Lesson 12 Timer

1. Project Overview

Timer is used for timing, which is widely used in microcomputer chip. In this lesson, we're going to learn how to use ESP32 in timer.

2. Working Principle

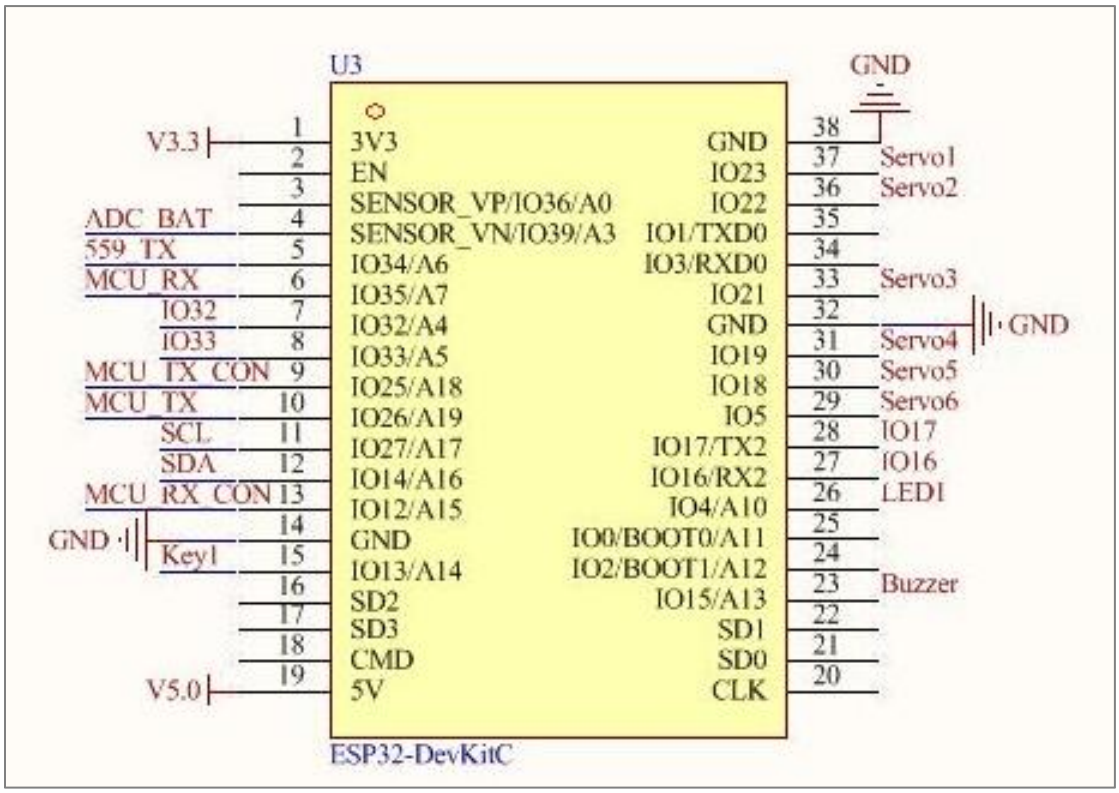
The path to the source code of the program is 5. Hardware Basic Learning/ Python Development/Program Files/Timer/main.py

```
from machine import Pin,Timer
import time
# Two LED lights are used. The LED on expansion board is set as LED1. The LED on ESP32 is set as LED2.
led1 = Pin(26,Pin.OUT)
led2 = Pin(2,Pin.OUT)
# Set the initial state of two LED
led1.value(1)
led2.value(1)
time.sleep_ms(100)
# Define a callback function to invert the outputs of led1 and led2
def fun(tim):
    led1.value(not led1.value())
    led2.value(not led2.value())
# Turn on ROS timer. The number is 2.
tim = Timer(2)
# period=1000 means execute once every 1000ms. "mode" represents execution mode.
# Timer.PERIODIC represents repeated execution. "callback" represents the callback function. "callback=fun" represents fun function is called after period ends.
tim.init(period=1000, mode=Timer.PERIODIC,callback=fun)
# Sleep only in the infinite loop
while True:
    time.sleep_ms(10)
```

Firstly, set the LED lights on expansion board and ESP32 controller to different status. Then flip the output of LED1 and LED2 by defining callback function. Finally, turn on the timer to make LED1 and LED2 alternatively light up, i.e, in this way, one is turned on and the other is turned off in one cycle, and the status of this two LEDs will be reversed in the next cycle, and so on.

According to the following circuit diagram, you can learn about the pin

information of buttons on ESP32 expansion board.



There are two operation modes in ESP32：

1) The timer runs with auto-reload. Interrupt-triggered callback events occur at intervals is based on a time value.

2) The timer runs under single mode, i.e, only run once.

RTOS timer is built in ESP32, so we just need to import Timer to use it easily.

