Ultrasonic Servo PTZ Basic Use

This tutorial demonstrates: how to externally connect and use an ultrasonic servo PTZ on the expansion board and then print ultrasonic ranging data via serial port.

1. Software and Hardware

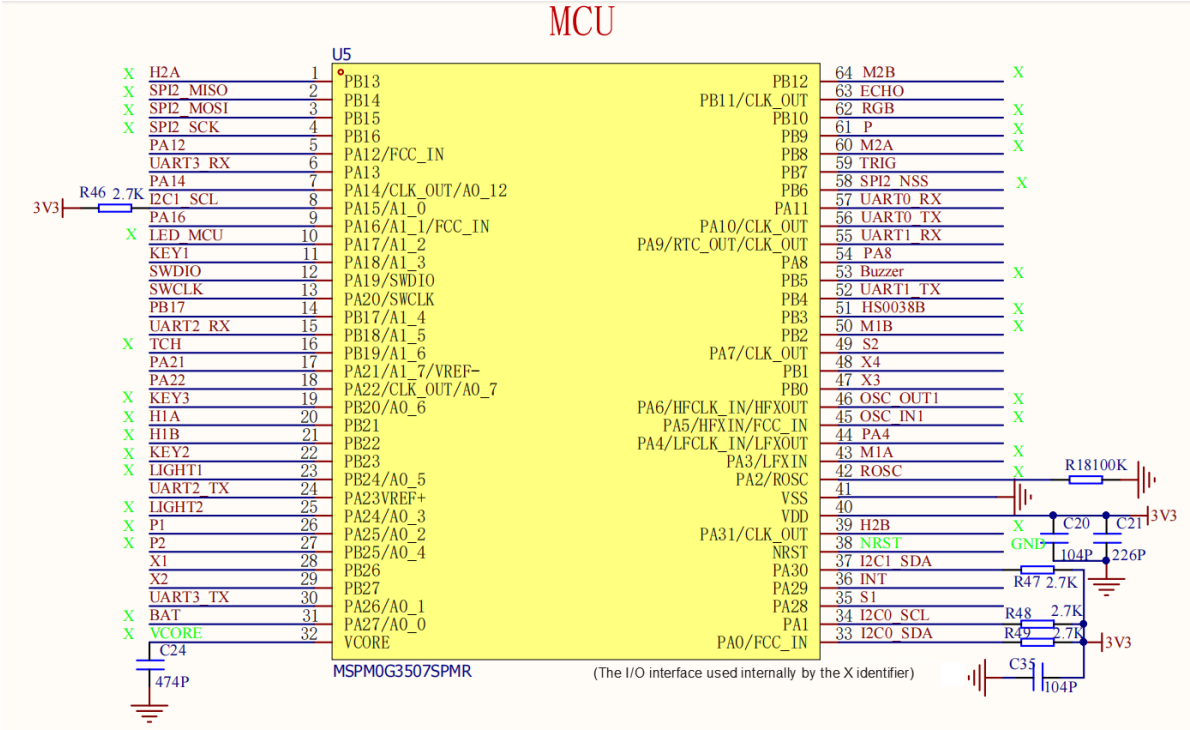
- **KEIL5**
- **MSPM0G3507 Expansion Board**
- **Ultrasonic Servo PTZ**
- **Type-C data cable or DAP-Link**

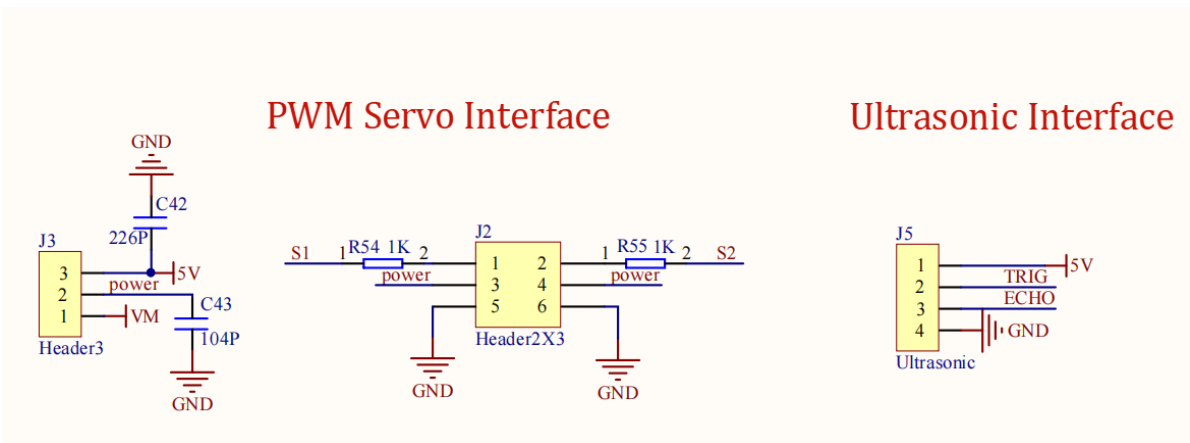
  For programming download or simulation to the development board
- **Serial Port Assistant**

  Receive and print serial port data

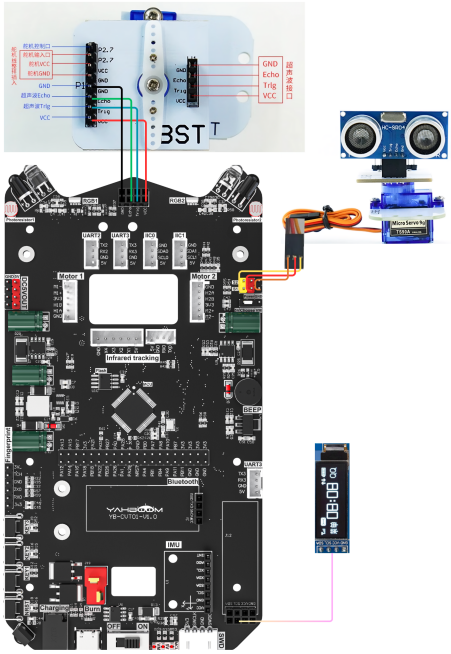
2. Basic Principles

2.1 Hardware Schematic





2.2 Physical Connection Diagram



Servo (S1)	Car	Specific Meaning
SIG(Yellow)	PA28(Yellow)	Signal Line
VCC(Red)	VCC(Red)	Power Supply
GND(Brown)	GND(Black)	Ground Line

Servo (S2)	Car	Specific Meaning
SIG(Yellow)	PA7(Yellow)	Signal Line
VCC(Red)	VCC(Red)	Power Supply
GND(Brown)	GND(Black)	Ground Line

Ultrasonic Module:

Note: Using the servo PTZ requires adding jumper caps to connect 5V and choose, see the white line.

