FM0+ S6E1C-Series

Starter Kit Guide

Doc. # 002-10542 Rev. *D

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
www.cypress.com

Copyrights

© Cypress Semiconductor Corporation, 2015-2017. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC (“Cypress”). This document, including any software or firmware included or referenced in this document (“Software”), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you under its copyright rights in the Software, a personal, non-exclusive, nontransferable license (without the right to sublicense) (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units. Cypress also grants you a personal, non-exclusive, nontransferable, license (without the right to sublicense) under those claims of Cypress’s patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely to the minimum extent that is necessary for you to exercise your rights under the copyright license granted in the previous sentence. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage (“Unintended Uses”). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and Company shall and hereby does release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. Company shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.

Contents

1. Introduction 4

1.1 Kit Contents 4
1.2 Board Details 5
1.3 Jumper and Connector 7
1.4 Getting Started 7
1.5 Additional Learning Resources 8
1.6 Technical Support 8
1.7 Acronyms 8

2. Installation and Test Operation 9

2.1 Install Software 9
2.2 Un-install Software 12
2.3 Test Operation 13

3. Hardware 18

3.1 System Block Diagram 18
3.2 Hardware Features 18
3.3 Hardware Details 19

4. Software Development 29

4.1 Tool Options 29
4.2 Example Projects 32
4.3 Flash Programming 44

A. Appendix 49

A.1 Schematic 49
A.2 Bill of Materials 53

Revision History 56

Document Revision History 56

1. Introduction

Thank you for your interest in the FM0+ S6E1C-Series Starter Kit. This kit enables customers to evaluate and develop projects using the FM0+ device family. FM0+ is a portfolio of ultra-low power ARM® Cortex®-M0+ MCUs. It is designed for low-power and cost-sensitive applications such as white goods, sensors, meters, HMI systems, power tools and internet of things (IoT) battery powered or energy harvesting wearable devices. There are several device series in this portfolio: the S6E1A-Series, the S6E1B-Series and the S6E1C-Series. This kit uses a device from the S6E1C-Series.

The S6E1C-Series microcontroller is based on the ARM® Cortex®-M0+ processor with on-chip flash memory. This series has peripherals such as High Definition Multimedia Interface-Consumer Electronics Control (HDMI-CEC), A/D converters, Descriptor System Data Transfer Controller (DTSC) and communication interfaces like USB, UART, CSIO (SPI), I2C and LIN. The FM0+S6E1C-Series Starter Kit offers footprint compatibility with Arduino™ shields, which provides options for application development.

1.1 Kit Contents

The FM0+ S6E1C-Series Starter Kit contains the following, as shown in Figure 1-1.

- FM0+ S6E1C-Series board
- USB Standard-A to Micro-B cable
- Quick Start Guide

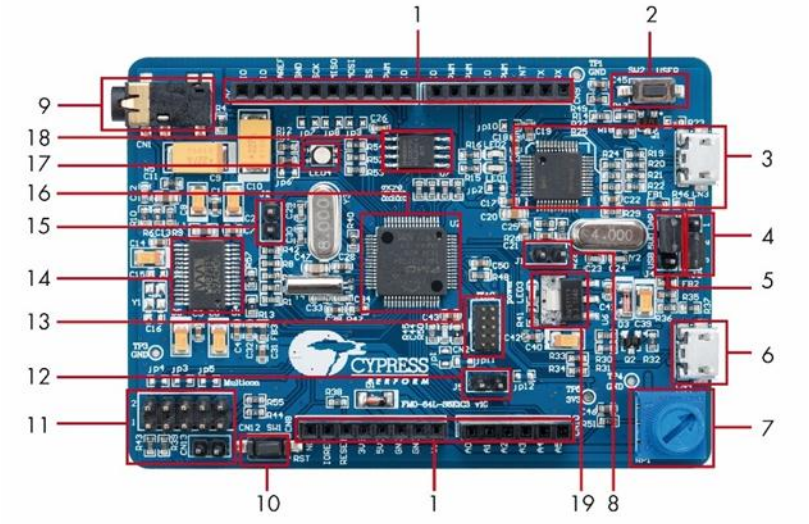
Figure 1-1: Kit Contents



Inspect the contents of the kit; if you find any part missing, contact your nearest Cypress sales office for help: www.cypress.com/support.

1.2 Board Details

Figure 1-2: FM0+ S6E1C-Series Starter Kit Markup



1. Arduino Interface (CN7-CN10)
2. User button
3. MB9AF312K programmer and debugger (CMSIS-DAP)
4. Serial programming mode select (J3)
5. Power supply resource select (J4)
6. USB device connector (CN4)
7. Potentiometer
8. MB9AF312K programming mode jumper (J1)

9. Headphone and microphone jack (CN1)
10. Reset button
11. Multicon connector (CN12, CN13)
12. Jumper for current measurement (J5)
13. 10-pin JTAG connector
14. Stereo codec
15. Programming mode jumper (J2)
16. Cypress FM0+ MCU S6E1C32D0A
17. RGB LED
18. NOR flash
19. 3.3 V voltage regulator

1.3 Jumper and Connector

Table 1-1: Jumper Description

Jumper	Function	Setting
J1	Sets the programming mode pin (MD0) of MB9AF312K (CMSIS-DAP)	Open: run mode
		Closed: serial programming mode
J2	Sets the programming mode pin (MD0) of S6E1C3	Open: run mode
		Closed: serial programming mode
J3	Serial programming mode Select for S6E1C3	Pin 2 to Pin 1: UART programming mode (not supported on this kit)
		Pin 2 to Pin 3: USB programming mode
J4	Power supply resource select	Pin 2 to Pin 1: power from CMSIS-DAP (CN3)
		Pin 2 to Pin 3: power from USB port of S6E1C3 (CN4)
J5	Jumper for power measurement	Open jp11 and connect a multi-meter with the terminals of J5 to measure the current that FM0+ MCU consumes.

Table 1-2: Connector Description

Connector	Description
CN1	Headphone and microphone jack
CN2	10-pin JTAG interface
CN3	USB port of CMSIS-DAP
CN4	USB port of FM0+ MCU
CN7,CN8,CN9,CN10	Arduino compatible headers
CN12,CN13	Multicon connector

1.4 Getting Started

This guide will help you get started with the FM0+S6E1C-Series Starter Kit:

- The Installation and Test Operation chapter describes the installation of the kit installer, and the test procedures for testing the board.
- The Hardware chapter describes the major features of the FM0+ S6E1C-Series Starter Kit and functionalities such as CMSIS-DAP debugger, Ethernet, USB, stereo codec, memories and sensors.
- The Software Development chapter describes how to open and run an example project in the IAR Embedded Workbench or Keil µVision IDE; it also describes how to use the example projects and how to program the devices using the USB DIRECT Programmer.
- The Appendix provides the kit schematics, and the bill of materials (BOM).

1.5 Additional Learning Resources

Cypress provides a wealth of data at www.cypress.com to help you to select the right MCU device for your design, and to quickly and effectively integrate the device into your design.

Visit the FM0+ product page for links to multiple resources, such as:

- Microcontroller Selector Guide
- Getting Started information
- Datasheets
- FM0+ Peripheral Manuals

1.6 Technical Support

For assistance, visit Cypress Support or contact customer support at +1(800) 541-4736 Ext. 2 (in the USA) or +1 (408) 943-2600 Ext. 2 (International).

1.7 Acronyms

Table 1-3. Acronyms Used in this Document

Acronym	Description
ADC	Analog-to-Digital Converter
CMSIS-DAP	Debug Access Port
DTSC	Descriptor System Data Transfer Controller
GPIO	General Purpose Input/output
HDMI-CEC	HighDefinition Multimedia Interface-Consumer Electronics Control
IDE	Integrated Development Environment
I2C	Inter-Integrated Circuit
I2S	Inter-IC Sound
JTAG	Joint Test Action Group
LDO	Low Drop Out (voltage regulator)
LED	Light-Emitting Diode
PDL	Peripheral Driver Library
PWM	Pulse Width Modulation
RGB	Red Green Blue
SWD	Serial Wire Debug
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus

2. Installation and Test Operation

This chapter describes the steps to install the software tools and drivers on a PC for using the FM0+ S6E1C-Series Starter Kit. After a successful installation, user can run the test code which is pre-programmed in the device.

2.1 Install Software

Follow the steps below to install the FM0+ S6E1C-Series Starter Kit software:

1. Download the FM0+ S6E1C-Series Starter Kit installer from the webpage:

   www.cypress.com/FM0-64L-S6E1C3.The Kit software is available for download in three formats.

   - **FM0+ S6E1C-Series Starter Kit Complete Setup**: This installation package contains the files related to the kit, including the Documentation, Hardware, Firmware, Software tools and drivers. However, it does not include the Windows Installer or Microsoft .NET framework packages. If these packages are not on your computer, the installer directs you to download and install them from the Internet.
   - **FM0+ S6E1C-Series Starter Kit Only Package:** This executable file installs only the kit contents, which include kit code examples, hardware files, and user documents. This package can be used if all the software prerequisites (listed in step 5) are installed on your PC.
   - **FM0+ S6E1C-Series Starter Kit DVD ISO**: This file is a complete package, stored in a DVD-ROM image format, which you can use to create a DVD or extract using an ISO extraction program such as WinZip® or WinRAR. The file can also be mounted like a virtual CD/DVD using virtual drive programs such as Virtual CloneDrive and MagicISO. This file includes all the required software, utilities, drivers, hardware files, and user documents.

2. If you have downloaded the ISO file, mount it in a virtual drive. Extract the ISO contents if you do not have a virtual drive to mount. Double-click **cyautorun.exe** in the root directory of the extracted content or the mounted ISO if “Autorun from CD/DVD” is not enabled on the PC. The installation window will appear automatically.

   Note: If you are using the “**Kit Complete Setup**” and “**Kit Only Package**”, then go to step 4 for installation.

3. Click **Install FM0-64L-S6E1C3** to start the kit installation, shown as Figure 2-1.

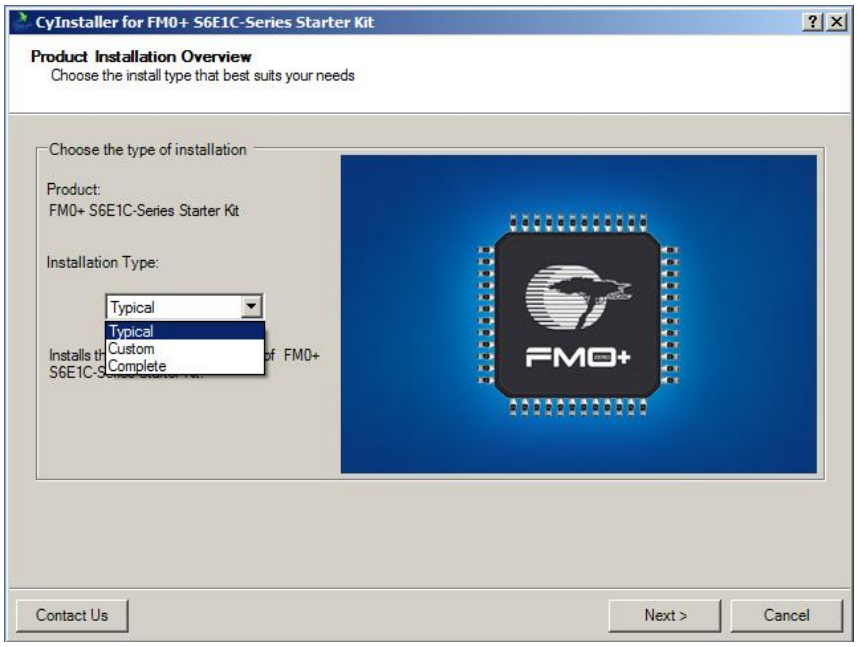
Figure 2-1: Kit Installation Window



4. Select the folder in which you want to install this package or use the default folder.
5. When you click **Next**, the FM0+ S6E1C-Series Starter Kit installer automatically installs the required software, if it is not present on your PC. Following are the required softwares and drivers:
   - FM Universal Peripheral Driver Library (PDL)
   - Serial Port Viewer
   - FLASH USB DIRECT Programmer
   - FLASH MCU Programmer
   - CMSIS-DAP driver

6. Choose the **Typical**, **Custom**, or **Complete** installation type (select '**Typical**' if you do not know which one to select) in the Product Installation Overview window, as shown in Figure 2-2. Click **Next** after you select the installation type.

Figure 2-2: Produce Installation Overview

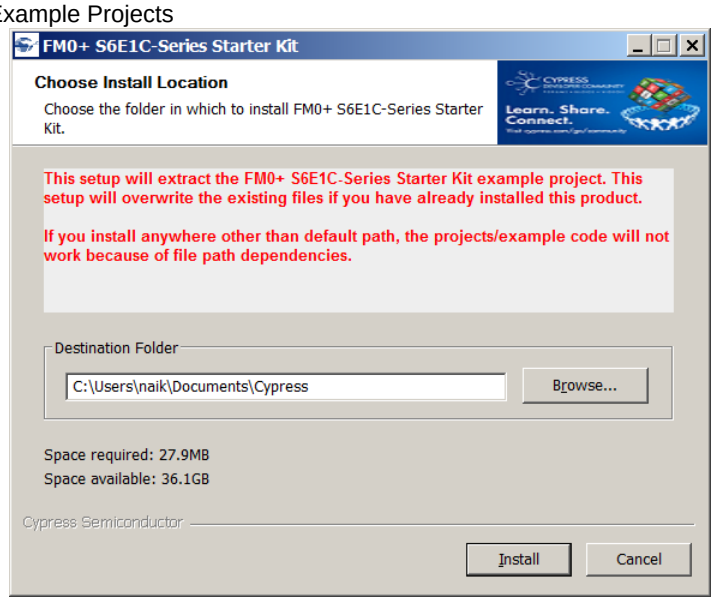


7. Read the License agreement and select **I accept the terms in the license agreement** to continue with installation. Then, click **Next**.
8. When the installation begins, a list of packages appears on the installation page. A green check mark appears next to each package after successful installation.
9. If you are an un-registered user either enter your contact information or select the check box **Continue without Contact Information**. If you are a registered user, then the installation procedure will not request you to enter the contact information. Click **Finish** to complete the kit installation.

   Note: Be sure to select the check boxes **Extract Example Projects** and **Extract FM PDL 2.0**.

10. Click **Install**.

Figure 2-3: Extract the Example Projects



11. Click **Close** to finish the extraction.

After the installation is complete, the kit contents are available at the following location:

Default location:

```
Windows OS (64-bit): C:\Program Files (x86)\Cypress
\FM0+ S6E1C-Series Starter Kit
Windows OS (32-bit): C:\Program Files\Cypress
\FM0+ S6E1C-Series Starter Kit
```

The Peripheral Driver Library (PDL) will be extracted to this default directory:

```
C:\Users\<User Name>\My Documents\Cypress\FM_PDL_2.0.1
```

And, the kit example projects will be extracted to this directory:

```
C:\Users\<User Name>\My Documents\Cypress
\FM0+ S6E1C-Series Starter Kit_Ver01
```

In the rest of the document, the following directory is termed as <User_Directory>:

```
C:\Users\<User Name>\My Documents\Cypress\
```

2.2 Un-install Software

The software can be uninstalled using one of the following methods:

- Go to **Start** ->**All Programs** ->**Cypress** ->**Cypress Update Manager** and select the **Uninstall** button that corresponds to the kit software.
- Go to **Start** ->**Control Panel** ->**Programs and Features for Windows 7** or **Add/Remove Programs for Windows XP**; select the **Uninstall/Change** button.

Note: Uninstalling the Kit software will not remove the PDL and Example Projects from <User_Directory>.

2.3 Test Operation

The FM0+ S6E1C-Series Starter Kit has been pre-programmed with a test demo code, which helps to test all of the on-board features. The Motorola-S format file, *tp_fm0-64l-s6e1c3.srec*, is provided in the following directory and can be programmed on the MCU by using the FLASH USB DIRECT Programmer.

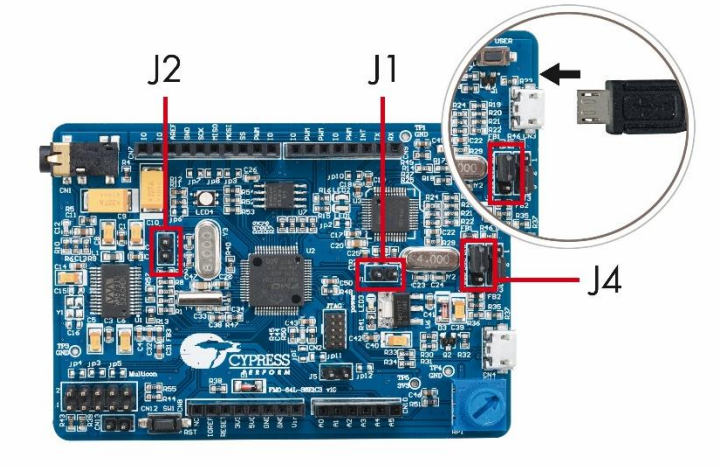
`<User_Directory>\FM0+ S6E1C-Series Starter Kit_Ver01\Firmware\Demo Projects\Test_Demo_Code`

2.3.1 Run the Test Demo

Follow the instructions to run the test code.

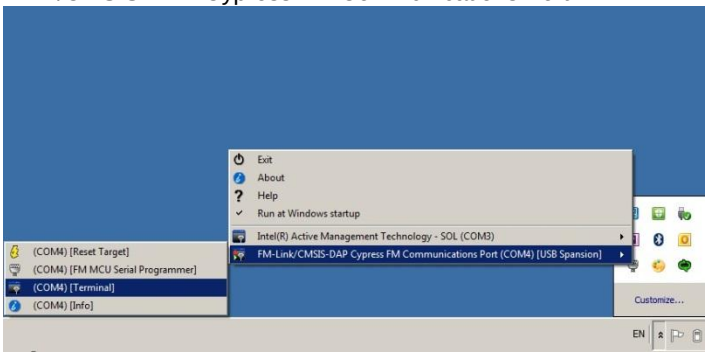
1. Ensure the jumpers J1 and J2 are open. Close Pin 1 and Pin 2 of J4 (default) and connect CN3 to a PC using the USB cable provided.

Figure 2-4: Power the Board from CN3



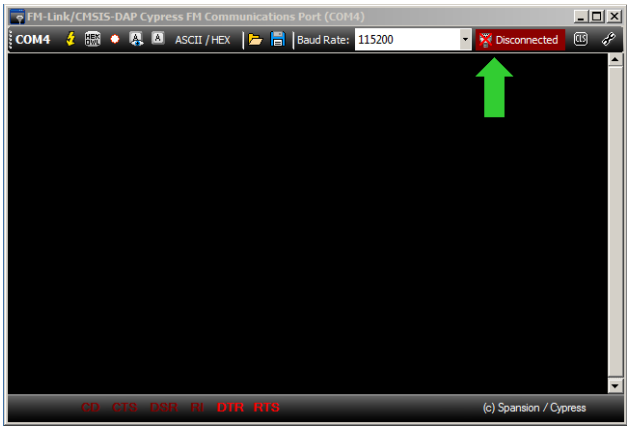
2. If not already launched, then launch the **Serial Port Viewer** from the start menu.
3. Click the Serial Port Viewer icon in the task bar and select **FM-Link/CMSIS-DAP Cypress FM Communications Port.**

Figure 2-5: FM-Link/CMSIS-DAP Cypress FM Communications Port



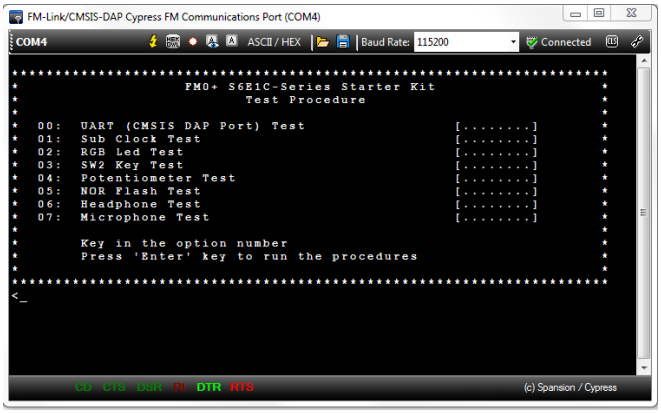
4. Select the baud rate **115200**, and click the **Disconnected** button to connect to the board.

Figure 2-6: Select the Baud Rate



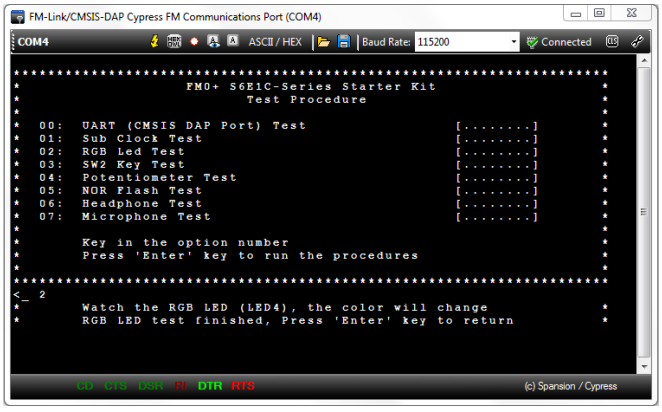
5. Press the **Enter** key on your keyboard to run the test procedure. Key in the option number and press the **Enter** key to run the test.

Figure 2-7: Test Procedure



6. For example, key in **2** and press the **Enter** key on the PC keyboard. The color of the RGB LED will change from red to green to blue.

Figure 2-8: RGB LED Test

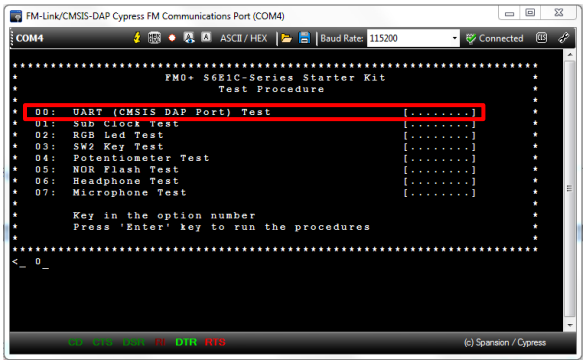


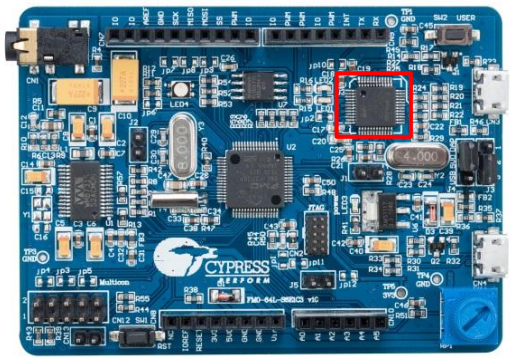
2.3.2 *Test Procedure Explanation*

This section explains the test procedure. This test procedure is based on the Serial Port Viewer. User has to key-in the test procedure number displayed on the menu and then press the **Enter** key. The firmware on the board will run the test procedure and display the results. There are eight test procedures and a brief description of each test procedure is given below:

- **UART (CMSIS DAP Port) Test:** This procedure will test the UART communication between CMSIS-DAP and S6E1C3 MCU. Key in **0** and press the **Enter** key.

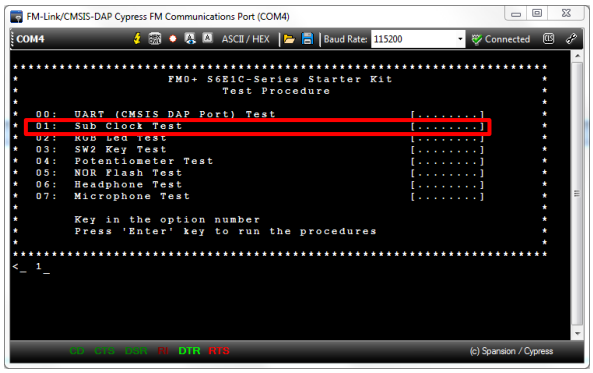
Figure 2-9: UART Test

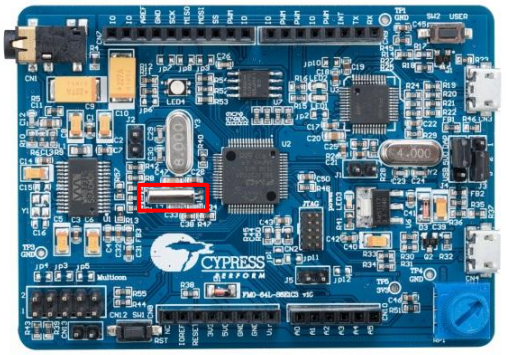




- **Sub Clock Test**: This procedure will test whether the sub clock is at 32.768 KHz. Key in **1** and press the **Enter** key, the main routine will shift the system clock to sub clock, and then shift back to main clock. It displays **OK** if the sub clock runs at 32.768 KHz, otherwise it will display **Fail.**

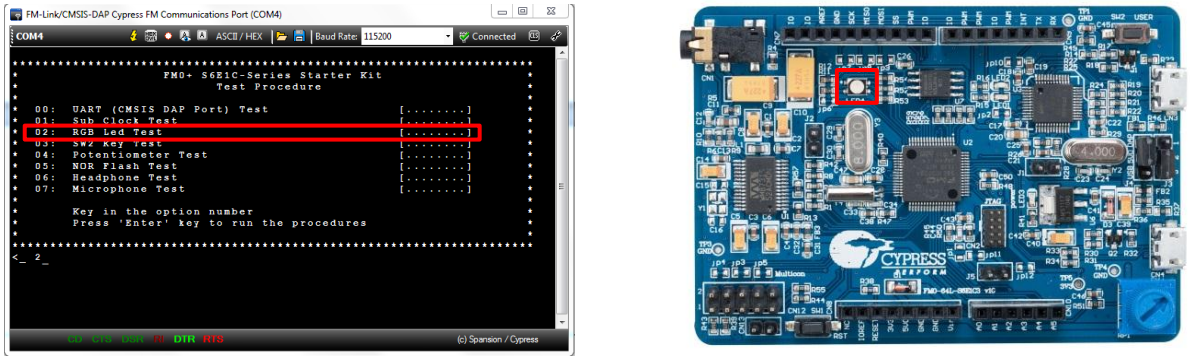
Figure 2-10: Sub Clock Test





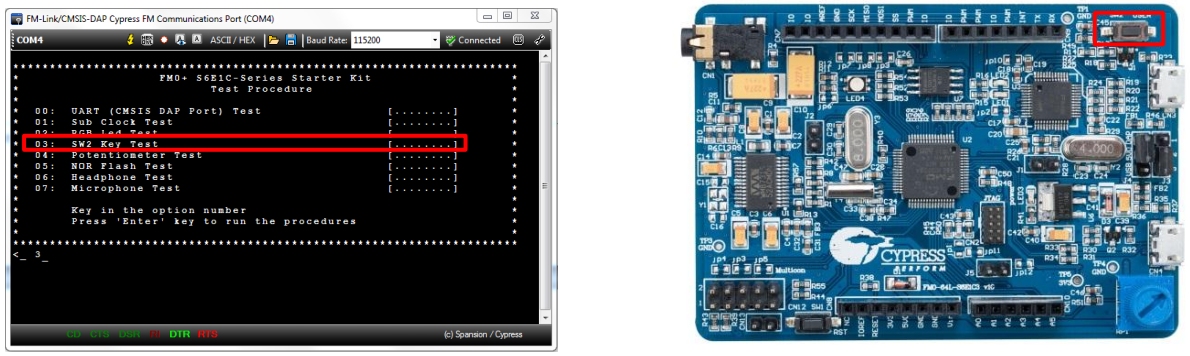
- **RGB LED Test**: This procedure tests the RGB LED. Key in **2** and press the **Enter** key, the RGB LED color will change from red to green to blue.

Figure 2-11: RGB LED Test



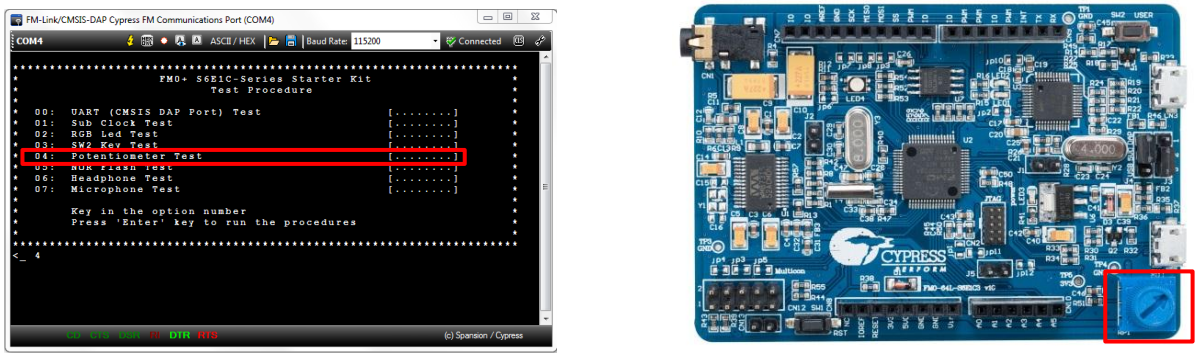
- **SW2 Key Test**: This procedure tests the switch SW2. Key in **3** and press the **Enter** key, the test routine will detect the press and release status of the SW2 switch.

Figure 2-12: SW2 Key Test



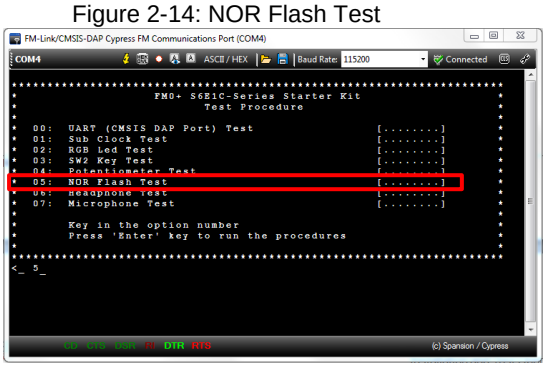
- **Potentiometer Test**: This procedure tests the potentiometer. Key in **4** and press the **Enter** key. The ADC value will be displayed on the terminal. Turn the potentiometer (RP1), the ADC value will change accordingly.

Figure 2-13: Potentiometer Test



- **NOR Flash Test**: This procedure tests the NOR flash. Key in **5** and press the **Enter** key, the code will write a pre-determined set of data into the flash and then reads and compares to check whether the data is the same. If it is same**,** then it will display **OK**, otherwise it will display **Fail.**

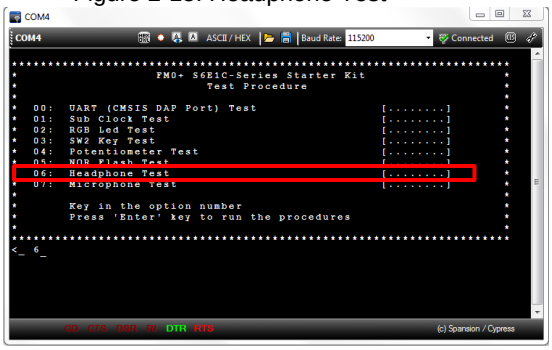
Figure 2-14: NOR Flash Test

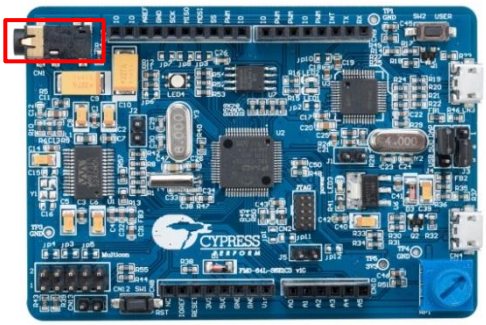




- **Headphone Test**: This procedure tests the headphone channel of the stereo codec. Key in **6** and press the **Enter** key, a pre-determined calling bell sound will be heard on the headphone connected to **CN1**.

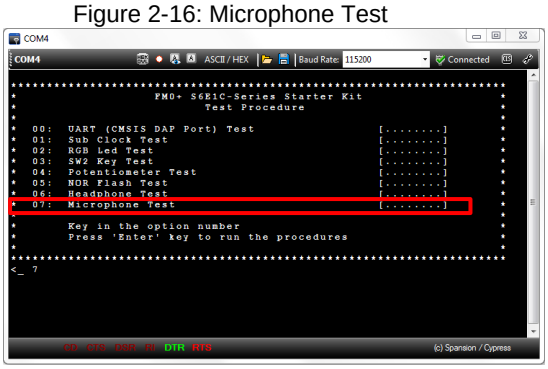
Figure 2-15: Headphone Test

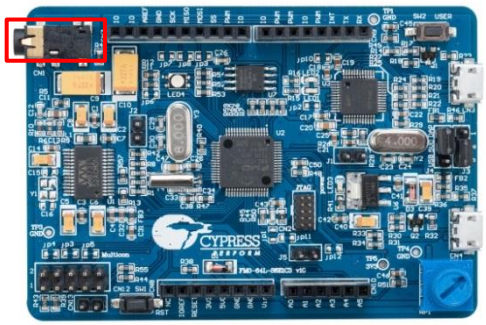




- **Microphone Test**: This procedure is to test the microphone channel of the stereo codec. Key in **7** and press the **Enter** key, you can hear your voice from the microphone on the headphone.

Figure 2-16: Microphone Test





Note: Connect a headset with a 4-conductor phone plug (American Headset Jack) to the CN1.

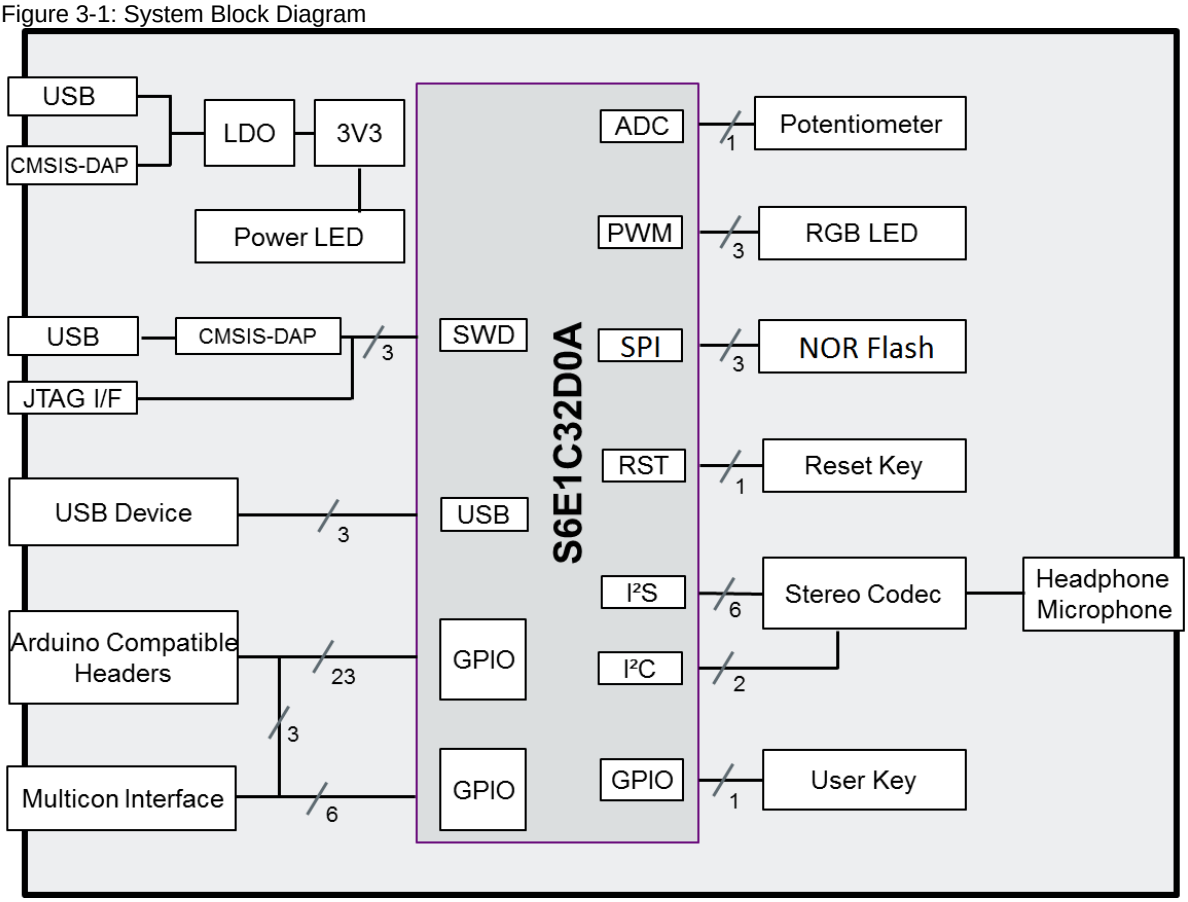
3. Hardware

This chapter describes the features and hardware details of the FM0+ S6E1C-Series Starter Kit.

3.1 System Block Diagram

Figure 3-1 shows the system block diagram of the FM0+ S6E1C-Series Starter Kit.

Figure 3-1: System Block Diagram



3.2 Hardware Features

- Cypress FM0+ S6E1C-Series MCU
- On-board ICE (CMSIS-DAP compatible)
- USB device interface
- Potentiometer

- NOR flash memory
- Stereo codec
- RGB LED
- User button
- Arduino compatible headers
- Multicon interface

3.3 Hardware Details

3.3.1 FM0+Series MCU

The S6E1C-Series Starter Kit features an ultra-low-power, highly integrated S6E1C3 MCU, a 32-bit ARM® Cortex®-M0+ MCU.

The S6E1C3 MCU is a member of the S6E1C-Series device family with 40MHz CPU, 128KB flash, 16KB SRAM and 54 GPIOs. The S6E1C3 MCU features a wide variety of peripherals such as timers, 12-bit SAR ADC and communication interfaces like USB (host and device), CSIO (SPI), I2S, I2C (slave only) and UART.

3.3.2 User Button and LED

The FM0+ S6E1C-Series Starter Kit features a user button and an RGB LED. The switch and the LED are connected to the S6E1C3 MCU via pins mentioned in Table 3-1.

Table 3-1: Button and LED

Pin No.	Port	External device
5	P30	SW2
14	P3D	LED4 – Red
15	P3E	LED4 – Green
16	P3F	LED4 – Blue

The port P3D/P3E/P3F are also assigned as the PWM output pins, and user can dim the LED by configuring the base timers in PWM mode to output PWM signals from the pins.

3.3.3 Arduino Compatible Interface

The FM0+ S6E1C-Series Starter Kit provides footprint compatibility with the Arduino interface. These headers expand the possibility for users to develop more applications based on this kit and different Arduino compatible shields. Figure 3-2 shows the pins.

Figure 3-2: Pins of Arduino Compatible Interface

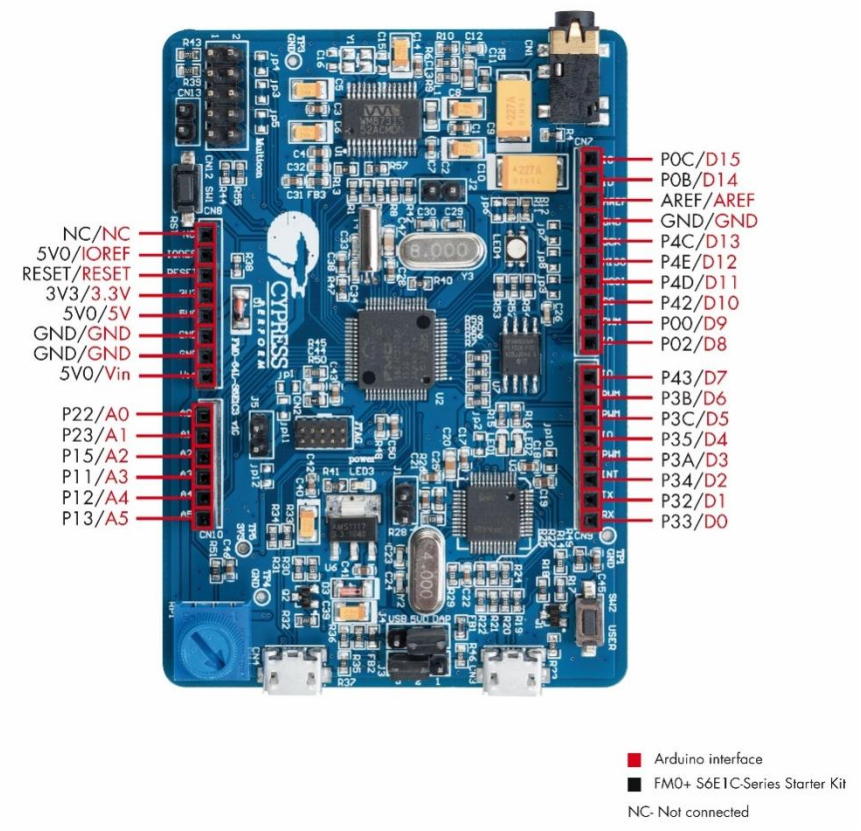


Table 3-2 shows full definitions of the pins connected with the Arduino headers.

Table 3-2: Pins of Arduino-Compatible Interface

Pin No.	Pin Name	Arduino Designation	Function (Part)
8	P33/ADTG_6/SIN6_1/INT04_0/MI2SDI6_1	D0	UART – RX
7	P32/SOT6_1/SI2CSDA6_1/TIOB2_1/INT05_2/MI2SDO6_1	D1	UART – TX
9	P34/SCS61_1/TIOB4_1/MI2SMCK6_1	D2	IO
11	P3A/TIOA0_1/INT03_0/RTCCO_2/SUBOUT_2/IC1_CIN_0	D3	PWM
10	P35/SCS62_1/TIOB5_1/INT08_1	D4	IO
12	P3B/TIOA1_1/IC1_DATA_0	D5	PWM
13	P3C/TIOA2_1/IC1_RST_0	D6	PWM
23	P43/ADTG_7/TIOA3_0	D7	IO
54	P02/WKUP5	D8	IO
52	P00/WKUP4	D9	IO
22	P42/TIOA2_0	D10	SPI -- #CS
25	P4D/SOT7_1	D11	SPI -- MOSI
26	P4E/SIN7_1/INT06_2	D12	SPI -- MISO
24	P4C/SCK7_1/TIOB3_0	D13	SPI -- CLK
62	P0B/TIOB6_1/WKUP6	D14	IO
63	P0C/TIOA6_1/WKUP7	D15	IO
47	P22/AN07/TIOB7_1	A0	AN7
46	P23/AN06/SCK0_1/TIOA7_1	A1	AN6
45	P15/AN05/SOT0_1/SCS11_1	A2	AN5
41	P11/AN01/SIN1_1/INT02_1/WKUP1	A3	AN1
42	P12/AN02/SOT1_1	A4	AN2/I2C--SDA
43	P13/AN03/SCK1_1/RTCCO_1/SUBOUT_1	A5	AN3/I2C--SCL
32	INITX	RESET	RESET

3.3.4 Stereo Codec

WM8731 is a low power stereo codec with an integrated headphone driver. On the digital side, it has an Inter-IC Sound (I2S) interface which is connected with the I2S macro. Table 3-3 explains the details of the signals. It also has an I2C interface for configuring the device and an I2S interface for audio data transmission. The I2C address for configuring the stereo codec is 0x1A.

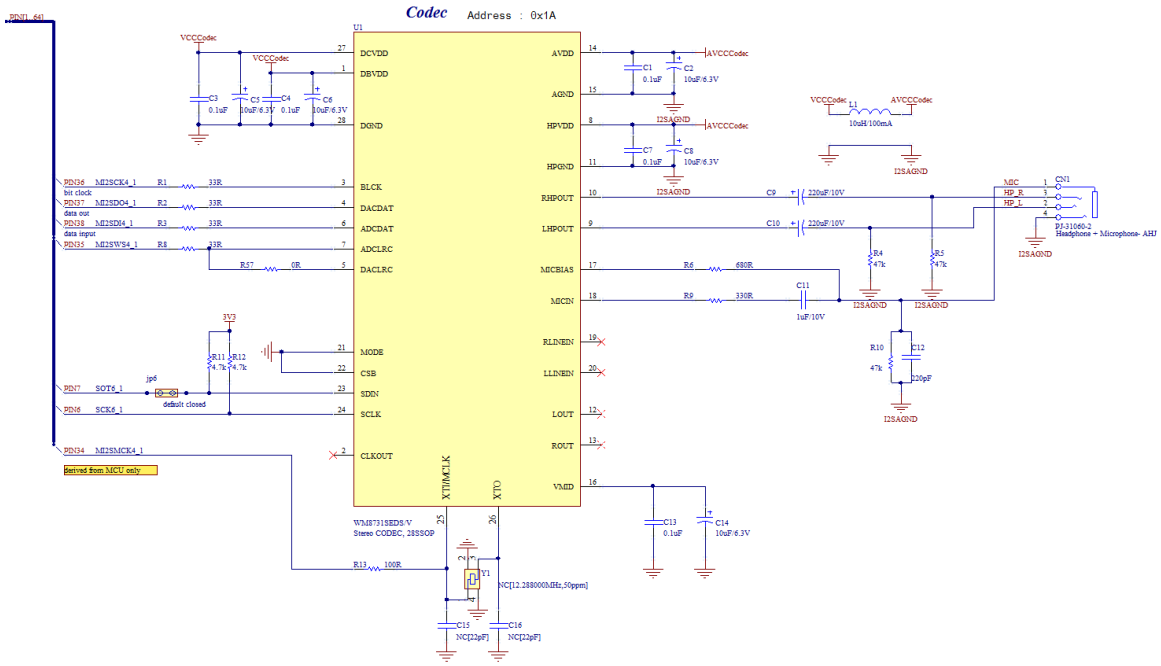
On the analog side, the codec has headphone, microphone, line-in and line-out channel. On this kit, only headphone and microphone are routed to jack, CN1.

Pin 7 is a multiplex pin which is connected to Pin 23 of the codec by a default closed wire jumper, jp6. See Multiplex Pins for details.

Table 3-3: I2S Interface

Pin No.	Port	Function
36	MI2SCK4_1	Bit clock out
37	MI2SDO4_1	Audio data out
38	MI2SDI4_1	Audio data in
35	MI2SWS4_1	Word select output pin
34	MI2SMCK4_1	Mast clock I/O pin

Figure 3-3: Stereo Codec Circuit



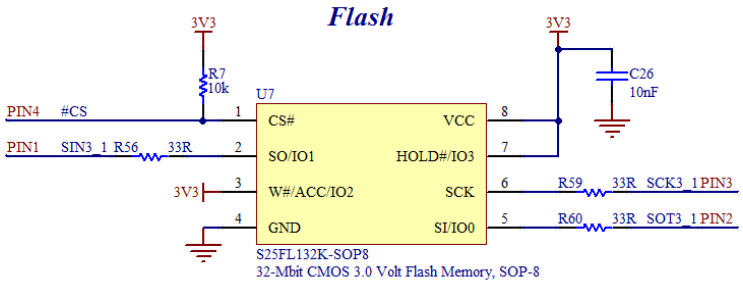
3.3.5 NOR Flash

The S25FL132K non-volatile flash memory device connects to a host system via a Serial Peripheral Interface (SPI). The NOR Flash supports SPI single bit serial input and output (single I/O or SIO) as well as optional two bit (Dual I/O or DIO) and four bit (Quad I/O or QIO) serial protocols. This multiple width interface is called SPI Multi-I/O or MIO. The 33R resistors are used to reduce EMI emissions.

