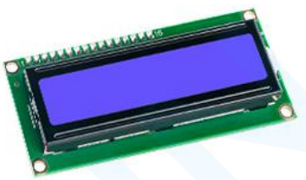
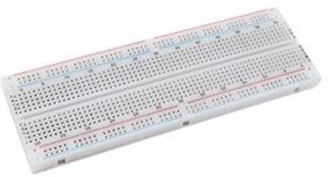
Raspberry Pi Pico Starter Kit

--Arduino IDE

Content

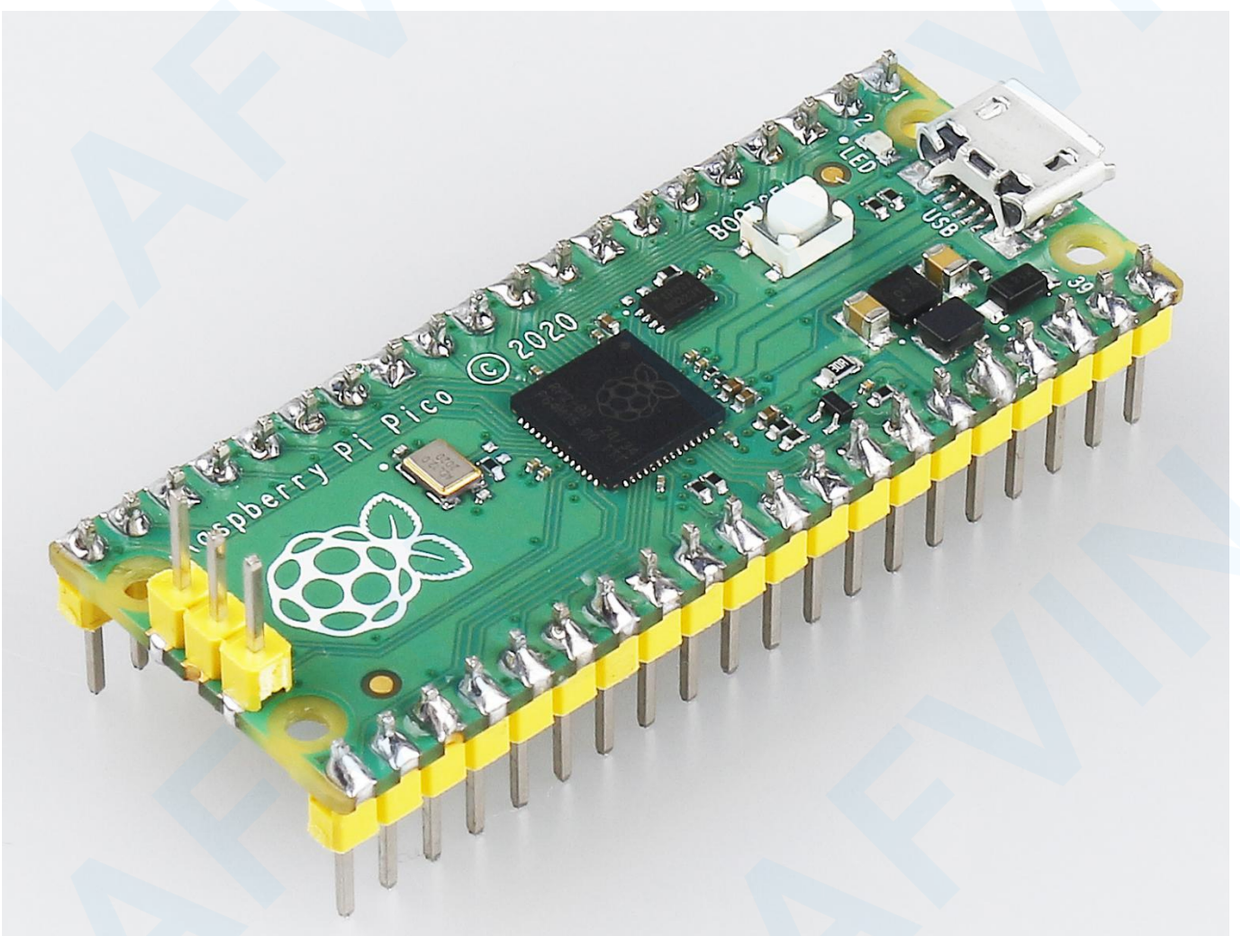
Packing List

 Raspberry Pi Pico x1	 IIC LCD 1602 x1	 830 Tie-Points Breadboard x1	 Obstacle Avoidance Module x1
 HC-SR501 PIR Motion Sensor x1	 SG90 Servo x1	 Potentiometer (10k) x1	 Passive Buzzer x1 Active Buzzer x1
 Resistor-220R/1k/10k x30	 Micro-USB Cable x1	 Button Switch x8	 WS2812 RGB 8 LEDSs Strip x1
 F-M DuPont Cable x10	 65 Jumper Wire x1	 Thermistor x1 Tilt Switch x1	 Photoresistor x2 S8050 Transistor x2
 LED-RGB x5	 LED-Yellow x5	 LED-Green x5	 LED-RGB x1

Raspberry Pi Pico Introduction

The Raspberry Pi Pico is a microcontroller board based on the Raspberry Pi RP2040 microcontroller chip.

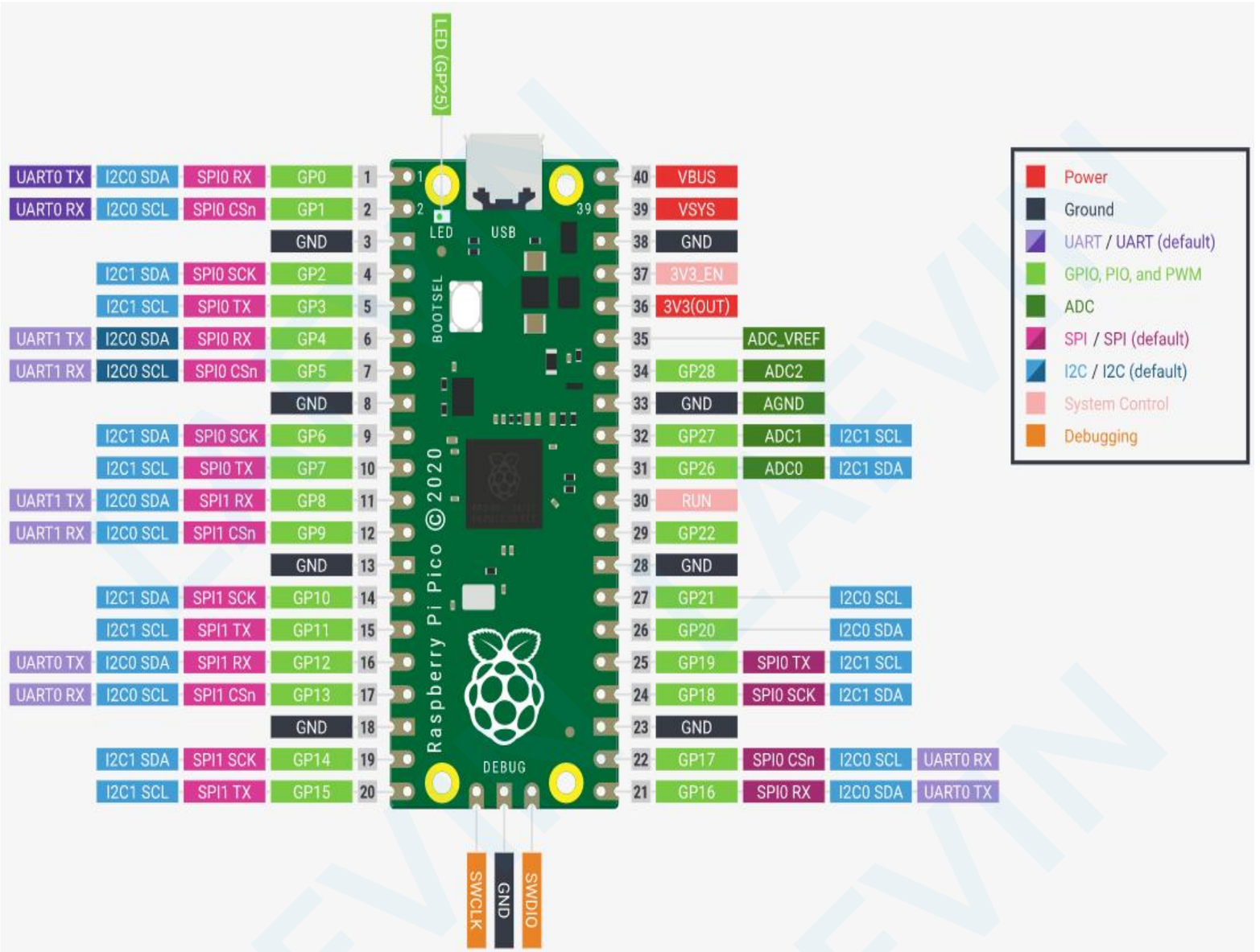
Whether you want to learn the MicroPython programming language, take the first step in physical computing, or want to build a hardware project, Raspberry Pi Pico and its amazing community – will support you every step of the way. In the project, it can control anything, from LEDs and buttons to sensors, motors, and even other microcontrollers.



Specifications

- 21 mm × 51 mm form factor
- RP2040 microcontroller chip designed by Raspberry Pi in the UK
- Dual-core Arm Cortex-M0+ processor, flexible clock running up to 133 MHz
- 264KB on-chip SRAM
- 2MB on-board QSPI Flash
- 26 multifunction GPIO pins, including 3 analog inputs
- 2 × UART, 2 × SPI controllers, 2 × I2C controllers, 16 × PWM channels
- 1 × USB 1.1 controller and PHY, with host and device support
- 8 × Programmable I/O (PIO) state machines for custom peripheral support
- Supported input power 1.8–5.5V DC
- Operating temperature -20°C to +85°C
- Castellated module allows soldering direct to carrier boards
- Drag-and-drop programming using mass storage over USB
- Low-power sleep and dormant modes
- Accurate on-chip clock
- Temperature sensor
- Accelerated integer and floating-point libraries on-chip

Pinout


LED(GP25)
Raspberry Pi Pico ©2020
BOOTSEL
USB
LED
DEBUG
SWCLK
GND
SWDIO
Power
Ground
UART / UART (default)
GPIO, PIO, and PWM
ADC
SPI / SPI (default)
I2C / I2C (default)
System Control
Debugging
1 UART0 TX I2C0 SDA SPI0 RX GP0
2 UART0 RX I2C0 SCL SPI0 CSn GP1
3 GND
4 I2C1 SDA SPI0 SCK GP2
5 I2C1 SCL SPI0 TX GP3
6 UART1 TX I2C0 SDA SPI0 RX GP4
7 UART1 RX I2C0 SCL SPI0 CSn GP5
8 GND
9 I2C1 SDA SPI0 SCK GP6
10 I2C1 SCL SPI0 TX GP7
11 UART1 TX I2C0 SDA SPI1 RX GP8
12 UART1 RX I2C0 SCL SPI1 CSn GP9
13 GND
14 I2C1 SDA SPI1 SCK GP10
15 I2C1 SCL SPI1 TX GP11
16 UART0 TX I2C0 SDA SPI1 RX GP12
17 UART0 RX I2C0 SCL SPI1 CSn GP13
18 GND
19 I2C1 SDA SPI1 SCK GP14
20 I2C1 SCL SPI1 TX GP15
40 VBUS
39 VSYS
38 GND
37 3V3_EN
36 3V3(OUT)
35 ADC_VREF
34 GP28 ADC2
33 GND AGND
32 GP27 ADC1 I2C1 SCL
31 GP26 ADC0 I2C1 SDA
30 RUN
29 GP22
28 GND
27 GP21 I2C0 SCL
26 GP20 I2C0 SDA
25 GP19 SPI0 TX I2C1 SCL
24 GP18 SPI0 SCK I2C1 SDA
23 GND
22 GP17 SPI0 CSn I2C0 SCL UART0 RX
21 GP16 SPI0 RX I2C0 SDA UART0 TX

Name	Description	Function
GP0-GP28	General-purpose input/output pins	Act as either input or output and have no fixed purpose of their own
GND	0 volts ground	Several GND pins around Pico to make wiring easier.
RUN	Enables or diables your Pico	Start and stop your Pico from another microcontroller.
GPxx_ADCx	General-purpose input/output or analog input	Used as an analog input as well as a digital input or output – but not both at the same time.
ADC_VREF	Analog-to-digital converter (ADC) voltage reference	A special input pin which sets a reference voltage for any analog inputs.
AGND	Analog-to-digital converter (ADC) 0 volts ground	A special ground connection for use with the ADC_VREF pin.
3V3(O)	3.3 volts power	A source of 3.3V power, the same voltage your Pico runs at internally, generated from the VSYS input.
3v3(E)	Enables or disables the power	Switch on or off the 3V3(O) power, can also switches your Pico off.
VSYS	2-5 volts power	A pin directly connected to your Pico's internal power supply, which cannot be switched off without also switching Pico off.
VBUS	5 volts power	source of 5 V power taken from your Pico's micro USB port, and used to power hardware which needs more than 3.3 V.

The best place to find everything you need to get started with your Raspberry Pi Pico.

Or you can click on the links below:

- Raspberry Pi Pico product brief
- Raspberry Pi Pico datasheet
- Getting started with Raspberry Pi Pico: C/C++ development
- Raspberry Pi Pico C/C++ SDK
- API-level Doxygen documentation for the Raspberry Pi Pico C/C++ SDK
- Raspberry Pi Pico Python SDK
- Raspberry Pi RP2040 datasheet
- Hardware design with RP2040
- Raspberry Pi Pico design files
- Raspberry Pi Pico STEP file

Getting Started with Arduino(Important)

This tutorial includes installing Arduino IDE, programming Raspberry Pi Pico with the most popular microcontroller development environment-Arduino IDE and some interesting and practical projects to help you learn Arduino code quickly.

Why Arduino IDE?

Arduino is an open source platform with powerful software and hardware features.

Even if you are a beginner, you can master it in no time. It provides an integrated development environment (IDE) for code compilation that is compatible with various control boards. In fact, the Arduino IDE supports almost all hobbyist microcontrollers, including the Raspberry Pi Pico. Therefore, all you need to do is download the Arduino IDE, upload the sketches (i.e. code files) to the control board, and then you can see the relative experimental phenomena.

For more information, refer to Arduino Website.

Install Arduino IDE

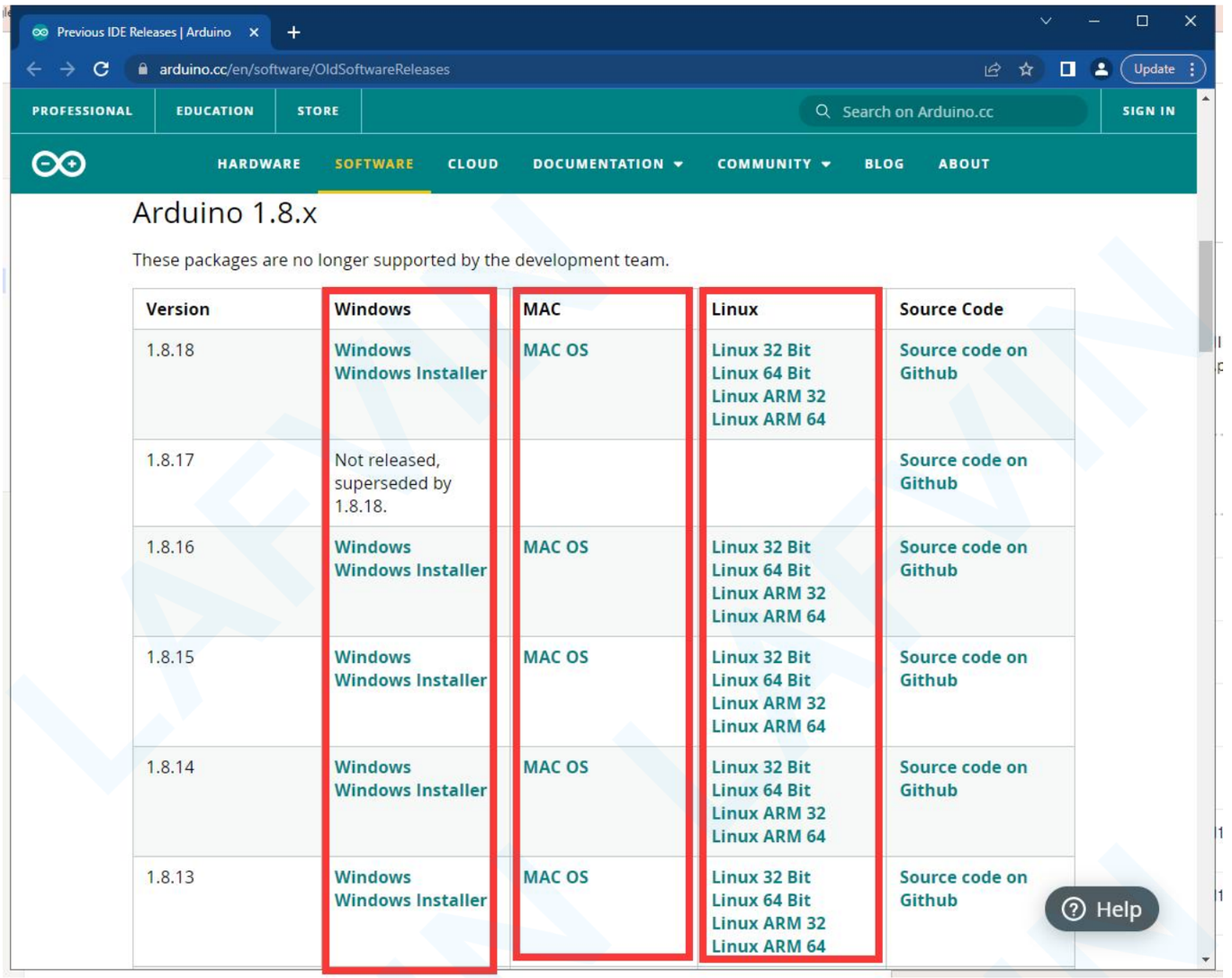
Note

The Raspberry Pi Pico is a newly released control board and may not be compatible on older versions of the Arduino IDE, so it is recommended that you download the latest version of the IDE, at least version 1.8.13 or higher.

If you are more explorative, you can also download Arduino IDE 2.0.x(**Not recommended**). Visit https://www.arduino.cc.This version is still in beta, so you may encounter some strange problems, but it is faster and even more powerful!

Download Arduino IDE

Visit Arduino IDE Software Releases. Download at least version 1.8.13 or higher.



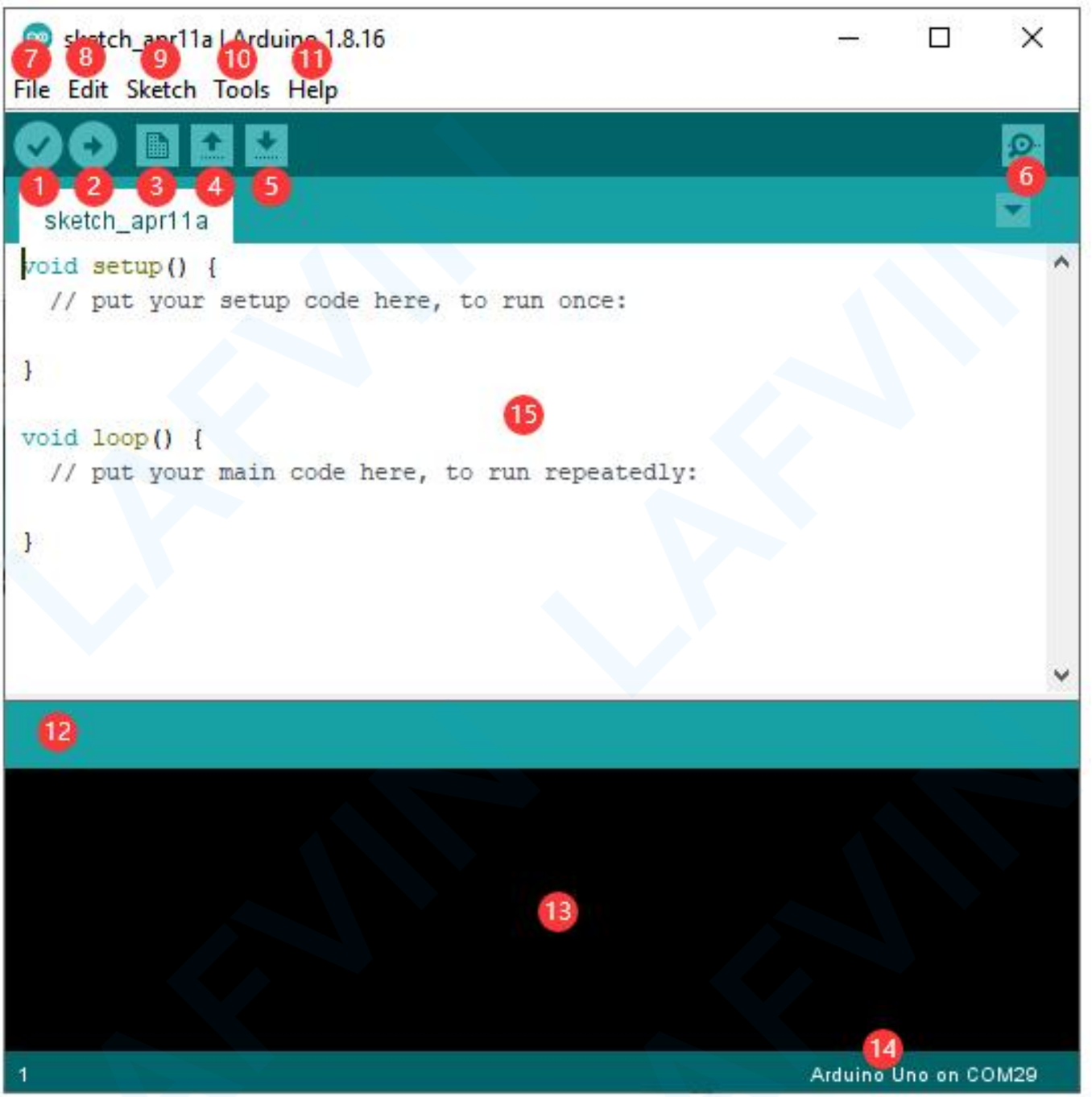
Note

Any questions about the installing Arduino IDE?

Refer to Arduino Guide docs Getting Started with Arduino products.

After the download completes, run the installer.After installation is complete, an Arduino Software shortcut will be generated in the desktop. Run the Arduino Software.

The interface of Arduino Software is as follows:



1. Verify: Compile your code. Any syntax problem will be prompted with errors.
2. Upload: Upload the code to your board. When you click the button, the RX and TX LEDs on the board will flicker fast and won't stop until the upload is done.
3. New: Create a new code editing window.
4. Open: Open an .ino sketch.
5. Save: Save the sketch.

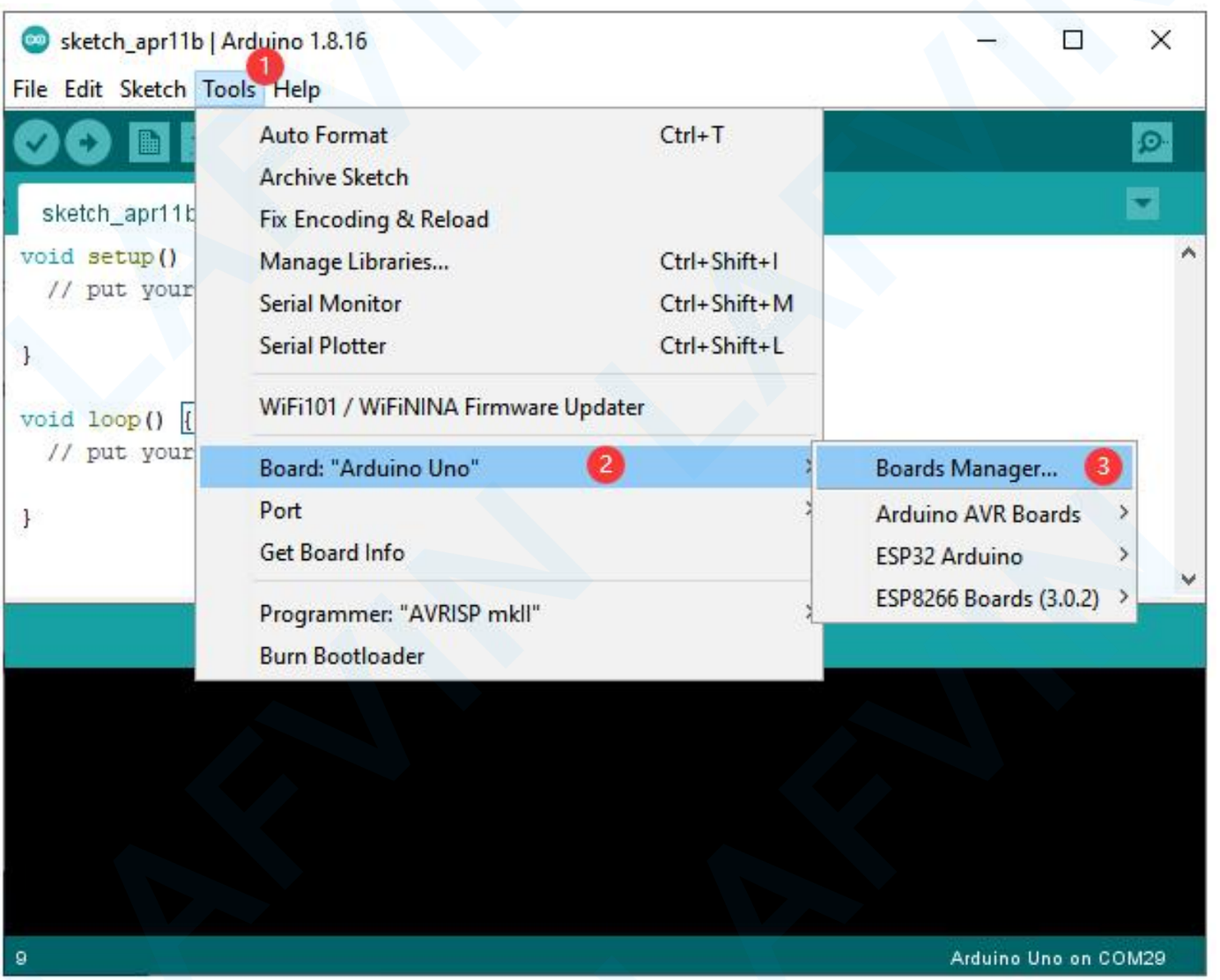
6. Serial Monitor: Click the button and a window will appear. It receives the data sent from your control board. It is very useful for debugging.
7. File: Click the menu and a drop-down list will appear, including file creating, opening, saving, closing, some parameter configuring, etc.
8. Edit: Click the menu. On the drop-down list, there are some editing operations like Cut, Copy, Paste, Find, and so on, with their corresponding shortcuts.
9. Sketch: Includes operations like Verify, Upload, Add files, etc. More important function is Include Library – where you can add libraries.
10. Tool: Includes some tools – the most frequently used Board (the board you use) and Port (the port your board is at). Every time you want to upload the code, you need to select or check them.
11. Help: If you're a beginner, you may check the options under the menu and get the help you need, including operations in IDE, introduction information, troubleshooting, code explanation, etc.
12. In this message area, no matter when you compile or upload, the summary message will always appear.
13. Detailed messages during compile and upload. For example, the file used lies in which path, the details of error prompts.
14. Board and Port: Here you can preview the board and port selected for code upload. You can select them again by Tools -> Board / Port if any is incorrect.
15. The editing area of the IDE. You can write code here.

Installation of Development Board Support Package

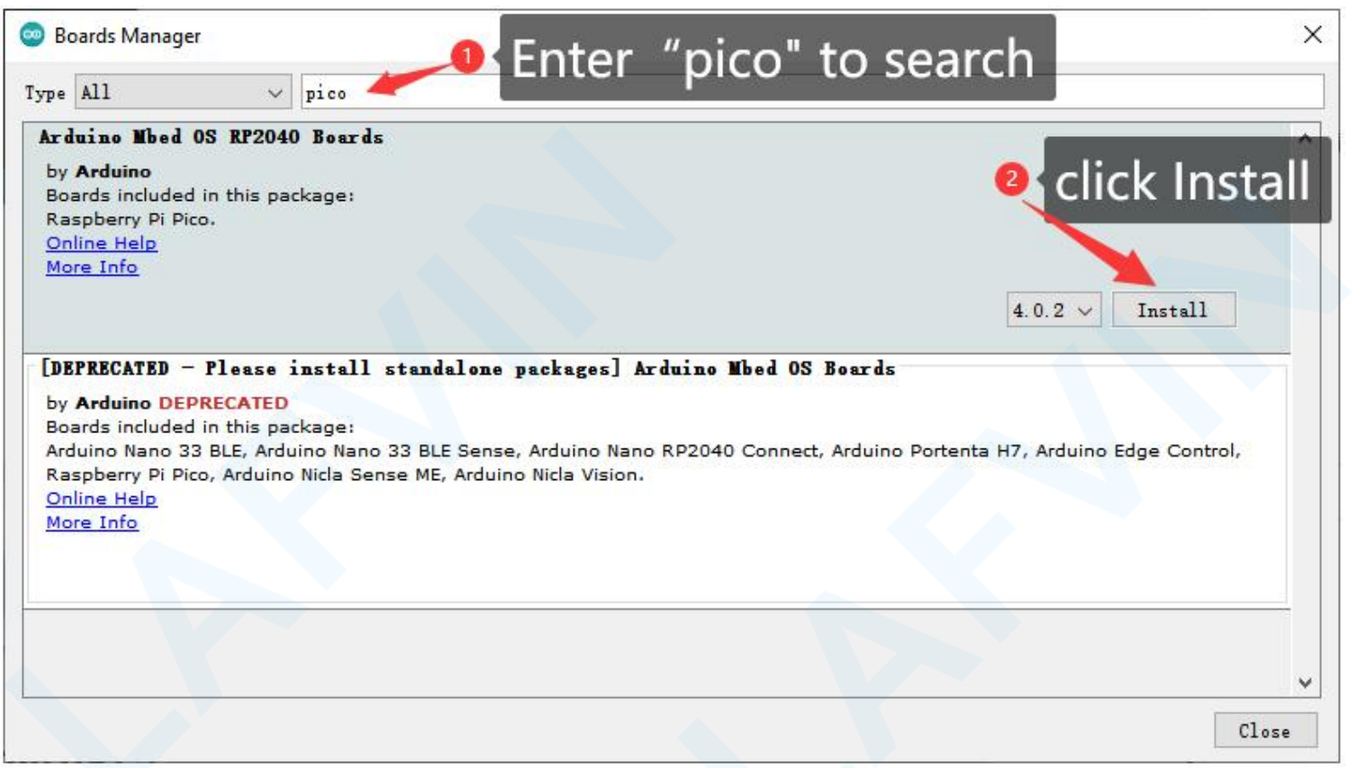
If your Pico is new and you want to use Arduino to learn and develop, you need to upload an Arduino-compatible Firmware for it. Please refer to the following steps to configure.

Step 1: Make sure your network is of good connection.

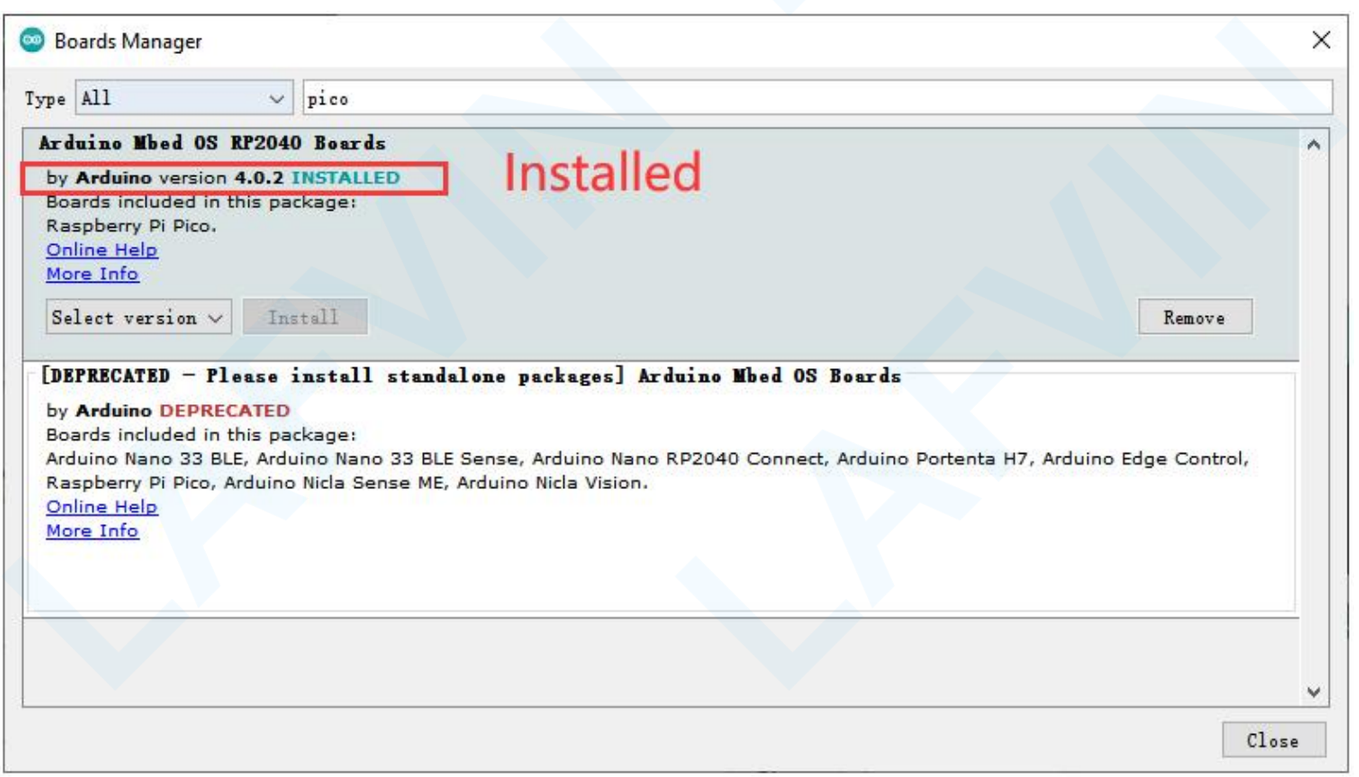
Step 2: Open the Boards Manager by clicking **Tools** -> **Board** -> **Boards Manager**.



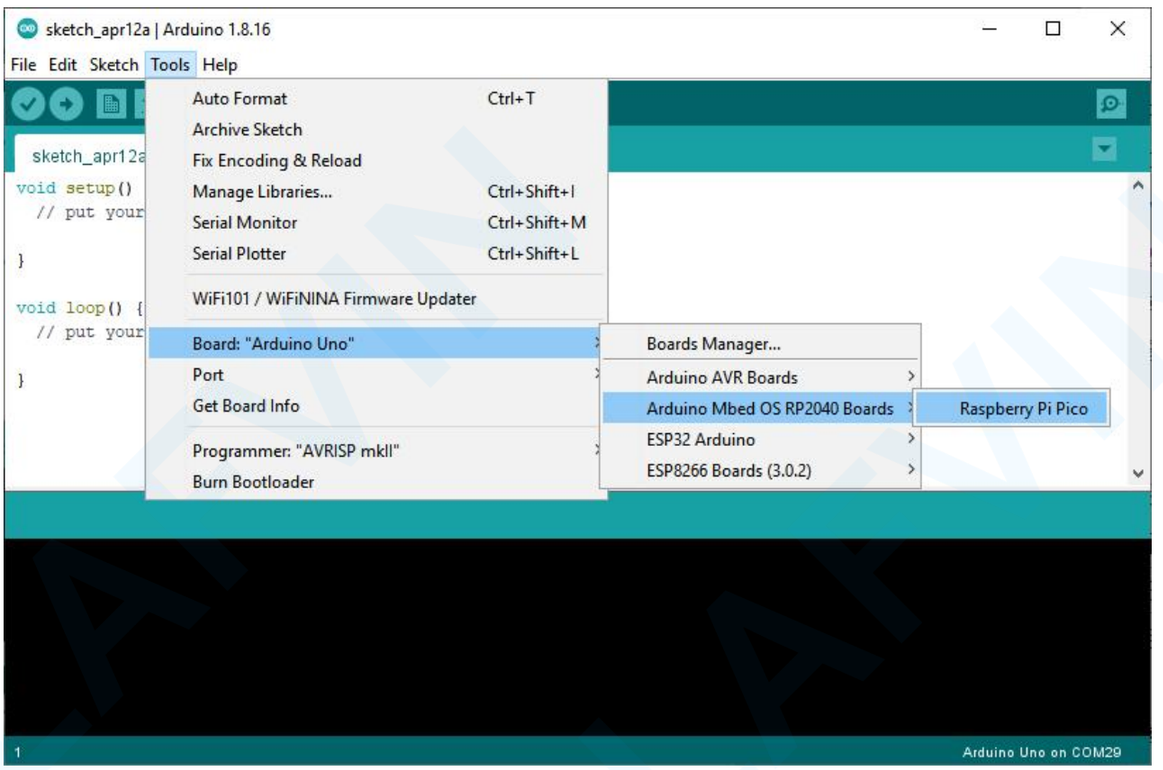
Step 3: Search for **Pico(Arduino Mbed OS RP2040 Boards)** and click **install** button.

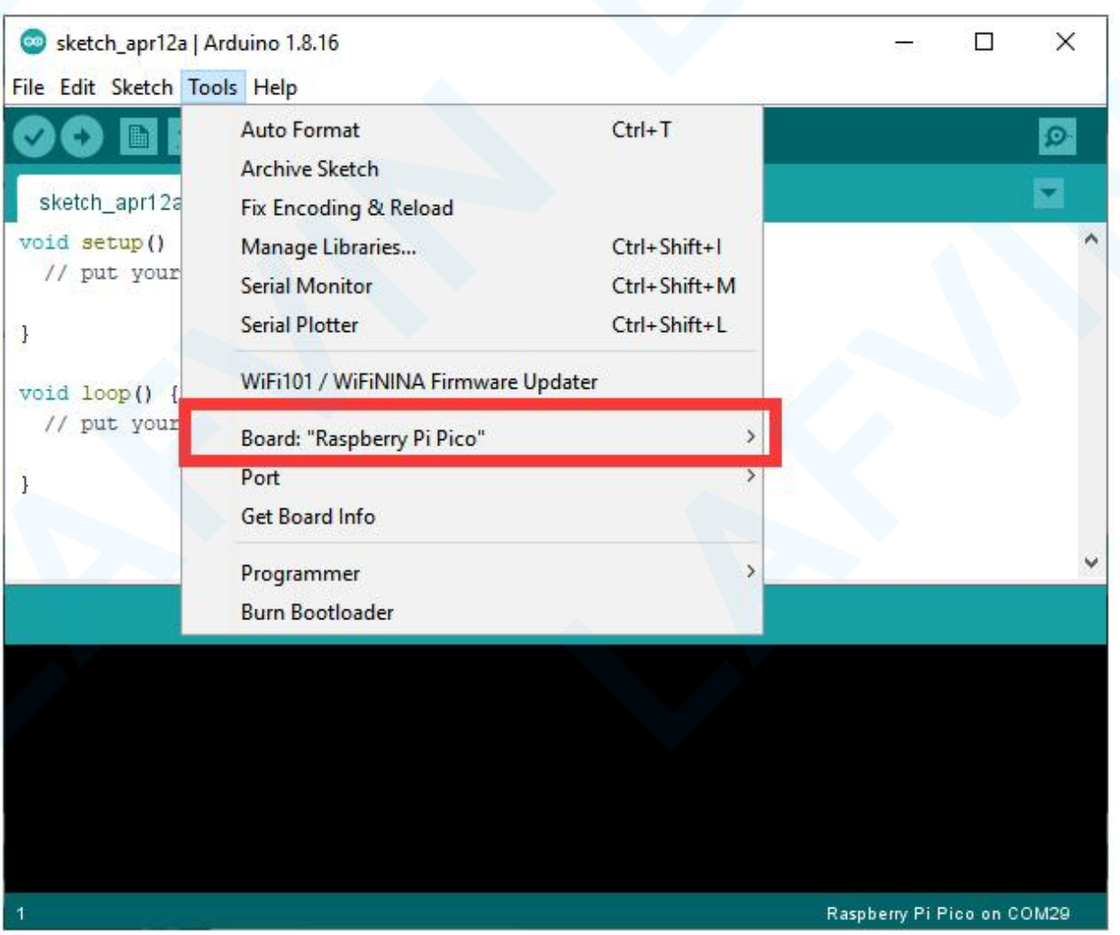


Installation takes a few minutes



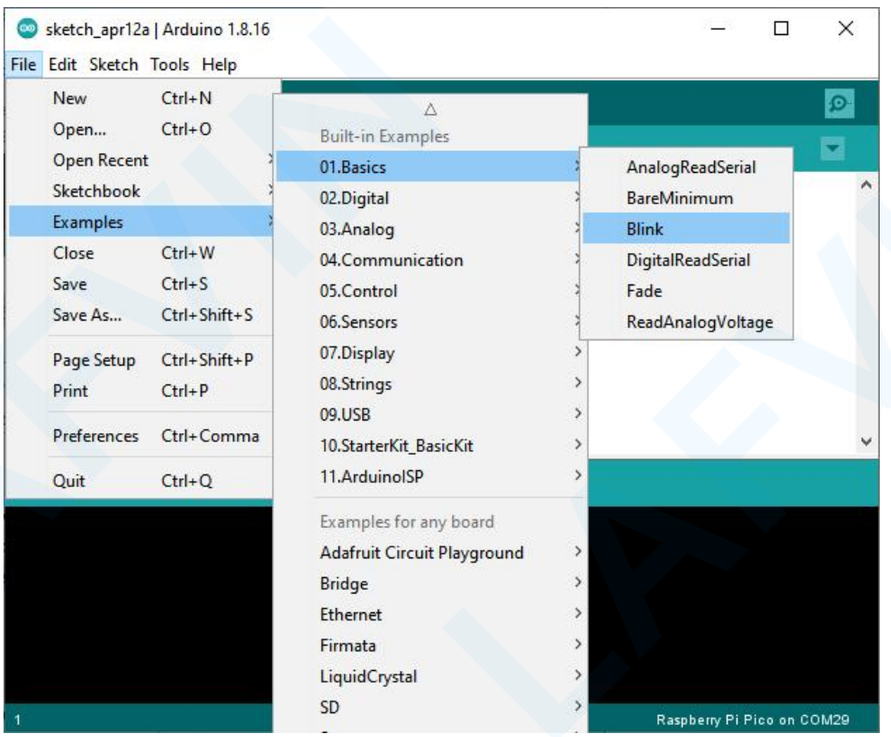
Step 4: Once the installation is complete, you can select the board as **Raspberry Pi Pico**.



