Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

## Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

# THIS SPEC IS OBSOLETE

Spec No: 002-04393

Spec Title: AN204393 - FM0+ Starter Kit 32-Bit Microcontroller

Replaced by: 002-05536

CYPRESS PERFORM

AN204393

# FM0+ Starter Kit 32-Bit Microcontroller

**Target Products: S6E1B8 Series**

This application note describes the starter kit SK-FM0-100L-S6E1B8 and how to use it.

OBSOLETE

## Contents

1 Introduction .......... 1
1.1 Document Purpose .......... 1
1.2 Definitions, Acronyms, and Abbreviations .......... 1
1.3 Document Overview .......... 2
2 Overview and Features .......... 2
2.1 Board Overview .......... 2
2.2 Board Features .......... 3
2.3 System Block Diagram .......... 3
3 Software Preparation and Installation .......... 4
3.1 CMSIS-DAP Driver Installation .......... 4
3.2 Virtual Communication Port Installation .......... 5
4 Board Description and Connection .......... 7
4.1 Hardware Description .......... 7
4.2 Jumpers Description .......... 7
4.3 4Connectors Description .......... 8
4.4 Hardware Connection and IAR Configuration .. 8
5 Firmware Architecture .......... 11
5.1 Firmware Architecture Diagram .......... 11
6 Module Labs Operation .......... 12
6.1 Control the RGB LED using Base Timer .......... 12
6.2 UART Receive/Send Data with CMSIS-DAP . 19
6.3 Measure the Potentiometer Analog Voltage .. 25
6.4 Measure Acceleration with KXCJK-1013 Accelerometer .......... 31
6.5 Read/Write SD card .......... 40
6.6 Play Pixie Dust Sound Data .......... 47
7 Additional Information .......... 52
8 Document History .......... 53

# 1 Introduction

## 1.1 Document Purpose

This application note describes the starter kit SK-FM0-100L-S6E1B8 and how to use it.

## 1.2 Definitions, Acronyms, and Abbreviations

| | |
|---|---|
| UART | Universal Asynchronous Receiver/Transmitter |
| USB | Universal Serial Bus |
| LED | Light Emitting Diode |
| LDO | Low-Drop-Out linear voltage regulator |
| I/F | Interface |
| I2S | Integrated Interchip Sound |
| I/O | Input and Output |
| PDL | Peripheral Driver Library |

OBSOLETE

### 1.3 Document Overview

The rest of document is organized like this:

Section 2 explains Overview and Features

Section 3 explains Software Preparation and Installation.

Section 4 explains Board Description and Connection.

Section 5 explains Firmware Architecture.

Section 6 explains Module Labs Operation.

Section 7 explains Additional Information

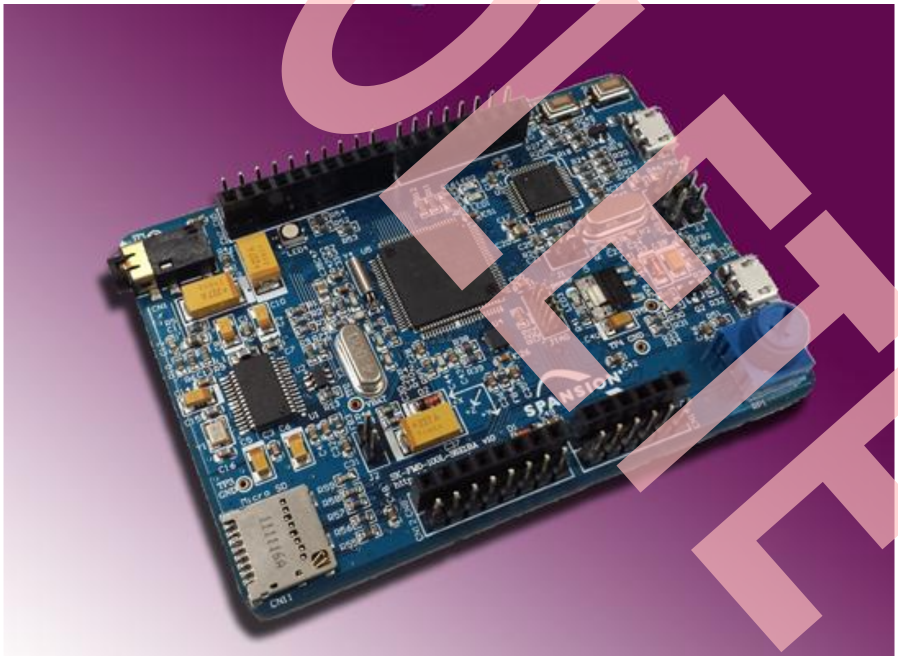
## 2 Overview and Features

### 2.1 Board Overview

The SK-FM0-100L-S6E1B8 is a Starter Kit for Cypress S6E1B8 Series microcontrollers.

The S6E1B8 Series is highly integrated 32-bit microcontrollers designed for embedded controllers aiming at low power consumption and low cost. This series has the ARM Cortex-M0+ Processor with on-chip Flash memory and SRAM, and consists of peripheral functions such as various timers, ADCs and communication interfaces (UART, CSIO, I$^2$C, LIN, I2S, and USB).

Figure 1. Board Overview



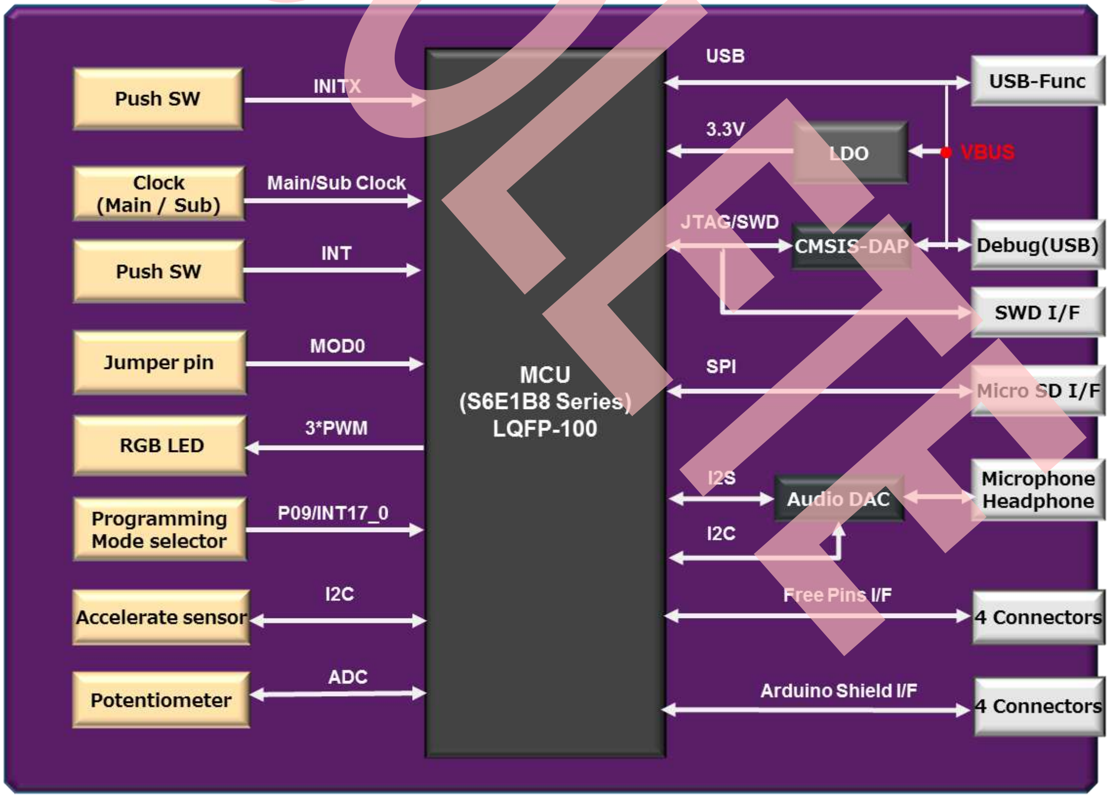
OBSOLETE

## 2.2 Board Features

- Cypress FM0+ S6E1B8 MCU
- CMSIS DAP interface
- JTAG interface
- Free Pins interface
- Arduino Shield interface
- USB Device (Micro type-B)
- SD card interface
- Acceleration Sensor
- Potentiometer
- Microphone/ Headphone interface
- RGB LED
- 2 x Push-button

