Servo Control

Servo Control
1. Software and Hardware
2. Basic Principles
    2.1 Hardware Schematic
    2.2 Physical Connection Diagram
    2.3 Control Principle
3. Project Experience
    3.1 Opening the Project
4. Main Functions
    4.1 User Functions
5. Experimental Phenomenon

This tutorial demonstrates: controlling a servo connected to the S1 interface on the development board through **timer interrupt** simulated PWM.

The first tutorial in this chapter will be more detailed than subsequent tutorials, with the purpose of demonstrating the complete process from creating a new project to achieving the final effect

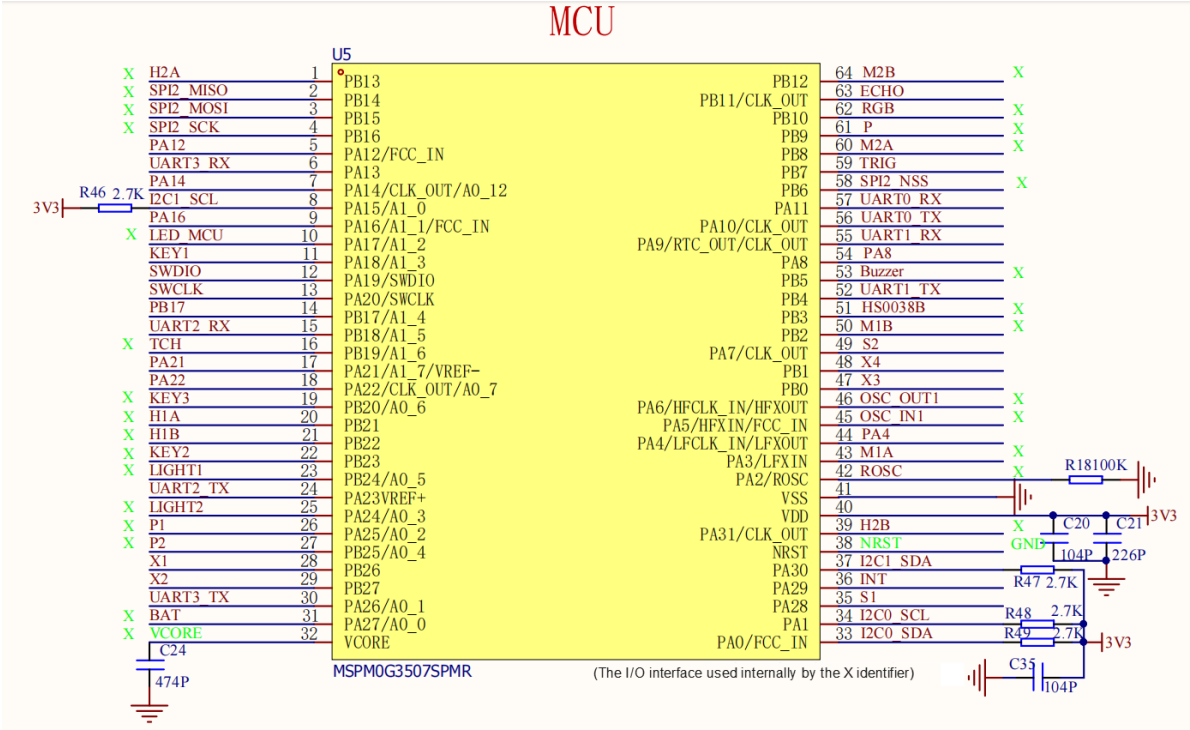
1. Software and Hardware

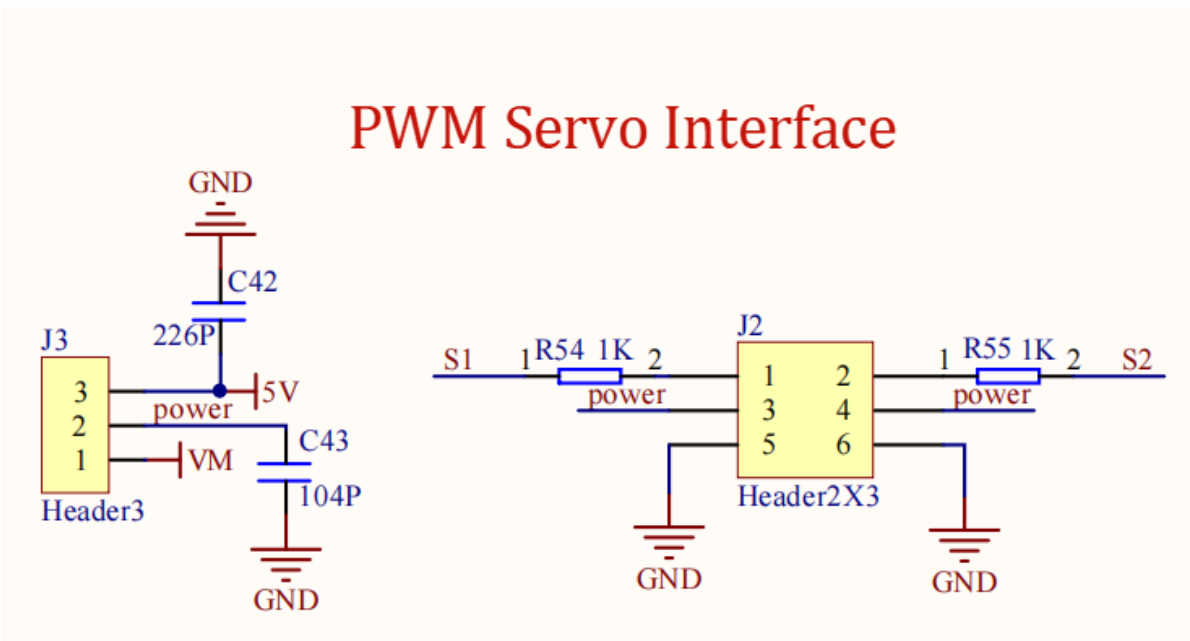
- **KEIL**
- **MSPM0G3507 Development Board**
- **180° Servo**