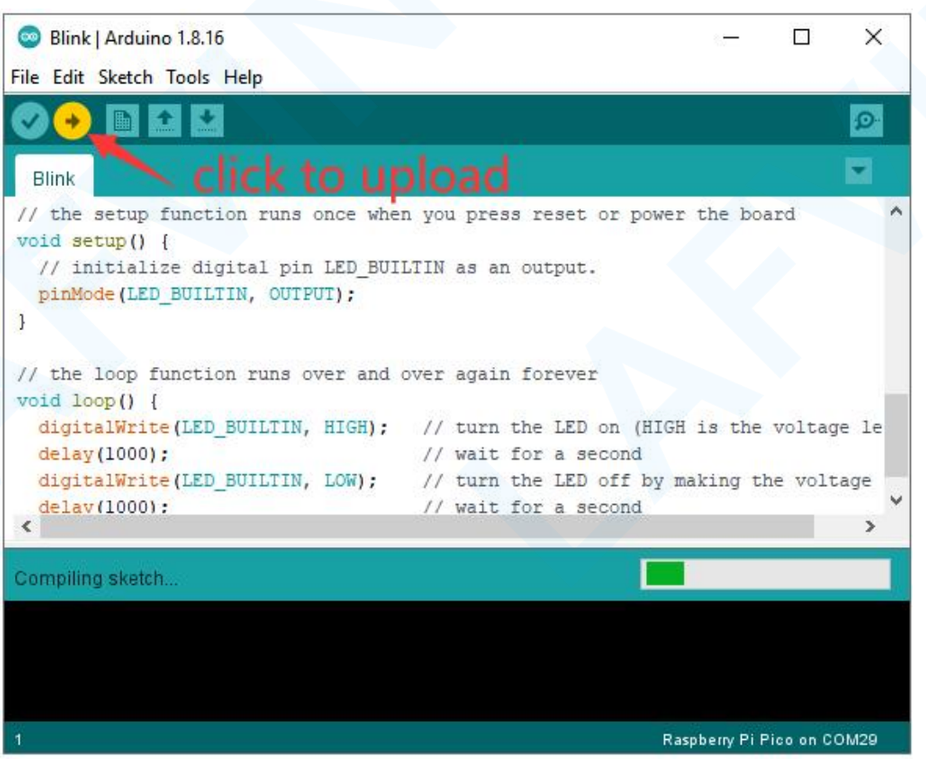


Uploading Arduino-compatible Firmware for Pico(important)

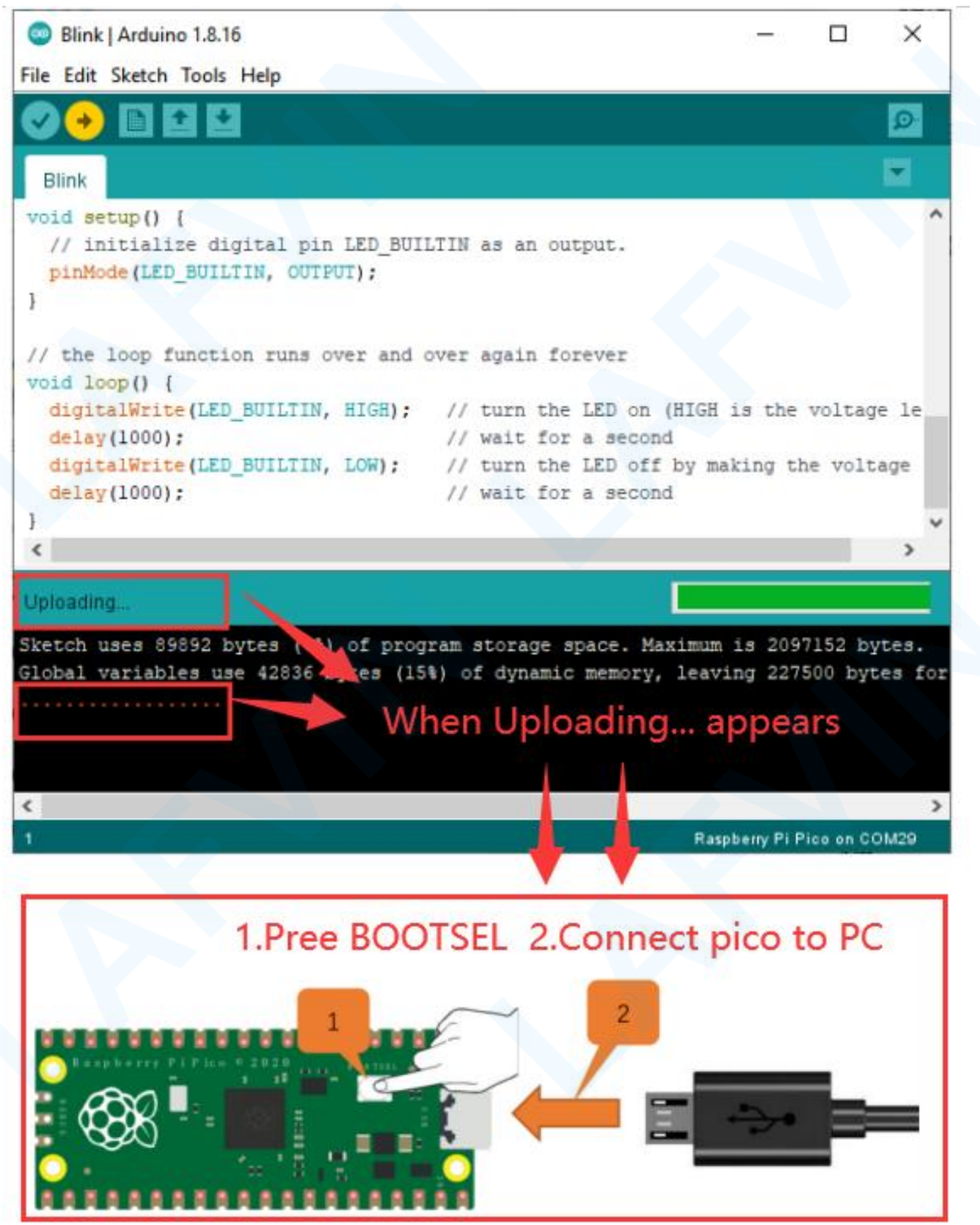
Step 1: Then open a blink sketch by click **Examples->Basics->Blink**.



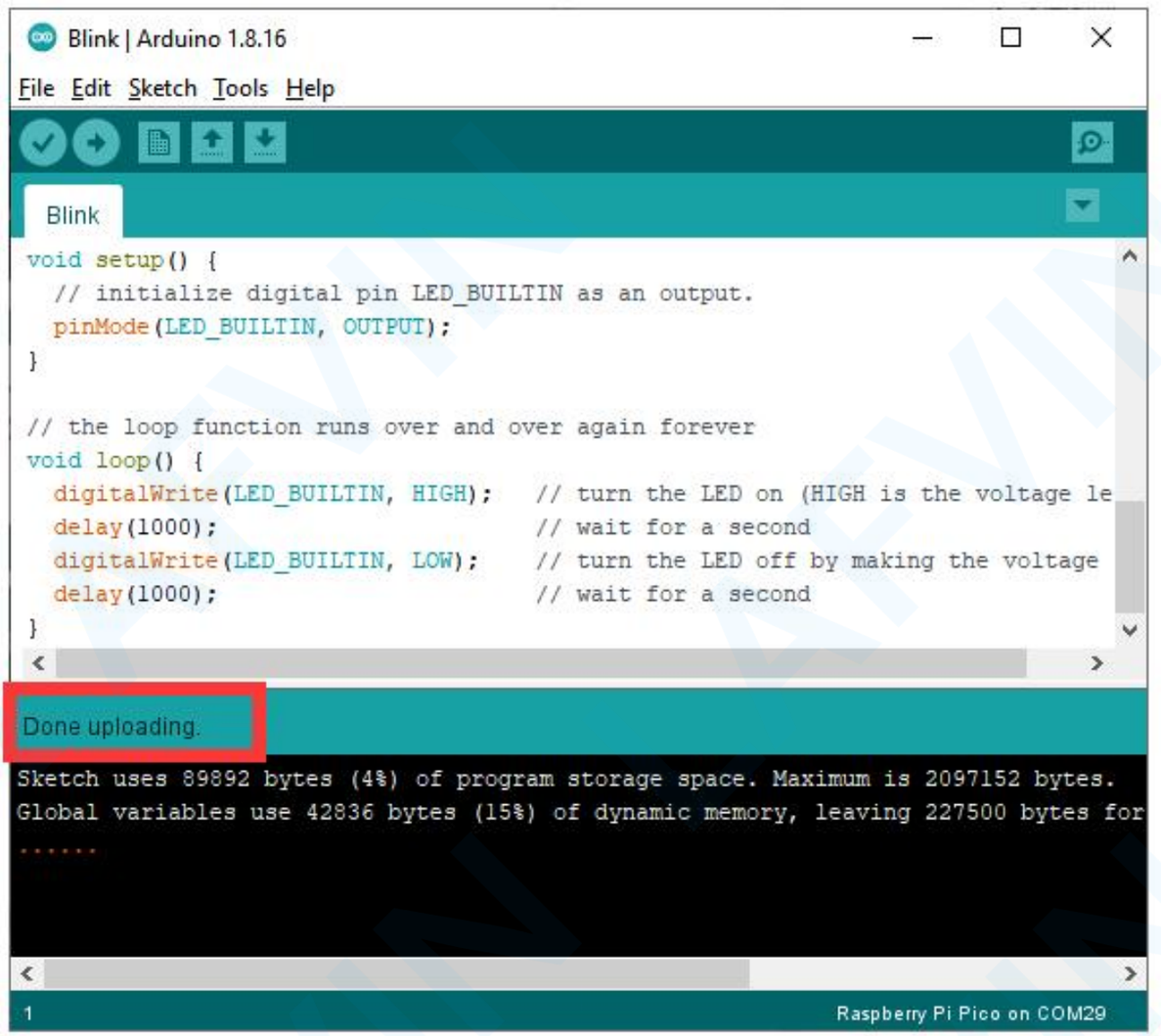
Step 2: Click on the upload icon



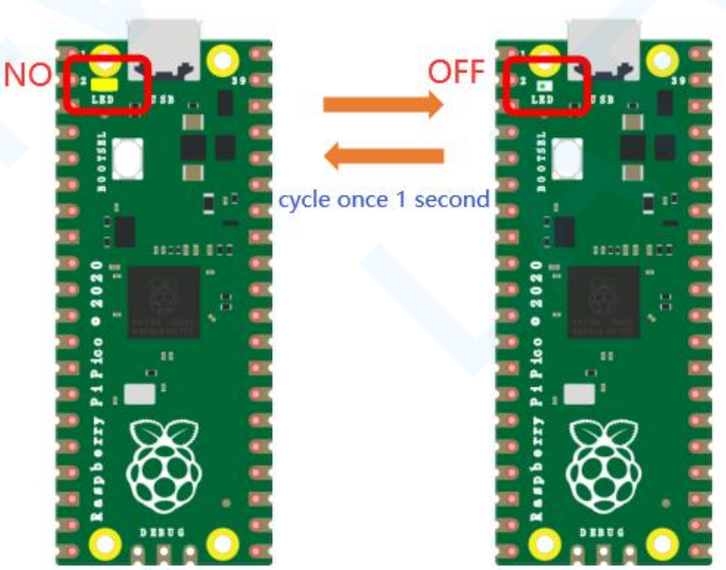
Step 3: When the summary message changes from **Compiling sketch** to **Uploading**. The compiling message appears • • • • ,Press **BOOTSEL** button immediately and connect Pico to the computer with a Micro USB cable.



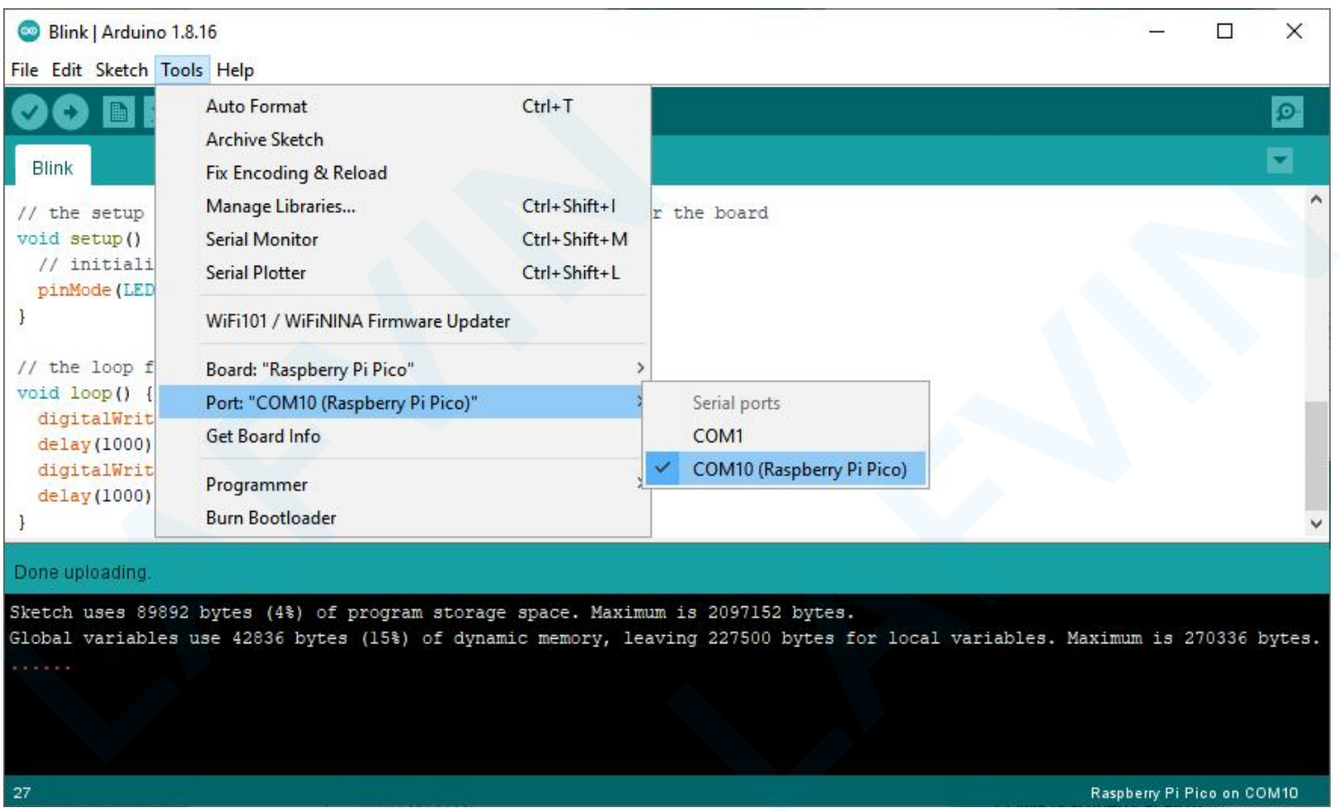
Step 4: After the **Done Uploading** appear,Firmware is successfully uploaded to pico.



And the indicator led on Pico starts to flash.

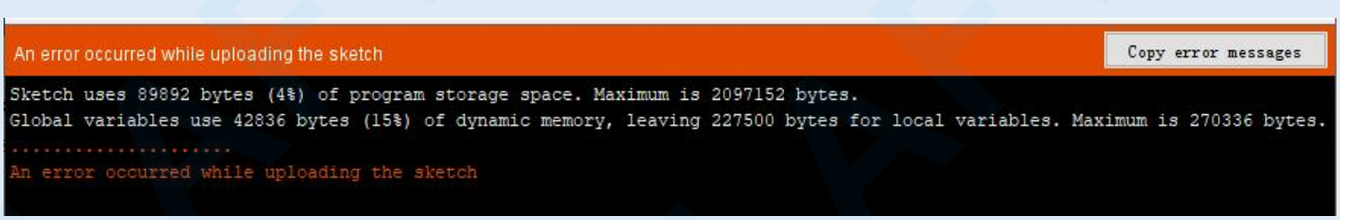


Pico will be recognized by the computer as COMx (Raspberry Pi Pico).Click **Tools>Port** to select it. X of COMx varies from different computers. In our case, it is COM10.



Note

1) **Uploading Firmware** is very important and only necessary for the first use on the Arduino IDE, otherwise your code will upload unsuccessfully.
2) If your indicator led is not blinking, or there is an error in the upload.Repeat the above steps.



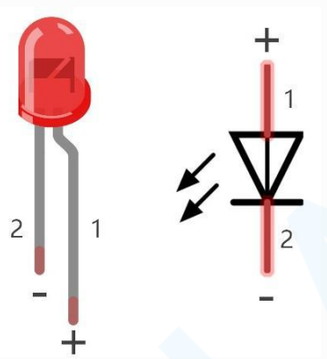
3) At the first time you use Arduino to upload sketch for Pico(Uploading Firmware for Pico), you don't need to select port. After that,each time before uploading sketch, please check whether the port has beed selected; otherwise, the downloading may fail.
4) Sometimes when using, Pico may lose firmware due to the code and fail to work. At this point, you can upload firmware for Pico as mentioned above.

Project 1 Hello, LED!(important)

Just as printing “Hello, world!” is the first step in learning programming, letting the LED light up is the traditional entry to learning physical programming.Knowledge point: **digital signal output**.

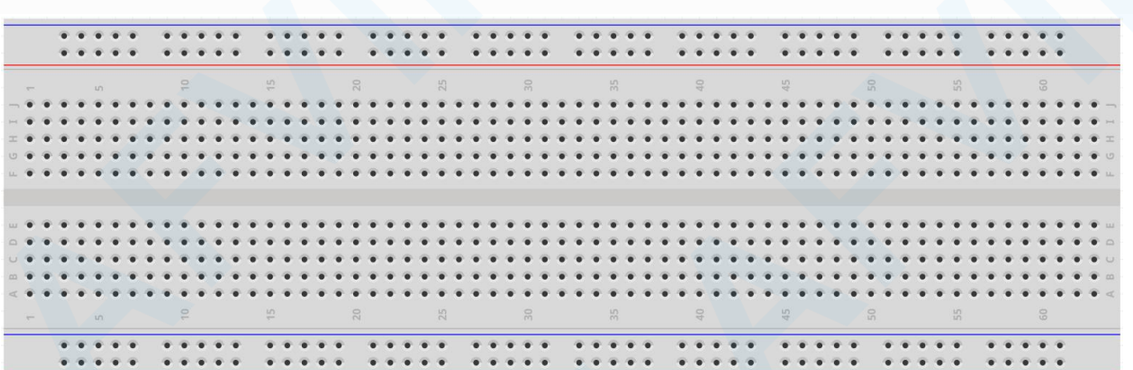
LED

An LED is a type of diode. All diodes only work if current is flowing in the correct direction and have two Poles.



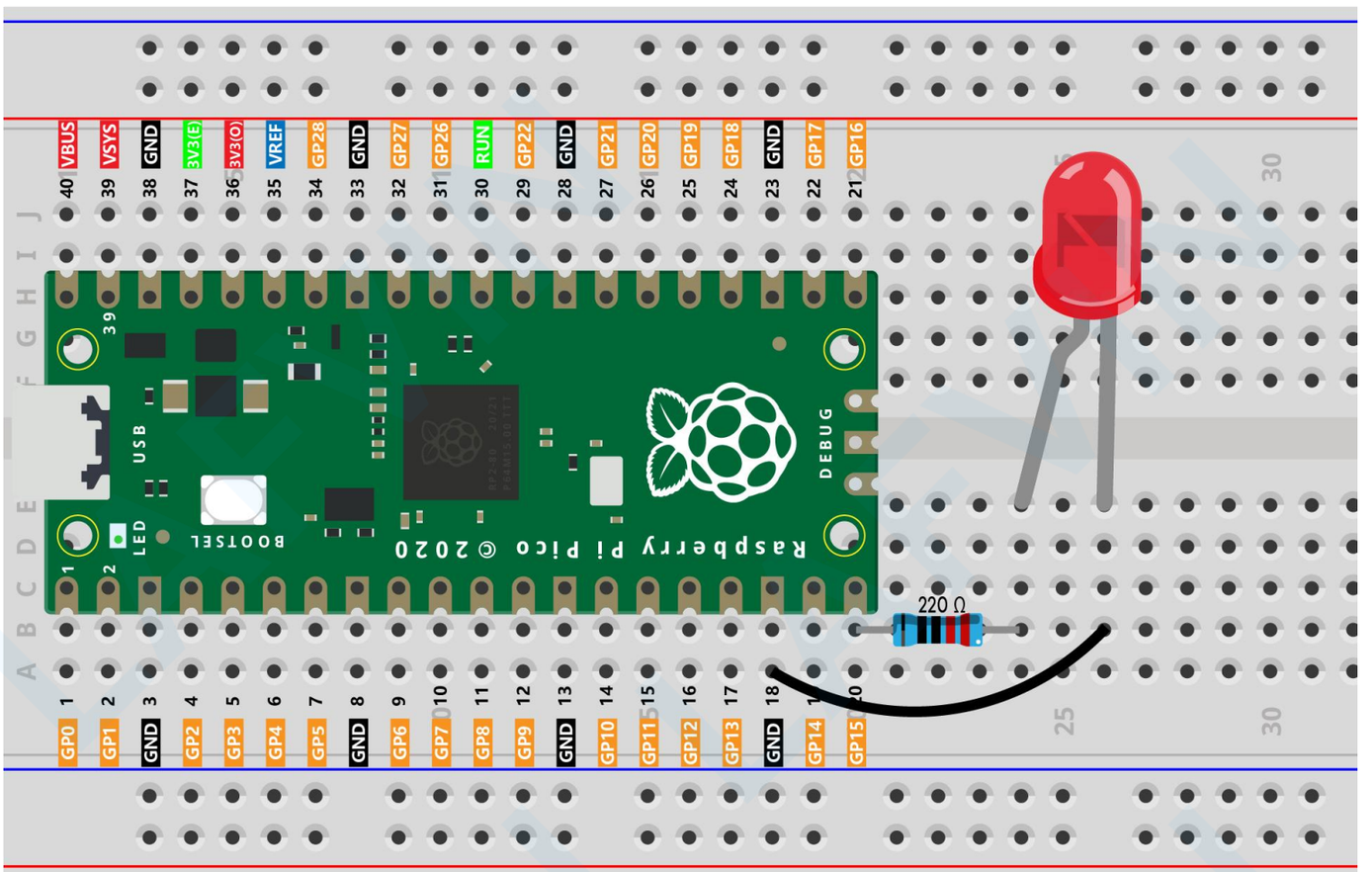
Breadboard

The breadboard is a rectangular plastic plate with a bunch of small holes in it. These holes allow us to easily insert electronic components and build electronic circuits. The breadboard does not permanently fix the electronic components, which makes it easy for us to repair the circuit and start over when we make a mistake.



Tip:Learn more about LED breadboard

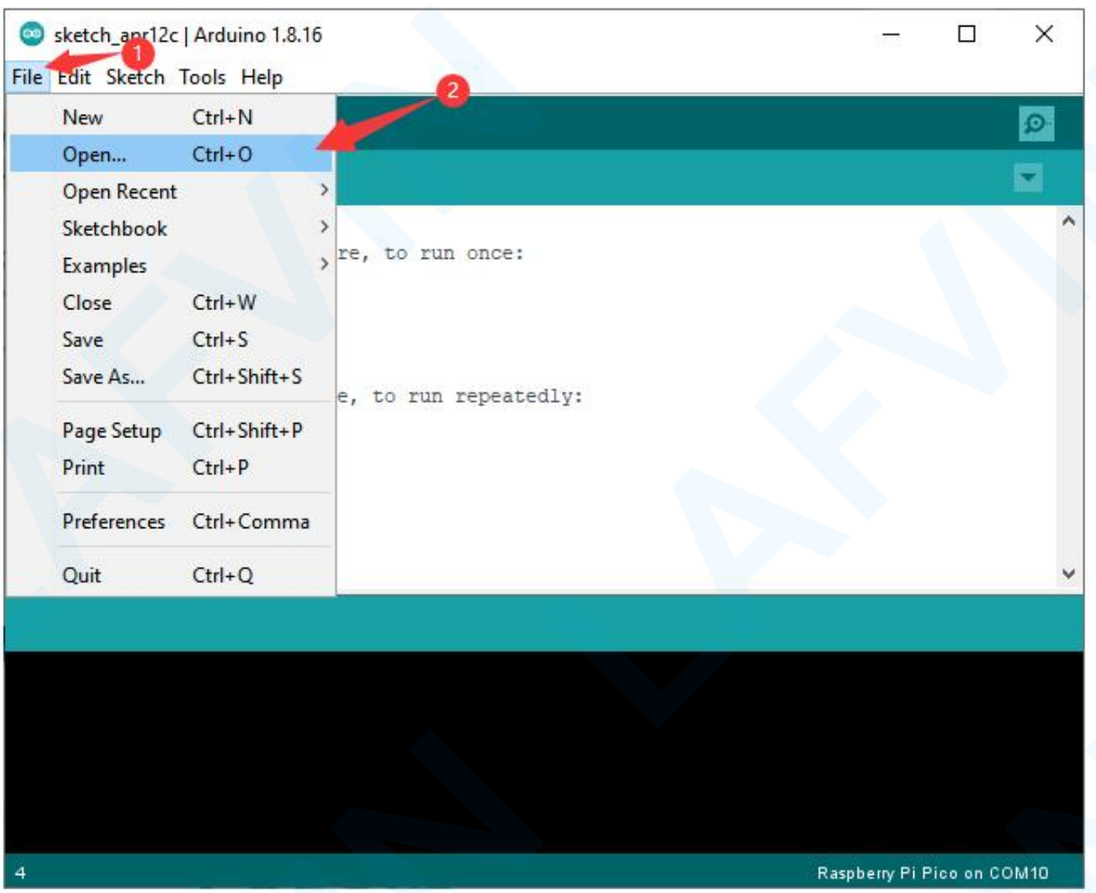
Wiring

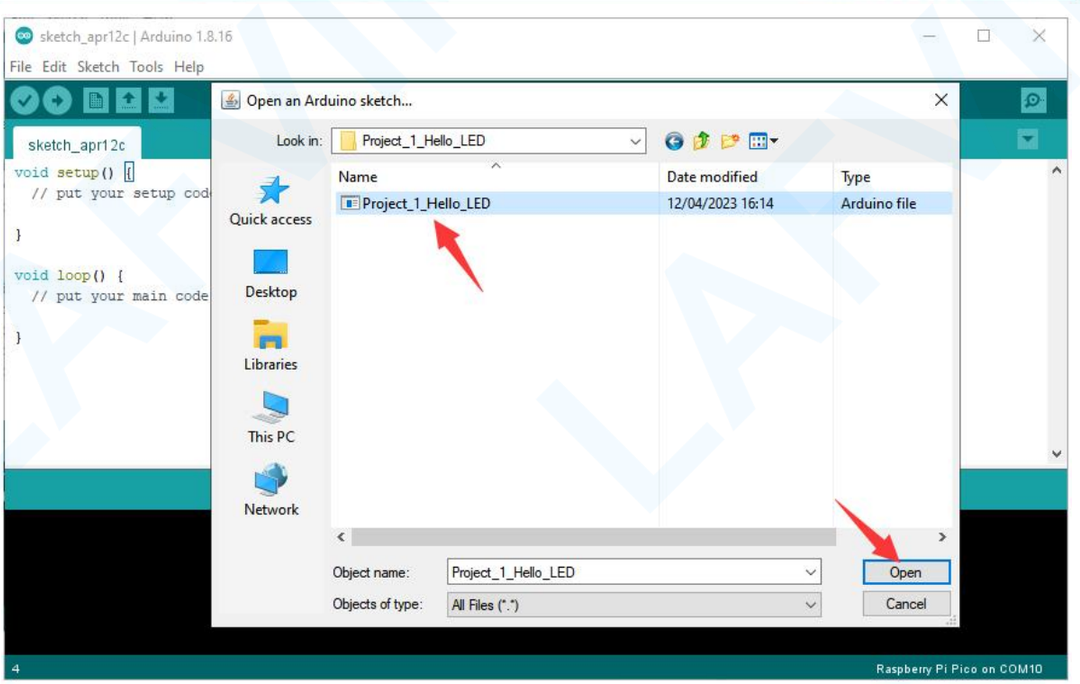

VBUS
VSYS
GND
3V3(E)
3V3(O)
VREF
GP28
GP27
GP26
RUN
GP22
GP21
GP20
GP19
GP18
GP17
GP16
GP0
GP1
GP2
GP3
GP4
GP5
GP6
GP7
GP8
GP9
GP10
GP11
GP12
GP13
GP14
GP15
USB
LED
BOOTSEL
DEBUG
Raspberry Pi Pico © 2020
220 Ω

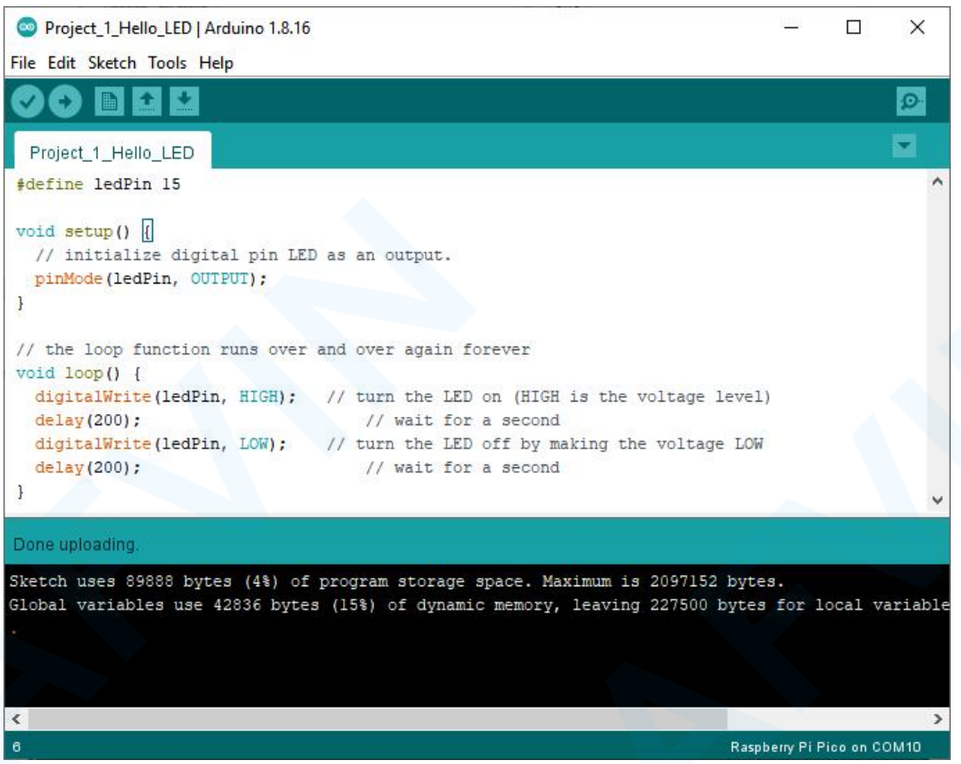
Code

Click the **File>>open** icon to open **Project 1_Hello_LED.ino** file.

*File path :**Raspberry Pi Pico Starter Kit\C\Codes***





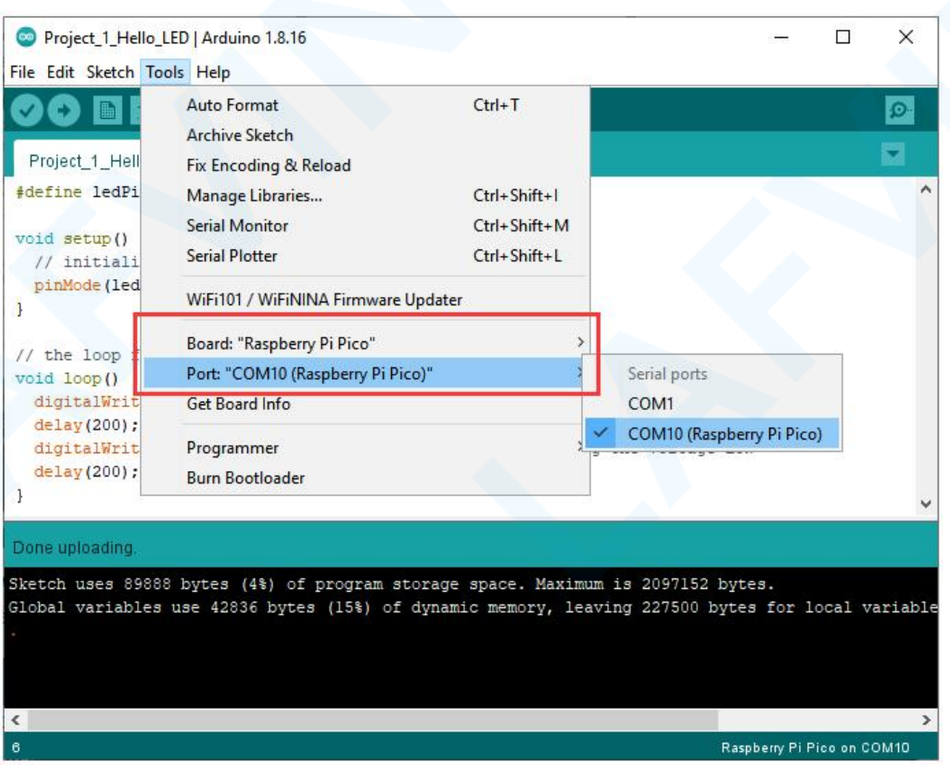


Before uploading code to Pico, please check the configuration of Arduino IDE.

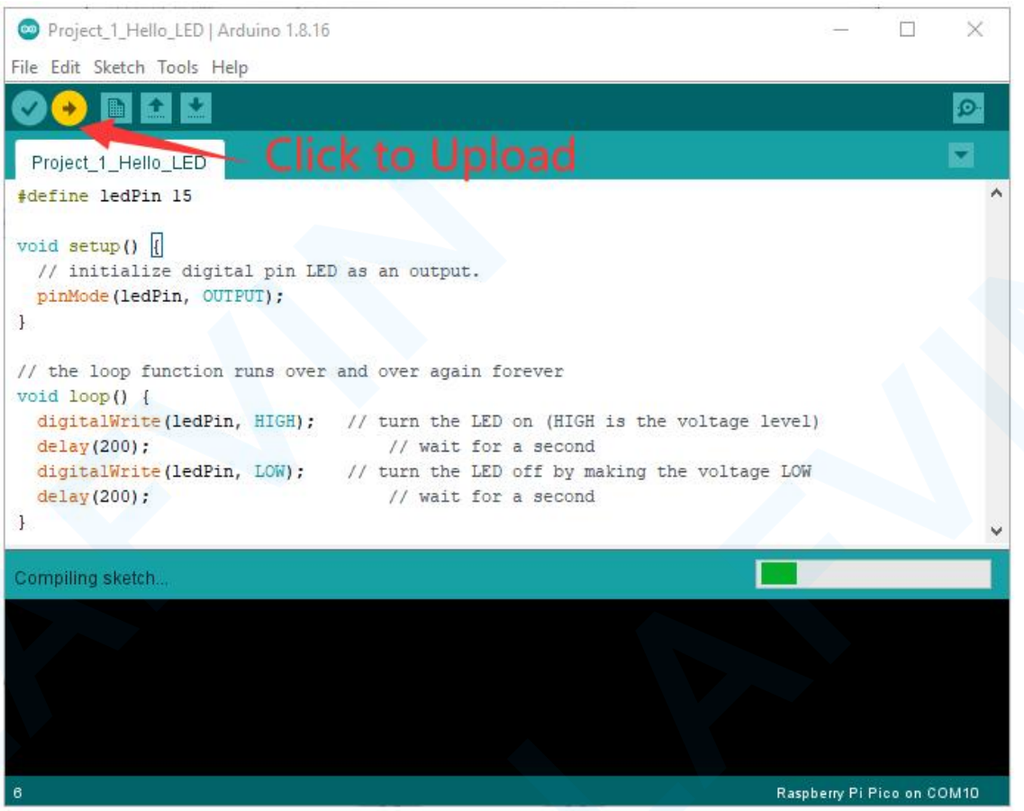
*Click **Tools**, make sure Board and Port are as follows:*

Board:"Raspberry Pi Pico"

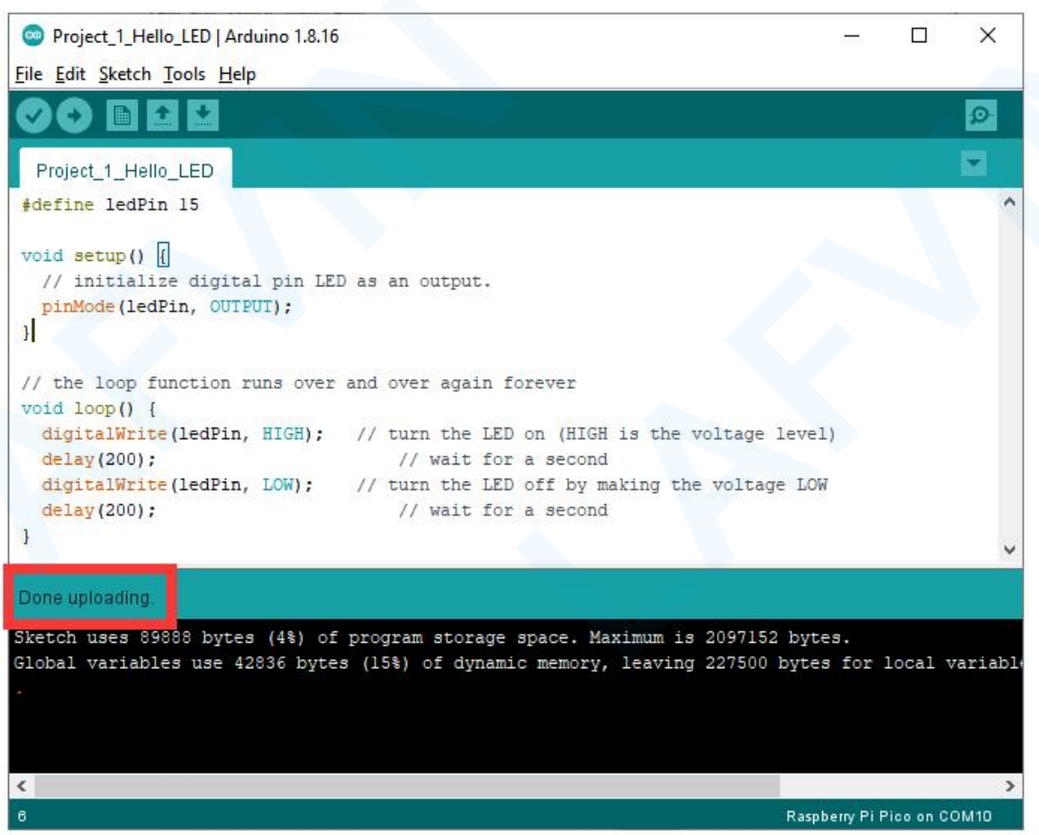
Port:"COMX(Raspberry Pi Pico)"

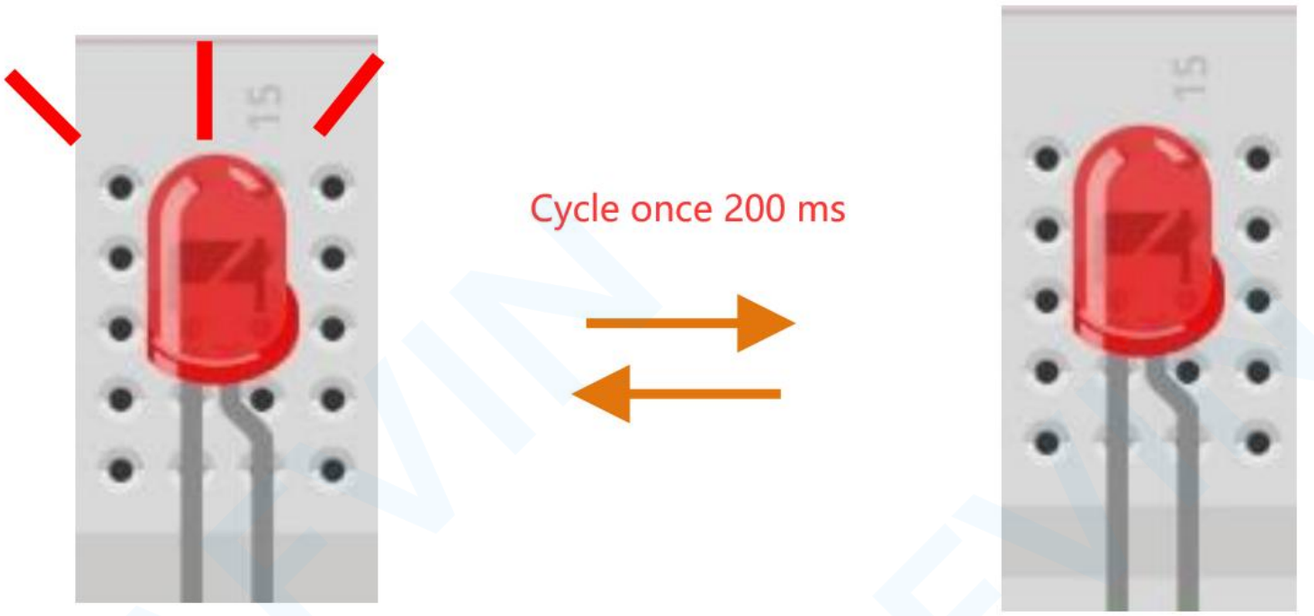


Click “Upload” to upload the sketch to Pico.



Done uploading.LED in the circuit starts Blink.





Note:

Code upload failed in Arduino IDE?

1. Make sure you have installed the Pico board in the Arduino IDE.
2. Check that the Board(Raspberry Pi Pico) or port（COMxx (Raspberry Pi Pico)）is selected correctly.
3. Make sure you have plugged the Pico into your computer.After trying to remove all sensors and modules connected to the pico, reconnect the pico to the computer.
4. If only COMxx (The complete one should be COMxx (Raspberry Pi Pico)) is displayed, it means that Pico is not correctly recognized by the computer. You need to Uploading Arduino-compatible Firmware for Pico.
5. FAQ

How it works?

```
#define ledPin 15
```

Here, we connect the LED to the GPIO15, so we define a variable **ledPin** to represent GPIO15. You could also just replace all the **ledPin** pins in the code with 15, but if you were to replace 15 with other pins you would have to modify 15 one by one, which would add a lot of work.

```
pinMode(ledPin, OUTPUT);
```

Now, you need to set the pin to OUTPUT mode in the setup() function.

- pinMode()

```
digitalWrite(ledPin, HIGH);
```

The above code has “set” the pin, but it will not light up the LED. Here, we use the digitalWrite() function to assign a high level signal to ledpin, which will cause a voltage difference between the LED pins, causing the LED to light up.

```
digitalWrite(ledPin, LOW);
```

If the level signal is changed to LOW, the ledPin’s signal will be returned to 0 V to turn LED off.

- digitalWrite()

```
delay(200);
```

An interval between on and off is required to allow people to see the change, so we use a delay(200) code to let the controller do nothing for 200 ms.

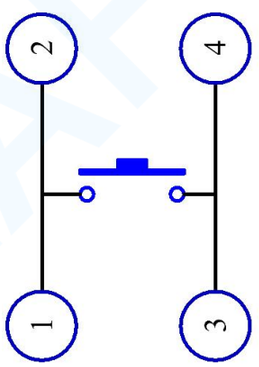
Project 2 Button Control LED

In the project, we will control the LED state through a Push Button Switch. When the button is pressed, our LED will turn ON, and when it is released, the LED will turn OFF. This describes a Momentary Switch.Knowledge point: **digital signal input**.

Component knowledge

Push button

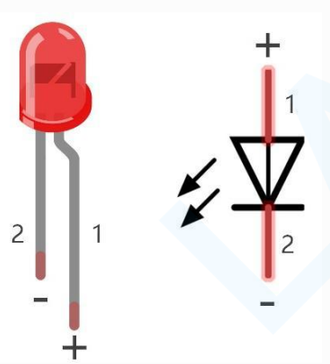
This type of Push Button Switch has 4 pins (2 Pole Switch). Two pins on the left are connected, and both left and right sides are the same per the illustration:



When the button on the switch is pressed, the circuit is completed (your project is Powered ON).