3. Preparation

3.1 Hardware

MaxArm robotic arm, power adapter, USB cable.

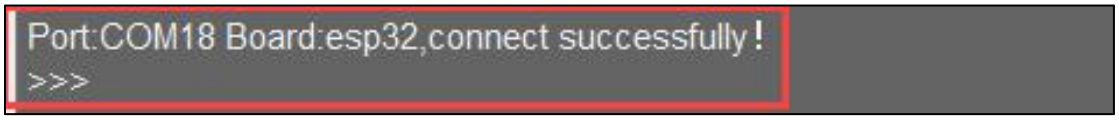
3.2 Software

Please refer to the material in folder “4.Underlying Program Learning/Python

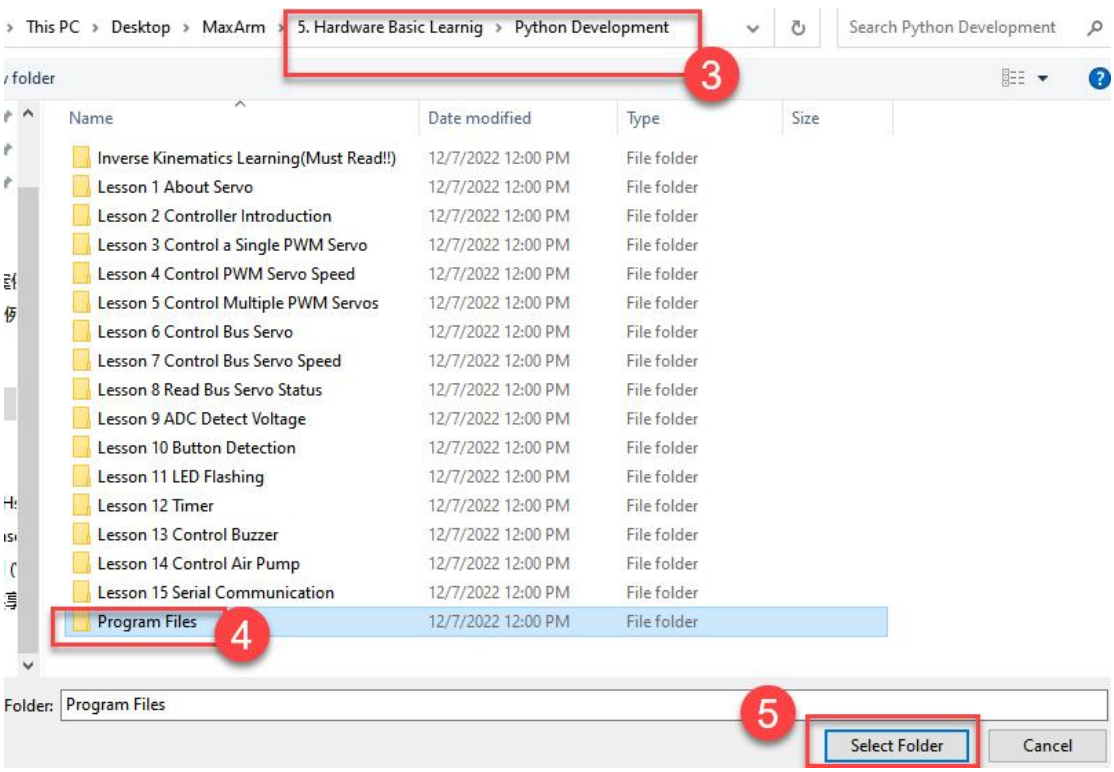
Development/Python Development/Lesson 1 Set Development Environment" to connect ESP32 controller to Python Editor.