## 2.3 System Block Diagram

Figure 2. System Block Diagram



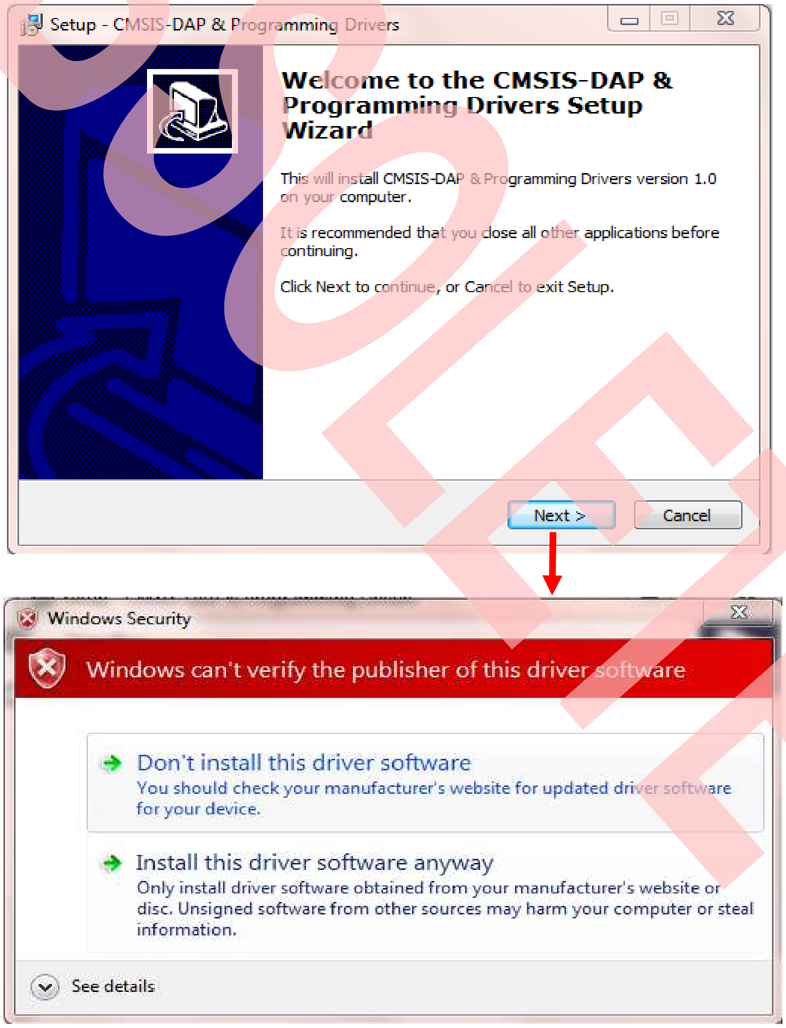
OBSOLETE

# 3 Software Preparation and Installation

## 3.1 CMSIS-DAP Driver Installation

- Please download the .zip file SK-FM0-100L-S6E1B8.vxyz.
- Open subfolder: \tools\ cmsisdap_fw_update.
- Double-click the “setup.exe” to install the driver.
- Click “**Next**” to continue.
- Select “**Install this driver software anyway**” (twice for two driver installations).
- Click the “**Finish**” button when the installation is completed.

Figure 3. CMSIS-DAP Driver Installation



**Welcome to the CMSIS-DAP & Programming Drivers Setup Wizard**

This will install CMSIS-DAP & Programming Drivers version 1.0 on your computer.

It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.

Windows can't verify the publisher of this driver software

Don't install this driver software
You should check your manufacturer's website for updated driver software for your device.

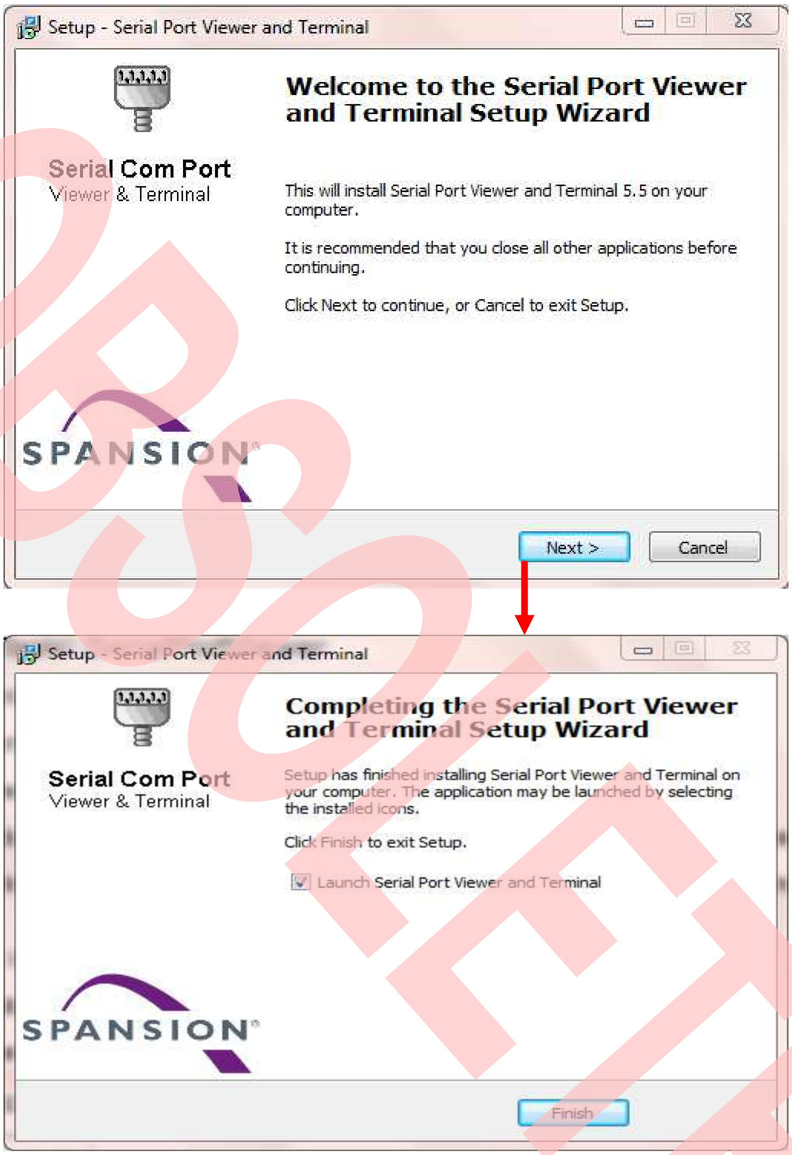
Install this driver software anyway
Only install driver software obtained from your manufacturer's website or disc. Unsigned software from other sources may harm your computer or steal information.

## 3.2 Virtual Communication Port Installation

- Open subfolder: Tools\ SerialPortViewer.
- Double-click “SerialPortViewerAndTerminalV5.5.exe” to install this software.
- Click “**Next**” to continue.
- Read the license agreement, and select “**I accept the agreement**”.
- Select a destination for the application or leave the default location, and click “**Next**”.
- Select a destination for the application shortcut or leave it in default location, and click “**Next**”.
- Select additional icon options and click “**Next**”.
- Select “**Install**” to start the installation.
- Check the box next to Launch Cypress Serial Port Viewer and Terminal and click “**Finish**”.

OBSOLETE

Figure 4. Virtual Communication Port Installation



Setup - Serial Port Viewer and Terminal

**Welcome to the Serial Port Viewer and Terminal Setup Wizard**

This will install Serial Port Viewer and Terminal 5.5 on your computer.

It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.

Setup - Serial Port Viewer and Terminal

**Completing the Serial Port Viewer and Terminal Setup Wizard**

Setup has finished installing Serial Port Viewer and Terminal on your computer. The application may be launched by selecting the installed icons.

Click Finish to exit Setup.

