RGB Light Drive

This tutorial demonstrates: controlling the onboard RGB through buttons.

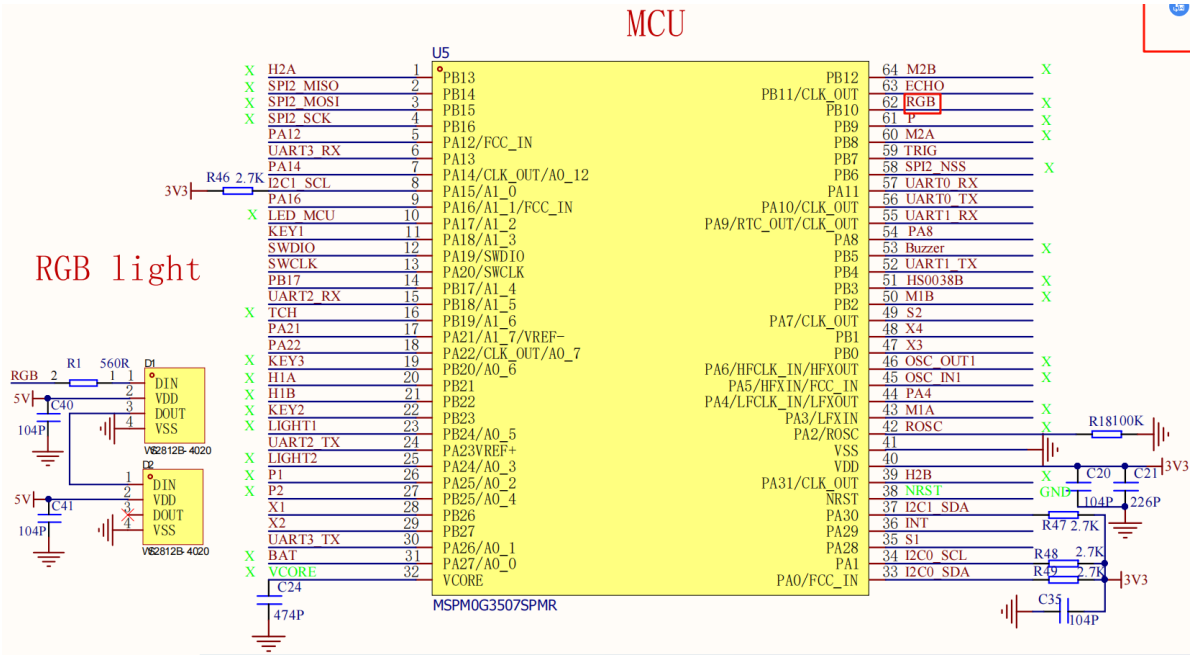
1. Software and Hardware

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

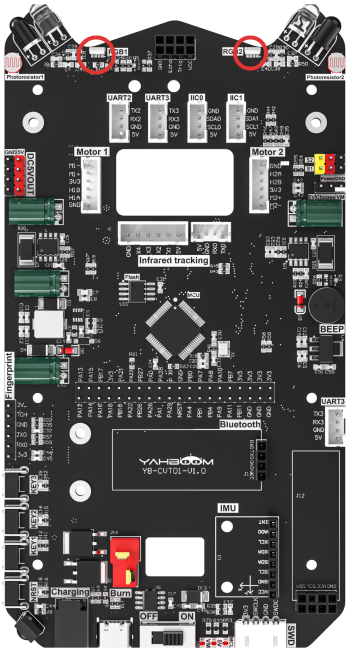
  For programming download or simulation to the development board