2.3 Control Principle

(Schematic Name)	Control Pin	Specific Meaning
TRIG	PB7	Trigger End
ECHO	PB11	Receiver End
S1	PA28	Servo Control Pin

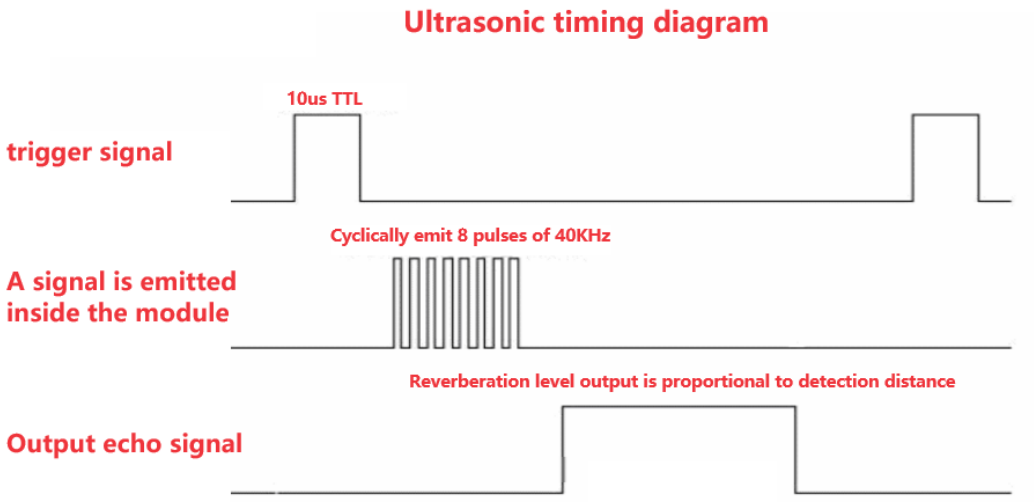
Ultrasonic Module:

It is a module that uses ultrasonic waves for non-contact physical quantity measurement. It can accurately measure physical quantities such as distance, speed, and flow rate by emitting and receiving ultrasonic signals, and convert the measurement results into digital signal outputs. This article will explain the working principle and function of the ultrasonic module.

This experiment's ultrasonic module information:

Model	HC-SR04	Detection Distance	2-400cm
Working Voltage	5V	High Precision	Up to 0.3cm
Working Current	15mA	Blind Zone	2cm
Working Frequency	40KHz	Pin Wiring Sequence	VCC, Trig (Control End), Echo (Receiver End), GND
Static Working Current	<2mA	Input Trigger Signal	10uS TTL pulse
Detection Angle	Not more than 15°	Input Echo Signal	Output TTL level signal, proportional to range
Range Range	2cm-4m (Peak)	Level Output	TTL level

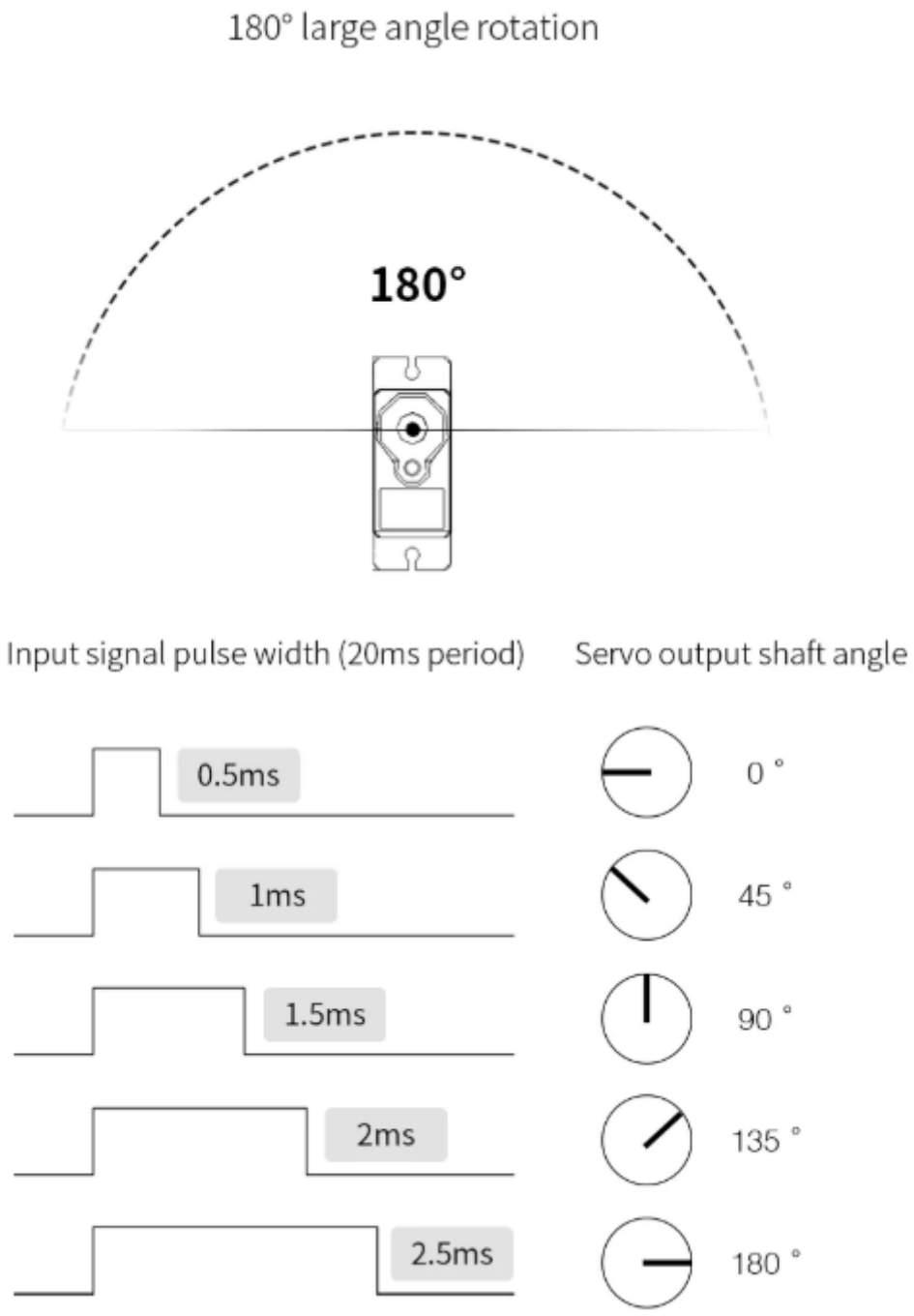
Ranging Principle: Input a high potential of more than 10 microseconds at the trigger end of the ultrasonic module to emit ultrasonic waves. After emitting the ultrasonic waves and before receiving the returned ultrasonic waves, the receiver end is at high potential. Therefore, the program can calculate the distance of the measured object from the high-level pulse duration of the "response" pin. **Test Distance = (High Level Time * Sound Speed (340M/S)) / 2;**



Note: The above timing diagram shows that you only need to provide a pulse trigger signal of more than 10us. The module will internally emit 8 40kHz cycle levels and detect echoes. Once an echo signal is detected, it will output an echo signal. The pulse width of the echo signal is proportional to the measured distance. Therefore, the distance can be calculated from the time interval between transmitting the signal and receiving the echo signal.

