

Homework #4

1. Read independent switch to control LEDs scrolling on/off blinking by external Interrupt.

*** Code:**

```
int interruptPin = 18;

int myLEDs[] = {12, 11, 10, 9, 8, 7, 6, 5};

int mySize = sizeof(myLEDs) / 2;

bool toLeft = true;

bool pressed = false;

int ledIndex = 0;

void setup() {
    Serial.begin(9600);

    for (int i = 0; i < mySize; i++) {
        pinMode(myLEDs[i], OUTPUT);
        digitalWrite(myLEDs[i], HIGH);
    }

    pinMode(interruptPin, INPUT);
    attachInterrupt(digitalPinToInterrupt(interruptPin), pause, RISING);
}

void loop() {
    if (!pressed) {
        blinking(50, 50);

        if (toLeft) {
            ledIndex++;
        }

        else {
```

```

    ledIndex--;
}
if (ledIndex == 0 || ledIndex == mySize - 1) {
    toLeft = !toLeft;
}
}
else {
    blinking(50, 1);
}
}

```

```

void blinking(int msOn, int msOff) {
    digitalWrite(myLEDs[ledIndex], LOW);
    delay(msOn);
    digitalWrite(myLEDs[ledIndex], HIGH);
    delay(msOff);
}

```

```

void pause() {
    delay(10000);
    pressed = !pressed;
}

```

* **Execution:** <https://goo.gl/oPhm5G>

2. Dynamically display 0000-9999 on 7-segment LED which is controlled start/stop by one switch.

* **Code:**

```

int digit[][2] = {{30, 1}, {31, 2}, {32, 3}, {33, 4}, {34, 5}, {35, 6}, {36, 7}};

```

```
int pinDs[] = {38, 39, 40, 41};
```

```
int mySize = sizeof(digit) / sizeof(digit[0]);
```

```
int dSize = sizeof(pinDs) / 2;
```

```
int interruptPin = 18;
```

```
int pin = 0, led = 0;
```

```
int counter = 0;
```

```
bool pause = false;
```

```
void setup() {
```

```
    Serial.begin(9600);
```

```
    for (int i = 0; i < mySize; i++) {
```

```
        //assign output pins and turn off all digits
```

```
        pinMode(digit[i][0], OUTPUT);
```

```
    }
```

```
    for (int i = 0; i < dSize; i++) {
```

```
        //assign output 4 Ds pins on 8051 7-segments
```

```
        pinMode(pinDs[i], OUTPUT);
```

```
    }
```

```
    pinMode(interruptPin, INPUT_PULLUP);
```

```
    attachInterrupt(digitalPinToInterrupt(interruptPin), pauseCounting, RISING);
```

```
}
```

```
void loop() {
```

```
if (!pause) {  
    counter++;  
    if (counter == 10000) {  
        counter = 0;  
    }  
}  
writeCounter();  
}
```

```
void writeCounter() {  
    int digit1 = 0, digit2 = 0, digit3 = 0, digit4 = 0;  
    digit1 = counter / 1000;  
    digit2 = (counter / 100) % 10;  
    digit3 = (counter / 10) % 10;  
    digit4 = counter % 10;  
    for (int i = 0; i < 10; i++) { //after 10 blinks only increase one number  
        writeDigits(digit1, digit2, digit3, digit4);  
    }  
}
```

```
void writeDigits(int digit1, int digit2, int digit3, int digit4) {  
    //quickly (2ms) turn on and off, write n3 to left most digit  
    for (int i = 0; i < dSize; i++) {  
        digitalWrite(pinDs[i], HIGH);  
    }  
    digitalWrite(pinDs[0], LOW);  
    for (int i = 0; i < mySize; i++) {
```

```
pin = digit[i][0];  
led = digit[i][1];  
writeNumbers(pin, led, digit1);  
}  
delay(2);
```

```
//quickly (2ms) turn on and off, write n2 to left second digit  
for (int i = 0; i < dSize; i++) {  
    digitalWrite(pinDs[i], HIGH);  
}  
digitalWrite(pinDs[1], LOW);  
for (int i = 0; i < mySize; i++) {  
    pin = digit[i][0];  
    led = digit[i][1];  
    writeNumbers(pin, led, digit2);  
}  
delay(2);
```

```
//quickly (2ms) turn on and off, write n1 to left third digit  
for (int i = 0; i < dSize; i++) {  
    digitalWrite(pinDs[i], HIGH);  
}  
digitalWrite(pinDs[2], LOW);  
for (int i = 0; i < mySize; i++) {  
    pin = digit[i][0];  
    led = digit[i][1];  
    writeNumbers(pin, led, digit3);
```

```

}

delay(2);

//quickly (2ms) turn on and off, write n to last digit
for (int i = 0; i < dSize; i++) {
    digitalWrite(pinDs[i], HIGH);
}
digitalWrite(pinDs[3], LOW);
for (int i = 0; i < mySize; i++) {
    pin = digit[i][0];
    led = digit[i][1];
    writeNumbers(pin, led, digit4);
}
delay(2);
}

