# Morse Project tob31

## Introduction

For this assignment I was given a new transmission code known as Vidco which I had to provide a back-and-forth translation for using the serial port as well as using the led's on the Arduino shield hardware to transmit the messages as led. Finally, allowing for some of these messages to act as commands allowing the user to interact with the I/O devices on the shield.

## The code

### Part 1 Encoding and decoding vidco code

For this part of the assignment, I first built the translation functions char2vidco ascci2vidco vidco2char vidco2ascii. The vidco code was put into a char * array with a corresponding char array of its translated ascii characters to match. The char arrays use a simple for loop to match the 2 arrays together to retrieve the translation. The ascii functions then use the char functions in a for loop to build up the translated string message.

I then built the read input code into the void loop function so that it could call the translation functions needed for the program and then It would loop continuously, after the serial port receives the input message an if statements determines whether the input was ascii or vidco the program then calls the correct function and sets the correct pin mode.

### Part 2 Transmitting vidco code by LED

The send digital function takes the vidco message as an input as well as the correct pin for the led, loops through the vidco string and sets the led on and off for the correct time corresponding to its input character. The program also reads the value for the potentiometer with analogRead and adds its value to the delay, so that turning potentiometer on the Arduino shield will change the speed of the output led transmitted message. If acsii is entered into the program the translated vidco is sent as led and if vidco is entered into the program that message is then sent by led.

### Part 3 Respond to command messages

For this part because the specification said it should perform part 3 in addition to and after part 1 and 2. The commands sent on into the input are translated to vidco/ascii and sent as an led transmission before the specific command interactions occur. All three commands can be called by simply inputting the ascii or vidco for each command. "PR" reads the analogy value from the potentiometer and turns the red led on at the same intensity as the value read from the potentiometer. "IRRR" simply reads the value from the IR receiver using digital read and outputs whether it is high or low on the serial port determined using an if statement. "L" followed by 12 decimal digits. CharAT(0) is used to identify this command, then charAt is used again to separated the characters from the input string and put them back together I order, so that there is 4 separated strings for each led. These strings are then converted to Integers which is then used to set the intensity of the 4 leds lit up.

## Conclusion

In conclusion I completed the functional requirements for this project fully and should expect 85% for this work. I planned and finished this work in a timely manner to the best of my ability. As for

problems and complications I did find that the specifications for part 3 were confusing since if I entered a command to the program, I wouldn't want to receive the translation of the command I inputted as ascii vidco or led however the specification did say to include it. Also, for part 3 IRRR I seemed to only be able to receive "High" back from the IR. Even though I encountered these problems I still feel I have completed this project fully.