PWM Control Servo Principle

Generate PWM signals using the timer/counter (TIM) peripheral and control the servo angle by adjusting the duty cycle.



3. Project Experience

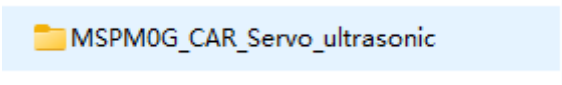
Use the project files we provide to directly experience the corresponding functions of the development board.

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

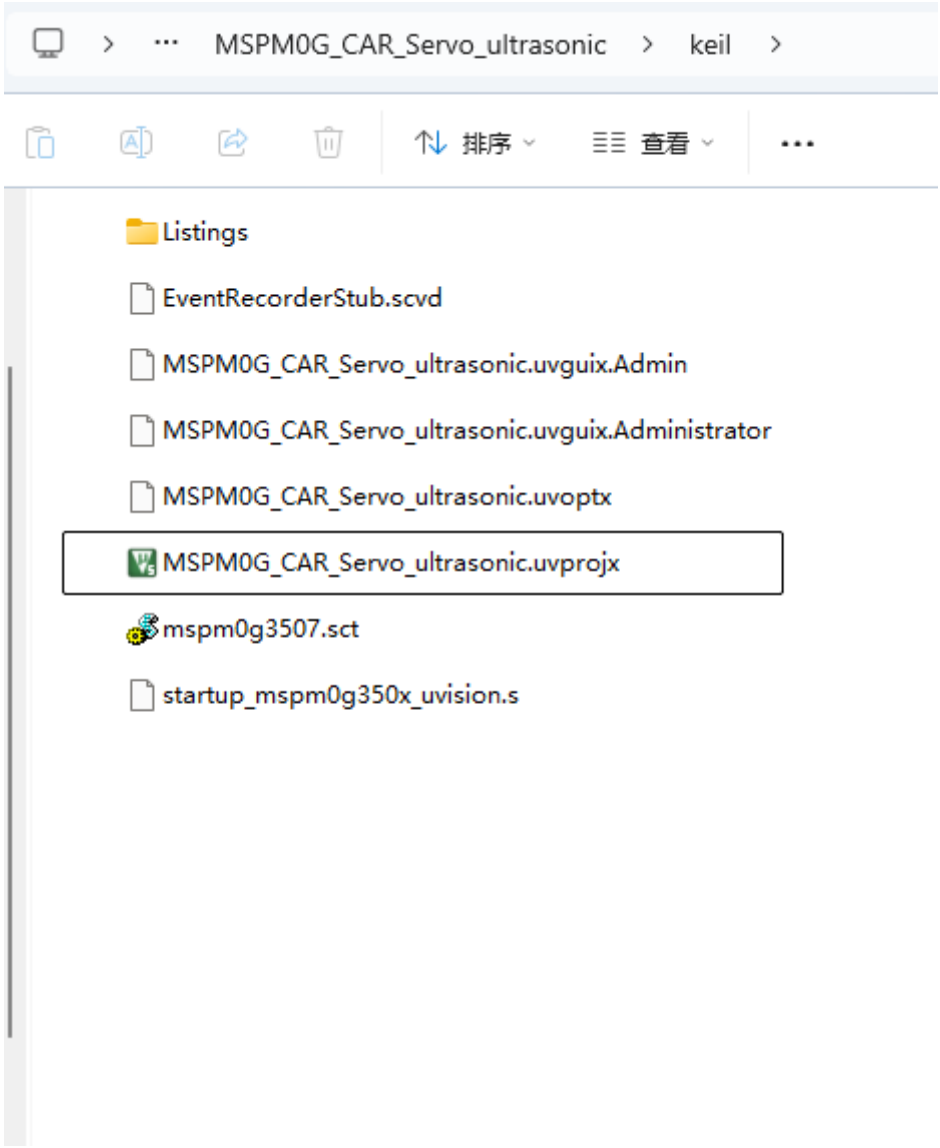