```

```

void writeNumbers(int pin, int led, int num) {
    digitalWrite(pin, HIGH);
    switch (num) {
        case 0:
            //Write 0
            if (led == 1 || led == 2 || led == 3 || led == 4 || led == 5 || led == 6) {
                digitalWrite(pin, LOW);
            }
            break;
        case 1:
            //Write 1

```

```
    if (led == 2 || led == 3) {  
        digitalWrite(pin, LOW);  
    }  
    break;  
case 2:  
    //Write 2  
    if (led == 1 || led == 2 || led == 4 || led == 5 || led == 7) {  
        digitalWrite(pin, LOW);  
    }  
    break;  
case 3:  
    //Write 3  
    if (led == 1 || led == 2 || led == 3 || led == 4 || led == 7) {  
        digitalWrite(pin, LOW);  
    }  
    break;  
case 4:  
    //Write 4  
    if (led == 2 || led == 3 || led == 6 || led == 7) {  
        digitalWrite(pin, LOW);  
    }  
    break;  
case 5:  
    //Write 5  
    if (led == 1 || led == 3 || led == 4 || led == 6 || led == 7) {  
        digitalWrite(pin, LOW);  
    }  
}
```

```
        break;
    case 6:
        //Write 6
        if (led != 2) {
            digitalWrite(pin, LOW);
        }
        break;
    case 7:
        //Write 7
        if (led == 1 || led == 2 || led == 3) {
            digitalWrite(pin, LOW);
        }
        break;
    case 8:
        //Write 8
        digitalWrite(pin, LOW);
        break;
    case 9:
        //Write 9
        if (led != 5) {
            digitalWrite(pin, LOW);
        }
        break;
}

void pauseCounting() {
```



```
delay(10000);  
pause = !pause;  
}
```

* **Execution:** <https://goo.gl/T58C7K>

3. In question 2, please add 1 second delay to display from 0000 to 9999 by Timer0.

* **Code:**

```
#include <TimerOne.h>  
  
int digit[][2] = {{30, 1}, {31, 2}, {32, 3}, {33, 4}, {34, 5}, {35, 6}, {36, 7}};  
int pinDs[] = {38, 39, 40, 41};  
int mySize = sizeof(digit) / sizeof(digit[0]);  
int dSize = sizeof(pinDs) / 2;  
int pin = 0, led = 0;  
int counter = 0;  
void setup() {  
    Serial.begin(9600);  
    for (int i = 0; i < mySize; i++) {  
        //assign output pins and turn off all digits  
        pinMode(digit[i][0], OUTPUT);  
    }  
    for (int i = 0; i < dSize; i++) {  
        //assign output 4 Ds pins on 8051 7-segments  
        pinMode(pinDs[i], OUTPUT);  
    }  
    Timer1.initialize(1000000);//0.1 second  
    Timer1.attachInterrupt(numberCounting);  
}
```

```

void loop() {
}

void writeDigits(int digit1, int digit2, int digit3, int digit4) {
    //quickly (2ms) turn on and off, write n3 to left most digit
    for (int i = 0; i < dSize; i++) {
        digitalWrite(pinDs[i], HIGH);
    }
    digitalWrite(pinDs[0], LOW);
    for (int i = 0; i < mySize; i++) {
        pin = digit[i][0];
        led = digit[i][1];
        writeNumbers(pin, led, digit1);
    }
    delay(2);
    //quickly (2ms) turn on and off, write n2 to left second digit
    for (int i = 0; i < dSize; i++) {
        digitalWrite(pinDs[i], HIGH);
    }
    digitalWrite(pinDs[1], LOW);
    for (int i = 0; i < mySize; i++) {
        pin = digit[i][0];
        led = digit[i][1];
        writeNumbers(pin, led, digit2);
    }
    delay(2);
    //quickly (2ms) turn on and off, write n1 to left third digit
    for (int i = 0; i < dSize; i++) {

