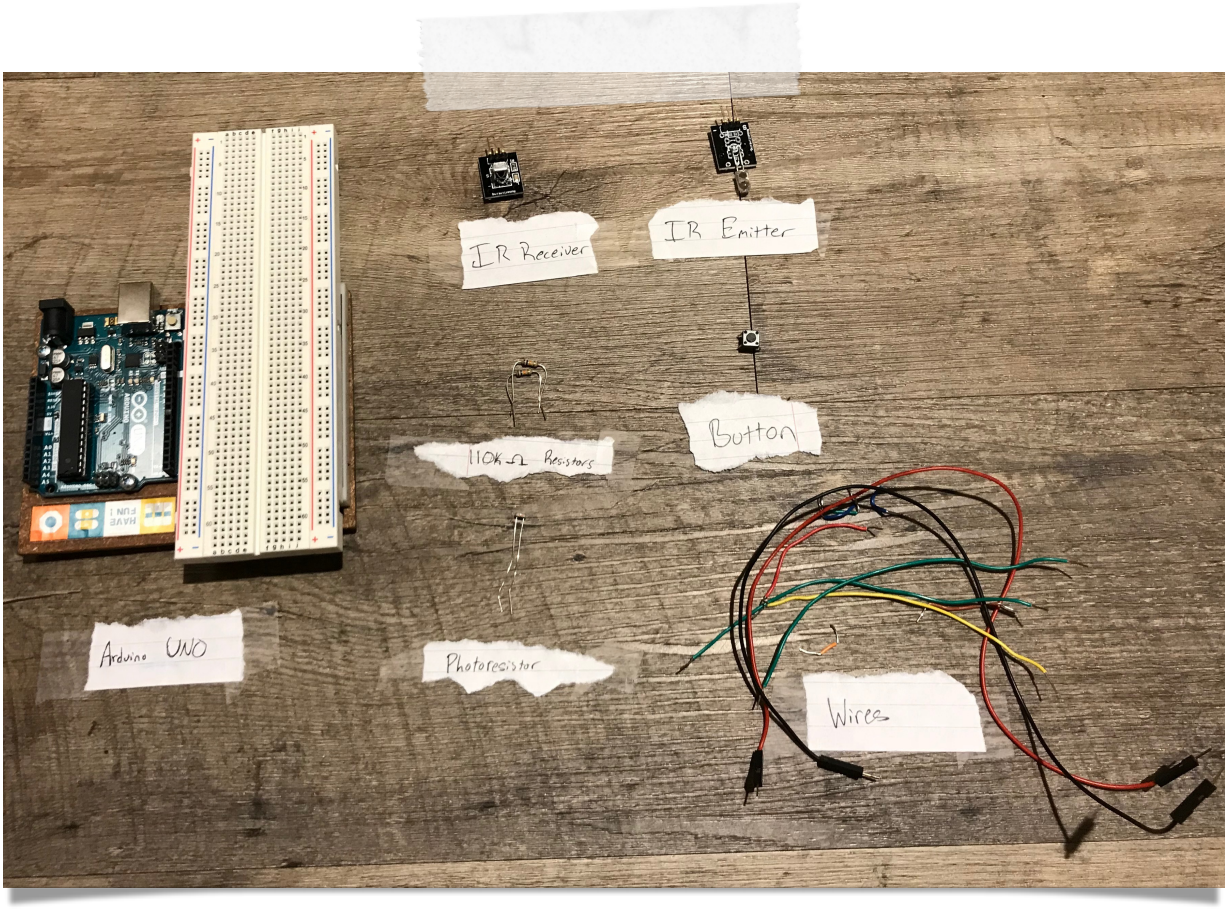
THE ‘NETFLIX & CHILL’ BUTTON

Computer Organization and Architecture



Project Write-up

G Leaden

Fall Semester 2017

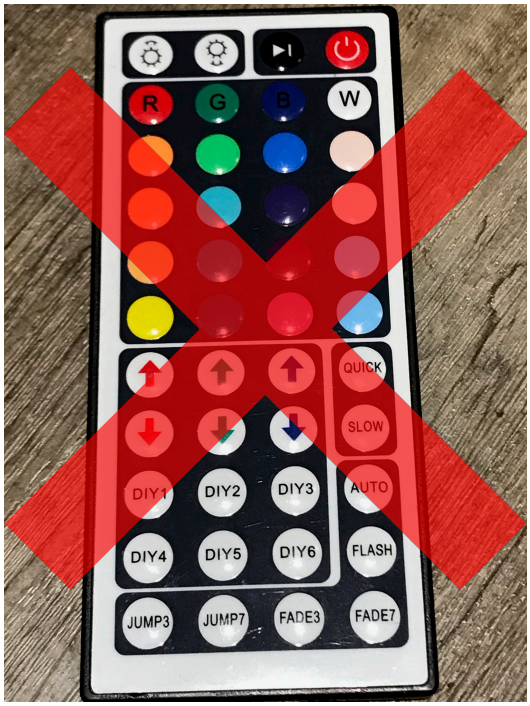
ABSTRACT

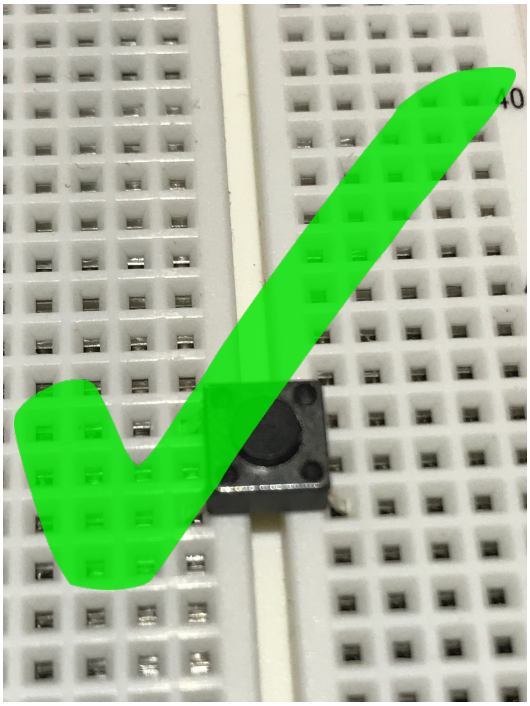
The ‘Netflix & Chill’ Button

The world today is filled with too many buttons, remotes, and controllers to get what you want. The amount of options available to us is commonly both empowering and overwhelming. This project, *The ‘Netflix & Chill’ Button* is meant to simplify. Through the use of infra-red (IR) receivers and emitters, the project seeks to create a One Button Experience™ for the user and eliminating the need for multiple remotes. Turning on the TV, turning on lights, changing the color of lights, opening Netflix, and adjusting lights based on current light levels has never been so easy.

INTRODUCTION

The Idea, The Solution, The Motivation





Idea

The idea behind this project was to eliminate the need for other, superfluous buttons and remotes and get right down to what people want. Netflix. In my house on campus my roommates and I only have the TV on when we are watching Netflix, and when we are watching Netflix we want a nice warm and inviting purple color coming from our lights to eliminate the need for the extremely aggressive daylight lights that are built into the house. To just utilize those two devices, the lights and the TV, we have two remotes and over 25 buttons to choose from! If we could eliminate the need to have more buttons and remotes than necessary to do the one thing we use our TV and lights for, that would be ideal.

Solution

There are several ways to approach this problem; given that we were given an Arduino and a sensor kit as part of the class, I decided to see if I could create a

solution using the tools available, nothing more. Everything was currently controlled by two IR remotes: one for the TV, and one for the lights. Our sensor kit provided me with all the tools I needed to get started: an IR receiver and an IR emitter. The IR receiver was first used to record the bits sent from the IR signal of the remote when a specific button was pressed. These signals were then recorded by myself to be used later when hardcoded into the emitter program. An example of what the received signal looks like can be seen in the Methodology section. The IR emitter was used to send previously recorded IR signals to the TV and lights respectively.

Motivation

The motivation for this project was twofold. First, I saw a void that needed filling. This device did not exist but would do nothing but benefit the immediate community. Second, I was inspired by Netflix's own, more robust, implementation of such a device called "The Switch"(1). The current design of The 'Netflix & Chill' Button device is not as effective as I would have liked, it requires very specific alignment with the two IR receivers for the TV and lights, but it is a proof of concept and I will have already laid the groundwork if I were to pursue a more refined version in the future.