☑ Launch Serial Port Viewer and Terminal

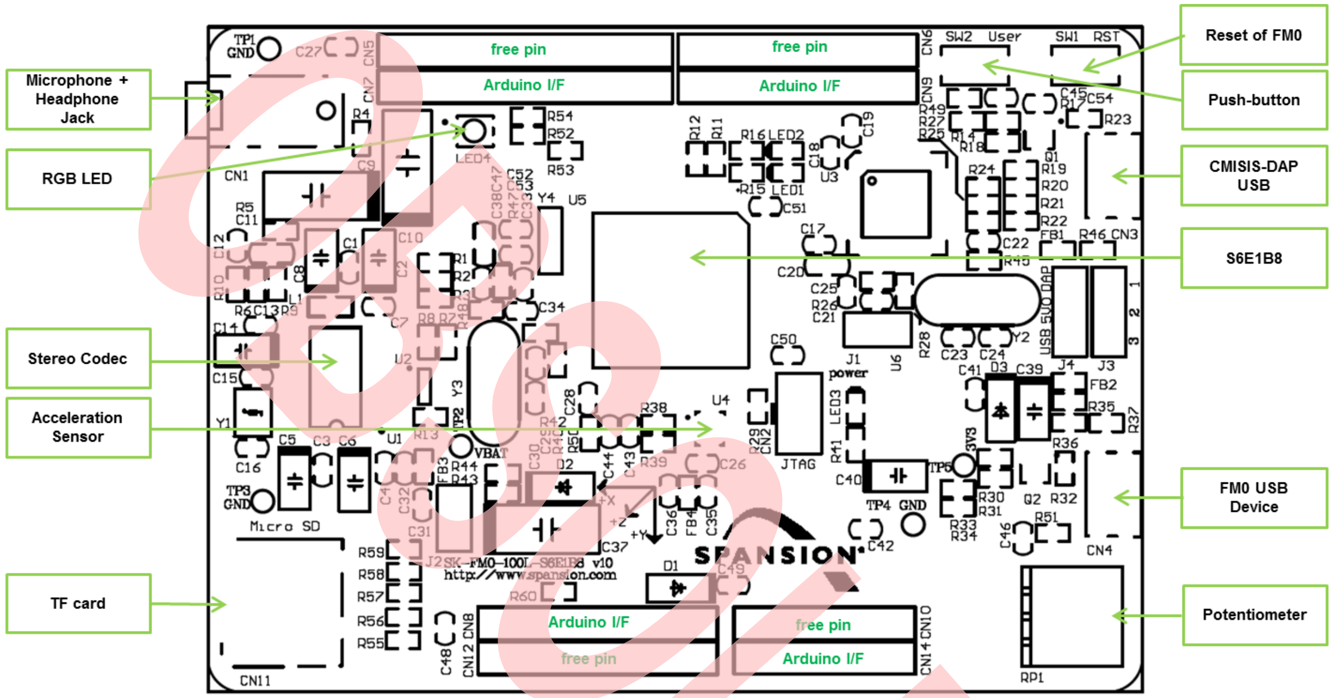
OBSOLETE

OBSOLETE

# 4 Board Description and Connection

## 4.1 Hardware Description

Figure 5. Components Layout



## 4.2 Jumpers Description

Jumper description is shown as below:

Table 1. Jumper Description

| Number | Name | Functions |
|---|---|---|
| J1 | CMSIS-DAP MD0 | Open: normal operation |
| | | Closed: serial programming mode |
| J2 | FM0+ MD0 | Open: normal operation |
| | | Closed: serial programming mode |
| J3 | UART or USB<br>programming mode selection | (1-2) UART |
| | | (2-3) USB |
| J4 | 5V power supply source selection (please select only one power source at the same time) | (1-2): DAP power supply (CN2) |
| | | (2-3): USB power supply (CN4) |

OBSOLETE

## 4.3 Connectors Description

CN1 to CN11 descriptions are:

Table 2. Connectors Description

| Number | Name | Functions |
|---|---|---|
| 1 | CN1 | Microphone + Headphone jack |
| 2 | CN2 | JTAG I/F (10pin) |
| 3 | CN3 | USB Device for CMSIS-DAP |
| 4 | CN4 | USB Device for FM0+ MCU |
| 5 | CN5/CN6/CN12/CN14 | Pin header of free pins |
| 6 | CN7/CN8/CN9/CN10 | Pin header of Arduino I/F |
| 7 | CN11 | Micro SD card socket |

## 4.4 Hardware Connection and IAR Configuration

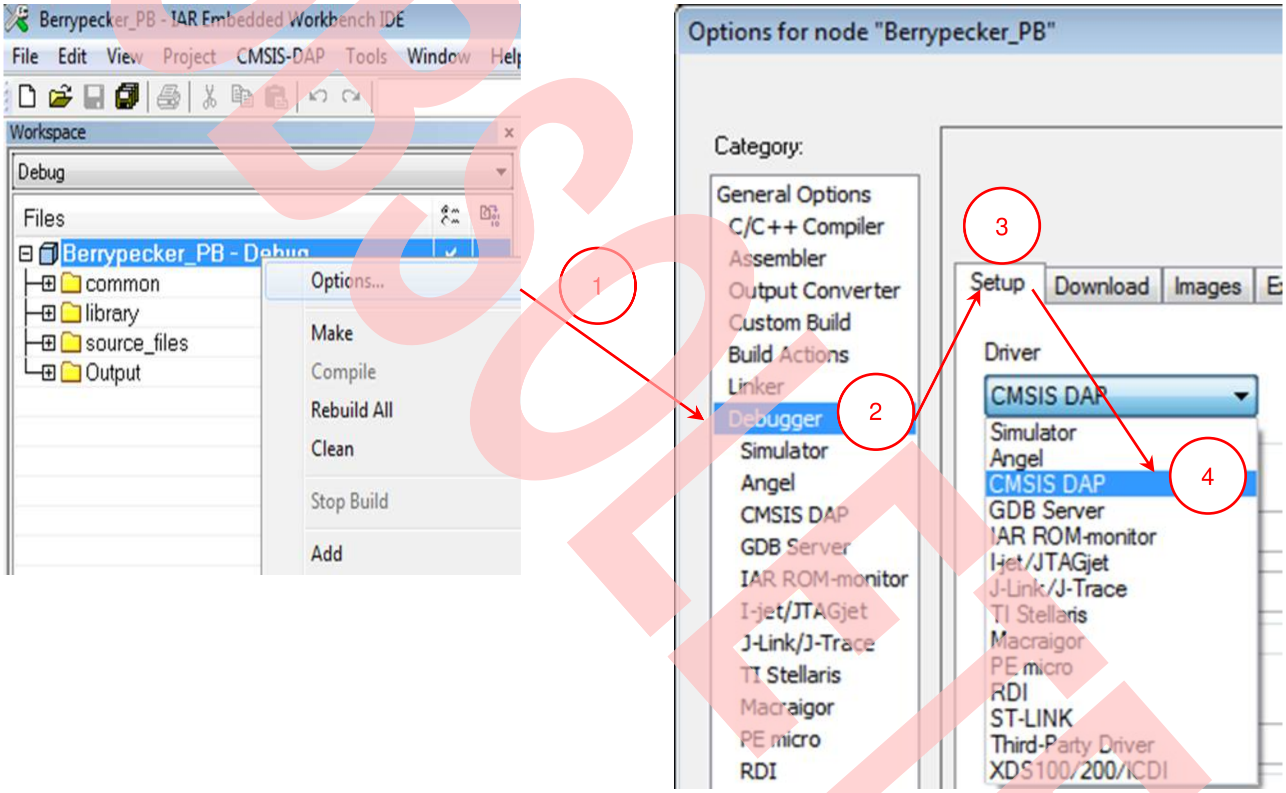
### 4.4.1 Power Supplies Selection

Because module lab will use CMSIS-DAP, close 1-2 of J4 jumper to select CMSIS-DAP USB power supply.

### 4.4.2 Select the CMSIS-DAP within IAR

- Open the project with IAR
- Right click on the project
- Select “**Options**”
- Select “**Debugger**”
- Click “**Setup**”
- Select “**CMSIS-DAP**”