2. Basic Principles

2.1 Hardware Schematic



2.2 Physical Connection Diagram



2.3 Control Principle

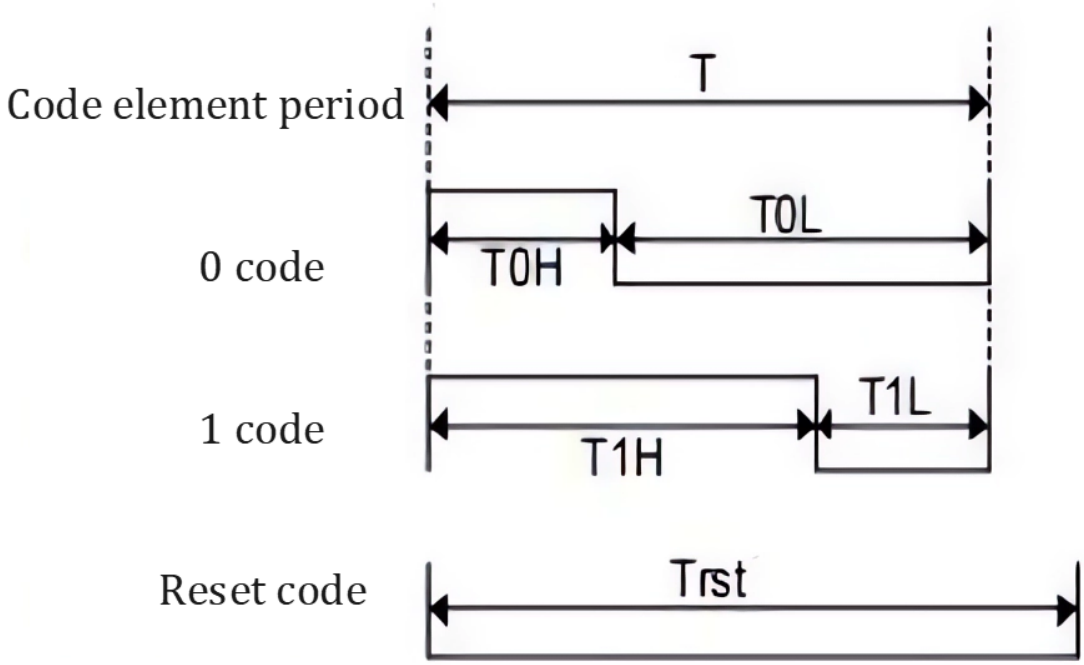
(Schematic Name)	Control Pin	Specific Meaning
RGB	PB10	Control data signal input pin

Chip Model: WS2812

Communication Protocol:

XGB-2812 is a single-wire transmission three-channel LED drive control chip that uses unipolar return-to-zero code data protocol. Each symbol must have a low level. Each symbol of this protocol starts with a high level, and the high-level time width determines "0" code or "1".

Input code pattern:



Symbol	Parameter	Min Value	Typical Value	Max Value	Unit
T	Symbol period	1200	-	-	ns
T0H	0 code, high level time	200	300	400	ns
T0L	0 code, low level time	800	900	-	ns
T1H	1 code, high level time	650	900	1000	ns
T1L	1 code, low level time	200	300	-	ns
Trst	Reset code, low level time	200	-	-	us

- When writing programs, the minimum symbol period requirement is 1.2us;
- The high-level time of 0 code and 1 code must follow the specified range in the table above, and the low-level time of 0 code and 1 code must be less than 20us;

Protocol Data Format

Trst + 24bits data of first chip + 24bits data of second chip + ... + 24bits data of Nth chip + Trst