```

```

    digitalWrite(pinDs[i], HIGH);
}
digitalWrite(pinDs[2], LOW);
for (int i = 0; i < mySize; i++) {
    pin = digit[i][0];
    led = digit[i][1];
    writeNumbers(pin, led, digit3);
}
delay(2);
//quickly (2ms) turn on and off, write n to last digit
for (int i = 0; i < dSize; i++) {
    digitalWrite(pinDs[i], HIGH);
}
digitalWrite(pinDs[3], LOW);
for (int i = 0; i < mySize; i++) {
    pin = digit[i][0];
    led = digit[i][1];
    writeNumbers(pin, led, digit4);
}
delay(2);
}

void writeNumbers(int pin, int led, int num) {
    digitalWrite(pin, HIGH);
    switch (num) {
        case 0:
            //Write 0
            if (led == 1 || led == 2 || led == 3 || led == 4 || led == 5 || led == 6) {

```

```
    digitalWrite(pin, LOW);
}
break;
case 1:
    //Write 1
    if (led == 2 || led == 3) {
        digitalWrite(pin, LOW);
    }
    break;
case 2:
    //Write 2
    if (led == 1 || led == 2 || led == 4 || led == 5 || led == 7) {
        digitalWrite(pin, LOW);
    }
    break;
case 3:
    //Write 3
    if (led == 1 || led == 2 || led == 3 || led == 4 || led == 7) {
        digitalWrite(pin, LOW);
    }
    break;
case 4:
    //Write 4
    if (led == 2 || led == 3 || led == 6 || led == 7) {
        digitalWrite(pin, LOW);
    }
    break;
```

case 5:

```
//Write 5
```

```
if (led == 1 || led == 3 || led == 4 || led == 6 || led == 7) {
```

```
    digitalWrite(pin, LOW);
```

```
}
```

```
break;
```

case 6:

```
//Write 6
```

```
if (led != 2) {
```

```
    digitalWrite(pin, LOW);
```

```
}
```

```
break;
```

case 7:

```
//Write 7
```

```
if (led == 1 || led == 2 || led == 3) {
```

```
    digitalWrite(pin, LOW);
```

```
}
```

```
break;
```

case 8:

```
//Write 8
```

```
digitalWrite(pin, LOW);
```

```
break;
```

case 9:

```
//Write 9
```

```
if (led != 5) {
```

```
    digitalWrite(pin, LOW);
```

```
}
```

```

        break;
    }
}

void numberCounting() {
    if (counter < 10000) {
        counter++;
    }
    else {
        counter = 0;
    }

    int digit1 = 0, digit2 = 0, digit3 = 0, digit4 = 0;

    digit1 = counter / 1000;
    digit2 = (counter / 100) % 10;
    digit3 = (counter / 10) % 10;
    digit4 = counter % 10;

    for (int i = 0; i < 100; i++) {
        writeDigits(digit1, digit2, digit3, digit4);
    }
}

```

* **Execution:** <https://goo.gl/KMQcUy>

4. Scroll 8 LEDs blinking and monitor Keypad. Once one key is pressed, let the program go to interrupt service routine function to scan keypad and display key number on 7-segment LED. Otherwise, don't periodically scan keypad in the main function.

* **Code:**

```

//declare LEDs

int myLEDs[] = {12, 11, 10, 9, 8, 7, 6, 5};

int ledSize = sizeof(myLEDs) / 2;

bool toLeft = true;