4. Program Download

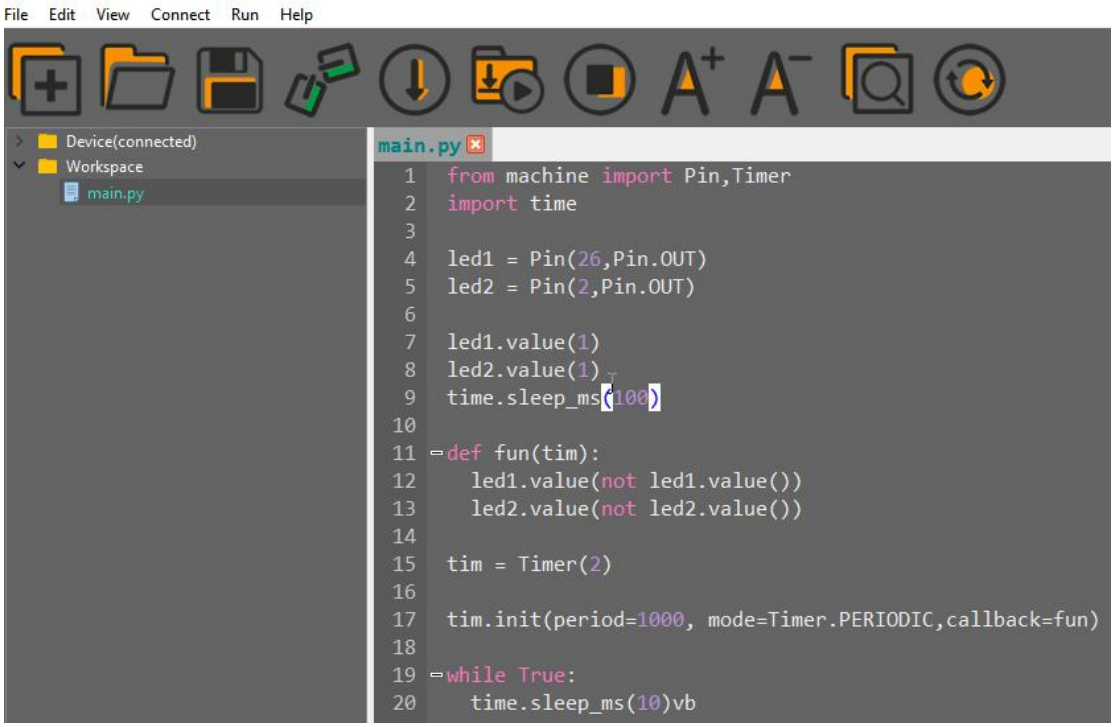
1) Please connect MaxArm to Python editor according to the tutorial in folder "4. Underlying Program Learning/Python Development/Lesson 1 Set Development Environment".



2) After connecting, change the path of Workspace to "5.Hardware Basic Learning/Python Development" and select "Program Files".



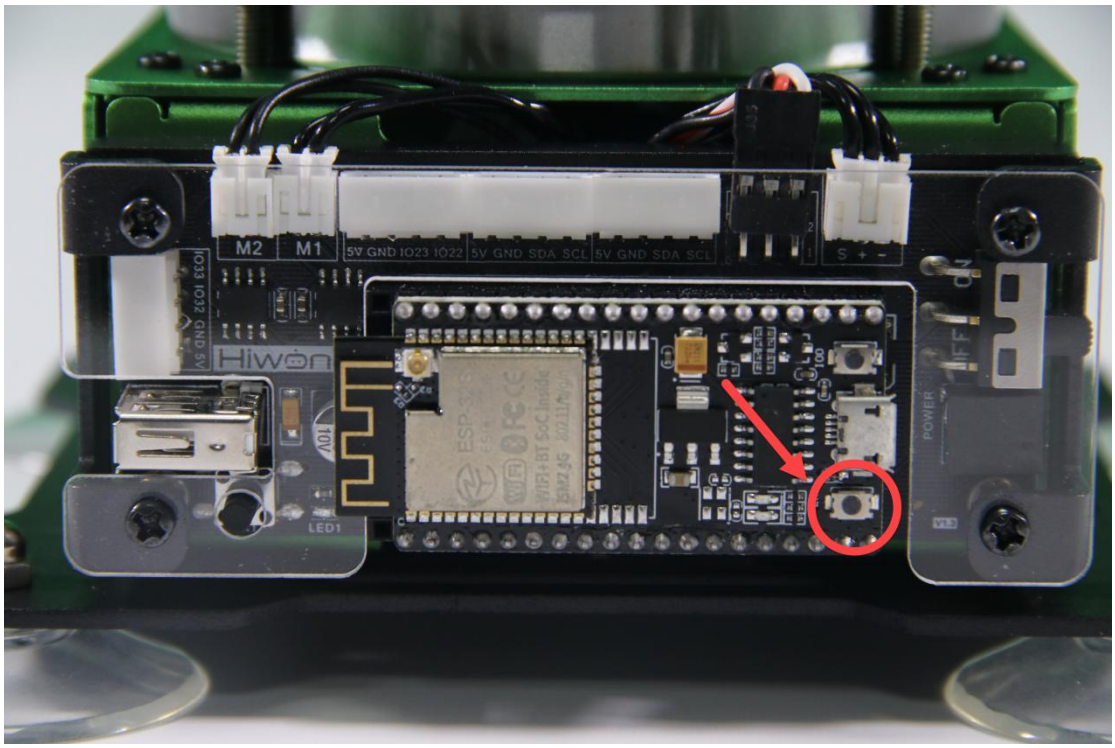
3) Double click the folder “Timer”, and then double click “main.py” to open program.



4) Click on the download icon to download program to ESP32 controller.

When the terminal prints the prompt, as shown in the image below, it means download completed.

5) After downloading, click on the reset icon or press the reset button on ESP32 controller to run program.



5. Project Outcome

The LED lights on controller turn on and off alternately.