Figure 6. Select the CMSIS-DAP

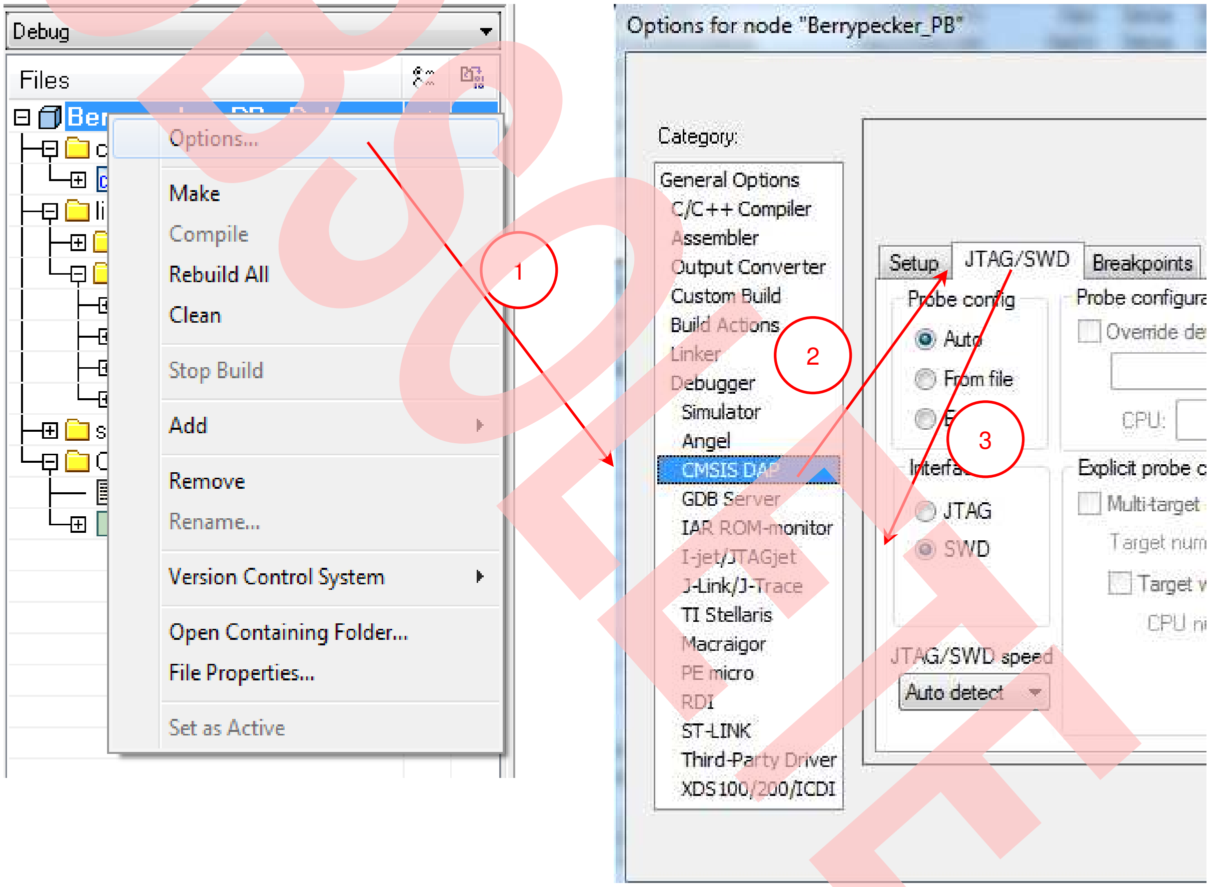


OBSOLETE

### 4.4.3 Select SWD Interface for CMSIS-DAP

- Open the project with IAR
- Right click on the project
- Select "**Options**"
- Click "**CMSIS-DAP**"
- Click "**JTAG/SWD**"
- Click "**SWD**"

Figure 7. Select SWD Interface



OBSOLETE

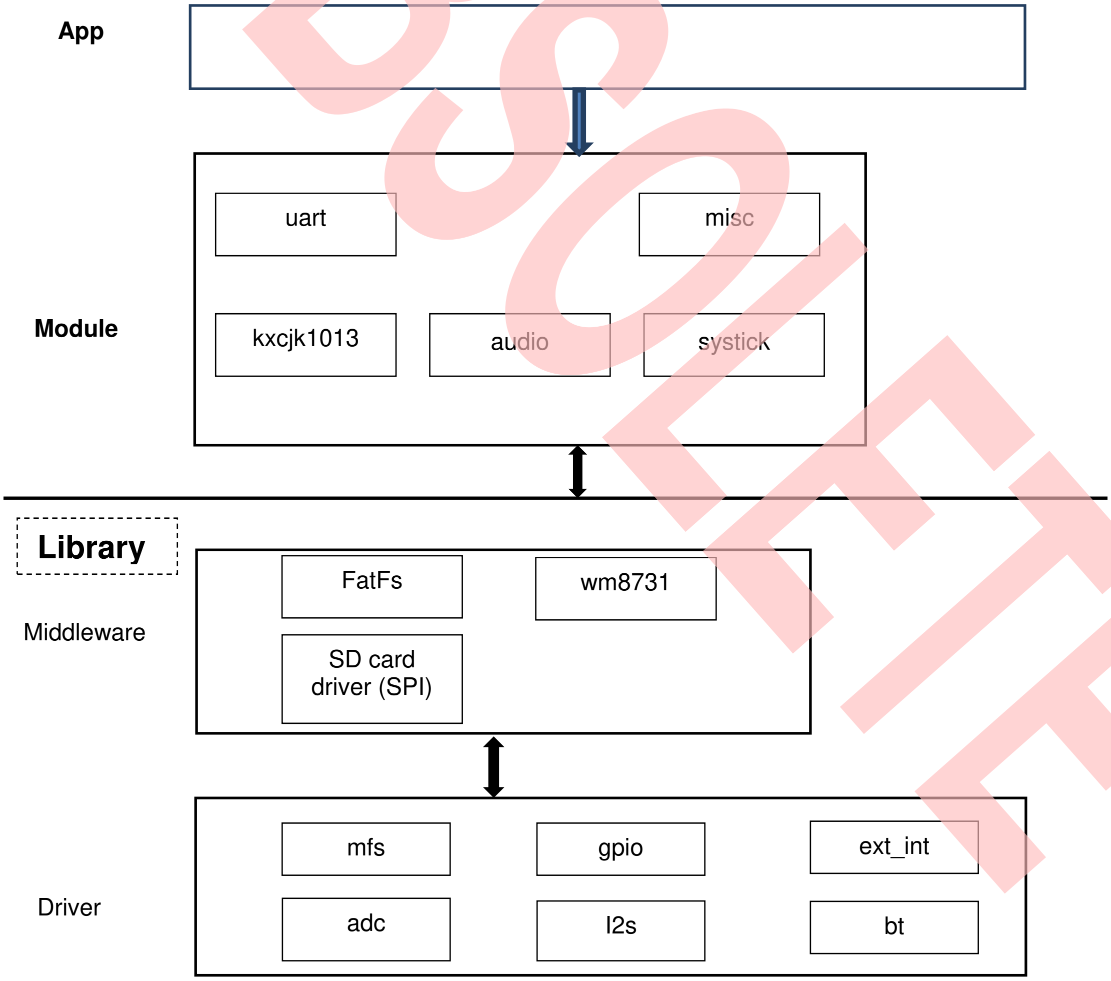
# 5 Firmware Architecture

## 5.1 Firmware Architecture Diagram

The FW subsystem architecture is shown as the Figure 8. There are 3 layers in subsystem:

1. **Library Layer:** This layer is a constant area. User must not change it. It provides all units' driving functions, such as GPIO driving function, timer driving function, UART driving function, etc. This layer is separated as 2 sub-layers: driver level and middleware level. For driver level, it directly operates the registers of the MCU. For middleware level, this layer implements adopted code (SD lib/fatFS/audio codec).
2. **Module Layer:** This layer is the highest layer of the system; it integrates the low level function and serves the system as the API (i2c, uart, audio, misc).
3. **App Layer:** The main function of the system is achieved by this layer

Figure 8. Firmware Architecture Diagram



OBSOLETE

OBSOLETE

# 6 Module Labs Operation

## 6.1 Control the RGB LED using Base Timer

### 6.1.1 Lab Objective

1. Learning the base principle and PWM.
2. Master the BT driver function about FM0+ driver

### 6.1.2 Lab Content

Use three PWM channels to light on or off RGB LED.

### 6.1.3 Preliminary Knowledge

1. Master the basic process of coding and debugging the program in the IAR integration development environment.
2. Understand FM0+ Peripheral Library framework

### 6.1.4 Preparation of Lab Equipment and Tools

1. Hardware
   a. Mini USB cable
   b. SK-FM0-100L-S6E1B8 board
   c. PC
2. Software
   a. IAR

### 6.1.5 Lab Principle and Instructions

1. The function of the base timer can be set to 16-bit PWM timer. When triggered, the 16-bit PWM timer starts decrementing from the cycle set value. First, it outputs a LOW level pulse. When the 16-bit down counter matches the value set in the PWM Duty Set Register, the output inverts to the HIGH level. Then, the output inverts again to the LOW level when a counter underflow occurs. This can generate waveforms with any cycle and duty. (More details to refer to FM0+ Family Peripheral Manual Timer Part). So set PWM duty to control the total current into LED
2. The Figure 9 shows that PIN22, PIN23 and PIN24 are selected to control RGB LED.

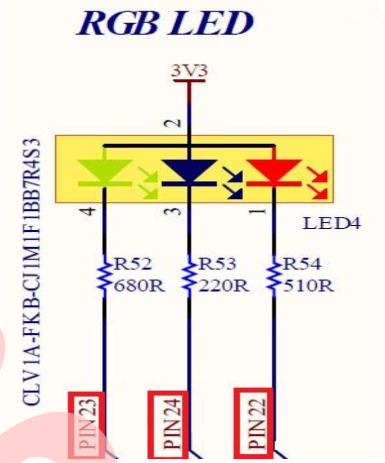
   The PIN22 has TIOA3_1 function at the output pin of BT ch.3 TIOA.