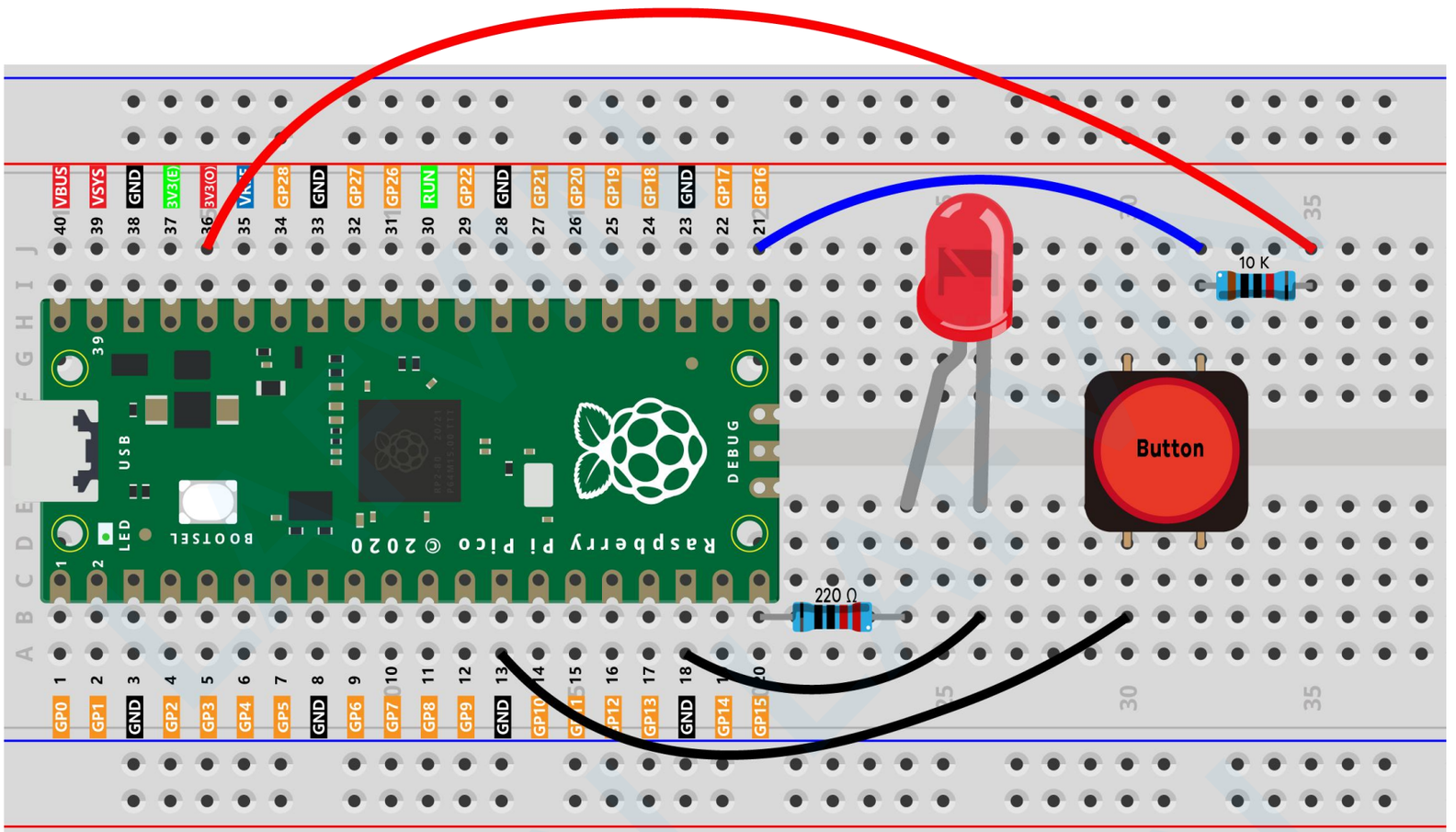
LED

An LED is a type of diode. All diodes only work if current is flowing in the correct direction and have two Poles.



Tip:Learn more about Button LED

Wiring

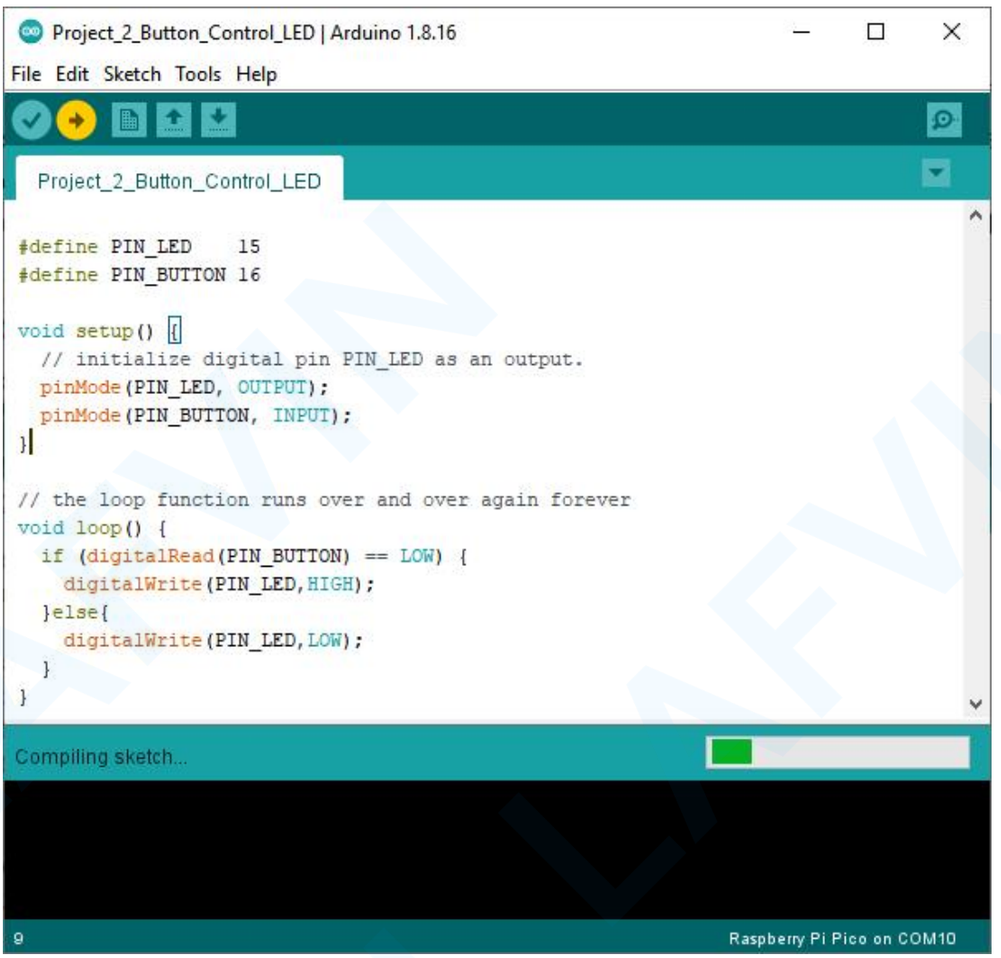


Code

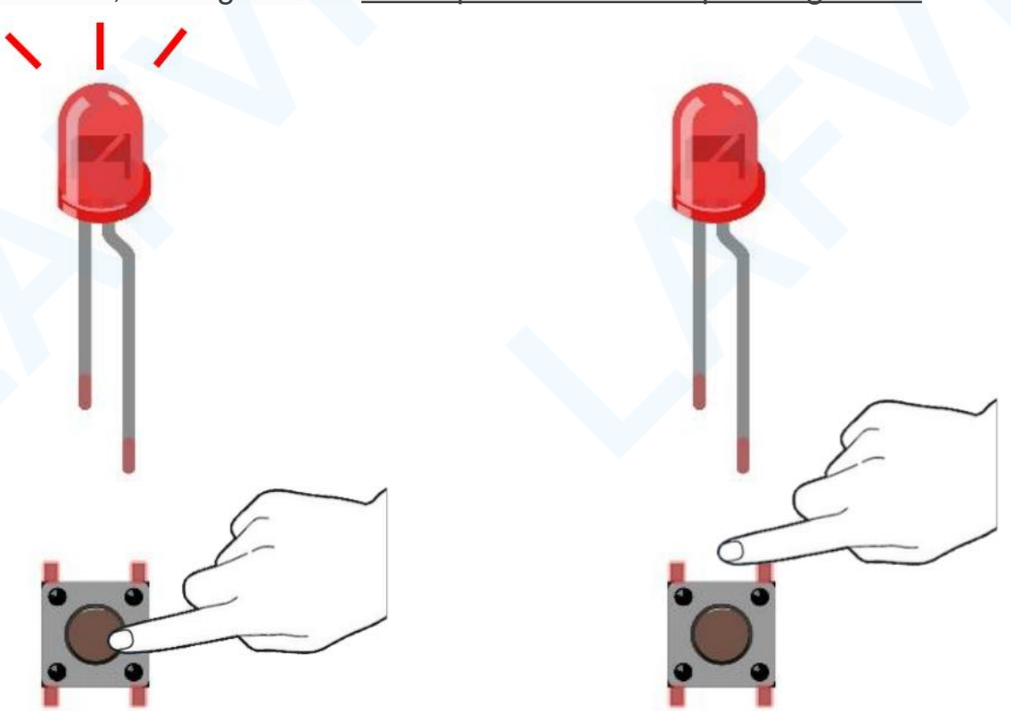
This project is designed to learn to control an LED with a push button switch. First, we need to read the state of the switch and then decide whether the LED is turned on or not based on it.

Click the **File>>Open** icon to open **Project_2_Button_Control_LED.ino** file.

*File path :**Raspberry Pi Pico Starter Kit\C\Codes***



Click  Upload the sketch to Pico.When pressing the button, LED lights up; when releasing the button, LED lights OFF.Have questions about uploading code?



How it works?

```
// initialize digital pin PIN_BUTTON as an input.
pinMode(PIN_BUTTON, INPUT);
```

For switchs, we need to set their mode to INPUT in order to be able to get their values.

```
digitalRead(PIN_BUTTON);
```

Read the status of the PIN_BUTTON in loop().

- digitalRead()

Project 3 PWM Control LED

So far, we have used only two output signals: high level and low level (or called 1 & 0, ON & OFF), which is called **digital output**. However, in actual use, many devices do not simply ON/OFF to work, for example, adjusting the speed of the motor, adjusting the brightness of the desk lamp, and so on. In the past, a slider that can adjust the resistance was used to achieve this goal, but this is always unreliable and inefficient. Therefore, Pulse width modulation (PWM) has emerged as a feasible solution to such complex problems.Knowledge point: **PWM signal output**.

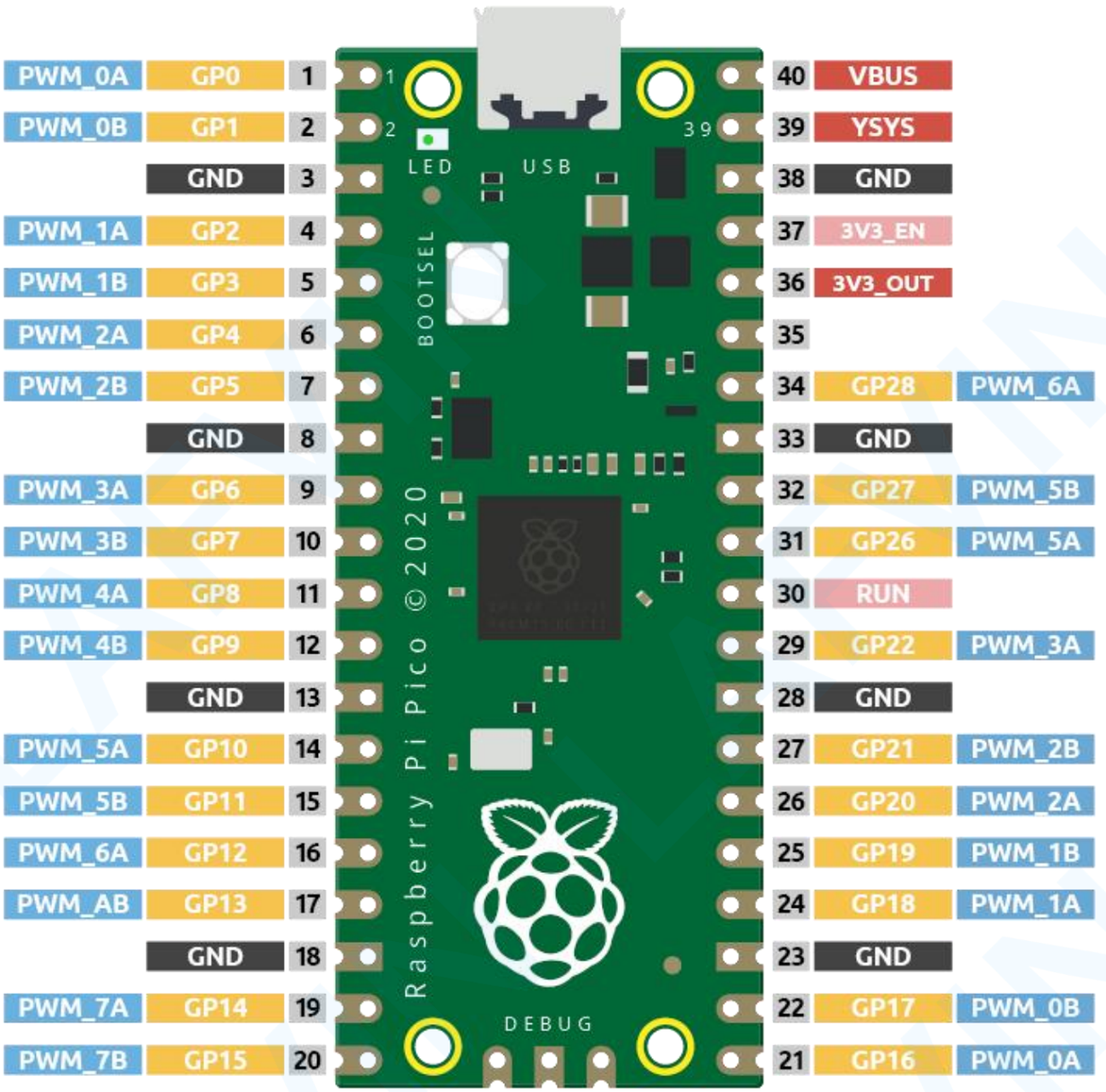
PWM signal

A digital output composed of a high level and a low level is called a pulse. The pulse width of these pins can be adjusted by changing the ON/OFF speed.Common processors cannot directly output analog signals. PWM technology makes it very convenient to achieve this conversion **(translation of digital to analog signals)**

Simply put, when we are in a short period (such as 20ms, most people's visual retention time), Let the LED turn on, turn off, and turn on again, we won't see it has been turned off, but the brightness of the light will be slightly weaker. During this period, the more time the LED is turned on, the higher the brightness of the LED. In other words, in the cycle, the wider the pulse, the greater the "electric signal strength" output by the microcontroller. This is how PWM controls LED brightness (or motor speed).

- Pulse-width modulation - Wikipedia

There are some points to pay attention to when Pico uses PWM. Let's take a look at this picture.

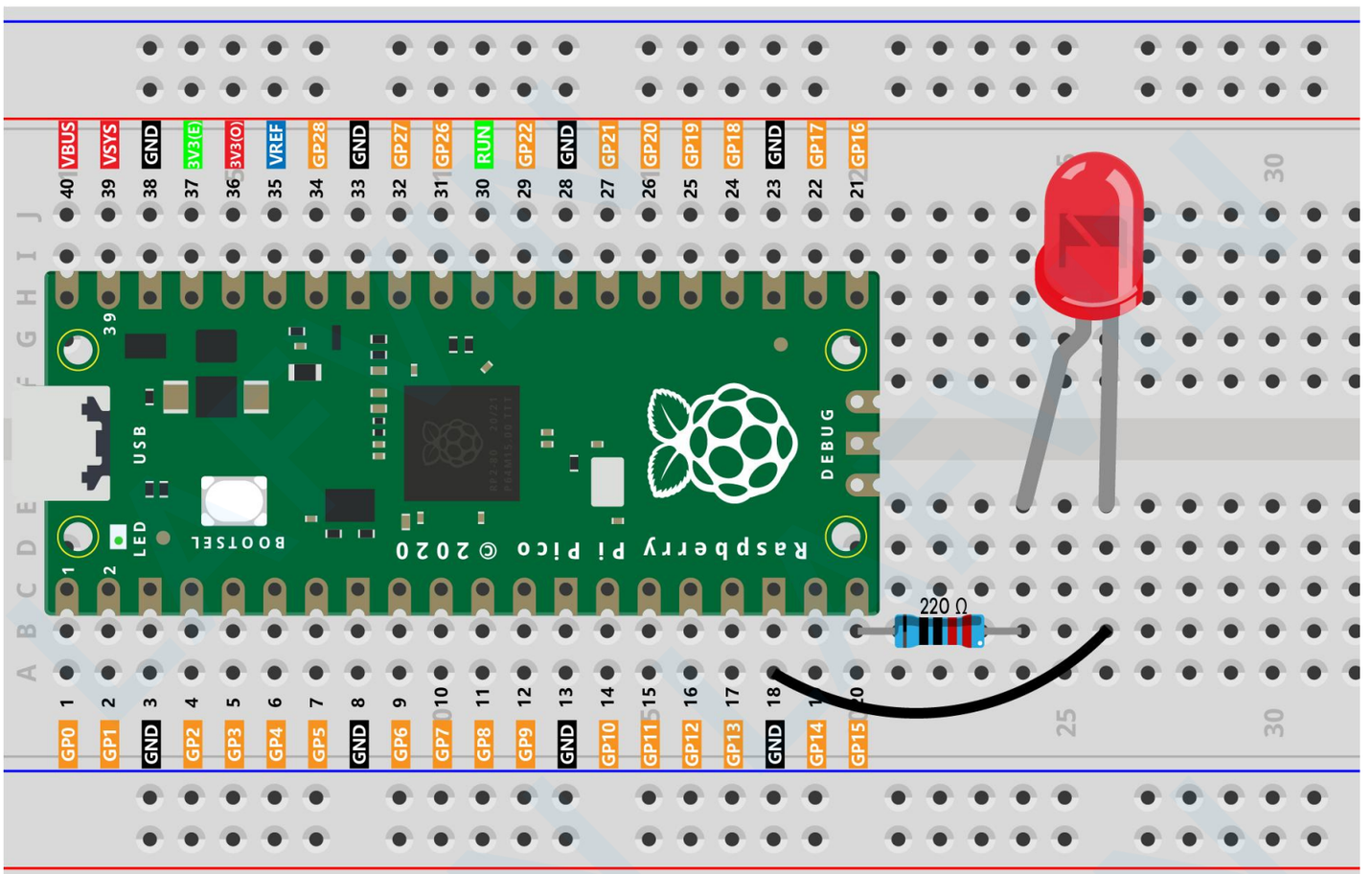


Each GPIO pin of Pico supports PWM, but it actually has a total of 16 independent **PWM outputs** (instead of 30), distributed between GP0 to GP15 on the left, and the PWM output of the right GPIO is equivalent to the left copy.

What we need to pay attention to is to avoid setting the same PWM channel for different purposes during programming. (For example, GP0 and GP16 are both PWM_0A)

After understanding this knowledge, let us try to achieve the effect of Fading LED.

Wiring

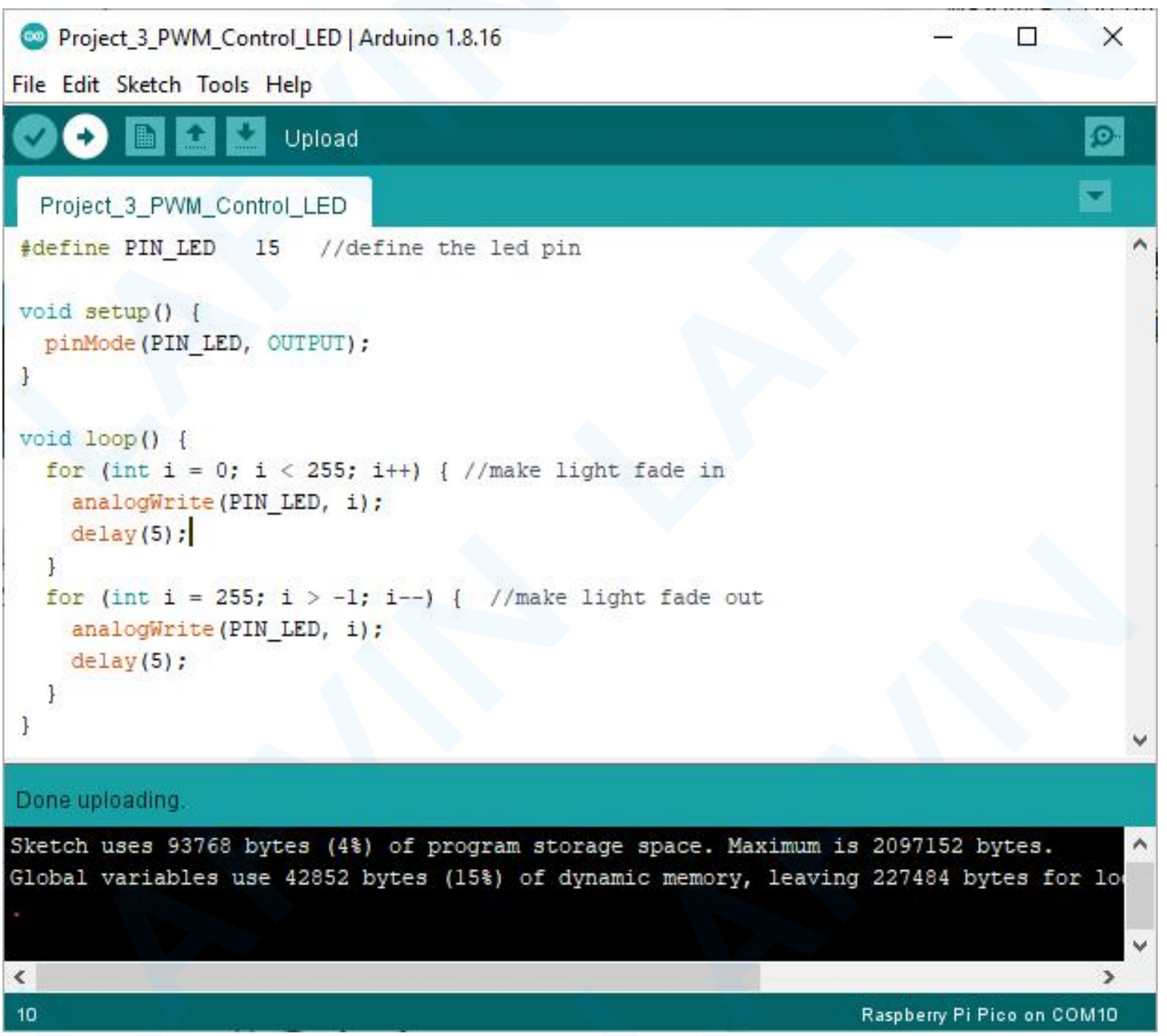

VBUS
VSYS
GND
3V3(E)
3V3(O)
VREF
GP28
GND
GP27
GP26
RUN
GP22
GND
GP21
GP20
GP19
GP18
GND
GP17
GP16
40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21
USB
LED
BOOTSEL
DEBUG
Raspberry Pi Pico ©2020
220 Ω
GP0 GP1 GND GP2 GP3 GP4 GP5 GND GP6 GP7 GP8 GP9 GND GP10 GP11 GP12 GP13 GND GP14 GP15
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
A B C D E F G H I J
25
30

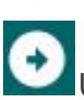
Code

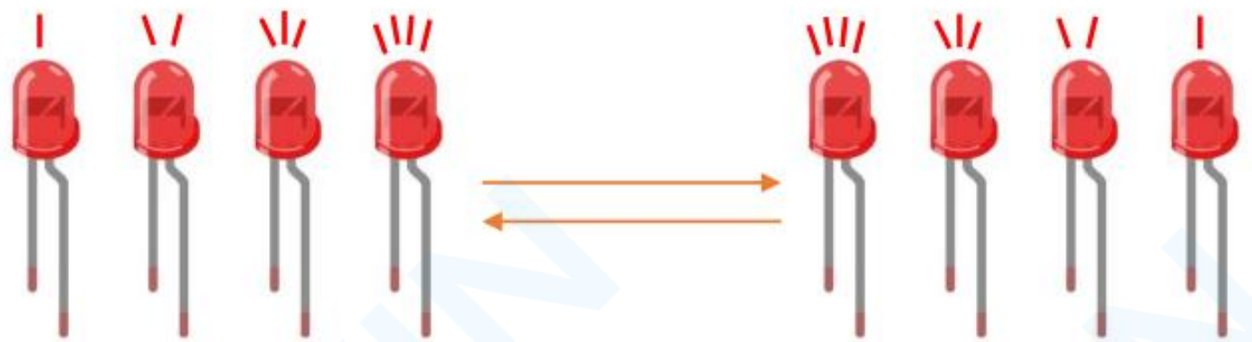
This project is designed to make PWM output GP15 with pulse width increasing from 0% to 100%, and then reducing from 100% to 0% gradually.

Click the **File>>Open** icon to open **Project_3_PWM_Control_LED.ino** file.

*File path :**Raspberry Pi Pico Starter Kit\C\Codes***



Click  Upload the sketch to Pico.you'll see that LED is turned from ON to OFF and then back from OFF to ON gradually like breathing.Have questions about uploading code?



How it works?

```
pinMode(PIN_BUTTON, OUTPUT);
```

Set the pin controlling LED to output mode.

```
analogWrite(pin, value);
```

Arduino IDE provides the function, analogWrite(pin, value), which can make ports directly output PWM waves. Every pin on Pico board can be configured to output PWM. In the function called analogWrite(pin,value), the parameter "pin" specifies the port used to output PWM wave. The range of value is 0-255, which represents the duty cycle of 0%-100%.

In order to use this function, we need to set the port to output mode.

Project 4 RGB LED

As we know, light can be superimposed. For example, mix blue light and green light give cyan light, red light and green light give yellow light. This is called “The additive method of color mixing”.

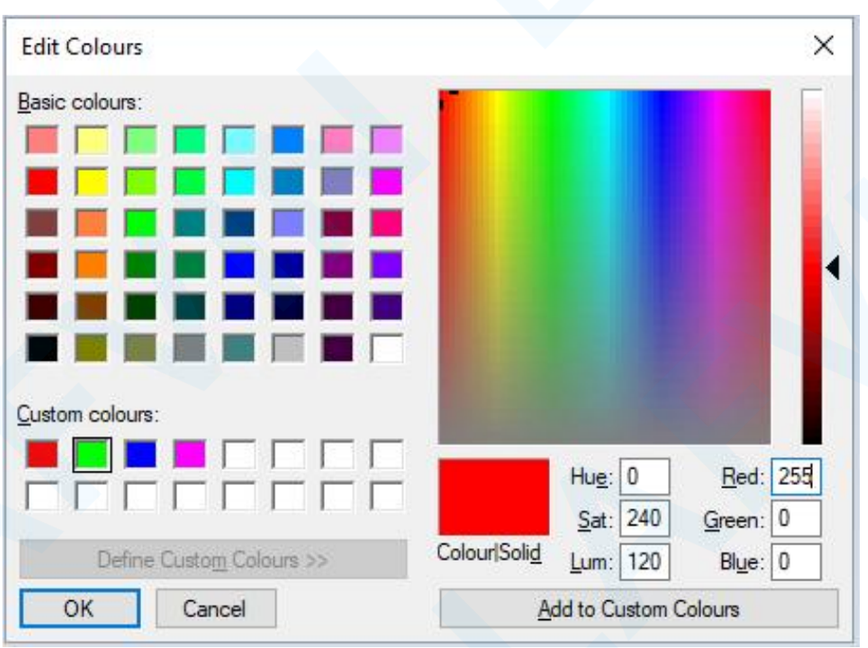
- Additive color - Wikipedia

Based on this method, we can use the three primary colors to mix the visible light of any color according to different specific gravity. For example, orange can be produced by more red and less green.

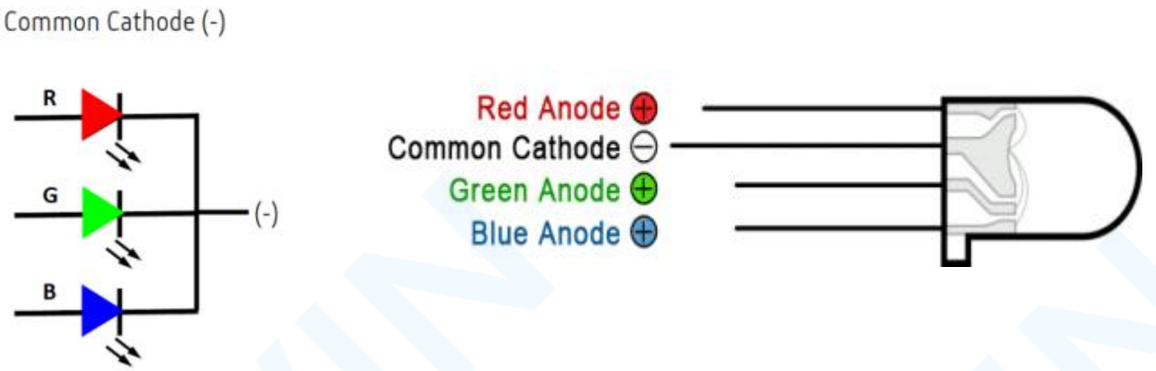
In this chapter, we will use RGB LED to explore the mystery of additive color mixing!

RGB LED is equivalent to encapsulating Red LED, Green LED, Blue LED under one lamp cap, and the three LEDs share one cathode pin. Since the electric signal is provided for each anode pin, the light of the corresponding color can be displayed. By changing the electrical signal intensity of each anode, it can be made to produce various colors.

Here, we can choose our favorite color in drawing software (such as paint) and display it with RGB LED.

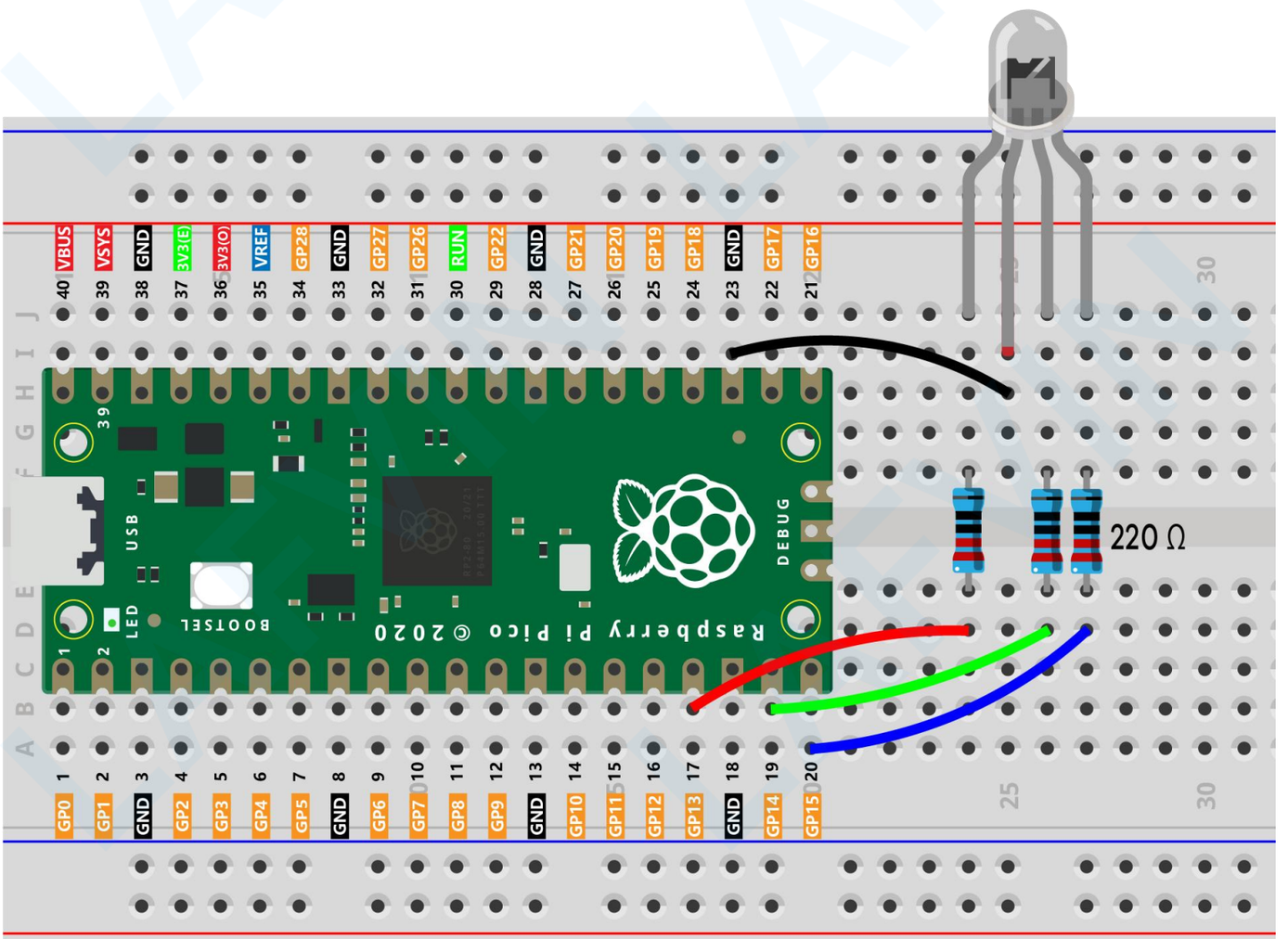


Put the RGB LED flat on the table, we can see that it has 4 leads of different lengths. Find the longest one (GND) and turn it sideways to the left. Now, the order of the four leads is Red, GND, Green, Blue from left to right.



Tip:Learn more about RGB LED

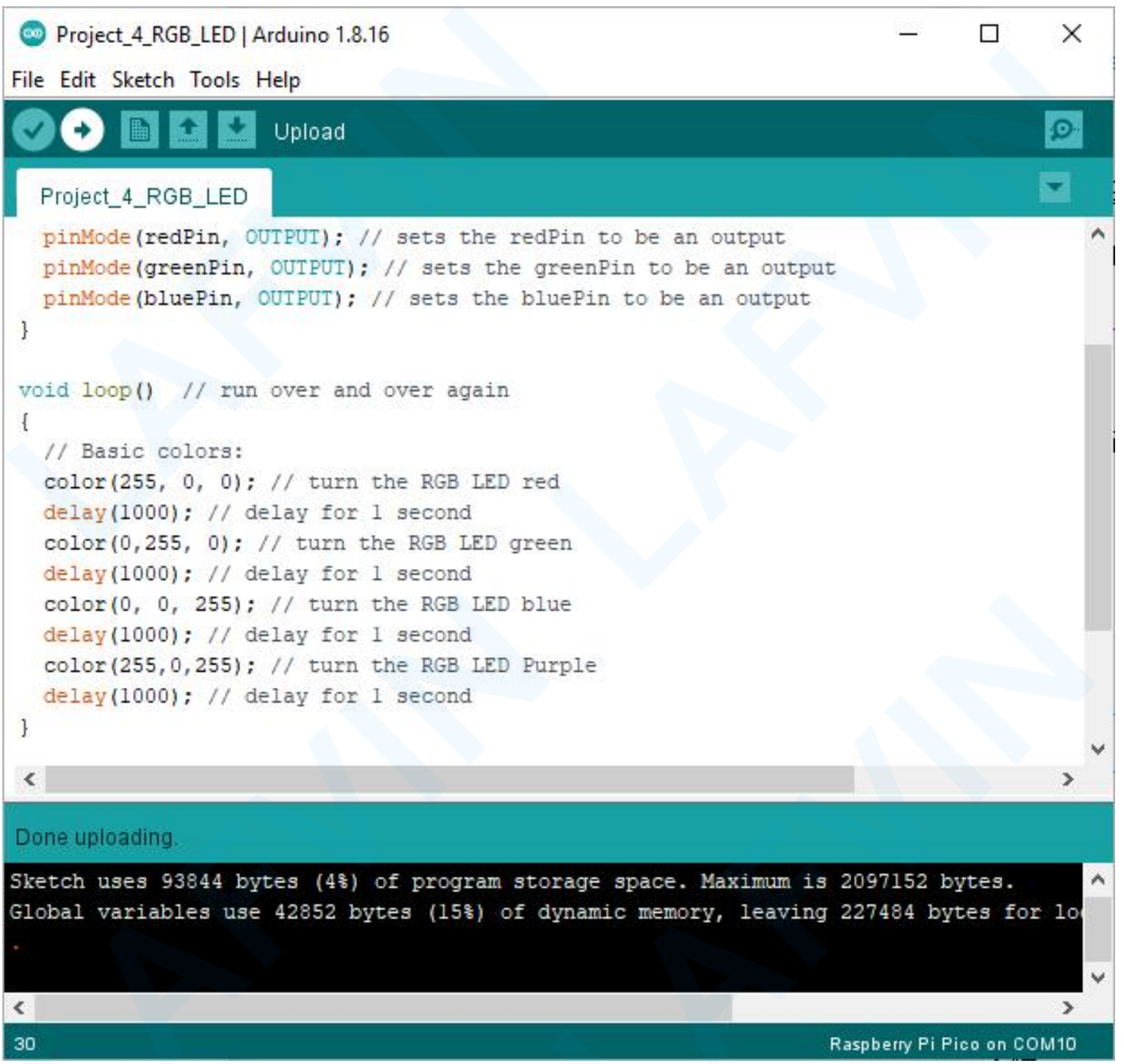
Wiring



Code

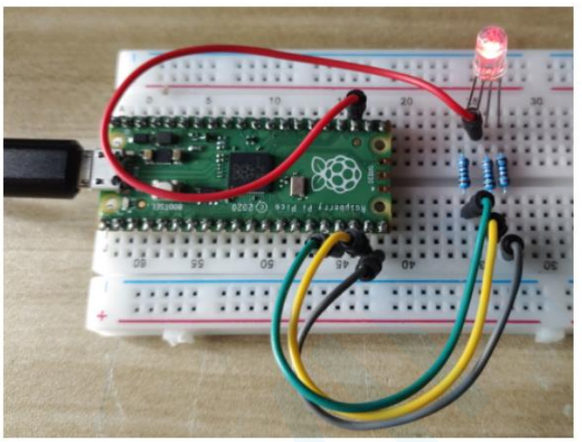
Click the **File>>Open** icon to open **Project_4_RGB_LED.ino** file.

*File path :**Raspberry Pi Pico Starter Kit\C\Codes***

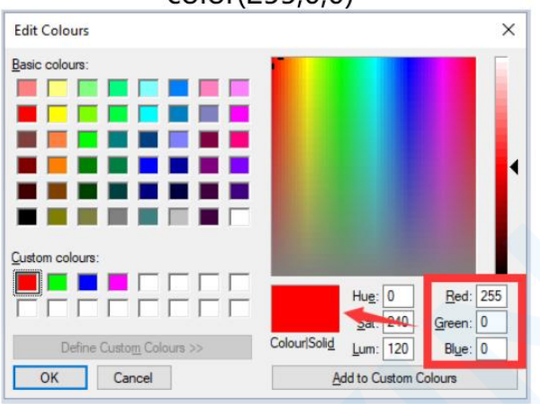


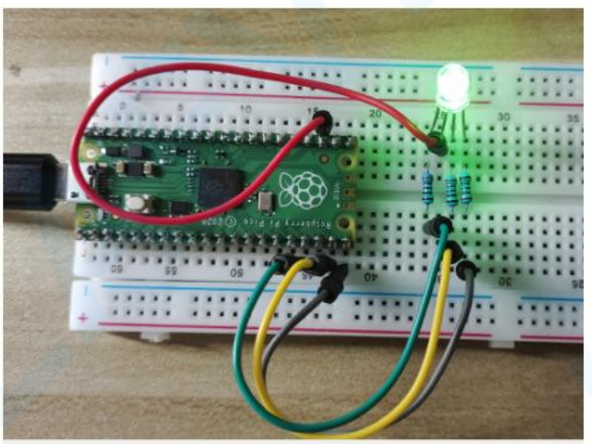
Click  Upload the sketch to Pico.The RGB light up the colors:Red->Green->Blue->Purple.

Have questions about uploading code?

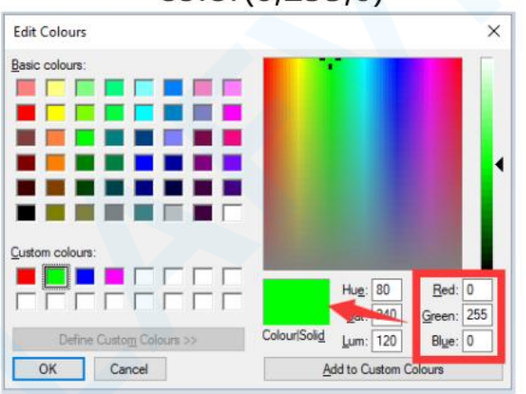


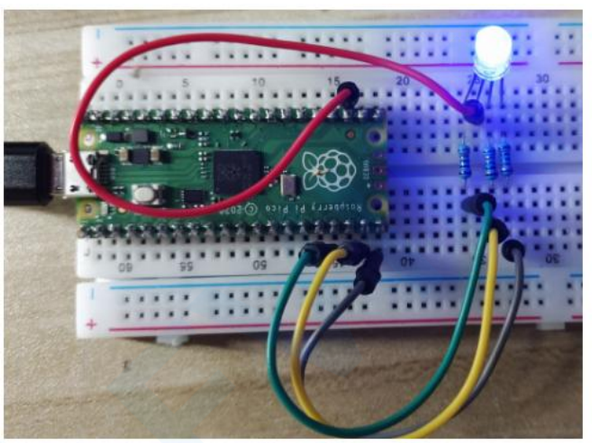
color(255,0,0)



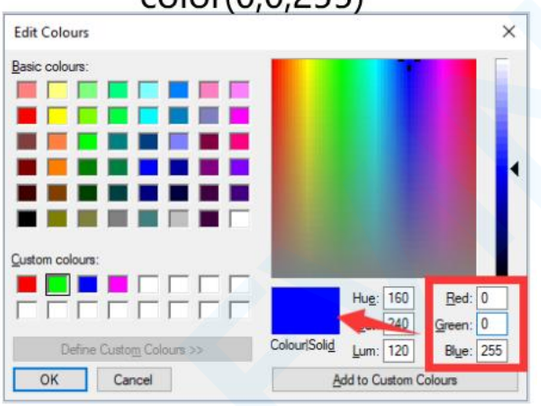


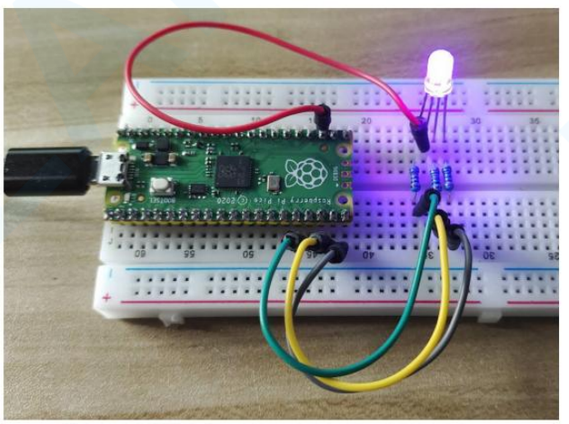
color(0,255,0)



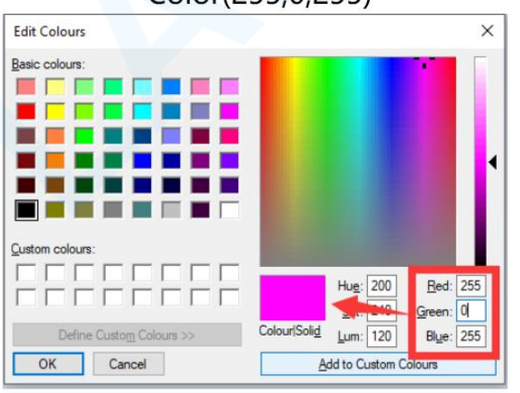


color(0,0,255)





Color(255,0,255)



Project 5 Custom Tone

In this project I use a passive buzzer to make different note sounds.

Component Knowledge

Buzzer

the passive buzzer also uses the phenomenon of electromagnetic induction to work. The difference is that a passive buzzer does not have oscillating source, so it will not beep if DC signals are used. But this allows the passive buzzer to adjust its own oscillation frequency and can emit different notes such as “doh, re, mi, fa, sol, la, ti”.