METHODOLOGY

The Implementation, The Diagrams, The Struggles, The Awesome

Implementation

For this project, I needed four distinct I/O devices. An IR receiver, an IR emitter, a button, and a photo-resistor. The final "functional" part of the project shown in the video only has three of these four parts. The IR receiver was deemed unnecessary and removed once it had performed its function of recording the inputs from the various remotes. The IR receiver/emitter and their specific models were used due to their availability in the sensor kit provided, both of these devices were crucial to the proper communication between the completed project, and the TV and lights. The button was a generic pushbutton included in the Arduino kit

and was used as the human interface to the project, triggering a chain reaction of events that will be described below, when pressed. The photoresistor came with the Arduino kit and was added to aid in the control of the brightness of the lights, it will sense how the bright the room is, and the Arduino will compare that reading to a goal of what the photoresistor *should* read. In short, if it is too bright it will dim the lights, if it is too dark it will increase the brightness of the lights. The programs used (Send and Recieve) in the project are publicly available online on my GitHub (2) and referenced below.

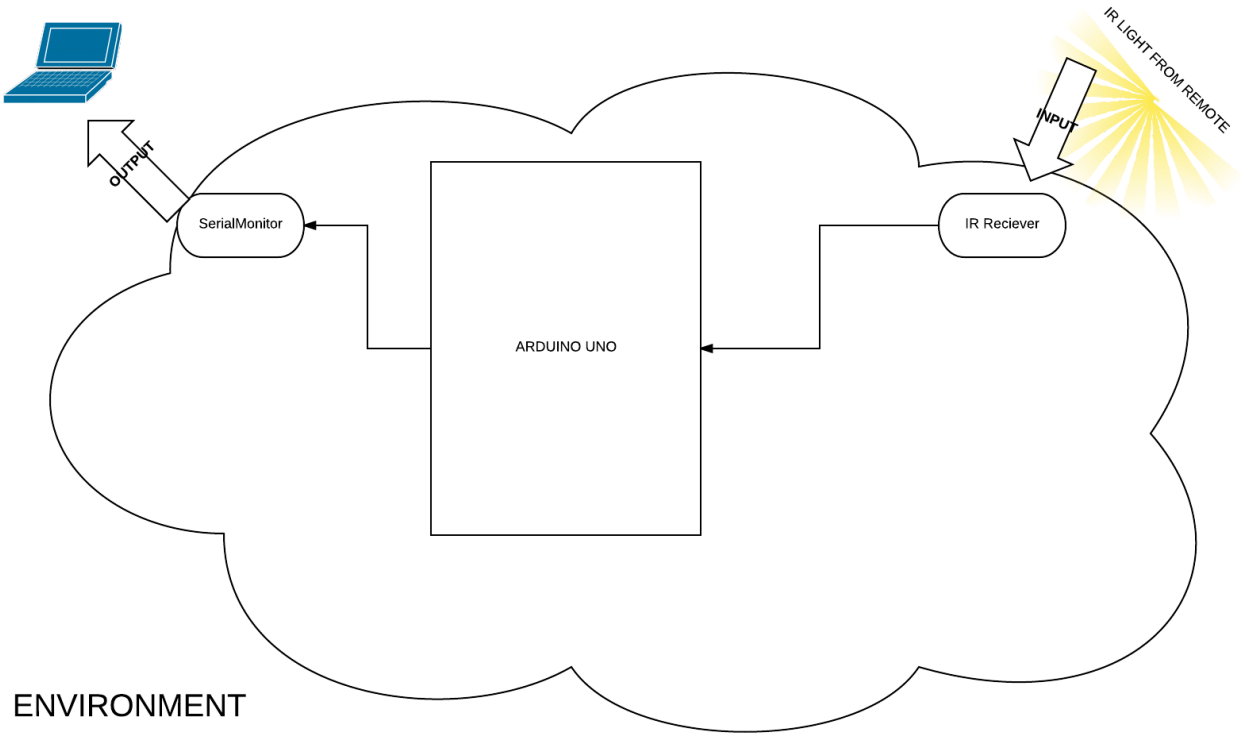
Below is an example of the readings received from the IR Receiver. The highlighted "Value: FF3AC5" portion is a code in hexadecimal that, once converted to Decimal, is given to the send function to mimic that signal as part of the IR library used (3).

```
Recieved!
Decoded NEC(1): Value:FF3AC5 Adrs:0 (32 bits)
Raw samples(68): Gap:38202
  Head: m8898  s4546
0:m490 s638     1:m466 s638             2:m494 s610    3:m514 s614
4:m494 s638     5:m490 s630             6:m494 s638    7:m470 s634
8:m494 s1762    9:m470 s1762            10:m490 s1738  11:m498 s1762
12:m466 s1762   13:m490 s1742            14:m494 s1762  15:m470 s1766

16:m490 s638    17:m466 s638            18:m490 s1762  19:m470 s1762
20:m494 s1742   21:m490 s634             22:m490 s1742  23:m494 s638
24:m486 s1742   25:m494 s1762            26:m470 s610   27:m518 s634
28:m490 s638    29:m470 s1762           30:m490 s642   31:m466 s1762

32:m490
Extent=67690
Mark  min:466    max:518
Space min:610    max:1766
```