   The PIN23 has TIOA4_1 function at the output pin of BT ch.4 TIOA.

   The PIN24 has TIOA5_1 function at the output pin of BT ch.5 TIOA.

OBSOLETE

Figure 9. RGB LED Circuit



### 6.1.6 Lab Steps

1. Create new project based on FM0+ Peripheral Library.
2. Edit file pdl_user. to activate PDL resource and interrupt.

Figure 10. Activate Requested BT Channels

```
// Base Timers
#define PDL_PERIPHERAL_ENABLE_BT3               PDL_ON
#define PDL_PERIPHERAL_ENABLE_BT4               PDL_ON
#define PDL_PERIPHERAL_ENABLE_BT5               PDL_ON
// GPIO header inclusion
#define PDL_PERIPHERAL_ENABLE_GPIO              PDL_ON
```

Figure 11. Activate Requested BT Channels Interrupts

```
// Base Timers
#define PDL_INTERRUPT_ENABLE_BT3              PDL_ON
#define PDL_INTERRUPT_ENABLE_BT4              PDL_ON
#define PDL_INTERRUPT_ENABLE_BT5              PDL_ON
```

3. Create new file misc.c and misc.h; add the new files to the project
4. Define macros about the requested BT channels and BT I/O (misc.h )

OBSOLETE

Figure 12. Define Requested BT Channels Macros

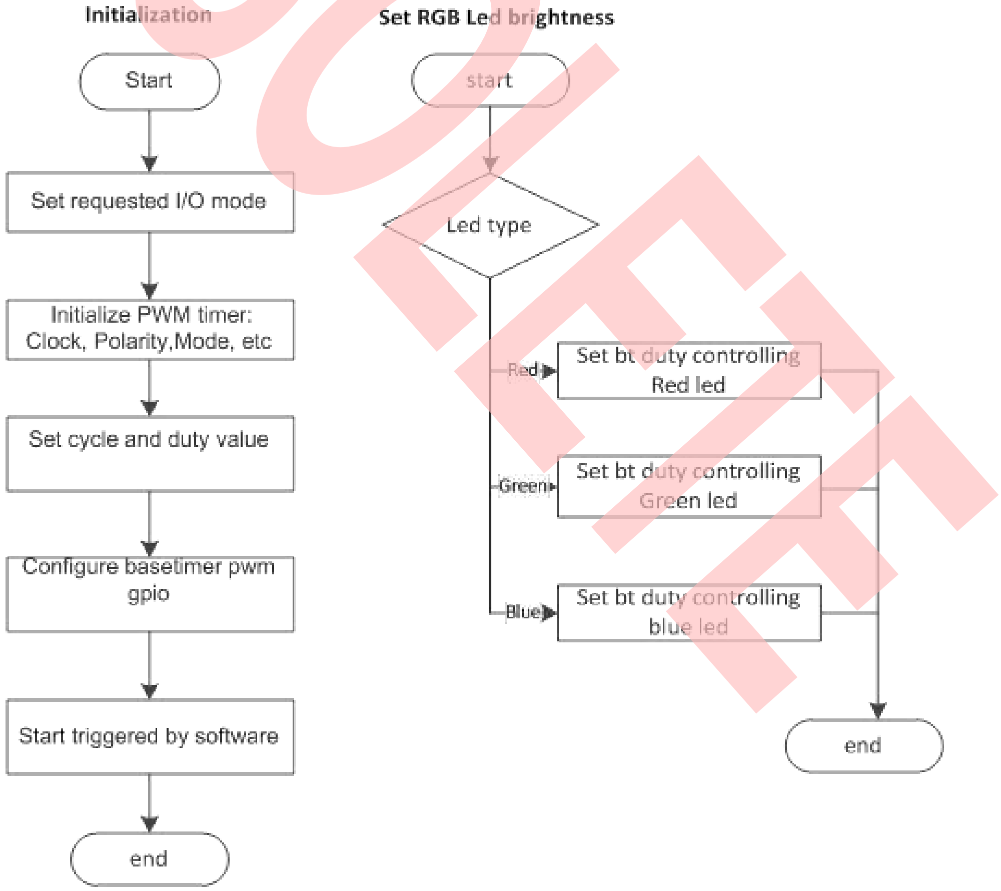
```
/* base timer - PWM channel configuration */
#define BT_LED_R            (BT3)
#define BT_LED_G            (BT4)
#define BT_LED_B            (BT5)
```

Figure 13. Define Requested BT Channels I/O Macros

```
/* base timer IO reallocation */
#define Bt_LedR_ConfigIO()   {SetPinFunc_TIOA3_1_OUT();}
#define Bt_LedG_ConfigIO()   {SetPinFunc_TIOA4_1_OUT();}
#define Bt_LedB_ConfigIO()   {SetPinFunc_TIOA5_1_OUT();}
```

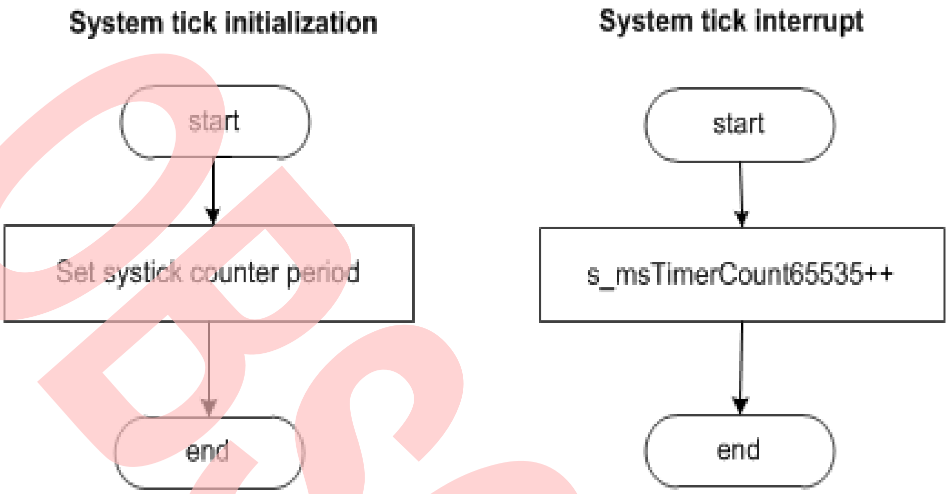
5. Code BT initialization function and setting RGB LED function (misc.c). The flow chart is as below

Figure 14. BT Initialization and Setting RGB LED Flow


Initialization
Set RGB Led brightness
Start
Set requested I/O mode
Initialize PWM timer: Clock, Polarity,Mode, etc
Set cycle and duty value
Configure basetimer pwm gpio
Start triggered by software
end
start
Led type
Red
Set bt duty controlling Red led
Green
Set bt duty controlling Green led
Blue
Set bt duty controlling blue led
end