- **Type-C data cable or DAP-Link**

  For programming download or simulation to the development board

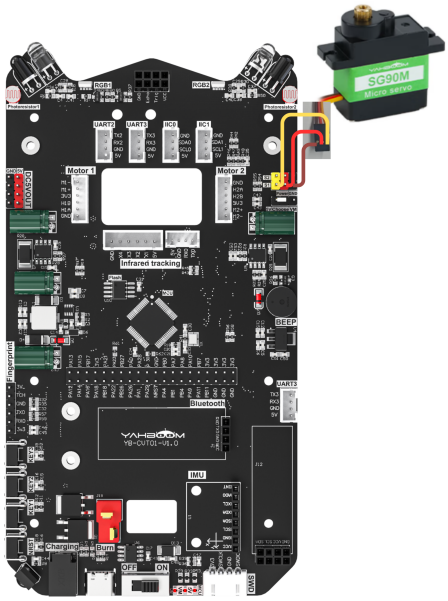
2. Basic Principles

2.1 Hardware Schematic





2.2 Physical Connection Diagram



Note: For servo wiring, refer to the servo wire colors. **This experiment requires connecting 5V and the choose jumper cap**

If using our 20k/25kg servo, you need to connect VM and choose to ensure proper operation.

Servo	Development Board
VCC(Red)	5V
SIG(Yellow)	S1/S2
GND(Brown)	GND

2.3 Control Principle

Control the servo rotation angle by changing the duty cycle of the PWM signal

- **PWM (Pulse Width Modulation)**

PWM is the abbreviation for Pulse Width Modulation, a technique that controls levels by adjusting the pulse width of signals.