- **Project File Location**

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

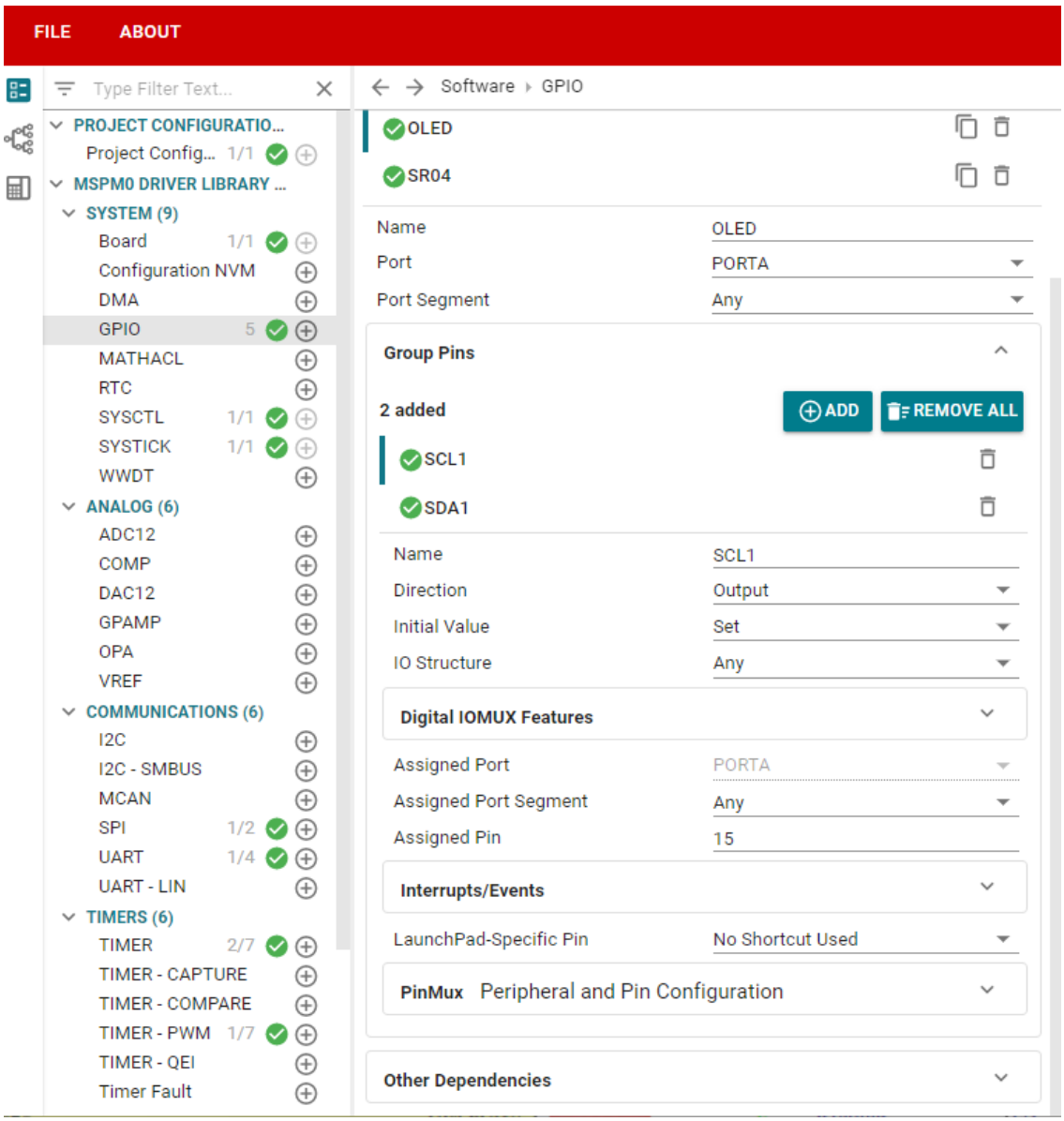


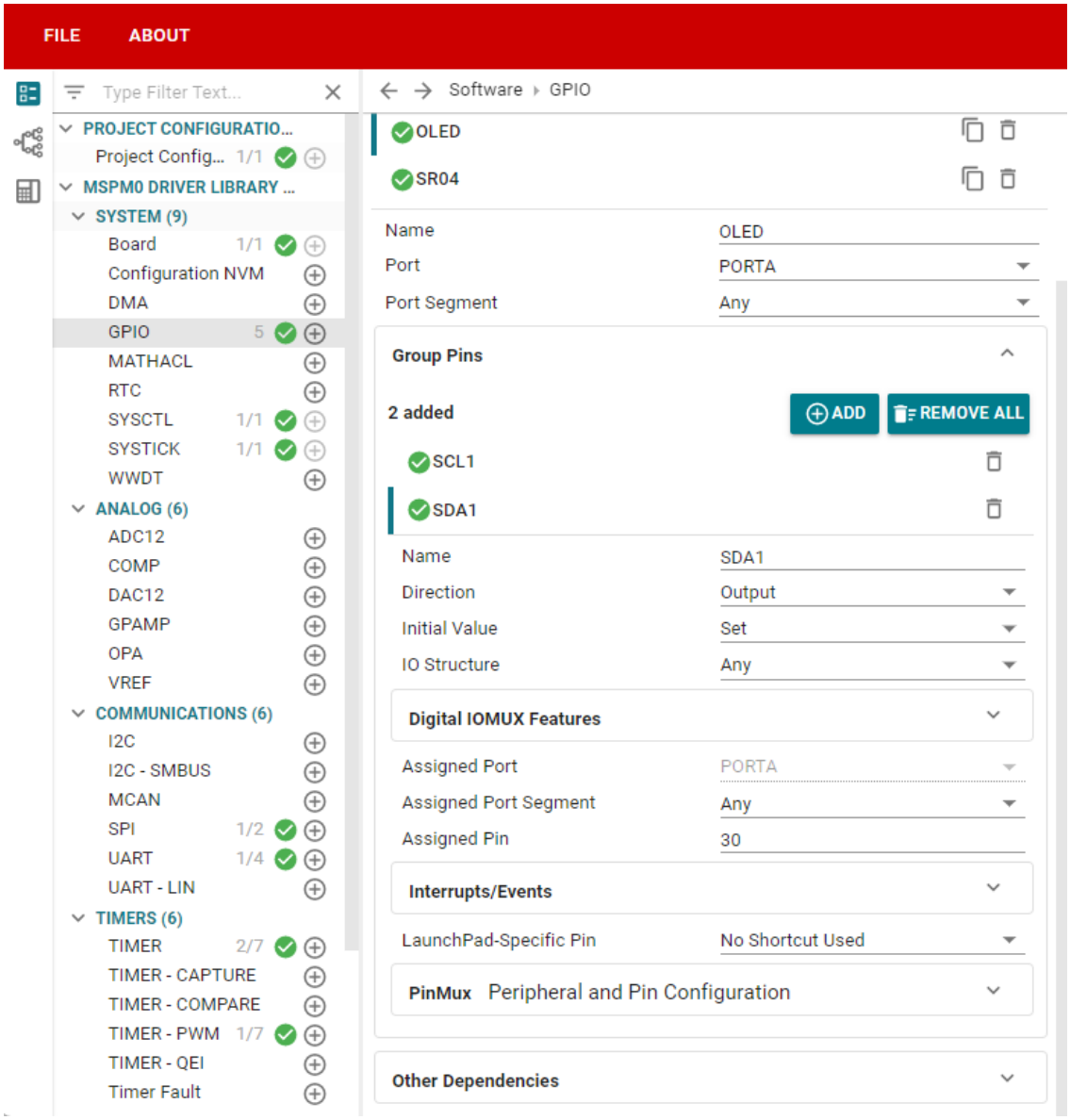
- **Opening the Project File**

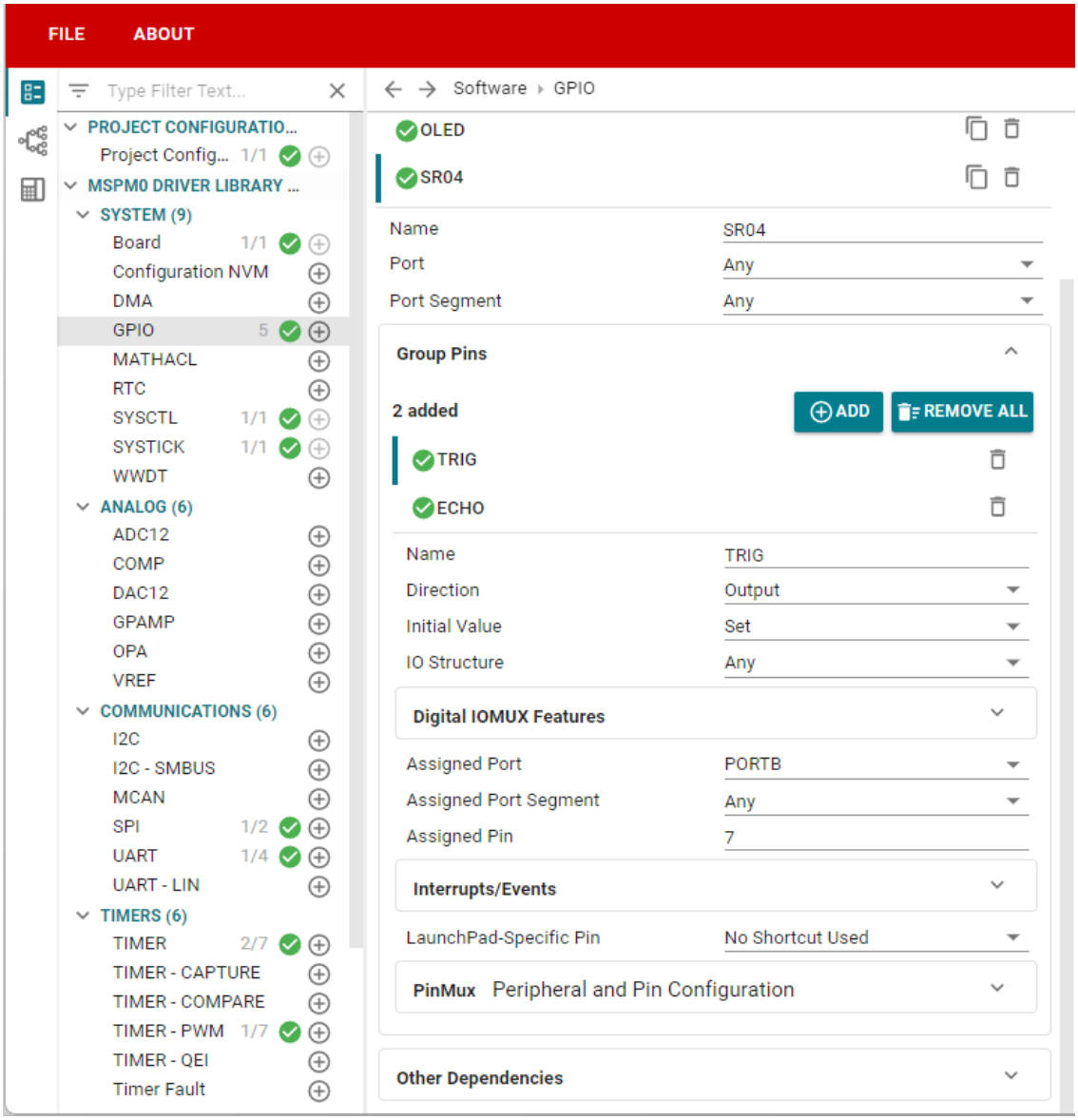
  Copy the project file to the **msp_sdk** directory under an **English path**, use KEIL5 to open the project file, and select the **.project** file when opening