24bit grayscale data structure: high order first

G7	G6	G5	G4	G3	G2	G1	G0	R7	R6	R5	R4	R3	R2	R1	R0	B7	B6	B5	B4	B3	B2	B1	B0

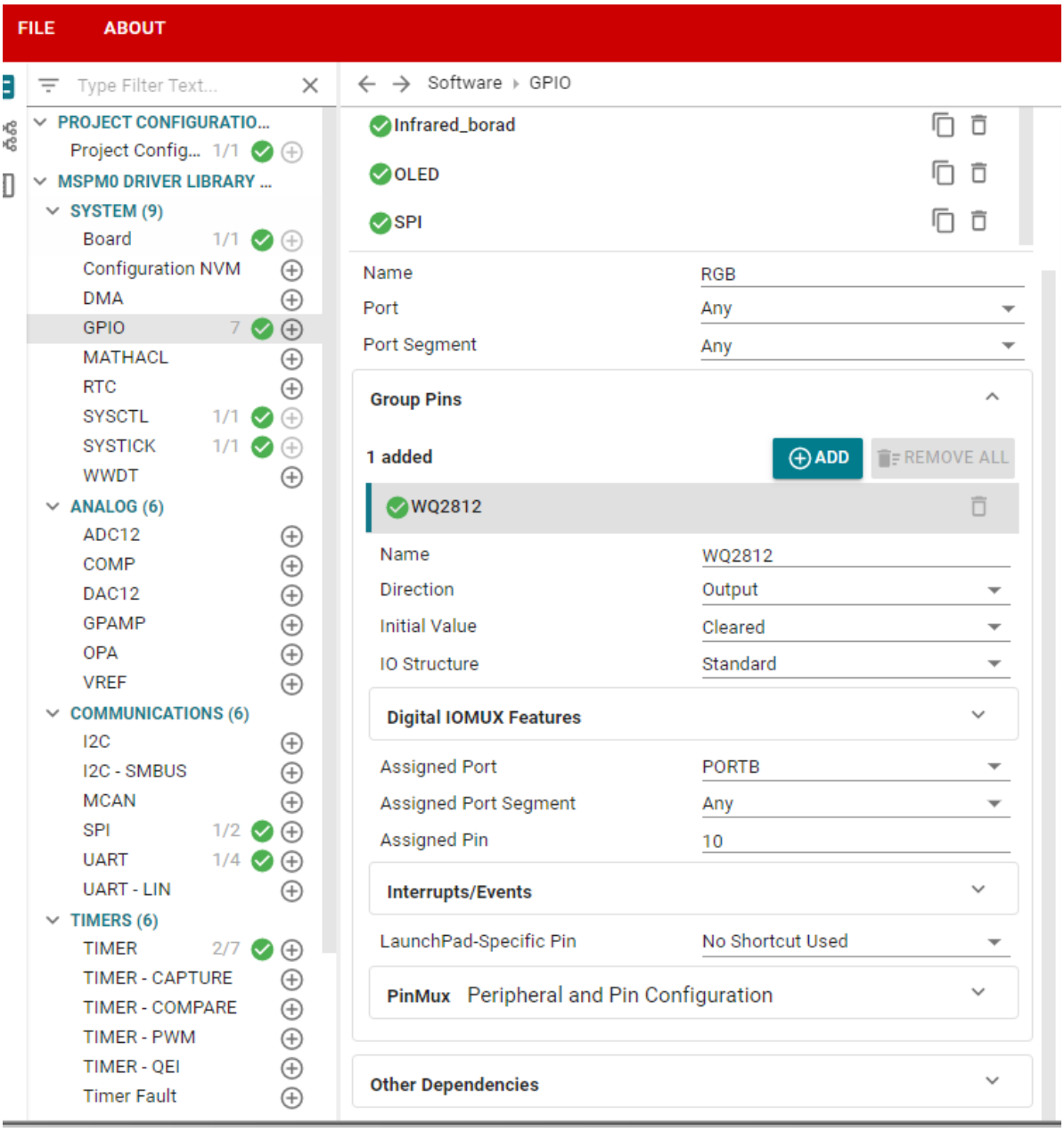
Note: High order is sent first, data is sent in GRB order.

