

# Homework #3

1. Read an independent switch to control LEDs scrolling blinking on/off.

\* **Code:**

```
int buttonPin = 13;
```

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

```
int mySize = sizeof(myLEDs)/2;
```

```
bool toLeft = true;
```

```
bool on = false;
```

```
int index = 0;
```

```
int buttonState;
```

```
int lastButtonState = LOW;
```

```
unsigned long lastDebounceTime = 0;
```

```
unsigned long debounceDelay = 50;
```

```
void setup() {
```

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

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

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

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

```
    }
```

```
    pinMode(buttonPin, INPUT);
```

```
}
```

```
void loop() {
```

```
int reading = digitalRead(buttonPin);
```

```
if (reading != lastButtonState) {  
    lastDebounceTime = millis();  
}
```

```
if ((millis() - lastDebounceTime) > debounceDelay) {  
    if (reading != buttonState) {  
        buttonState = reading;
```

```
        if(buttonState==HIGH) {  
            on = !on;  
        }  
    }  
}
```

```
if(on==false){  
    digitalWrite(myLEDs[index], LOW);  
}
```

```
    blinking();
```

```
    lastButtonState = reading;  
}
```

```
void blinking(){  
    if(toLeft==true){
```

```

toLeft = false;
for(int i=0; i<mySize; i++){
    if(on==true){
        digitalWrite(myLEDs[index], HIGH);
        digitalWrite(myLEDs[i], LOW);
        delay(50);
        digitalWrite(myLEDs[i], HIGH);
        delay(50);
        index = i;
    }
}
}
else{
    toLeft = true;
    for(int i=mySize-2; i>0; i--){
        if(on==true){
            digitalWrite(myLEDs[index], HIGH);
            digitalWrite(myLEDs[i], LOW);
            delay(50);
            digitalWrite(myLEDs[i], HIGH);
            delay(50);
            index = i;
        }
    }
}
}

```

\* **Execution:** <https://goo.gl/pdsTgs>

**2. Design scanning dual keys pressing that you select in different row and column on keypad and display a character defined by you for this pressing on 7-Segment LEDs.**

**\* Code:**

```
int digit[][2] = {{30, 1}, {31, 2}, {32, 4}, {33, 10}, {34, 9}, {35, 7}, {36, 6}};
```

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

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

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

```
int btnPins[] = {12, 13};
```

```
const int btnSize = sizeof(btnPins)/2;
```

```
int onStatus[btnSize];
```

```
int btnState[btnSize];
```

```
int lastBtnState[btnSize];
```

```
int reading[btnSize];
```

```
unsigned long lastDebounceTime = 0;
```

```
unsigned long debounceDelay = 50;
```

```
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);
```

```
        digitalWrite(digit[i][0], HIGH);
```

```
}
```

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

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

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

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

```
}
```

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

```
    //assign buttons pins, lastBtnState & onStatus for debounce
```

```
    pinMode(btnPins[i], INPUT);
```

```
    lastBtnState[i] = LOW;
```

```
    onStatus[i] = 0;
```

```
}
```

```
}
```

```
void loop() {
```

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

```
        reading[i] = digitalRead(btnPins[i]);
```

```
        if (reading[i] != lastBtnState[i]) {
```

```
            lastDebounceTime = millis();
```

```
        }
```

```
        if ((millis() - lastDebounceTime) > debounceDelay) {
```

```
            if (reading[i] != btnState[i]) {
```

```
                btnState[i] = reading[i];
```

```

    if(btnState[i]==LOW) {
        onStatus[i] = !onStatus[i];
    }
}
}
}
}

```

```

for(int i=0; i<btnSize; i++){
    if(onStatus[i]){
        switch(i){
            case 0: // 1st button, write A
                digitalWrite(pinDs[0], HIGH);
                for(int i=0; i<mySize; i++){
                    digitalWrite(digit[i][0], HIGH);
                }
                digitalWrite(pinDs[0], LOW);
                for(int i=0; i<mySize; i++){
                    if(digit[i][1]==1 || digit[i][1]==2 || digit[i][1]==4 || digit[i][1]==9 || digit[i][1]==7 ||
digit[i][1]==6){
                        digitalWrite(digit[i][0], LOW);
                    }
                }
                break;
            case 1: // 2nd button, write b
                digitalWrite(pinDs[0], HIGH);
                for(int i=0; i<mySize; i++){
                    digitalWrite(digit[i][0], HIGH);

```

```

    }
    digitalWrite(pinDs[0], LOW);
    for(int i=0; i<mySize; i++){
        if(digit[i][1]==4 || digit[i][1]==10 || digit[i][1]==9 || digit[i][1]==7 || digit[i][1]==6){
            digitalWrite(digit[i][0], LOW);
        }
    }
    break;
}
}
else{//2nd press in each button, turn off all
    digitalWrite(pinDs[0], HIGH);
    for(int i=0; i<mySize; i++){
        digitalWrite(digit[i][0], HIGH);
    }
}
lastBtnState[i] = reading[i];
}
}

