

# SPI-Host Demo Getting Started Guide

## What You Need

---

- 1 x MQM730i board
- 1 x STM32U575ZIT6Q board
- 1 x PC with Windows
- 1 x Micro USB cable
- 1 x Type-C cable
- 6 x Dupont lines

## Overview

---

MQM730i is Qualcomm ultra-low micro-power WiFi development board that integrates dual band 2.4/5Ghz Wi-Fi function, 1.5MB RRAM, 640KB SRAM and 2M NOR flash.

STM32U575ZIT6Q provides an affordable and flexible way for users to try out new concepts and build prototypes by choosing from the various combinations of performance and power-consumption features provided by the STM32U5 microcontroller.

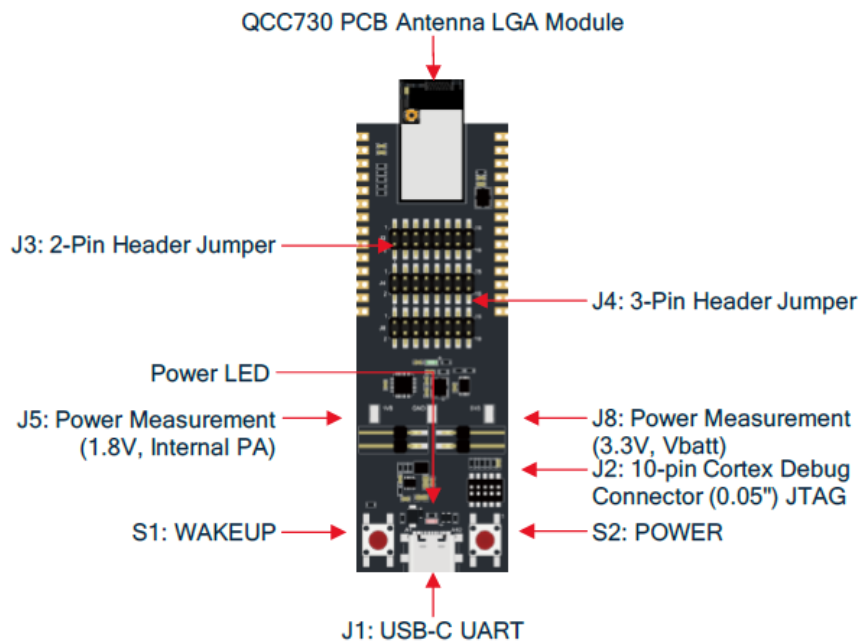
Attach MQM730i to STM32U575ZIT6Q through SPI.

## Hardware Preparation

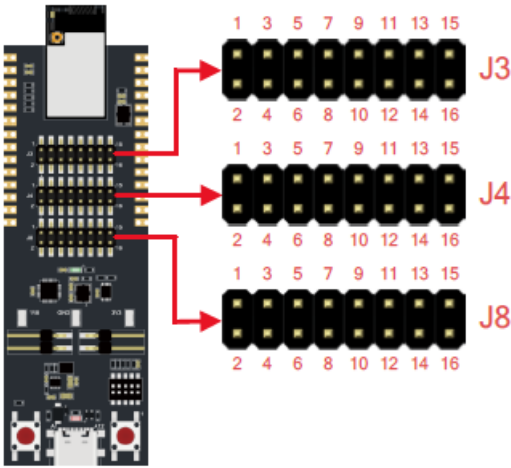
---

The list and figure below describe the key components, interfaces and functions of the MQM730i development board:

### Component Layout



**Jumper Setting**



Each individual jumper setting is defined in the table below:

| Jumper             | Pin 1 | Pin 2 | Function                                              |
|--------------------|-------|-------|-------------------------------------------------------|
| J3<br>(2x8 Jumper) | 1     | 2     | CH347F TCK to QCC730 TCK connect enable               |
|                    | 3     | 4     | CH347F TRST to QCC730 TRST_N connect enable           |
|                    | 5     | 6     | CH347F TDO to QCC730 TDO connect enable               |
|                    | 7     | 8     | CH347F TDI to QCC730 TDI connect enable               |
|                    | 9     | 10    | CH347F TMS to QCC730 TMS connect enable               |
|                    | 11    | 12    | CH347F TXD to GPIO14 (UART_TX) connect enable         |
|                    | 13    | 14    | CH347F RXD to GPIO13 (UART_RX) connect enable         |
|                    | 15    | 16    | 32.768kHz oscillator to GPIO8 (32k_IN) connect enable |
| J4<br>(2x8 Jumper) | 1     | 2     | 1V8 (iPA) power supply enable                         |
|                    | 3     | 4     | xFEM power supply enable                              |
|                    | 5     | 7     | VBAT set to 3.3V                                      |
|                    | 7     | 9     | VBAT set to 1.8V                                      |
|                    | 6     | 8     | VDDIO set to 3.3V                                     |
|                    | 8     | 10    | VDDIO set to 1.8V                                     |
|                    | 11    | 12    | CH347F VDD set to 1.8V                                |
|                    | 11    | 13    | CH347F VDD set to 3.3V                                |
|                    | 15    | 16    | USB 5V power supply enable                            |
| J8<br>(2x8 Jumper) | 1     | 2     | XPG5G control by GPIO7                                |
|                    | 1     | 3     | XPG5G control by GPIO0                                |
|                    | 3     | 4     | CH347F SCK to GPIO0 (SPI_SCK) connect enable          |
|                    | 5     | 6     | XPG2G control by GPIO6                                |
|                    | 5     | 7     | XPG2G control by GPIO2                                |
|                    | 7     | 8     | CH347F MOSI to GPIO2 (SPI_MOSI) connect enable        |
|                    | 9     | 10    | CH347F SCS0 to GPIO1 (SPI_CS) connect enable          |
|                    | 11    | 12    | CH347F MISO to GPIO3 (SPI_MISO) connect enable        |
|                    | 13    | 14    | EEPROM SDA to GPIO9 (I2C_SDA) connect enable          |
|                    | 15    | 16    | EEPROM SCL to GPIO10 (I2C_SCL) connect enable         |

The list and figure below describe the key components, interfaces and functions of the STM32U575ZIT6Q development board, please refer xxx.

**Software Development**

**Preparation**

- Go through [QCC730.FR.1.0 Programming Guide](#) to set up the development environment of MQM730i.
- Go through [00010.1 Release Note for QCC730.FR.1.0](#) to get the SDK of MQM730i.

- Go through xxx to set up the development environment of STM32U575ZIT6Q, in which STM32CubeIDE is used.
- Connect MQM730i to STM32U575ZIT6Q with Dupon Lines including SPI, interrupt and GND pins following with figure below.

| function                     | GPIO  | STM32U575ZIT6Q |
|------------------------------|-------|----------------|
| SPI_CLK                      | GPIO0 | PA5            |
| SPI_CS                       | GPIO1 | PA4            |
| SPI_MOSI                     | GPIO2 | PA7            |
| SPI_MISO                     | GPIO3 | PA6            |
| Interrupt (MQM730i to STM32) | GPIO8 | PC12 or others |
| GND                          |       |                |
|                              |       |                |

- Connect MQM730i to your PC with a Type-C cable
- Connect STM32U575ZIT6Q to your PC with a Micro SUB cable

### Setup MQM730i software for SPI host demo

Follow these steps to prepare the build and flash the image:

1. Set the following configuration in the qccsdk\demo\qcli\_demo\prj.conf file:

CONFIG\_RING\_IF\_ONLY=y

CONFIG\_QCSPI\_HFC\_TEST=y

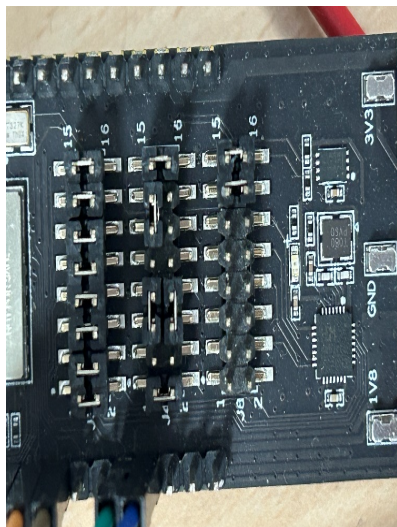
2. Build an image by using the following commands:

```
python qccsdk.py set -S=demo/qcli_demo -b=mqm730i python qccsdk.py build --rebuild
```

