CS3514 Laboratory Session:

Name: Nathan Crowley

Student ID: 118429092

Question: Shift Registers

Repeat the following sections for each question

Answer:

****

Link to YouTube video showing program running on arduino:
https://www.youtube.com/watch?v=2nPNC0OrNHs

****

Part A:

Firstly set the **latchPin** to **pin8**, **clockPin** to **pin12**, **dataPin** to **pin11**. And set them all as OUTPUTs.

I also begin the Serial monitor with a data rate of 9600 bits/second. Aswell as initialising the IR reciever.

In my loop function I first check if there is data being recievered by the IR reciever. Then filter out the results by checking if it is **not recurring** (4294967295 is the result for holding down a button).

First write the latchPin LOW, followed by shfiting out the result to the **shfit register**. Then writing the latchPin HIGH following the shfitOut.

To get the **Least Significant byte** I mask the result with 0xFF to select only the final byte.

Part B:

To get the **Middle byte** I mask the result with 0xFF00 and shift left 8 bits to select only the middle byte.

Part C:

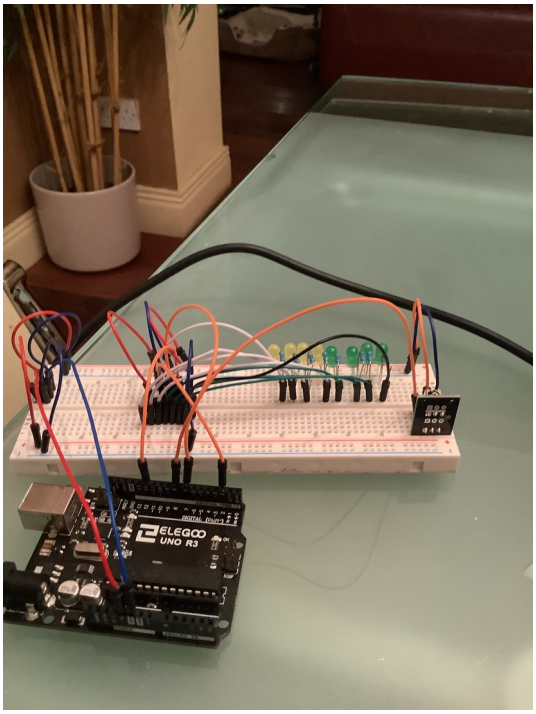
For myShiftOut function I first **define** the pd2 pin to 2 and set it as OUTPUT. Then in my function I loop through 8 bits and if the bit is 1 set pin2 HIGH, else if bit is 0 set pin2 LOW.

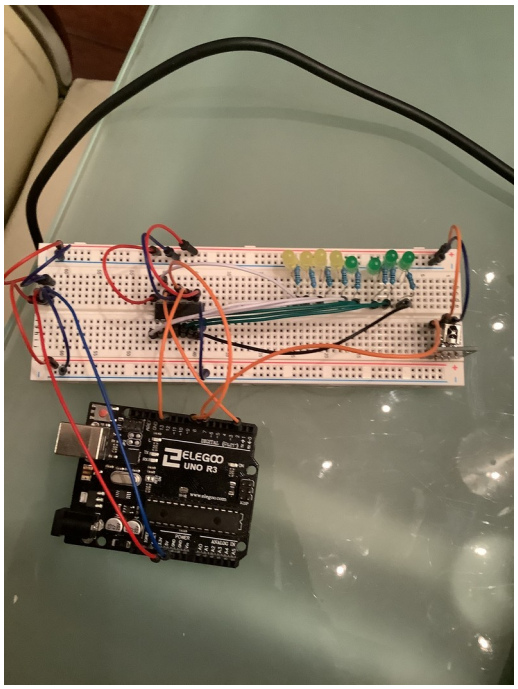
Code:

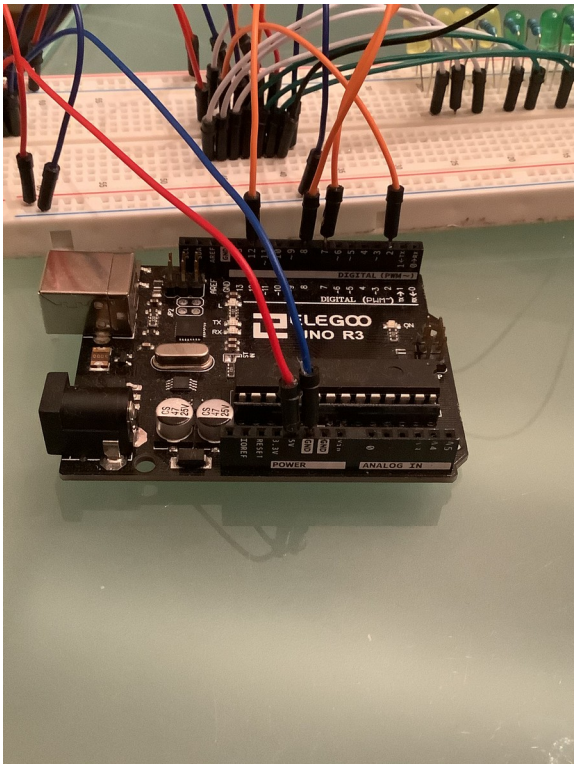
```
#include <IRremote.h>           //include infrared remote hander file.
int RECIERVER_PIN = 7;          //set the pin that is connected to IR
Reciever.
IRrecv irrecv(RECIERVER_PIN);
decode_results results;

//Pin connected to ST_CP of 74HC595
int latchPin = 8;
//Pin connected to SH_CP of 74HC595
int clockPin = 12;
////Pin connected to DS of 74HC595
int dataPin = 11;

#define pd2 2

void setup() {
  Serial.begin(9600);           //initalise the serial monitor with 9600 bits
per second data rate.
  irrecv.enableIRIn();          //initalise to recieve IR signals.
  pinMode(latchPin,OUTPUT);     //set all pins as OUTPUT for the main
loop.
  pinMode(clockPin,OUTPUT);
  pinMode(dataPin,OUTPUT);
  pinMode(pd2,OUTPUT);
}

byte LSBmask = 0xff;
void loop() {
  if(irrecv.decode(&results)){              //returns 0 if no data ready, 1 if
data ready.
    if(results.value != 4294967295){
/*PART A: */
      digitalWrite(latchPin, LOW);
      myShiftOut(clockPin, LSBFIRST, results.value&LSBmask);
      digitalWrite(latchPin, HIGH);
      /*Uncomment these to see the program running in the Serial
Moniotr.
      Serial.println(results.value,HEX);
      Serial.print("HEX Least significant Byte: ");
      Serial.println(results.value&LSBmask, HEX);
      Serial.println(results.value&LSBmask, BIN);*/
```

```
      delay(1000);
/* PART B*/
      digitalWrite(latchPin, LOW);
      myShiftOut(clockPin, LSBFIRST, (results.value&0xff00)>>8);
      digitalWrite(latchPin, HIGH);
      /* Uncomment these to see the program running in the Serial
Moniotr.
      Serial.println(results.value,HEX);
      Serial.print("HEX Middle Byte: ");
      Serial.println((results.value&0xff00)>>8, HEX);
      Serial.println((results.value&0xff00)>>8, BIN);*/
    }
   Serial.println();
   irecv.resume();
  }
}

/* PART C*/
void myShiftOut(uint8_t clockPin, uint8_t bitOrder, uint8_t val){
  uint8_t i;
  for(i = 0; i<8; i++){
    if(bitOrder == LSBFIRST){
      if(val&(1<<i)){   //check if the bit is 1
        PORTD &= B00000100;   //set pin 2 to 1
      }else{            //else its 0
        PORTD &= B00000000;   //set pin 2 to 0.
      }
    }else{
      if(val&(1<<(7-i))){
        PORTD &= B00000100;   //set pin 2 to 1
      }else{
        PORTD &= B00000000;   //set pin 2 to 0.
      }
    }
      digitalWrite(clockPin, HIGH);
      digitalWrite(clockPin, LOW);
  }
}
```

Photos:


ELEGOO
UNO R3


ELEGOO
UNO R3


DIGITAL (PWM~)
ELEGOO
UNO R3
POWER
ANALOG IN