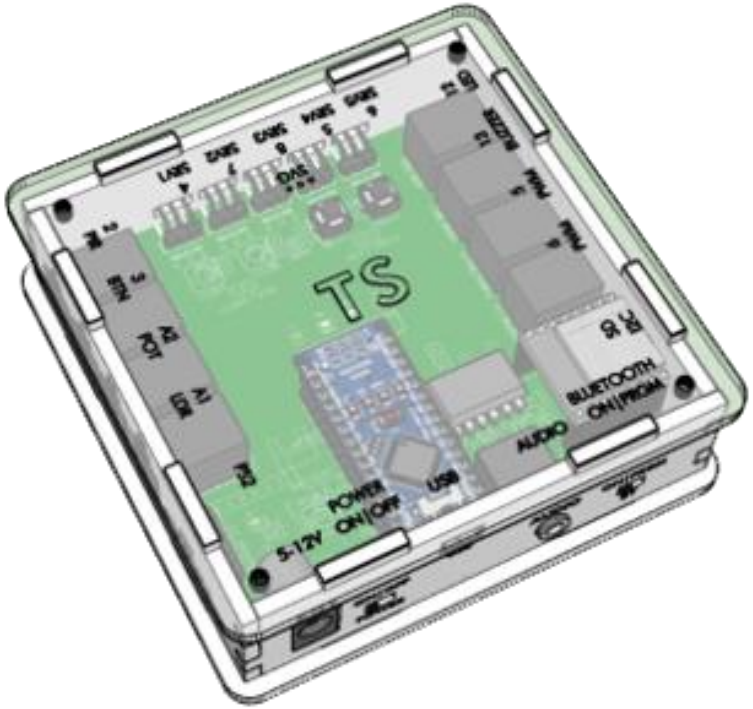
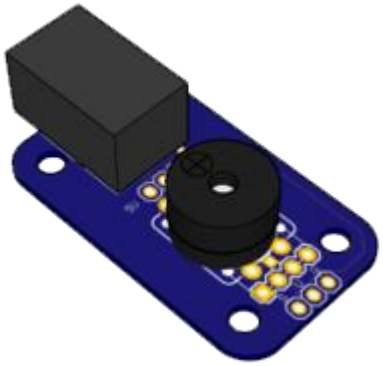


TScratch Basics

Coding with Arduino IDE (Software)

Learning Objective

In this lesson you will learn:



TScratch
(TSense Buzzer)

- Include a Buzzing output into your project!
- Coding another Digital output with Arduino

Code TSense (Buzzer)

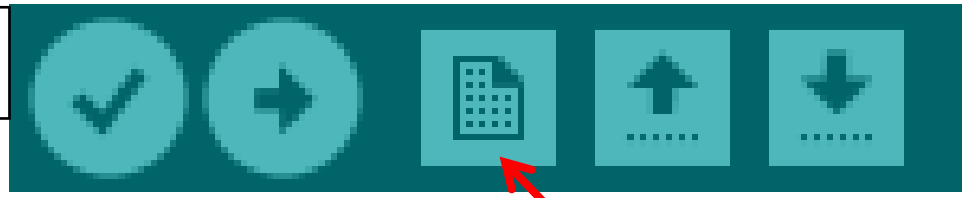
Write your control program with TScratch!

In a simple step, connect the TScratch with TSense (Buzzer) and TSense (Button)

Since buzzer is an output, we code using digitalWrite().

- Create a new project on the QUICK TOOLS

Quick Tools



New Project

Coding TSense(Buzzer)

- Declare the following in the declaration space
  1. Define BUTTON as pin 3
  2. Define BUZZER as pin 13
  3. A variable named “buttonstate” to store the state of the button
     Note: Comments appear after // or within /**/

TScratch3_Buzzer

```
/*
 * Program name : TScratch3_Buzzer
 * Description  : This program turns the buzzer on when the button is pressed.
 */

#define BUTTON_PIN 3                    // Defines pin of button as a constant
#define BUZZER_PIN 13                   // Defines pin of led as a constant
int buttonstate = 0;                    // Declares a integer variable to store the state of the button
```

