Infrared control sound

1. Learning target

1.1 In this course, we will learn how to use pins of the Raspberry Pi Pico board.

1.2 How to use infrared receiver, infrared remote controller, buzzer make an infrared remote control tuner.

2. Preparation

Raspberry Pi Pico board *1

Pico sensor expansion board *1

PC *1

USB data cable *1

Infrared receiver *1

Infrared remote controller *1

Buzzer module *1

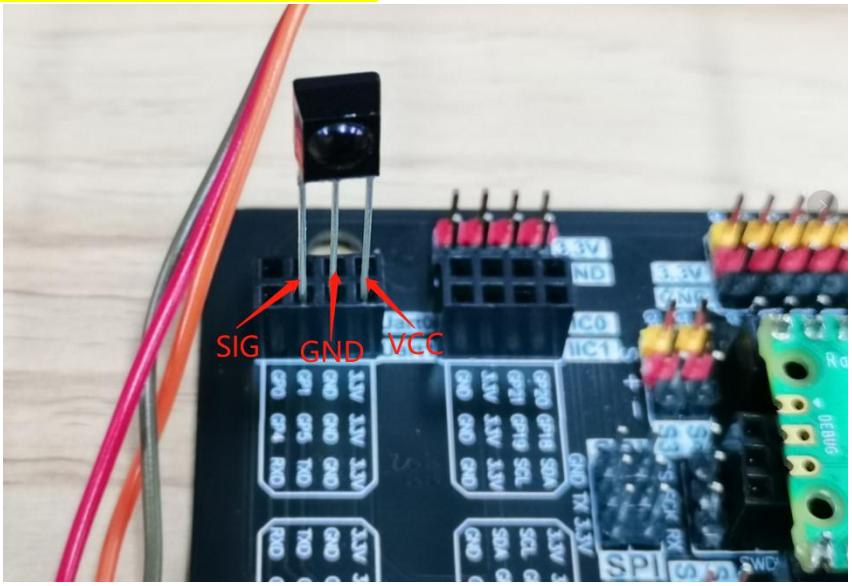
Female-to-male DuPont line *3

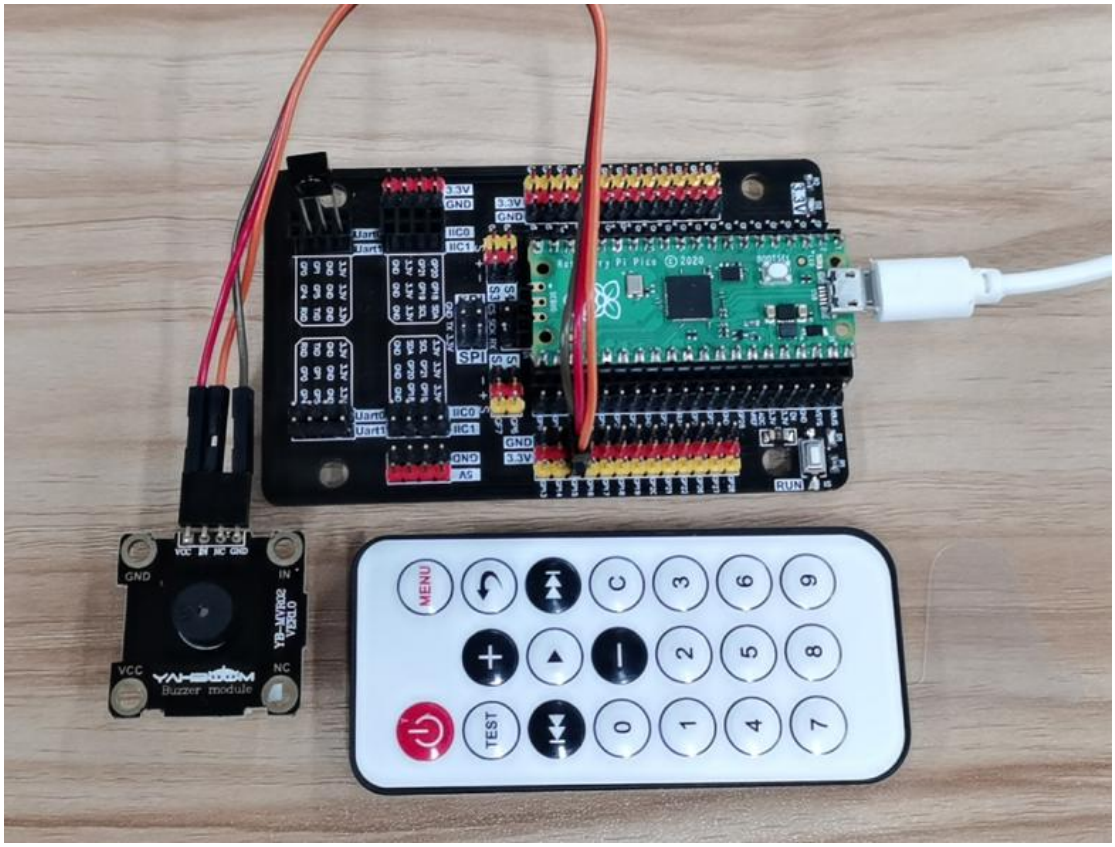
3. About wiring

Infrared receiver	Pico sensor expansion board
VCC	3.3V
GND	GND
SIG	GP5

Buzzer module	Pico sensor expansion board
IN	GP15
VCC	3.3V
GND	GND

Note: Do not insert the infrared receiver in the wrong direction!
Inserting it in the wrong direction will destroy the infrared receiver module. If the module is hot, it means that the module has been damaged.





4. About code

Thonny programming

About how to using ThonnyIDE, please check the tutorials in【2.Development environment】.

```
from machine import Pin, PWM
import utime
from ir import ir

# Initialize the buzzer pin to PWM function
buzzer = PWM(Pin(15))
buzzer.freq(262)
buzzer.duty_u16(0)

# Play the frequency of midrange tones 1-7
freq = [262, 294, 330, 350, 393, 441, 495]

#Configure infrared receiving pin
pin = Pin(5, Pin.IN, Pin.PULL_UP)

#Configure infrared receiver library
Ir = ir(pin)

#Corresponding code value of infrared remote control keyboard
```

```
POWER = 69
MENU = 71
TEST = 68
PLUS = 64
RECALL = 67
BACKWARD = 7
PLAY = 21
FORWARD = 9
NUM_0 = 22
REDUCE = 25
CLEAR = 13
NUM_1 = 12
NUM_2 = 24
NUM_3 = 94
NUM_4 = 8
NUM_5 = 28
NUM_6 = 90
NUM_7 = 66
NUM_8 = 82
NUM_9 = 74

# Set the buzzer to emit different tones.
# index=[0-7], where 0 is closed, and 1-7 respectively represent middle C, middle D, middle E, middle F, middle G, middle A, middle B.
# time represents the function delay time (a positive integer), in milliseconds.
# auto_off indicates whether the buzzer will be turned off automatically after the delay time.
def tone(index, time=0, auto_off=False):