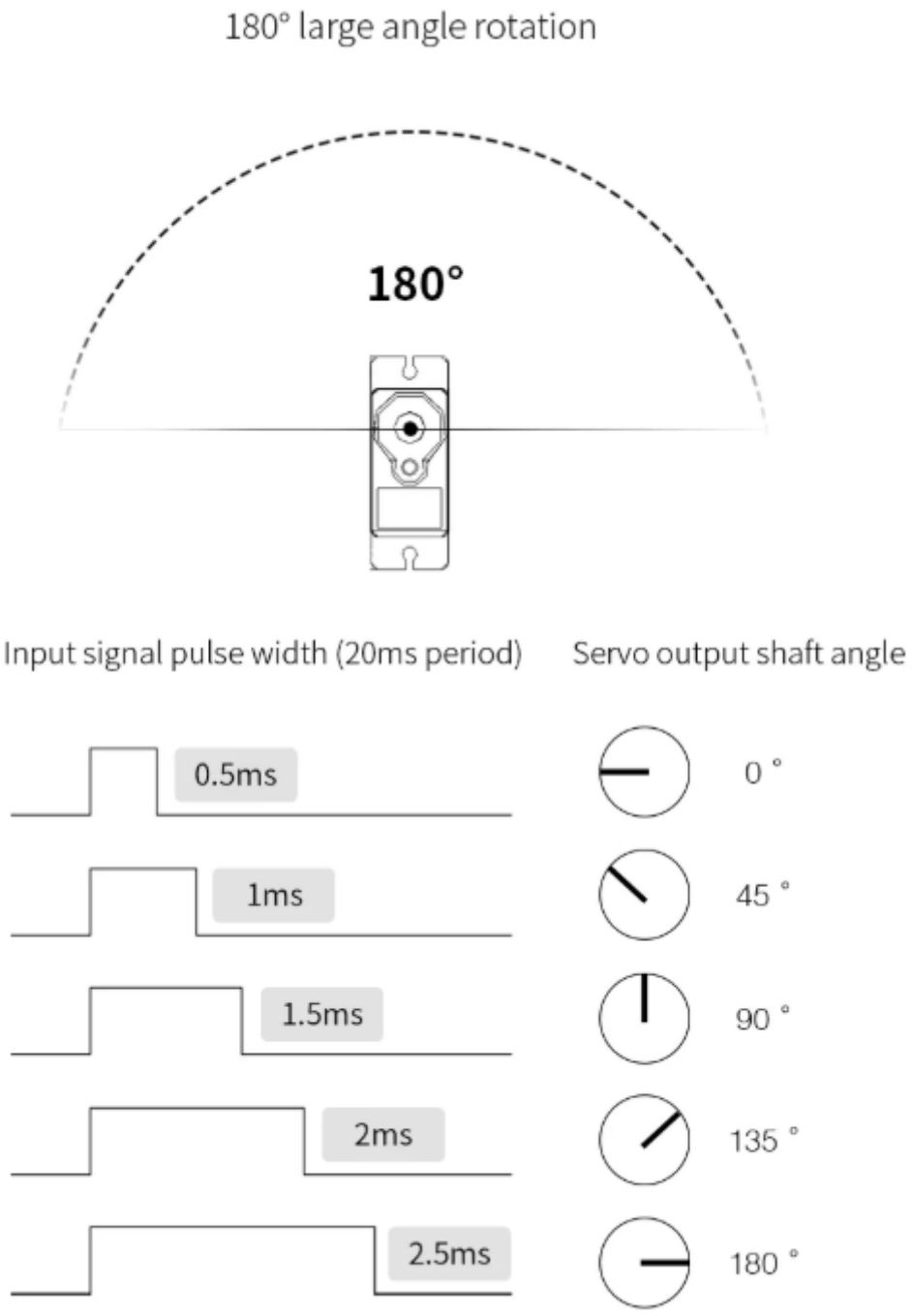
Period: The duration of one complete PWM waveform;

Duty Cycle: The ratio of high-level duration to the total period time;

Frequency: The reciprocal of the period is called frequency, i.e., the number of PWM cycles generated per second;

- **PWM Servo**

Set the PWM control signal period to 20ms, i.e., 50Hz frequency; the high-level time of the pulse determines the servo rotation angle



Common angles and corresponding high-level pulse widths

Servo (180°)	High-level Pulse Width (us)
0°	500
45°	1000
90°	1500
135°	2000
180°	2500

- **Basic Timer**

Using the TimerA timer interrupt function on the MSPM0G3507 development board

PWM Servo (Schematic Name)	Control Pin	Function
S1	PA28	Simulated PWM output to control S1 servo
S2	PA7	Simulated PWM output to control S2

3. Project Experience

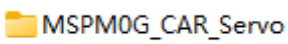
Use the project files we provide to directly experience the corresponding functions of the development board.