Tsense (Buzzer) programming

- Declare the input/output used in the setup
  1. BUTTON → INPUT
  2. BUZZER → OUTPUT

```
void setup() {
  pinMode(BUTTON_PIN, INPUT);          // Initializes the button as an INPUT
  pinMode(BUZZER_PIN, OUTPUT);         // Initializes the buzzer as an OUTPUT
}
```

TSense(Buzzer) programming

- Lastly, use conditional programming in the loop() function

  <u>if</u> the button is pressed, the buzzer will make a sound of **2000Hz;**

  <u>else</u> it will make a sound of **5000Hz** (more high-pitched).

```
void loop() {
  buttonstate = digitalRead(BUTTON_PIN);
  if (buttonstate == HIGH) {
    tone(BUZZER_PIN, 2000);           // Turns the buzzer on at a frequency of 2000Hz
  }                                   // Frequency of the buzzer can be set to 31-20000Hz
  else {
    tone(BUZZER_PIN, 5000);           // Turns the buzzer on at a frequency of 5000Hz
    //noTone(BUZZER_PIN);             // Uncomment this line to turn off buzzer instead
  }
}
```

TSense(Buzzer) programming

- Note the last line of the program!
- Remove the comment // to turn off the buzzer (no sound)
- In other words, use the comment // to determine your desired outcome for the buzzer (different tone or no tone)

```
void loop() {
  buttonstate = digitalRead(BUTTON_PIN);
  if (buttonstate == HIGH) {
    tone(BUZZER_PIN, 2000);           // Turns the buzzer on at a frequency of 2000Hz
  }                                   // Frequency of the buzzer can be set to 31-20000Hz
  else {
    tone(BUZZER_PIN, 5000);           // Turns the buzzer on at a frequency of 5000Hz
    //noTone(BUZZER_PIN);             // Uncomment this line to turn off buzzer instead
  }
}
```

Complete Program

```
#define BUTTON_PIN 3                    // Defines pin of button as a constant
#define BUZZER_PIN 13                   // Defines pin of led as a constant
int buttonstate = 0;                    // Declares a integer variable to store the state of the button

void setup() {
  pinMode(BUTTON_PIN, INPUT);           // Initializes the button as an INPUT
  pinMode(BUZZER_PIN, OUTPUT);          // Initializes the buzzer as an OUTPUT
}

void loop() {
  buttonstate = digitalRead(BUTTON_PIN);
  if (buttonstate == HIGH) {
    tone(BUZZER_PIN, 2000);             // Turns the buzzer on at a frequency of 2000Hz
  }                                     // Frequency of the buzzer can be set to 31-20000Hz
  else {
    tone(BUZZER_PIN, 5000);             // Turns the buzzer on at a frequency of 5000Hz
    //noTone(BUZZER_PIN);               // Uncomment this line to turn off buzzer instead
  }
}
```

Try it yourself!

Control both the TSense(LED) and TSense(Buzzer)

- Make both the LED and buzzer switch “ON” when the button is pressed.
- Declare the LED as pin13.

Solution – TSense(Button, Buzzer, LED)

```
#define BUTTON_PIN 3
#define BUZZER_PIN 13
#define LED_PIN 12
int buttonstate = 0;    // Declares a variable to store the state of the button

void setup() {
  pinMode(BUTTON_PIN, INPUT);        // initialize button as an input
  pinMode(LED_PIN, OUTPUT);          // initialize led and buzzer as an output.
  pinMode(BUZZER_PIN, OUTPUT);
}

void loop() {
  buttonstate = digitalRead(BUTTON_PIN);
  if (buttonstate == HIGH) {
    digitalWrite(LED_PIN, HIGH);       // Turns the led on (HIGH is the voltage level)
    tone(BUZZER_PIN, 2000);            // Turns the buzzer on (2000 is the frequency of the tone)
  }
  else {
    digitalWrite(LED_PIN, LOW);        // Turns the led off
    noTone(BUZZER_PIN);                // Turns the buzzer off
  }
}
```