3. Flash the image to a QCC730 board by using the following command:

```
python qccsdk.py flash --flash --bdf--reset
```

The jumper on QCC730:



### Setup STM32U575ZIT6Q software for SPI host demo

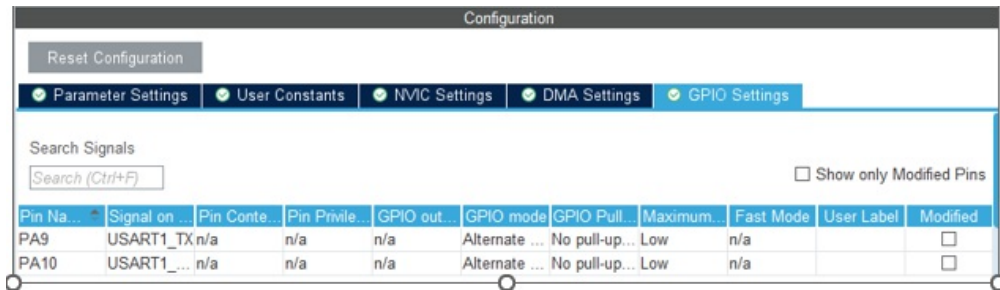
Follow these steps to prepare the build and flash the image using STM32CubeIDE:

1. Create STM32 project for SPI host demo following with guide [STM32CubeIDE: Multi-OS development tool](#).

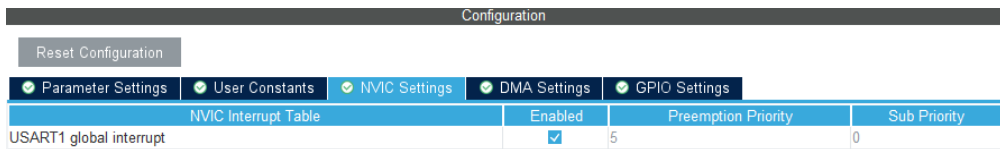
- Commercial Part Number: STM32U575ZIT6Q

2. Clock Configuration



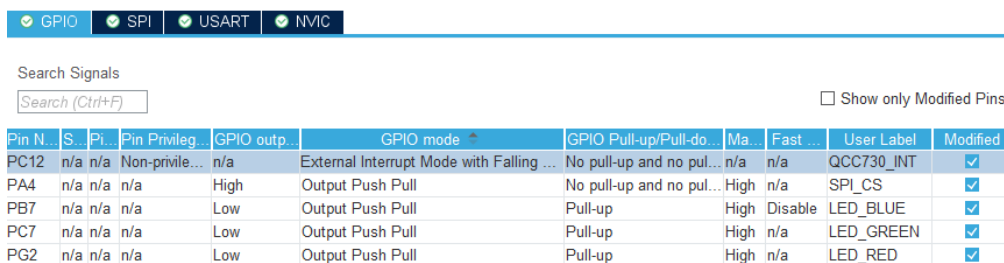


- UART Interrupt

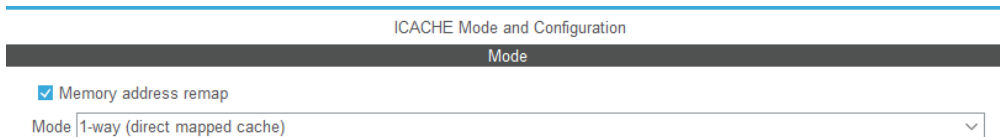


## 5. Enable GPIO

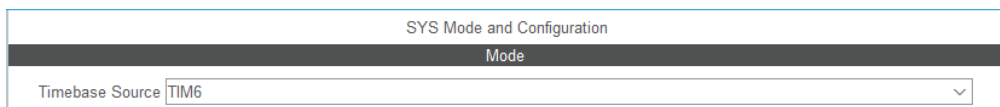
- PA4: SPI\_CS
- PB7: LED\_BLUE
- PC7: LED\_GREEN
- PG2: LED\_RED
- PC12: QCC730\_INT



