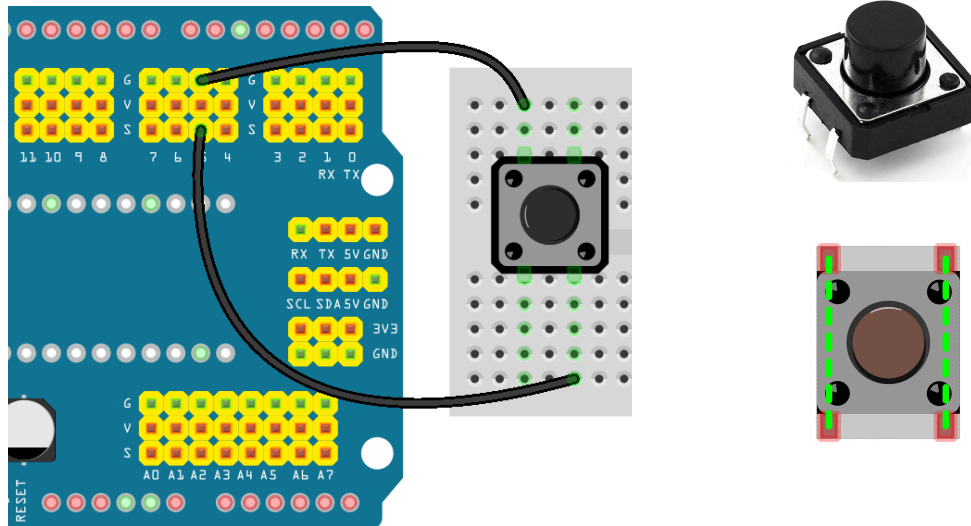


Exercise no 2: Digital inputs

Task 1. Connect the circuit as shown in the picture:



Code example:

```
#define BAUDRATE 115200
#define BUTTON1_PIN 5
int buttonstate;

void setup() {
  pinMode(BUTTON1_PIN, INPUT_PULLUP);
  Serial.begin(BAUDRATE); }

void loop() {
  buttonstate = digitalRead(BUTTON1_PIN);
  if(buttonstate == LOW)
    Serial.println("Button is down");
  else
    Serial.println("Button is up");
  delay(250); }
```

What's new:

digitalRead function, ***if*** statement, ***int*** type variable

Task 2. Using the circuit from Task 1, write a program that controls the built-in LED. The LED should be turned on if the button connected to DIO 5 is pressed.

```
#define BAUDRATE 115200
#define BUTTON1_PIN 5
int buttonstate;

void setup() {
    pinMode(LED_BUILTIN,OUTPUT);
    pinMode(BUTTON1_PIN,INPUT_PULLUP);
    Serial.begin(BAUDRATE); }

void loop() {
    buttonstate = digitalRead(BUTTON1_PIN);
    if(buttonstate == LOW) {
        digitalWrite(LED_BUILTIN,HIGH);
        Serial.println("Button is down"); }
    else {
        digitalWrite(LED_BUILTIN,LOW);
        Serial.println("Button is up"); }
    delay(250);
}
```

Task 3. Write a program that will blink (turn on and then turn off) the built-in LED 5 times, every time the DIO5 button is pressed.

```
#define BAUDRATE 115200
#define BUTTON1_PIN 5
int buttonstate;

void setup() {
```

Exercise no 2: Digital inputs

```
pinMode(LED_BUILTIN, OUTPUT);
pinMode(BUTTON1_PIN, INPUT_PULLUP);
Serial.begin(BAUDRATE); }

void loop() {
    buttonstate = digitalRead(BUTTON1_PIN);
    if(buttonstate == LOW)
        for(int i=0; i<5; i++) {
            Serial.println(i);
            digitalWrite(LED_BUILTIN, HIGH);
            delay(500);
            digitalWrite(LED_BUILTIN, LOW);
            delay(500); }
        delay(250);
}
```

What's new:

for expression, increment operator

Task 4. Using the code from the previous task, create a function that allows the user to define a number and period of blinks.

```
#define BAUDRATE 115200
#define BUTTON1_PIN 5
int buttonstate;

void setup() {
    pinMode(LED_BUILTIN, OUTPUT);
    pinMode(BUTTON1_PIN, INPUT_PULLUP);
    Serial.begin(BAUDRATE);
}
```

Exercise no 2: Digital inputs

```
void loop() {
  buttonstate = digitalRead(BUTTON1_PIN);
  if(buttonstate == LOW)
    led_blink(5,1000);
  delay(250);
}

void led_blink(int times,int duration) {
  for(int i=0;i<times;i++) {
    Serial.println(i);
    digitalWrite(LED_BUILTIN,HIGH);
    delay(duration/2);
    digitalWrite(LED_BUILTIN,LOW);
    delay(duration/2); }
}
```

What's new:

void function

Task 5. Write a program that works in the following way - if the DIO5 button is pressed, the built-in LED should be turned off, if the button is not pressed, the LED should be turned on.

```
#define BUTTON1_PIN 5

void setup() {
  pinMode(LED_BUILTIN,OUTPUT);
  pinMode(BUTTON1_PIN,INPUT_PULLUP); }

void loop() {
  digitalWrite(LED_BUILTIN,digitalRead(BUTTON1_PIN)); }
```

Task 6. Write a program that switches off the built-in LED, after 4 seconds from when the user releases the button.

