MEMO

Drexel University

To: Dr. Christopher Peters, ECE 303 Fall 2020

From: Dinh Nguyen

Date: Oct 1st, 2020

Re: Lab 2 Timers and Interrupts Project – Codebreaker Project

Purpose

The lab is a simulation of a code breaker project where we are trying to crack a safety code with some 4-digit random number. Each digit will control one of the timers and LED and there is a total of limited 5 attempts to guess the code. Whenever the digit does not match, the speed of the LED will increase and will light up solidly if it reaches the maximum attempts. The lab goal is to enhance the experience programming with Arduino and understanding of timers and interrupts to produce the desired behavior of the program.

Discussion

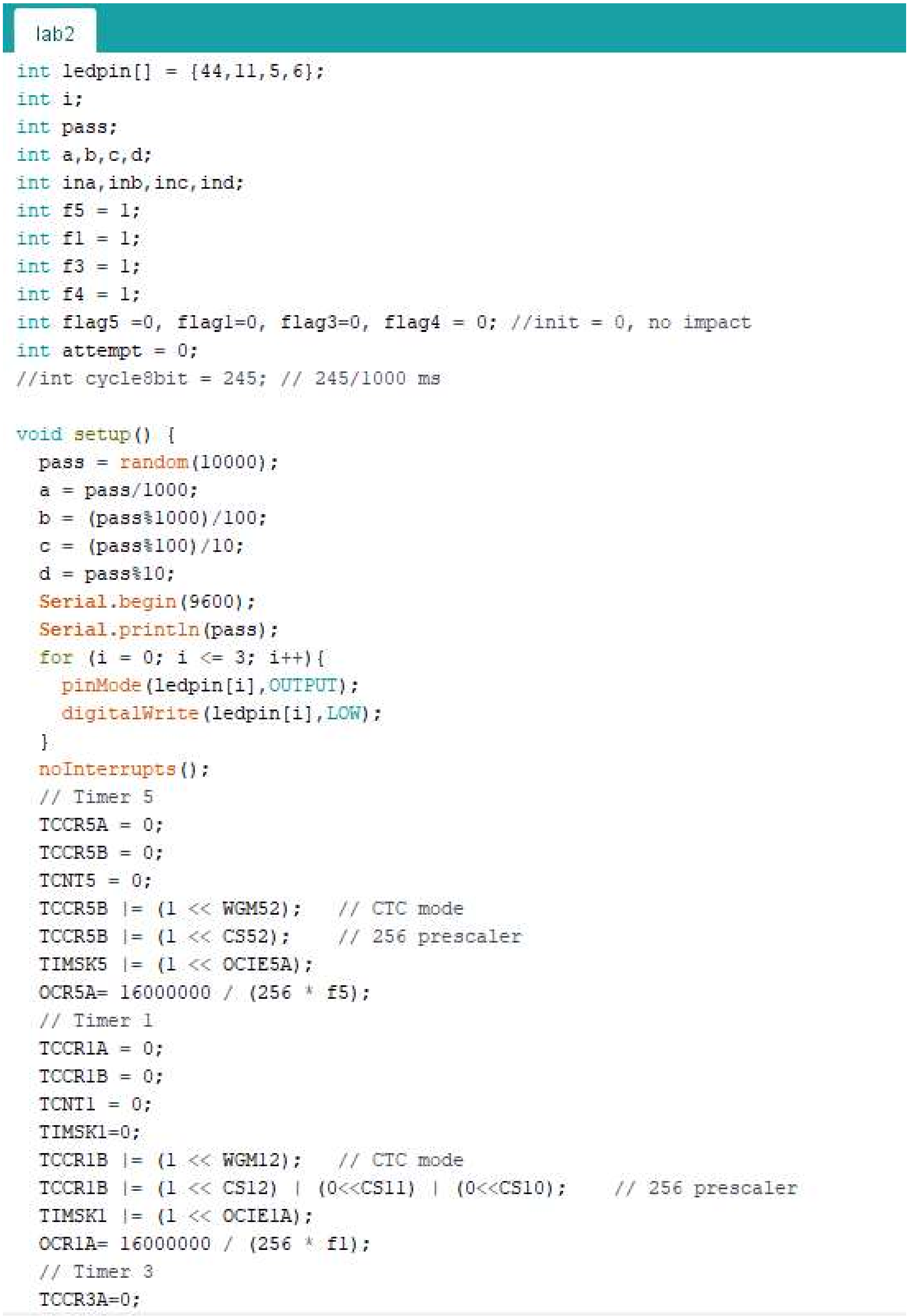
Hardware: Each LED will be connected in series with a resistor to ensure the low current through the LED. After that, the positive leg of LEDs will be connected to pin 44, 11, 5 and 6 where 16 bit timers are operating. The negative leg of LEDs will connect to ground.