## 6. Enable ICACHE

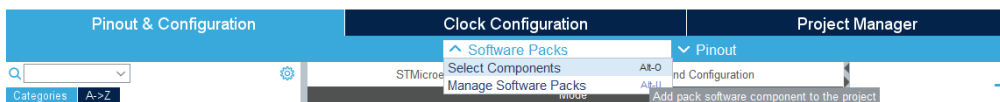


## 7. Enable SYS



## 8. Enable X-CUBE-FREERTOS (Middleware and Software Packs )

- Software Packs: Select Components



- CMSIS RTOS2

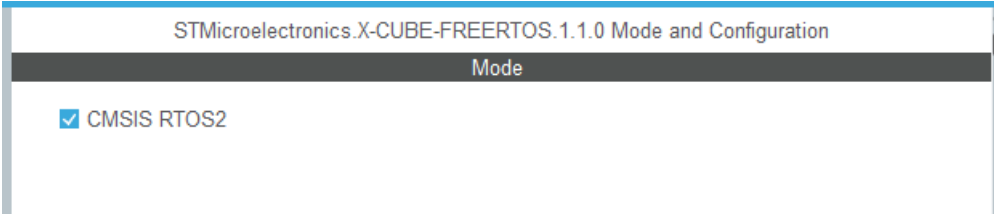
Core: TZ\_Non\_Support

Heap: Heap\_4

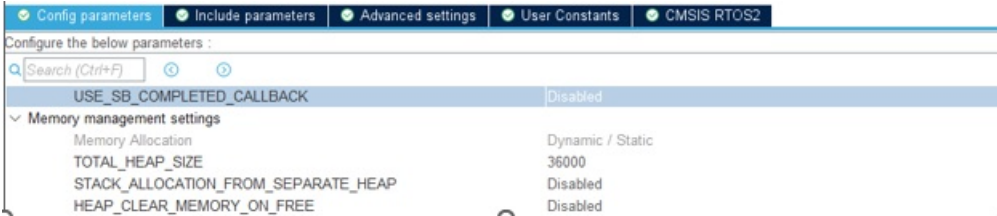
Packs

| Pack / Bundle / Component            | Status | Version | Selection     |  |
|--------------------------------------|--------|---------|---------------|--|
| ▼ STMicroelectronics.X-CUBE-FREERTOS | ✓      | 1.1.0   | ▼             |  |
| ▼ CMSIS RTOS2                        | ✓      | 10.5.1  |               |  |
| ▼ RTOS2                              | ✓      |         |               |  |
| Core                                 | ✓      | 10.5.1  | TZ_Non_S... ▼ |  |
| Heap                                 | ✓      | 10.5.1  | Heap_4 ▼      |  |

- CMSIS RTOS2 mode



- Memory management setting: TOTAL\_HEAP\_SIZE 36000



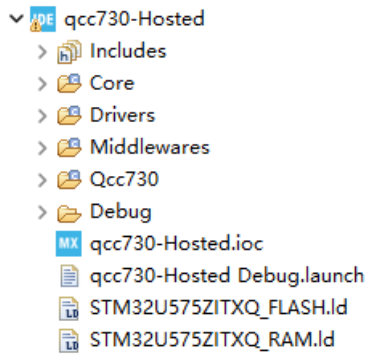
9. Project Generate Code

10. Import the code of SPI host demo

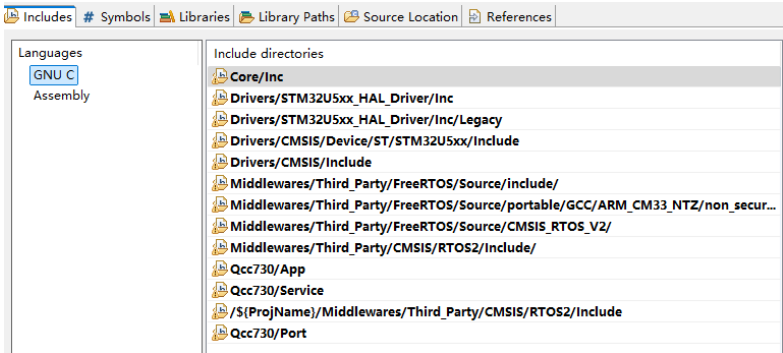
- Import “Qcc730” folder from “qcc730\_sdk\comp\exhost\host\_stm32Nucleo\ Qcc730” to project.