Later tutorials will not provide this content to avoid repetition. You can refer to 【3. Development Environment Setup and Usage: 5. Project Porting】 for operations

3.1 Opening the Project

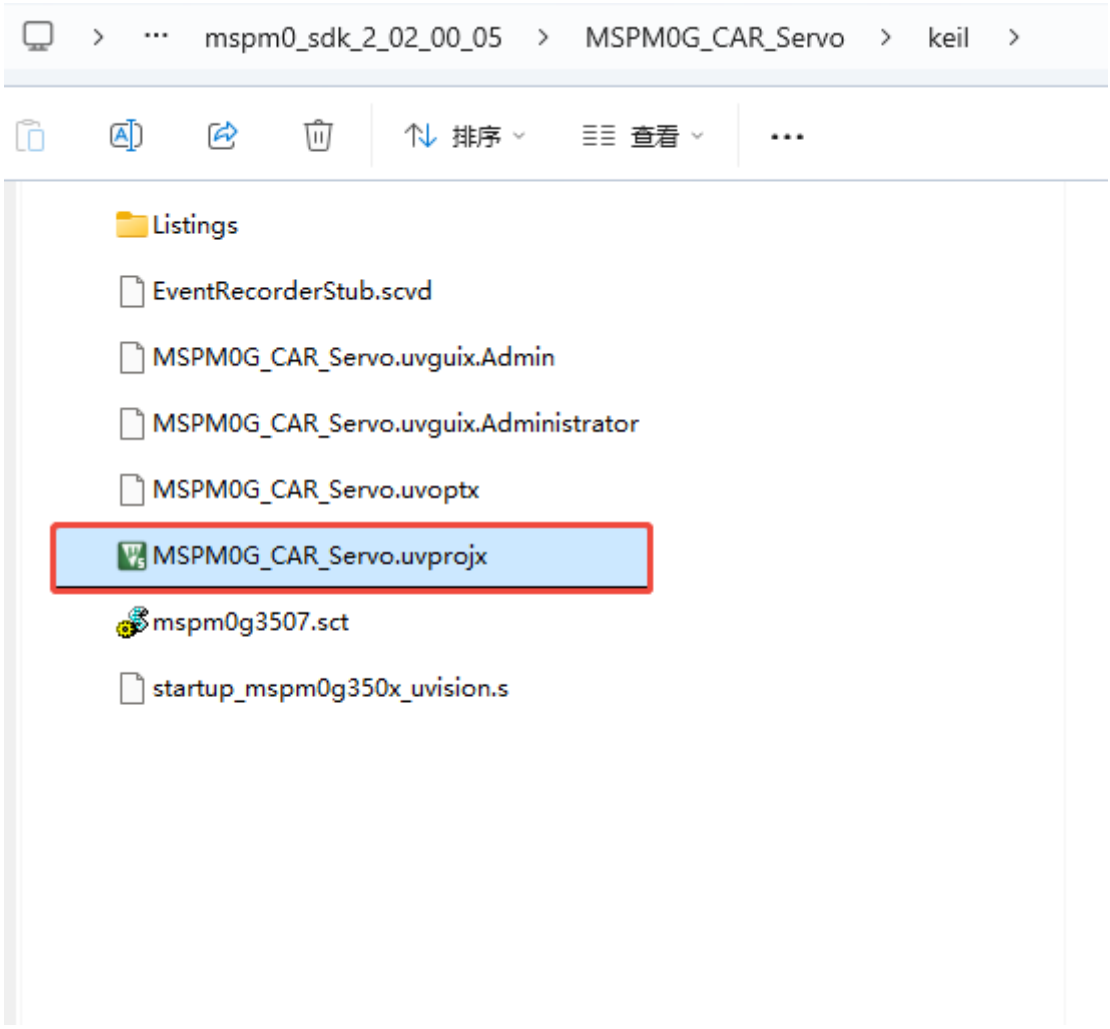
- **Project File Location**

  Project file path: Under the corresponding course folder in the 【Program Source Code Summary】 folder