Solution #1:

```
#define BUTTON1_PIN 5

int buttonstate;

void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(BUTTON1_PIN, INPUT_PULLUP);
  digitalWrite(LED_BUILTIN, HIGH);
}

void loop() {
  buttonstate = digitalRead(BUTTON1_PIN);
  if(buttonstate == LOW) {
    delay(100);
    while(buttonstate == LOW) {
      buttonstate = digitalRead(BUTTON1_PIN);
      delay(100); }
    delay(4000);
    digitalWrite(LED_BUILTIN, LOW);
    while(1);
  }
}
```

Solution #2:

```
#define BUTTON1_PIN 5

void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(BUTTON1_PIN, INPUT_PULLUP);
```

Exercise no 2: Digital inputs

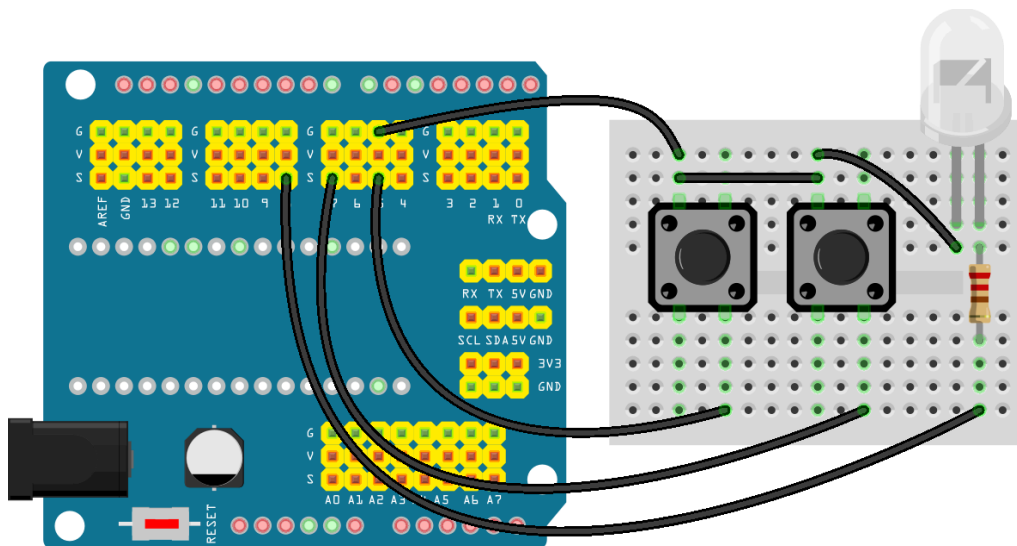
```
digitalWrite(LED_BUILTIN,HIGH); }

void loop() {
  if(digitalRead(BUTTON1_PIN) == LOW) {
    delay(100);
    while(digitalRead(BUTTON1_PIN) == LOW)
      delay(100);
    delay(4000);
    digitalWrite(LED_BUILTIN,LOW);
    while(1); } }
```

What's new:

while expression

Task 7. Connect the circuit as shown in the picture. Write a program that works as follows - one button should turn on the LED, and the other should turn it off.

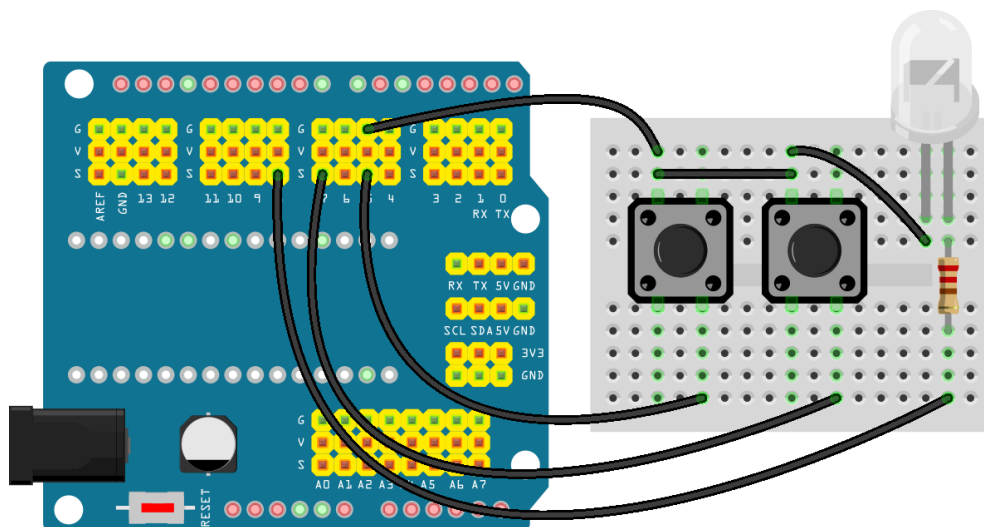


```
#define BUTTON1_PIN 5
#define BUTTON2_PIN 7
#define LED_PIN 8
```