Software: To generate a passcode, we can use function random() to get a number between 0000 and 9999. We will declare some global variables to control the frequency of the LEDs and the flags to control the behavior of the LED. If flag = 0, it will blink. If flag = 1, it will be off and it will HIGH when the flag = 2. Then, we set up 4 16-bit timers: Timer 5, Timer 1, Timer 3 and Timer 4 with the corresponding waveform, prescaler and OCR. In the loop function, we will get input from the user and compare the digits with the passcode. If they are correct, the flag variable will equal to 1 and the LED is off. If not, the flag will stay 0 and the LED will increase the speed of blinking by updating the OCR. The variable attempt also count how many attempts the users have tried and if it reach 5, all LEDs that are still blinking (flag=0) will be set to HIGH (flag=2). The last part is the interrupt function where the timers control the LED blinking behavior following the flag.

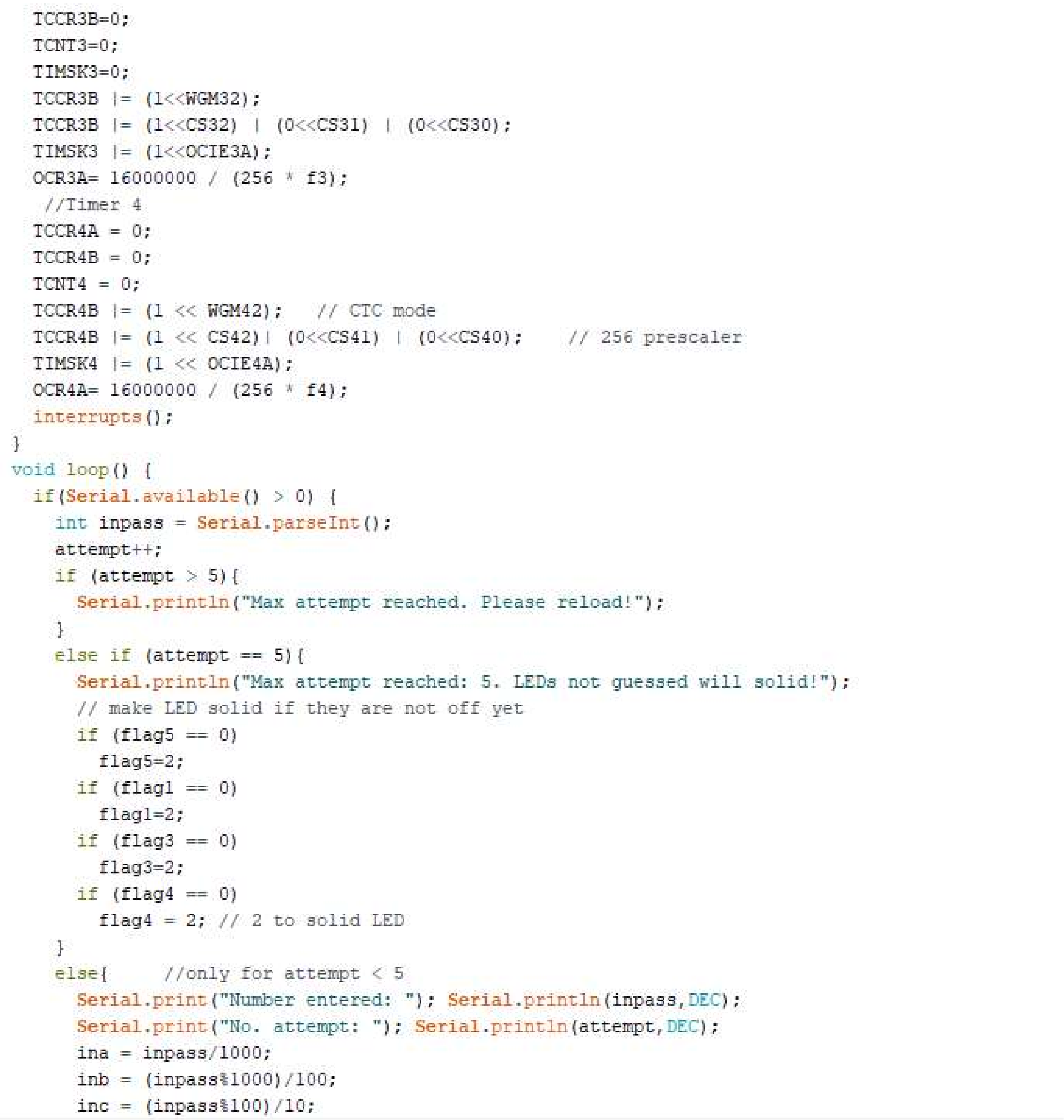
Recommendation

The lab provided a comprehensive study about timers and interrupt in Arduino similar to a lot of how other hardware in the industry work as well. However, the speed of the 8-bit timer is extremely fast that makes it impossible to see the blinking with human eyes. Thus, future research should be carried out to find out a way to implement the project with 8-bit timers. Nevertheless, the project in overall is a great learning source for future development of clock control between hardware and software.