11. The configure of Project

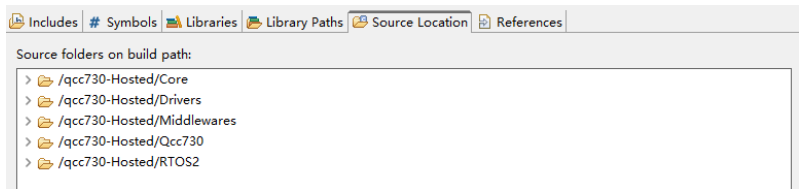
### Project File



### Includes



## Source Location



### Enable SPI host demo

#### main.h

```
#define SHELL_FEATURE
```

#### main.c

```
/* USER CODE BEGIN Includes */
```

```
#include "stdio.h"
```

```
#include "led.h"
```

```
#include "spi.h"
```

```
#ifdef SHELL_FEATURE
```

```
#include "shell.h"
```

```
#endif
```

```
/* USER CODE END Includes */
```

```
/* USER CODE BEGIN PV */
```

```
#ifdef __GNUC__
```

```
/* With GCC, small printf (option LD Linker->Libraries->Small printf
```

```
set to 'Yes') calls __io_putchar() */
```

```
#define PUTCHAR_PROTOTYPE int __io_putchar(int ch)
```

```
#else
```

```
#define PUTCHAR_PROTOTYPE int fputc(int ch, FILE *f)
```

```
#endif /* __GNUC__ */
```

```
PUTCHAR_PROTOTYPE
```

```
{
```

```
/* Place your implementation of fputc here */
```

```
/* e.g. write a character to the EVAL_COM1 and Loop until the end of transmission */
```

```
HAL_UART_Transmit(&huart1, (uint8_t *)&ch, 1, 0xFFFF);
```

```
return ch;
```

```
}
```

```

static void EXTI12_IRQHandler_Config(void);

/* USER CODE END PV */

/* USER CODE BEGIN 4 */

static void EXTI12_IRQHandler_Config(void)
{
    GPIO_InitTypeDef  GPIO_InitStructure;

    /* Enable GPIOC clock */

    __HAL_RCC_GPIOC_CLK_ENABLE();

    /* Configure PC.13 pin as input floating */

    GPIO_InitStructure.Mode = GPIO_MODE_IT_FALLING;

    GPIO_InitStructure.Pull = GPIO_NOPULL;

    GPIO_InitStructure.Pin = QCC730_INT_Pin;

    HAL_GPIO_Init(GPIOC, &GPIO_InitStructure);

    /* Enable and set line 13 Interrupt to the lowest priority */

    HAL_NVIC_SetPriority(EXTI12_IRQn, 2, 0);

    HAL_NVIC_EnableIRQ(EXTI12_IRQn);
}

extern F2A_int_handler();

void HAL_GPIO_EXTI_Falling_Callback(uint16_t GPIO_Pin)
{
    if(GPIO_Pin == QCC730_INT_Pin)
    {
        /* Toggle LED1 */

        // BSP_LED_Toggle(LED_GREEN_Pin);

        //printf("Button pressed\r\n");

        F2A_int_handler();

        //HAL_GPIO_WritePin(GPIOC, QCC730INT_Pin, GPIO_PIN_SET); //chip_on test

        //HAL_GPIO_TogglePin(GPIOC, QCC730INT_Pin);
    }
}

```

```

#ifdef SHELL_FEATURE

void HAL_UART_RxCpltCallback(UART_HandleTypeDef* huart){

    if(huart->Instance == USART1){

        xQueueSendFromISR(qShell, &c, NULL);

        HAL_UART_Receive_IT(&huart1, (uint8_t*)&c, 1);

    }

}

#endif

/* USER CODE END 4 */

```

## function:

```

int main(void)

{

...

/* USER CODE BEGIN 2 */

EXTI12_IRQHandler_Config();

transport_init();

```

```

#ifdef SHELL_FEATURE

    task_init_all();

#endif

/* USER CODE END 2 */