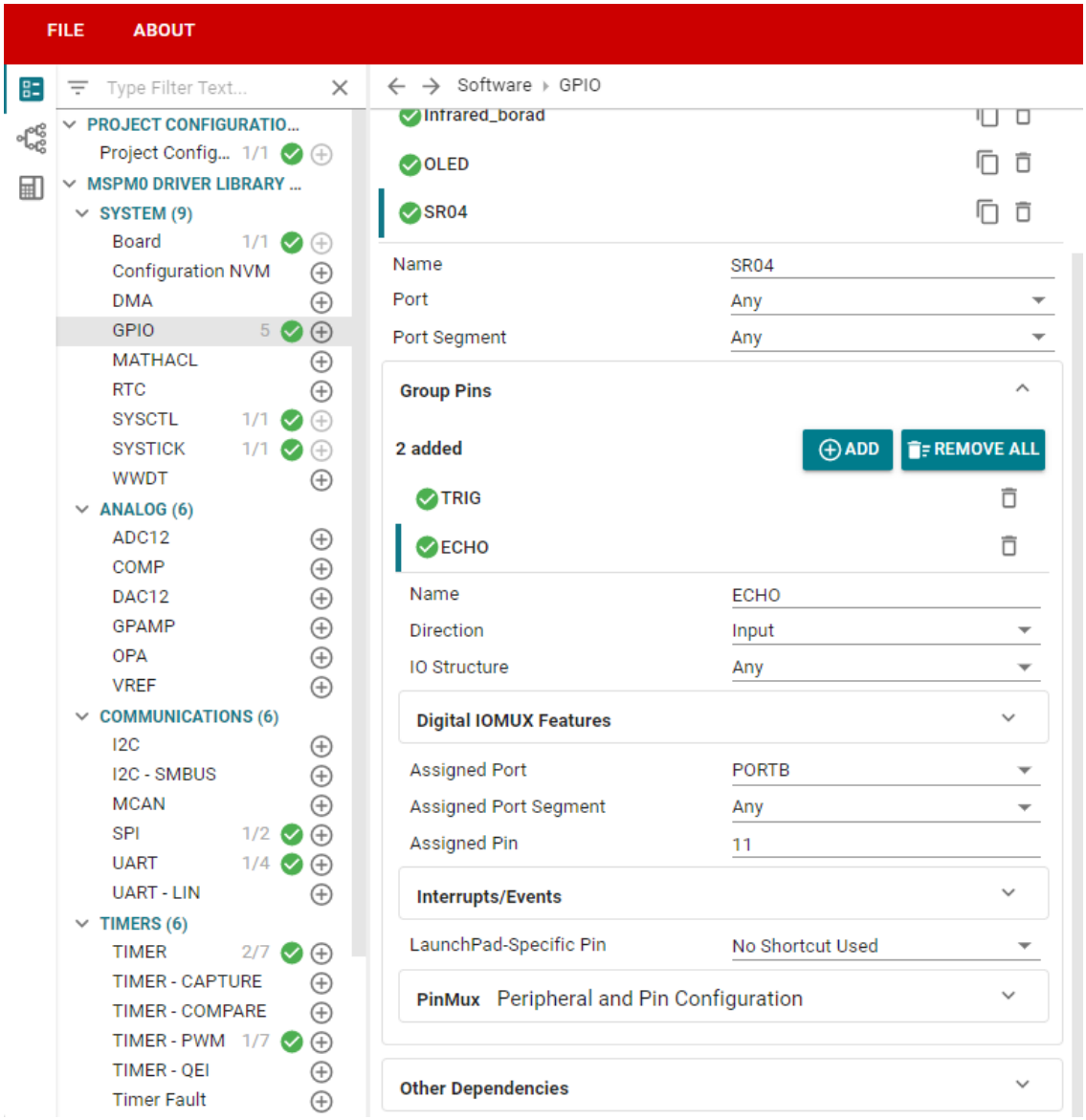


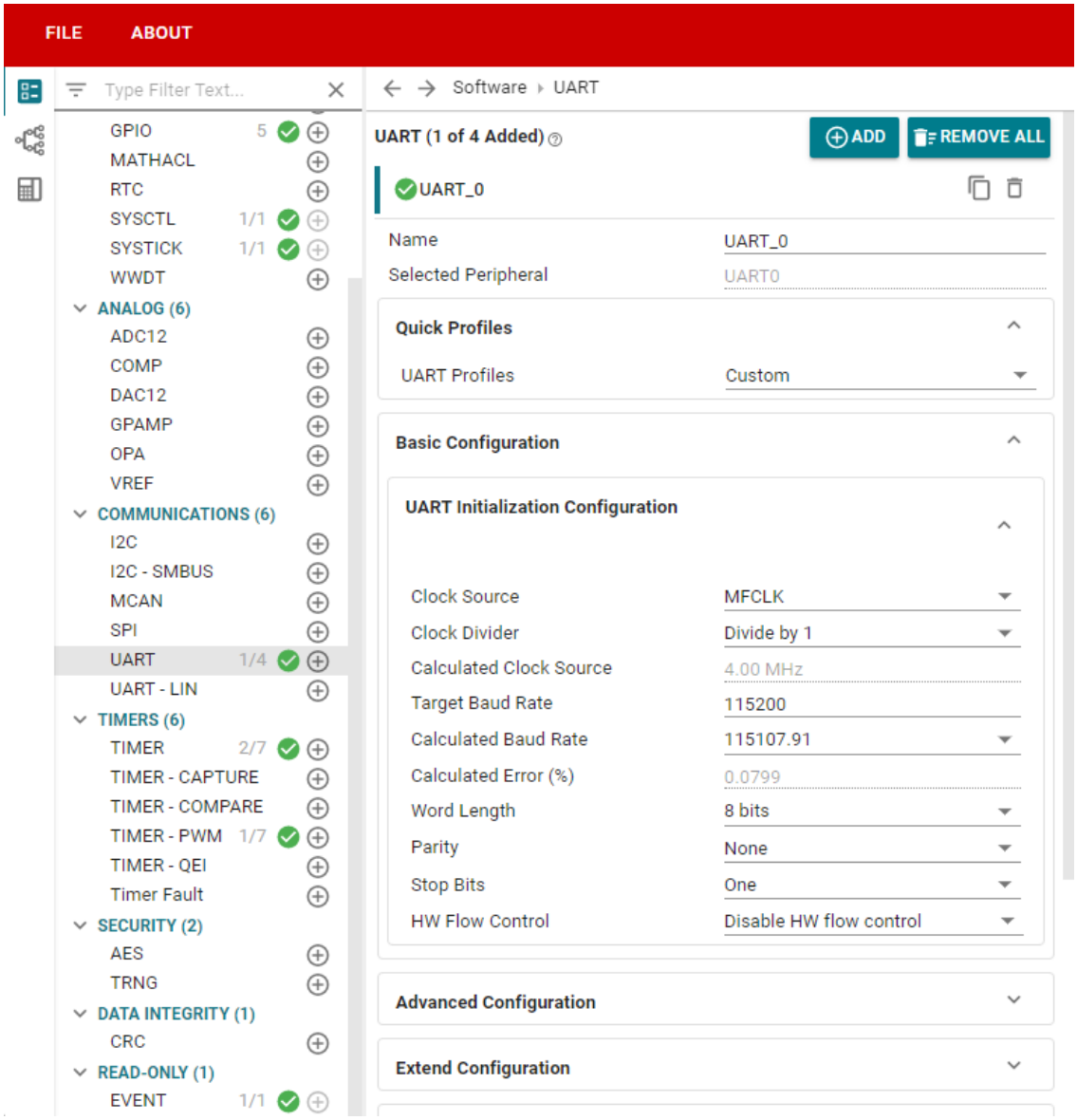
- Pin Configuration

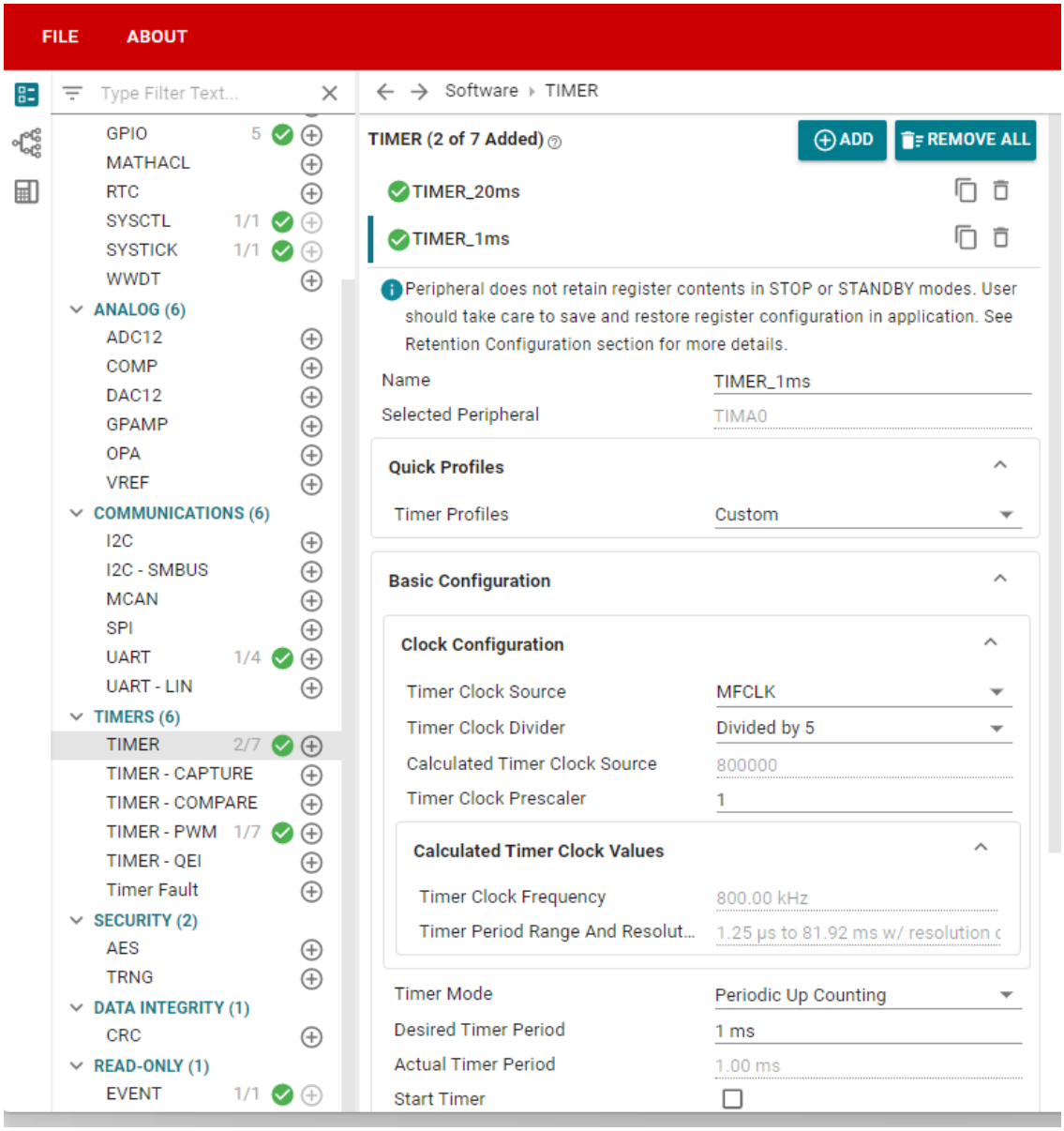

FILE
ABOUT
Type Filter Text...
PROJECT CONFIGURATIO...
Project Config... 1/1
MSPM0 DRIVER LIBRARY ...
SYSTEM (9)
Board 1/1
Configuration NVM
DMA
GPIO 5
MATHACL
RTC
SYSCTL 1/1
SYSTICK 1/1
WWDT
ANALOG (6)
ADC12
COMP
DAC12
GPAMP
OPA
VREF
COMMUNICATIONS (6)
I2C
I2C - SMBUS
MCAN
SPI 1/2
UART 1/4
UART - LIN
TIMERS (6)
TIMER 2/7