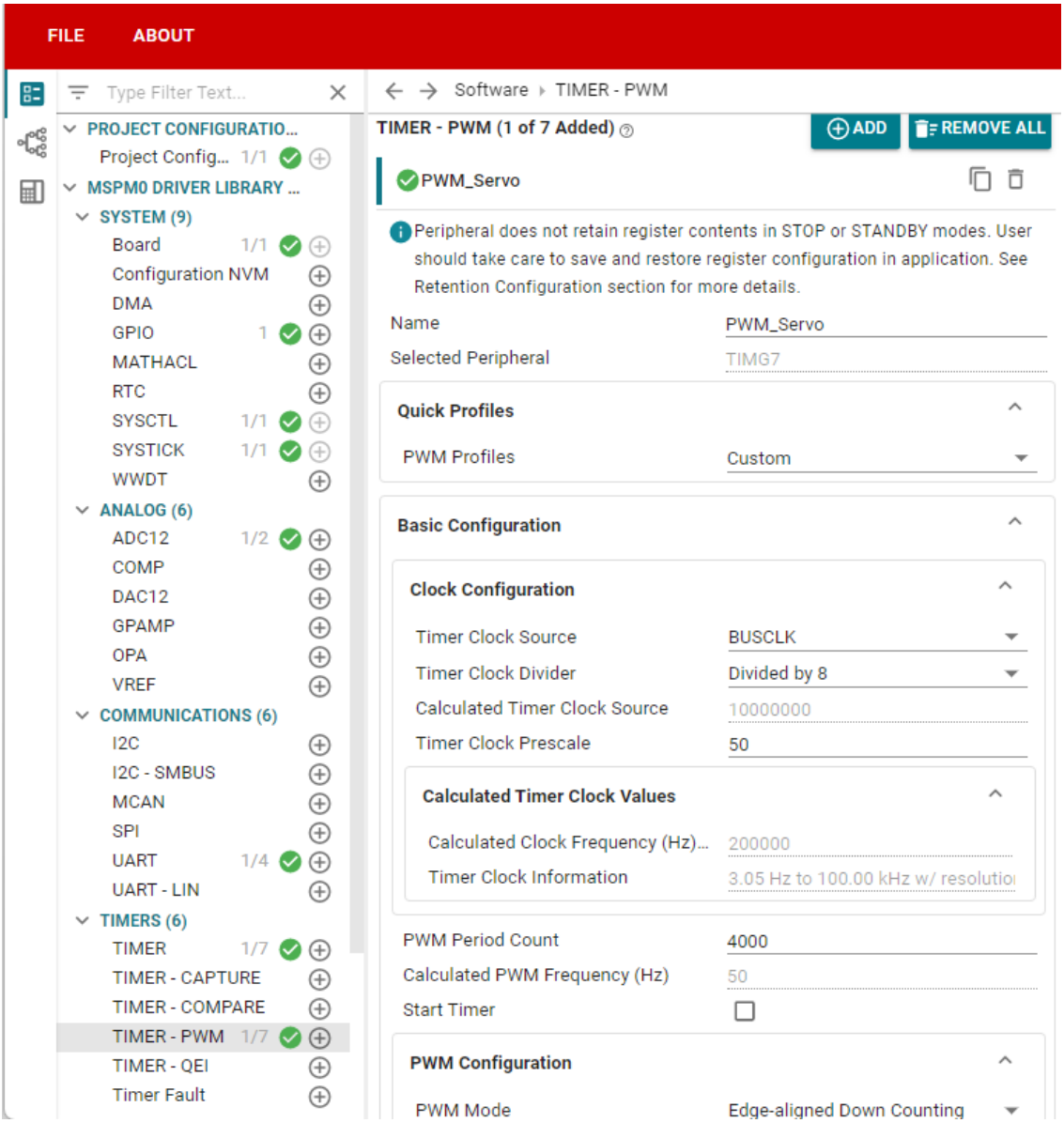


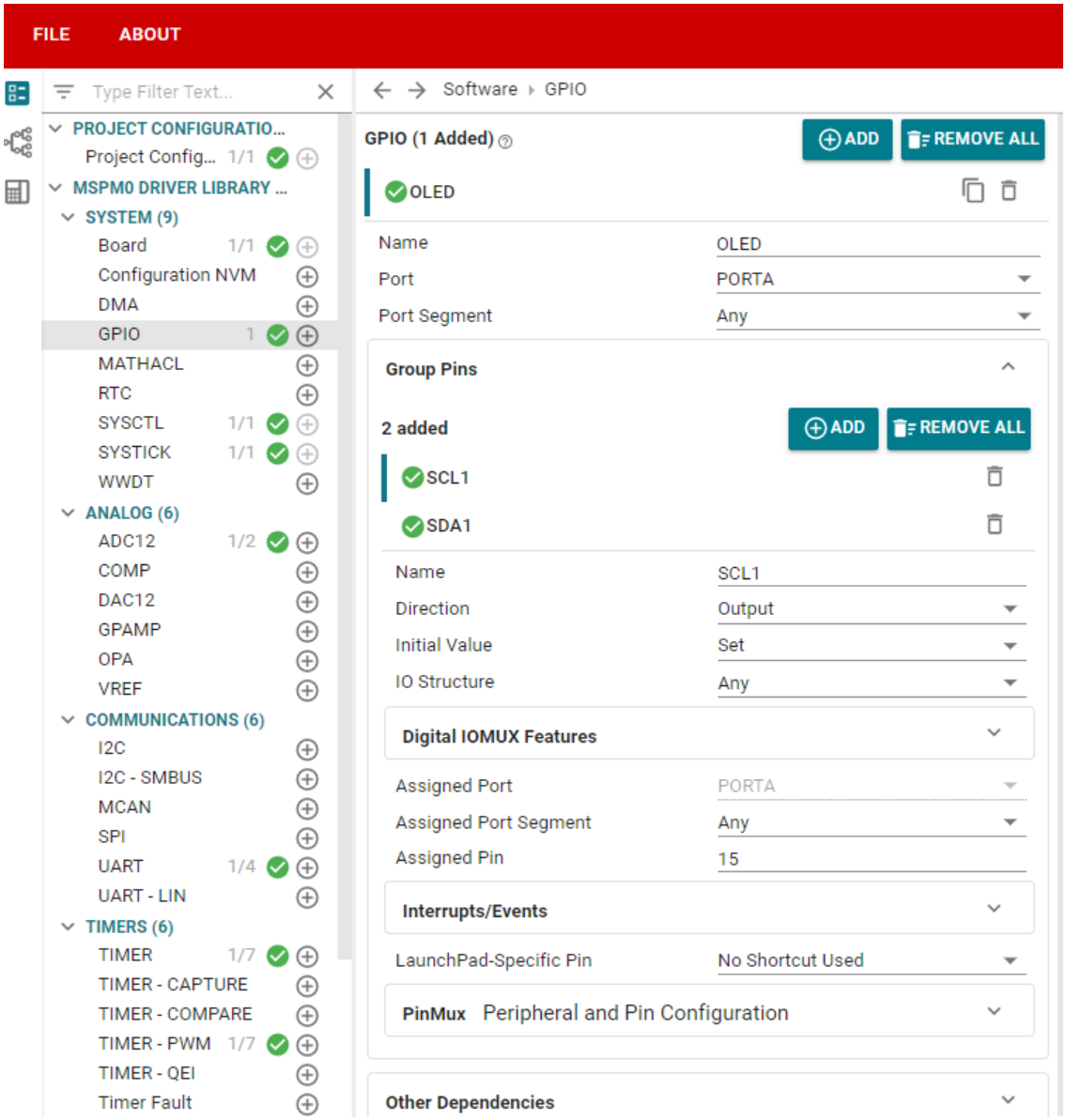
- **Opening the Project File**