```

```

...

}

static void MX_GPIO_Init(void)

{

...

__HAL_RCC_GPIOB_CLK_ENABLE();

/*Configure GPIO pin Output Level */

HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_SET);

/*Configure GPIO pin : SPI_CS_Pin */

}

```

```
/* USER CODE BEGIN 1 */
```

```
void EXTI12_IRQHandler(void)

{

    HAL_GPIO_EXTI_IRQHandler(QCC730_INT_Pin);

}

/* USER CODE END 1 */
```

12. **BUILD and Flash the SPI host demo image**

# SPI Host Demo

**Command on STM32**

- 1. **start 2**
- 2. **spi2 10 1000**
- 3. **stop 2**

**Test Log on STM32:**

STM32 >> **start 2**

test\_qcspi\_start arc=2 210010 [210010]

STM32 >> **spi2 10 1000**

spi w/r with interrupt

app\_tsrx\_demo size=1000

Start write RING 192208

APP tx add 20006a10, tx len=1000 data: a5 a5 a5 a5

send 10000 bytes time:[now]=192269,time1=192208,interval:61 ms

STM32 >> **stop 2**

demo\_stop\_test=2 20 [20]

END: info\_type=5,type=1,size=24, mode=2, rcv\_bytes=10000, send\_bytes=10000

**Test Log on QCC730:**

>RingIf: hfc msg type 0 id 0 (p\_buf: 5fc90 len:16 data:5fc90)

\*\*\*test start mode 2

>

>RingIf: hfc msg type 1 id 0 (p\_buf: 87218 len:1000 data:87218)

rcv msg\_id 0 data 87218 len 1000: a5a5a5a5a5...a5a5a5a5a5

```
send data 87218 len 1000.

RingIf: hfc msg type 1 id 0 (p_buf: 73d48 len:1000 data:73d48)

recv msg_id 0 data 73d48 len 1000: a5a5a5a5a5...a5a5a5a5a5

send data 73d48 len 1000.

RingIf: hfc msg type 1 id 0 (p_buf: 87860 len:1000 data:87860)

recv msg_id 0 data 87860 len 1000: a5a5a5a5a5...a5a5a5a5a5

send data 87860 len 1000.

RingIf: hfc msg type 1 id 0 (p_buf: 87ea8 len:1000 data:87ea8)

recv msg_id 0 data 87ea8 len 1000: a5a5a5a5a5...a5a5a5a5a5

send data 87ea8 len 1000.

RingIf: hfc msg type 1 id 0 (p_buf: 74390 len:1000 data:74390)

recv msg_id 0 data 74390 len 1000: a5a5a5a5a5...a5a5a5a5a5

send data 74390 len 1000.

RingIf: hfc msg type 1 id 0 (p_buf: 884f0 len:1000 data:884f0)

recv msg_id 0 data 884f0 len 1000: a5a5a5a5a5...a5a5a5a5a5

send data 884f0 len 1000.

RingIf: hfc msg type 1 id 0 (p_buf: 749d8 len:1000 data:749d8)

recv msg_id 0 data 749d8 len 1000: a5a5a5a5a5...a5a5a5a5a5

send data 749d8 len 1000.

RingIf: hfc msg type 1 id 0 (p_buf: 75020 len:1000 data:75020)

recv msg_id 0 data 75020 len 1000: a5a5a5a5a5...a5a5a5a5a5

send data 75020 len 1000.

RingIf: hfc msg type 1 id 0 (p_buf: 776d8 len:1000 data:776d8)

recv msg_id 0 data 776d8 len 1000: a5a5a5a5a5...a5a5a5a5a5

send data 776d8 len 1000.

RingIf: hfc msg type 1 id 0 (p_buf: 77d20 len:1000 data:77d20)

recv msg_id 0 data 77d20 len 1000: a5a5a5a5a5...a5a5a5a5a5

send data 77d20 len 1000.


RingIf: hfc msg type 0 id 1 (p_buf: 5fd98 len:16 data:5fd98)

test end: mode 2 send_len 10000 recv_len 10000
```