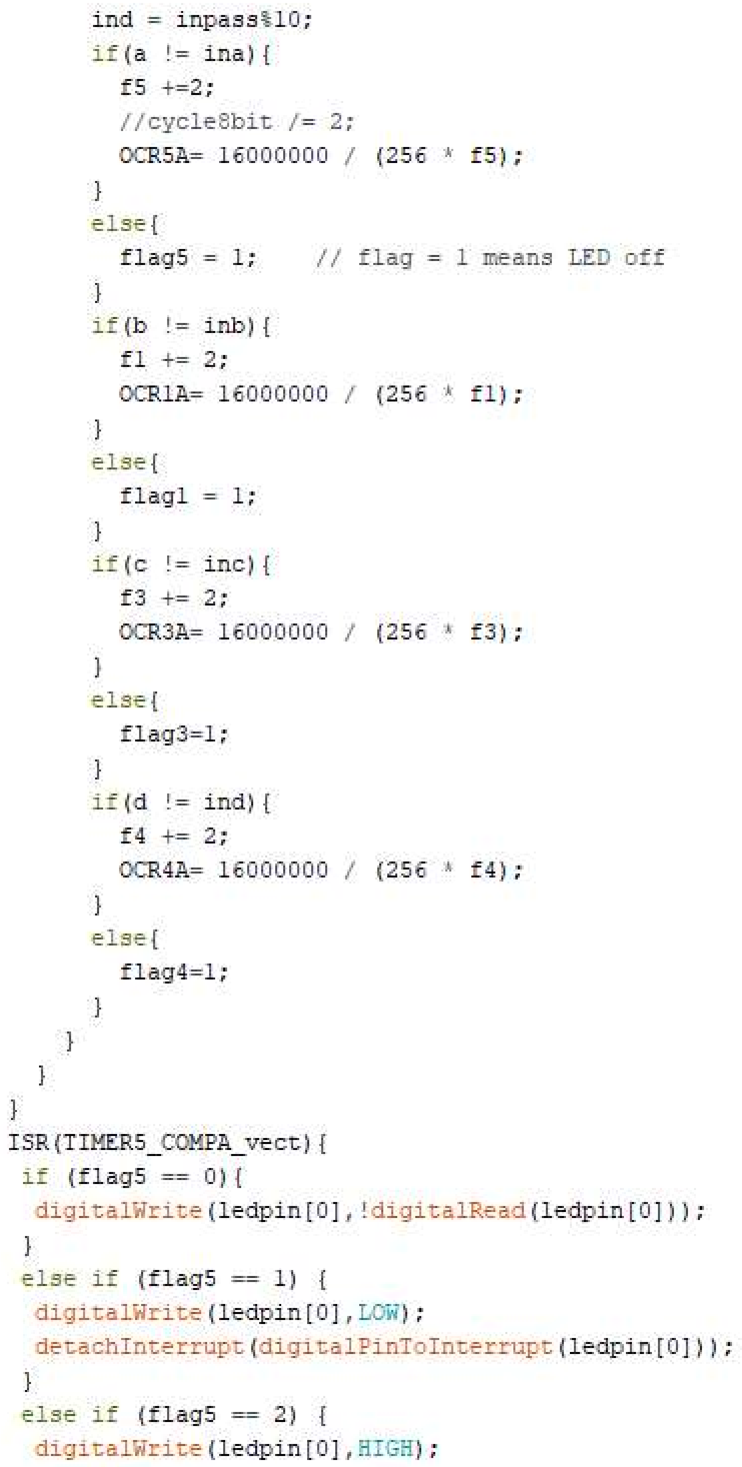
Arduino Sketch



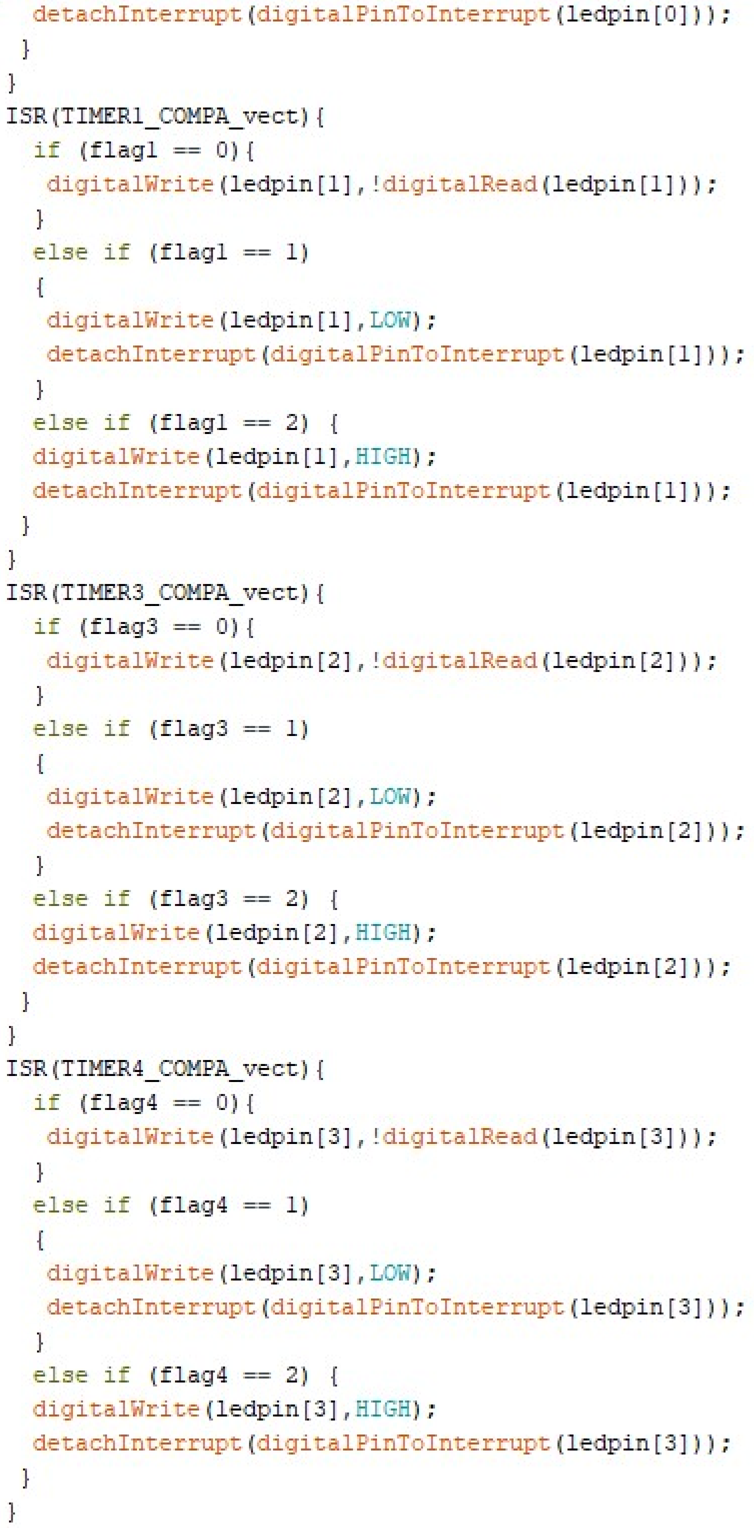
```
int ledpin[] = {44,11,5,6};
int i;
int pass;
int a,b,c,d;
int ina,inb,inc,ind;
int f5 = 1;
int f1 = 1;
int f3 = 1;
int f4 = 1;
int flag5 =0, flag1=0, flag3=0, flag4 = 0; //init = 0, no impact
int attempt = 0;
//int cycle8bit = 245; // 245/1000 ms

void setup() {
  pass = random(10000);
  a = pass/1000;
  b = (pass%1000)/100;
  c = (pass%100)/10;
  d = pass%10;
  Serial.begin(9600);
  Serial.println(pass);
  for (i = 0; i <= 3; i++){
    pinMode(ledpin[i],OUTPUT);
    digitalWrite(ledpin[i],LOW);
  }
  noInterrupts();
  // Timer 5
  TCCR5A = 0;
  TCCR5B = 0;
  TCNT5 = 0;
  TCCR5B |= (1 << WGM52);   // CTC mode
  TCCR5B |= (1 << CS52);    // 256 prescaler
  TIMSK5 |= (1 << OCIE5A);
  OCR5A= 16000000 / (256 * f5);
  // Timer 1
  TCCR1A = 0;
  TCCR1B = 0;
  TCNT1 = 0;
  TIMSK1=0;
  TCCR1B |= (1 << WGM12);   // CTC mode
  TCCR1B |= (1 << CS12) | (0<<CS11) | (0<<CS10);    // 256 prescaler
  TIMSK1 |= (1 << OCIE1A);
  OCR1A= 16000000 / (256 * f1);
