Lesson 2 Underlying Program Overview

1. Underlying Program Composition

The following programs and library files must be contained to implement the complete function of MaxArm.

boot.py	main.py	BusServo.py	Key.py
Buzzer.py	Led.py	PWMServo.py	RobotControl.py
USBDevice.py	SuctionNozzle.py	__espmax.mpy	espmax.py

2. Program Starting Process

When turning on MaxArm, that it, power on ESP32 controller, it will experience the following steps:

1) "boot" program is started first. It is the first executed file after starting the system, which is usually an initialization process. If boot is null, just skip it.

2) Then execute main program and create Bluetooth object. The function libraries in "main.py" are called. When enabling Bluetooth service, the initialization process takes about 10s, which is normal.

3) After executing the main program, the LED on main controller will flash blue.

Note: All the functions of MaxArm need to be started by "main.py" including reset and boot up operations. If this program is stuck, no other operations can be performed later!

If user want to develop other functions, it is recommended to modify the program name first, such as "Test_main.py", and run this file manually every time.

Therefore, if some special stuck phenomenon happen (pressing Ctrl+C and Ctrl+D do not work it out), then you only need to reset controller and delete this program in "device" list, re-edit and download it!

3. Program Function Distribution

The following table is the corresponding function of the underlying programs, which is for reference. Please refer to the detailed code for secondary development.

Program	Function
boot.py	Initialization Program
main.py	Main program: use amount of function libraries
BusServo.py	Control bus servo: based on bus servo bus servo communication protocol
Buzzer.py	Control buzzer
Key.py	Control key
Led.py	Control LED

PWMServo.py	Control PWM servo
RobotControl.py	Call the action group of robotic arm
USBDevice.py	Controller USB driver program: parse serial data
SuctionNozzle.py	Control suction nozzle
__espmax.mpy	Kinematics base library: get solution through kinematics
espmax.py	Kinematic package library: call kinematics base library to get solution and drive servo