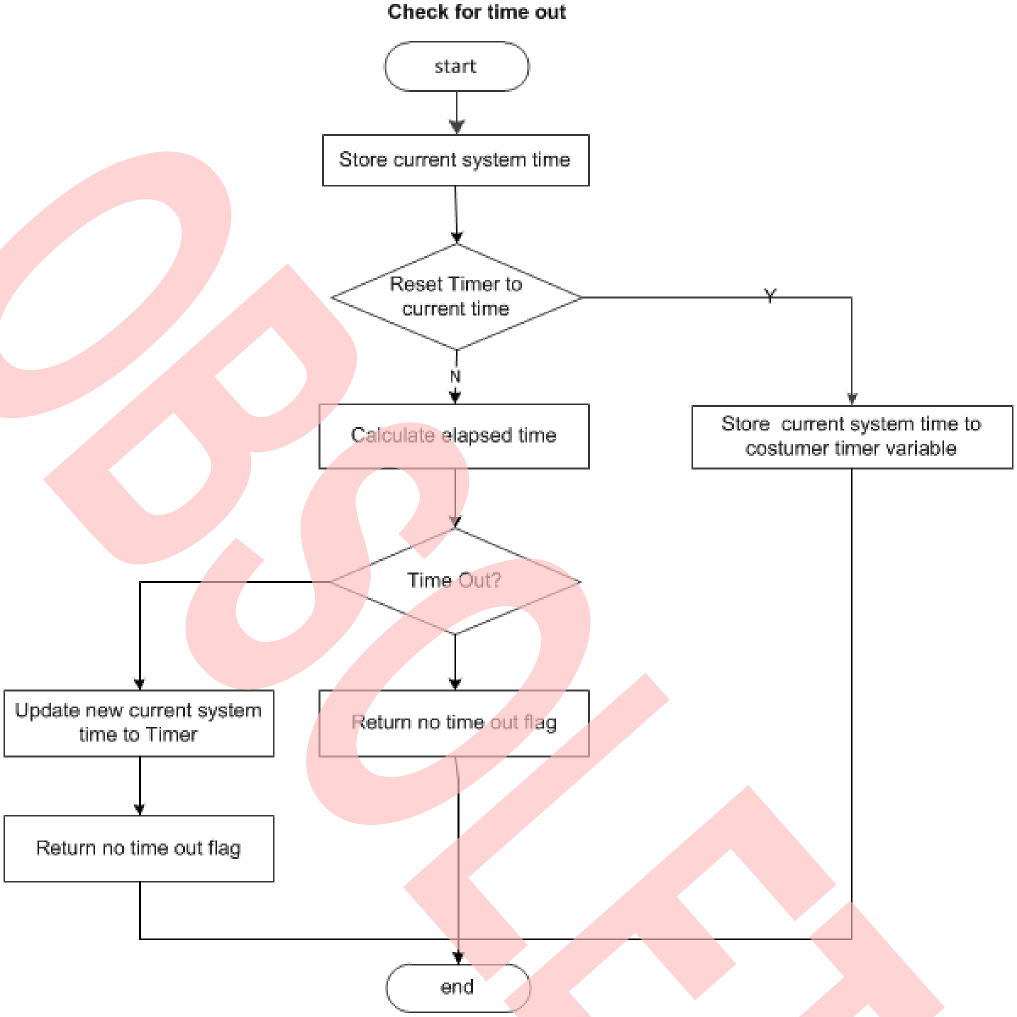
6. Create files sys_tick.c and sys_tick.h; add the new files to the project.
7. Code system tick initialization and interrupt function; code checking for timeout occurrence function (sys_tick.c).

Figure 15. BT Initialization and Setting RGB Led Flow



OBSOLETE

Figure 16. System Tick Timeout Function Flow


Check for time out
start
Store current system time
Reset Timer to current time
Y
N
Calculate elapsed time
Store current system time to costumer timer variable
Time Out?
Update new current system time to Timer
Return no time out flag
Return no time out flag
end

OBSOLETE

8. Create file main.c
   a. Initializes the System tick counter.
   b. Initializes the base timer for RGB LED driving.
   c. In the infinite loop
      i. RGB LED all light off when time out count equals 0.
      ii. Only red LED lights on when time out count equals 1.
      iii. Only green LED lights on when time out count equals 2.
      iv. Only blue LED lights on when time out count equals 3.
      v. Reset time out count when time out count is larger than 3.

Figure 17. Main Control Code

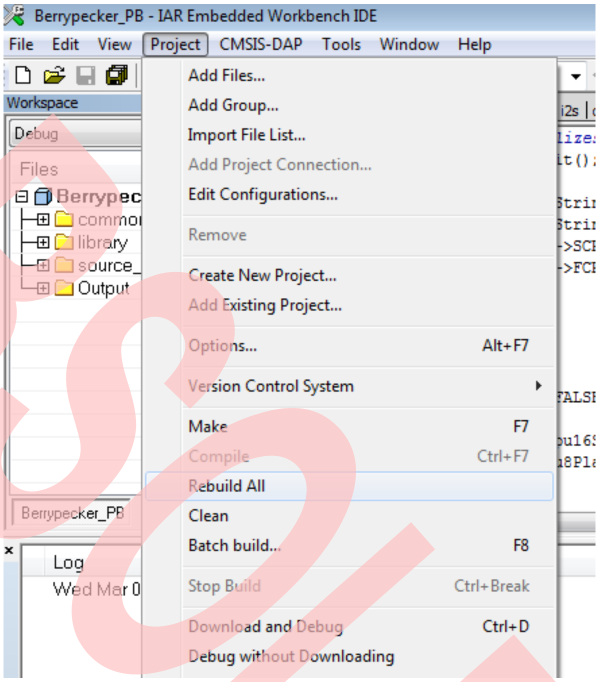
```
int32_t main(void)
{
    uint8_t   u8LedState = 0x00;
    uint16_t  RgbTimer;
    /* Initial system tick */
    SysTick_Init(100);
    /* Initial RGB */
    RgbLed_Init();
    TimerMax65535ms(&RgbTimer, 0x0);
    while (1)
    {
        switch(u8LedState)
        {
            case 0u:
            RgbLed_SetLumi(Led_Red, 100);    // LED R off
            RgbLed_SetLumi(Led_Green, 100);  // LED G off
            RgbLed_SetLumi(Led_Blue, 100);   // LED B off
            break;

            case 1u:
            RgbLed_SetLumi(Led_Red, 80);     // LED  R on
            RgbLed_SetLumi(Led_Green, 100);  // LED G off
            RgbLed_SetLumi(Led_Blue, 100);   // LED B off
            break;

            case 2u:
            RgbLed_SetLumi(Led_Red, 100);    // LED R off
            RgbLed_SetLumi(Led_Green, 80);  // LED G on
            RgbLed_SetLumi(Led_Blue, 100);  // LED B off
            break;

            case 3u:
            RgbLed_SetLumi(Led_Red, 100);     // LED R off
            RgbLed_SetLumi(Led_Green, 100);  // LED G off
            RgbLed_SetLumi(Led_Blue, 80);   // LED B on
            break;

            default:
                break;
        }
        if ( TIME_OUT_FLAG == TimerMax65535ms(&RgbTimer,100))
        {
            u8LedState++;
            if(u8LedState > 3)
              u8LedState = 0;
            TimerMax65535ms(&RgbTimer, 0x0);
        }
    }
}
```

OBSOLETE

OBSOLETE

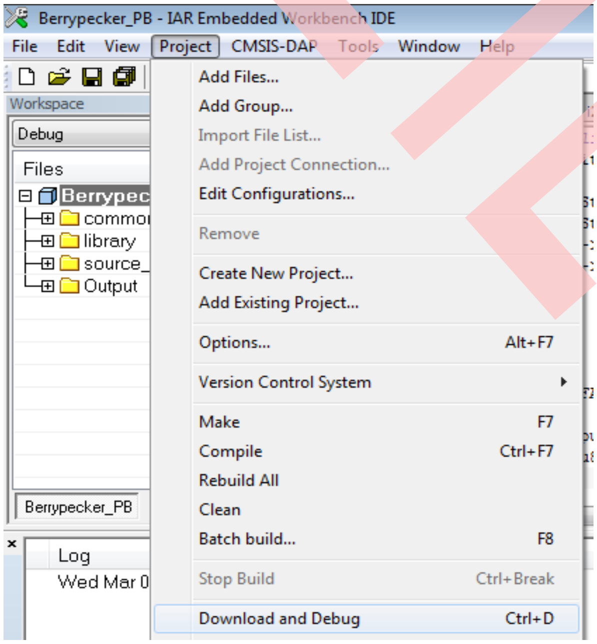
9. Connect CMSIS-DAP interface with PC to debug
10. Rebuild all.
    a. Click “**Project**”
    b. Select “**Rebuild All**”

Figure 18. Rebuild Project



11. Download and debug
    a. Click “Project”
    b. Select “Download and Debug”

Figure 19. Download and Debug



12. Watch RGB LED.
    a. RGB LED lights off.
    b. After some time, only red LED lights on.
    c. After some time, only green LED lights on.
    d. After some time, only blue LED lights on.
    e. After some time, turn to a

## 6.2 UART Receive/Send Data with CMSIS-DAP

### 6.2.1 Lab Objective

1. Understand multi-function serial interface module.
2. Study UART communication and program
3. Master the UART driver function about FM0+ PDL.

### 6.2.2 Lab Content

Receive the character from Cypress Serial Port Viewer and Terminal; and send to display on Terminal.

### 6.2.3 Preliminary Knowledge

1. Master the basic process of coding and debugging the program in the IAR integration development environment.
2. Understand FM0+ Peripheral Library framework.
3. Understand the serial bus.