  // Timer 3
  TCCR3A=0;
```



```
  TCCR3B=0;
  TCNT3=0;
  TIMSK3=0;
  TCCR3B |= (1<<WGM32);
  TCCR3B |= (1<<CS32) | (0<<CS31) | (0<<CS30);
  TIMSK3 |= (1<<OCIE3A);
  OCR3A= 16000000 / (256 * f3);
   //Timer 4
  TCCR4A = 0;
  TCCR4B = 0;
  TCNT4 = 0;
  TCCR4B |= (1 << WGM42);   // CTC mode
  TCCR4B |= (1 << CS42)| (0<<CS41) | (0<<CS40);    // 256 prescaler
  TIMSK4 |= (1 << OCIE4A);
  OCR4A= 16000000 / (256 * f4);
  interrupts();
}
void loop() {
  if(Serial.available() > 0) {
    int inpass = Serial.parseInt();
    attempt++;
    if (attempt > 5){
      Serial.println("Max attempt reached. Please reload!");
    }
    else if (attempt == 5){
      Serial.println("Max attempt reached: 5. LEDs not guessed will solid!");
      // make LED solid if they are not off yet
      if (flag5 == 0)
        flag5=2;
      if (flag1 == 0)
        flag1=2;
      if (flag3 == 0)
        flag3=2;
      if (flag4 == 0)
        flag4 = 2; // 2 to solid LED
    }
    else{     //only for attempt < 5
      Serial.print("Number entered: "); Serial.println(inpass,DEC);