Table 3-4: NOR Flash Interface

Pin No.	Port	Function
1	P50/SIN3_1/INT00_0	SPI data in
2	P51/SOT3_1/INT01_0	SPI data out
3	P52/SCK3_1/INT02_0	SPI clock out
4	P53/TIOA1_2/INT07_2	Chip Select

Figure 3-4: NOR Flash Circuit



3.3.6 Multicon interface

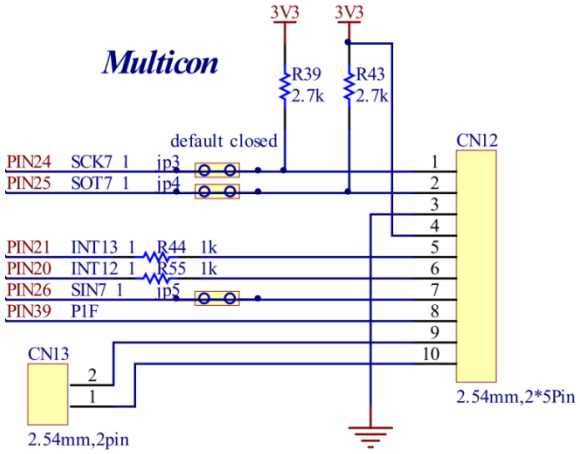
The Multicon interface, CN12, is a 2x5 pin connector interface that brings the appropriate signals for UART, SPI, I2C and external interrupts. A simple 10-pin cable can be used to route the serial signals and power to a secondary board or another system. Pins 9 and 10 connect to an additional connector (CN13) to provide another connection point.

Pin 24, Pin 25, Pin 26 are multiplexed pins which are connected to CN12 by default via wire jumpers jp3, jp4 and jp5. See Multiplex Pins for details.

Table 3-5: Serial Ports in Multicon Interface

Pin No.	Pin	Port	UART	SPI	I2C	Interrupt
CN12_1	Pin24	SCK7_1		CLK	SCL	
CN12_2	Pin25	SOT7_1	TX	MOSI	SDA	
CN12_3	GND					
CN12_4	3V3					
CN12_5	Pin21	INT13_1				INT13_1
CN12_6	Pin20	INT12_1				INT12_1
CN12_7	Pin26	SIN7_1	RX	MISO		
CN12_8	Pin39	PIF		#CS		
CN12_9	CN13_2					
CN12_10	CN13_1					

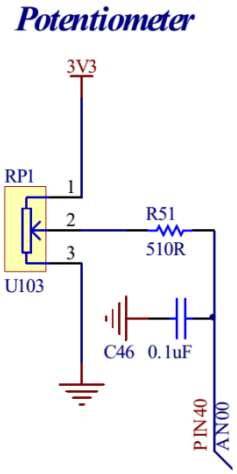
Figure 3-5: Multicon Interface



3.3.7 Potentiometer

The FM0+ S6E1C-Series Starter kit provides a potentiometer. Its resistance value ranges from 0 to 10k.The middle terminal is connected to the ADC channel AN00 (Pin40).

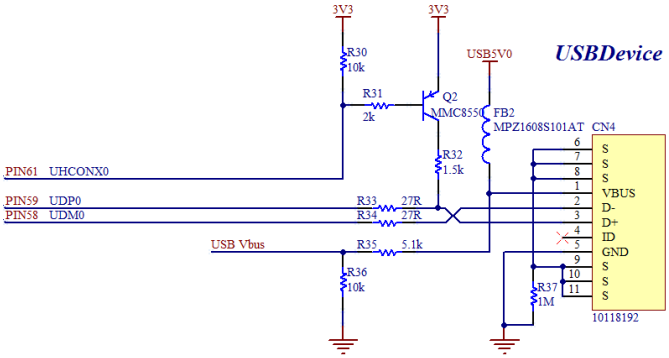
Figure 3-6: Potentiometer



3.3.8 USB Interface

The FM0+ S6E1C3 MCU has a USB channel that can work as a host or a device. In the FM0+ S6E1C-Series Starter Kit, the USB port is configured as a USB device and is connected to CN4, a micro USB type B connector.

Figure 3-7: USB Device

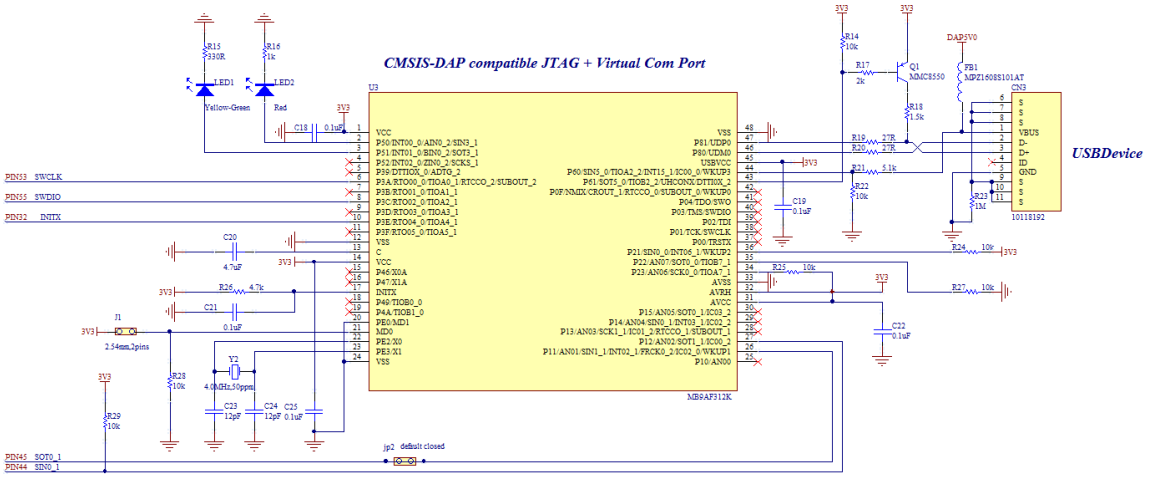


3.3.9 CMSIS-DAP

The FM0+ S6E1C-Series Starter Kit features an on-board CMSIS-DAP module to enable programming and debugging of the FM0+ S6E1C3 MCU. The CMSIS-DAP firmware solution supports full JTAG configuration and two-wire Serial Wire Debug (SWD) interface. On this kit the CMSIS-DAP supports SWD only as the FM0+ S6E1C3 MCU has just the SWD interface. The CMSIS-DAP module can also power the FM0+S6E1C-Series Starter kit via the CN3 connector, when pin 1 and pin 2 of Jumper J4 are shorted.

Pin 45 is a multiplexed pin, it is connected to Pin 26 of MB9AF312K by default via Jumper jp2. See Multiplex Pins for details.

Figure 3-8: CMSIS-DAP Circuit

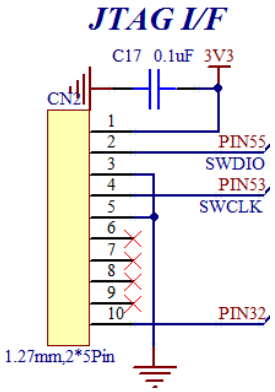


Note: The SOT1_1 and SIN1_1 are connected to S6E1C3 SIN0_1 and SOT0_1 respectively, which enables the UART communication between S6E1C3 and CMSIS-DAP. However this pin arrangement does not support serial programming of the S6E1C3 using the FLASH MCU Programmer on this board.

3.3.10 JTAG

The FM0+ S6E1C-Series Starter Kit provides an interface, CN2, to connect an external programmer and debugger for programming and debugging the FM0+ S6E1C3 MCU. CN2 is a standard ARM 0.05'' 10-pin Cortex debug header.

Figure 3-9: 10-pin JTAG Interface



3.3.11 Multiplex Pins

This board has some multiplexed pins. They have connections with different peripherals or headers using wire jumpers. User can select the connection by opening or closing the wire jumpers on the board. The wire jumper's footprint is compatible with 0603 footprint. It is easy to close them by soldering a 0R 0603 resistor on it.

Table 3-6 shows the wire jumper information.

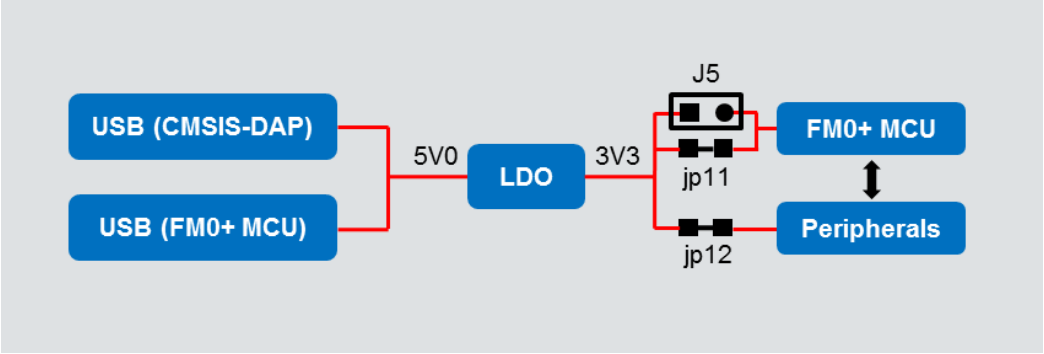
Table 3-6: Wire Jumpers for Multiplex Pins

Wire Jumper	Function
jp1 (default open)	Connect MCU pin45 to pin3 of CN10
jp2 (default closed)	Connect MCU pin45 to pin 26 of MB9AF312K
jp3 (default open)	Connect MCU pin24 to pin1 of Multicon interface (CN12)
jp7 (default closed)	Connect MCU pin24 to pin6 of CN7
jp4 (default open)	Connect MCU pin25 to pin2 of Multicon interface (CN12)
jp9 (default closed)	Connect MCU pin25 to pin4 of CN7
jp5 (default open)	Connect MCU pin26 to pin7 of Multicon interface (CN12)
jp8 (default closed)	Connect MCU pin26 to pin5 of CN7
jp6 (default closed)	Connect MCU pin7 to pin23 of stereo codec
jp10 (default open)	Connect MCU pin7 to pin2 of CN9

3.3.12 Jumpers for Current Measurement

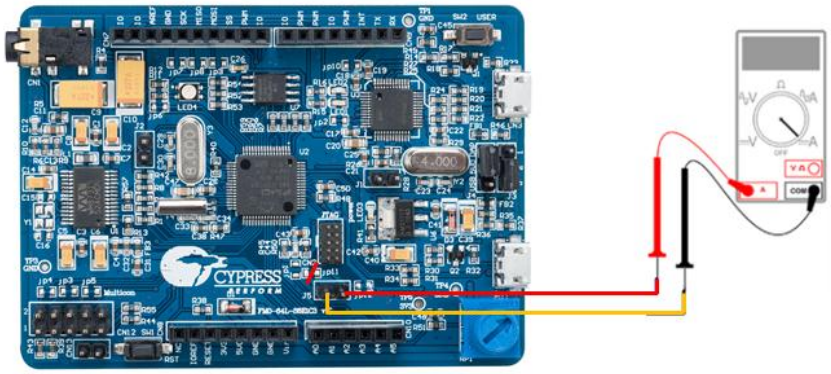
Wire jumpers jp11 and jp12 are used for connecting the 3V3 supply for FM0+ S6E1C3 MCU and peripherals respectively. The jumper J5 is in parallel with jp11.

Figure 3-10: Power Supply Connection



The power supply connection shown above enables customer to measure the current that FM0+ S6E1C3 MCU consumes. Open the jumper jp11 and connect an ammeter at the terminals of J5 as shown in Figure 3-11.

Figure 3-11: Current Measurement



The user can recover the connection of 3V3 wire by closing J5 with a jumper after current measurement.

4. Software Development

4.1 Tool Options

The FM0+ S6E1C-Series device is supported by several third party tools/IDEs, and the user can choose his preferred tool for development. Any one of below listed IDEs can used for opening and building the example projects packaged with this kit:

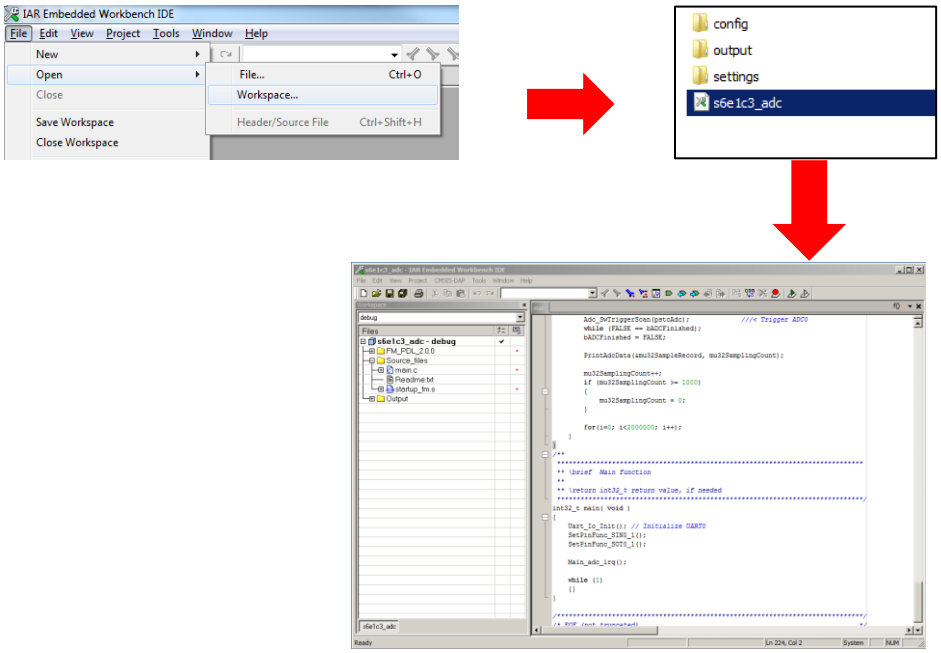
- IAR Embedded Workbench for ARM
- Keil ARM RealView® Microcontroller Development System

Download evaluation versions of these tools from the vendor's website. A full license may be required to build or debug some of the example projects. For detailed information on using the tools, see the documentation in the Help section of the tool chain or the website of the tools supplier.

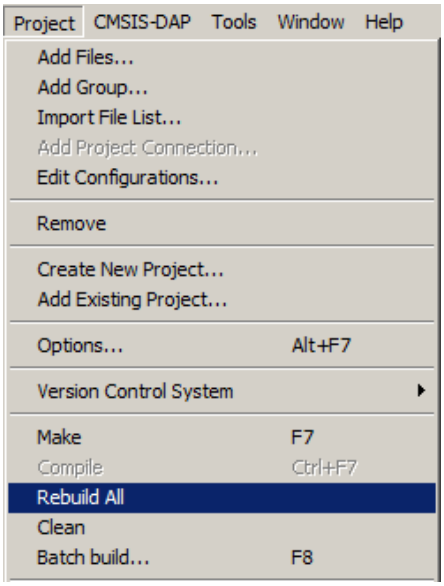
4.1.1 Open the Example Projects in IAR IDE

The following steps describe how to open, build, and run an example project in IAR IDE. The *s6e1c3_adc* is one of example projects available after a successful installation.

1. Launch IAR Embedded Workbench IDE V7.40.5.9739 (or later).
2. Click **File→Open→Workspace** and select the workspace file *s6e1c3_adc.eww* from `<User_Directory>\FM0+ S6E1C-Series Starter Kit_Ver01\Firmware \Demo Projects\s6e1c3_adc\IAR`



3. Click **Project** ->**Rebuild All** to build the project.



4. Make sure the jumpers on the FM0+S6E1C-Series Starter board are placed according to the following table:

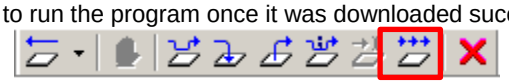
Table 4-1: Debugging Jumper Setting

Jumper	Position	Description
J1	Open	Sets MB9AF312K (CMSIS-DAP) in run mode.
J2	Open	Sets S6E1C3 in run mode.
J4	Pin 1 to Pin 2	Power from USB port of CMSIS-DAP (CN3)

5. Connect the USB cable to CN3 port.
6. Observe that the Power LED (LED3) is glowing green.
7. Click the **Debug** icon in the tool bar; use Shortcut **Crtl+D** or choose **Project-> Download and Debug** to start downloading and debugging.



8. Click the **Run** icon to run the program once it was downloaded successfully.



9. Click the **Stop** icon to stop the program when you want.



For more information about the IAR Embedded Workbench IDE, please click **Help**.

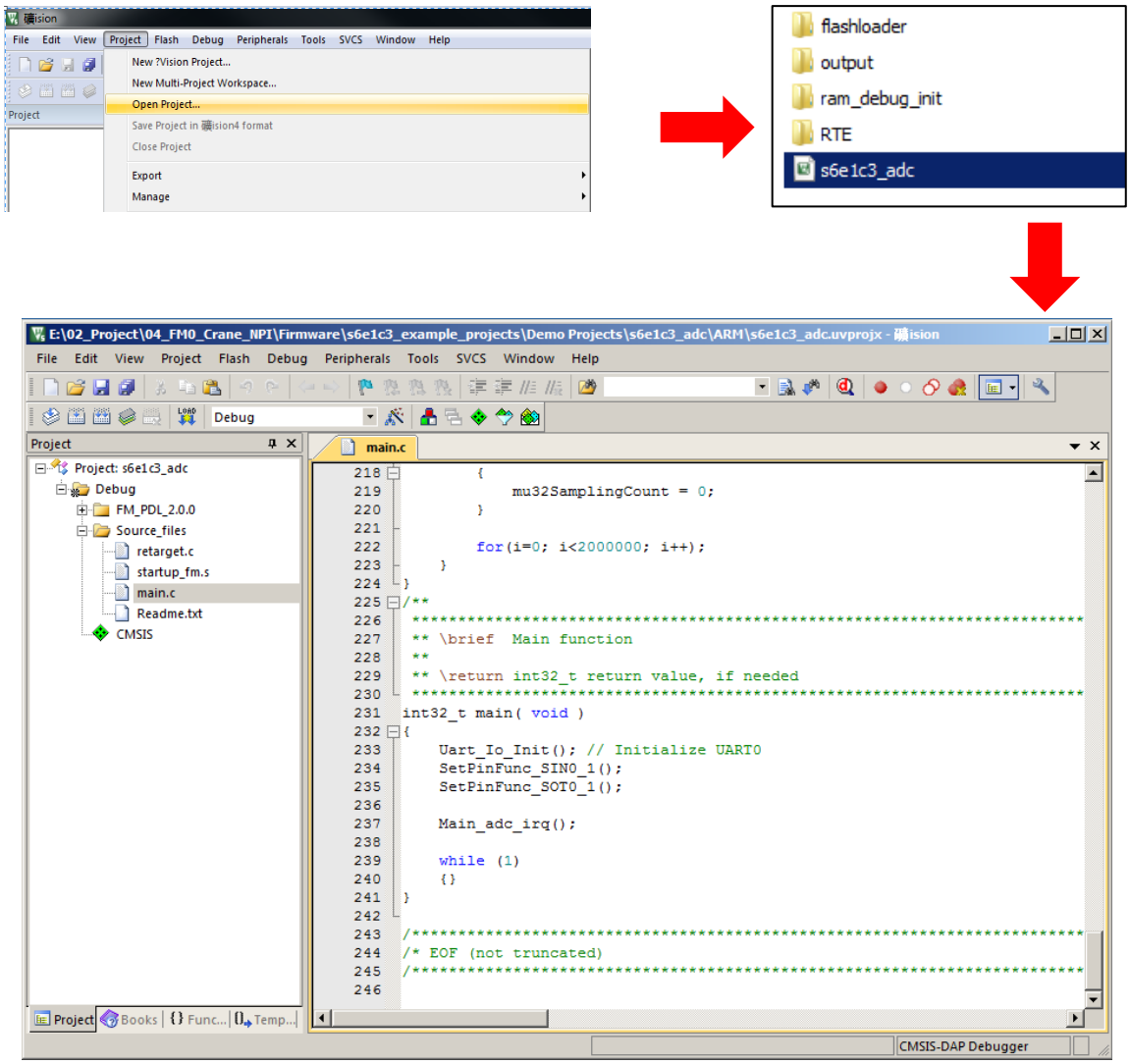
4.1.2 Open the Example Projects in Keil µVision IDE

The following steps describe how to open, build and run an example project in Keil µVision IDE.

Before proceeding, please check the availability of the flash loader file of the S6E1C3 device (*S6E1C32X0.FLM*) in this directory: `<Keil_Install_Directory>:\ARM\flash.`

If the flash loader file is not available, copy from following location to Keil install directory mentioned above:
`<User_Directory>:\FM0+ S6E1C-Series Starter Kit_Ver01\Firmware\Demo Projects\<Project>\ARM\flashloader\S6E1C32X0.FLM`

1. Launch Keil µVision IDE v5.16a (or later).
2. Click **Project→Open Project** and select the workspace file *s6e1c3_dac.uvprojx* from `<User_Directory>:\FM0+ S6E1C-Series Starter Kit_Ver01\Firmware \Demo Projects\s6e1c3_adc\ARM`.



3. Click the **Build** icon to build this project.



4. Make sure the jumpers on the FM0+S6E1C-Series Starter board are placed according to the Table 4-2.

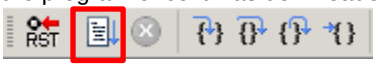
Table 4-2: Debugging Jumper Setting

Jumper	Position	Description
J1	Open	Sets MB9AF312K (CMSIS-DAP) in run mode.
J2	Open	Sets S6E1C3 in run mode.
J4	Pin 1 to Pin 2	Power from USB port of CMSIS-DAP (CN3)

5. Connect the USB cable to CN3 port.
6. Observe that the Power LED (LED3) is glowing green.
7. Click the **Debug** icon, use shortcut **Crtl+F5,** or choose **Debug-> Start/Stop Debug Session** to start downloading and debugging.



8. Click the **Run** icon to run the program once it was downloaded successfully.



9. Click the **Stop** icon to stop the program when you want.



For more information about the Keil µVision IDE, please click **Help**.

4.2 Example Projects

The FM0+ S6E1C-Series Starter Kit includes twelve example projects to help the user get a quick start with the S6E1C3 device. They are in this directory:

```
<User_Directory>:\FM0+ S6E1C-Series Starter Kit_Ver01\Firmware
\Demo Projects
```

These examples listed in Table 4-3 are based on the Peripheral Driver Library (PDL). The PDL provides the API for initializing and operating on-chip peripherals. PDL documentation is available at

```
<User_Directory>:\FM_PDL_2.0.1\doc
```

Use either IAR Embedded Workbench v7.40.5.9739 (or later) or Keil µVision IDE v5.16a (or later) to open these example projects.

Table 4-3: Example Projects

#	Projects	Title/Description
1	s6e1c3_adc	Title: Analog-to-Digital Converter Description: This project demonstrates the AD conversion of the S6E1C3 device. This project enables the ADC channel AN00 to measure the voltage across the potentiometer. The measured value is sent out using UART0. Refer to AD Converter for details.

2	s6e1c3_bt_pwm	Title: Base Timer Description: This project demonstrates the base timer operation of the S6E1C3 device. The project configures a base timer in PWM mode to generate a PWM sequence. The PWM output sequence from TIOA10_1 drives the green LED of LED4. The PWM duty cycle is updated every 2ms on another base timer, producing a breathing LED effect.
3	s6e1c3_dstc	Title: Descriptor System Data Transfer Controller(DSTC) Description: This project demonstrates the Descriptor System Data Transfer Controller (DSTC) operation of the S6E1C3 device. The program configures DSTC to move the data from a source array(*au32SourceData)* to a destination array*(au32DestinationData*), and then compares the content of the arrays to verify the data.Refer to Descriptor System Data Transfer Controller (DSTC) for details.
4	s6e1c3_dt	Title: Dual Timer Description: This project demonstrates the dual timer operation of the S6E1C3 device. This program configures a dual timer to interrupt at a certain interval. In interrupt service routine, the P3E pin is toggled, causing the green LED to blink.
5	s6e1c3_ext_int	Title: External Interrupt Description: This project demonstrates the external interrupt operation of the S6E1C3 device. SW2 key press is detected by the external interrupt (INT03_2). On pressing SW2 key, RGB LED (LED4) will change the color from red to green to blue.
6	s6e1c3_flash	Title: Flash Write Description: This project demonstrates the flash writing operation of the S6E1C3 device. A specific set of four values each of four bytes in size will be written into a specific address location in the flash memory. Refer to Flash Write for the details.
7	s6e1c3_gpio	Title: GPIO Description: This project demonstrates the GPIO operations of the S6E1C3 device by driving an LED. The PB2 pin sinks the current from the green LED of the RGB LED (LED4). The PB2 pin will output a pulse sequence to blink the LED continuously.
8	s6e1c3_mfs_uart	Title: Multi-function Serial Interface Description: This project demonstrates the UART communication of the S6E1C3 device. This program enables the MFS0 as a UART to communicate with the CMSIS-DAP. The CMSIS-DAP serves as the bridge between the MCU and the PC. Refer to UART Communication for details.
9	s6e1c3_rtc	Title: Real Time Clock Description: This project demonstrates the RTC operation of the S6E1C3 device. The program enables the RTC in calendar mode, and sends out the data through UART0.The calendar starts from 2015/9/13 23:59:01 Wednesday. The calendar data will be displayed in Serial Port Viewer window. Refer to RTC Calendar for details.
10	s6e1c3_sleep_mode	Title: Sleep Mode Description: This project demonstrates the sleep mode operation of the S6E1C3 device. The MCU will enter sleep mode after blinking the green LED five times. It can be woken up by pressing SW2. After wakeup, the green LED will turn on. Refer to Sleep Mode for the details.

11	s6e1c3_sw_wdt	Title: Software Watchdog Description: This project demonstrates the operation of the S6E1C3 watchdog by considering two different situations with the watchdog enabled: a) when the watchdog is fed and b) when the watchdog is not fed. If the watchdog is enabled and is fed in time, the program will run properly, and the RGB LED will blink green. If the watchdog is enabled and not fed in time, the device will reset, and the green LED will remain continuously on. Refer to Software Watchdog for the details.
12	s6e1c3_wc	Title: Watch Timer Description: This project demonstrates the Watch Timer function of the S6E1C3 device. The Watch Timer generates an interrupt every second. In the interrupt service routine, the PB2 will drive the green LED to blink.

4.2.1 AD Converter

4.2.1.1 Project Description

This project demonstrates the Analog to Digital conversion of the S6E1C3 device. This project enables the ADC channel AN00 to measure the voltage across potentiometer. The measured voltage converted into decimal format and sent out using UART0.

4.2.1.2 Hardware Connection

No specific hardware connections are required for this project. All connections are hardwired on the board.

4.2.1.3 Verify Output

1. Power the FM0+ S6E1C-Series Starter board from CN3 using USB cable, refer to Figure 2-4.
2. Open the project file in IAR Embedded Workbench or Keil uVision IDE from the following directory:
   ```
   IAR project: <User_Directory>:\FM0+ S6E1C-Series Starter
   Kit_Ver01\Firmware\Demo Projects\s6e1c3_adc\IAR\s6e1c3_adc.eww.

   Keil project: <User_Directory>:\FM0+ S6E1C-Series Starter
   Kit_Ver01\Firmware\Demo Projects\s6e1c3_adc\ARM
   \s6e1c3_adc.uvprojx.
   ```
3. Build the project and download the code to S6E1C3 device.

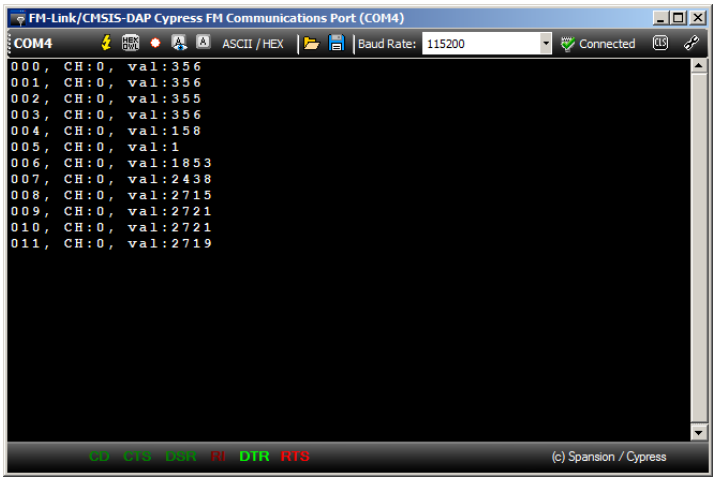
4. Run the Serial Port Viewer, set the baud rate as **115200**, and click the **Disconnected** button to connect the board with PC, as described in Run the Test Demo.

Figure 4-1: Select the Baud Rate



5. Run the program and the ADC value will be displayed in Serial Port Viewer window.

Figure 4-2: ADC value



6. Turn the potentiometer (RP1), the ADC value will change.

4.2.2 Descriptor System Data Transfer Controller (DSTC)

4.2.2.1 Project Description

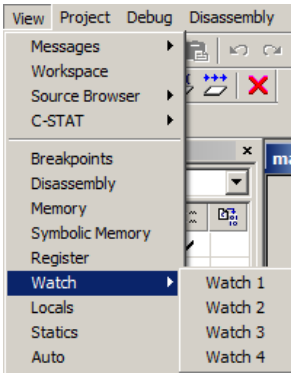
This project demonstrates the Descriptor System Data Transfer Controller (DSTC) operation of the S6E1C3 device. The program configures DSTC to move the data from a source array (*au32SourceData*) to a destination array (*au32DestinationData*), and then compares the content of the arrays to verify the data.

4.2.2.2 Hardware Connection

No specific hardware connections are required for this project. All connections are hardwired on the board.

4.2.2.3 Verify Output

- Open the project in IAR Embedded Workbench
  - Power the FM0+ S6E1C-Series Starter board from CN3 using USB cable, refer to Figure 2-4.
  - Open the project file in IAR Embedded Workbench from the following directory on your PC:
    `IAR project: <User_Directory>:\FM0+ S6E1C-Series Starter Kit_Ver01\Firmware\Demo Projects\s6e1c3_dstc\IAR \s6e1c3_dstc.eww.`
  - Build the project and download the code to the S6E1C3 device.
  - Open Watch1 window from **View**->**Watch**.



  - Add the arrays *m_au32SourceData*and *m_au32DestinationData*in Watch1 window.

Watch 1

Expression	Value	Location	Type
⊞ m_au32SourceD...	<array>	0x20000000	uint32_t[16]
⊞ m_au32Destinati...	<array>	0x20000040	uint32_t[16]
<click to edit>			

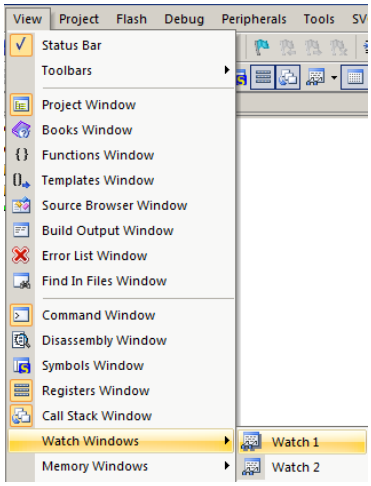
  - Run the program for a while (>10s).
  - Stop the program and check the arrays mentioned above. The Program Counter (PC) will stop at the routine as below which means the content of the arrays are the same.

```
    if (TRUE == bCompareError) // Should never happen ...
    {
        while(1)
        {}
    }

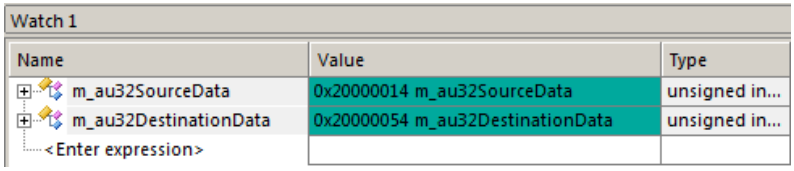
    while(1)
    {}
}
```

- Open the project in Keil uVision IDE

- Power the FM0+ S6E1C-Series Starter board from CN3 using USB cable, refer to Figure 2-4.
- Open the project filein Keil uVision IDE from the following directory on your PC: `Keil project: <User_Directory>:\FM0+ S6E1C-Series Starter Kit_Ver01\Firmware\Demo Projects\s6e1c3_dstc\ARM \s6e1c3_dstc.uvprojx.`
- Build the project and download the code to the S6E1C3 device.
- Open Watch1 window from **View->Watch Windows**.



- Add the arrays *au32SourceData* and *au32DestinationData* in Watch1 window.



- Run the program for a while (>10s).
- Stop the program and check the arrays mentioned above. The Program Counter (PC) will stop at the routine as below, which means the content of the arrays are the same.

```
206     if (TRUE == bCompareError) // Should never happen ...
207     {
208         while(1)
209         {}
210     }
211
212     while(1)
213     {}
214 }
```

4.2.3 Flash Write

4.2.3.1 Project Description

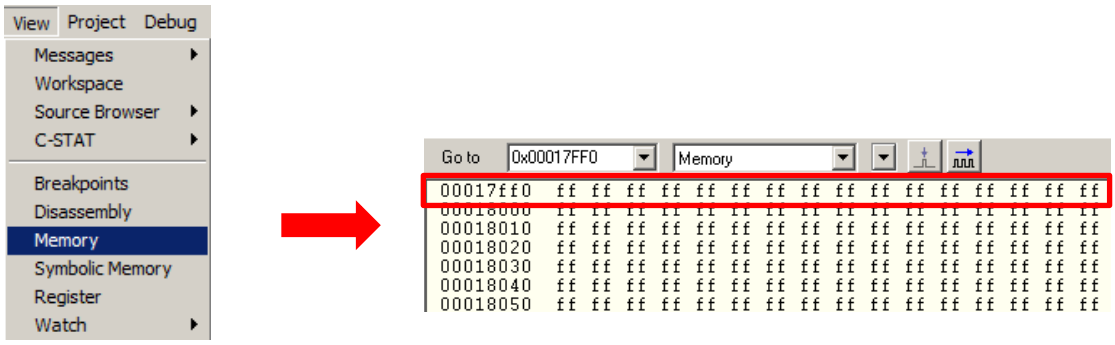
This project demonstrates the flash writing operation of the S6E1C3 device. A specific set of four values each of four bytes in size will be written into a specific address location in the flash memory.

4.2.3.2 Hardware Connection

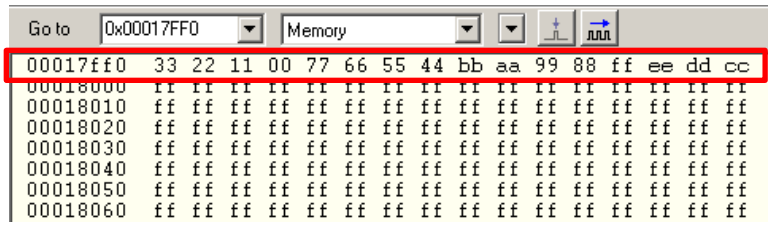
No specific hardware connections are required for this project. All connections are hardwired on the board.

4.2.3.3 Verify Output

- Open the project in IAR Embedded Workbench
  - Power the FM0+ S6E1C-Series Starter board from CN3 using USB cable, refer to Figure 2-4.
  - Open the project file in IAR Embedded Workbench from the following directory on your PC:
    `IAR project: <User_Directory>:\FM0+ S6E1C-Series Starter Kit_Ver01\Firmware\Demo Projects\s6e1c3_flash\IAR \s6e1c3_flash.eww.`
  - Build the project and download the code into the S6E1C3 device.
  - Open the memory window from the **View** ->**Memory**. Enter 0x00017FF0 in the **Go to** table and press the **Enter** key on your PC.

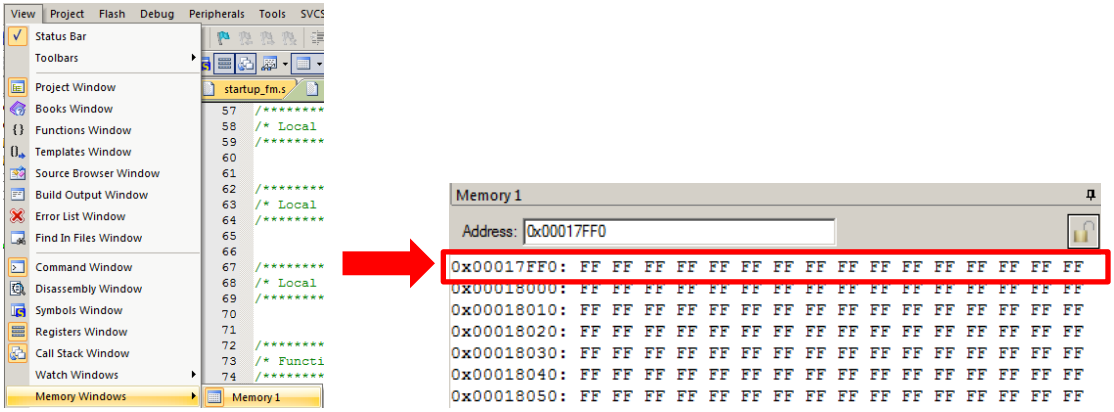


  - Run the program for a while (>10s).
  - Stop the program and check the content of 0x00017FF0 in flash.

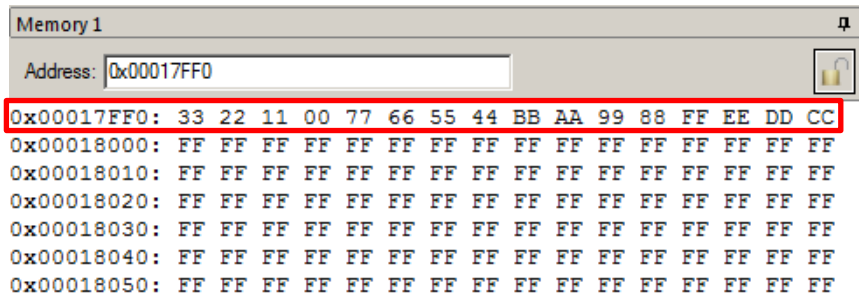


- Open the project in Keil uVision IDE.
  - Power the FM0+ S6E1C-Series Starter board from CN3 using USB cable, refer to Figure 2-4.

- Open the project file in Keil uVision IDE from the following directory on your PC: `Keil project: <User_Directory>:\FM0+ S6E1C-Series Starter Kit_Ver01\Firmware\Demo Projects\s6e1c3_flash\ARM \s6e1c3_flash.uvprojx.`
- Build the project and download the code to the S6E1C3 device.
- Open the **Memory1** window from the **View** ->**Memory Windows**. Enter 0x00017FF0 in the **Address** table and press the **Enter** key on your PC.



- Run the program for a while (>10s).
- Stop the program and check the content of 0x00017FF0 in flash.



4.2.4 UART Communication

4.2.4.1 Project Description

This project demonstrates the UART communication of the S6E1C3 device. This program enables the MFS0 as UART mode to communicate with the CMSIS-DAP. The CMSIS-DAP serves as the bridge between the MCU and PC.

4.2.4.2 Hardware Connection

No specific hardware connections are required for this project. All connections are hardwired on the board.

4.2.4.3 Verify Output

1. Power the FM0+ S6E1C-Series Starter board from CN3 using USB cable, refer to Figure 2-4.
2. Open the project file in IAR Embedded Workbench or Keil uVision IDE from the following directory:
   `IAR project: <User_Directory>:\FM0+ S6E1C-Series Starter Kit_Ver01\Firmware\Demo Projects\s6e1c3_mfs_uart\IAR \s6e1c3_mfs_uart.eww.`

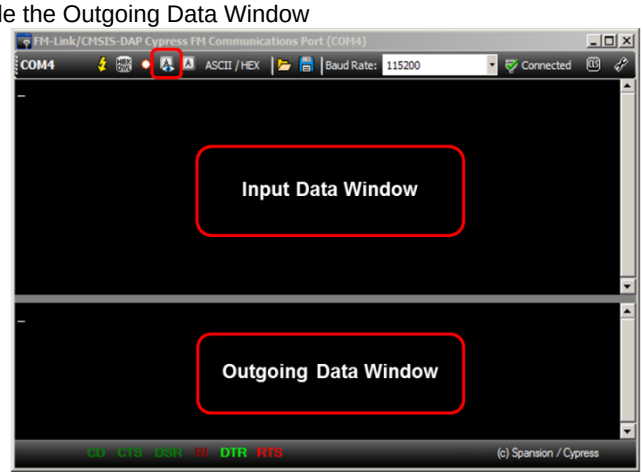
   `Keil project: <User_Directory>:\FM0+ S6E1C-Series Starter Kit_Ver01\Firmware\Demo Projects\s6e1c3_mfs_uart\ARM \s6e1c3_mfs_uart.uvprojx.`
3. Build the project and download the code into S6E1C3 device.
4. Run the program.
5. Run the Serial Port Viewer, set the baud rate as **115200**, and click the **Disconnected** button to connect the board with PC, as described in Run the Test Demo.

Figure 4-3: Select the Baud Rate



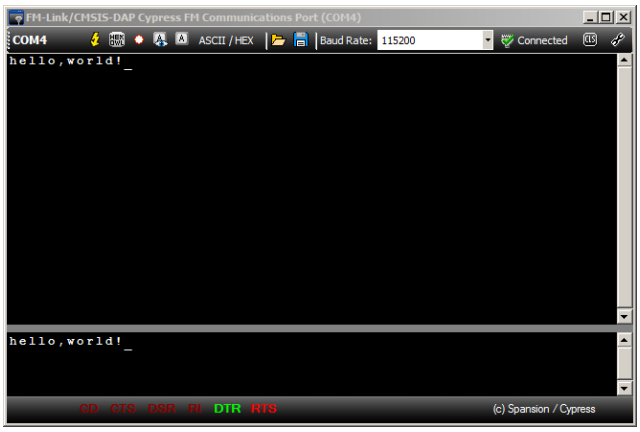
6. Click the button shown below to toggle the Outgoing Data Window.

Figure 4-4: Toggle the Outgoing Data Window



7. Key in any characters in the Outgoing Data Window, the same characters will be echoed in the Input Data Window.

Figure 4-5: Echo Test



4.2.5 RTC Calendar

4.2.5.1 Project Description

This project demonstrates the RTC operation of the S6E1C3 device. The program enables the RTC in calendar mode, and sends out the current calendar through UART0.The calendar starts from 2015/9/13 23:59:01 Wednesday.

4.2.5.2 Hardware Connection

No specific hardware connections are required for this project. All connections are hardwired on the board.

4.2.5.3 Verify Output

1. Power the FM0+ S6E1C-Series Starter board from CN3 using USB cable, refer to Figure 2-4.
2. Open the project file in IAR Embedded Workbench or Keil uVision IDE from the following directory:

   `IAR project: <User_Directory>:\FM0+ S6E1C-Series Starter Kit_Ver01\Firmware\Demo Projects\s6e1c3_rtc\IAR\s6e1c3_rtc.eww.`

   `Keil project: <User_Directory>:\FM0+ S6E1C-Series Starter Kit_Ver01\Firmware\Demo Projects\s6e1c3_rtc\ARM\s6e1c3_rtc.uvprojx.`
3. Build the project and download the code on the S6E1C3 device.

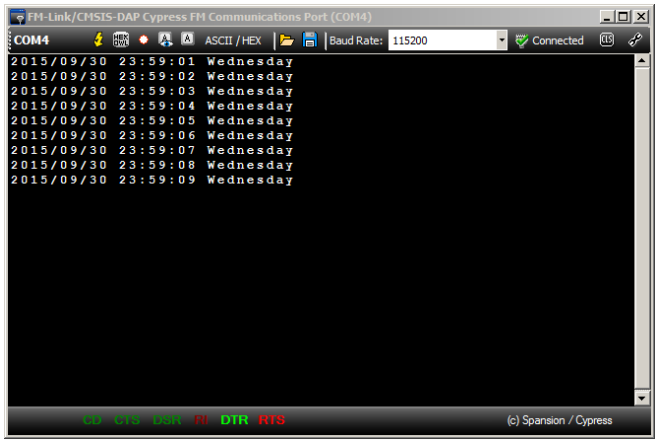
4. Run the Serial Port Viewer, set the baud rate as **115200**, and click the **Disconnected** button to connect the board with PC, as described in Run the Test Demo.

Figure 4-6: Select the Baud Rate



5. Run the program.
6. The calendar data will be displayed in Serial Port Viewer window.

Figure 4-7: Calendar



4.2.6 Sleep Mode

4.2.6.1 Project Description

This project demonstrates the sleep mode operation of the S6E1C3 device. The MCU will enter sleep mode after the RGB LED (LED4) blink five times (green color). It can be awakened by pressing the SW2 switch. After wakeup, the green LED will turn on.

4.2.6.2 Hardware Connection

No specific hardware connections are required for this project. All connections are hardwired on the board.

4.2.6.3 Verify Output

1. Power the FM0+ S6E1C-Series Starter board from CN3 using USB cable, refer to Figure 2-4.
2. Open the project file in IAR Embedded Workbench or Keil µVision IDE from the following directory on your PC:
   ```
   IAR project: <User_Directory>:\FM0+ S6E1C-Series Starter
   Kit_Ver01\Firmware\Demo Projects\s6e1c3_sleep_mode\IAR
   \s6e1c3_sleep_mode.eww.

   Keil project: <User_Directory>:\FM0+ S6E1C-Series Starter
   Kit_Ver01\Firmware\Demo Projects\s6e1c3_sleep_mode\ARM
   \s6e1c3_sleep_mode.uvprojx.
   ```
3. Build the project and download the code into the S6E1C3 device.
4. Run the program.
5. The RGB LED (LED4) will blink 5 times (green color), and then the MCU enters sleep mode.
6. Press the SW2 key to wake up the MCU.
7. The RGB LED (LED4) will glow with green color.

4.2.7 Software Watchdog

4.2.7.1 Project Description

This project is to demonstrate the operation of the S6E1C3 watchdog. The project will demonstrate the impact of two different situations (with watchdog enabled): when the watchdog is fed and when the watchdog is not fed.

If the watchdog is enabled, but the watchdog is fed in time, the program will run properly, and the RGB LED will blink in green.

If the watchdog is enabled, but the watchdog is not fed in time, the chip is reset, and the green LED will remain on.

4.2.7.2 Hardware Connection

No specific hardware connections are required for this project. All connections are hardwired on the board.

4.2.7.3 Verify Output

1. Power the FM0+ S6E1C-Series Starter board from CN3 using USB cable, refer to Figure 2-4.
2. Open the project file in IAR Embedded Workbench or Keil µVision IDE from the following directory on your PC:
   ```
   IAR project: <User_Directory>:\FM0+ S6E1C-Series Starter
   Kit_Ver01\Firmware\Demo Projects\s6e1c3_st_wdt\IAR
   \s6e1c3_st_wdt.eww.

   Keil project: <User_Directory>:\FM0+ S6E1C-Series Starter
   Kit_Ver01\Firmware\Demo Projects\s6e1c3_st_wdt\ARM
   \s6e1c3_st_wdt.uvprojx.
   ```
3. Build the project and download the code into the S6E1C3 device.

4. Run the program.
5. The RGB LED (LED4) will blink with green color.
6. Stop the program, comment out the line *Swwdg_Feed();* in main.c file and click **File** ->**Save**.

```
static void WdgSwCallback(void)
{
    //Shield "Swwdg_Feed()"
    //Swwdg_Feed();   // Clear Irq and Reset Timer
    ++u8CountWdg;
    if (TRUE != FM_SWWDT->WDOGRIS_f.RIS) // If the watchdog interrupt flag had
    {
        SetLed(u8CountWdg);
    }
}
```

7. Exit from debug mode, then repeat the step 3~4.
8. The RGB LED (LED4) will glow with green color.

4.3 Flash Programming

4.3.1 Programming the S6E1C3 Using FLASH USB DIRECT Programmer

1. FLASH USB DIRECT Programmer gets installed on your PC as part the Kit installer.
2. Make sure the jumpers on the FM0+ S6E1C-Series Starter board are placed according to the below table.

Table 4-4: Programming Jumper Settings using FLASH USB DRIECT Programmer

Jumper	Position	Description
J1	Open	Sets MB9AF312K (CMSIS-DAP) in run mode.
J2	Closed	Sets S6E1C3 in programming mode
J3	Pin 2 to Pin 3	Sets USB programming mode
J4	Pin 2 to Pin 3	Power from USB port of the S6E1C3 (CN4)

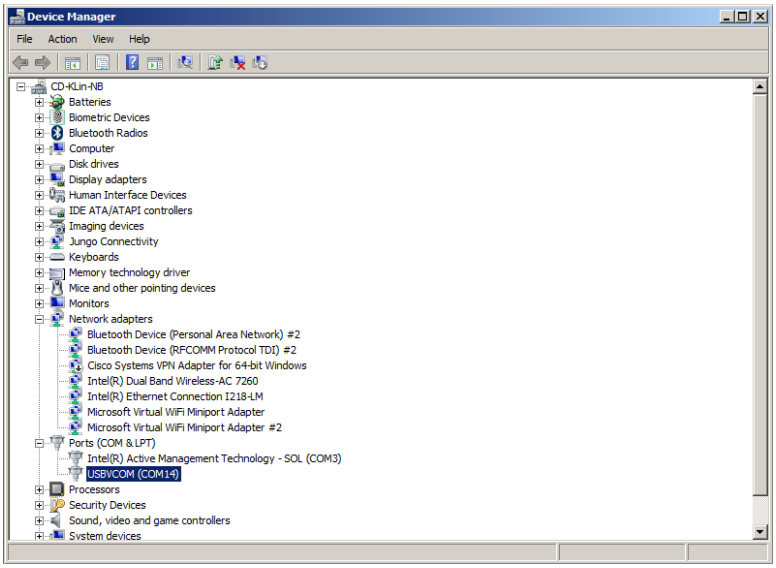
3. Connect the USB cable to CN4 port.
4. Observe that the Power LED (LED3) is glowing green.
5. Launch the FLASH USB DIRECT Programmer from Windows

   Start Menu*-> *All Programs* > *Cypress* > *FLASH USB DIRECT Programmer* > *USBDirect
6. Select **Target MCU** as **S6E1C32B/C/D**.
7. Select the Motorola S-record file or Intel-HEX format file to be programmed to FLASH memory in the MCU.

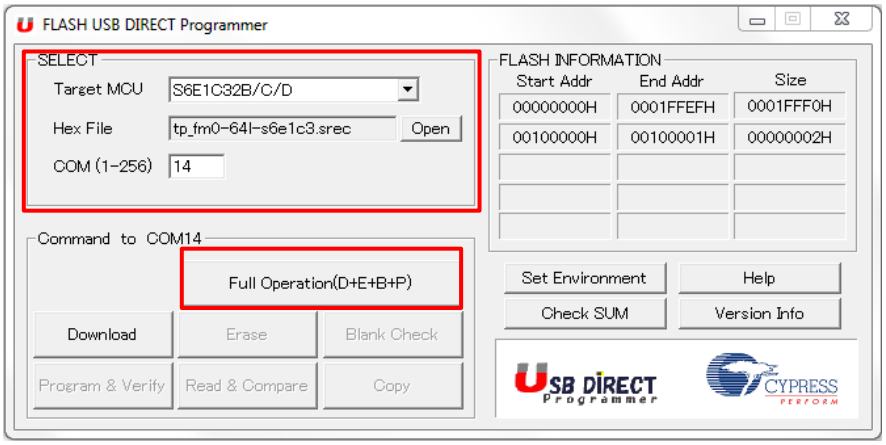
   Note: The HEX file selected in this example is the Test Demo firmware. It is available in directory

   `<User_Directory>:\FM0+ S6E1C-Series Starter Kit_Ver01\Firmware \Demo Projects\Test_Demo_Code`

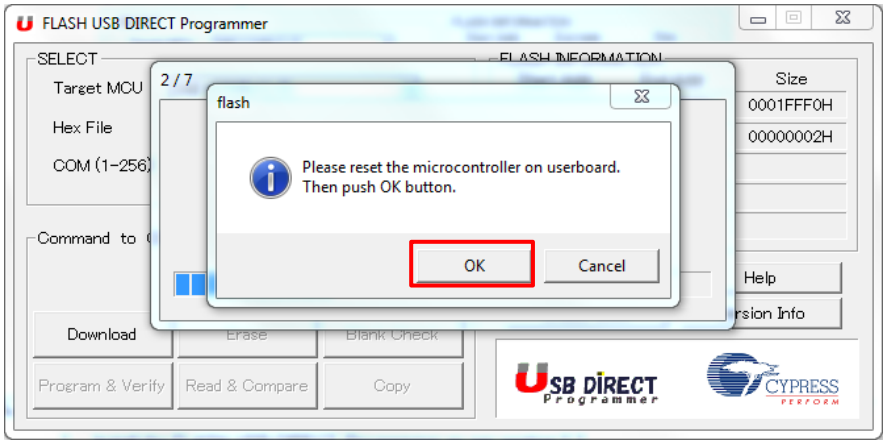
8. Check the COM Port number in the Windows Device Manager.



9. Enter the Virtual COM Port listed in the Ports of Device Manager in **COM** box.
10. Click the **Full Operation (D+E+B+P)** button to start programming.



11. Reset the S6E1C3 by pressing the reset button (SW1) on the board, and click **OK**.



Note: Please click on **Help** for any issues or errors encountered during programming.

4.3.2 Programming CMSIS-DAP (MB9AF312K) Using FLASH USB DIRECT Programmer

By default, the latest CMSIS-DAP firmware is programmed on the MB9AF312K already. It is not required for the user to re-program firmware again. Follow the steps below to update the firmware, if needed.

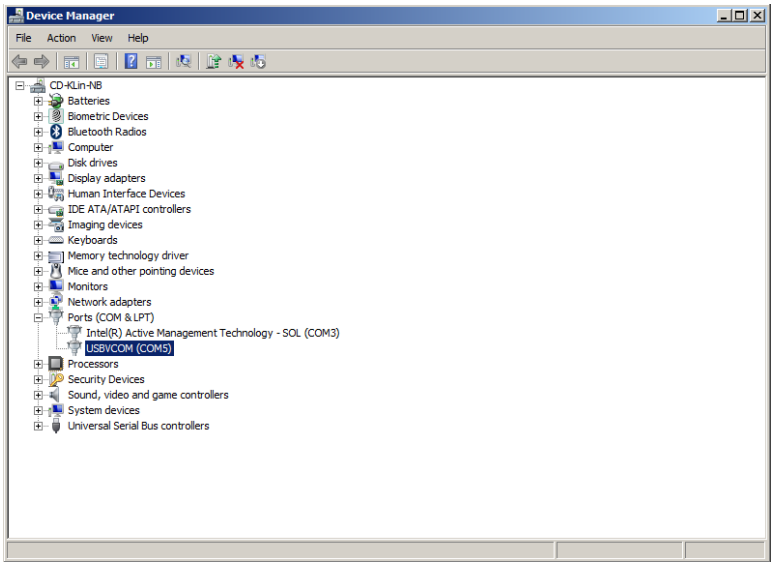
1. Make sure the jumpers on the FM0+ S6E1C-Series Starter board are placed according to the below table:

Table 4-5: Programming Jumper Settings using FLASH USB DIRECT Programmer

Jumper	Position	Description
J1	Closed	Sets MB9AF312K (CMSIS-DAP) in programming mode.
J2	Open	Sets S6E1C3 in run mode.
J4	Pin 1 to Pin 2	Power from USB port of CMSIS-DAP (CN3)

2. Connect the USB cable to CN3 port.
3. Observe that Power LED (LED3) is glowing green.

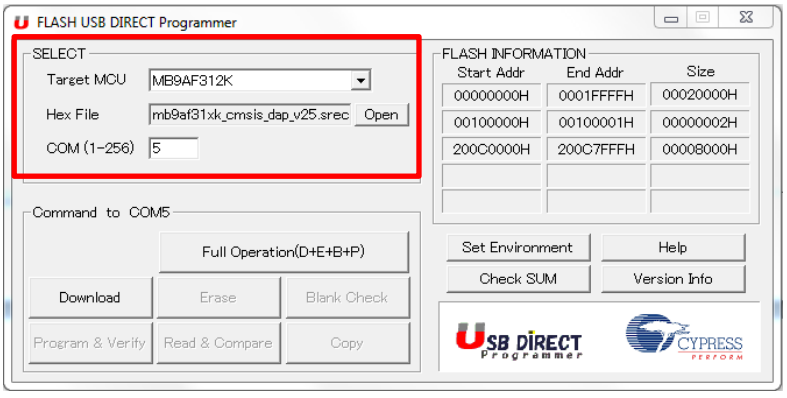
4. Check the COM Port number in the Windows Device Manager.



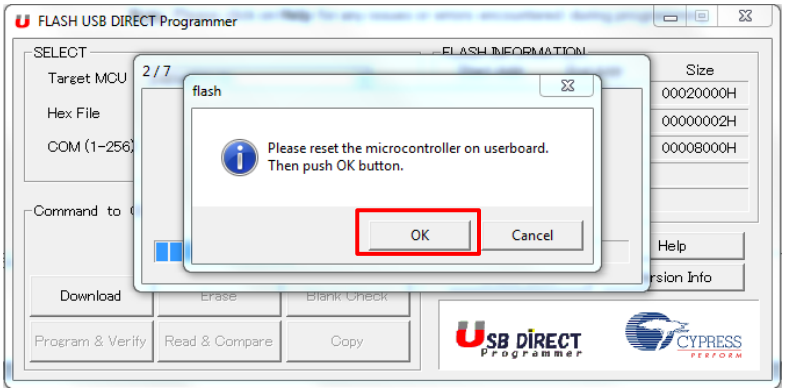
5. Launch the FLASH USB DIRECT Programmer from Windows

   Start Menu > ***All Programs*** > ***Cypress*** > ***FLASH USB DIRECT Programmer*** > ***USBDirect***

6. Select “Target MCU” to **MB9AF312K**.
7. Select the Motorola-S format file or Intel-HEX format file to be programmed to FLASH memory in the MCU. The Hex file is included in the following directory:
   `<User_Directory>:\FM0+ S6E1C-Series Starter Kit_Ver01\Firmware \CMSIS-DAP`
8. Enter the Virtual COM Port listed in the Ports of Device Manager in **COM** box.
9. Click the **Full Operation (D+E+B+P)** button to start programming.



12. Reset the CMSIS-DAP microcontroller by removing and reconnecting the USB cable, and click **OK**.



Note: Please click on **Help** for any issues or errors encountered during programming.

4.3.3 Programming the S6E1C3 Using the FLASH MCU Programmer

The FLASH MCU Programmer is not supported on the S6E1C3 Starter Kit board.

A. Appendix

A.1 Schematic

Figure A-1. MCU

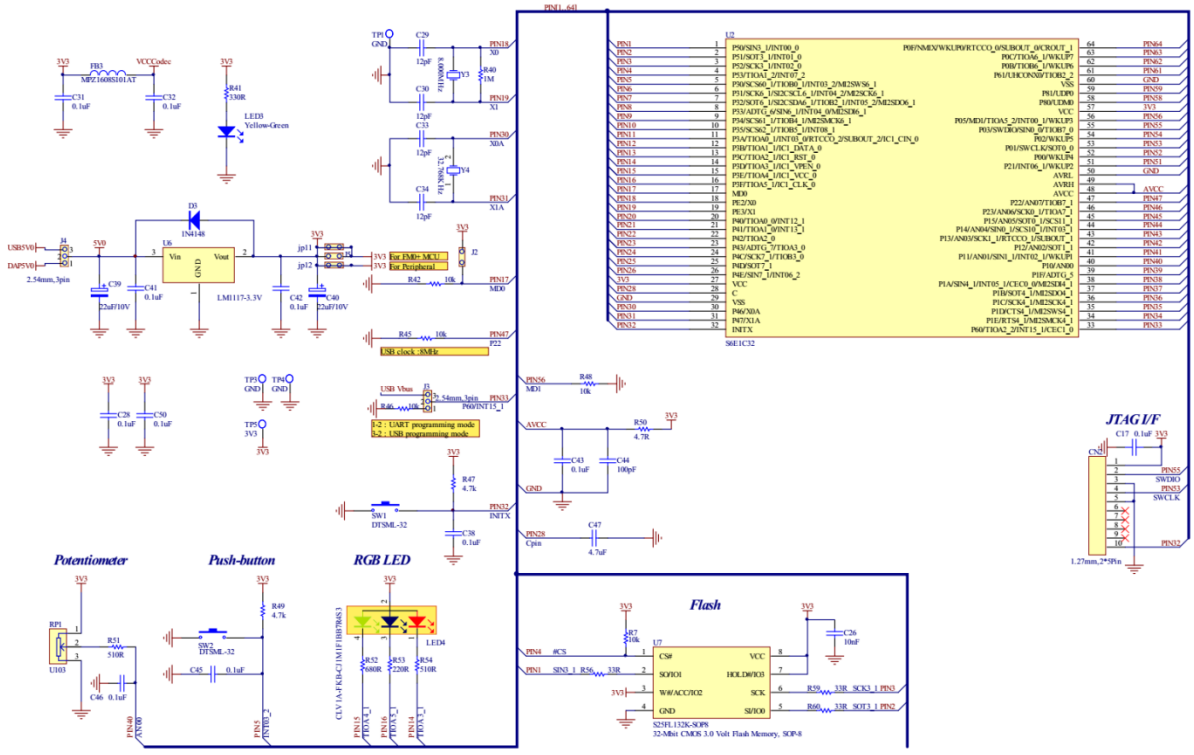


Figure A-2. CMSIS-DAP & USB

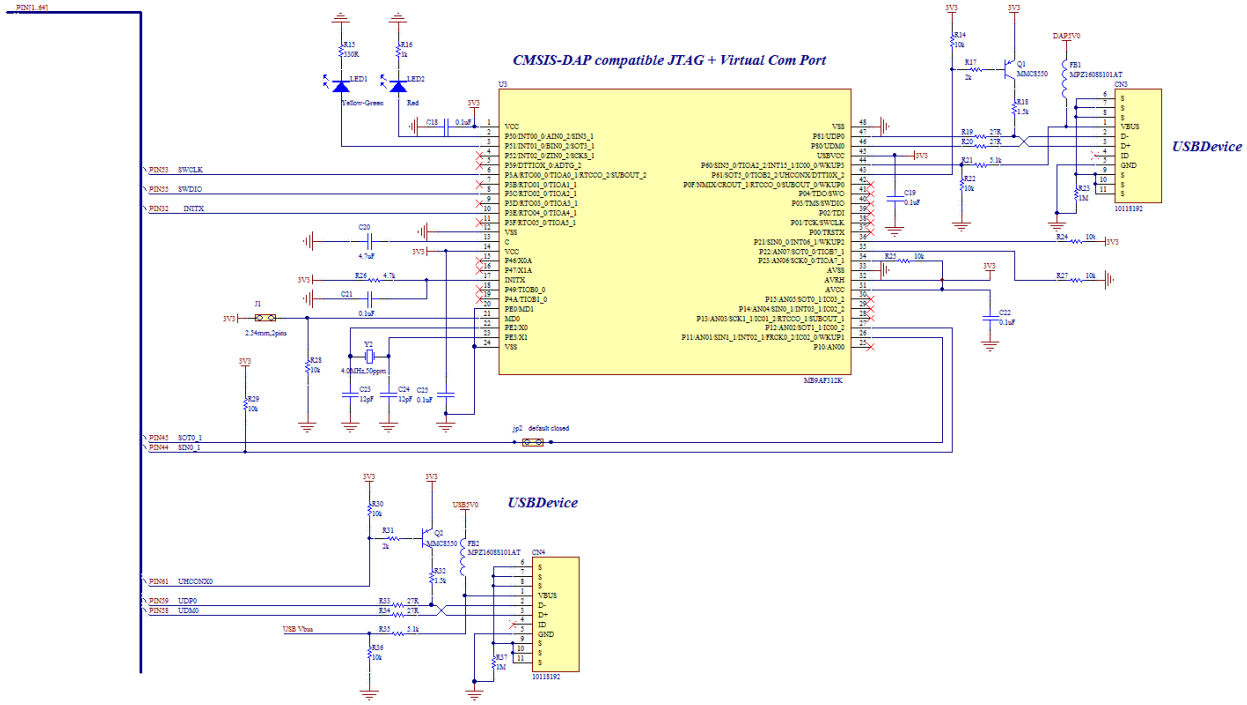


Figure A-3: Stereo Codec

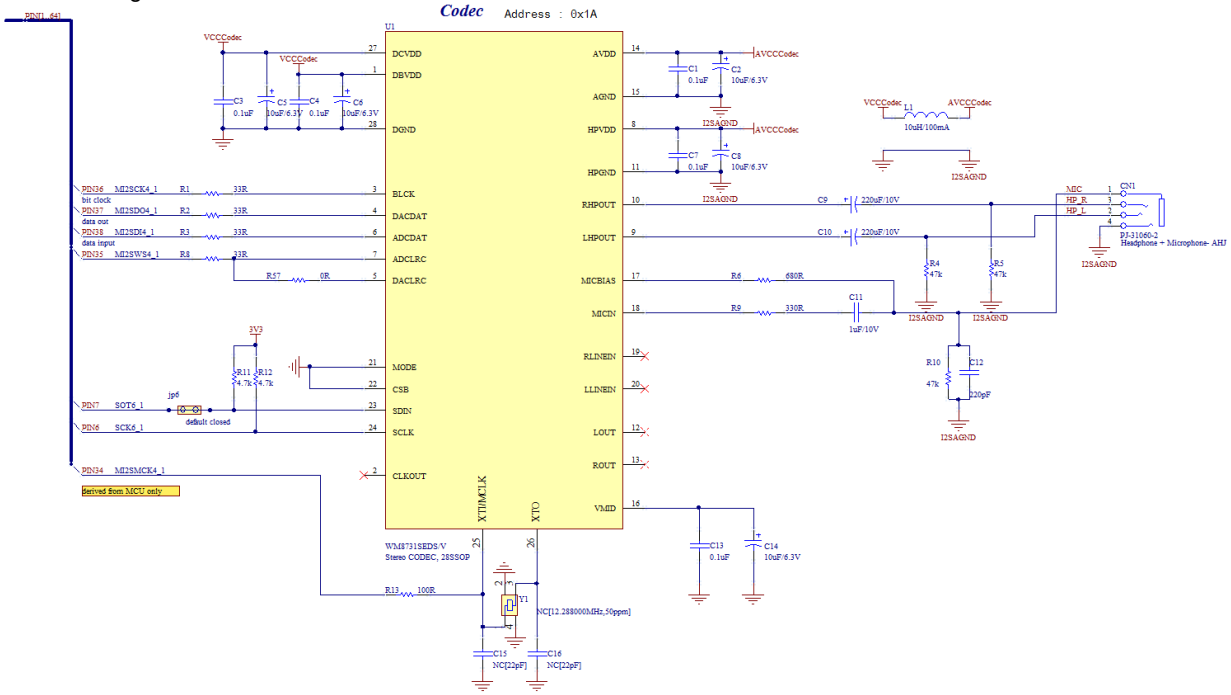
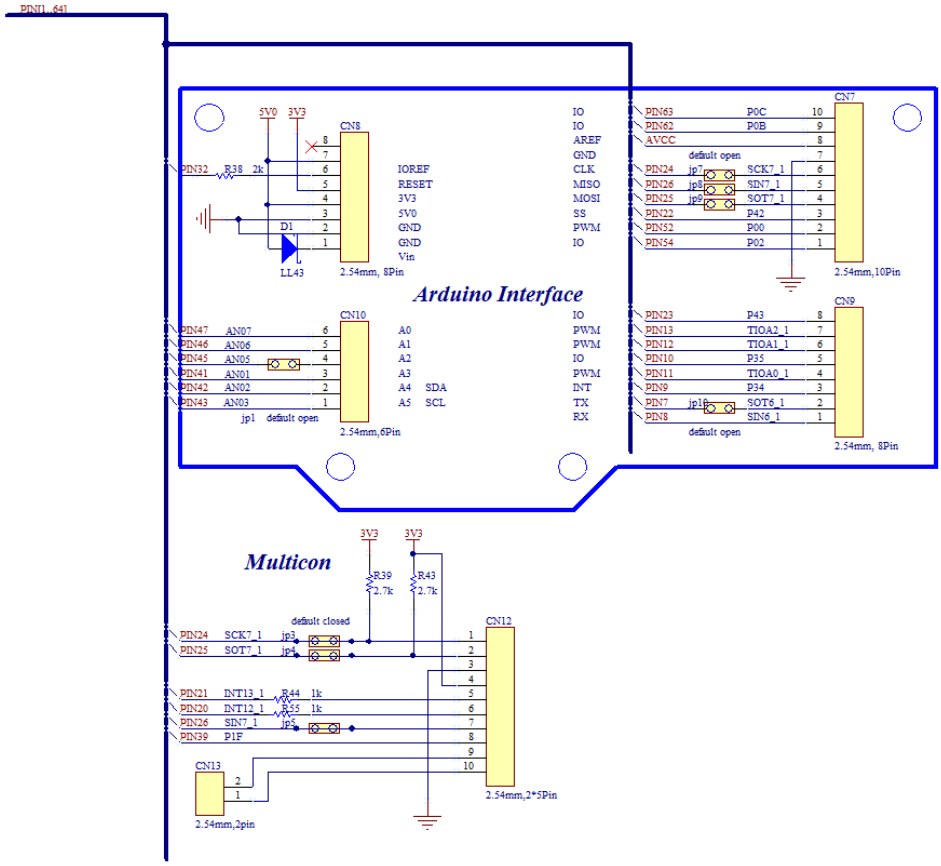


Figure A-4: Interfaces



A.2 Bill of Materials

Item	Qty	Reference	Value	Description	Mfg	Mfg part number
1	21	C1, C3, C4, C7, C13, C17, C18, C19, C21, C22, C25, C28, C31, C32, C38, C41, C42, C43, C45, C46, C50	0.1uF	Capacitor without polarity	YAGEO	CC0603JRX7R9BB104
2	5	C2, C5, C6, C8, C14	10uF/6.3V	Tan capacitor	AVX	TAJA106K006RNJ
3	2	C9, C10	220uF/10V	Tan capacitor	AVX	TAJD227K010R
4	1	C11	1uF/10V	Capacitor without polarity	Murata	GRM21BR71A105KA01L
5	1	C12	220pF	Capacitor without polarity	YAGEO	CC0603JRNPO9BB221
6	2	C20, C47	4.7uF	Capacitor without polarity	YAGEO	CC0805KKX5R8BB475
7	6	C23, C24, C29, C30, C33, C34	12pF/50V	Capacitor without polarity	YAGEO	CC0603JPNPO9BN120
8	1	C26	10nF/50V	Capacitor without polarity	YAGEO	CC0603KRX7R9BB103
9	2	C39, C40	22uF/10V	Tan capacitor	AVX	TAJA226K010RNJ
10	1	C44	100pF/50V	Capacitor without polarity	YAGEO	CC0603JRNPO9BN101
11	1	CN1	PJ-31060-2	3.5mm, micro+ headphone jack	AIMO	PJ-31060-2
12	1	CN2	1.27mm,2*5Pin	1.27mm, 2*5 pin header	AIMO	1415-1205CNGOS3.01.52.301
13	2	CN3, CN4	10118192	Micro USB-B type, FCI, Micro USB-B type,FCI	FCI	10118192
14	1	CN7	2.54mm,10Pin	Arduino header	AIMO	2285-0110ANGO01
15	2	CN8, CN9	2.54mm, 8Pin	Arduino header	AIMO	2285-0108ANGO01
16	1	CN10	2.54mm,6Pin	Arduino header	AIMO	2285-0106ANGO01
17	1	CN12	2.54mm,2*5Pin	2*5 pin headers, 2.54mm	AIMO	1225-1210ANG0S11501
18	1	D1	LL43-GS08	Socktety diode	Vishay	LL43-GS08
19	1	D3	DL4148	Recifier diode	MCC	DL4148
20	3	FB1, FB2, FB3	MPZ1608S101AT	Ferrite bead, 3A,100R@100MHz,TDK	TDK	MPZ1608S101AT

Item	Qty	Reference	Value	Description	Mfg	Mfg part number
21	3	J1, J2, J5	2.54mm,2pins	2.54mm,2pins	AIMO	1225-1102ANGOS11.501
22	2	J3, J4	2.54mm,3pin	2.54mm,3pins	AIMO	1225-1103ANGOS11.501
23	5	jp1, jp7, jp8, jp9, jp10	default open	Wire jumper		N/A
24	2	jp2, jp6	default closed	Wire jumper		N/A
25	5	jp3, jp4, jp5, jp11, jp12	default closed	Wire jumper		N/A
26	1	L1	MLZ2012N100LT	Ferrite Bead,TDK	TDK	MLZ2012N100LT
27	2	LED1, LED3	Yellow-Green	LED	Everlight	19-21SYGC/S530-E3/TR8
28	1	LED2	Red	LED	Everlight	19-21SURC/S530-A5/TR8
29	1	LED4	CLV1A-FKB-CJ1M1F1BB7R4S3	RGB LED,SMD,4pin	Cree	CLV1A-FKB-CJ1M1F1BB7R4S3
30	2	Q1, Q2	MMC8550	PNP transistor	MCC	MMS8550-H-TP
31	7	R1, R2, R3, R8, R56, R59, R60	33R	Resistor	YAGEO	RC0603FR-0733RL
32	3	R4, R5, R10	47k	Resistor	YAGEO	RC0603FR-0747KL
33	2	R6, R52	680R	Resistor	YAGEO	RC0603FR-07680RL
34	14	R7, R14, R22, R24, R25, R27, R28, R29, R30, R36, R42, R45, R46, R48	10k	Resistor	YAGEO	RC0603FR-0710KL
35	3	R9, R15, R41	330R	Resistor	YAGEO	RC0603FR-07330RL
36	5	R11, R12, R26, R47, R49	4.7k	Resistor	YAGEO	RC0603FR-074K7L
37	1	R13	100R	Resistor	YAGEO	RC0603FR-07100RL
38	3	R16, R44, R55	1k	Resistor	YAGEO	RC0603FR-071KL
39	3	R17, R31, R38	2k	Resistor	YAGEO	RC0603FR-072KL
40	2	R18, R32	1.5k	Resistor	YAGEO	RC0603FR-071K5L
41	4	R19, R20, R33, R34	27R	Resistor	YAGEO	RC0603FR-0727RL
42	2	R21, R35	5.1k	Resistor	YAGEO	RC0603FR-075K1L

Item	Qty	Reference	Value	Description	Mfg	Mfg part number
43	3	R23, R37, R40	1M	Resistor	YAGEO	RC0603FR-071ML
44	2	R39, R43	2.7k	Resistor	YAGEO	RC0603FR-072K7L
45	1	R50	4.7R	Resistor	YAGEO	RC0603FR-074R7L
46	2	R51, R54	510R	Resistor	YAGEO	RC0603FR-07510RL
47	1	R53	220R	Resistor	YAGEO	RC0603FR-07220RL
48	1	R57	0R	Resistor	YAGEO	RC0603FR-070RL
49	1	RP1	3386P-1-103T	Potentiometer	BURANS	3386P-1-103T
50	2	SW1, SW2	K2-1101ST-C4SA-01	3.5*6*5mm, SMT, 2Pin Push-button	HARO	K2-1101ST-C4SA-01
51	1	U1	WM8731SEDS/V	Stereo CODEC, 28SSOP, Wolfson,ROHS	Wolfson	WM8731SEDS/V
52	1	U2	S6E1C32D0AGV20000	0.5mm,LQFP64,Spansion	Cypress	S6E1C32D0AGV20000
53	1	U3	MB9AF312KPMC	MB9AF314KPMC/QN 48 Pin two-sided	Cypress	MB9AF314KPMC
54	1	U6	LM1117-3.3V	LDO,TI	TI, AMS	LM1117IMPX-3.3/NOPB
55	1	U7	S25FL132K0XMFI011	32-Mbit CMOS 3.0 Volt Flash Memory, SOP-8	Cypress	S25FL132K0XMFI011
56	1	Y2	4.0MHz,50ppm	Crystal Oscillator ,4.0000MHz, 49s	Wisdom	QRS-4M00A5020B
57	1	Y3	8.000MHz,50ppm	Crystal Oscillator ,8.0000MHz,49s	Wisdom	QRS-8M00A5020B
58	1	Y4	32.768KHz	Crystal Oscillator	Wisdom	QRA-32768A20125B
59	1	PCB	PCB	2-layers,1.6mm,FR-4;84*60mm, ROHS	UEC	
60	4	rubber		6.0*3.0 mm ,black		

Revision History

Document Revision History

Document Title: FM0+ S6E1C-Series Starter Kit Guide Document Number: 002-10542				
Revision	**ECN Number**	**Issue Date**	**Origin of Change**	**Description of Change**
**	5062321	12/23/2015	CCTA	Initial revision.
*A	5151290	02/25/2016	CCTA	1. modify Figure 1-2, have an error for the #Item9 2. update pdl 2.0.0 to pdl 2.0.1
*B	5160824	03/03/2016	BHAT / CCTA	Updated to new template.
*C	5171622	03/15/2016	CCTA / JETT	Replaced phototransistor with potentiometer across the document. Updated Installation and Test Operation: Updated Test Operation: Updated Test Procedure Explanation (Updated description). Updated Hardware: Updated Hardware Details: Updated NOR Flash (Added Table 3-4). Updated Potentiometer (Updated Figure 3-6). Updated Software Development: Updated Tool Options (Updated description). Updated Open the Example Projects in Keil µVision IDE (Updated description). Updated Example Projects (Updated Table 4-3). Updated Flash Programming: Updated Programming the S6E1C3 Using FLASH USB DIRECT Programmer (Updated description). Updated Programming CMSIS-DAP (MB9AF312K) Using FLASH USB DIRECT Programmer (Updated description). Added Programming the S6E1C3 Using the FLASH MCU Programmer.
*D	5713151	04/26/2017	SHEA	Updated logo and copyright