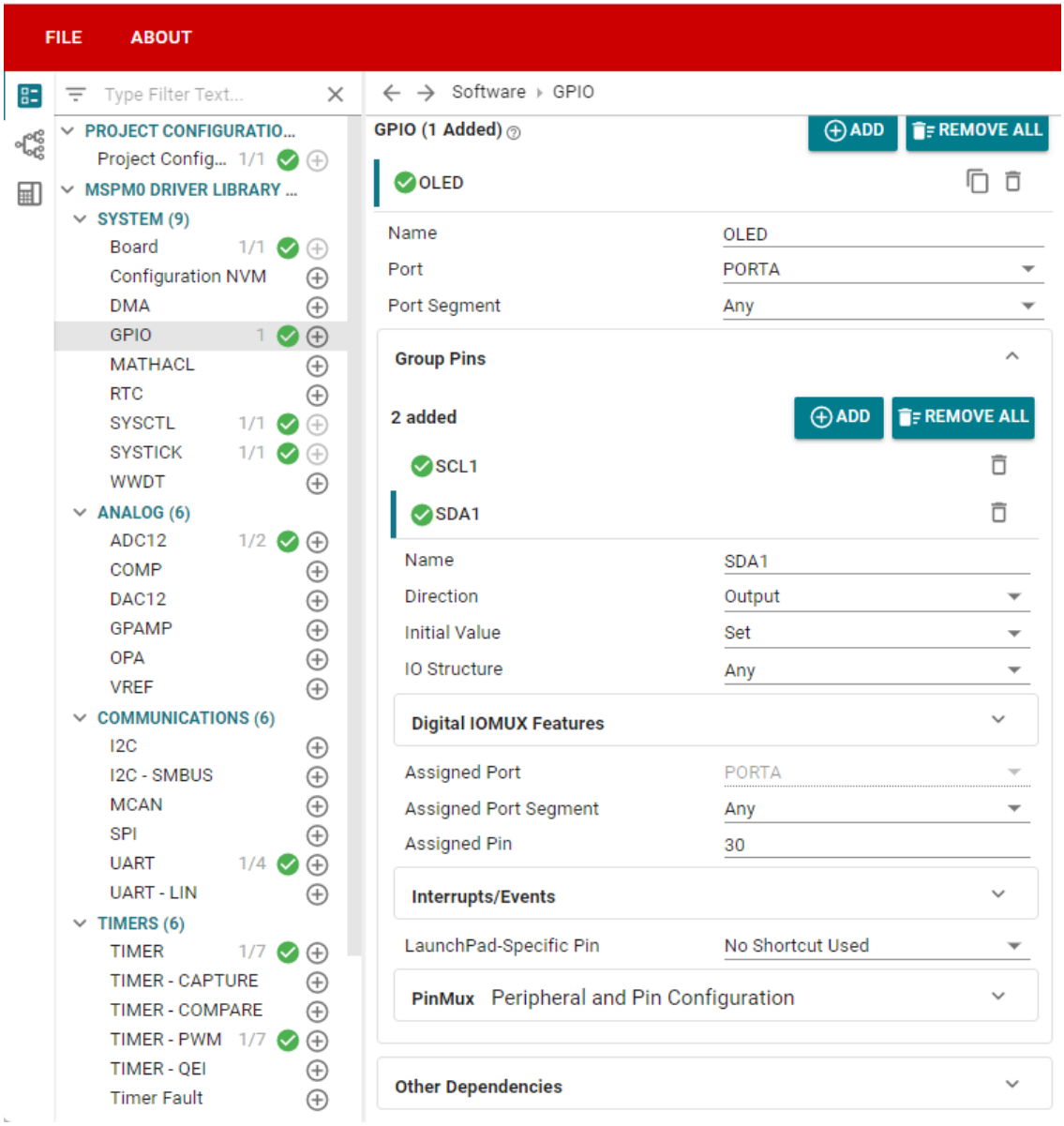
  Copy the project file to the **msp_sdk** directory under an **English path**, use KEIL5 to open the project file, and select the **.project** file when opening