Exercise no 2: Digital inputs

```
void setup() {  
  pinMode(LED_PIN, OUTPUT);  
  pinMode(BUTTON1_PIN, INPUT_PULLUP);  
  pinMode(BUTTON2_PIN, INPUT_PULLUP);  
  digitalWrite(LED_PIN, LOW); }  
  
void loop() {  
  if(digitalRead(BUTTON1_PIN) == LOW) {  
    digitalWrite(LED_PIN, HIGH);  
    delay(100);  
    while(digitalRead(BUTTON1_PIN) == LOW);  
    while(digitalRead(BUTTON2_PIN) == HIGH);  
    digitalWrite(LED_PIN, LOW);  
  }  
  delay(100);  
}
```

Task 8. Built a prototype of a device that controls the state of an LED with one button.



Exercise no 2: Digital inputs

```
#define BUTTON_PIN 5
#define LED_PIN 8
bool led_state=LOW;

void setup() {
    pinMode(LED_PIN,OUTPUT);
    pinMode(BUTTON_PIN,INPUT_PULLUP);
    digitalWrite(LED_PIN,led_state); }

void loop() {
    if(button_pressed(BUTTON_PIN)) {
        led_state = !led_state;
        digitalWrite(LED_PIN,led_state);
    }
    delay(100);
}

bool button_pressed(int button_pin) {
    if(digitalRead(button_pin) == LOW) {
        delay(100);
        while(digitalRead(button_pin) == LOW);
        return true;
    }
    else
        return false;
}
```

Task 9. Built a prototype of a device that measures the user's reaction time.

Hint:

```
#define BAUDRATE 115200
unsigned long timer;
```



```
void setup() {  
  Serial.begin(BAUDRATE);  
  Serial.println(timer); }  
  
void loop() {  
  timer = millis();  
  Serial.println(timer);  
  delay(500); }
```

What's new:

millis() function, ***unsigned long*** data type.

Function ***millis()*** returns the number of milliseconds passed since the Arduino board began running the program. This number will overflow (go back to zero), after approximately 50 days.

Reference:

www.arduino.cc/reference/en/language/functions/time/millis/

Task 10. Built a prototype of a device that blinks 3 LEDs. The first LED should blink once per second, the second LED twice, and the third LED three times per second. Using *millis()* function is mandatory.

Task 11. Built a human reaction time meter.

For those interested:

1. Basics of Tactile Switches:

components.omron.com/us-en/ds_related_pdf/A293-E1.pdf

2. The debounce debacle:

www.edn.com/the-debounce-debacle/

3. Pull-up Resistors:

learn.sparkfun.com/tutorials/pull-up-resistors

4. Video - What is Switch Bounce and How to Debounce

www.youtube.com/watch?v=IvU8m_30iK0&t=111s

Solution for task 11

```
#define BAUDRATE 9600
#define BUTTON1_PIN 7
int buttonstate;
int start_time;
int result;

void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(BUTTON1_PIN, INPUT_PULLUP);
  Serial.begin(BAUDRATE);
  digitalWrite(LED_BUILTIN, HIGH); }

void loop() {
  digitalWrite(LED_BUILTIN, HIGH);
  Serial.println("Reaction time meter");
  Serial.println("Press the button after the LED is off.");
  delay(4000);
  buttonstate = digitalRead(BUTTON1_PIN);
  while(buttonstate != HIGH) {
    Serial.println("Please remove your finger from the button");
    buttonstate = digitalRead(BUTTON1_PIN);
    delay(1000); }
  digitalWrite(LED_BUILTIN, LOW);
  start_time = millis();
  buttonstate = digitalRead(BUTTON1_PIN);
  while(buttonstate == HIGH)
    buttonstate = digitalRead(BUTTON1_PIN);
  result = millis() - start_time;
  Serial.print("Result: ");
  Serial.println(result);
  while(1); }
```