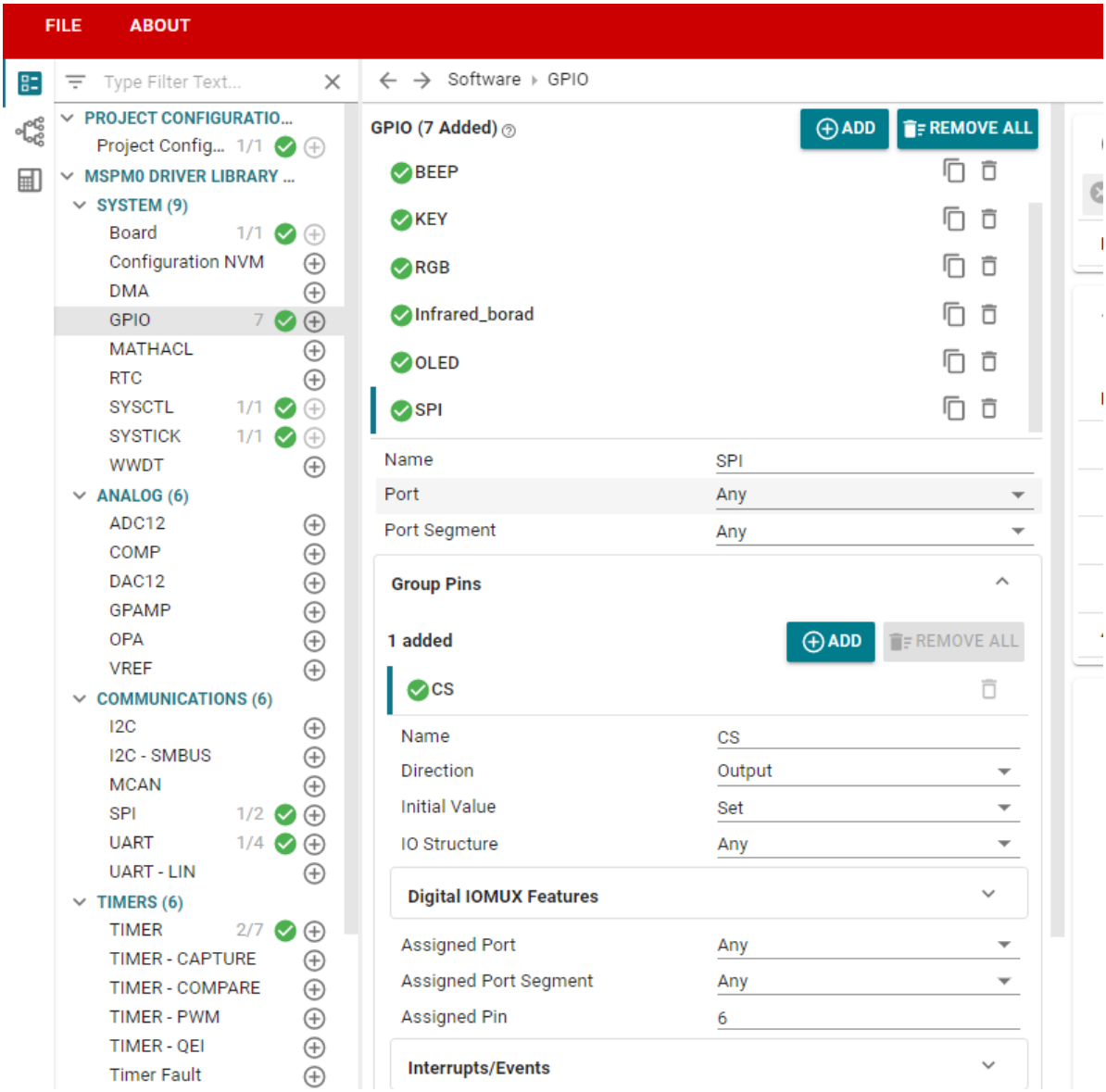
3. Project Configuration

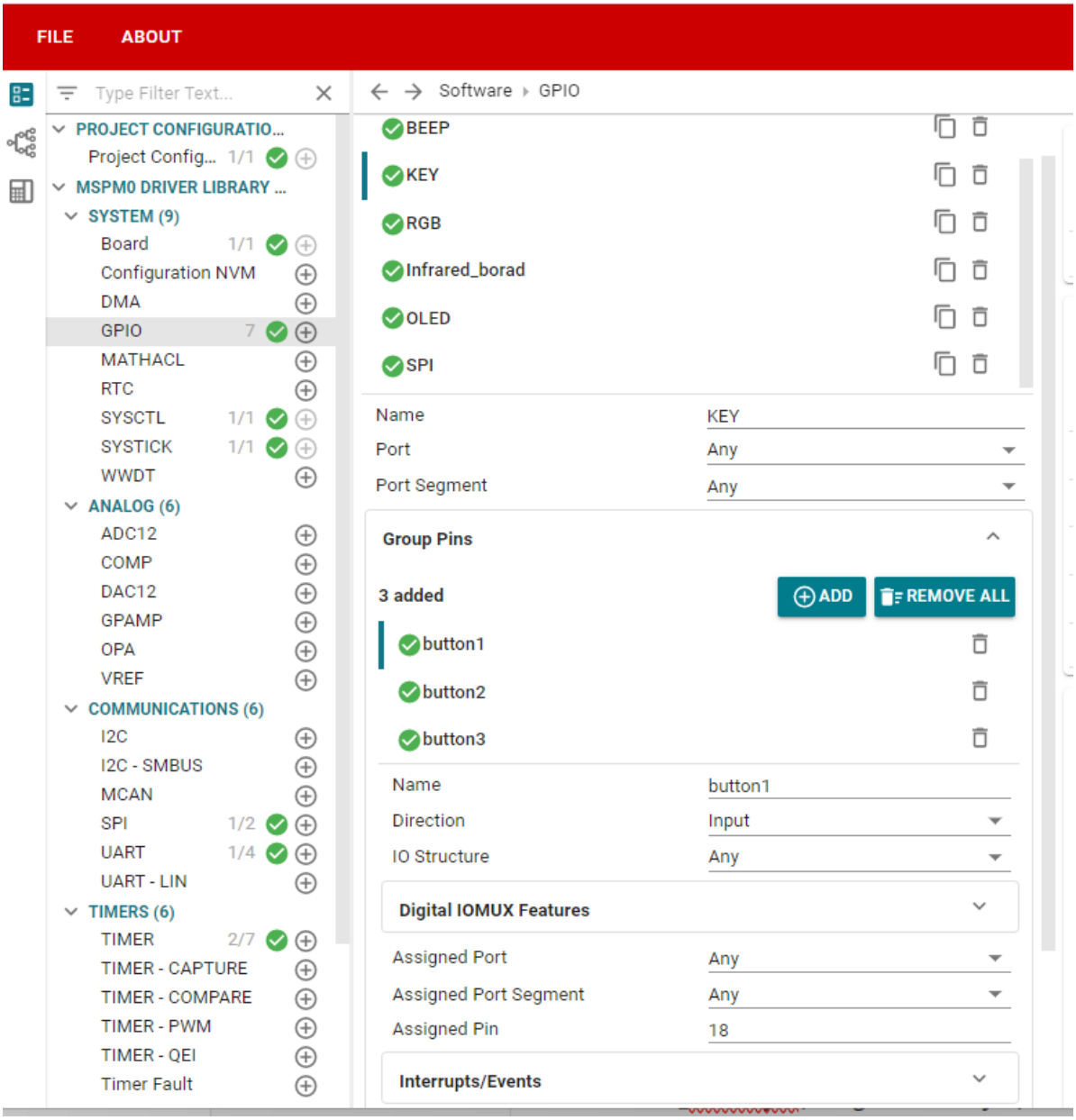
3.1 Description

