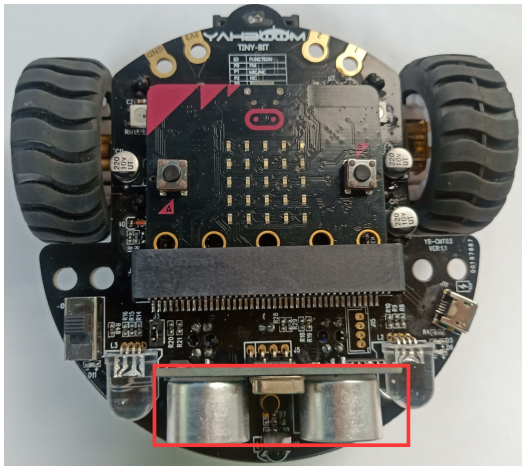
8.Measuring worker

Learning goal：

This lesson learns how to use the Ultrasonic sensor on the Tiny-bit car.

Preparation：

1.The position of the Ultrasonic module in the robot, as shown below.



2.Ultrasonic ranging principle:

The ultrasonic wave has two ultrasonic probes, which are used for transmitting and receiving ultrasonic waves, respectively, and the measurement range is about 3-450 cm. First, input a 15us high level signal to the SCL (TRIG) pin to start the ranging function of module.

After the ranging function is started, the module will automatically send out 8 40 kHz ultrasonic pulses and automatically detect whether there is a signal return. This step is automatically done internally by the module. When the echo signal is detected, the echo terminal SDA (ECHO) pin will output a high level.

The high level duration is the time from the transmission to the return of the ultrasonic wave. We can calculate the current distance by the high level duration.

Formula: Distance = High Time * Sound Speed (340M/S)/2.

From the hardware interface manual, we can know that the tracking sensor is directly driven by the micro:bit P15,P16 pin.

Category	Function	Number	Drive	The number of Drive pin	The number of connected to the controller	micro:bit
Buzzer	Buzzer	FM			FM	P0
Voice sensor	Voice sensor	MIC			MIC	P1
LED light	Water light	LED-RGB			LED-RGB	P12
Tracking sensor	Left tracking	L-DET			L-DET	P13
	Right tracking	R-DET		Micro:bit drive directly	R-DET	P14
Ultrasonic module	Echo pin	ECHO			ECHO	P15
	Trigger pin	TRIG			TRIG	P16
Infrared receiver	Infrared remote control	RX			RX	P8
I2C interface	I2C interface	SCL			SCL	P19
		SDA			SDA	P20
Motor	Left motor Forward	L-INA	STM8S	PC6/TIM1_CH1	SCL, SDA	P19, P20
	Left motor Reverse	L-INB		PC7/TIM1_CH2		
	Right motor Forward	R-INA		PC3/TIM1_CH3		
	Right motor Reverse	R-INB		PC4/TIM1_CH4		
RGB Searching light	Red	LED-R		PC5/TIM2_CH1		
	Green	LED-G		PD3/TIM2_CH2		
	Blue	LED-B		PD2/TIM2_CH3		

Code：

```
from microbit import *

length = 200

class HCSR04:
    def __init__(self, tpin=pin16, epin=pin15):
        spi.init(baudrate=125000, sclk=tpin, mosi=tpin, miso=epin)
        self.r = bytearray(length)

    def distance(self):
        pre = 0
        post = 0
        k = -1
        self.r[0] = 255
        spi.write_readinto(self.r, self.r)
        # find first non zero value
        try:
            i, value = next((ind, v) for ind, v in enumerate(self.r) if v)
        except StopIteration:
            i = -1
        if i > 0:
            pre = bin(value).count("1")
            try:
                k, value = next((ind, v)
                for ind, v in enumerate(self.r[i:length - 2]) if self.r[i + ind + 1] == 0)
                post = bin(value).count("1") if k else 0
                k = k + i
            except StopIteration:
                i = -1
        dist = -1 if i < 0 else round(((pre + (k - i) * 8. + post) * 8 * 0.172) / 2)
        return dist

sonar = HCSR04()

while True:
    distance = round(sonar.distance()/10)
    display.scroll(str(distance))
    sleep(500)
```

Programming and downloading：

1.You should open the Mu software, and enter the code in the edit window, , as shown in Figure 8-1.

Note! All English and symbols should be entered in English, and the last line must be a space.