    if index == 0:
        buzzer.duty_u16(0)
        utime.sleep_ms(time)
    elif index >= 1 and index <= 7:
        tone_freq = freq[int(index - 1)]
        buzzer.freq(tone_freq)
        buzzer.duty_u16(32768)
        utime.sleep_ms(time)
        if auto_off == True:
            buzzer.duty_u16(0)
        # print("----freq:", index, tone_freq)
    else:
        print("Tones must be 0-7")


delay = 0
```

```
tone(1, 100, True)

while True:
    #Read remote control data
    value = Ir.Getir()
#       if value != None:
#           print(int(value))

    # Determine whether there is a button that meets the needs
    if value == NUM_1:
        tone(1, delay)
        print("NUM_1")
    elif value == NUM_2:
        tone(2, delay)
        print("NUM_2")
    elif value == NUM_3:
        tone(3, delay)
        print("NUM_3")
    elif value == NUM_4:
        tone(4, delay)
        print("NUM_4")
    elif value == NUM_5:
        tone(5, delay)
        print("NUM_5")
    elif value == NUM_6:
        tone(6, delay)
        print("NUM_6")
    elif value == NUM_7:
        tone(7, delay)
        print("NUM_7")
    else:
        tone(0)
```

Before running this program, you need to load IR library, please check the specific steps in 【2.Development environment】

5. Phenomenon

Click the green run button  of Thonny IDE to start running the program. Click the red stop button  to stop the program. When the program is running, if you press 1~7 key on infrared remote controller, you can hear the buzzer playing do, re, mi, fa, sol, la, si tones. Release the button

and the buzzer will stop playing.

(Note: Before using, we need remove the plastic sheet at the bottom of the infrared remote controller)