### 6.2.4 Preparation of Lab Equipment and Tools

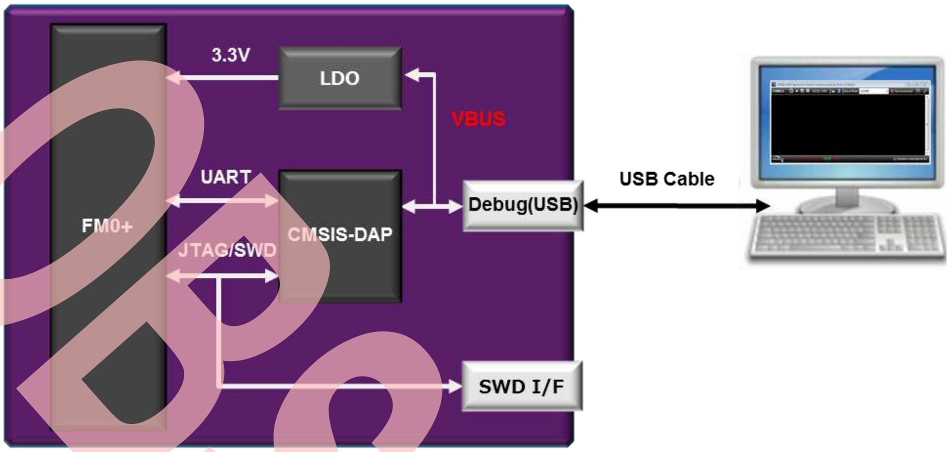
1. Hardware
    a. Mini USB cable
    b. SK-FM0-100L-S6E1B8 board
    c. PC
2. Software
    a. IAR
    b. Cypress Serial Port Viewer and Terminal

OBSOLETE

### 6.2.5 Lab Principle and Instructions

1. The Figure 20 shows that CMSIS-DAP module can receive/send characters from/to PC and then forwards data to FM0+ with UART communication

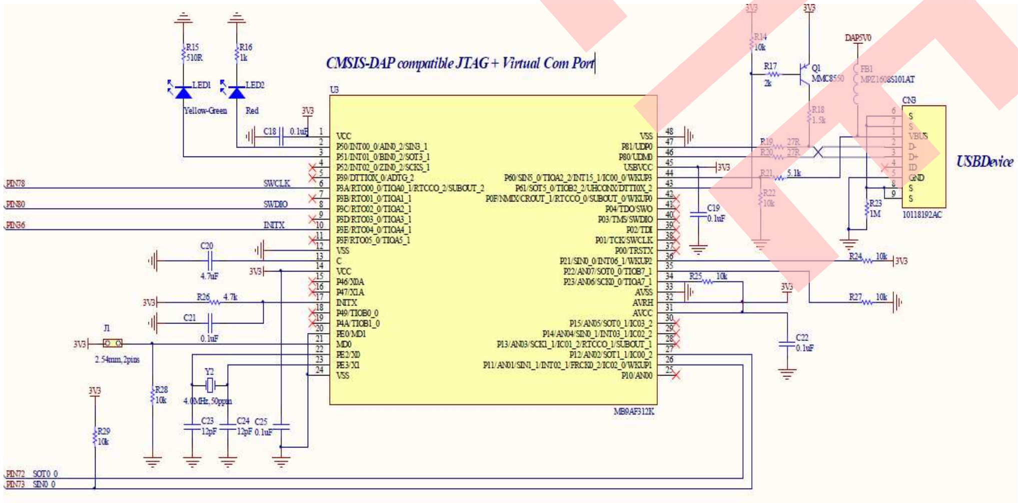
Figure 20. CMSIS-DAP Block



2. The multi-function serial interface has the following characteristics
   a. UART0 (Asynchronous normal serial interface)
   b. UART1 (Asynchronous multi-processor serial interface)
   c. CSIO (Clock synchronous serial interface) (SPI can be supported)
   d. LIN(LIN bus interface)
   e. I2C (I2C bus interface)

   Firstly, set multi-function serial interface mode to switch UART mode.
3. UART is a general-purpose serial data communications interface for asynchronous communications (start/stop synchronization) with external devices. It supports a bi-directional communications function (normal mode) and a master/slave type communications function (multi-processor mode: both master and slave modes supported). It also has transmitted/received FIFO (More details to refer to FM0+ Family Peripheral Manual Communication Macro Part).
4. The Figure 21 shows that SOT0_0 and SIN0_0 are selected to communicate with CMSIS-DAP.

Figure 21. CMSIS-DAP Compatible JTAG + Virtual Com Port Circuit



OBSOLETE

OBSOLETE

### 6.2.6 Lab Step

1. Open the project based on the last lab.
2. Edit file pdl_user.h to activate PDL resource and interrupt

Figure 22. Activate Requested MFS Channel

```
// Multi Function Serial Interfaces
#define PDL_PERIPHERAL_ENABLE_MFS0          PDL_ON
```

Figure 23. Activate Requested MFS Channel Interrupts

```
// Multi Function Serial Interfaces
#define PDL_INTERRUPT_ENABLE_MFS0           PDL_ON
```

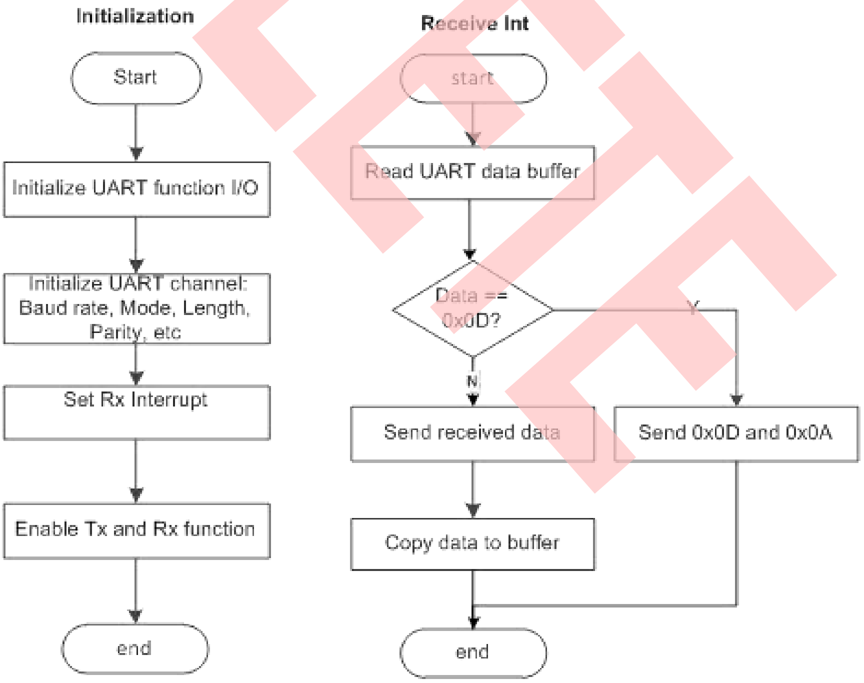
3. Create new file uart.c and uart.h; add the new files to the project.
4. Define macros about the requested UART I/O and channel(uart.c )

Figure 24. Define Requested UART Channel I/O Macros

```
#define InitUartIo()    {SetPinFunc_SIN0_0();SetPinFunc_SOT0_0();}
```

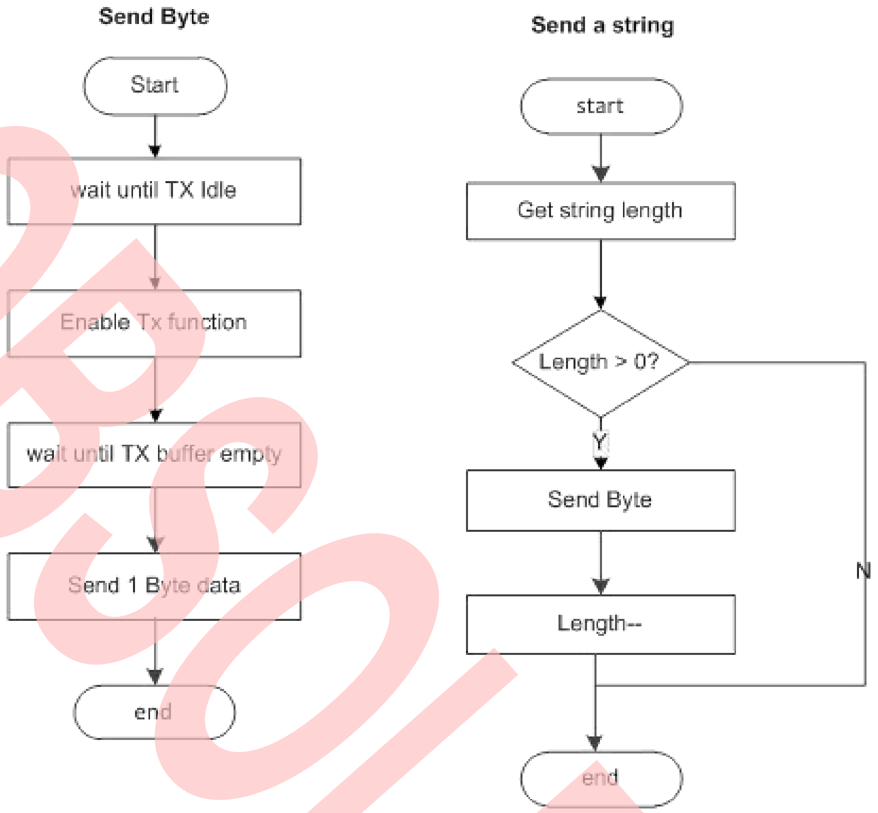
5. Code UART initialization function and Receive/Transmit function (uart.c). The flow chart is as below.