      Serial.print("No. attempt: "); Serial.println(attempt,DEC);
      ina = inpass/1000;
      inb = (inpass%1000)/100;
      inc = (inpass%100)/10;
```



```
      ind = inpass%10;
      if(a != ina){
        f5 +=2;
        //cycle8bit /= 2;
        OCR5A= 16000000 / (256 * f5);
      }
      else{
        flag5 = 1;    // flag = 1 means LED off
      }
      if(b != inb){
        f1 += 2;
        OCR1A= 16000000 / (256 * f1);
      }
      else{
        flag1 = 1;
      }
      if(c != inc){
        f3 += 2;
        OCR3A= 16000000 / (256 * f3);
      }
      else{
        flag3=1;
      }
      if(d != ind){
        f4 += 2;
        OCR4A= 16000000 / (256 * f4);
      }
      else{
        flag4=1;
      }
    }
  }
}
ISR(TIMER5_COMPA_vect){
 if (flag5 == 0){
  digitalWrite(ledpin[0],!digitalRead(ledpin[0]));
 }
 else if (flag5 == 1) {
  digitalWrite(ledpin[0],LOW);
  detachInterrupt(digitalPinToInterrupt(ledpin[0]));
 }
 else if (flag5 == 2) {
  digitalWrite(ledpin[0],HIGH);
```



```
  detachInterrupt(digitalPinToInterrupt(ledpin[0]));
 }
}
ISR(TIMER1_COMPA_vect){
  if (flag1 == 0){
   digitalWrite(ledpin[1],!digitalRead(ledpin[1]));
  }
  else if (flag1 == 1)
  {
   digitalWrite(ledpin[1],LOW);
   detachInterrupt(digitalPinToInterrupt(ledpin[1]));
  }
  else if (flag1 == 2) {
  digitalWrite(ledpin[1],HIGH);
  detachInterrupt(digitalPinToInterrupt(ledpin[1]));
 }
}
ISR(TIMER3_COMPA_vect){
  if (flag3 == 0){
   digitalWrite(ledpin[2],!digitalRead(ledpin[2]));
  }
  else if (flag3 == 1)
  {
   digitalWrite(ledpin[2],LOW);
   detachInterrupt(digitalPinToInterrupt(ledpin[2]));
  }
  else if (flag3 == 2) {
  digitalWrite(ledpin[2],HIGH);
  detachInterrupt(digitalPinToInterrupt(ledpin[2]));
 }
}
ISR(TIMER4_COMPA_vect){
  if (flag4 == 0){
   digitalWrite(ledpin[3],!digitalRead(ledpin[3]));
  }
  else if (flag4 == 1)
  {
   digitalWrite(ledpin[3],LOW);
   detachInterrupt(digitalPinToInterrupt(ledpin[3]));
  }
  else if (flag4 == 2) {
  digitalWrite(ledpin[3],HIGH);
  detachInterrupt(digitalPinToInterrupt(ledpin[3]));
 }
}
```