TIMER - CAPTURE
TIMER - COMPARE
TIMER - PWM 1/7
TIMER - QEI
Timer Fault
Software ▸ GPIO
OLED
SR04
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

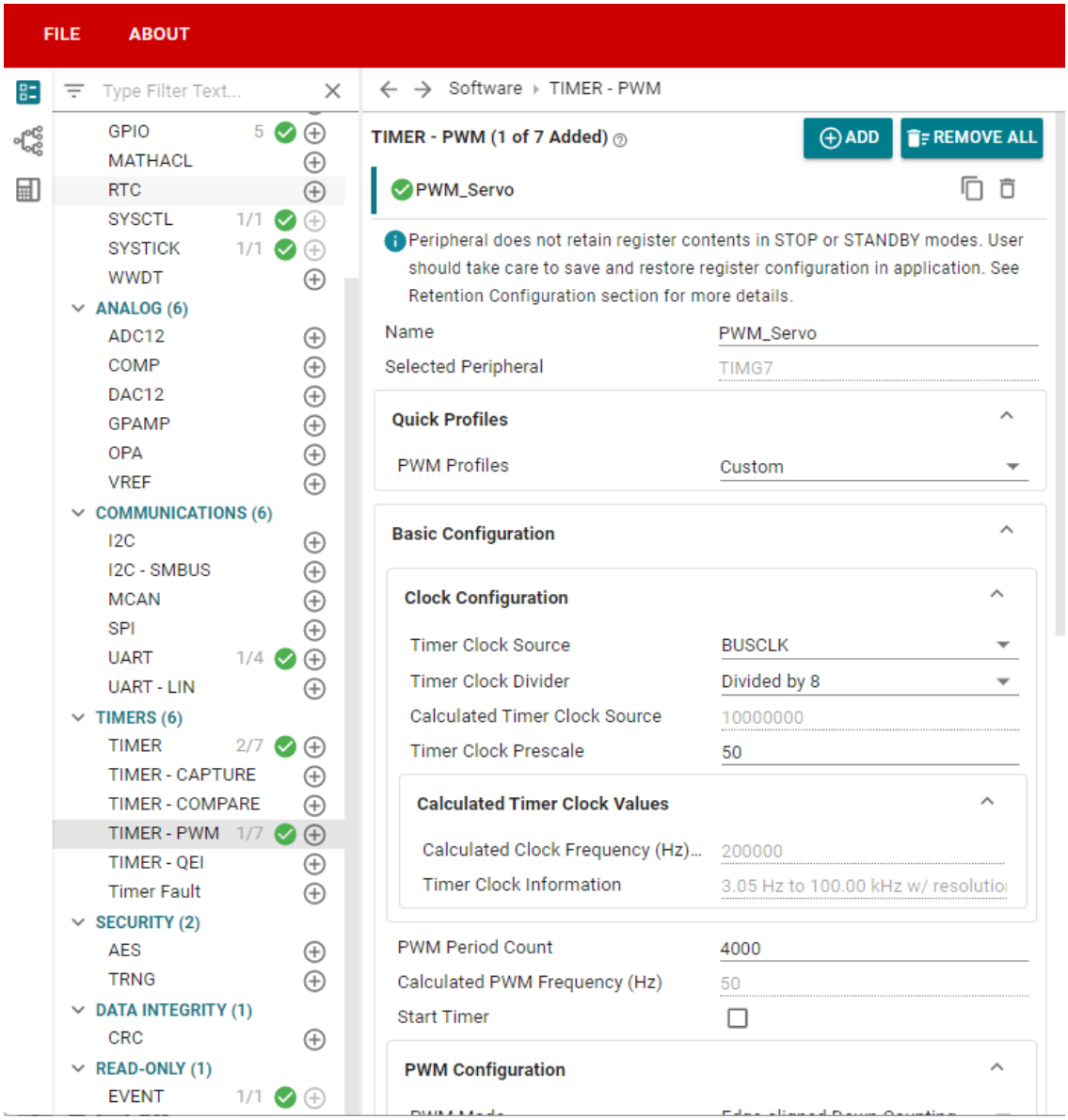

FILE
ABOUT
Type Filter Text...
PROJECT CONFIGURATIO...
Project Config... 1/1
MSPM0 DRIVER LIBRARY ...
SYSTEM (9)
Board 1/1
Configuration NVM
DMA
GPIO 5
MATHACL
RTC
SYSCTL 1/1
SYSTICK 1/1
WWDT
ANALOG (6)
ADC12
COMP
DAC12
GPAMP
OPA
VREF
COMMUNICATIONS (6)
I2C
I2C - SMBUS
MCAN
SPI 1/2
UART 1/4
UART - LIN
TIMERS (6)
TIMER 2/7
TIMER - CAPTURE
TIMER - COMPARE
TIMER - PWM 1/7
TIMER - QEI
Timer Fault
Software ▸ GPIO
OLED
SR04
Name OLED
Port PORTA
Port Segment Any
Group Pins
2 added
ADD
REMOVE ALL
SCL1
SDA1
Name SDA1
Direction Output
Initial Value Set
IO Structure Any
Digital IOMUX Features
Assigned Port PORTA
Assigned Port Segment Any
Assigned Pin 30
Interrupts/Events
LaunchPad-Specific Pin No Shortcut Used
PinMux Peripheral and Pin Configuration
Other Dependencies