Figure 25. UART Initialization and Receive Interrupt Flow



OBSOLETE

Figure 26. UART Send Byte and String Flow



13. Create new file main.c to replace the old one.

a. Initializes Uart.

b. In the infinite loop

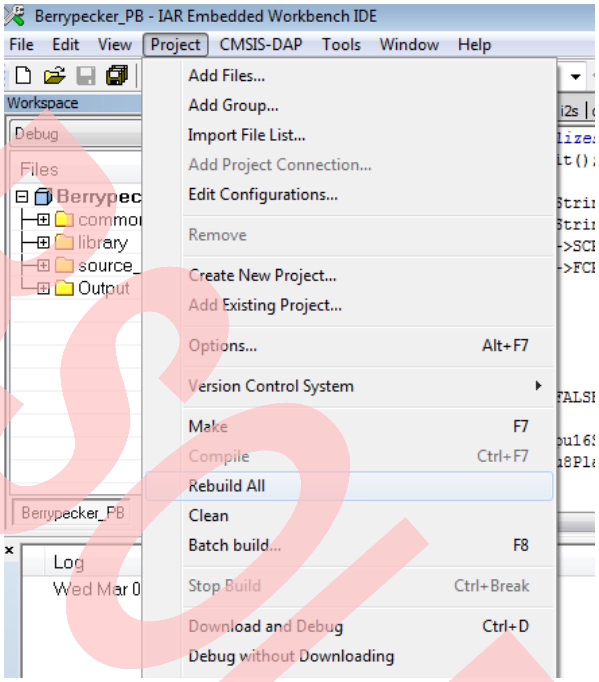
   i. If UART receiving counter overflow, reset counter.

Figure 27. Main Control Code

```
int32_t main(void)
{
    /* Initial uart function*/
    UartInit();

    while (1)
    {
     /* If UART receiving counter overflow, reset counter.*/
        if ( stcUartRxBuf.RxCnt >= (UARTMAXBUFFRTSIZE-1))
        {
          stcUartRxBuf.RxCnt = 0;
        }
    }
}
```

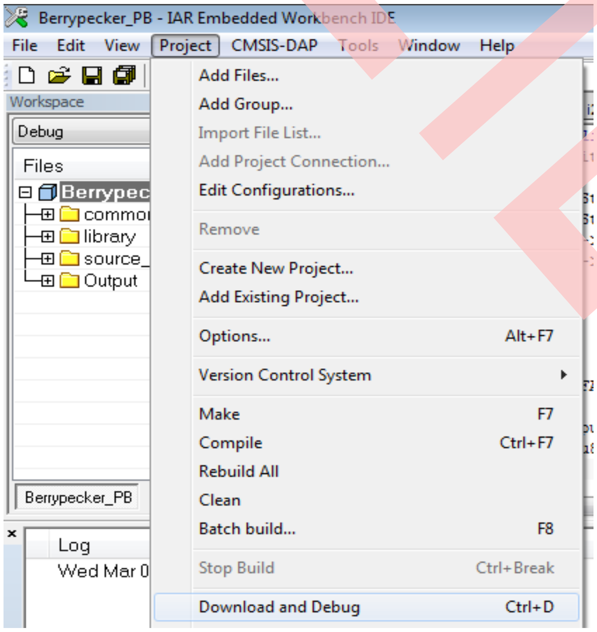
14. Connect CMSIS-DAP interface with PC to debug
15. Rebuild all.
    a. Click “**Project**”
    b. Select “**Rebuild All**”

Figure 28. Rebuild Project



16. Download and debug
    a. Click “**Project**”
    b. Select “**Download and Debug**

Figure 29. Download and Debug

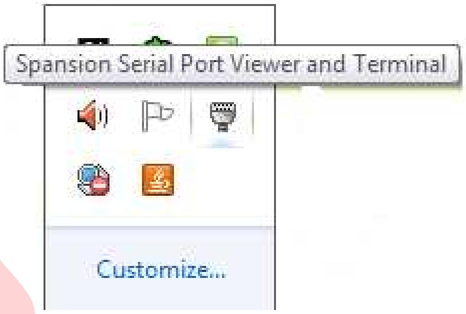


OBSOLETE

OBSOLETE

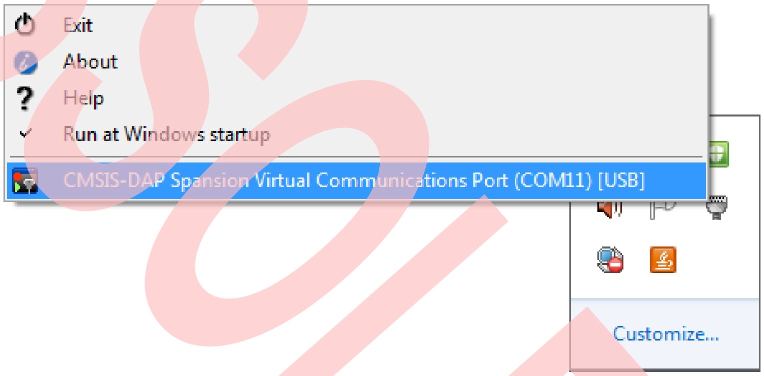
17. Open Cypress Serial Port Viewer and Terminal through icon in task menu or desktop shortcut

Figure 30. Cypress Serial Port Viewer and Terminal



18. Select the virtual COM of CMSIS-DAP

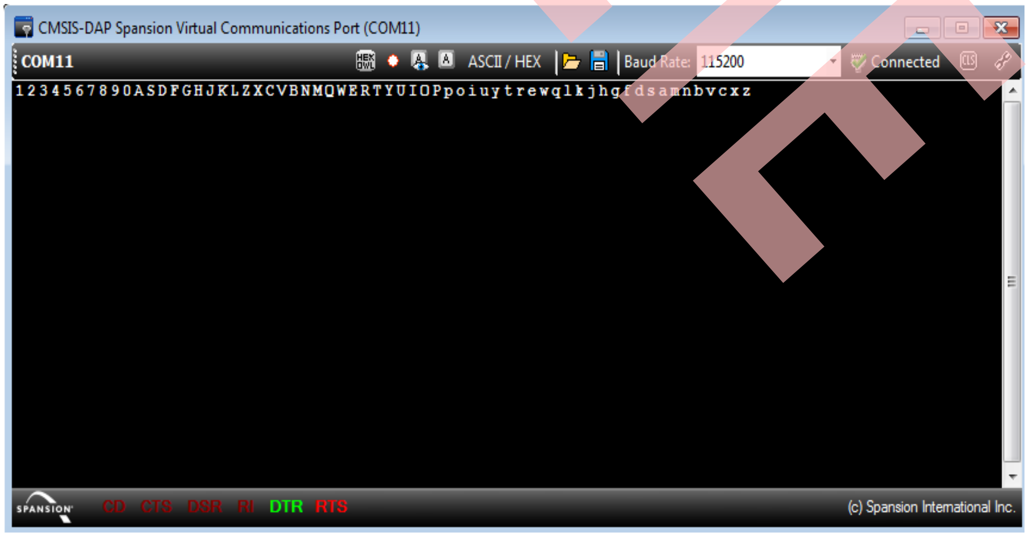
Figure 31. CMSIS-DAP Cypress Virtual Communications Port


Exit
About
Help
✓ Run at Windows startup
CMSIS-DAP Spansion Virtual Communications Port (COM11) [USB]
Customize...

19. Select the baud rate "115200", and Click the "disconnect" above the green arrow to connect

20. Click Cypress Serial Port Viewer and Terminal; press any key of PC Key Board, and check it

Figure 32. Cypress Serial Port Viewer and Terminal Display


CMSIS-DAP Spansion Virtual Communications Port (COM11)
COM11
ASCII / HEX
Connected
SPANSION
CD CTS DSR RI DTR RTS
(c) Spansion International Inc.

Baud Rate: 115200

```
1234567890ASDFGHJKLZXCVBNMQWERTYUIOPpoiuytrewqlkjhgfdsamnbvcxz
```