- Pin Configuration


FILE
ABOUT
Type Filter Text...
PROJECT CONFIGURATIO...
Project Config... 1/1
MSPM0 DRIVER LIBRARY ...
SYSTEM (9)
Board 1/1
Configuration NVM
DMA
GPIO 1
MATHACL
RTC
SYSCTL 1/1
SYSTICK 1/1
WWDT
ANALOG (6)
ADC12 1/2
COMP
DAC12
GPAMP
OPA
VREF
COMMUNICATIONS (6)
I2C
I2C - SMBUS
MCAN
SPI
UART 1/4
UART - LIN
TIMERS (6)
TIMER 1/7
TIMER - CAPTURE
TIMER - COMPARE
TIMER - PWM 1/7
TIMER - QEI
Timer Fault
Software
TIMER - PWM
TIMER - PWM (1 of 7 Added)
ADD
REMOVE ALL
PWM_Servo
Peripheral does not retain register contents in STOP or STANDBY modes. User should take care to save and restore register configuration in application. See Retention Configuration section for more details.
Name
PWM_Servo
Selected Peripheral
TIMG7
Quick Profiles
PWM Profiles
Custom
Basic Configuration
Clock Configuration
Timer Clock Source
BUSCLK
Timer Clock Divider
Divided by 8
Calculated Timer Clock Source
10000000
Timer Clock Prescale
50
Calculated Timer Clock Values
Calculated Clock Frequency (Hz)...
200000
Timer Clock Information
3.05 Hz to 100.00 kHz w/ resolutio
PWM Period Count
4000
Calculated PWM Frequency (Hz)
50
Start Timer
PWM Configuration
PWM Mode
Edge-aligned Down Counting


FILE
ABOUT
Type Filter Text...
PROJECT CONFIGURATIO...
Project Config... 1/1
MSPM0 DRIVER LIBRARY ...
SYSTEM (9)
Board 1/1
Configuration NVM
DMA
GPIO 1
MATHACL
RTC
SYSCTL 1/1
SYSTICK 1/1
WWDT
ANALOG (6)
ADC12 1/2
COMP
DAC12
GPAMP
OPA
VREF
COMMUNICATIONS (6)
I2C
I2C - SMBUS
MCAN
SPI
UART 1/4
UART - LIN
TIMERS (6)
TIMER 1/7
TIMER - CAPTURE
TIMER - COMPARE
TIMER - PWM 1/7
TIMER - QEI
Timer Fault
Software ▸ GPIO
GPIO (1 Added)
ADD
REMOVE ALL
OLED
Name
OLED
Port
PORTA
Port Segment
Any
Group Pins
2 added
ADD
REMOVE ALL
SCL1
SDA1
Name
SCL1
Direction
Output
Initial Value
Set
IO Structure
Any
Digital IOMUX Features
Assigned Port
PORTA
Assigned Port Segment
Any
Assigned Pin
15
Interrupts/Events
LaunchPad-Specific Pin
No Shortcut Used
PinMux
Peripheral and Pin Configuration
Other Dependencies



4. Main Functions

Mainly introduces the functional code written by users. **For detailed code, you can open the project files we provide and view the source code in the Bsp folder**.

4.1 User Functions

> **Function: servo_init**

Function Prototype	**void servo_init(void)**
Function Description	Initialize timer simulated PWM signal output
Input Parameters	None
Output Parameters	None

> **Function: Set_Servo270_Angle**

Function Prototype	void Set_Servo270_Angle(unsigned int angle)
Function Description	Control 270-degree servo rotation
Input Parameters	**angle**: rotation angle
Output Parameters	None

> **Function: Set_Servo_Angle**

Function Prototype	void Set_Servo_Angle(unsigned int angle)
Function Description	Control 180-degree servo rotation
Input Parameters	**angle**: rotation angle
Output Parameters	None

> **Function: get_servo**

Function Prototype	void get_servo ()
Function Description	Control servo to rotate from 0° then sequentially 0°,90°,180°,90°,0° back and forth
Input Parameters	None
Output Parameters	None

5. Experimental Phenomenon

After successfully downloading the program, press the RESET button on the development board and observe the phenomenon!

```
For program download, refer to 【3. Development Environment Setup and Usage: 3. Uniflash burning】
```

Phenomenon:

S1 Interface Servo: Rotates cyclically from 0°→ 90° →180° → 90° → 0°.

```
The provided project only controls the S1 interface servo. To control other servos, you can modify the interface function parameters yourself
```