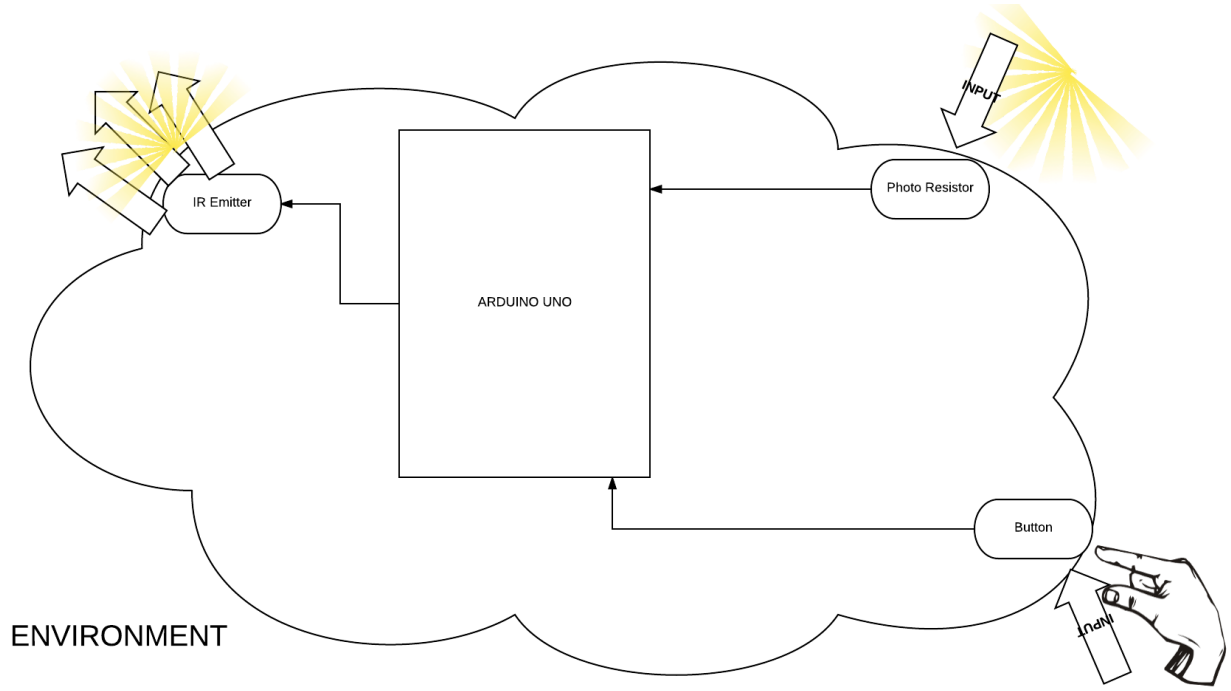
Diagrams

The architecture of the project is as follows:

Receive:

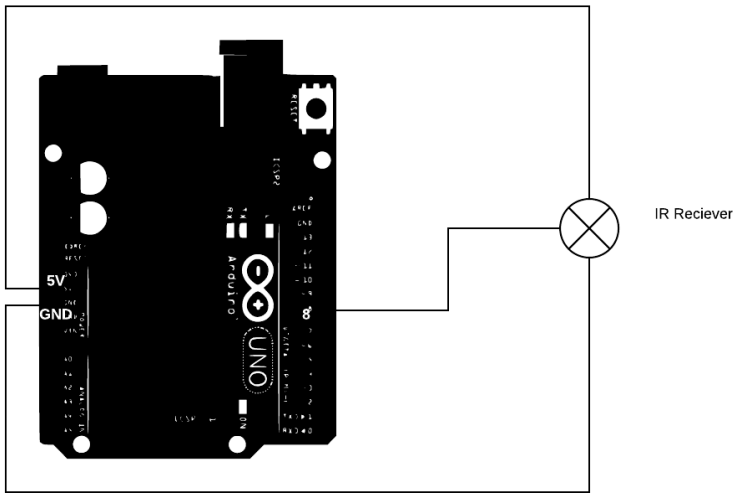


Send:



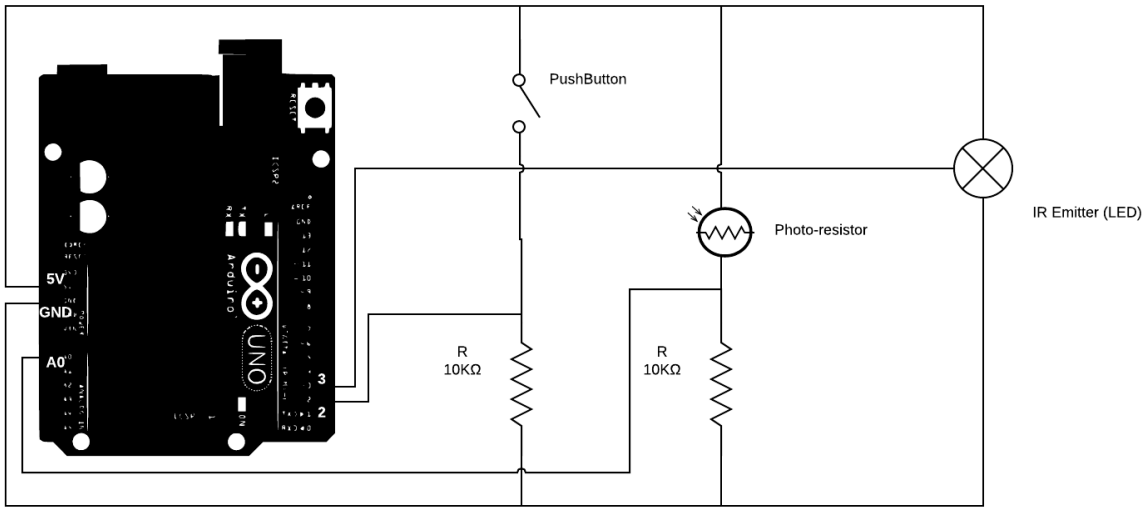
Here are the accompanying circuit diagrams:

Receive: The IR Receiver is just connected to digital pin 8, 5V Power, and Ground.



Send: The sending architecture is a bit more nuanced. The pushbutton is connected to a 10KΩ resistor connected to ground, digital pin 2, and 5V power. The Photo-resistor is connected to a 10KΩ resistor connected to ground, analog pin 0, and 5V power. Lastly, the IR Emitter (LED) is connected to digital pin 3, 5V power.

All diagrams created in Lucidchart (4).



Struggle

I ran into three main challenges during the course of this project. Firstly, the IR emitter does not have the largest range, so it was hard to get both IR devices (TV and lights) to read the signal from the emitter at the same time. I fixed this by careful placement of the Arduino project, as well as some tape on the light's IR sensor so it would not point downward.

Second, it was harder than I imagined to implement the photoresistor. I wired it just fine, but I was having trouble figuring out how to go about adjusting

the light levels in the room. My logic was off for the longest time because I was trying to take the current light level and use that as a target for the lights to achieve. I foolishly left the current light level variable inside the loop method, so the lights were constantly going up and down to adjust to the new target level. I resolved this issue by setting a target light level in the setup method prior to the loop. No matter the starting light level the project will attempt to be as close to the target light level as it can by adjusting the brightness of the lights via the IR emitter.

My third issue was one that presented itself fairly early in the development process. I had a pushbutton that only sent a HIGH signal to the digital pin while it was pressed down, but I did not want to have to press and hold the button during the entire loading processes. The Netflix and chill button was designed to be easier, this would not do. I resolved this issue by having a conditional in my loop that checked if the digital pin was HIGH and if it was it would increment a counter, if that counter was an odd number (it started at 0) the Arduino would run the process that powers on the tv, lights, colors the lights, and opens Netflix. If the counter was not odd it would continue looping until it was.

Awesome

My project is inspired by an April fools joke from Netflix one year. Although it does not order dominos for you (perhaps a future update?) it does have significant control over the entertainment system and is a novel concept. This project taught me a lot about IR signaling and although I didn't end up doing much besides the basics within the IRLibrary I used, I practically read through the entire documentation and learned a lot about the processes involved with many different devices.

REFERENCES

1. Netflix The Switch: http://makeit.netflix.com/projects/the-switch
2. GitHub: https://github.com/GLeaden/Netflix-Chill.git
3. IR Library: https://github.com/cyborg5/IRLib2
4. Lucidchart: https://www.lucidchart.com/
5. Arduino Projects Book edited by Scott Fitzgerald and Michael Shiloh: Pages 34, 54