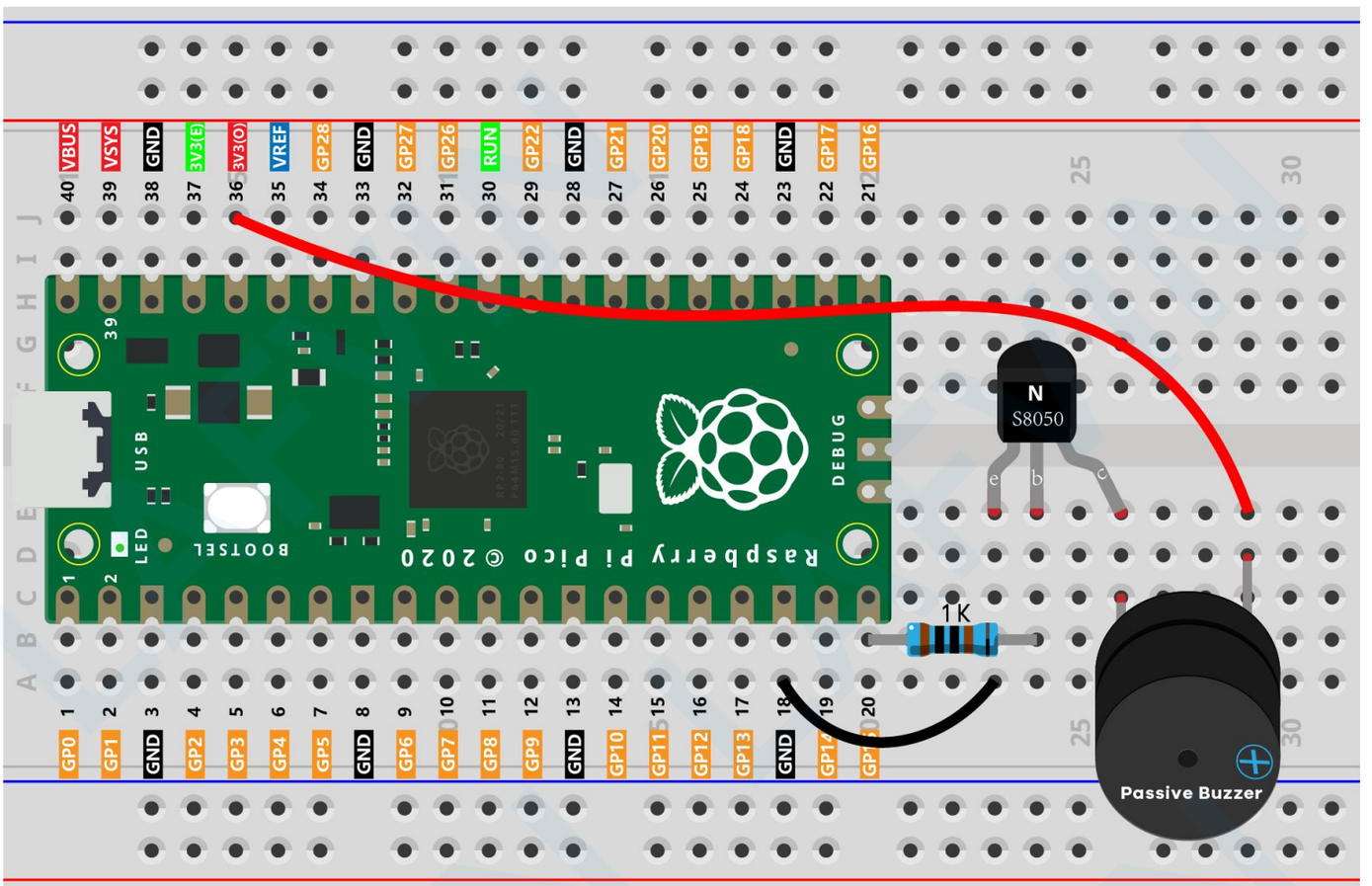
Tip:Learn more about Buzzer

Transistor

In simple words, some components use high-current (such as Buzzer). If the power is directly supplied from the GPIO of the microcontroller, the power may be insufficient or the microcontroller may be damaged. Then, the transistor has played a “dam” role here. Transistor receives the weak electrical signal from the GPIO pin to control the turn-on and turn-off of the large current (from VCC to GND). In this way, high-current components can be driven and the microcontroller can be protected.

Tip:Learn more about Transistor

Wiring



Note:

①Two buzzers are included in the kit, we use the one with exposed PCB behind.

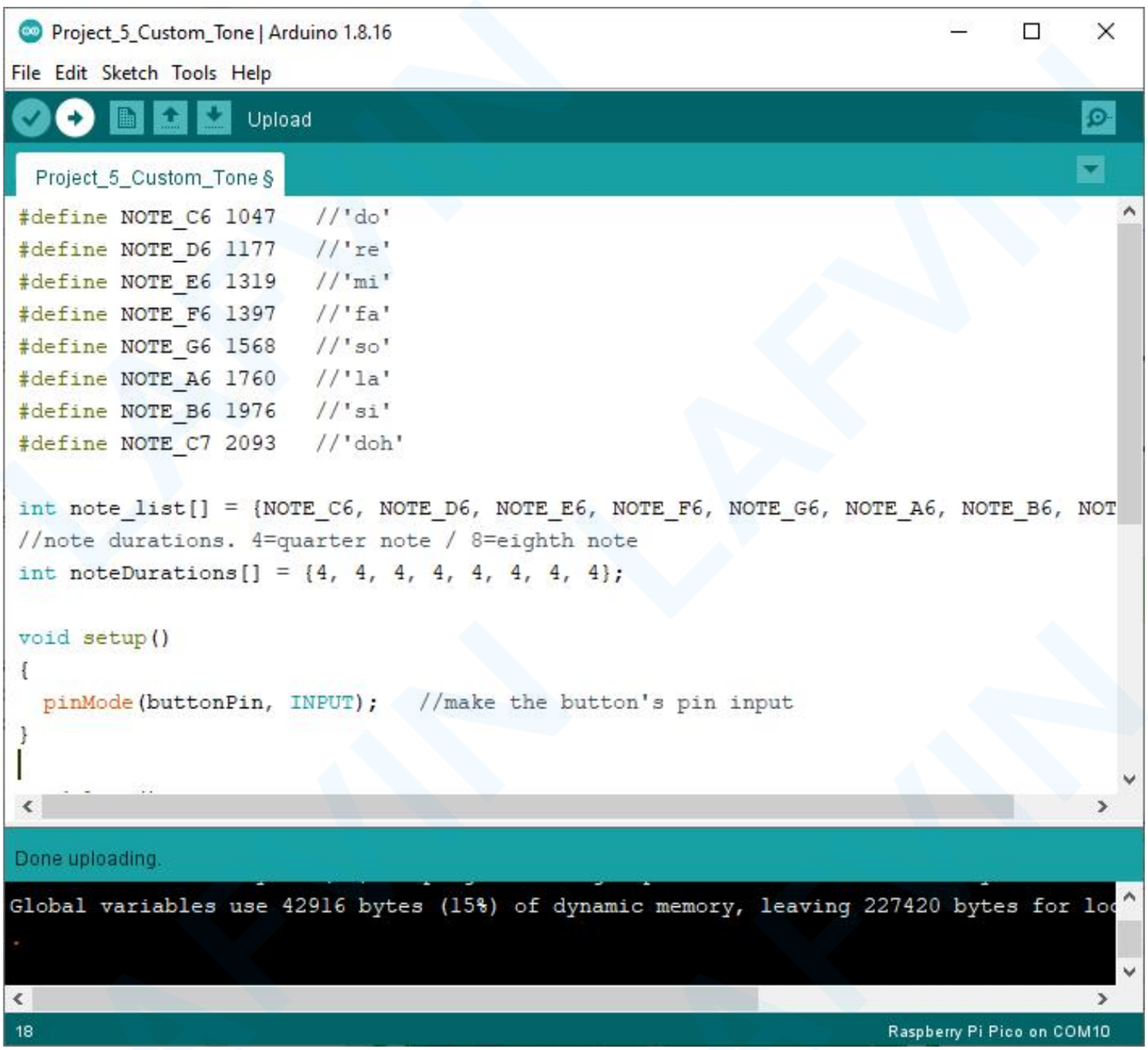
②The buzzer needs a transistor to work, and here we use S8050.

③The 1K resistor between the NPN transistor and GP15 is used for current limiting when the transistor is energized.

Code

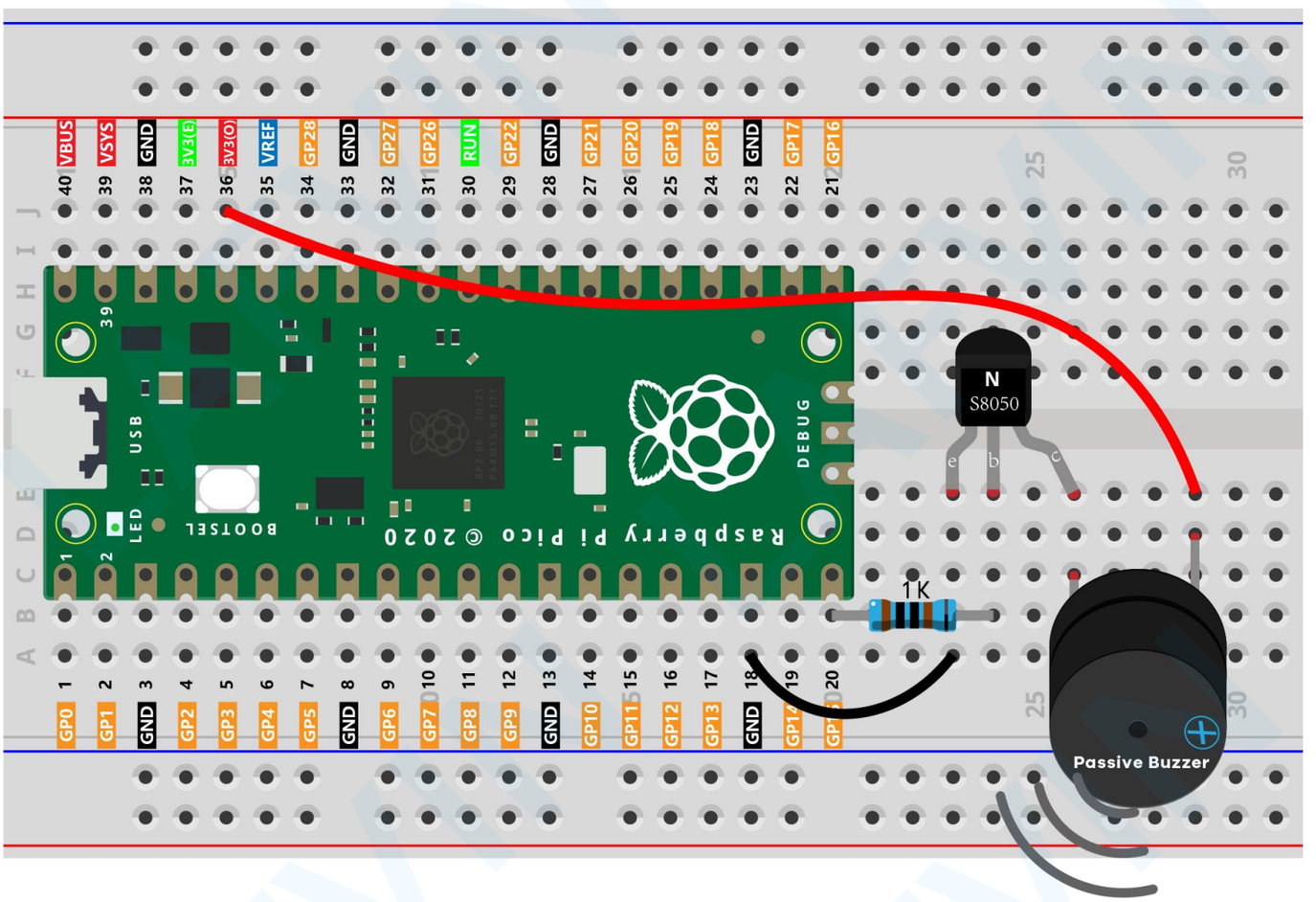
Click the **File>>Open** icon to open **Project_5_Custom_Tone.ino** file.

*File path :**Raspberry Pi Pico Starter Kit\C\Codes***



Click  Upload the sketch to Pico. Buzzer will play note:"do, re, mi, fa, so, la, si, doh".

Have questions about uploading code?



How it works?

```
int note_list[] = {NOTE_C6, NOTE_D6, NOTE_E6, NOTE_F6, NOTE_G6, NOTE_A6,
NOTE_B6, NOTE_C7};
//note durations. 4=quarter note / 8=eighth note
int noteDurations[] = {4, 4, 4, 4, 4, 4, 4, 4};
```

The array note_list[] stores 7 notes, and noteDurations[] is the duration corresponding to these notes, 4=quarter note, 8=eighth note.

- Quarter note
- Eighth note

```
for (int i = 0; i < 8; i++)
    {
      // to calculate the note duration, take one second divided by the note type.
      //e.g. quarter note = 1000 / 4, eighth note = 1000/8, etc.
      int noteDuration = 1000 / noteDurations [i];
      tone(14, note_list [i], noteDuration);
      //to distinguish the notes, set a minimum time between them
      //the note's duration +30% seems to work well
      int pauseBetweenNotes = noteDuration * 1.30;
      delay(pauseBetweenNotes);
    }
```

In the for() statement, a tone() is used to let the buzzer play one note at a time, and then after 8 times, the buzzer can play the notes in the array note_list[] one by one.

```
tone(14, note_list [i], noteDuration);
```

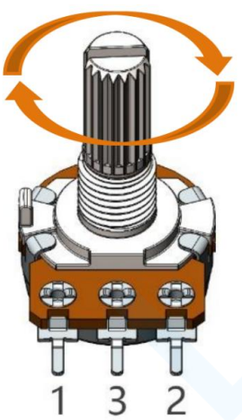
14: The pin on which to generate the tone (the buzzer pin).
note_list[i]: The frequency of the tone in hertz.
noteDuration: The duration of the tone in milliseconds (optional).

Project 6 Analog Input

In the previous projects, we have used the digital input on the Pico. For example, a button can change the pin from low level (off) to high level (on). This is a binary working state.

However, Pico can receive another type of input signal: analog input. It can be in any state from fully closed to fully open, and has a range of possible values. The analog input allows the microcontroller to sense the light intensity, sound intensity, temperature, humidity, etc. of the physical world.

In this project, we try to read the analog value of potentiometer.



Tip:Learn more about potentiometer

Related knowledge

ADC

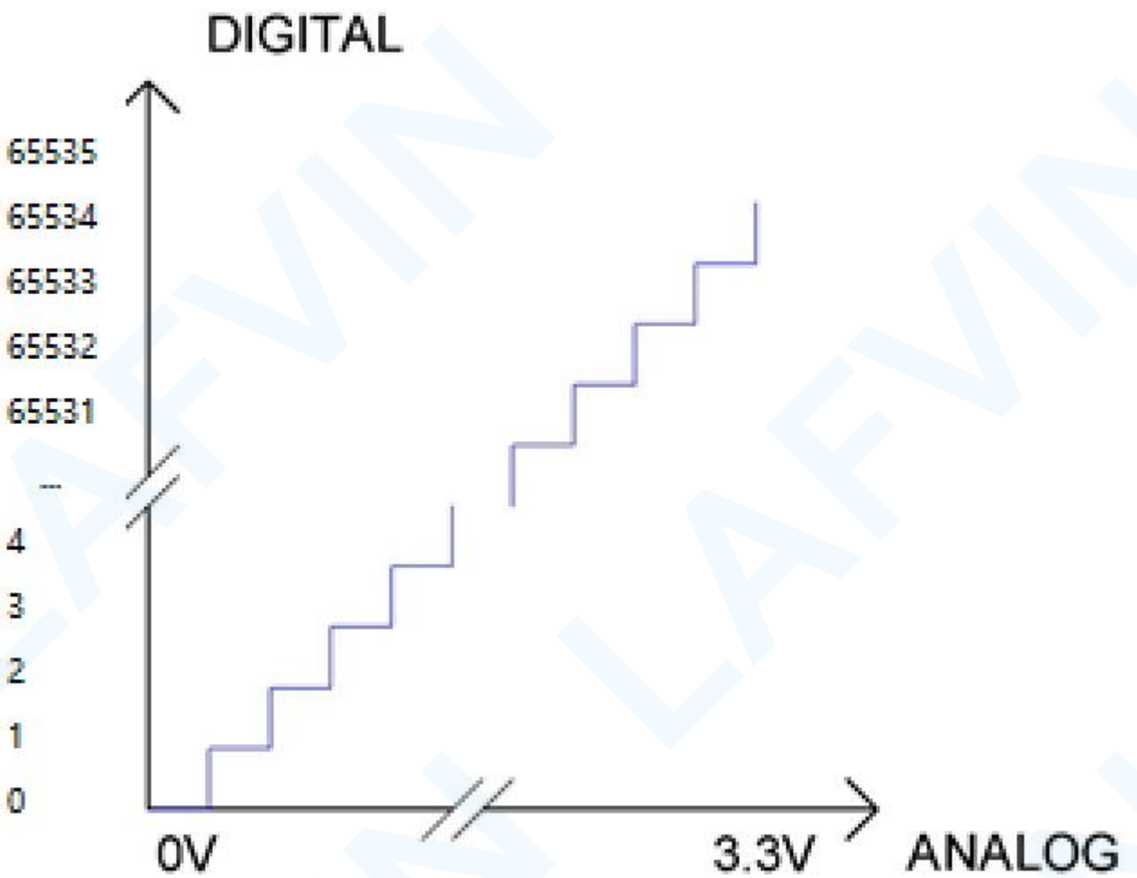
An analog-to-digital converter(ADC) converts a measured analog signal into a digital code. ADC has two key features: resolution and channels.

ADC resolution of Raspberry Pi Pico

Raspberry Pi Pico uses RP2040 chip. With a 12-bit ADC resolution, it can convert any analog signal to digital signal, ranging from 0-4095. For example, if the analog voltage range it measures is 0-3.3V, the ADC can divide it into 4096 equal parts.

However, when using Micropython firmware to call Raspberry Pi Pico ADC, the digital signal it obtains ranges from 0 to 65535. This is because Micropython internally processes Pico's ADC resolution to 16 bits, and the values is also changed to 0-65535, so as to make it the same as the ADC of other Micropython

microcontrollers. The way that ADC converts doesn't change but only the resolution changes. So if the measured analog voltage ranges from 0-3.3V, ADC can devide it into 65536 equal parts.



Subsection 1: the analog in a range of 0V---3.3/65535 V corresponds to digital 0;

Subsection 2: the analog in a range of 3.3/65535 V---2*3.3 /65535 V corresponds to digital 1;

…

Subsection 65535: the analog in range of 65534*3.3/65535 V---65535*3.3 /65535 V corresponds to digital

65534;

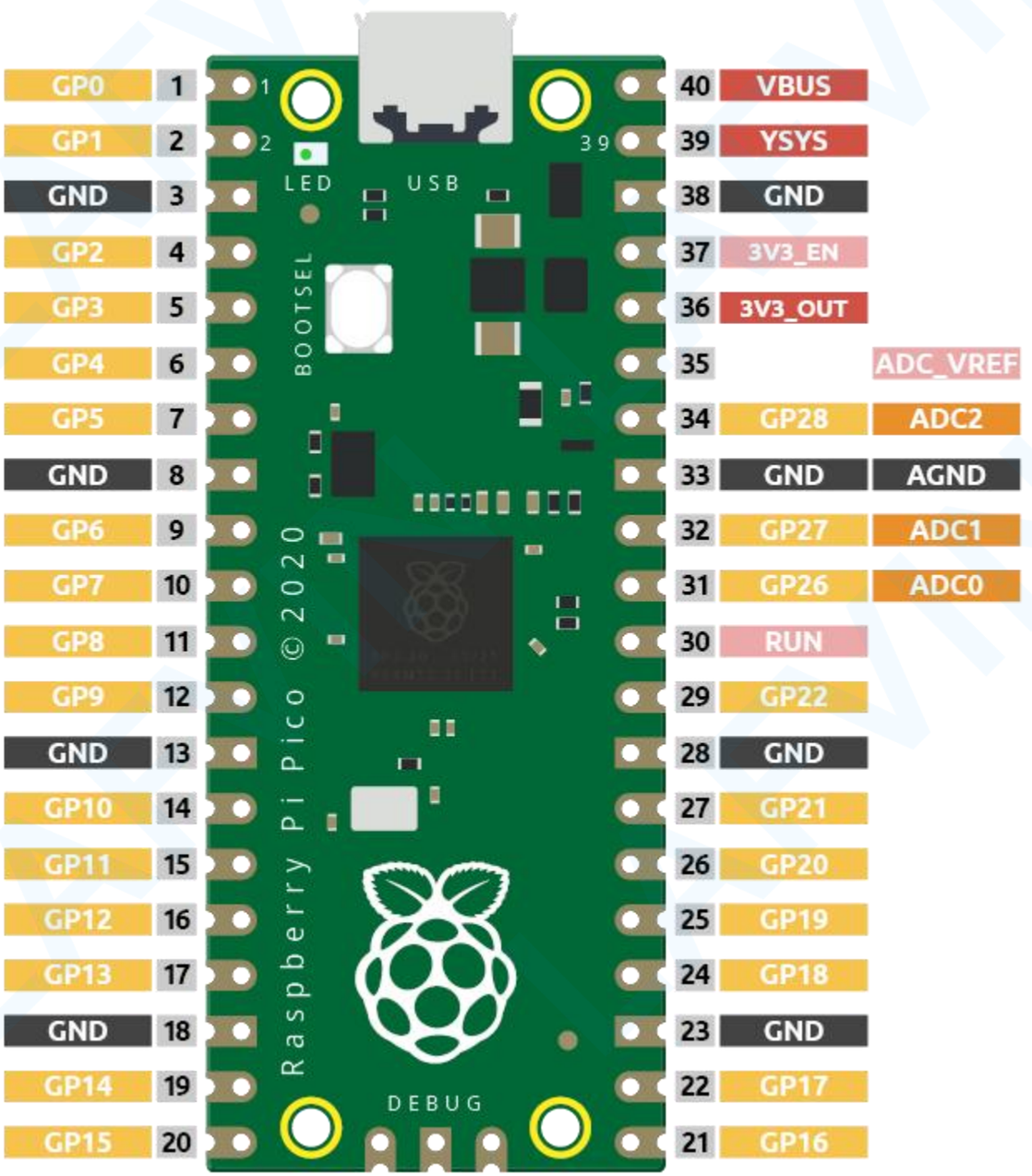
The following analog will be divided accordingly.

The conversion formula is as follows:

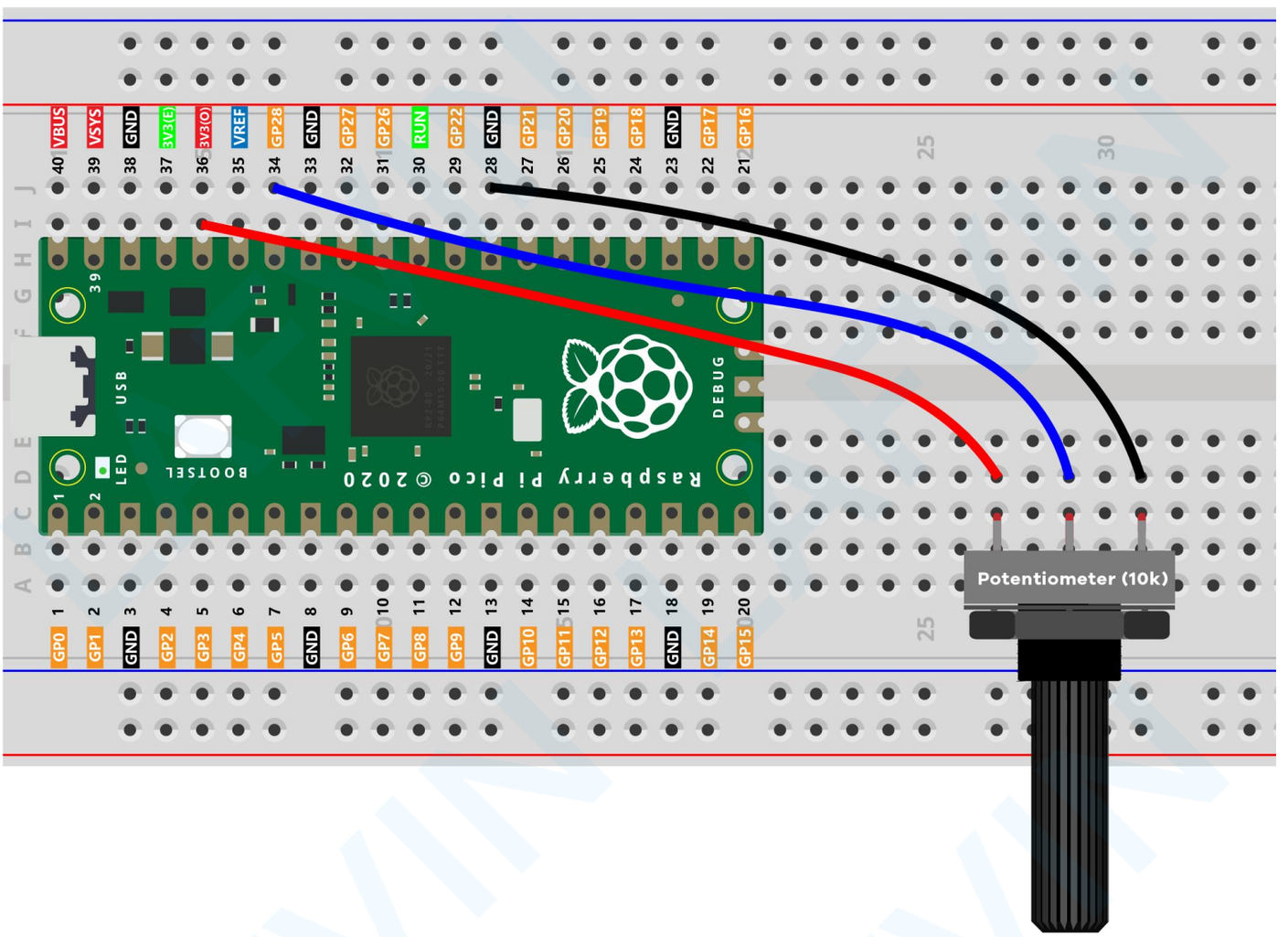
$$ADCValue = \frac{\text{Analog Voltage}}{3.3} * 65535$$

ADC ChannelsRaspberry Pi Pico

Raspberry Pi Pico has 5 ADC channels, which are ADC0(GP26), ADC1(GP27), ADC2(GP28), ADC3(GP29): used to measure VSYS on Pico board, and ADC4, which directly connects to the built-in temperature sensor of RP2040 chip. Therefore, there are only three generic ADC channels that can be directly used, namely, ADC0,ADC1 and ADC2.



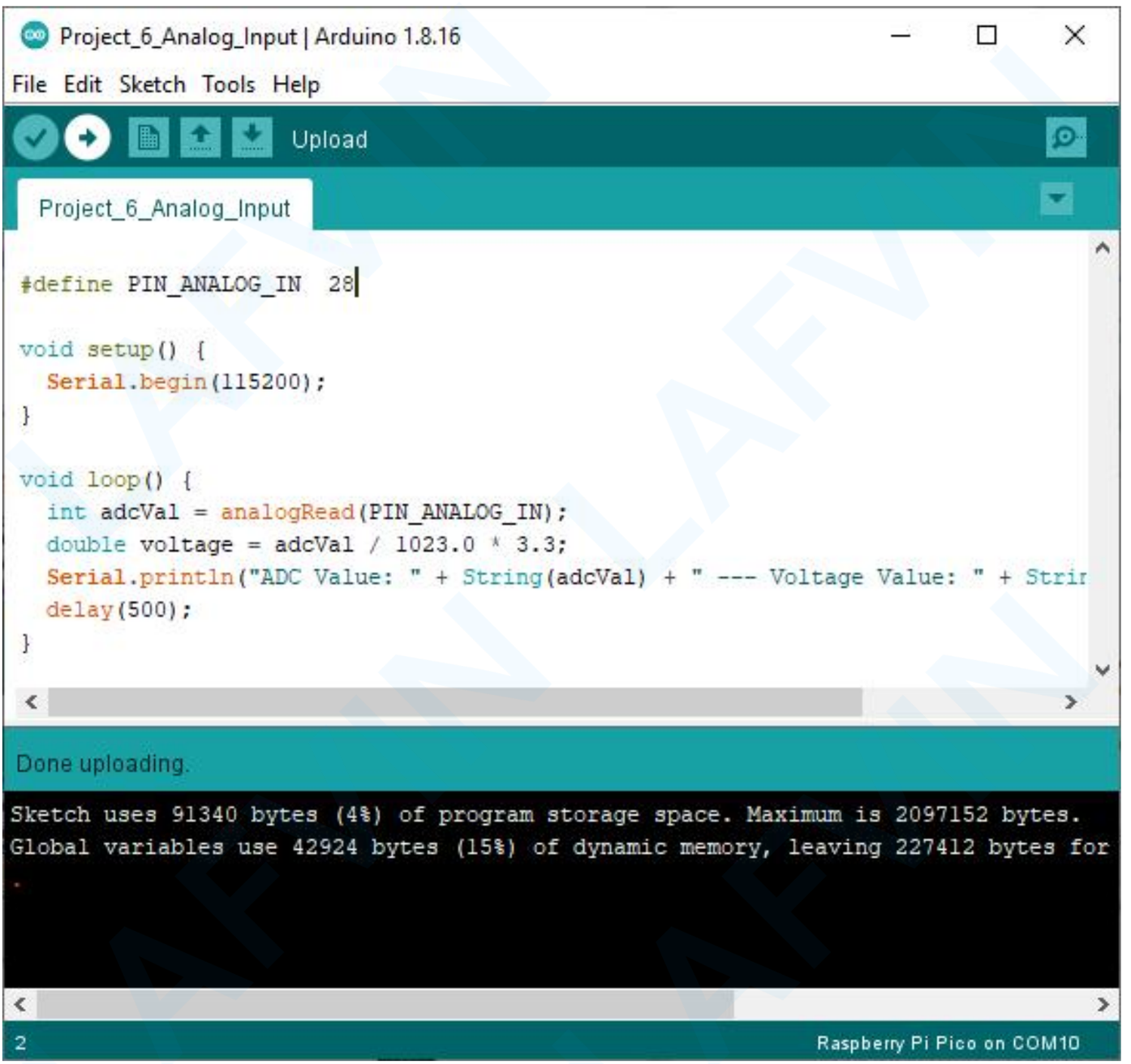
Wiring


VBUS
VSYS
GND
3V3(E)
3V3(O)
VREF
GP28
GND
GP27
GP26
RUN
GP22
GND
GP21
GP20
GP19
GP18
GND
GP17
GP16
GP0
GP1
GND
GP2
GP3
GP4
GP5
GND
GP6
GP7
GP8
GP9
GND
GP10
GP11
GP12
GP13
GND
GP14
GP15
USB
LED
BOOTSEL
DEBUG
Raspberry Pi Pico ©2020
Potentiometer (10k)

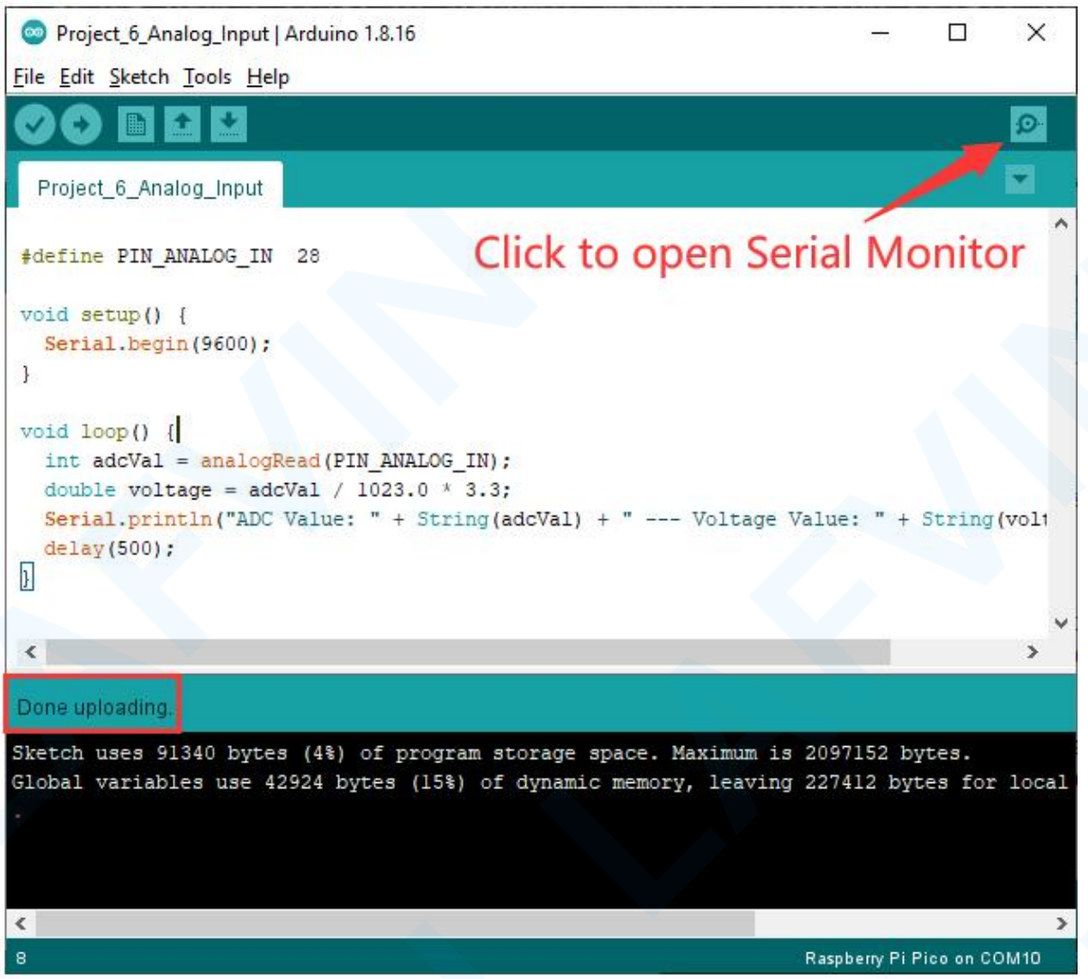
Code

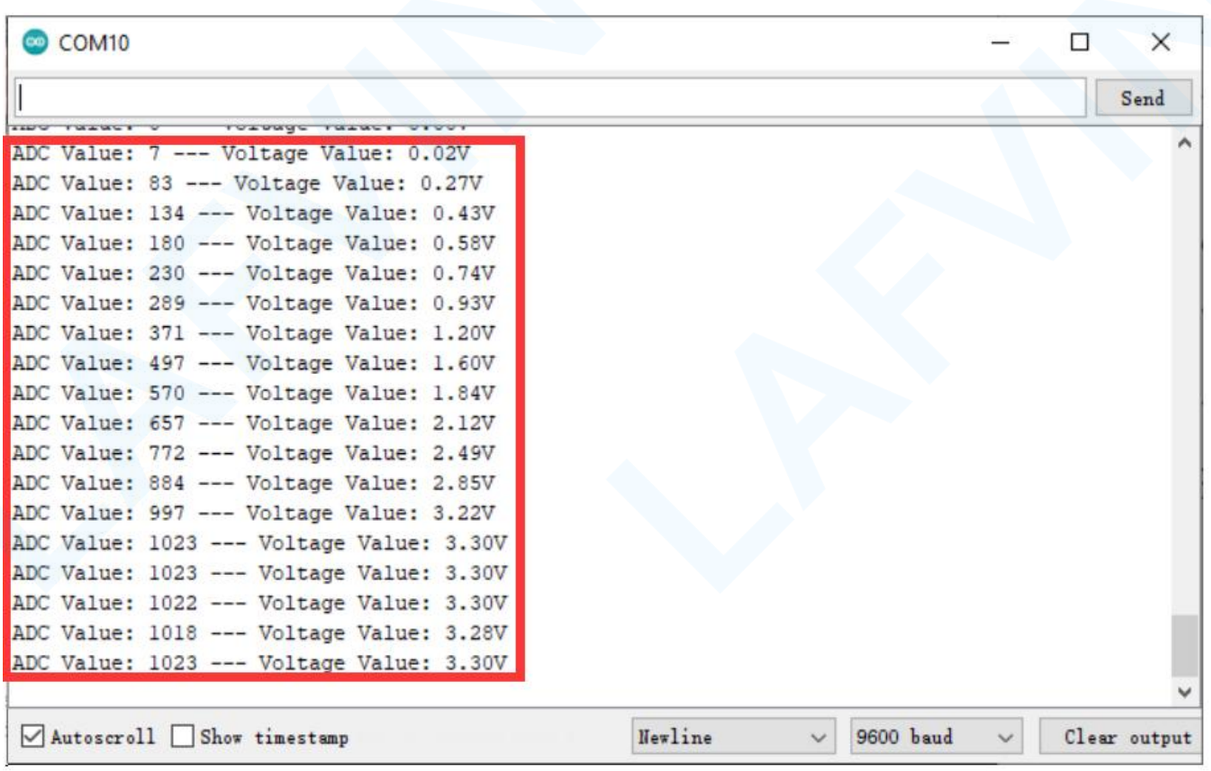
Click the **File>>Open** icon to open **Project_6_Analog Input.ino** file.

*File path :**Raspberry Pi Pico Starter Kit\C\Codes***



Click  Upload the sketch to Pico.After Done Uploading appears. Click  to open the Serial Monitor.The ADC value will be printed to the serial monitor.Have questions about uploading code?


Project_6_Analog_Input | Arduino 1.8.16
File Edit Sketch Tools Help
Project_6_Analog_Input
Click to open Serial Monitor
#define PIN_ANALOG_IN 28
void setup() {
Serial.begin(9600);
}
void loop() {
int adcVal = analogRead(PIN_ANALOG_IN);
double voltage = adcVal / 1023.0 * 3.3;
Serial.println("ADC Value: " + String(adcVal) + " --- Voltage Value: " + String(volt
delay(500);
}
Done uploading.
Sketch uses 91340 bytes (4%) of program storage space. Maximum is 2097152 bytes.
Global variables use 42924 bytes (15%) of dynamic memory, leaving 227412 bytes for local
8
Raspberry Pi Pico on COM10


COM10
Send
ADC Value: 7 --- Voltage Value: 0.02V
ADC Value: 83 --- Voltage Value: 0.27V
ADC Value: 134 --- Voltage Value: 0.43V
ADC Value: 180 --- Voltage Value: 0.58V
ADC Value: 230 --- Voltage Value: 0.74V
ADC Value: 289 --- Voltage Value: 0.93V
ADC Value: 371 --- Voltage Value: 1.20V
ADC Value: 497 --- Voltage Value: 1.60V
ADC Value: 570 --- Voltage Value: 1.84V
ADC Value: 657 --- Voltage Value: 2.12V
ADC Value: 772 --- Voltage Value: 2.49V
ADC Value: 884 --- Voltage Value: 2.85V
ADC Value: 997 --- Voltage Value: 3.22V
ADC Value: 1023 --- Voltage Value: 3.30V
ADC Value: 1023 --- Voltage Value: 3.30V
ADC Value: 1022 --- Voltage Value: 3.30V
ADC Value: 1018 --- Voltage Value: 3.28V
ADC Value: 1023 --- Voltage Value: 3.30V
Autoscroll
Show timestamp
Newline
9600 baud
Clear output

How it works?

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

To enable Serial Monitor, you need to start serial communication in setup() and set the datarate to 9600.

- Serial

```
analogRead(PIN_ANALOG_IN);
```

Reads the value from the specified analog pin. Return the analog reading on the pin. (0-1023 for 10 bits).

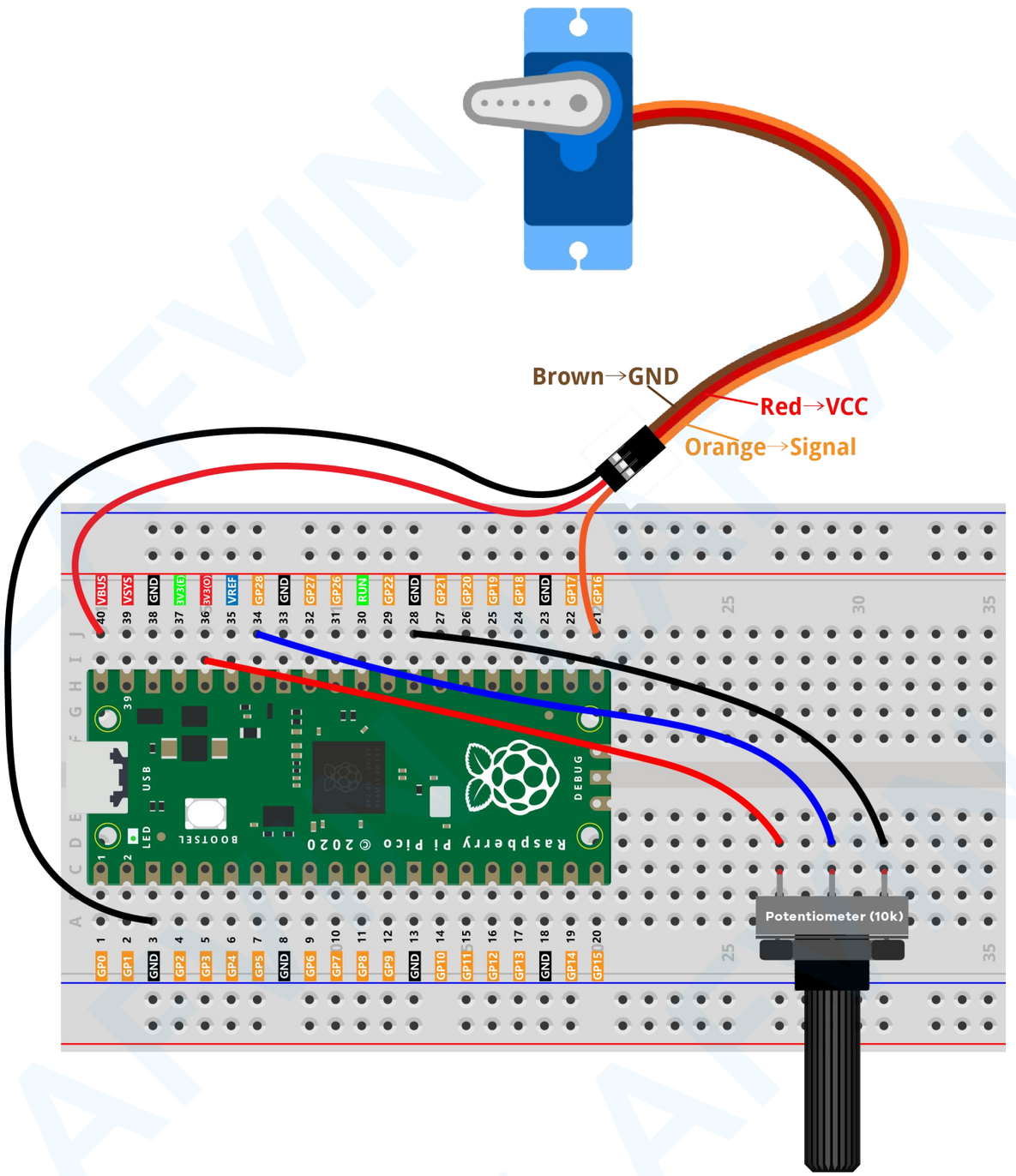
Project 7 Potentiometer Control Servo

We have learned ADC analog input and PWM analog output concept. This time, we use some new analog input device and PWM output device.

In this project, we will show you how to read analog signal from a potentiometer(adjustable resistor) through a Pico ADC pin(GP28) . We will also connect a Servo motor to GP16 which can generate PWM signal. The servo will rotate its arm when you rotate potentiometer.

Tip:Learn more about potentiometer servo

Wiring



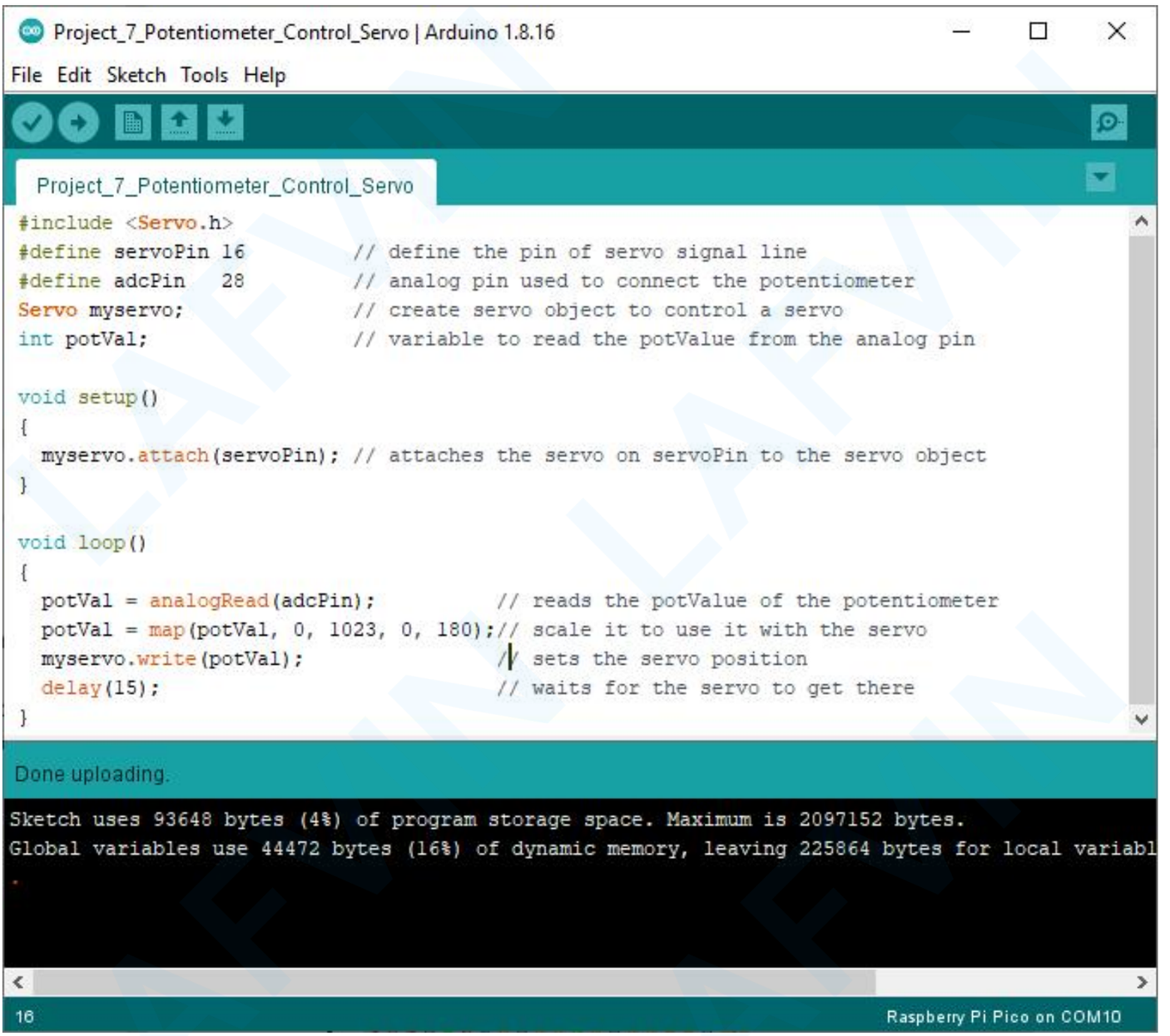
Note:

The working voltage of Servo is 5V. Use the VBUS pin to power it.

