AN211122

Getting Started with FM4 Firmware Development

Author: James Trudeau

Associated Part Portfolio: All FM4 parts

Related Application Note & Code Examples: See FM4 Portfolio Resources

AN211122 introduces you to the FM4 portfolio of 32-bit general-purpose microcontrollers based on the ARM® Cortex®-M4 processor core, which features DSP and Floating Point Unit (FPU) functions. This note provides an overview of hardware features and capabilities, firmware development, and technical resources available to you.

Contents

1 FM4 Portfolio Overview 1
2 Firmware Development 3
2.1 Peripheral Driver Library Overview 4
2.2 Software Development Overview 4
3 Programming Embedded Flash 6
3.1 Before You Begin 6
3.2 Using the FLASH MCU Programmer 7
3.3 Using the FLASH USB Direct Programmer 10
4 FM4 Portfolio Resources 12

1 FM4 Portfolio Overview

Cypress' FM4 is a portfolio of 32-bit, general-purpose and high-performance microcontrollers based on the ARM Cortex-M4 processor with FPU and DSP functionality. FM4 microcontrollers operate at frequencies up to 200 MHz and support a diverse set of on-chip peripherals for motor control, factory automation, and home appliance applications. The portfolio delivers low-latency, reliable, machine-to-machine communication required for network-computing technologies that advance design and manufacturing.

There are six series within the FM4 Portfolio. Each series represents multiple device packages with different capabilities. Table 1 lists the maximum value for some of the defining characteristics of each series.

Table 1. FM4 Portfolio Series

Series	S6E2C	S6E2D	S6E2G	S6E2H	MB9BFx6xM/N/R	MB9BFx6xK/L
Frequency (MHz)	200	160	180	160	160	160
Flash/SRAM(KB)	2048/256	384/256 512 KB VRAM for graphics display	1024/196	512/64	1024/128	512/64
Work Flash (KB)	-	-	-	32	32	32
GPIO	190	154	153	100	100	51
Base Timer	16	8	16	8	8	8
Multi-function Timer	3	1	2	3	2	2
Quadrature Counter	4	1	2	3	2	1
Graphics Controller	-	1	-	-	-	-
Multi-function Serial	16	8	10	8	8	6
USB	2	1	2	-	1	1
CAN	3	1	1	2	2	1

Series	S6E2C	S6E2D	S6E2G	S6E2H	MB9BFx6xM/N/R	MB9BFx6xK/L
Ether-MAC	1	-	1	-	-	-
DMA/DSTC	8/256	8/128	8/256	8/256	8/128	8/128
ADC Inputs	32	24	32	24	24	15
DAC	2	-	-	2	2	2
SD Card Interface	1	1	1	1	1	-
I2S	1	2	1	-	-	-

This portfolio is designed for applications that require advanced, high-speed computing performance such as:

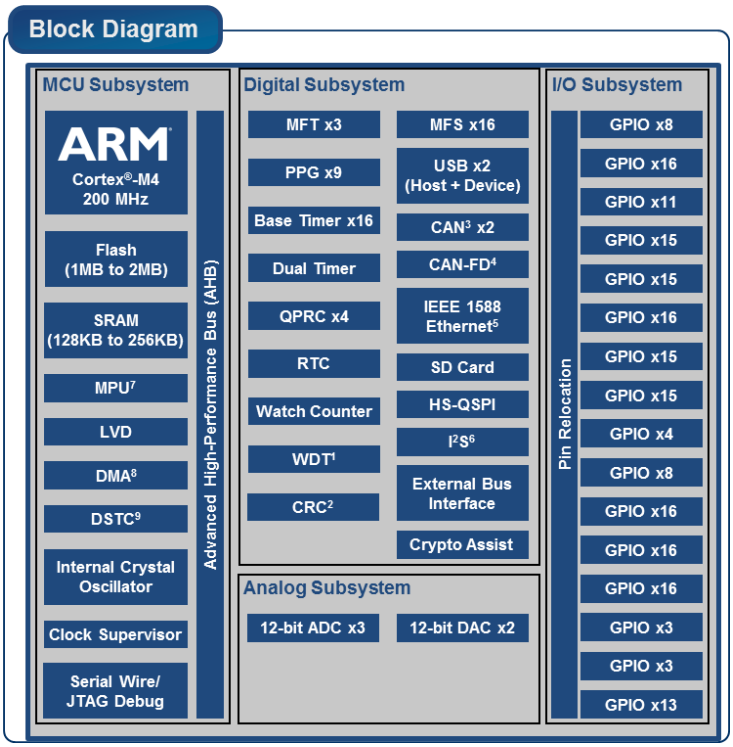
- Inverter-based home appliances such as washing machines and air conditioners
- Servomotors
- Programmable logic controllers and other industrial equipment
- Medical products
- Meters
- Printers

Key features include:

- Operating voltage: 2.7-5.5 V
- Low power consumption: 365 µA/MHz, 1.5 µA in real-time clock (RTC) mode
- 48- through 216-pin packages

Figure 1 shows the block diagram of the FM4 S6E2C series as an example. The FM4 portfolio provides a range of peripherals such as Ethernet, CAN, USB2.0, DMA, and A/D Converters. Peripheral support varies among the series in this portfolio. For details on variations within each series, such as pin packages, voltage range, or peripheral support, review the Product Selector Guide. Read AN203277 to learn more about hardware design considerations.

Figure 1. Block Diagram for the FM4 S6E2C Series



1. Watchdog Timer
2. Cyclical Redundancy Check
3. Controller Area Network
4. Controller Area Network with Flexible Data-Rate
5. Ethernet Communications with IEEE 1588 Precision Time Protocol (PTP) Standard
6. Inter-IC Sound
7. Memory Protection Unit
8. Direct Memory Access
9. Descriptor System Transfer Controller

2 Firmware Development

This section discusses the Cypress FM Peripheral Driver Library (PDL). The PDL is central to firmware development for all FM portfolios. The PDL simplifies software development for the extensive set of peripherals available. It reduces the need to understand registers and bit structures. You configure the library for the desired functionality, and then use API function calls to initialize and use a peripheral. In addition to the FM4 portfolio, the PDL supports Cypress FM0+ and FM3 processors and peripherals. Using the PDL makes it easier to port code from one portfolio to the other.

Developers who wish to work at the register level should also install the PDL. The PDL is where you get device-specific header files, startup code, configuration files, and IDE project files for every FM4 device. You can use these files with or without the PDL.

The PDL is provided as source code. Reviewing the PDL source code is a useful way to approach the detailed knowledge required to program a microcontroller at a low level. Combined with the review of the appropriate data sheet and peripheral manual, you can learn the information you need to use a peripheral. See the FM4 Portfolio Resources section of this document for links to the extensive technical documentation available.

Because the PDL is central to all FM portfolios, you will find in-depth information on firmware development in the *PDL Quick Start Guide*. This includes simple step-by-step instructions on how to build and run a PDL code example. The *PDL Quick Start Guide* is installed along with the PDL. It is also available separately at the Cypress PDL product page.

2.1 Peripheral Driver Library Overview

The PDL is a superset of all the code required to build any driver for any supported device. This superset design means:

- All APIs needed to initialize, configure, and use a peripheral are available.
- The PDL includes error checking to ensure that the targeted peripheral is present on the selected device.

The superset design means the PDL is useful across all devices irrespective of the available peripherals. This enables the code to maintain compatibility across platforms where peripherals remain present. If you configure the PDL to include a peripheral that is unavailable on the specified hardware, your project would fail at compile time, rather than at runtime. The PDL configuration logic knows the target processor and removes the peripheral register headers for unsupported peripherals from the build.

Before writing code to use a peripheral, consult the datasheet for the particular series or device to confirm support for the peripheral.

2.1.1 Getting and Installing the PDL

Download the PDL Installer from the Cypress PDL product page. Launch the installer, and follow the prompts.

2.1.2 PDL Structure

The PDL is organized into several folders as shown in Table 2.

Table 2. PDL Folder Structure

Path\Folder	Description
cmsis	cmsis header files
devices	For each device package: common header files configuration, startup, and project files for each IDE
doc	PDL documentation
driver	Driver source code and header files
examples	Code examples for each peripheral on each supported starter kit
utilities	Various utility files

When you use the PDL, typically the only files you modify are *pdl_user.h* and *main.c.*

2.2 Software Development Overview

Table 3 lists supported toolchains.

Table 3. Supported Toolchains

Vendor	Tool	Version
IAR Systems	Embedded Workbench	7.50.1 or higher
ARM Keil	µVision	5.17 or higher
Open Source/ARM	GCC ARM Embedded	4.9-d015-Q1-Update or higher
Atollic	TrueSTUDIO	5.4.1 or higher
iSystem	WinIDEA	9.12 or higher

2.2.1 Using PDL Code Examples

Beginning with PDL v2.1, the PDL includes code examples configured for particular starter kits. You find the code examples in the examples folder organized by starter kit. Each example demonstrates the basic initialization and configuration for a peripheral. Some peripherals have multiple examples.

Note: For step-by-step instructions on how to build and run a PDL project, see the *PDL Quick Start Guide,* installed along with the PDL. It is also available separately at the Cypress PDL product page.

Most Cypress FM4 starter kits, such as the FM4 S6E2GM Pioneer Kit, install a version of the PDL as part of the kit. The kits may install an older version of the PDL. Starter kit code examples work only with the version of the PDL used by the kit. They will not work with a different version of the PDL.

2.2.2 Writing Your Own Code Using the PDL

For detailed information on topics such as creating a custom project, configuring the PDL, configuring a peripheral, and using a peripheral,, see the PDL Quick Start Guide available in the *doc* folder in the PDL directory. It is also available separately at the Cypress PDL product page.

2.2.3 PDL API Documentation

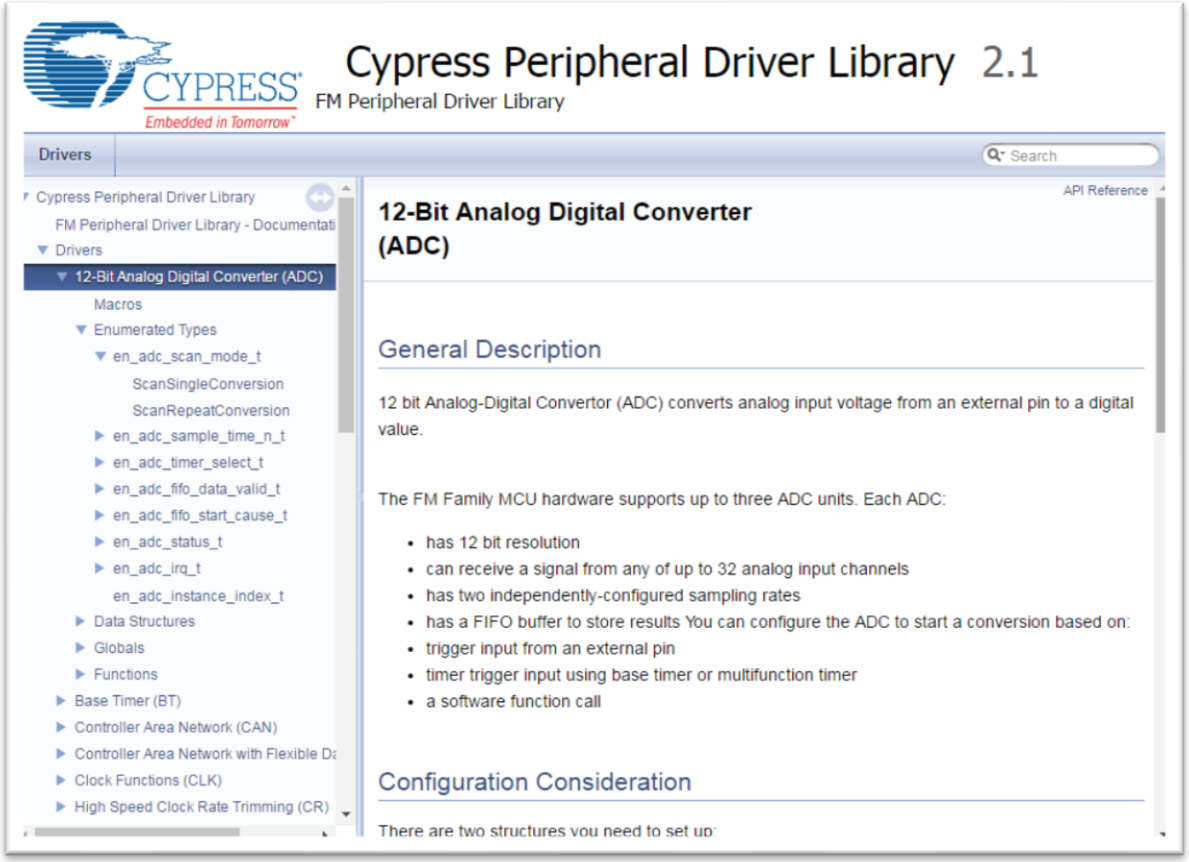
PDL API documentation is HTML-based and generated from the source code. The PDL installer puts the documentation here:

<PDL directory>\doc\pdl_api_reference_manual.html

The first time you open the documentation, make a bookmark in your browser for easy access.

In the documentation, use the left navigation menu to find the information you need. The **Drivers** section lists all the information for a particular peripheral. Expand any driver to see the macros, types, structures, global variables, and API functions. Figure 2 shows the documentation home page.

Figure 2. PDL Documentation



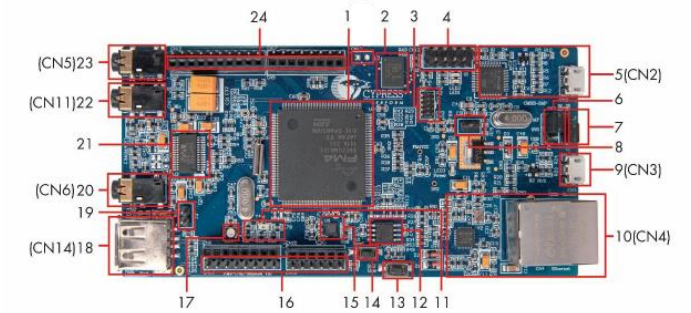
See the FM4 Portfolio Resources section of this document for additional links to helpful information.

3 Programming Embedded Flash

Full information on firmware development for FM portfolio microcontrollers is in the *PDL Quick Start Guide*. Most IDEs are capable of programming embedded flash. However, a flash programmer may be your preferred or only solution in some cases. This section shows you how to program embedded flash using either a serial or a USB connection. The USB connection requires USB support on the target.

As an example, these instructions use the S6E2GM processor found in the FM4 S6E2GM Pioneer Kit. Figure 3 has a key to the components on the hardware.

Figure 3. The S6E2GM-Series Pioneer Kit Board



1. Cypress FM4 S6E2GM8J0A MCU
2. 4-Mb SRAM
3. 10-pin JTAG connector (CN1)
4. Multicon Interface connector (CN12,CN13)
5. MB9F312K Programmer and Debugger (CMSIS-DAP) (CN2)
6. Power supply source select (J4)
7. Serial programming mode select (J3)
8. Programming mode jumper of MB9F312K (J1)
9. USB device connector (CN3)
10. Ethernet PHY and RJ45 connector (CN4)
11. Phototransistor
12. 32-Mb Quad-SPI NOR Flash
13. User button (SW2)
14. Reset button (SW1)
15. Accelerometer
16. Additional GPIO headers (CN15-CN18)
17. RGB LED (LED4)
18. USB host connector (CN14)
19. Programming mode jumper of S6E2GM (J2)
20. Line-in jack (CN6)
21. Stereo codec
22. Microphone jack (CN11)
23. Headphone jack (CN5)
24. Arduino™ interface (CN7-CN10)

If you are not using this kit, you must modify the instructions to fit your specific target hardware. Check the documentation provided with your board for jumper configuration and other details.

3.1 Before You Begin

Ensure that you have a programmer. These instructions cover the tools listed here.

- FLASH MCU Programmer for FM0+/FM3/FM4
- FLASH USB Direct Programmer

There are two ways to use these flash programmers: single-step or automatic programming (full operation). Note that only single-step works for secured flash devices that need chip erase. In this example, we use automatic programming.

You also need a file to download. The file format must be either Motorola S-Record or Intel-HEX. This example uses a Motorola S-Record file provided with the kit.

When you build code in an IDE, you may be able to generate an S-Record or Intel-HEX format file. Consult the documentation for your IDE. For example, in IAR Embedded Workbench, use the **Project** > **Options** > **Output Converter** panel. In Keil µVision, use the **Project** > **Options for Target** > **Output** panel.

3.2 Using the FLASH MCU Programmer

These instructions assume you have downloaded and installed the starter kit so you have access to the required S-Record file. If not, locate an S-Record or Intel-HEX file, and use that file instead.

1. **Configure the jumpers.**

   Make sure that the jumpers on the FM4 S6E2G-Series Pioneer board are placed according to Table 4.

   Table 4. Jumper Settings for S6E2GM programming by FLASH MCU Programmer.

Jumper	Default	Program by Serial	Purpose
J1	Open	Open	Sets MB9AF312K (CMSIS-DAP) to run mode.
J2	Open	Closed	Sets S6E2GM to programming mode.
J3	Pin 1 to Pin 2	Pin 1 to Pin 2	Sets for UART programming mode.
J4	Pin 1 to Pin 2	Pin 1 to Pin 2	Get power from the CMSIS-DAP connector

2. **Provide power to the board.**

   Connect the USB cable to the CN2 connector. The Power LED (LED3) should be lit (green). See Figure 3 for the location of the correct connector.

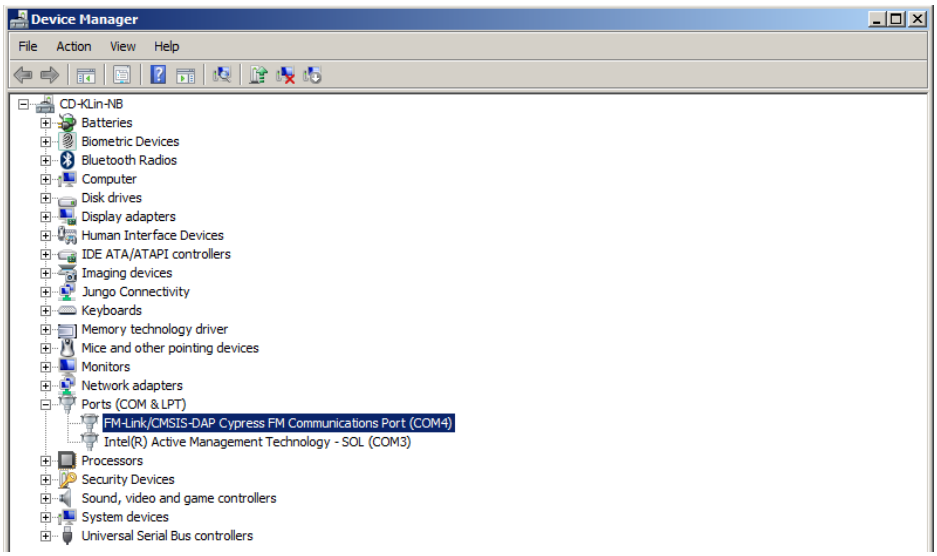
3. **Identify the COM port in use.**

   You need to know which COM port your board is connected to. You will use this number to configure the Flash programmer.

   If you use the Cypress Serial Port Viewer and Terminal tool, you will see a popup notification that contains this information as you connect the board. If you don't see it, or don't know the COM port, open the Device Manager and look for **Ports (COM & LPT).** You should see an entry for FM-Link/CMSIS-DAP. The COM port is listed at the end of that entry, as shown in Figure 4.

   Remember the number.

   Figure 4. Identify the Com Port In Use



4. **Launch the FLASH MCU Programmer.**

   You can do this from the Start menu, using this path:

   Start Menu > All Programs > Cypress > FLASH MCU Programmer > FM0+ FM3 FM4

5. **Configure the programmer.**

   In this and the next step you configure the programmer for the target device. See Figure 5.

   A. Set **Target MCU** to S6E2GM8H/J.

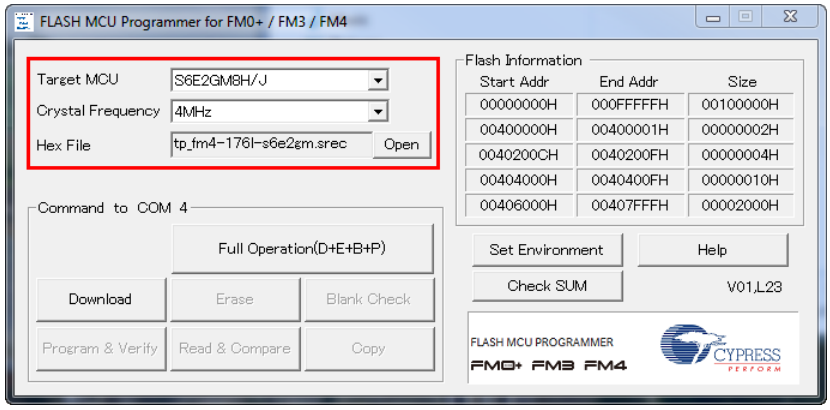
   A. Set Crystal Frequency to 4MHz.

   B. Set **Hex File** to the file you wish to flash to the board.

      For purposes of this example we use *tp_fm4-176l-s6e2gm.srec*. This file restores the starter kit board to its initial state.

      The S-Record file is here: *<Kit Directory> \Firmware\Demo Projects\Test_Demo_Code.*

Figure 5. Configure the Programmer

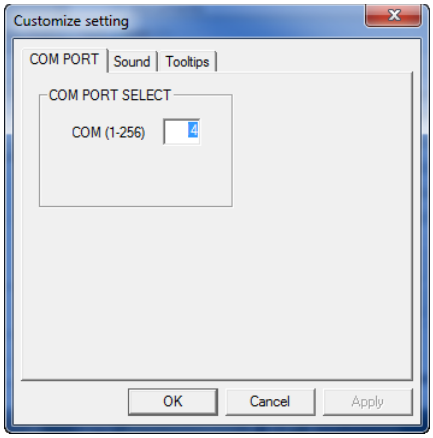


6. **Set the COM Port in the programmer.**

   A. Click Set Environment.

   C. Set **COM (1-256)** to the value you saw in the Device Manager. See Figure 6.

Figure 6. Set the COM Port in the Programmer

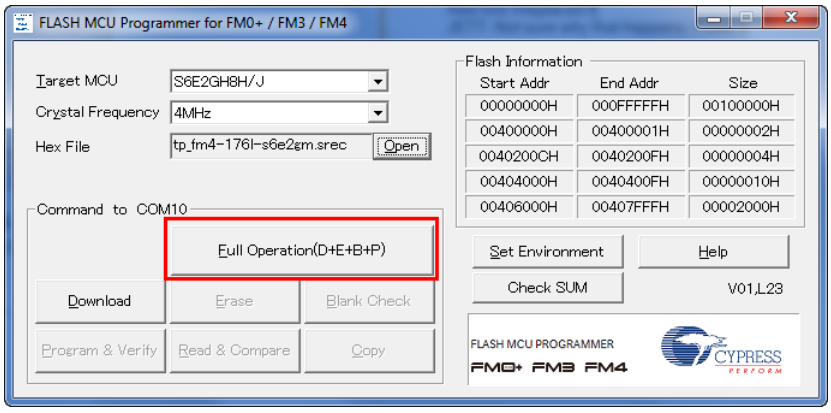


7. **Program the FLASH.**

   You will need to reset the board as it is programmed. See Figure 7 and Figure 8.

   A. Click **Full Operation (D+E+B+P)** button to start programming. (**Note**: Full Operation does not work on secured flash. You must use single steps.)

Figure 7. Click the Full Operation Button

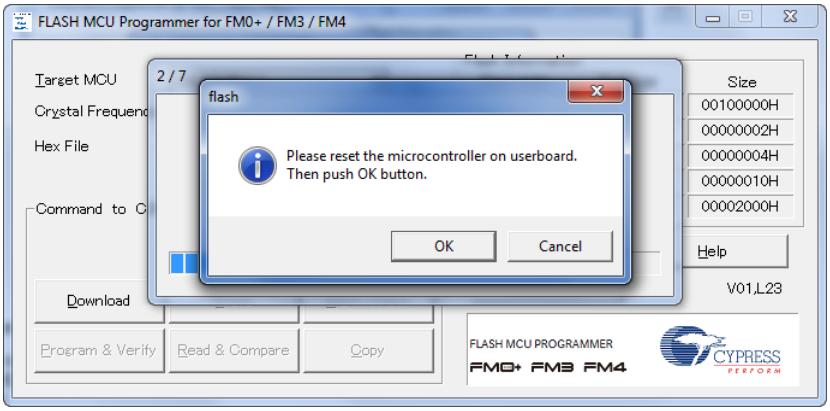


The programming process begins, and a dialog window appears, as shown in Figure 8.

D. Reset the board.

Press the reset switch (SW1) on the board, and then click OK.

Figure 8. Reset the Microcontroller



The programmer downloads the selected file to the board.

8. Restore the board to normal operation.

When done, restore the jumpers to their original configuration, or to default values as shown in Table 4.

To confirm success, use the Serial Port Viewer tool to connect to the board and run the demo code. Full instructions are in the FM4 S6E2G-Series Pioneer Kit Guide. Click **Help** for any issues or errors encountered during programming.

3.3 Using the FLASH USB Direct Programmer

These instructions assume you have downloaded and installed the starter kit so you have access to the required S-Record file. If not, locate an S-Record or Intel-HEX file, and use that file instead. The USB connection requires USB support on the target.

1. **Configure the jumpers.**

   Make sure the jumpers on the FM4 S6E2G-Series Pioneer board are placed according to Table 5.

   Table 5. Jumper Settings for S6E2GM programming by FLASH USB DIRECT Programmer

Jumper	Default	Program by USB	Purpose
J1	Open	Open	Sets MB9AF312K (CMSIS-DAP) to run mode.
J2	Open	Closed	Sets S6E2GM to programming mode.
J3	Pin 1 to Pin 2	Pin 2 to Pin 3	Sets for USB programming mode.
J4	Pin 1 to Pin 2	Pin 2 to Pin 3	Get power from the USB connector

2. **Provide power to the board.**

   Connect the USB cable to the CN3 connector. The Power LED (LED3) should be lit (green). See Figure 3 for the location of the correct connector.

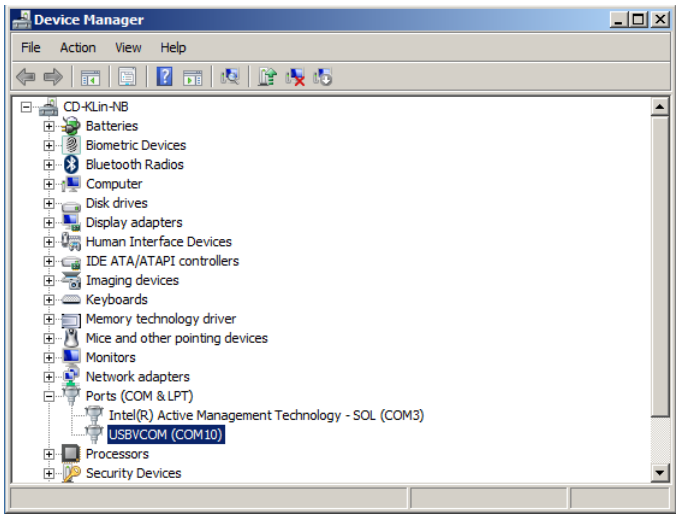
3. **Identify the COM Port in use.**

   You need to know which COM port your board is connected to. You will use this to configure the Flash programmer.

   If you use the Cypress Serial Port Viewer and Terminal tool, you will see a popup notification that contains this information as you connect the board. If you don't see it, or don't know the COM port, open the Device Manager and look for **Ports (COM & LPT).).** You should see an entry for USBVCOM. The COM port is listed at the end of that entry, as shown in Figure 9.

   Remember the number.

   Figure 9. Identify the COM Port In Use



4. **Launch the FLASH USB DIRECT Programmer.**

   You can do this from the Start menu, using this path:

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

5. **Configure the programmer.**

   In this step, you set up the programmer for the target device. See Figure 10.

   A. Set **Target MCU** to S6E2GM8H/J.

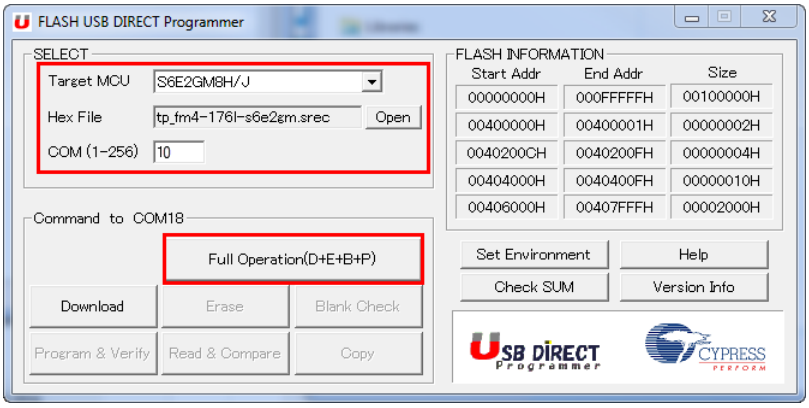
   E. Set **Hex File** to the file you wish to flash to the board.

   For purposes of this step we use *tp_fm4-176l-s6e2gm.srec*. This file restores the kit to its initial state.

   The S-Record file is here: *<Kit Directory>\Firmware\Demo Projects\Test_Demo_Code*.

   F. Set **COM (1-256)** to the value you saw in the Device Manager.

Figure 10. Configure the Programmer



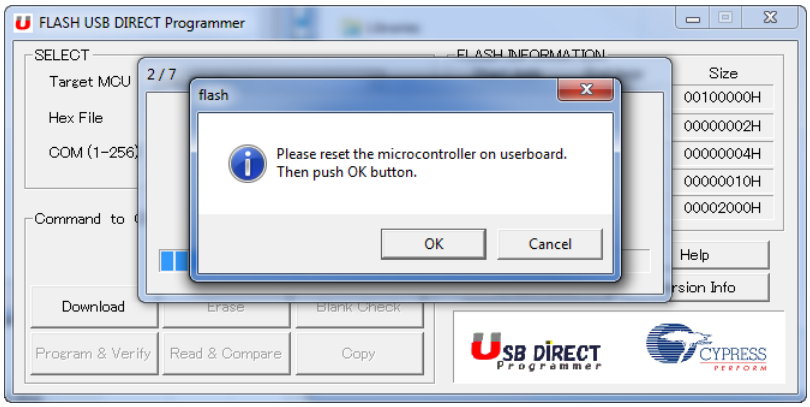
6. **Program the Flash.**

   You will need to reset the board during the process.

   A. Click **Full Operation (D+E+B+P)** button to start programming. (**Note**: Full Operation does not work on secured flash. You must use single steps.) The programming process begins, and a dialog window appears.

   G. Reset the board. Press the reset switch (SW1) on the board, and then click OK, as shown in Figure 11.

Figure 11. Reset the Microcontroller



7. **Restore the board to normal operation.**

   When done, restore the jumpers to their original configuration, or to default values as shown in Table 5.

   To confirm success, use the Serial Port Viewer and Terminal tool to connect to the board and run the demo code. Full instructions are in the FM4 S6E2G-Series Pioneer Kit Guide.

4 FM4 Portfolio Resources

Cypress provides many resources to help you learn about and become productive with the FM4 portfolio. Use Table 6 to identify and choose the resource you want based on where you are in the design process.

Table 6. FM4 Portfolio Resources Navigator

I Want To	Resources
Evaluate FM4	Read this document. Watch the FM4 introductory video. Explore the FM4 product page on the Cypress website. Purchase an FM4 Starter Kit. Click the Kits tab on the product page. Refer to FM4 Datasheets. Read AN202487- Differences Among FM0+, FM3, and FM4 Families
Select an FM Part	Download and review the Product Selector Guide. Read AN202487 - Differences Among FM0+, FM3 and FM4 Families
Learn About Hardware Design	Read AN203277 – FM 32-bit Microcontroller Family Hardware Design Considerations
Learn About Available Tools	IAR Embedded Workbench Keil µVision IDE iSYSTEM winIDEA Atollic TrueSTUDIO GCC ARM Embedded
Learn About the Peripheral Driver Library and Code Examples	Purchase an FM4 Starter Kit. Read the Peripheral Driver Library Overview in this document. Download the PDL and read the PDL Quick Start Guide. Read the *Build and Run a PDL Project* section of the *PDL Quick Start Guide*. Explore the PDL code examples installed with the PDL
Learn about particular FM4 peripherals	Search for an FM4-related application note. Some examples include: AN202488 – Servo Motor Speed Control AN203980 - S6E2Cx Series Over the Air Update AN99218 - Multi Function Serial Interface of FM MCU AN204468 – FM4 I2S USB MP3 Player Application
Develop Low-level Software for FM4	Read the *Creating a Custom Project* section of the *PDL Quick Start Guide*. Use project files and startup code from the PDL *devices* folder. Use PDL source code to see low-level programming techniques Refer to FM4 Datasheets. Use the FM4 Peripheral Manuals as a technical reference.
Learn About Flash Programming	Get a Flash programmer. FLASH MCU Programmer for FM0+/FM3/FM4 FLASH USB Direct Programmer Read the Programming Embedded Flash section of this document. Read the Flash Programming Manual for your FM4 series: MB9Bx S6E2Cx S6E2Dx S6E2Gx S6E2Hx Read AN204438 - How to Setup Flash Security for FM0+, FM3 and FM4 Families.

About the Author

Name:	James Trudeau
Title:	Senior Principal Application Engineer
Background:	Jim Trudeau started at Cypress in 2015, continuing a long career in technical support, technical communication, and developer relations in the software tools and semiconductor industry.

Document History

Document Title: AN211122 - Getting Started with FM4 Firmware Development

Document Number: 002-11122

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	5127784	JETT	02/23/2016	New application note
*A	5347663	JETT	02/08/2016	Update to new AN template. Update table 1.1, portfolio features (from Cypress website) Updated for information related to PDL v2.1, in sections 1 and 2 Deleted former section 3, Build and Run a PDL Project (information is in the PDL Quick Start Guide) Updated hyperlinks throughout document to current resources.
*B	5465971	JETT	10/07/2016	Fixed the link to the FM4 starter kits Updated template
*C	5727940	BENV	05/05/2017	Updated logo and copyright
*D	6450467	JETT	01/18/2019	Updated template

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

Arm® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6 MCU

Cypress Developer Community

Community | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

© Cypress Semiconductor Corporation, 2016-2019. 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 a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (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, and (2) 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 for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. No computing device can be absolutely secure. Therefore, despite security measures implemented in Cypress hardware or software products, Cypress does not assume any liability arising out of any security breach, such as unauthorized access to or use of a Cypress product. In addition, the products described in these materials may contain design defects or errors known as errata which may cause the product to deviate from published specifications. To the extent permitted by applicable law, 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 you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You 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.