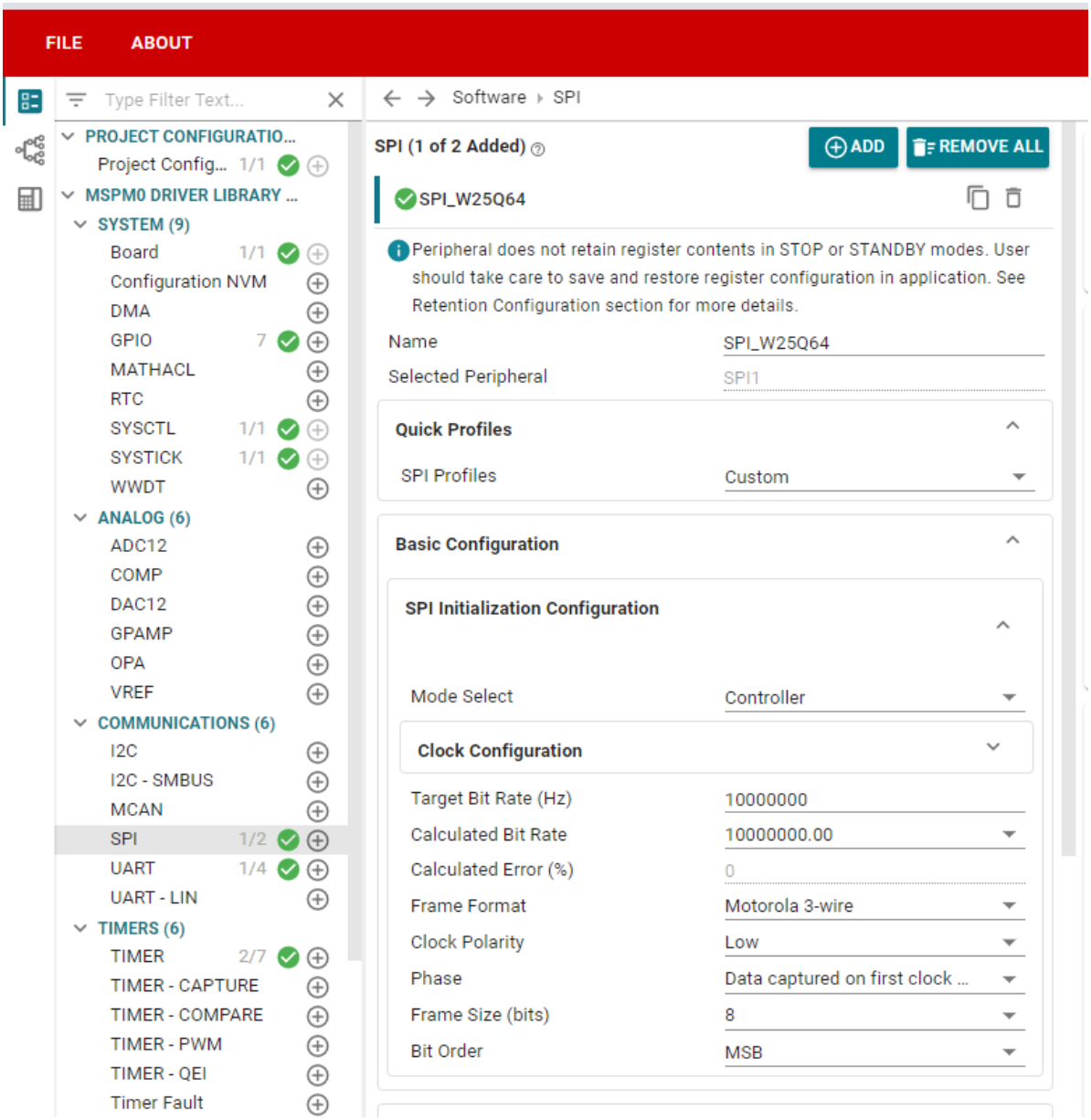
You can refer to the basic tutorial to complete the development environment setup.