```

```
int ledIndex = 0;
```

```
//declare 7-segments
```

```
int digit[][2] = {{30, 1}, {31, 2}, {32, 3}, {33, 4}, {34, 5}, {35, 6}, {36, 7}};
```

```
int pinDs[] = {38, 39, 40, 41};
```

```
int sevSegSize = sizeof(digit) / sizeof(digit[0]);
```

```
int dSize = sizeof(pinDs) / 2;
```

```
int pin = 0, led = 0;
```

```
int digit1 = 0, digit2 = 0;
```

```
//declare keypad buttons
```

```
int interruptPins[] = {18, 19, 20, 21};
```

```
int interruptSizes = sizeof(interruptPins) / 2;
```

```
bool pressed = false;
```

```
void setup() {
```

```
    Serial.begin(9600);
```

```
    //initialize LEDs
```

```
    for (int i = 0; i < ledSize; i++) {
```

```
        pinMode(myLEDs[i], OUTPUT);
```

```
        digitalWrite(myLEDs[i], HIGH);
```

```
    }
```

```
    //inititalize 7-segments
```

```
    for (int i = 0; i < sevSegSize; i++) {
```

```
        //assign output pins and turn off all digits
```



```
    digit1 = 1;
    digit2 = 9;
    break;
case 2:
    digit1 = 2;
    digit2 = 0;
    break;
case 3:
    digit1 = 2;
    digit2 = 1;
    break;
}
}
}
for (int i = 0; i < 10; i++) {
    blinking(5, 0);
    for (int i = 0; i < 2; i++) {
        writeDigits(digit1, digit2);
    }
}
}
else {
    turnOffSevSeg();
    blinking(50, 50);
    if (toLeft) {
        ledIndex++;
    }
}
```

```
    else {  
        ledIndex--;  
    }  
    if (ledIndex == 0 || ledIndex == ledSize - 1) {  
        toLeft = !toLeft;  
    }  
}  
}
```

```
void blinking(int msOn, int msOff) {  
    digitalWrite(myLEDs[ledIndex], LOW);  
    delay(msOn);  
    digitalWrite(myLEDs[ledIndex], HIGH);  
    delay(msOff);  
}
```

```
void buttonPress() {  
    delay(10000);  
    pressed = !pressed;  
}
```

```
void turnOffSevSeg() {  
    for (int i = 0; i < dSize; i++) {  
        digitalWrite(pinDs[i], HIGH);  
    }  
}
```

```

//determine which digit will be on
void writeDigits(int digit1, int digit2) {
    //quickly (2ms) turn on and off, write n3 to left most digit
    for (int i = 0; i < dSize; i++) {
        digitalWrite(pinDs[i], HIGH);
    }
    digitalWrite(pinDs[0], LOW);
    for (int i = 0; i < sevSegSize; i++) {
        pin = digit[i][0];
        led = digit[i][1];
        writeNumbers(pin, led, digit1);
    }
    delay(2);

    //quickly (2ms) turn on and off, write n2 to left second digit
    for (int i = 0; i < dSize; i++) {
        digitalWrite(pinDs[i], HIGH);
    }
    digitalWrite(pinDs[1], LOW);
    for (int i = 0; i < sevSegSize; i++) {
        pin = digit[i][0];
        led = digit[i][1];
        writeNumbers(pin, led, digit2);
    }
    delay(2);
}

```

```
//determine which segments will be on
void writeNumbers(int pin, int led, int num) {
    digitalWrite(pin, HIGH);
    switch (num) {
        case 0:
            //Write 0
            if (led == 1 || led == 2 || led == 3 || led == 4 || led == 5 || led == 6) {
                digitalWrite(pin, LOW);
            }
            break;
        case 1:
            //Write 1
            if (led == 2 || led == 3) {
                digitalWrite(pin, LOW);
            }
            break;
        case 2:
            //Write 2
            if (led == 1 || led == 2 || led == 4 || led == 5 || led == 7) {
                digitalWrite(pin, LOW);
            }
            break;
        case 3:
            //Write 3
            if (led == 1 || led == 2 || led == 3 || led == 4 || led == 7) {
                digitalWrite(pin, LOW);
            }
    }
```

```
    break;
case 4:
    //Write 4
    if (led == 2 || led == 3 || led == 6 || led == 7) {
        digitalWrite(pin, LOW);
    }
    break;
case 5:
    //Write 5
    if (led == 1 || led == 3 || led == 4 || led == 6 || led == 7) {
        digitalWrite(pin, LOW);
    }
    break;
case 6:
    //Write 6
    if (led != 2) {
        digitalWrite(pin, LOW);
    }
    break;
case 7:
    //Write 7
    if (led == 1 || led == 2 || led == 3) {
        digitalWrite(pin, LOW);
    }
    break;
case 8:
    //Write 8
```

```
    digitalWrite(pin, LOW);  
    break;  
case 9:  
    //Write 9  
    if (led != 5) {  
        digitalWrite(pin, LOW);  
    }  
    break;  
}  
}
```

* **Execution:** <https://goo.gl/kBShaR>