```

\* **Execution:** <https://goo.gl/oDZgTd>

**3. (Bonus) Design single key scanning calculator (Calculation Arrange: 0+9; define +, -, x, / , = and square keys and display on 7-Segment LEDs).**

\* **Code:**

```
int digit[][2] = {{30, 1}, {31, 2}, {32, 4}, {33, 10}, {34, 9}, {35, 7}, {36, 6}};
```

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

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

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

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

```
const int btnSize = sizeof(btnPins)/2;
```

```
int num1=-1;
```

```
int num2=-1;
```

```
char oper;
```

```
int result;
```

```
bool resultShown = 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);
```

```
        digitalWrite(digit[i][0], HIGH);
```

```
    }
```

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

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

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

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

```
    }
```

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



```

//assign buttons pins, lastBtnState & onStatus for debounce
pinMode(btnPins[i], INPUT_PULLUP);
}

attachInterrupt(digitalPinToInterrupt(btnPins[0]), write1, CHANGE);
attachInterrupt(digitalPinToInterrupt(btnPins[1]), writeP, CHANGE);
attachInterrupt(digitalPinToInterrupt(btnPins[2]), write2, CHANGE);
attachInterrupt(digitalPinToInterrupt(btnPins[3]), writeResult, CHANGE);
}

void loop() {

}

void write1(){
  if(resultShown){
    num1 = num2 = -1;
    resultShown = false;
  }
  if(num1== -1)
    num1 = 1;
  else
    num2 = 1;
  digitalWrite(pinDs[0], HIGH);
  for(int i=0; i<mySize; i++){
    digitalWrite(digit[i][0], HIGH);
  }
}

```

```

digitalWrite(pinDs[0], LOW);
for(int i=0; i<mySize; i++){//write 1
    if(digit[i][1]==2 || digit[i][1]==4){
        digitalWrite(digit[i][0], LOW);
    }
}
}

```

```

void writeP(){
    oper = '+';
    digitalWrite(pinDs[0], HIGH);
    for(int i=0; i<mySize; i++){
        digitalWrite(digit[i][0], HIGH);
    }
    digitalWrite(pinDs[0], LOW);
    for(int i=0; i<mySize; i++){//write P
        if(digit[i][1]==1 || digit[i][1]==2 || digit[i][1]==9 || digit[i][1]==7 || digit[i][1]==6){
            digitalWrite(digit[i][0], LOW);
        }
    }
}

```

```

void write2(){
    if(resultShown){
        num1 = num2 = -1;
        resultShown = false;
    }
}

```

```

if(num1== -1)
    num1 = 2;
else
    num2 = 2;
digitalWrite(pinDs[0], HIGH);
for(int i=0; i<mySize; i++){
    digitalWrite(digit[i][0], HIGH);
}
digitalWrite(pinDs[0], LOW);
for(int i=0; i<mySize; i++){//write 2
    if(digit[i][1]==1 || digit[i][1]==2 || digit[i][1]==10 || digit[i][1]==9 || digit[i][1]==6){
        digitalWrite(digit[i][0], LOW);
    }
}
}
}

```

```

void writeResult(){
    Serial.println("num1");
    Serial.println(num1);
    Serial.println("num2");
    Serial.println(num2);
    if(oper=='+')
        result = num1 + num2;
    digitalWrite(pinDs[0], HIGH);
    for(int i=0; i<mySize; i++){
        digitalWrite(digit[i][0], HIGH);
    }
}

```

```

digitalWrite(pinDs[0], LOW);
Serial.println("result");
Serial.println(result);
switch(result){
    case 2:
        digitalWrite(pinDs[0], LOW);
        for(int i=0; i<mySize; i++){//write 2
            if(digit[i][1]==1 || digit[i][1]==2 || digit[i][1]==10 || digit[i][1]==9 || digit[i][1]==6){
                digitalWrite(digit[i][0], LOW);
            }
        }
        break;
    case 3:
        for(int i=0; i<mySize; i++){//write 3
            if(digit[i][1]==1 || digit[i][1]==2 || digit[i][1]==4 || digit[i][1]==10 || digit[i][1]==6){
                digitalWrite(digit[i][0], LOW);
            }
        }
        break;
    case 4:
        for(int i=0; i<mySize; i++){//write 3
            if(digit[i][1]==2 || digit[i][1]==4 || digit[i][1]==7 || digit[i][1]==6){
                digitalWrite(digit[i][0], LOW);
            }
        }
        break;
}

```

```
resultShown = true;  
}
```

\* **Execution:** <https://goo.gl/ykUrDi>