3.2 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 7
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
TIMER - QEI
Timer Fault
Software ▸ GPIO
GPIO (7 Added)
ADD
REMOVE ALL
BEEP
KEY
RGB
Infrared_borad
OLED
SPI
Name SPI
Port Any
Port Segment Any
Group Pins
1 added
ADD
REMOVE ALL
CS
Name CS
Direction Output
Initial Value Set
IO Structure Any
Digital IOMUX Features
Assigned Port Any
Assigned Port Segment Any
Assigned Pin 6
Interrupts/Events


FILE
ABOUT
Type Filter Text...
PROJECT CONFIGURATIO...
Project Config... 1/1
MSPM0 DRIVER LIBRARY ...
SYSTEM (9)
Board 1/1
Configuration NVM
DMA
GPIO 7
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
TIMER - QEI
Timer Fault
Software ▸ GPIO
BEEP
KEY
RGB
Infrared_borad
OLED
SPI
Name KEY
Port Any
Port Segment Any
Group Pins
3 added
ADD
REMOVE ALL
button1
button2
button3
Name button1
Direction Input
IO Structure Any
Digital IOMUX Features
Assigned Port Any
Assigned Port Segment Any
Assigned Pin 18
Interrupts/Events



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: Ws2812b_WriteByte**

Function Prototype	**void Ws2812b_WriteByte(unsigned char byte)**
Function Description	Write single byte data to WS2812
Input Parameters	byte: byte data to be written
Output Parameters	None

> **Function: rgb_SendArray**

Function Prototype	void rgb_SendArray(void)
Function Description	Send color light data
Input Parameters	None
Output Parameters	None

Function: BSP_Loop

Function Prototype	void BSP_Loop(void)
Function Description	Use buttons to control onboard RGB light
Input Parameters	None
Output Parameters	None

Function: breathing_effect

Function Prototype	void breathing_effect(uint8_t r, uint8_t g, uint8_t b)
Function Description	Set RGB breathing light
Input Parameter 1	r: first parameter of rgb
Input Parameter 2	g: second parameter of rgb
Input Parameter 3	b: third parameter of rgb
Output Parameters	None

Function: BSP_Loop

Function Prototype	void BSP_Loop(void)
Function Description	Use buttons to control onboard RGB light
Input Parameters	None
Output Parameters	None

Function: Key1_State

Function Prototype	uint8_t Key1_State(uint8_t mode)
Function Description	Read the state of button K1
Input Parameters	mode: set mode, 0: returns 1 continuously when pressed; 1: returns 1 only once when pressed
Output Parameters	Returns 1 when pressed, returns 0 when released.

5. Experimental Phenomenon

After downloading the program, the red light will be on. When key1 is pressed, the onboard RGB will switch between red light and off; when key2 is pressed, the onboard RGB will cycle through colors; when key3 is pressed, the onboard RGB will flash in breathing light mode.