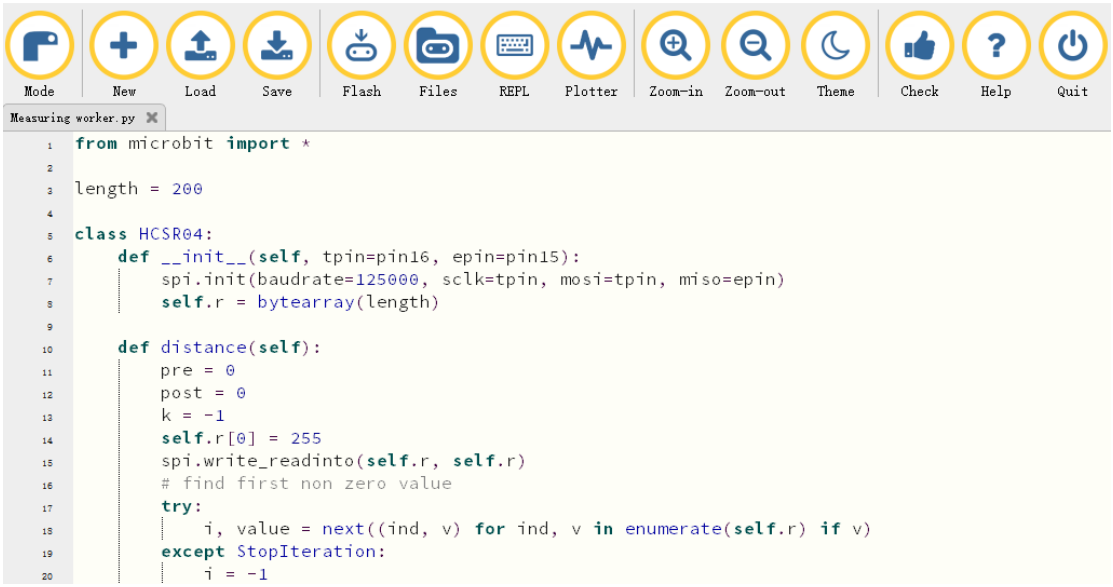


Figure 8-1

2.As shown in Figure 8-2, you need to click the Check button to check if our code has an error. If a line appears with a cursor or an underscore, the program indicating this line is wrong.

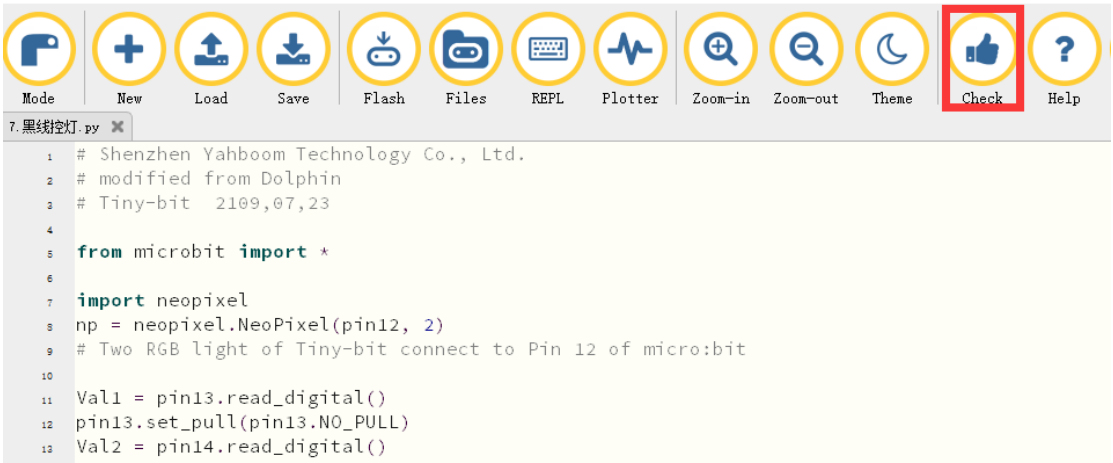


Figure 8-2

3.You need to connect the micro data cable to micro:bit and the computer, then click the Flash button to download the program to micro:bit as shown in Figure 8-3.

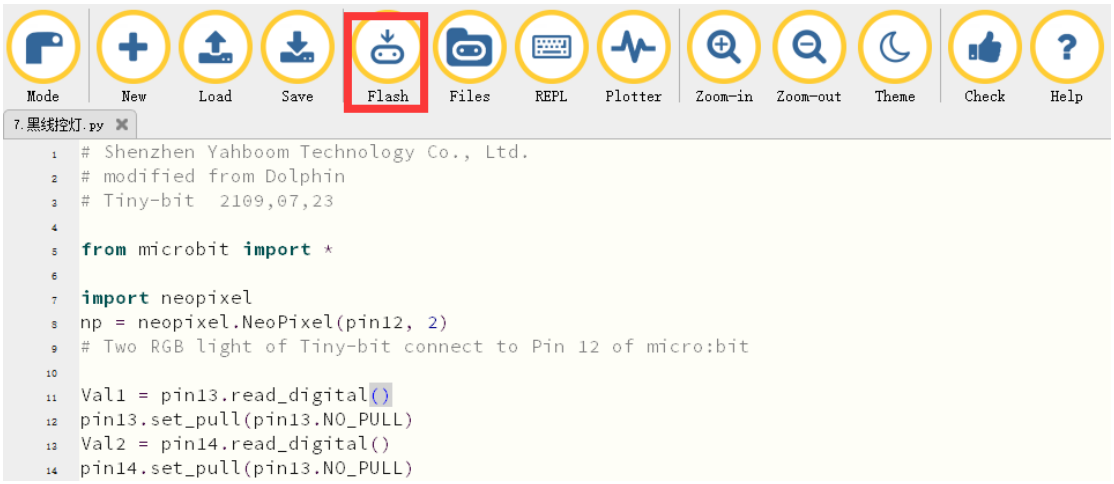


Figure 8-3

4. After downloading the program, we will see that the distance will be dispaly on the micro:bit dox matrix.