Code

Click the **File>>Open** icon to open **Project_7_Potentiometer_Control_Servo.ino** file.

*File path :**Raspberry Pi Pico Starter Kit\C\Codes***



Click  Upload the sketch to Pico and you can rotate the potentiometer, and you will see the servo arm rotate accordingly.Have questions about uploading code?

How it works?

```
#include <Servo.h>
```

Servo uses the Servo library, #include reference to Servo library

```
Servo myServo;     // create servo object to control a servo
```

Servo library provides the Servo class that controls it.Servo class must be instantiated before using.

```
myServo.attach(servoPin);          // attaches the servo on pin 16 to the servo object
```

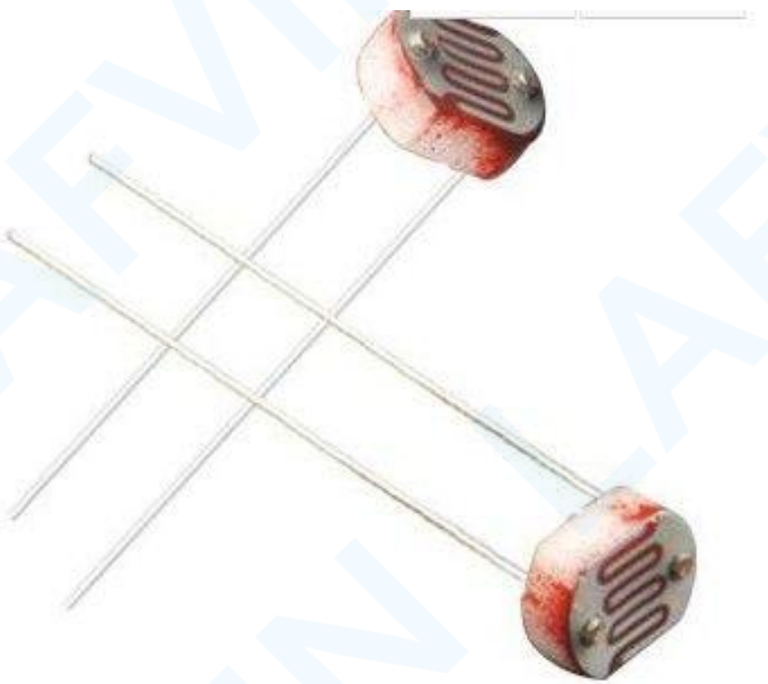
Initialize the servo, the parameter is the port connected to servo signal line;

```
myServo.write( posVal );
```

Control servo to rotate to the specified angle; parameter here is to specify the angle.

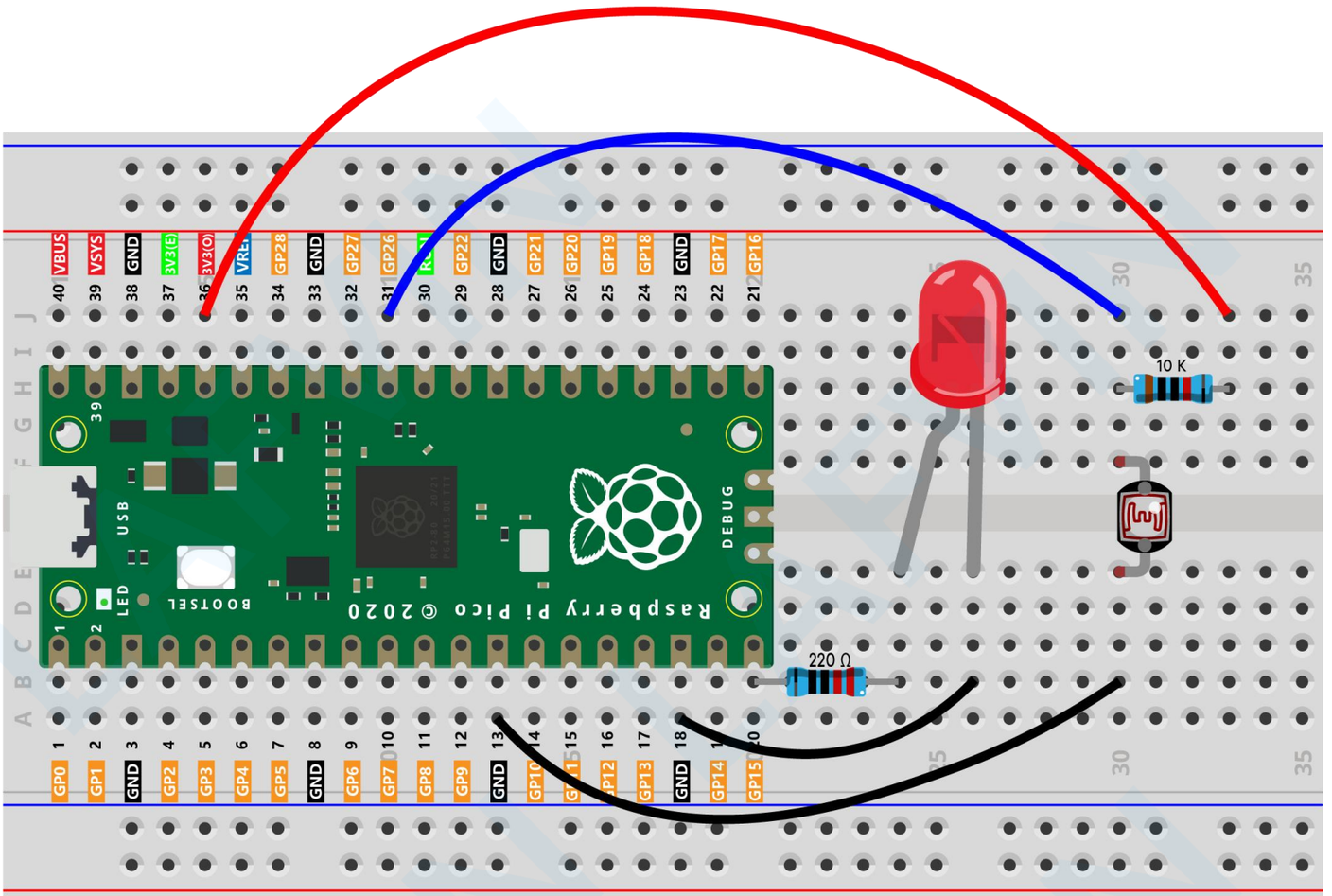
Project 8 Photoresistor Control LED

A Photoresistor is very sensitive to the amount of light present. We can take advantage of the characteristic to make a night lamp with the following function: when the ambient light is less (darker environment) the LED will automatically become brighter to compensate and when the ambient light is greater (brighter environment) the LED will automatically dim to compensate.



Tip:Learn more about Photoresistor

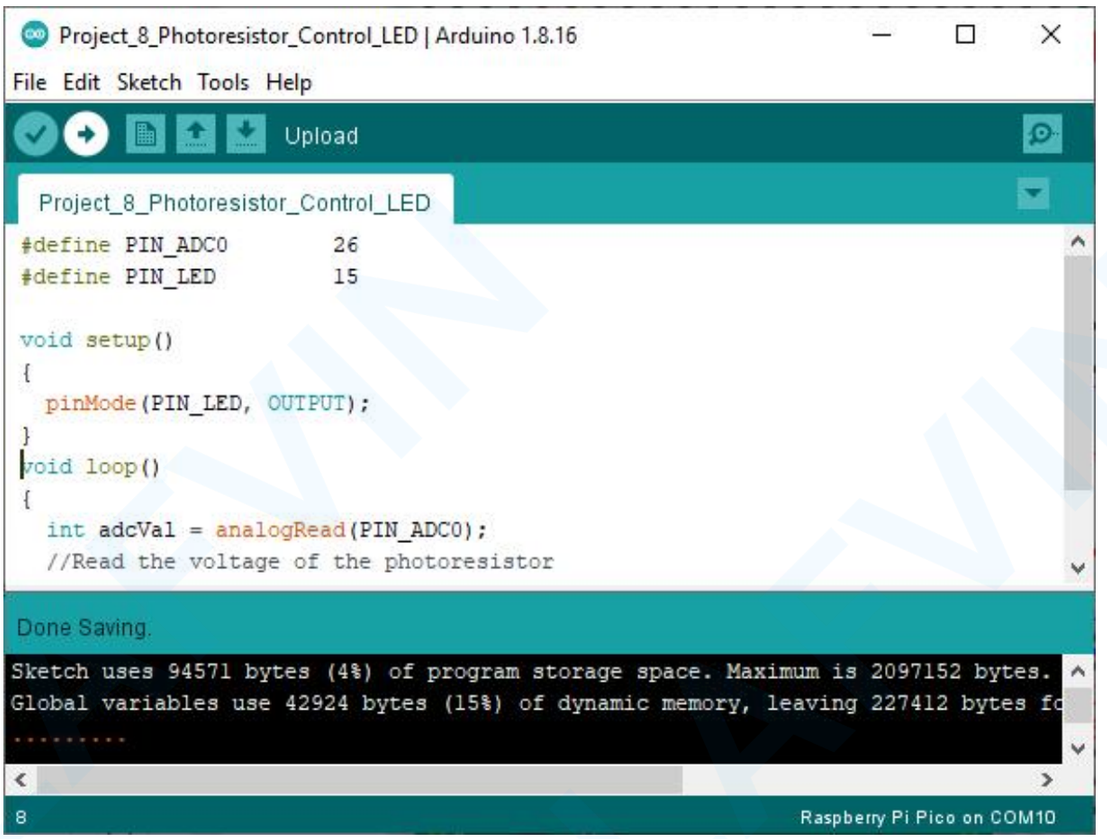
Wiring



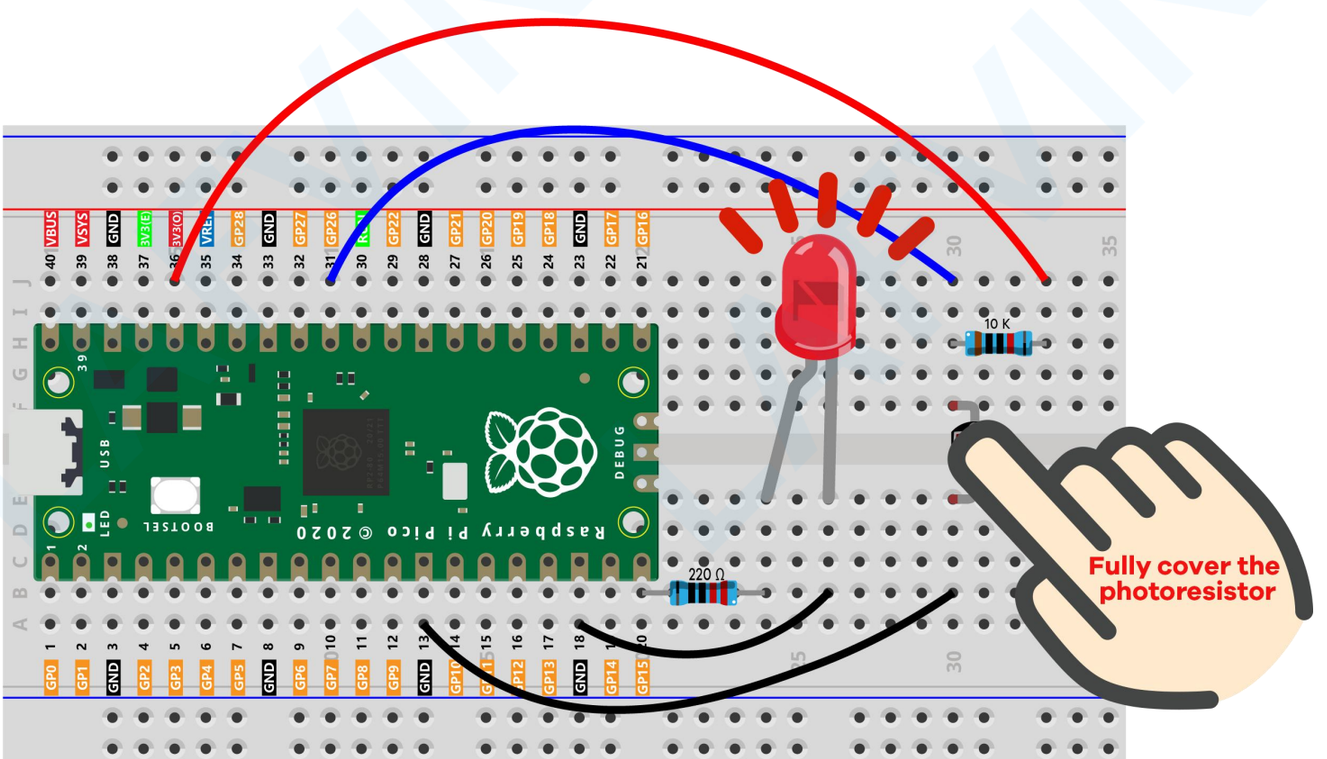
Code

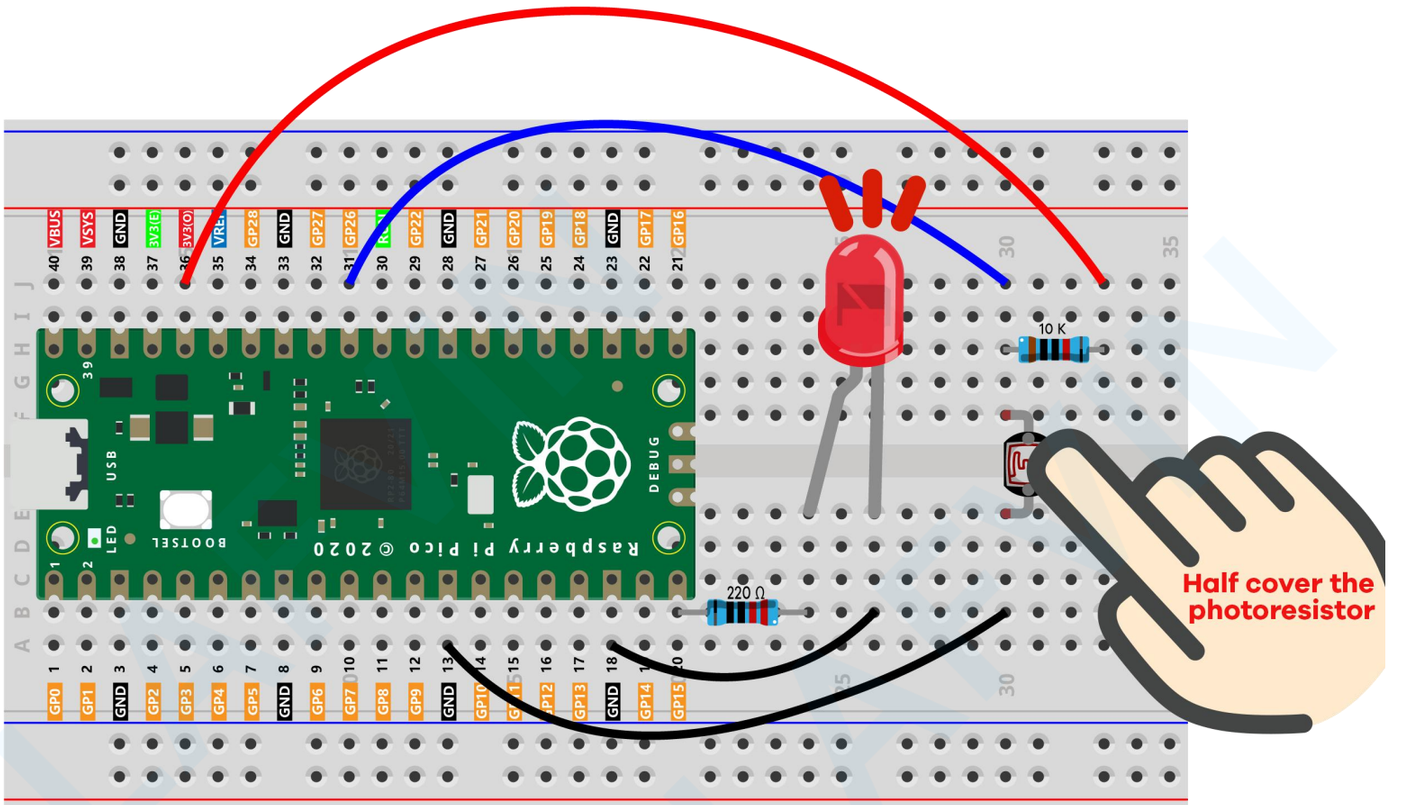
Click the **File>>Open** icon to open **Project_8_Photoresistor_Control_LED.ino** file.

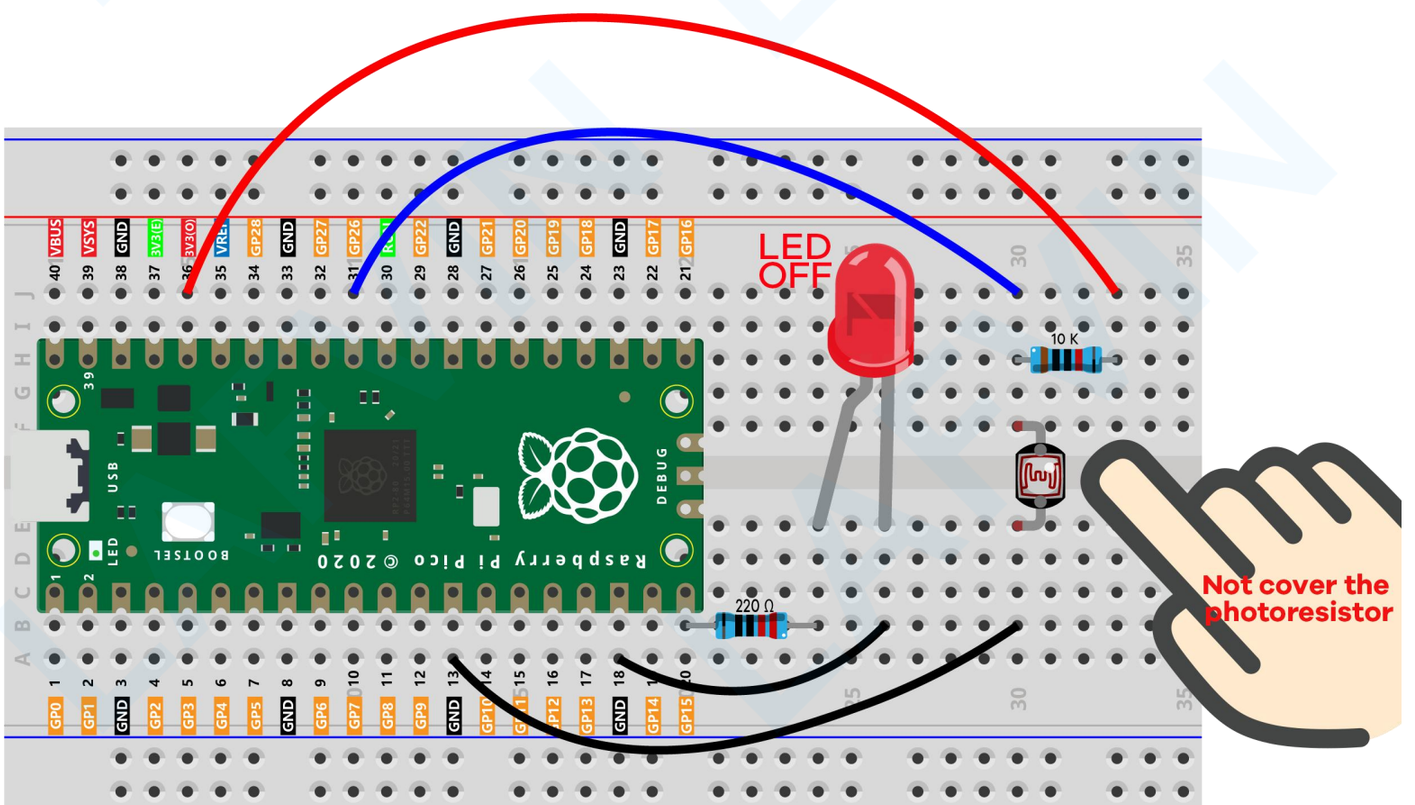
*File path :**Raspberry Pi Pico Starter Kit\C\Codes***



Click  Upload the sketch to Pico. Cover the photoresistor with your hands or illuminate it with lights, the brightness of LEDs will change.Have questions about uploading code?




10 K
220 Ω
Half cover the photoresistor

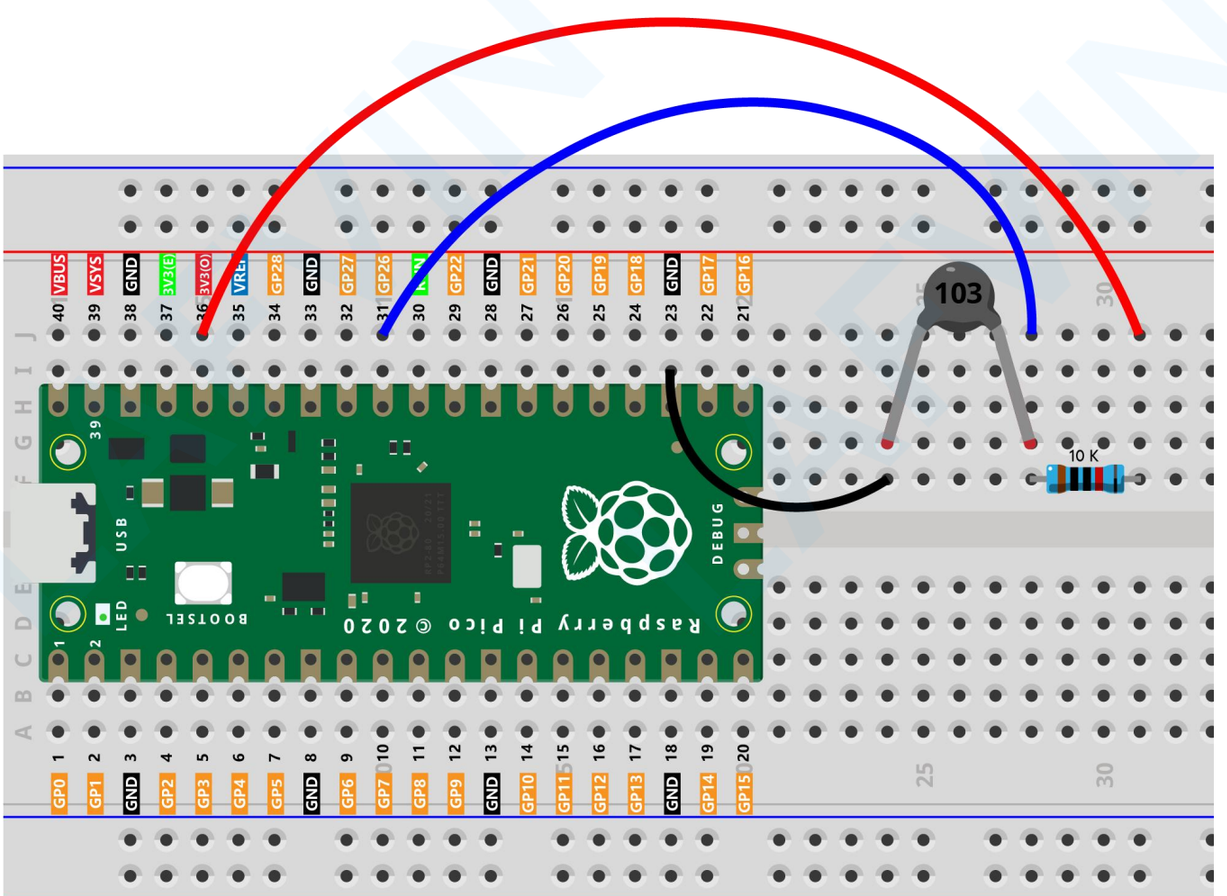

LED OFF
10 K
220 Ω
Not cover the photoresistor

Project 9 Thermeometer

A Thermistor is a type of Resistor whose resistance value is dependent on temperature and changes in temperature. Therefore, we can take advantage of this characteristic to make a Thermometer.

Tip:Learn more about Thermistor

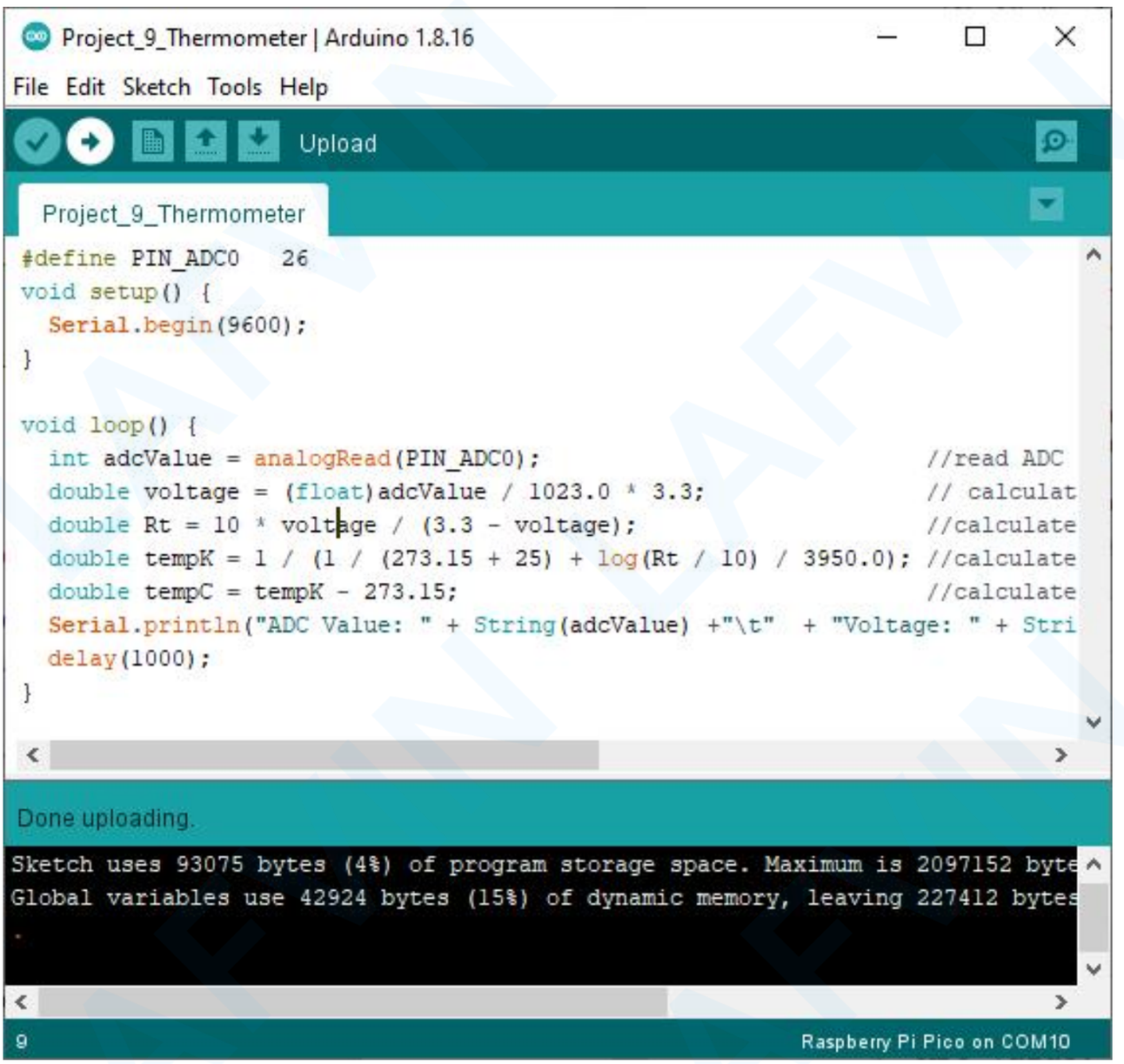
Wiring



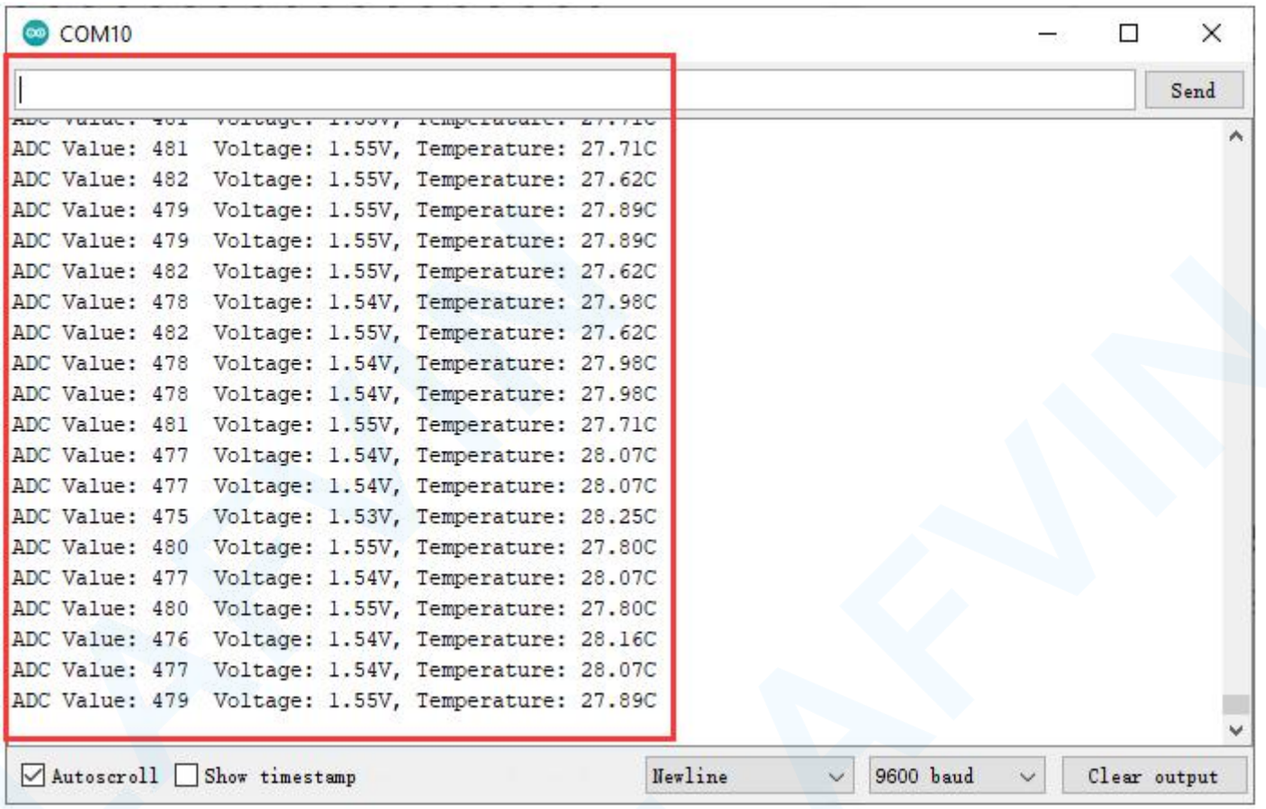
Code

Click the **File>>Open** icon to open **Project_9_Thermometer.ino** file.

*File path :**Raspberry Pi Pico Starter Kit\C\Codes***



Click  Upload the sketch to Pico.The ADC value and Celsius temperature value will be printed to the serial monitor.Have questions about uploading code?



How it works?

Each thermistor has a normal resistance. Here it is 10k ohm, which is measured under 25 degree Celsius. When the temperature gets higher, the resistance of the thermistor decreases. Then the voltage data is converted to digital quantities by the A/D adapter.The temperature in Celsius or Fahrenheit is output via programming.

```
int adcValue = analogRead(PIN_ADC0);
```

analogRead() function is called to read the value of ADC0

```
double voltage = (float)adcValue / 1023.0 * 3.3;
```

Convert the read ADC0 value to get the current Thermistor voltage value.

```
double Rt = 10 * voltage / (3.3 - voltage);                    //calculate resistance value of thermistor
double tempK = 1 / (1 / (273.15 + 25) + log(Rt / 10) / 3950.0); //calculate temperature (Kelvin)
```

According to the formula:

$$T2 = 1/(\frac{1}{T1} + \ln(\frac{Rt}{R})/B)$$

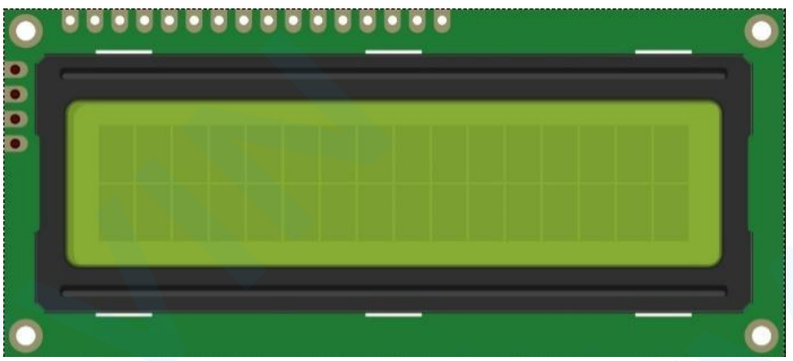
where T1 = 25℃, R = 10KΩ, B = 3950 and the Rt calculated in the previous step, substitute the formula to calculate tempK(T2) . Get the value of the temperature unit K. Learn More

```
tempC = int(tempK - 273.15)
```

Finally, tempK (unit: K) is converted to tempC (unit: °C). You can also convert to Fahrenheit based on your needs.

Project 10 LCD1602 Show Temperature

Using a thermistor and an I2C LCD1602, we can create a room temperature meter.

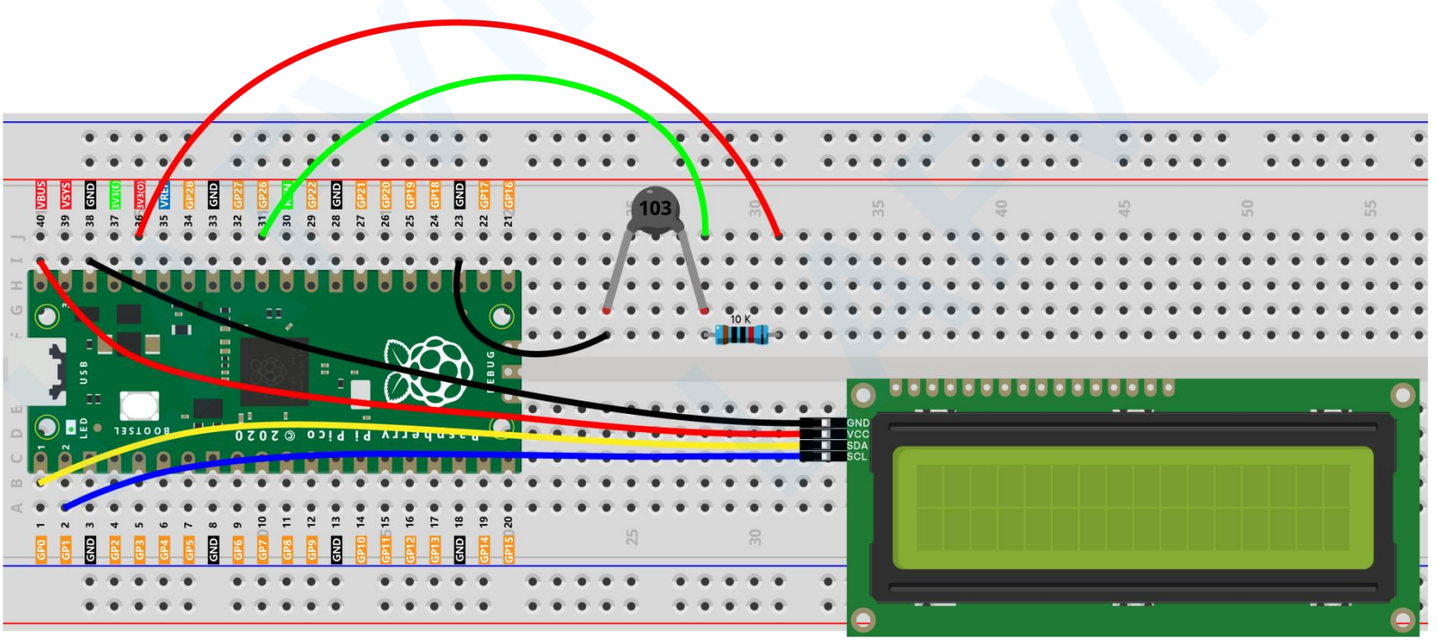


LCD1602 is a character type liquid crystal display, which can display 32 (16*2) characters at the same time.

I2C LCD1602 Display Screen integrates an I2C interface, which connects the serial-input & parallel-output module to the LCD1602 Display Screen. This allows us to use only 4 lines to the operate the LCD1602.

Tip:Learn more about LCD1602

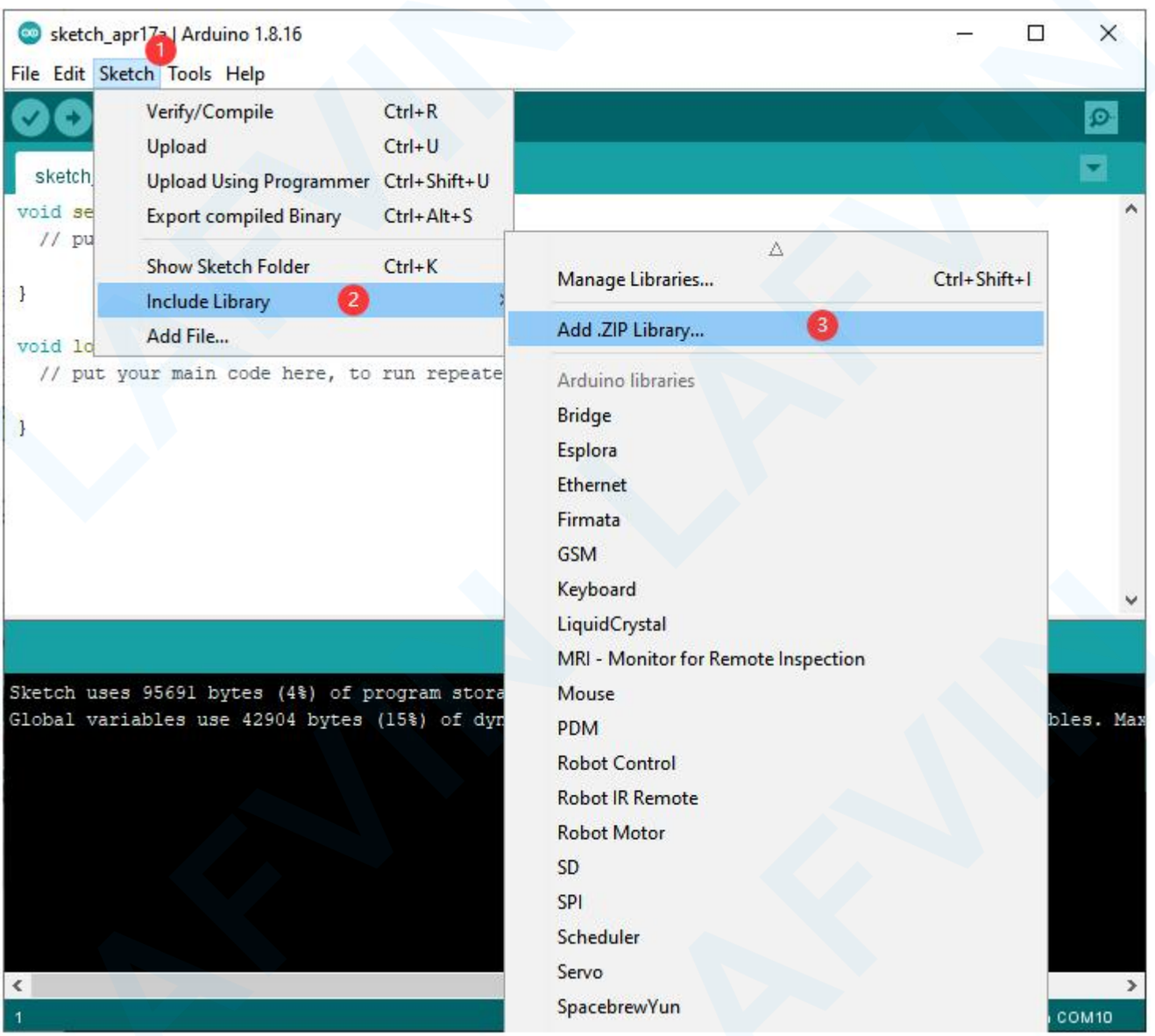
Wiring



Install LiquidCrystal_I2C.zip Library(Important)

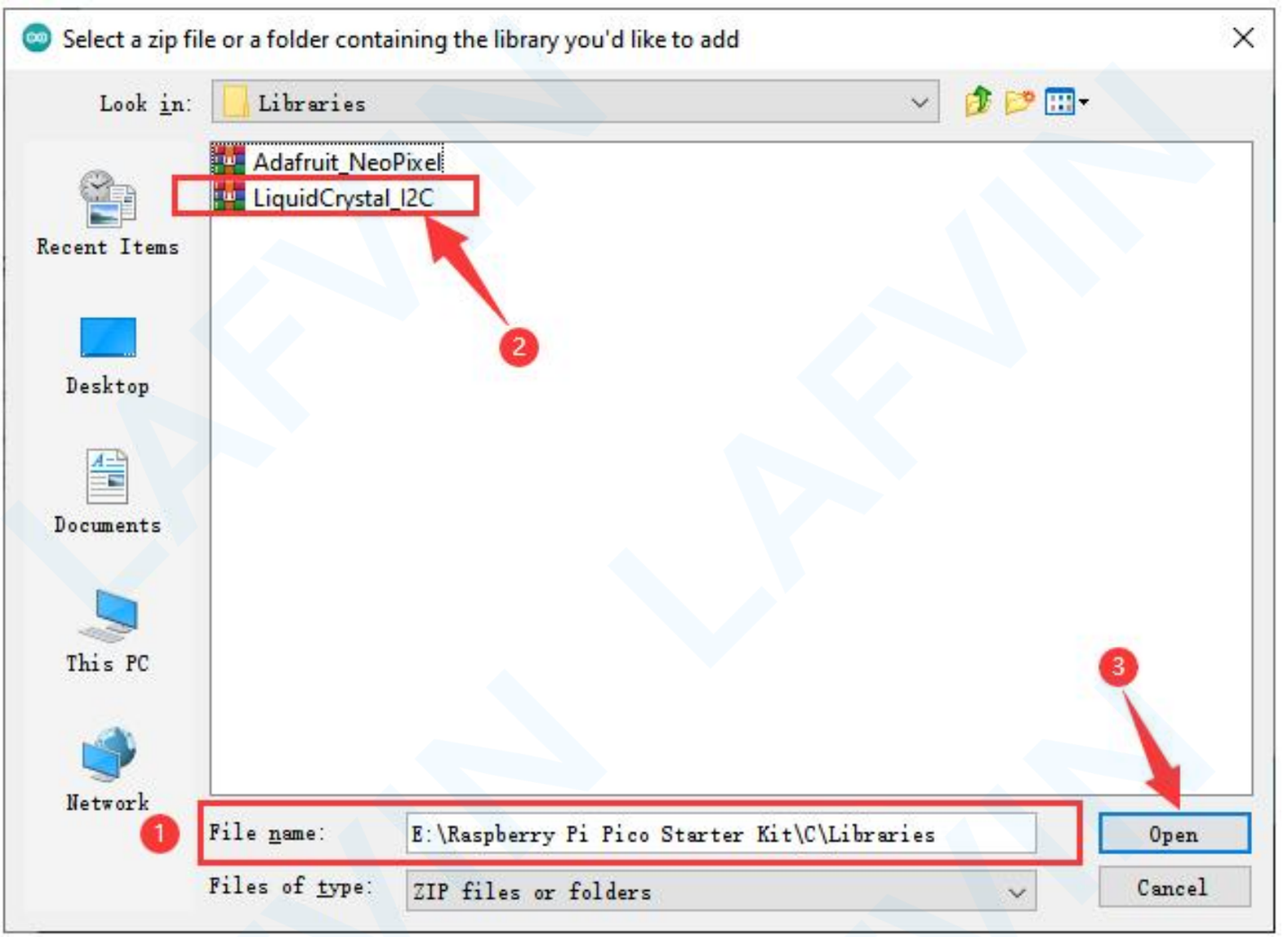
We use the third party library LiquidCrystal I2C. If you haven't installed it yet, please do so before learning.The steps to add third-party Libraries are as follows:

Step①:Open Arduino IDE—>Sketch—>Include Library—>Add .ZIP Library

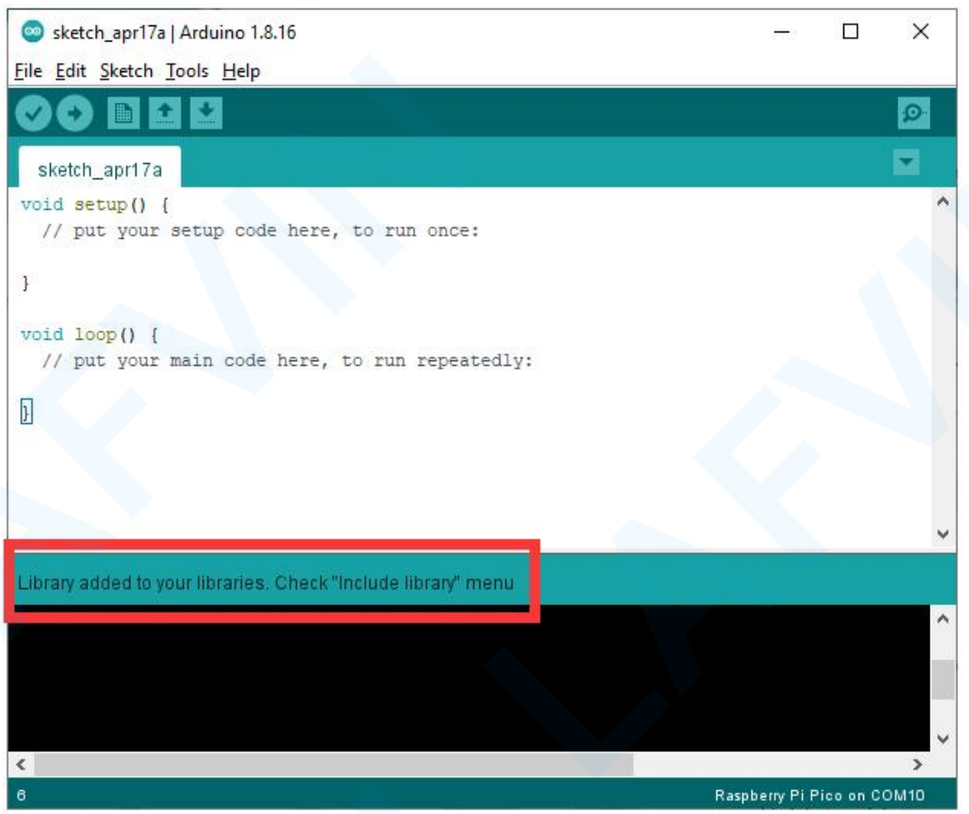


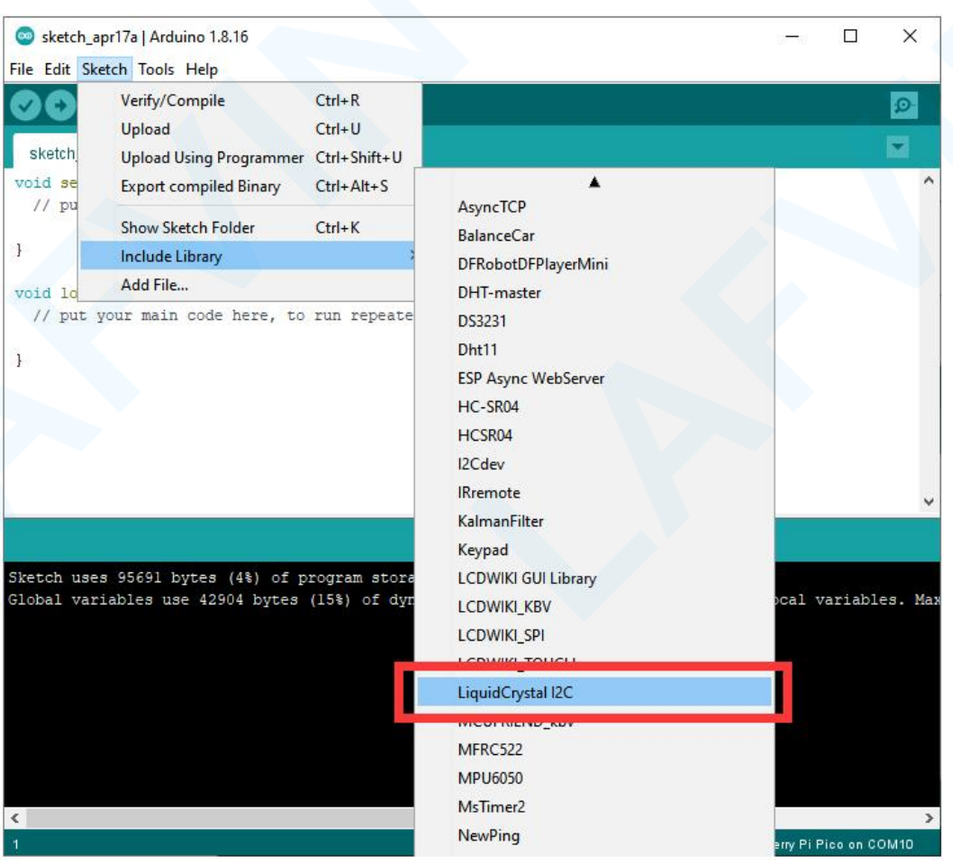
Step②:Select LiquidCrystal_I2C.zip file.

File path : **Raspberry Pi Pico Starter Kit\C\Libraries**

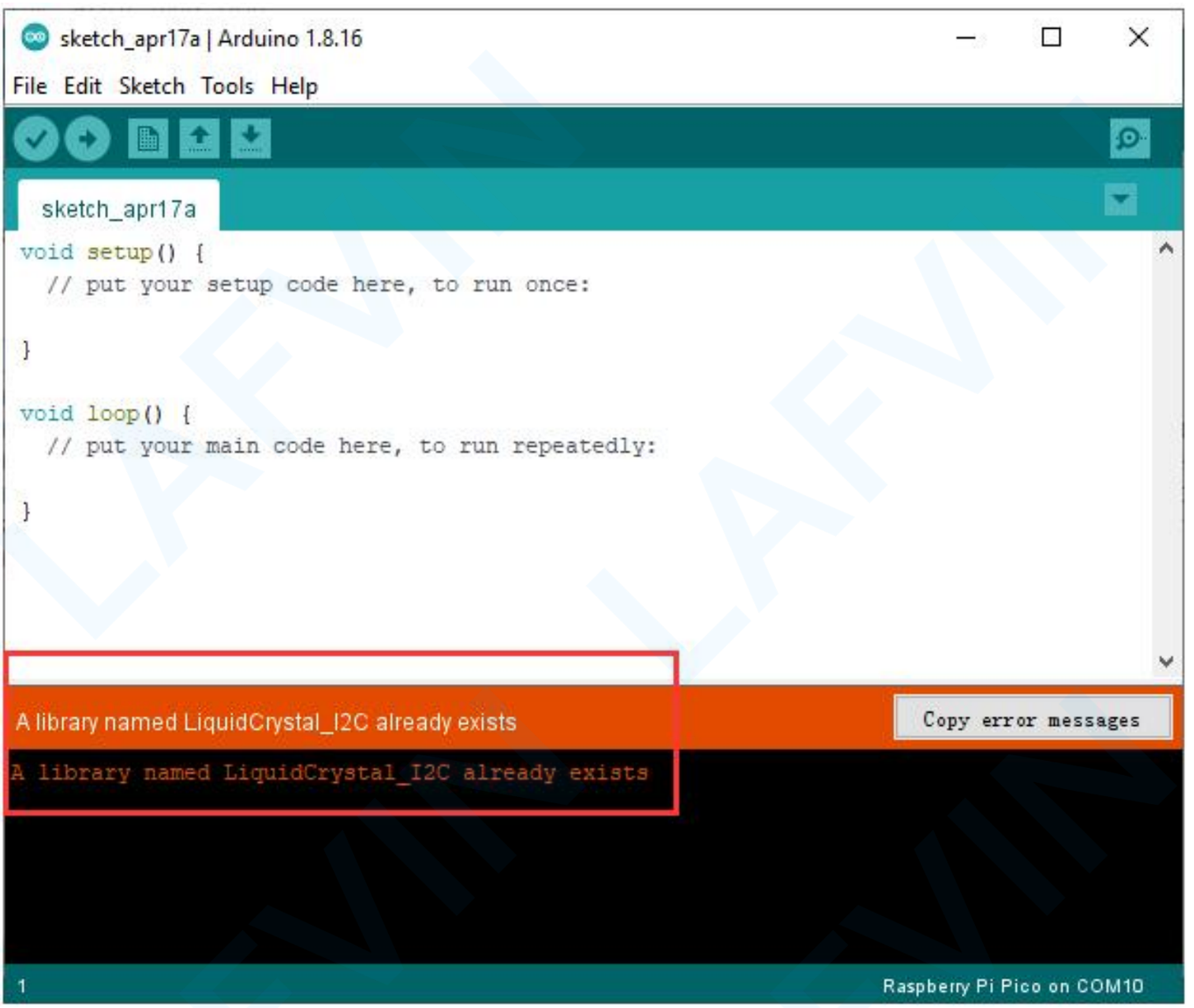


Step③:If the following information is printed, LiquidCrystal_I2C.zip library file is installed successfully.





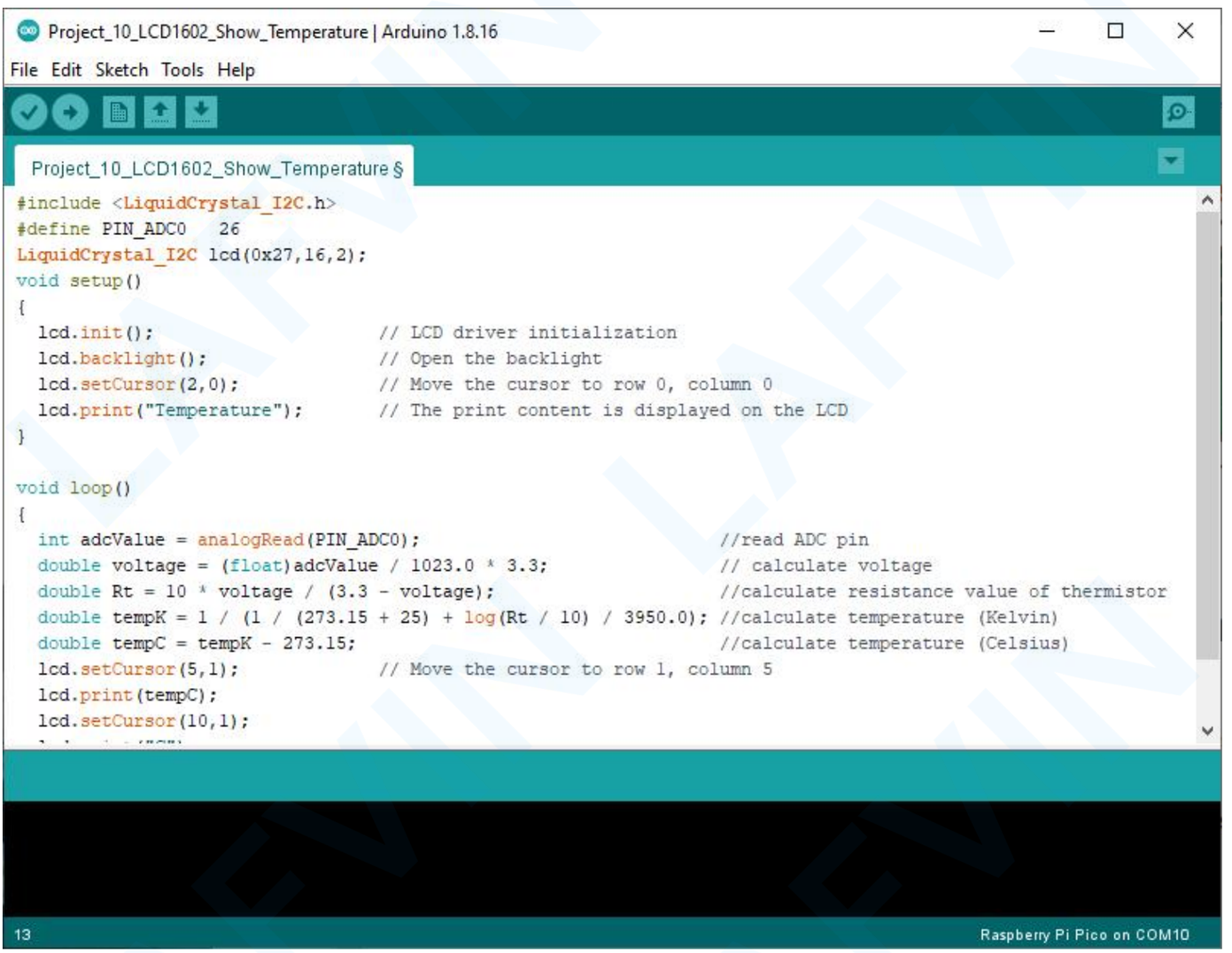
If the following information is printed, you have installed the same library file before, and you need to delete the previous library file before reinstalling.



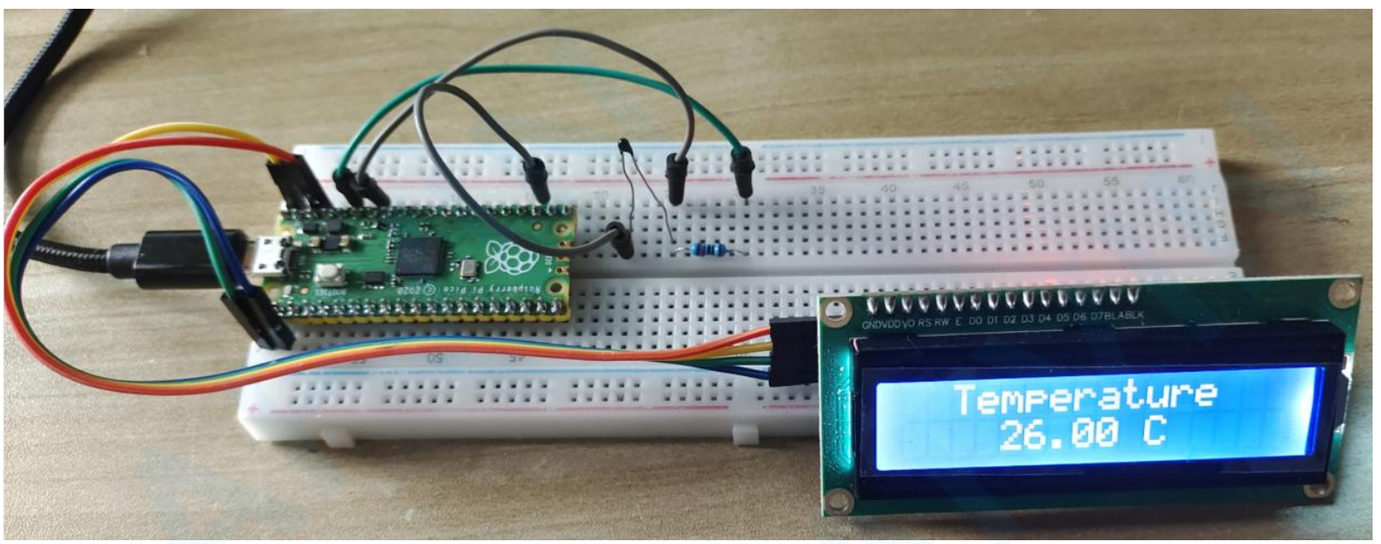
Code

Click the **File>>Open** icon to open **Project_10_LCD1602_Show_Temperature.ino** file.

*File path :**Raspberry Pi Pico Starter Kit\C\Codes***



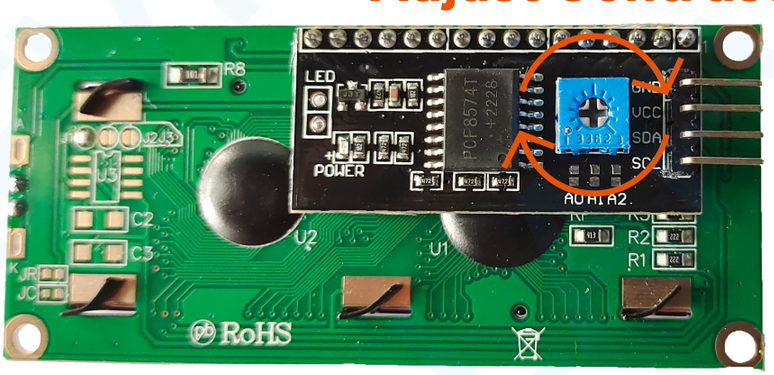
Click  Upload the sketch to Pico.The LCD will display the temperature value in the current environment..Have questions about uploading code?



Note:

If the code and wiring are fine, but the LCD still does not display content or the display is not clear, you can turn the potentiometer on the back to increase the contrast.

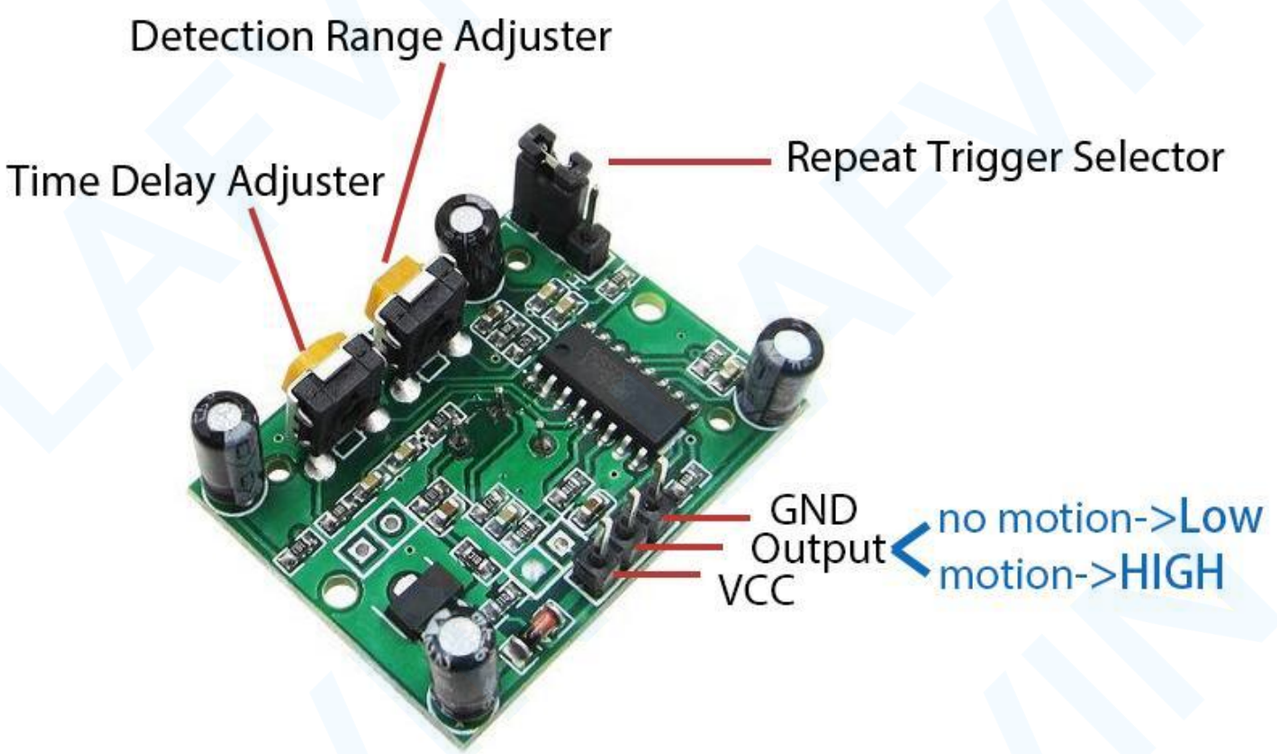
Adjust Contrast



Project 11 Intruder Alarm

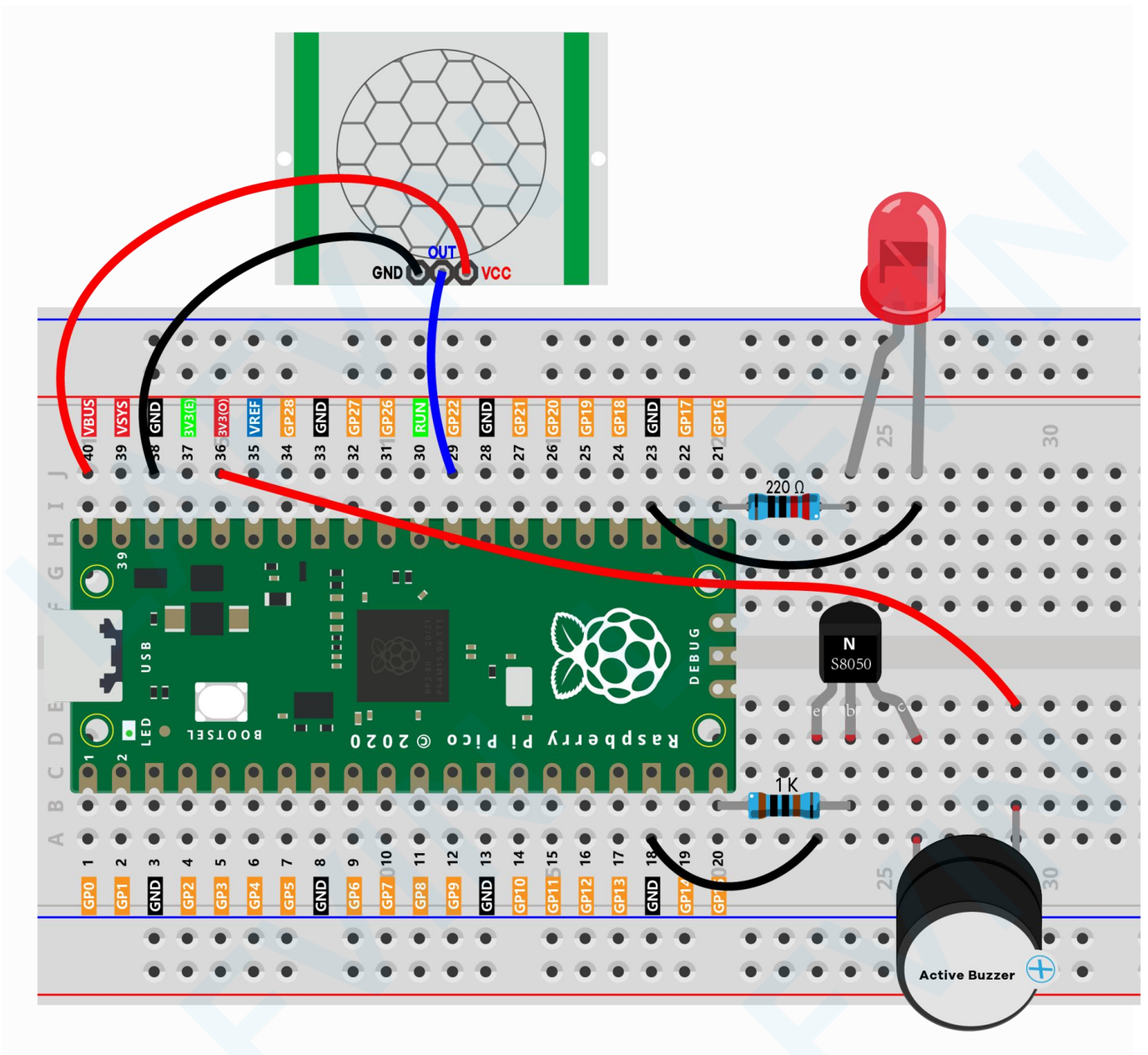
In this project,use PIR,LED and active buzzer to build an Intruder Alarm.

Passive infrared sensor (PIR sensor) is a common sensor that can measure infrared (IR) light emitted by objects in its field of view. Simply put, it will receive infrared radiation emitted from the body, thereby detecting the movement of people and other animals. More specifically, it tells the main control board that someone has entered your room.



Tip:Learn more about PIR Motion Sensor Transistor Buzzer

Wiring



Note:

①The working voltage of PIR senso is 5V. Use the VBUS pin to power it.

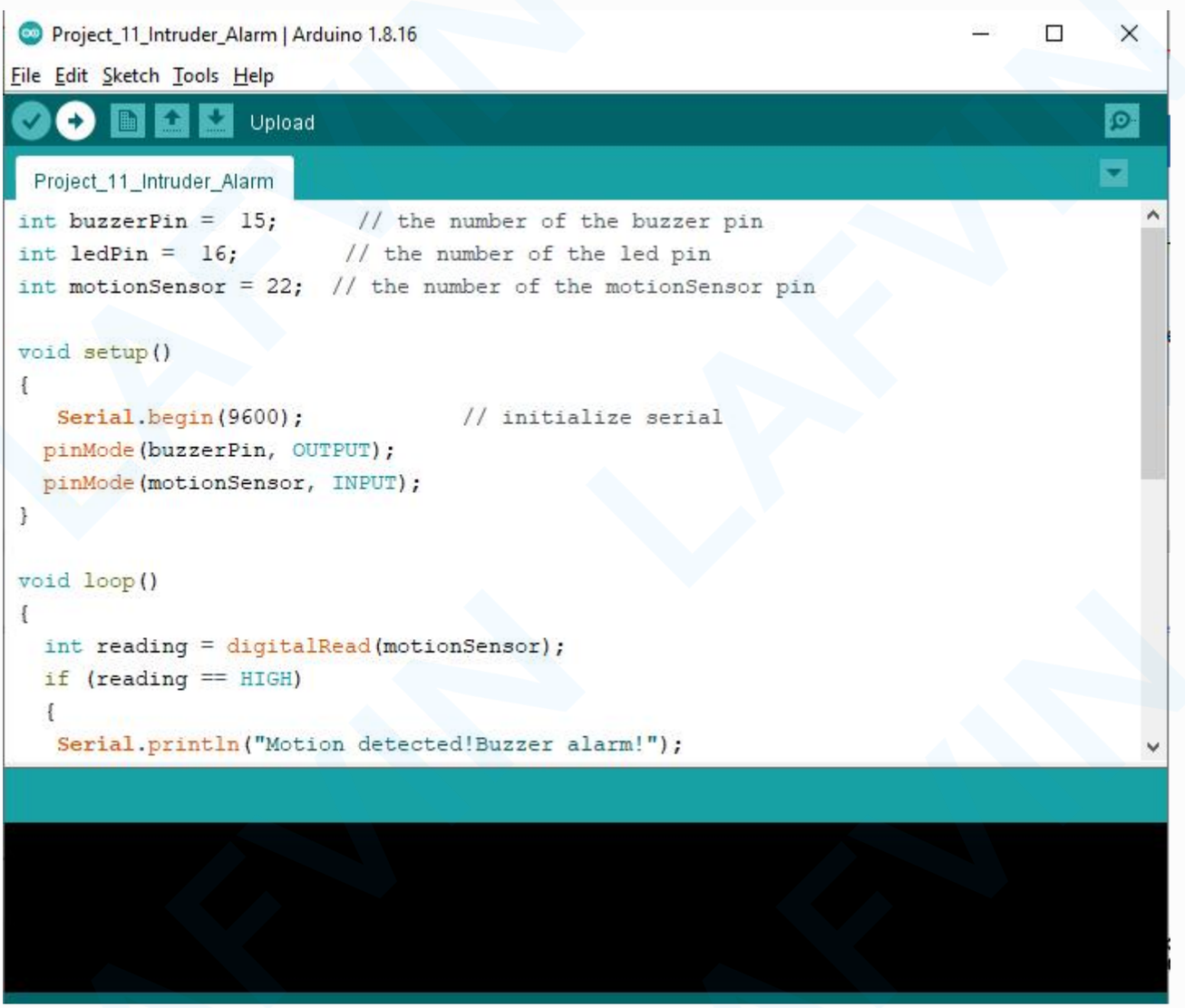
②Two types of buzzers are included in the kit. We need to use active buzzer. Turn them around, the sealed back (not the exposed PCB) is the one we want.

③The 1K resistor between the NPN transistor and GP15 is used for current limiting when the transistor is energized.

Code

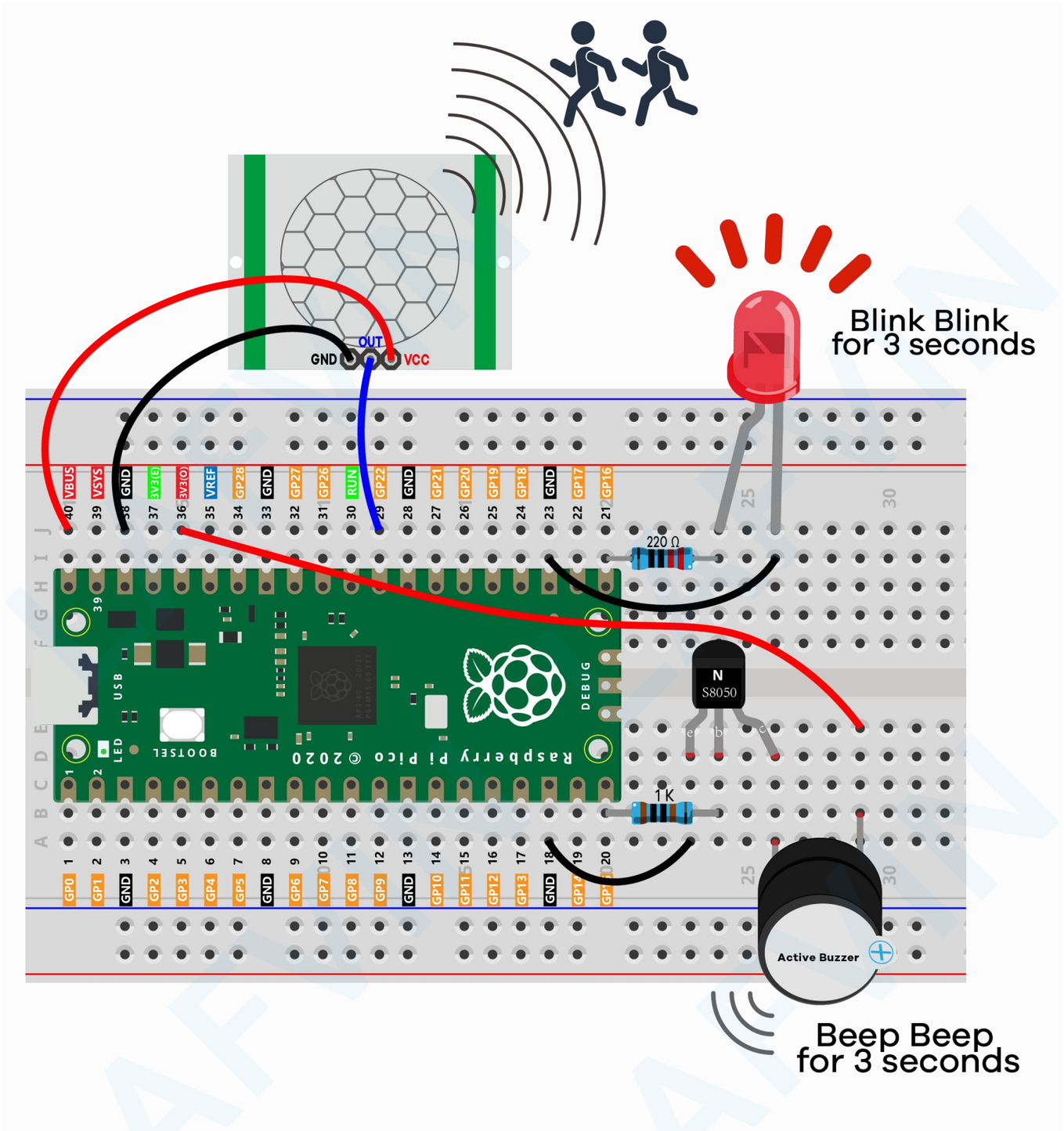
Click the **File>>Open** icon to open **Project_11_Intruder_Alarm.ino** file.

*File path :**Raspberry Pi Pico Starter Kit\C\Codes***



Click  Upload the sketch to Pico.When the program is executed, if someone walks into the PIR detection range, the buzzer will be ’BEEP BEEP’ for 3 seconds.the led will be ’Blink Blink’ for 3 seconds.

Have a question about using PIR Motion Sensor? Have questions about uploading code?


OUT
GND
VCC
Blink Blink
for 3 seconds
220 Ω
N
S8050
1K
Active Buzzer
Beep Beep
for 3 seconds

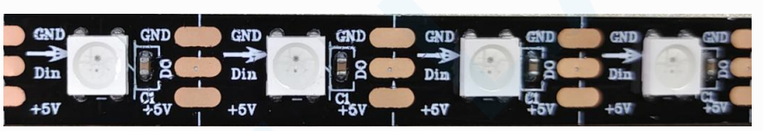
Project 12 RGB LED Strip

In this project, learn to use the infrared sensor to control the display mode of the led light strip.

RGB LED Strip

WS2812 is a intelligent control LED light source that the control circuit and RGB chip are integrated in a package of 5050 components. It internal include intelligent digital port data latch and signal reshaping amplification drive circuit. Also include a precision internal oscillator and a 12V voltage programmable constant current control part, effectively ensuring the pixel point light color height consistent.

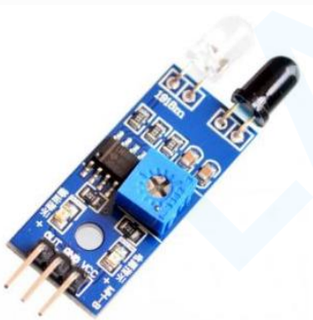
The data transfer protocol use single NZR communication mode. After the pixel power-on reset, the DIN port receive data from controller, the first pixel collect initial 24bit data then sent to the internal data latch, the other data which reshaping by the internal signal reshaping amplification circuit sent to the next cascade pixel through the DO port. After transmission for each pixel，the signal to reduce 24bit. pixel adopt auto reshaping transmit technology, making the pixel cascade number is not limited the signal transmission, only depend on the speed of signal transmission.



Tip:Learn more about RGB LED Strip

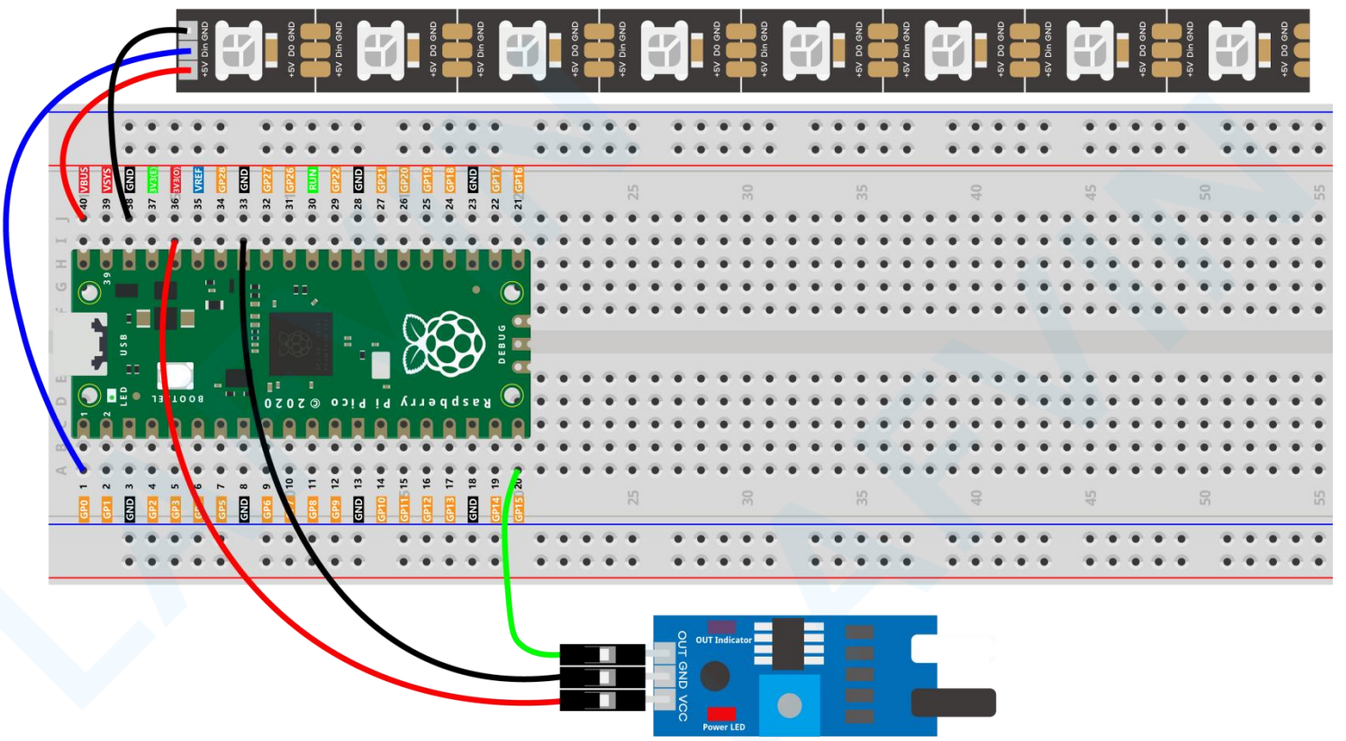
IR Proximity Sensor Module

Proximity Sensor are used to detect objects and obstacles in front of sensor. Sensor keeps transmitting infrared light and when any object comes near, it is detected by the sensor by monitoring the reflected light from the object



Tip:Learn more about IR Proximity Sensor Module

Wiring



Note:

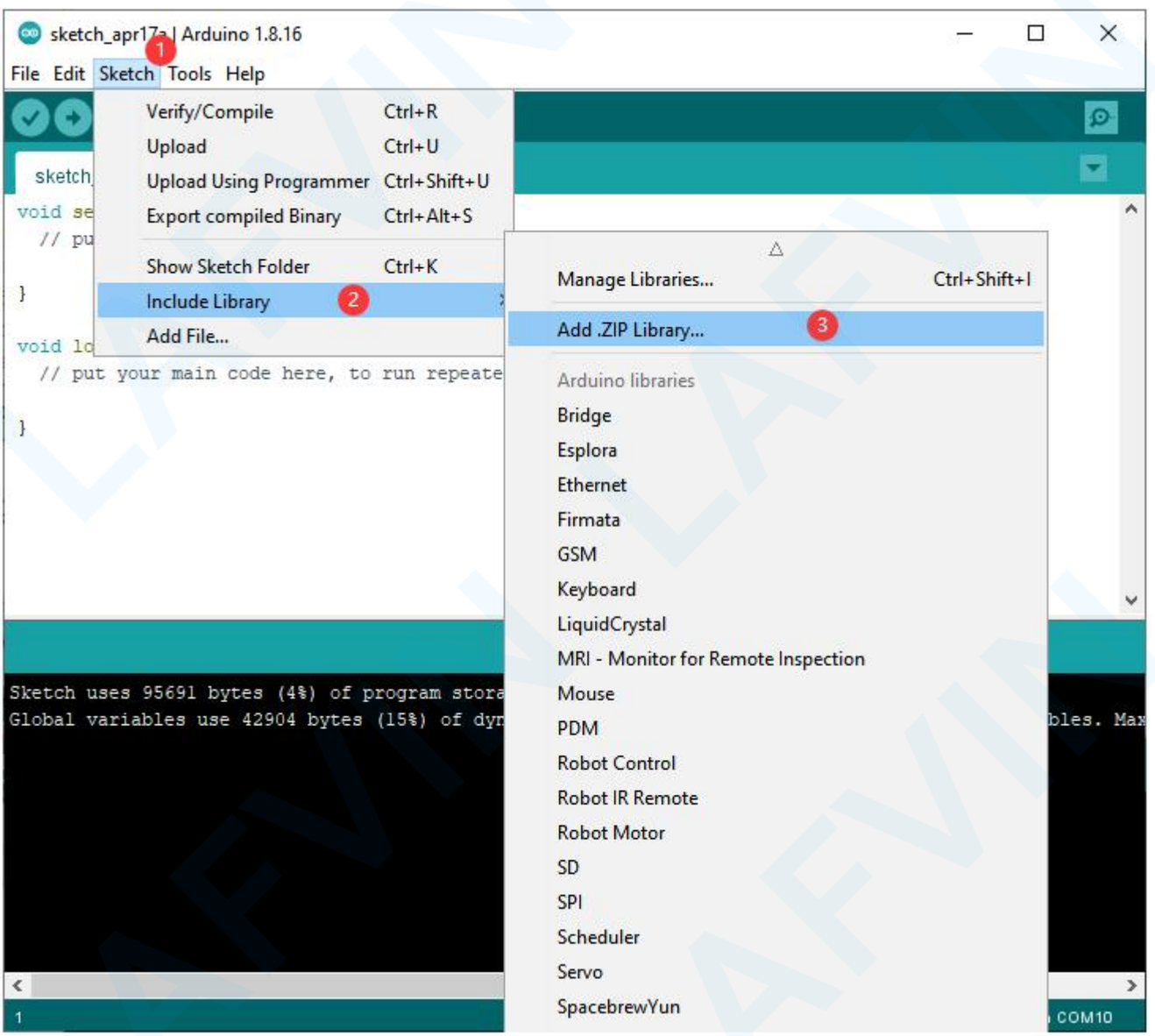
The working voltage of servo is 5V. Use the VBUS pin to power it.

Although the LED Strip with any number of LEDs can be used in Pico, the power of its VBUS pin is limited. Here, we will use eight LEDs, which are safe. But if you want to use more LEDs, you need to add a separate power supply.

Install Adafruit_NeoPixel.zip Library(Important)

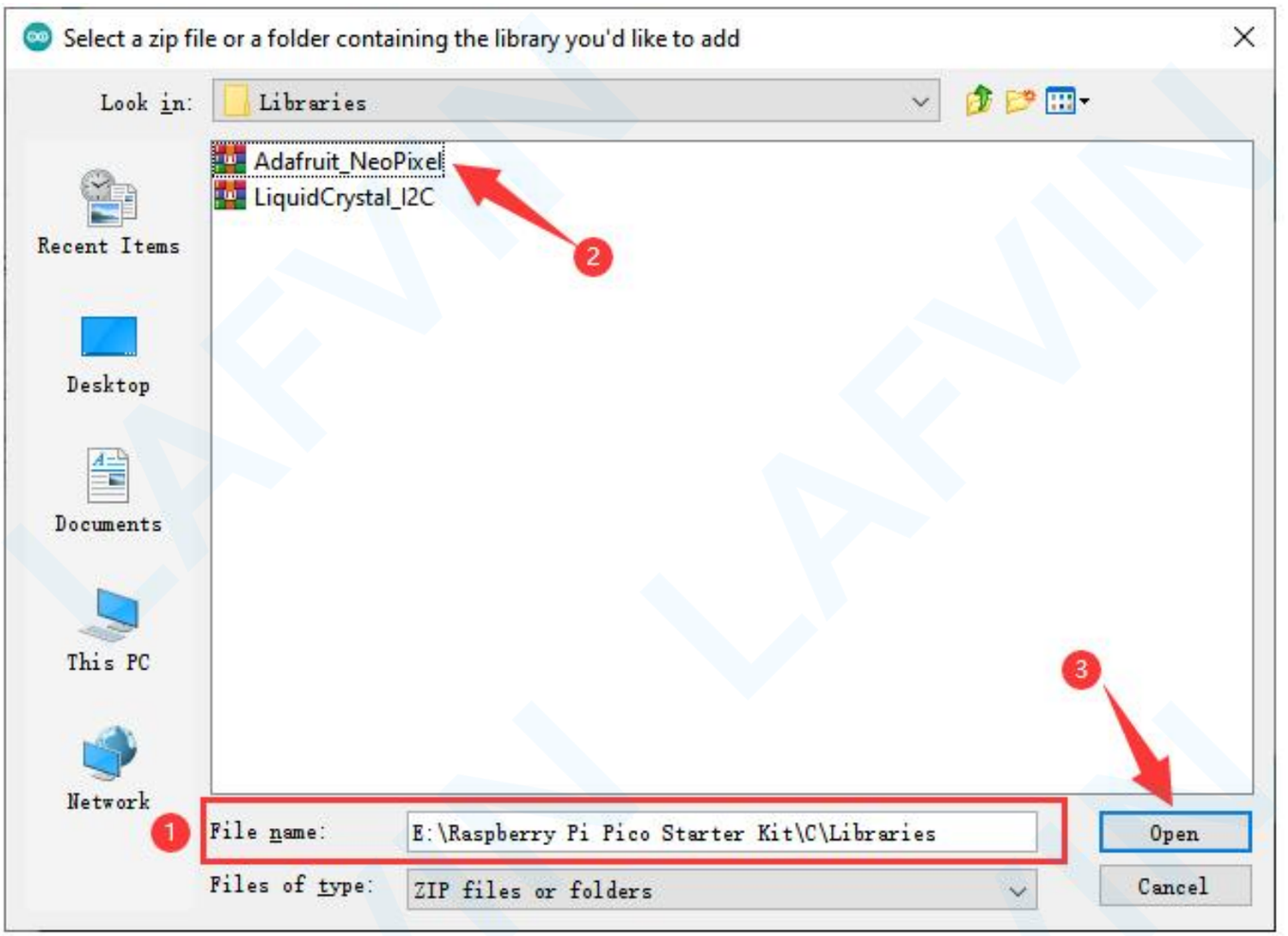
We use the third party library Adafruit_NeoPixel. If you haven't installed it yet, please do so before learning.The steps to add third-party Libraries are as follows:

Step①:Open Arduino IDE—>Sketch—>Include Library—>Add .ZIP Library

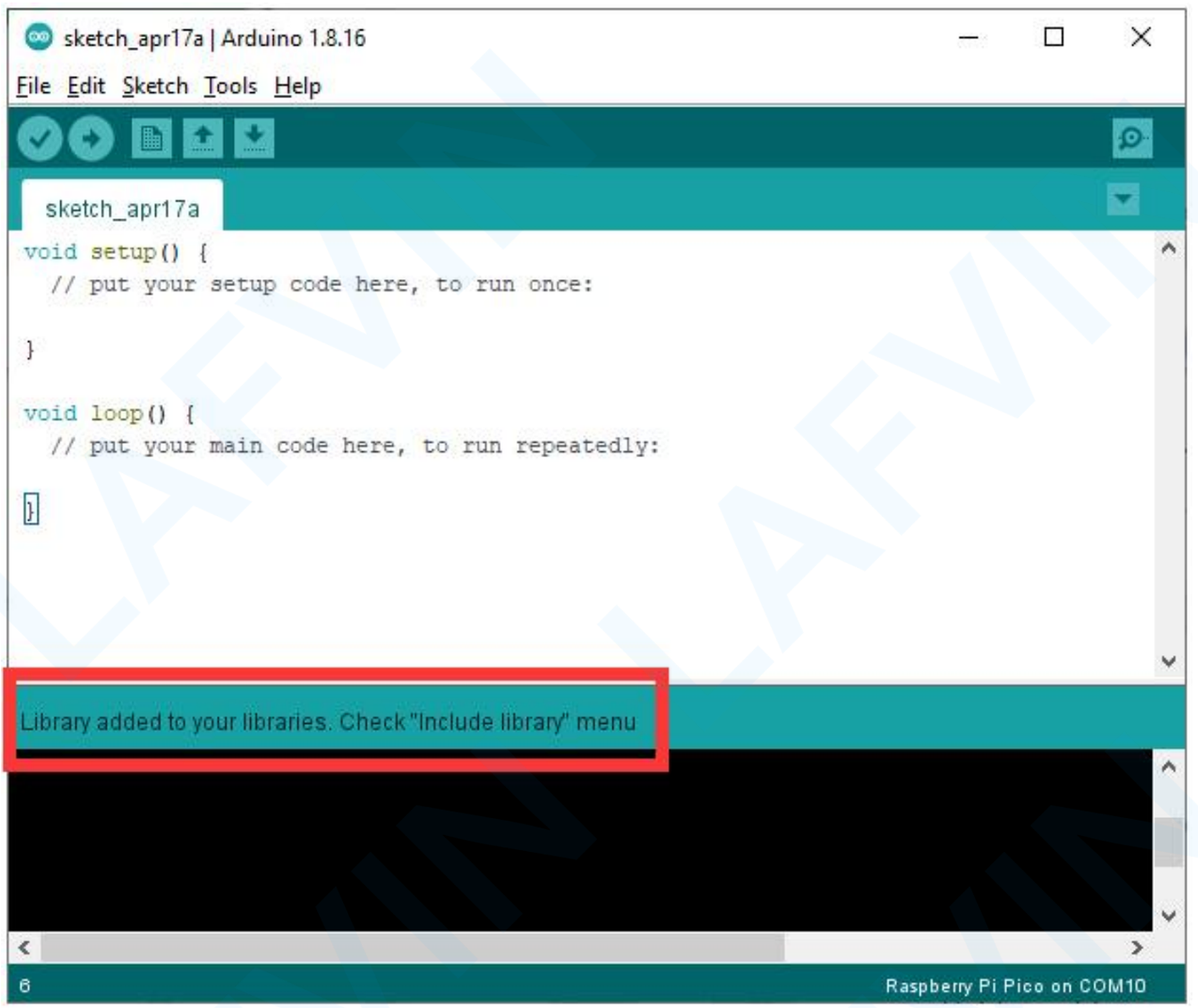


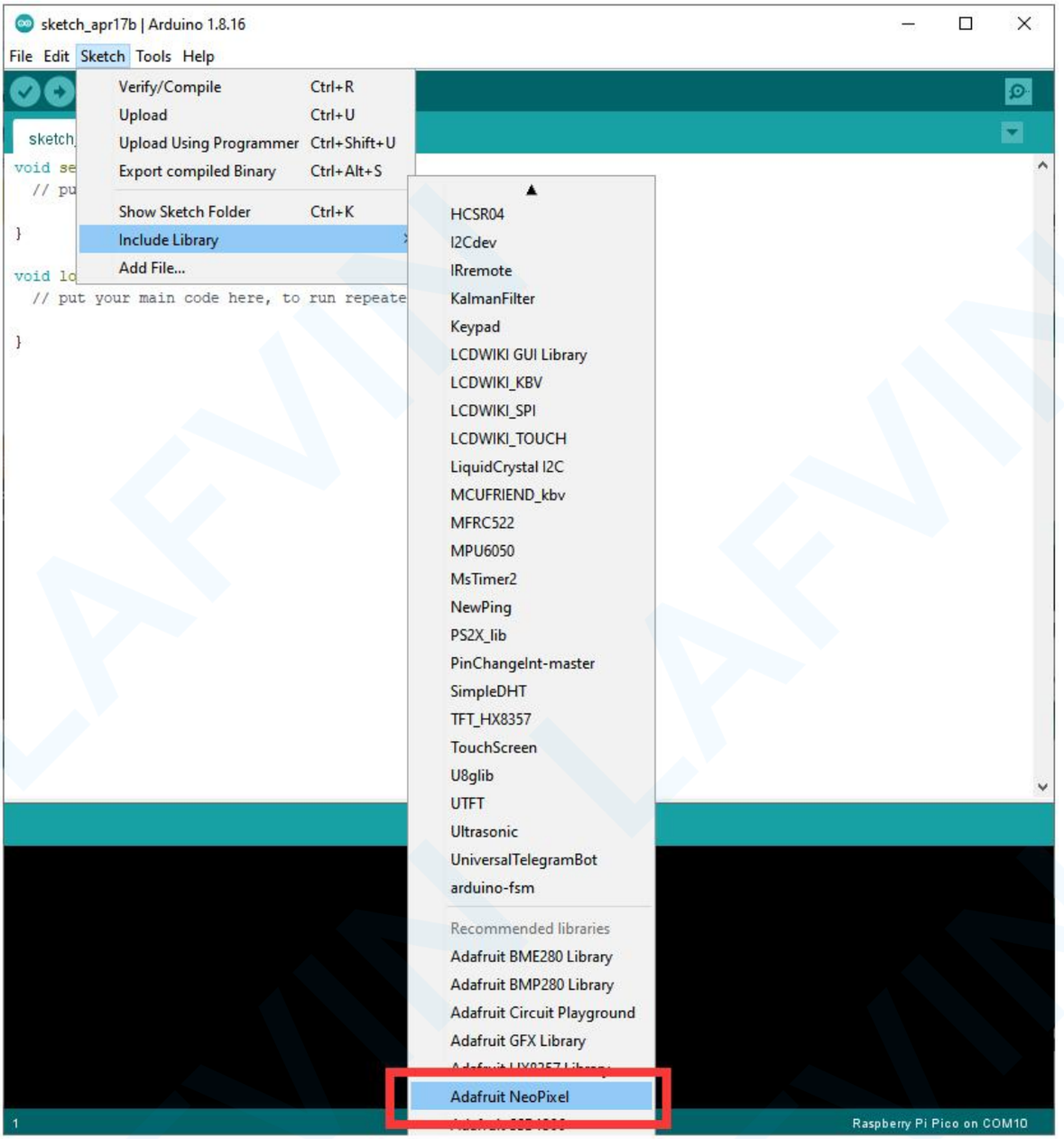
Step②:Select Adafruit_NeoPixel.zip file.

File path : **Raspberry Pi Pico Starter Kit\C\Libraries**

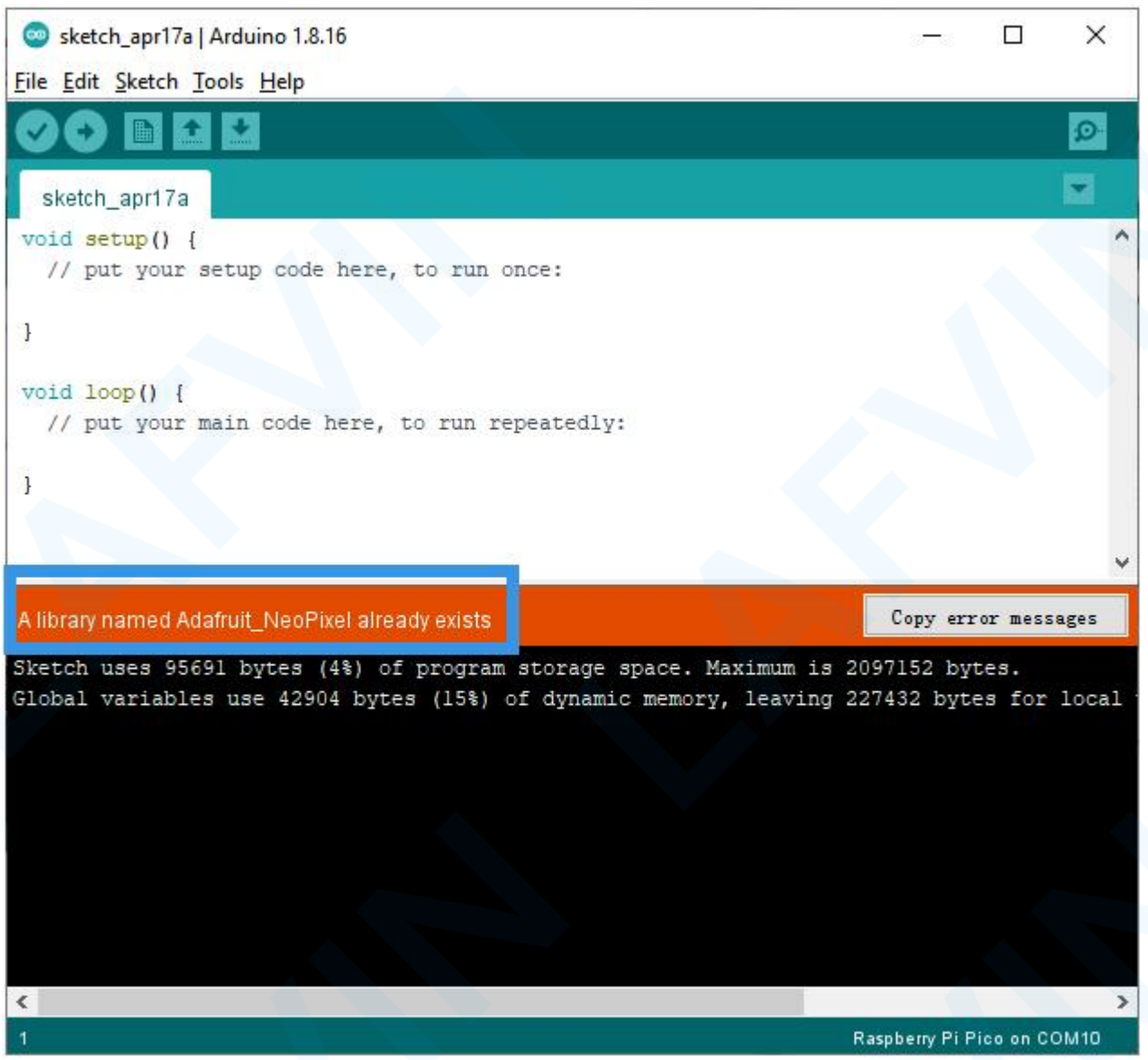


Step③:If the following information is printed, LiquidCrystal_I2C.zip library file is installed successfully.




sketch_apr17b | Arduino 1.8.16
File Edit Sketch Tools Help
Verify/Compile Ctrl+R
Upload Ctrl+U
Upload Using Programmer Ctrl+Shift+U
Export compiled Binary Ctrl+Alt+S
Show Sketch Folder Ctrl+K
Include Library
Add File...
HCSR04
I2Cdev
IRremote
KalmanFilter
Keypad
LCDWIKI GUI Library
LCDWIKI_KBV
LCDWIKI_SPI
LCDWIKI_TOUCH
LiquidCrystal I2C
MCUFRIEND_kbv
MFRC522
MPU6050
MsTimer2
NewPing
PS2X_lib
PinChangeInt-master
SimpleDHT
TFT_HX8357
TouchScreen
U8glib
UTFT
Ultrasonic
UniversalTelegramBot
arduino-fsm
Recommended libraries
Adafruit BME280 Library
Adafruit BMP280 Library
Adafruit Circuit Playground
Adafruit GFX Library
Adafruit NeoPixel
Raspberry Pi Pico on COM10

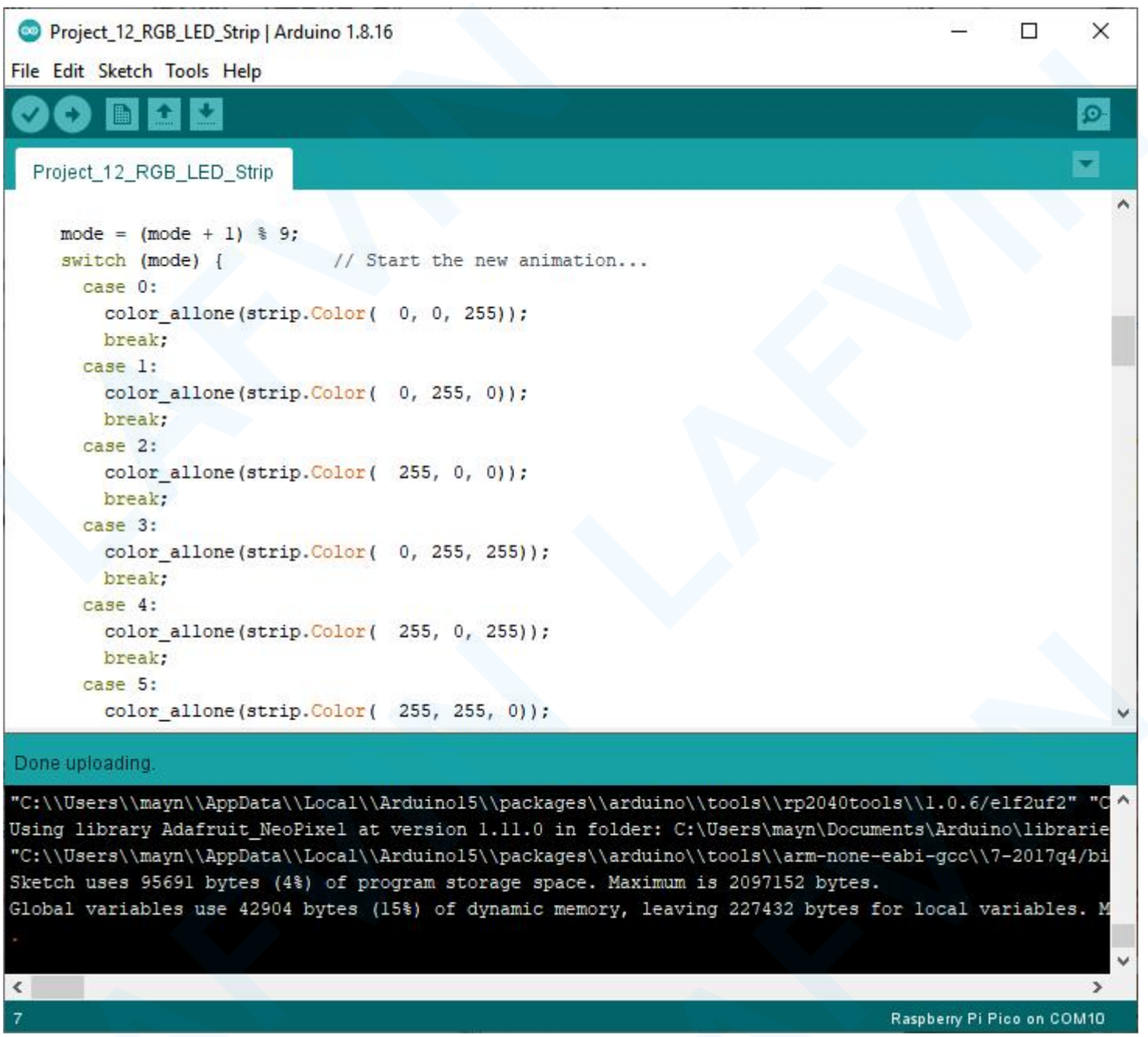
If the following information is printed, you have installed the same library file before, and you need to delete the previous library file before reinstalling.



Code

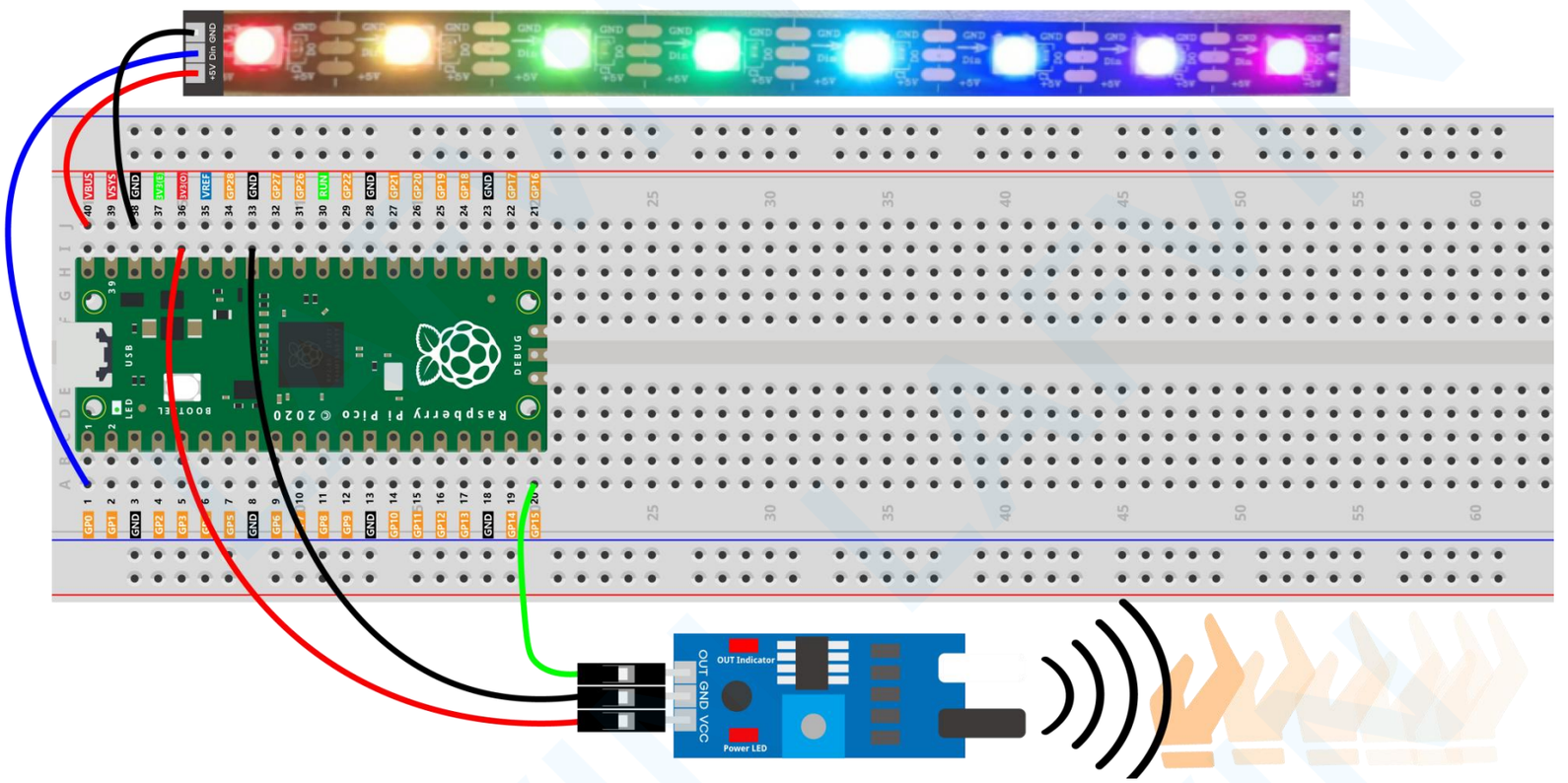
Click the **File>>Open** icon to open **Project_12_RGB_LED_Strip.ino** file.

*File path :**Raspberry Pi Pico Starter Kit\C\Codes***



Click  Upload the sketch to Pico.Every time the hand moves to the detection range of the IR sensor, the LED light strip will change to the next display mode.

Have questions about uploading code?



How it works?

```
Adafruit_NeoPixel strip(PIXEL_COUNT, PIXEL_PIN, NEO_GRB + NEO_KHZ800);
```

Create object strip to control ws2812 and declare the number of LED pixels, pin number and Pixel type flags.

```
strip.setPixelColor(i, color);
```

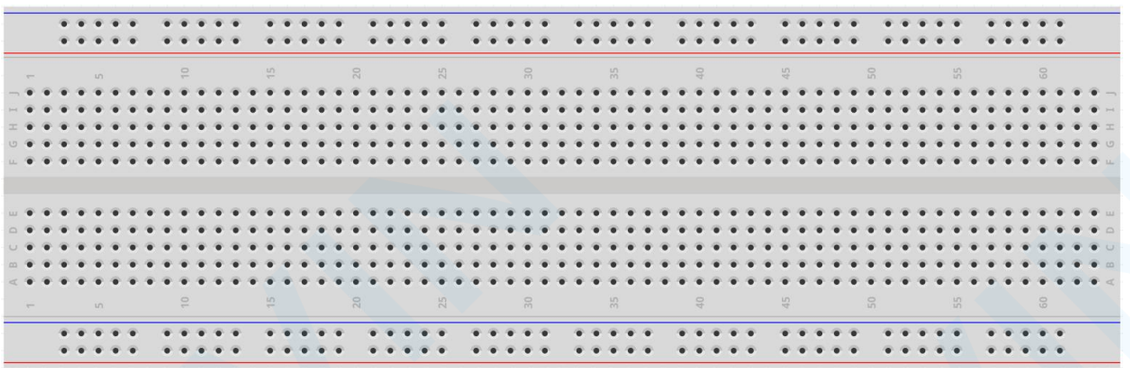
Set the color of a certain RGB pixel, "i" indicates the serial number of the light group, and "color" indicates the value of the color.

```
colorWipe(strip.Color(  0,   255,   255), 50);

theaterChase(strip.Color(  0,   255, 255), 100);

rainbow(10);

theaterChaseRainbow(100);
```

- colorWipe(): From the left (wired side) to the right, light up one led every 50ms, after all light up, wait 500ms, then from left to right, turn off one led every 50ms(variable wait), until all off.
- theaterChase(): First light up the WS2812 Strip on the 1st (wired side), 4, 7 position of the LED, wait 50ms after extinguishing; then light up 2, 5, 8 of the led, wait 50ms after extinguishing; finally light up 3, 6 of the led, wait 50ms after extinguishing. The whole process is cycled 10 times, due to the short interval thus achieving the effect of flow.
- rainbow(): Different colors are displayed on 8 LEDs, and then these 8 colors flow from right to left (wired side), flowing 3 times and then stopping.
- theaterChaseRainbow(): Add flow effect to rainbow().

Components Introduction

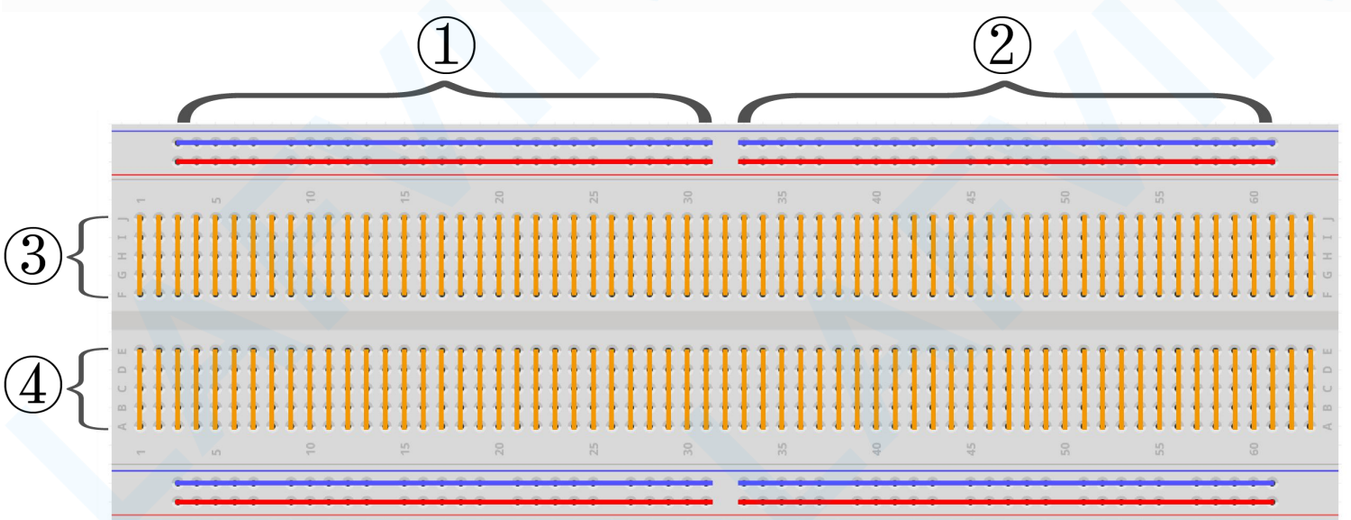
➢ Breadboard



A breadboard is a construction base for prototyping of electronics. Originally the word referred to a literal bread board, a polished piece of wood used for slicing bread.[1] In the 1970s the solderless breadboard (a.k.a. plugboard, a terminal array board) became available and nowadays the term “breadboard” is commonly used to refer to these.

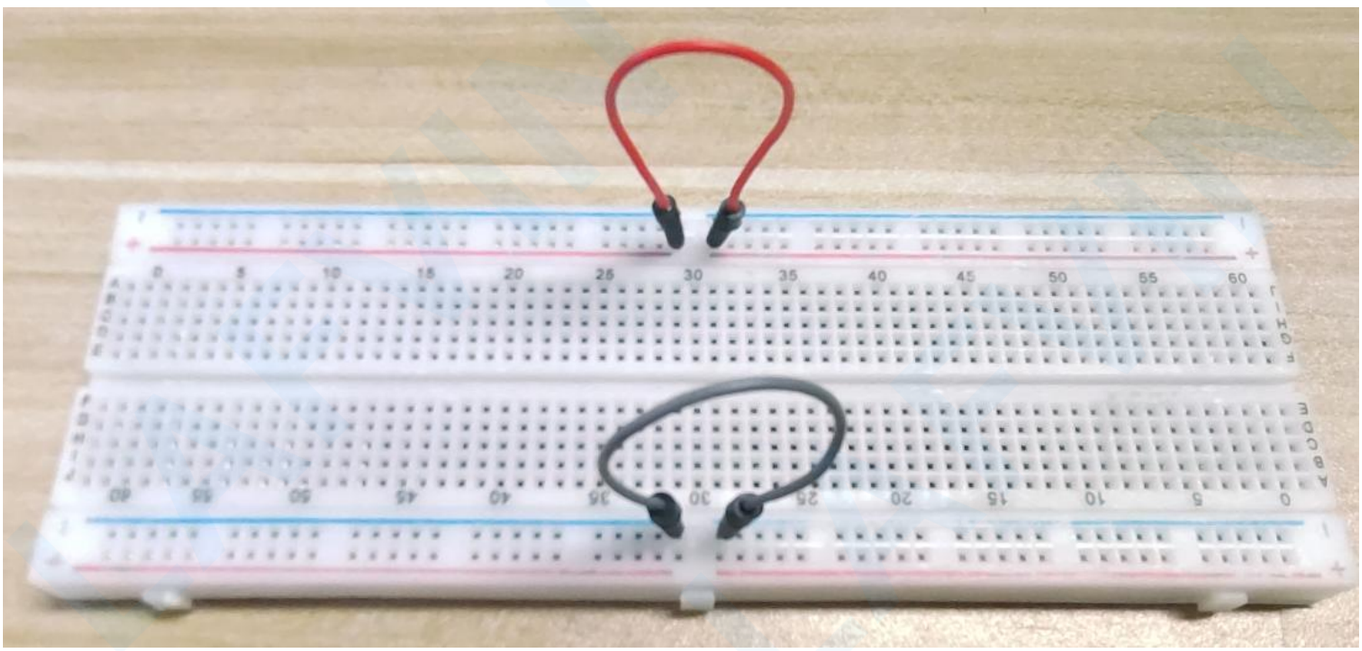
It is used to build and test circuits quickly before finishing any circuit design. And it has many holes into which components mentioned above can be inserted like ICs and resistors as well as jumper wires. The breadboard allows you to plug in and remove components easily.

The picture shows the internal structure of a breadboard. Although these holes on the breadboard appear to be independent of each other, they are actually connected to each other through metal strips internally.



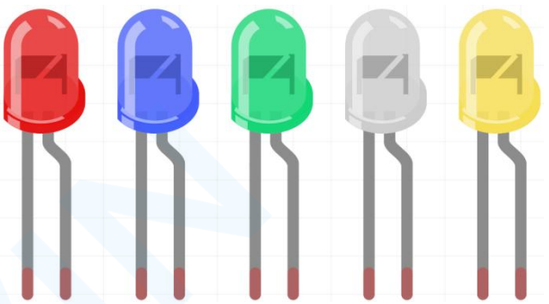
Note:

Area ① and area ② are not connected, and area ① and area ② can be connected through a jumper.

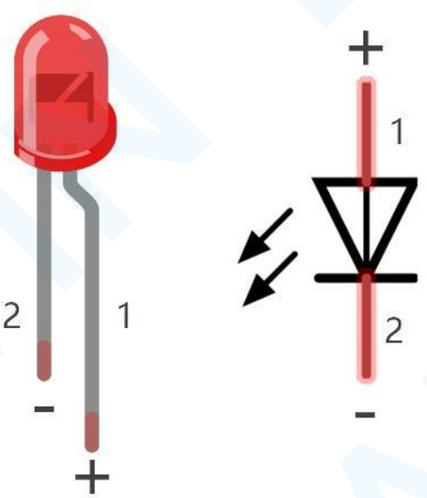


If you want to know more about breadboard, refer to: How to Use a Breadboard - Science Buddies

➢ LED



An LED is a type of diode. All diodes only work if current is flowing in the correct direction and have two Poles.An LED will only work (light up) if the longer pin (+) of LED is connected to the positive output from a power source and the shorter pin is connected to the negative (-). Negative output is also referred to as Ground (GND). This type of component is known as “Polar” (think One-Way Street).All common 2 lead diodes are the same in this respect. Diodes work only if the voltage of its positive electrode is higher than its negative electrode and there is a narrow range of operating voltage for most all common diodes of 1.9 and 3.4V. If you use much more than 3.3V the LED will be damaged and burn out.



Note:

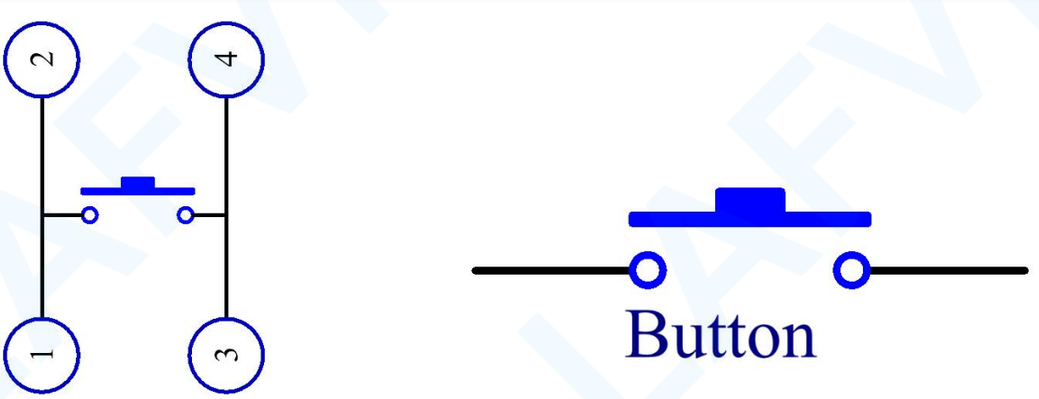
LEDs cannot be directly connected to a power supply, which usually ends in a damaged component. A

resistor with a specified resistance value must be connected in series to the LED you plan to use.

➢ Button

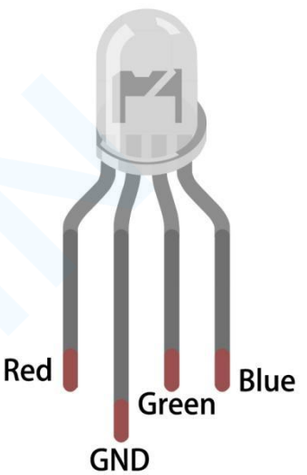
Buttons are a common component used to control electronic devices. They are usually used as switches to connect or break circuits. Although buttons come in a variety of sizes and shapes, the one used here is a 6mm mini-button as shown in the following pictures. Pin 1 is connected to pin 2 and pin 3 to pin 4. So you just need to connect either of pin 1 and pin 2 to pin 3 or pin 4.

The following is the internal structure of a button. The symbol on the right below is usually used to represent a button in circuits.

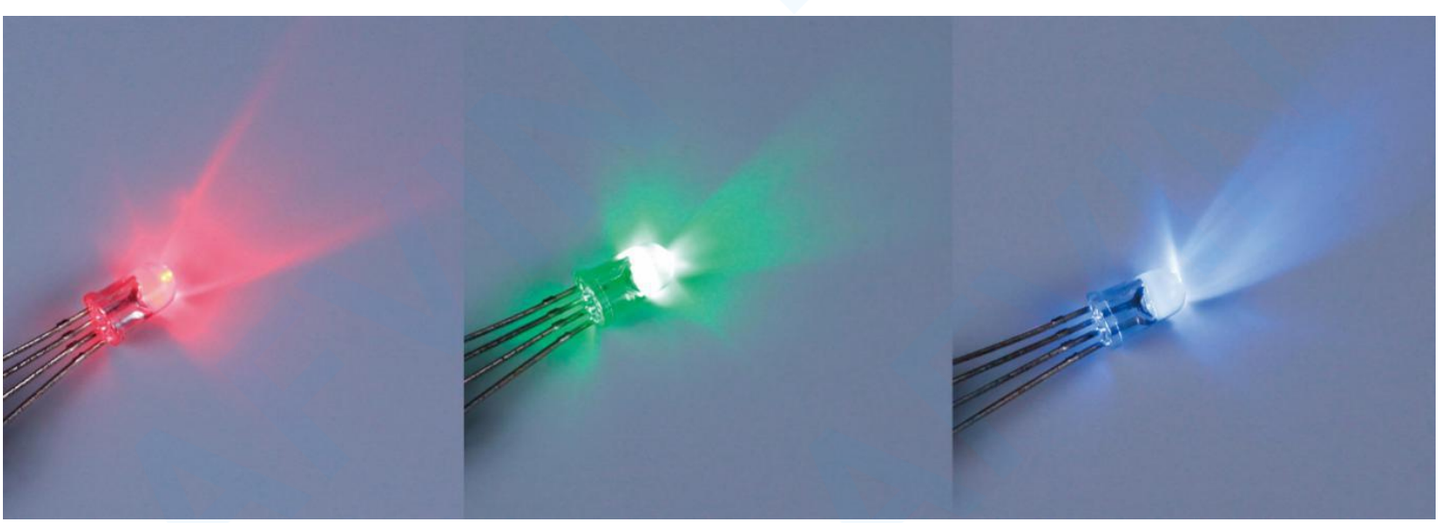


Since the pin 1 is connected to pin 2, and pin 3 to pin 4, when the button is pressed, the 4 pins are connected, thus closing the circuit.

➢ RGB LED

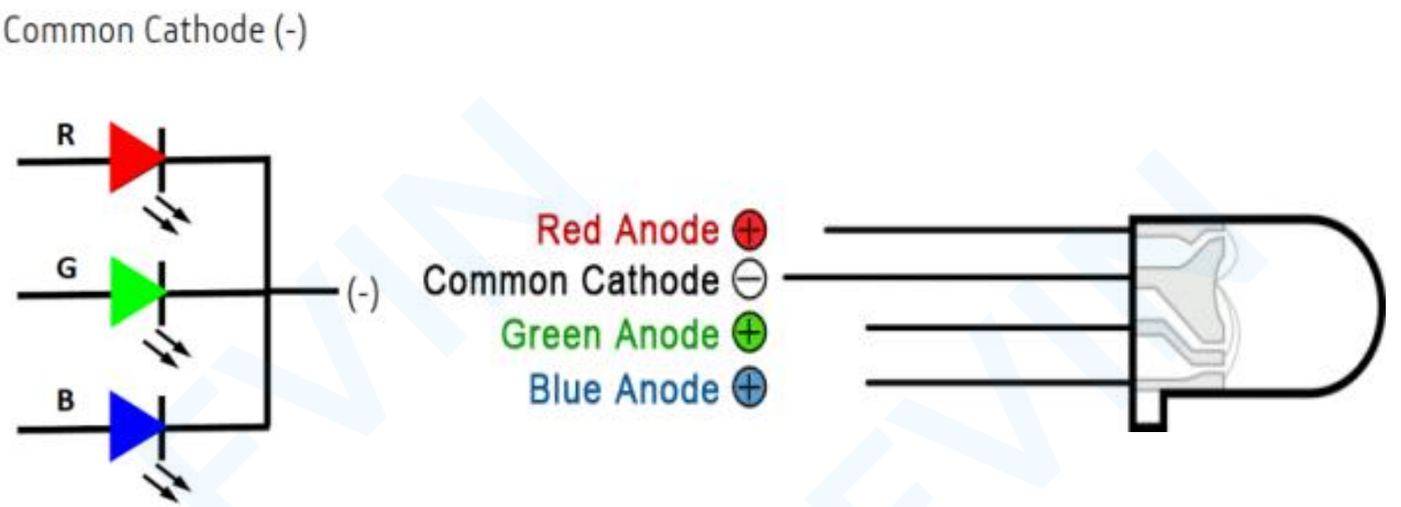


RGB LEDs emit light in various colors. An RGB LED packages three LEDs of red, green, and blue into a transparent or semitransparent plastic shell. It can display various colors by changing the input voltage of the three pins and superimpose them, which, according to statistics, can create 16,777,216 different colors.



RGB LEDs can be categorized into common anode and common cathode ones. In this kit, the latter is used. The common cathode, or CC, means to connect the cathodes of the three LEDs. After you connect it with GND and plug in the three pins, the LED will flash the corresponding color.

Its circuit symbol is shown as figure.

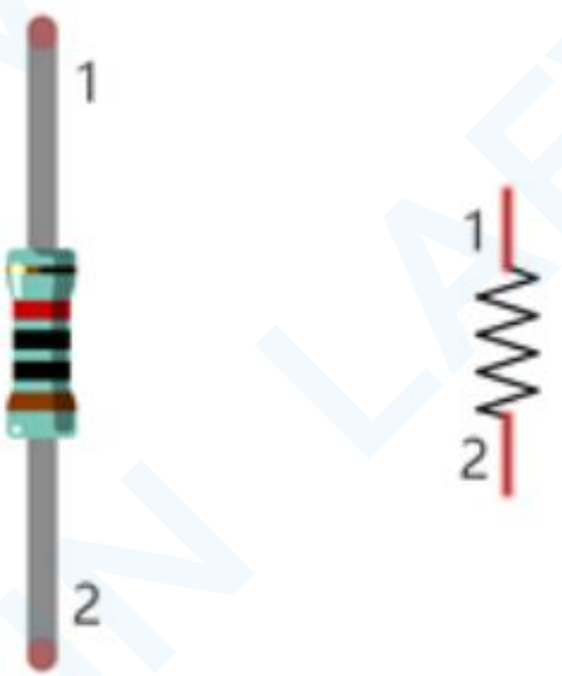


An RGB LED has 4 pins: the longest one is GND; the others are Red, Green and Blue. Touch its plastic shell and you will find a cut. The pin closest to the cut is the first pin, marked as Red, then GND, Green and Blue in turn.

➢ Resistor

Resistors use Ohms (Ω) as the unit of measurement of their resistance (R). 1MΩ=1000kΩ, 1kΩ=1000Ω.

A resistor is a passive electrical component that limits or regulates the flow of current in an electronic circuit.On the left, we see a physical representation of a resistor, and the right is the symbol used to represent the presence of a resistor in a circuit diagram or schematic.



The bands of color on a resistor is a shorthand code used to identify its resistance value. For more details of resistor color codes, please refer to the appendix of this tutorial.

With a fixed voltage, there will be less current output with greater resistance added to the circuit. The relationship between Current, Voltage and Resistance can be expressed by this formula: I=V/R known as Ohm’ s Law where I = Current, V = Voltage and R = Resistance. Knowing the values of any two of these allows you to solve the value of the third.

In the following diagram, the current through R1 is: I=U/R=5V/10kΩ=0.0005A=0.5mA.

WARNING:

Never connect the two poles of a power supply with anything of low resistance value (i.e. a metal object or bare wire) this is a Short and results in high current that may damage the power supply and electronic components.

Note:

Unlike LEDs and Diodes, Resistors have no poles and re non-polar (it does not matter which direction you insert them into a circuit, it will work the same)

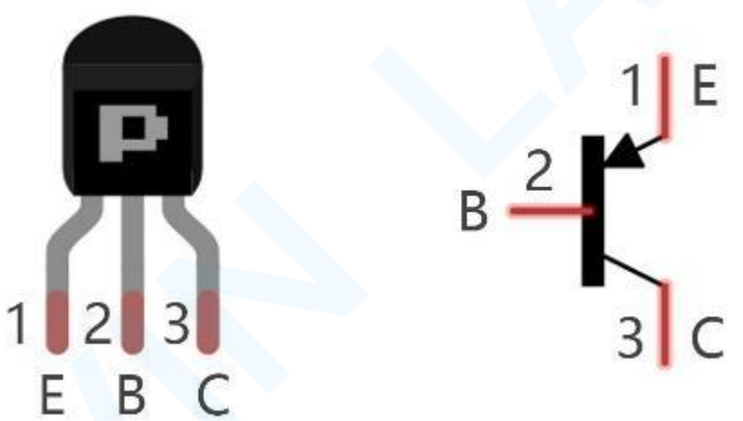
➢ Transistor

Because the buzzer requires such large current that GP of Raspberry Pi Pico output capability cannot meet the requirement, a transistor of NPN type is needed here to amplify the current.

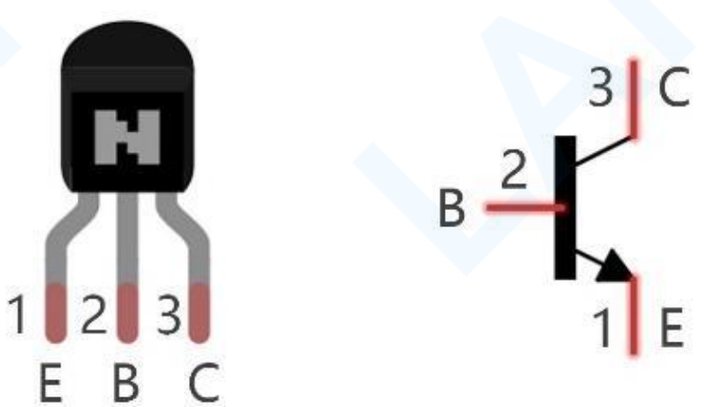
Transistor, the full name: semiconductor transistor, is a semiconductor device that controls current. Transistor

can be used to amplify weak signal, or works as a switch. It has three electrodes(PINs): base (b), collector (c) and emitter (e). When there is current passing between "be", "ce" will allow several-fold current (• t between "be" exceeds a certain value, "ce" will not allow current to increase any longer, at this point, transistor works in the saturation area. Transistor has two types as shown below: PNP and NPN.

PNP transistor



NPN transistor

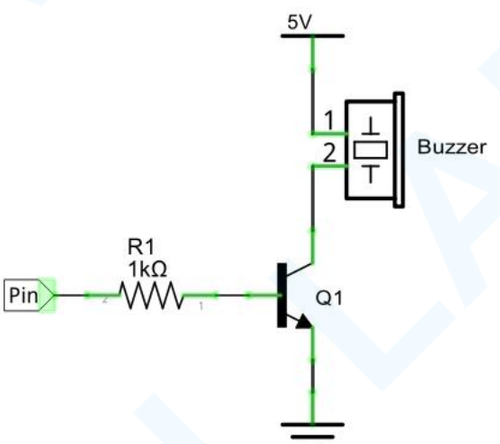


In our kit, There are two NPN transistor is marked with 8050.

Based on the transistor's characteristics, it is often used as a switch in digital circuits. As micro-controller's capacity to output current is very weak, we will use transistor to amplify current and drive large-current components.

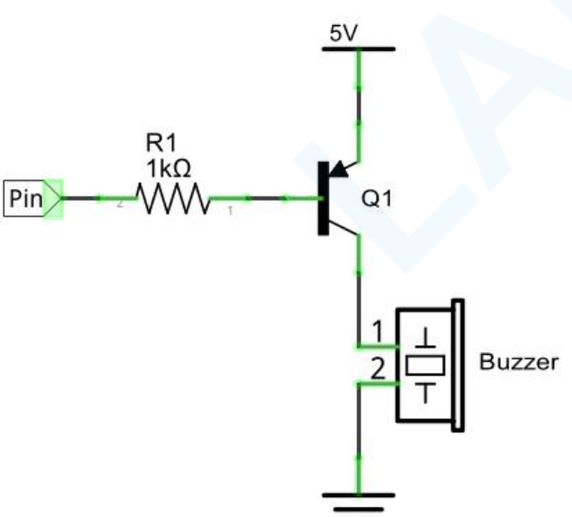
- When using NPN transistor to drive buzzer, we often adopt the following method. If GP outputs high level,current will flow through R1, the transistor will get conducted, and the buzzer will sound. If GP outputs low level, no current flows through R1, the transistor will not be conducted, and buzzer will not sound.

NPN transistor to drive buzzer



- When using PNP transistor to drive buzzer, we often adopt the following method. If GP outputs low level,current will flow through R1, the transistor will get conducted, and the buzzer will sound. If GP outputs high level, no current flows through R1, the transistor will not be conducted, and buzzer will not sound.

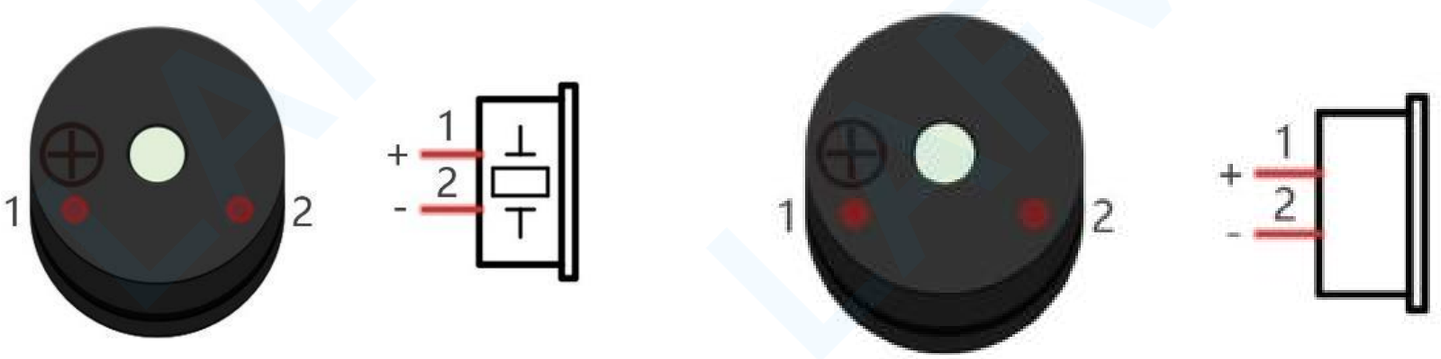
PNP transistor to drive buzzer



➢ Buzzer

Buzzer is a sounding component, which is widely used in electronic devices such as calculator, electronic warning clock and alarm. Buzzer has two types: active and passive. Active buzzer has oscillator inside, which will sound as long as it is supplied with power. Passive buzzer requires external oscillator signal (generally use PWM with different frequency) to make a sound.

Active buzzer **Passive buzzer**



Active buzzer is easy to use. Generally, it can only make a specific frequency of sound. Passive buzzer requires an external circuit to make a sound, but it can be controlled to make a sound with different frequency. The resonant frequency of the passive buzzer is 2kHz, which means the passive buzzer is loudest when its resonant frequency is 2kHz.

How to identify active and passive buzzer?

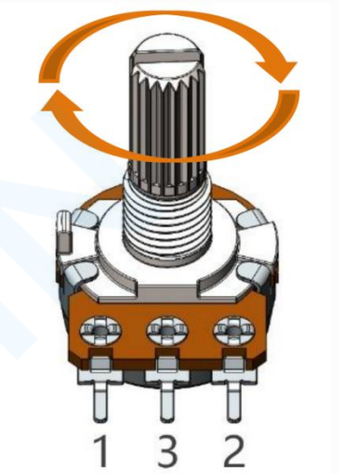
1. Usually, there is a label on the surface of active buzzer covering the vocal hole, but this is not an absolute judgment method.
2. Active buzzers are more complex than passive buzzers in their manufacture. There are many circuits and crystal oscillator elements inside active buzzers; all of this is usually protected with a waterproof coating(and a housing) exposing only its pins from the underside. On the other hand, passive buzzers do not have protective coatings on their

underside. From the pin holes viewing of a passive buzzer, you can see the circuit board, coils, and a permanent magnet (all or any combination of these components depending on the model.

Active buzzer

Passive buzzer

➢ Potentiometer

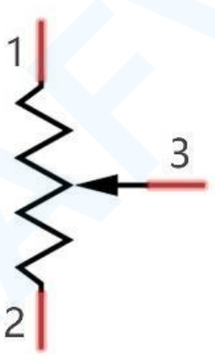


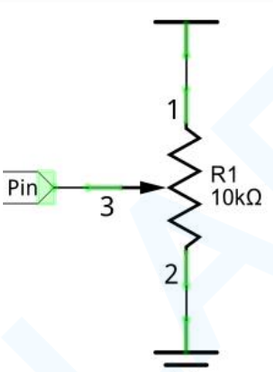
Potentiometer is also a resistance component with 3 terminals and its resistance value can be adjusted according to some regular variation.

Potentiometers come in various shapes, sizes, and values, but they all have the following things in common:

- They have three terminals (or connection points).
- They have a knob, screw, or slider that can be moved to vary the resistance between the middle terminal and either one of the outer terminals.
- The resistance between the middle terminal and either one of the outer terminals varies from 0 Ω to the maximum resistance of the pot as the knob, screw, or slider is moved.

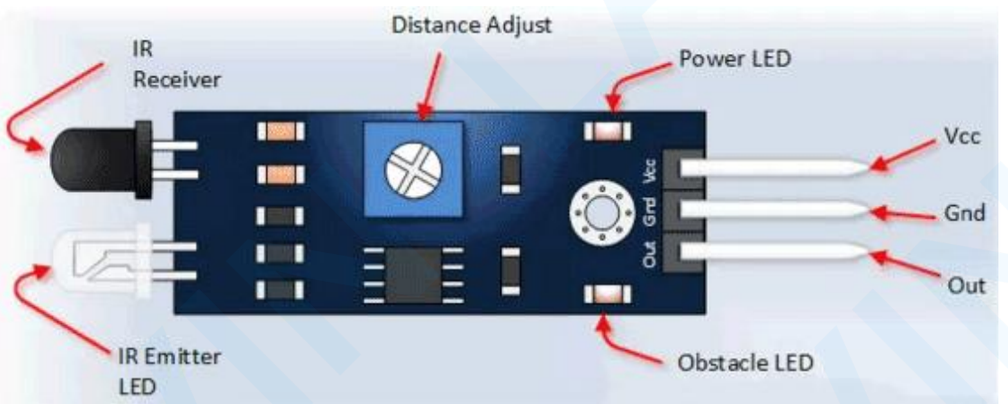
Here is the circuit symbol of potentiometer.10K ohm potentiometer is included in the kit





➢ IR Proximity Sensor Module

Proximity Sensor are used to detect objects and obstacles in front of sensor. Sensor keeps transmitting infrared light and when any object comes near, it is detected by the sensor by monitoring the reflected light from the object. It can be used in robots for obstacle avoidance, for automatic doors, for parking aid devices or for security alarm systems, or contact less tachometer by measuring RPM of rotation objects like fan blades.

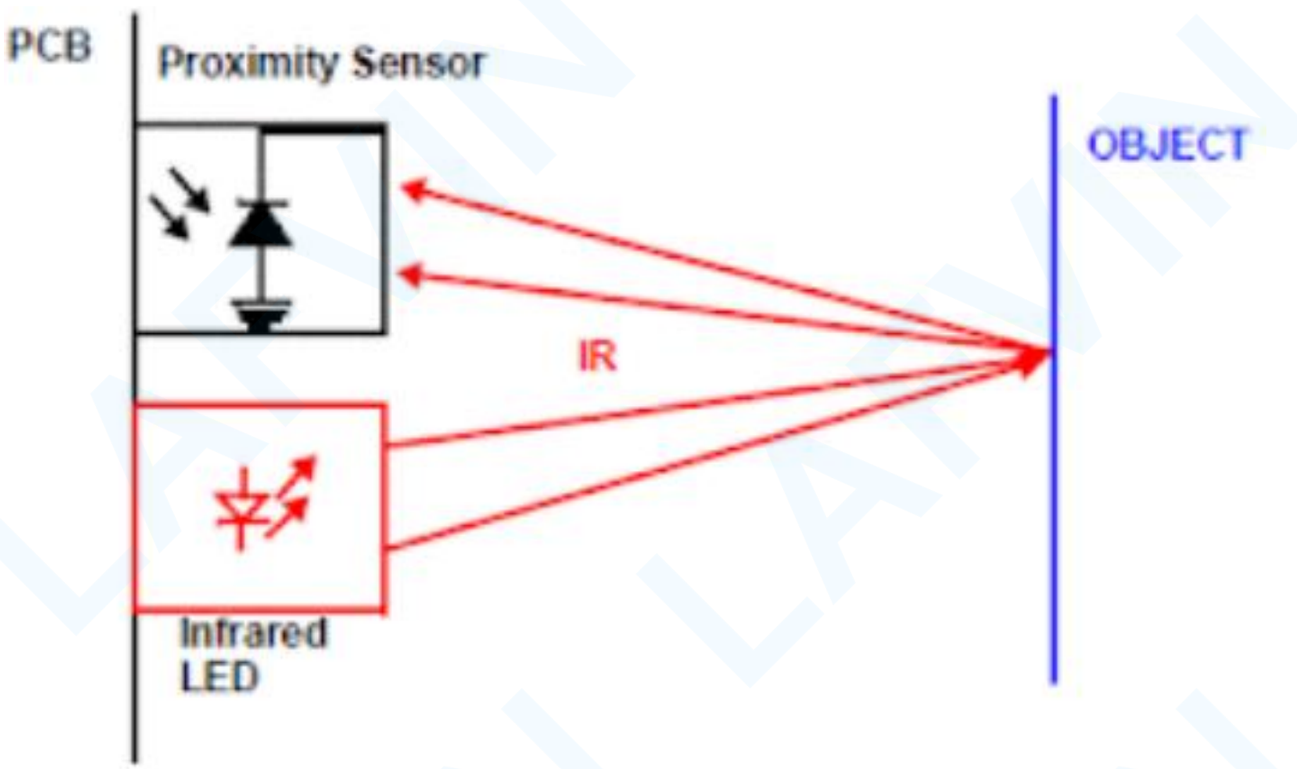


Pin, Control Indicator	Description
Vcc	3.3 to 5 Vdc Supply Input
Gnd	Ground Input
Out	Output that goes low when obstacle is in range
Power LED	Illuminates when power is applied
Obstacle LED	Illuminates when obstacle is detected
Distance Adjust	Adjust detection distance. CCW decreases distance. CW increases distance.
IR Emitter	Infrared emitter LED
IR Receiver	Infrared receiver that receives signal transmitted by Infrared emitter.

Digital OUTPUT

Digital **LOW** output on detecting objects in front.

Digital **HIGH** output on detecting no objects in front.



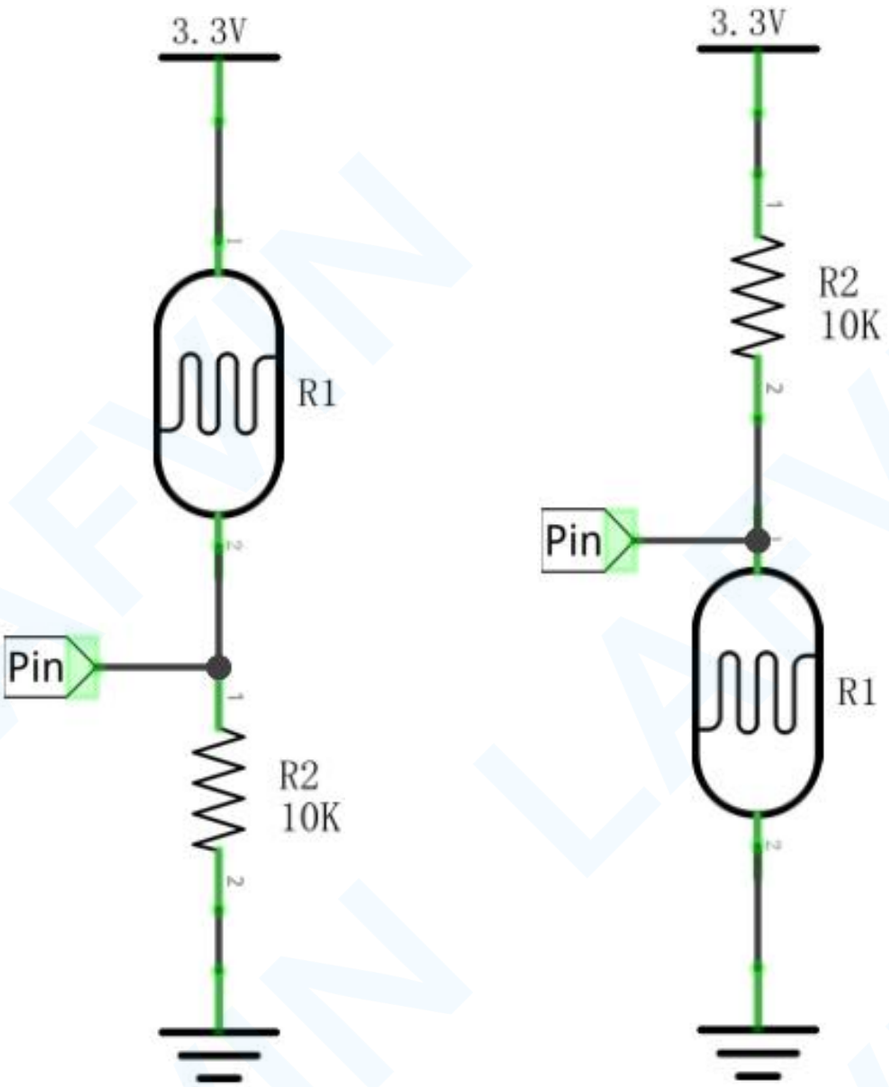
Distance Adjust

IR sensors are highly susceptible to ambient light and the IR sensor on this sensor is suitably covered to reduce effect of ambient light on the sensor. To For maximum, range the on board potentiometer should be used to calibrate the sensor. To set the potentiometer, use a screw driver and turn the potentiometer till the output LED just turns off.

➢ Photoresistor

Photoresistor is simply a light sensitive resistor. It is an active component that decreases resistance with respect to receiving luminosity (light) on the component's light sensitive surface. Photoresistor’s resistance value will change in proportion to the ambient light detected. With this characteristic, we can use a Photoresistor to detect light intensity. The Photoresistor and its electronic symbol are as follows.

The circuit below is used to detect the change of a Photoresistor’s resistance value:



In the above circuit, when a Photoresistor's resistance value changes due to a change in light intensity, the voltage between the Photoresistor and Resistor R1 will also change. Therefore, the intensity of the light can be obtained by measuring this voltage.

➢ Thermistor

A Thermistor is a temperature sensitive resistor. When it senses a change in temperature, the resistance of the Thermistor will change. We can take advantage of this characteristic by using a Thermistor to detect temperature intensity. A Thermistor and its electronic symbol are shown below.

➢ Thermistor - Wikipedia

The relationship between resistance value and temperature of a thermistor is:

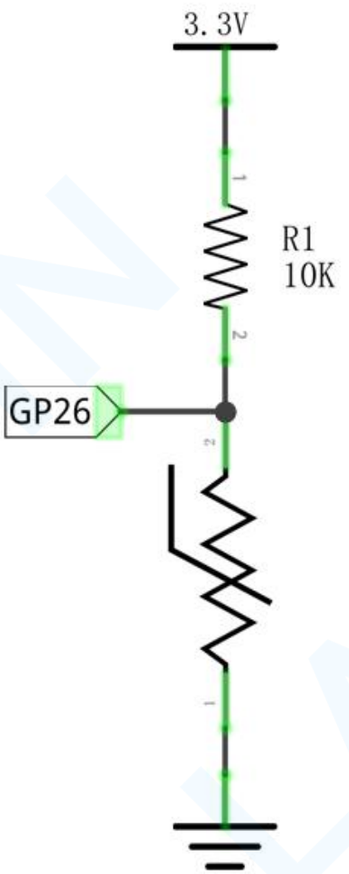
$$Rt = R * EXP[\ B * (\frac{1}{T2} - \frac{1}{T1})\ ]$$

Where:

- **Rt** is the thermistor resistance under T2 temperature;
- **R** is the nominal resistance of thermistor under T1 temperature;
- **EXP[n]** is nth power of e;
- **B** is for thermal index;
- **T1**, **T2** is Kelvin temperature (absolute temperature). Kelvin temperature=273.15 + Celsius temperature.

For the parameters of the Thermistor, we use: B=3950, R=10kΩ, T1=25℃.

The circuit connection method of the Thermistor is similar to photoresistor, as the following:



We can use the value measured by the ADC converter to obtain the resistance value of Thermistor, and then we can use the formula to obtain the temperature value.

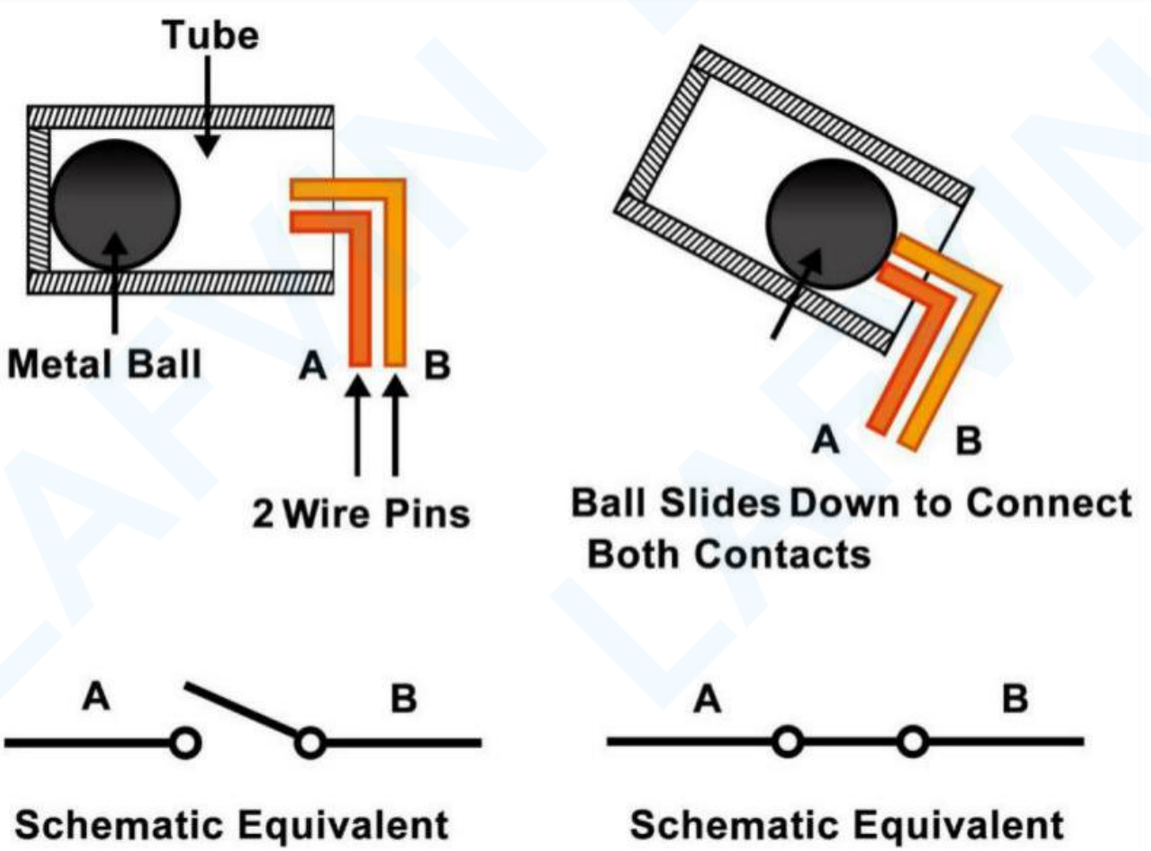
Therefore, the temperature formula can be derived as:

$$T2 = 1/(\frac{1}{T1} + \ln(\frac{Rt}{R})/B)$$

➢ Tilt Switch

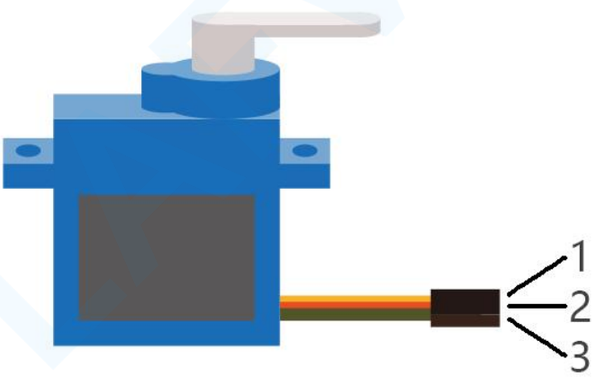
The tilt switch used here is a ball one with a metal ball inside. It is used to detect inclinations of a small angle.

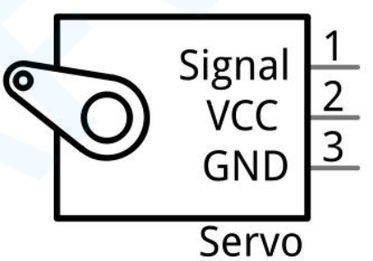
The principle is very simple. When the switch is tilted in a certain angle, the ball inside rolls down and touches the two contacts connected to the pins outside, thus triggering circuits. Otherwise the ball will stay away from the contacts, thus breaking the circuits.



➢ Servo

Servo is a compact package which consists of a DC Motor, a set of reduction gears to provide torque, a sensor and control circuit board. Most Servos only have a 180-degree range of motion via their “horn”. Servos can output higher torque than a simple DC Motor alone and they are widely used to control motion in model cars,model airplanes, robots, etc. Servos have three wire leads which usually terminate to a male or female 3-pin plug. Two leads are for electric power: Positive (2-VCC, Red wire), Negative (3-GND, Brown wire), and the signal line (1-Signal, Orange wire) as represented in the Servo provided in your Kit.





We will use a 50Hz PWM signal with a duty cycle in a certain range to drive the Servo. The time interval of 0.5ms-2.5ms of PWM single cycle high level corresponds to the Servo angle 0 degrees - 180 degrees linearly.

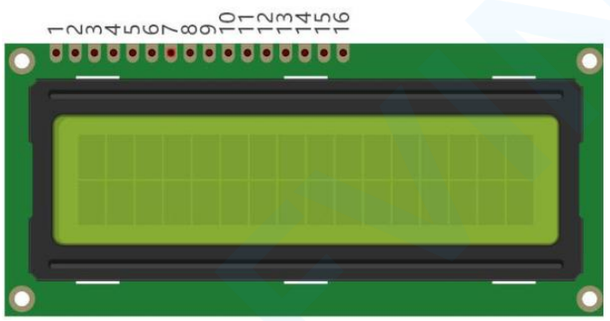
Part of the corresponding values are as follows:

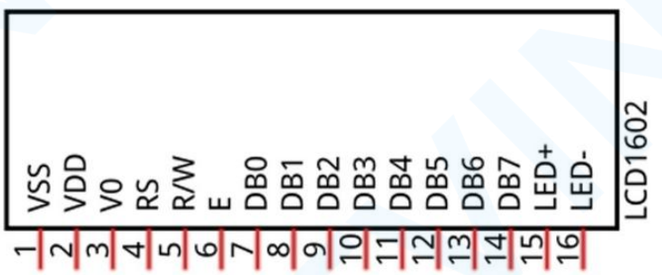
High level time	Servo angle
0.5ms	0 degree
1ms	45 degree
1.5ms	0 degree
2ms	45 degree
2.5ms	180 degree

When you change the Servo signal value, the Servo will rotate to the designated angle.

➢ I2C LCD1602

The LCD1602 Display Screen can display 2 lines of characters in 16 columns. It is capable of displaying numbers,letters, symbols, ASCII code and so on. As shown below is a monochrome LCD1602 Display Screen along with its circuit pin diagram.

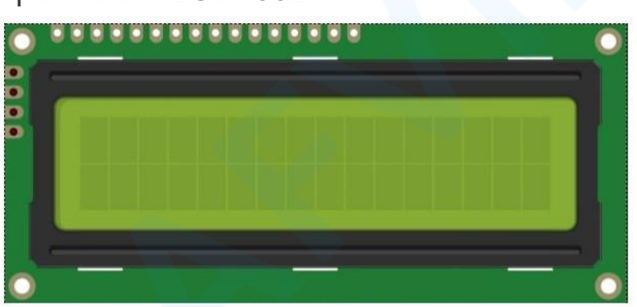


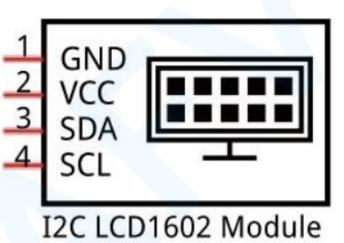


As we all know, though LCD and some other displays greatly enrich the man-machine interaction, they share a common weakness. When they are connected to a controller, multiple IOs will be occupied of the controller which has no so many outer ports. Also it restricts other functions of the controller. Therefore, LCD1602 with an I2C bus is developed to solve the problem.

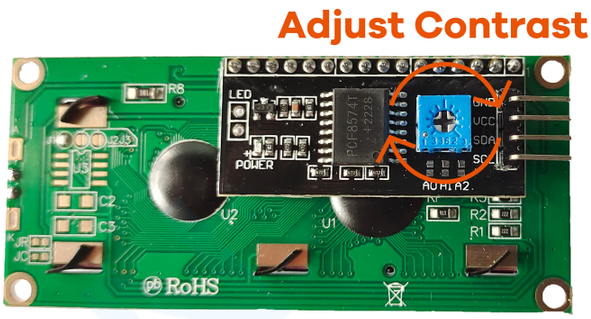
➢ Inter-Integrated Circuit - Wikipedia

I2C LCD1602 Display Screen integrates an I2C interface, which connects the serial-input & parallel-output module to the LCD1602 Display Screen. This allows us to use only 4 lines to the operate the LCD1602.





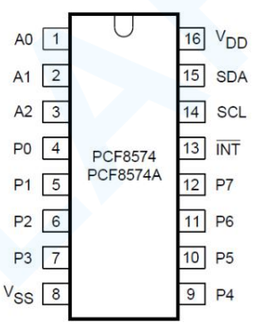
The serial-to-parallel IC chip used in this module is PCF8574T (PCF8574AT), and its default I2C address is 0x27(0x3F).



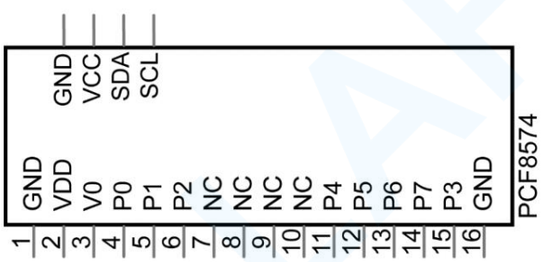
Potentiometer: It is used to adjust the contrast (the clarity of the displayed text), which is increased in the clockwise direction and decreased in the counterclockwise direction.

Below is the PCF8574 pin schematic diagram and the block pin diagram:

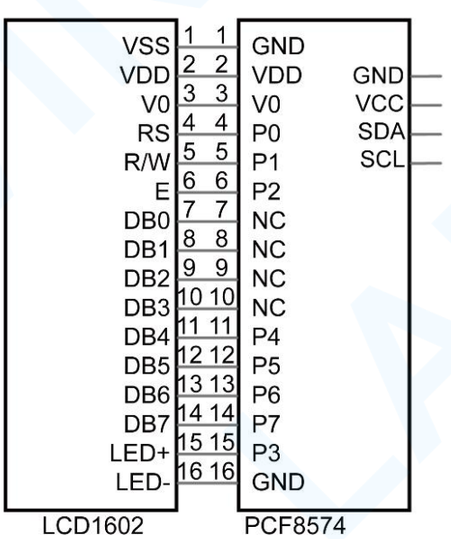
PCF8574 chip pin diagram:



PCF8574 module pin diagram



PCF8574 module pin and LCD1602 pin are corresponding to each other and connected with each other:

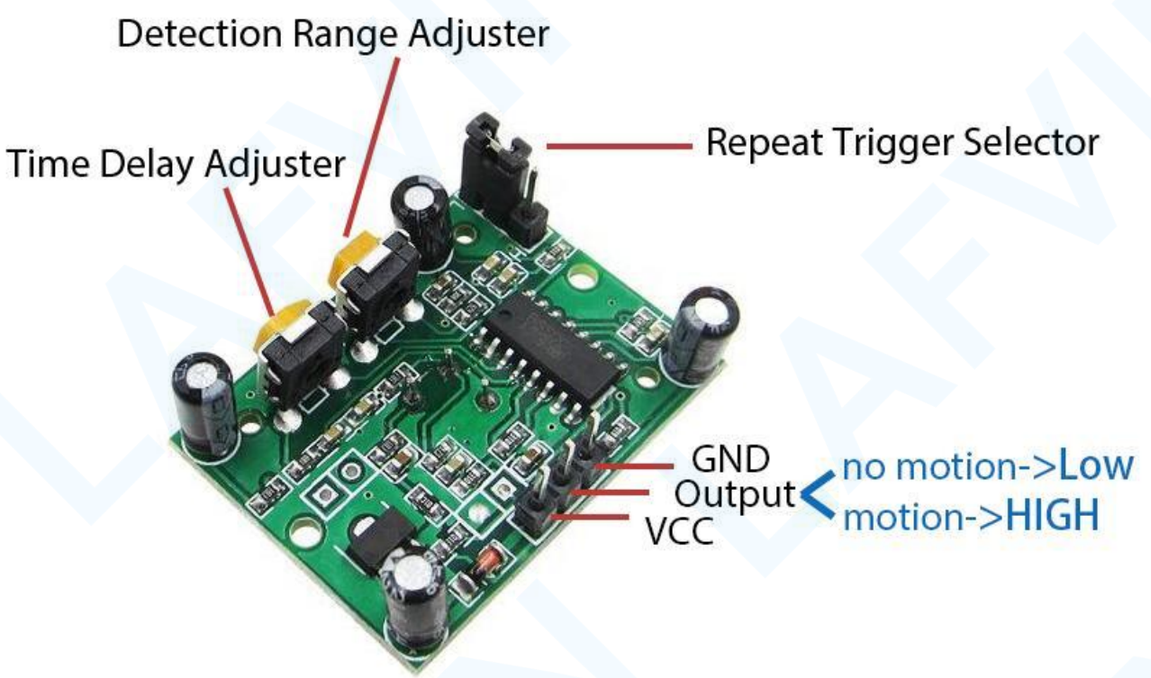


So we only need 4 pins to control the 16 pins of the LCD1602 Display Screen through the I2C interface.

we will use the I2C LCD1602 to display some static characters and dynamic variables.

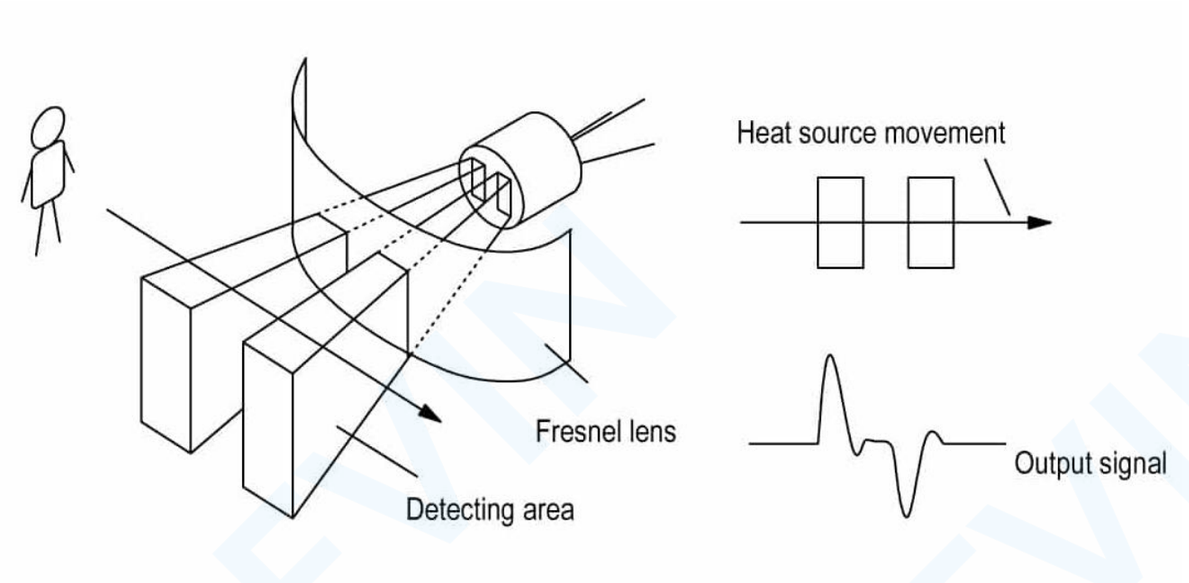
➢ PIR Motion Sensor

Passive infrared sensor (PIR sensor) is a common sensor that can measure infrared (IR) light emitted by objects in its field of view. Simply put, it will receive infrared radiation emitted from the body, thereby detecting the movement of people and other animals. More specifically, it tells the main control board that someone has entered your room.



The PIR sensor detects infrared heat radiation that can be used to detect the presence of organisms that emit infrared heat radiation.

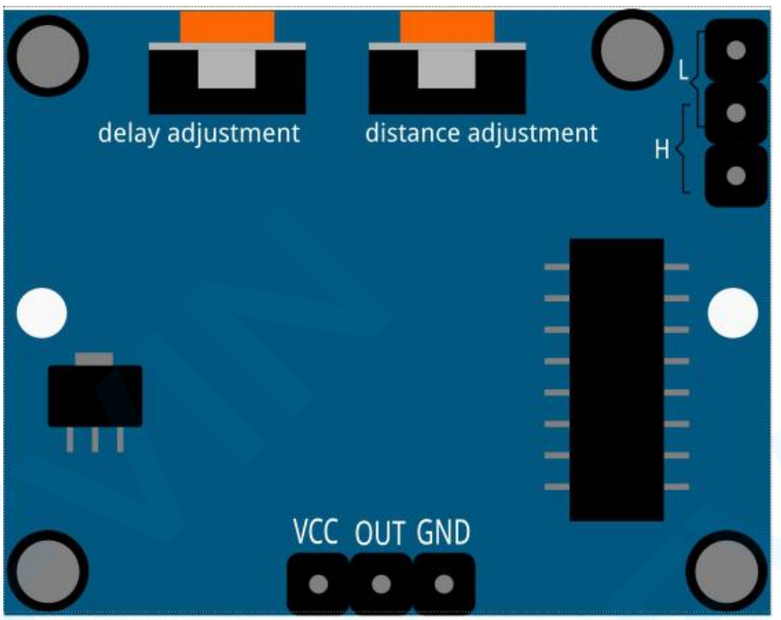
The PIR sensor is split into two slots that are connected to a differential amplifier. Whenever a stationary object is in front of the sensor, the two slots receive the same amount of radiation and the output is zero. Whenever a moving object is in front of the sensor, one of the slots receives more radiation than the other , which makes the output fluctuate high or low. This change in output voltage is a result of detection of motion.



Note:（Importance）

After the sensing module is wired, there is a one-minute initialization.After initialization,then the module will be in the standby mode.During the initialization,do not let any triggered infrared signal appear in the PIR monitoring range, including your hand. Otherwise in standby mode, it may cause false trigger detection.During the initialization, module will output for 0~3 times at intervals.This is not a real trigger result and you can ignore it until standby mode .

Please keep the interference of light source and other sources away from the surface of the module so as to avoid the misoperation caused by the interfering signal. Even you'd better use the module without too much wind, because the wind can also interfere with the sensor.



Distance Adjustment

Turning the knob of the distance adjustment potentiometer clockwise, the range of sensing distance increases, and the maximum sensing distance range is about 0-7 meters. If turn it anticlockwise, the range of sensing distance is reduced, and the minimum sensing distance range is about 0-3 meters.

Delay adjustment

Rotate the knob of the delay adjustment potentiometer clockwise, you can also see the sensing delay increasing. The maximum of the sensing delay can reach up to 300s. On the contrary, if rotate it anticlockwise, you can shorten the delay with a minimum of 1s.Usually you need to set the delay time to a minimum of 1s.

Two Trigger Modes

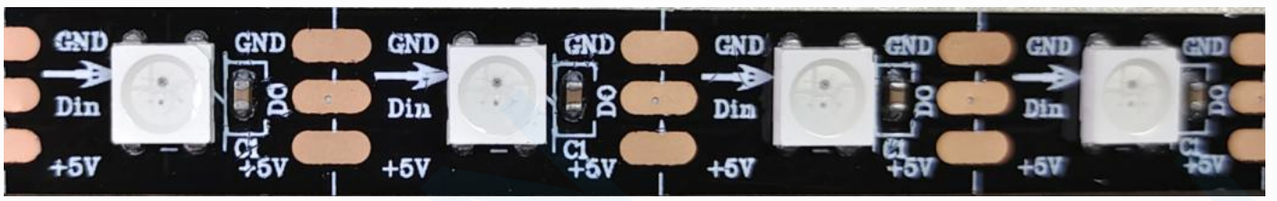
Choosing different modes by using the jumper cap.

H: Repeatable trigger mode, after sensing the human body, the module outputs high level.

During the subsequent delay period, if somebody enters the sensing range,the output will keep being the high level.

L：Non-repeatable trigger mode, outputs high level when it senses the human body. After the delay, the output will change from high level into low level automatically.

➢ WS2812 RGB 8 LEDs Strip



The WS2812 RGB 8 LEDs Strip is composed of 8 RGB LEDs. Only one pin is required to control all the LEDs. Each RGB LED has a WS2812 chip, which can be controlled independently. It can realize 256-level brightness display and complete true color display of 16,777,216 colors. At the same time, the pixel contains an intelligent digital interface data latch signal shaping amplifier drive circuit, and a signal shaping circuit is built in to effectively ensure the color height of the pixel point light Consistent.

It is flexible, can be docked, bent, and cut at will, and the back is equipped with adhesive tape, which can be fixed on the uneven surface at will, and can be installed in a narrow space.

Features

- Work Voltage: DC5V
- IC: One IC drives one RGB LED
- Consumption: 0.3w each LED
- Working Temperature: -15-50
- Color: Full color RGB
- RGB Type：5050RGB（Built-in IC WS2812B）
- Light Strip Thickness: 2mm
- Each LED can be controlled individually

WS2812B Introduction

- WS2812B Datasheet

WS2812B is a intelligent control LED light source that the control circuit and RGB chip are integrated in a package of 5050 components. It internal include intelligent digital port data latch and signal reshaping amplification drive circuit. Also include a precision internal oscillator and a 12V voltage programmable constant current control part, effectively ensuring the pixel point light color height consistent.

The data transfer protocol use single NZR communication mode. After the pixel power-on reset, the DIN port receive data from controller, the first pixel collect initial 24bit data then sent to the internal data latch, the other data which reshaping by the internal signal reshaping amplification circuit sent to the next cascade pixel through the DO port. After transmission for each pixel，the signal to reduce 24bit. pixel adopt auto reshaping transmit technology, making the pixel cascade number is not limited the signal transmission, only depend on the speed of signal transmission.

LED with low driving voltage, environmental protection and energy saving, high brightness, scattering angle is large, good consistency, low power, long life and other advantages. The control chip integrated in LED above becoming more simple circuit, small volume, convenient installation.

FAQ

Question:

Can Raspberry Pi Pico be used on Thonny, Arduino IDE and Piper Make at the same time?

Answer:

No, Raspberry Pi Pico only supports one kind of firmware installation at a time, you need to do some different operations according to the following situations.

①If you use it on Arduino IDE or Piper Make first, and now you want to use it on Thonny IDE, you need to **Burn Micropython Firmware** to your pico. Check out the "**Tutorial For MicroPython User**" folder.

②If you use it on Thonny or Piper Make first, and now you want to use it on Arduino IDE, you need to **Upload Arduino-compatible Firmware** for Pico. Check out the "**Tutorial For Arduino User**" folder.

③If you use it on Arduino IDE or Thonny first, and now you want to use it on Piper Make, you need to **Burn piper circuitpython Firmware** to your pico. Check out the "**Tutorial For Piper Make User**" folder.