FILE
ABOUT
Type Filter Text...
PROJECT CONFIGURATIO...
Project Config... 1/1
MSPM0 DRIVER LIBRARY ...
SYSTEM (9)
Board 1/1
Configuration NVM
DMA
GPIO 5
MATHACL
RTC
SYSCTL 1/1
SYSTICK 1/1
WWDT
ANALOG (6)
ADC12
COMP
DAC12
GPAMP
OPA
VREF
COMMUNICATIONS (6)
I2C
I2C - SMBUS
MCAN
SPI 1/2
UART 1/4
UART - LIN
TIMERS (6)
TIMER 2/7
TIMER - CAPTURE
TIMER - COMPARE
TIMER - PWM 1/7
TIMER - QEI
Timer Fault
Software ▸ GPIO
OLED
SR04
Name SR04
Port Any
Port Segment Any
Group Pins
2 added
ADD
REMOVE ALL
TRIG
ECHO
Name TRIG
Direction Output
Initial Value Set
IO Structure Any
Digital IOMUX Features
Assigned Port PORTB
Assigned Port Segment Any
Assigned Pin 7
Interrupts/Events
LaunchPad-Specific Pin No Shortcut Used
PinMux Peripheral and Pin Configuration
Other Dependencies


FILE
ABOUT
Type Filter Text...
PROJECT CONFIGURATIO...
Project Config... 1/1
MSPM0 DRIVER LIBRARY ...
SYSTEM (9)
Board 1/1
Configuration NVM
DMA
GPIO 5
MATHACL
RTC
SYSCTL 1/1
SYSTICK 1/1
WWDT
ANALOG (6)
ADC12
COMP
DAC12
GPAMP
OPA
VREF
COMMUNICATIONS (6)
I2C
I2C - SMBUS
MCAN
SPI 1/2
UART 1/4
UART - LIN
TIMERS (6)
TIMER 2/7
TIMER - CAPTURE
TIMER - COMPARE
TIMER - PWM 1/7
TIMER - QEI
Timer Fault
Software ▸ GPIO
Infrared_borad
OLED
SR04
Name SR04
Port Any
Port Segment Any
Group Pins
2 added
ADD
REMOVE ALL
TRIG
ECHO
Name ECHO
Direction Input
IO Structure Any
Digital IOMUX Features
Assigned Port PORTB
Assigned Port Segment Any
Assigned Pin 11
Interrupts/Events
LaunchPad-Specific Pin No Shortcut Used
PinMux Peripheral and Pin Configuration
Other Dependencies


FILE
ABOUT
Type Filter Text...
GPIO 5
MATHACL
RTC
SYSCTL 1/1
SYSTICK 1/1
WWDT
ANALOG (6)
ADC12
COMP
DAC12
GPAMP
OPA
VREF
COMMUNICATIONS (6)
I2C
I2C - SMBUS
MCAN
SPI
UART 1/4
UART - LIN
TIMERS (6)
TIMER 2/7
TIMER - CAPTURE
TIMER - COMPARE
TIMER - PWM 1/7
TIMER - QEI
Timer Fault
SECURITY (2)
AES
TRNG
DATA INTEGRITY (1)
CRC
READ-ONLY (1)
EVENT 1/1
Software ▸ UART
UART (1 of 4 Added)
ADD
REMOVE ALL
UART_0
Name
UART_0
Selected Peripheral
UART0
Quick Profiles
UART Profiles
Custom
Basic Configuration
UART Initialization Configuration
Clock Source
MFCLK
Clock Divider
Divide by 1
Calculated Clock Source
4.00 MHz
Target Baud Rate
115200
Calculated Baud Rate
115107.91
Calculated Error (%)
0.0799
Word Length
8 bits
Parity
None
Stop Bits
One
HW Flow Control
Disable HW flow control
Advanced Configuration
Extend Configuration


FILE
ABOUT
Type Filter Text...
GPIO 5
MATHACL
RTC
SYSCTL 1/1
SYSTICK 1/1
WWDT
ANALOG (6)
ADC12
COMP
DAC12
GPAMP
OPA
VREF
COMMUNICATIONS (6)
I2C
I2C - SMBUS
MCAN
SPI
UART 1/4
UART - LIN
TIMERS (6)
TIMER 2/7
TIMER - CAPTURE
TIMER - COMPARE
TIMER - PWM 1/7
TIMER - QEI
Timer Fault
SECURITY (2)
AES
TRNG
DATA INTEGRITY (1)
CRC
READ-ONLY (1)
EVENT 1/1
Software ▸ TIMER
TIMER (2 of 7 Added)
ADD
REMOVE ALL
TIMER_20ms
TIMER_1ms
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
TIMER_1ms
Selected Peripheral
TIMA0
Quick Profiles
Timer Profiles
Custom
Basic Configuration
Clock Configuration
Timer Clock Source
MFCLK
Timer Clock Divider
Divided by 5
Calculated Timer Clock Source
800000
Timer Clock Prescaler
1
Calculated Timer Clock Values
Timer Clock Frequency
800.00 kHz
Timer Period Range And Resolut...
1.25 µs to 81.92 ms w/ resolution c
Timer Mode
Periodic Up Counting
Desired Timer Period
1 ms
Actual Timer Period
1.00 ms
Start Timer



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: Hcsr04GetLength**

Function Prototype	float Hcsr04GetLength (void)
Function Description	Calculate and obtain ultrasonic data
Input Parameters	None
Output Parameters	Ultrasonic data

> **Function: get_servo_Printf**

Function Prototype	void get_servo_Printf ()
Function Description	Servo rotates from 90° to 0°, then to 90° and 180°, during which ultrasonic data is obtained and respectively printf output and displayed on OLED
Input Parameters	None
Output Parameters	None

5. Experimental Phenomenon

During installation, you need to center the servo first before installation; the tutorial folder provides a 【MSPM0G_Car_Servo_ultrasonic.hex】 file for calibration. After burning this program, the servo will rotate to 90°. At this time, install the connection board between the ultrasonic and servo PTZ (physical object as shown below) on the servo. When installing the ultrasonic, it needs to face forward (slight deviation is acceptable).

After servo calibration, we can burn the code for this chapter. Then after powering on the car, press the Reset key. We need to connect to the host computer through the Type-C interface, open the serial port assistant, set the parameters as shown in the figure below, and then we can view the measured ultrasonic distance through the serial port assistant.

After the car is powered on, the PTZ will first turn to the middle, then turn left, and then turn right. It cycles back and forth in these three directions. The serial port will print the distance between the ultrasonic measurement and obstacles.

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

Servo wiring is as follows:


BEEP
R100
REMOVE SEAL AFTER WASHING

Effect is as follows: