Copyright @ 2019

EM9304 SOC SDK GETTING STARTED GUIDE

Product Family: **EM9304**

Part Number: EM9304 Revision B

Keywords: EM9304, SOC, SDK, DVK, Software, Development, Getting Started

1. INTRODUCTION

The EM9304 System On Chip (SOC) Software Development Kit (SDK) and the Synopsys Metaware Integrated Development Environment provide a flexible and convenient way to develop applications using the EM9304. This application note will help you start developing applications for the EM9304 device using the EM9304 SOC SDK. This application note contains a step-by-step guide on how to set up the required development tools, build the example projects, execute an example project with the debugger, and interact with the EM9304 example application with a Smartphone.

2. REQUIREMENTS

The following hardware components are required for this tutorial:

- PC running Windows 7, 8, 8.x, or 10
- EM9304 SOC DVK (EMDVK9304SOC version 4.0 or newer)
- Smartphone running iOS or Android

The following software components are required for this tutorial:

- DesignWare ARC Metaware Toolkit Lite (Version **2017.12** or newer)
- EM9304 SOC SDK (Version **6.0** or newer)
- Alpwise Mobile App for iOS or Android (Build version **7.3.1** or newer)
- EM9304 Configuration Editor (Version **1.0.0** or newer as installed with this SDK)

3. OVERVIEW

This tutorial covers the following topics.

1. Installing the development tools and the EM9304 SOC SDK.
2. Reviewing EM9304 SOC SDK organization.
3. Setting up the example application project files.
4. Building on the FindMe example project.
5. Preparing the EM9304 DVK for the example FindMe application.
6. Connecting the EM9304 DVK to a computer.
7. Using the EM9304 Configuration Editor to update the EM9304 firmware with the EM9304 meta SOC patch.
8. Using the EM9304 Configuration Editor to program a configuration patch into the One-Time Programmable (OTP) memory.
9. Launching the debugger to run the example FindMe application.
10. Installing the Alpwise Bluetooth Low Energy app on a smartphone.
11. Interacting the example FindMe application with the smart phone.

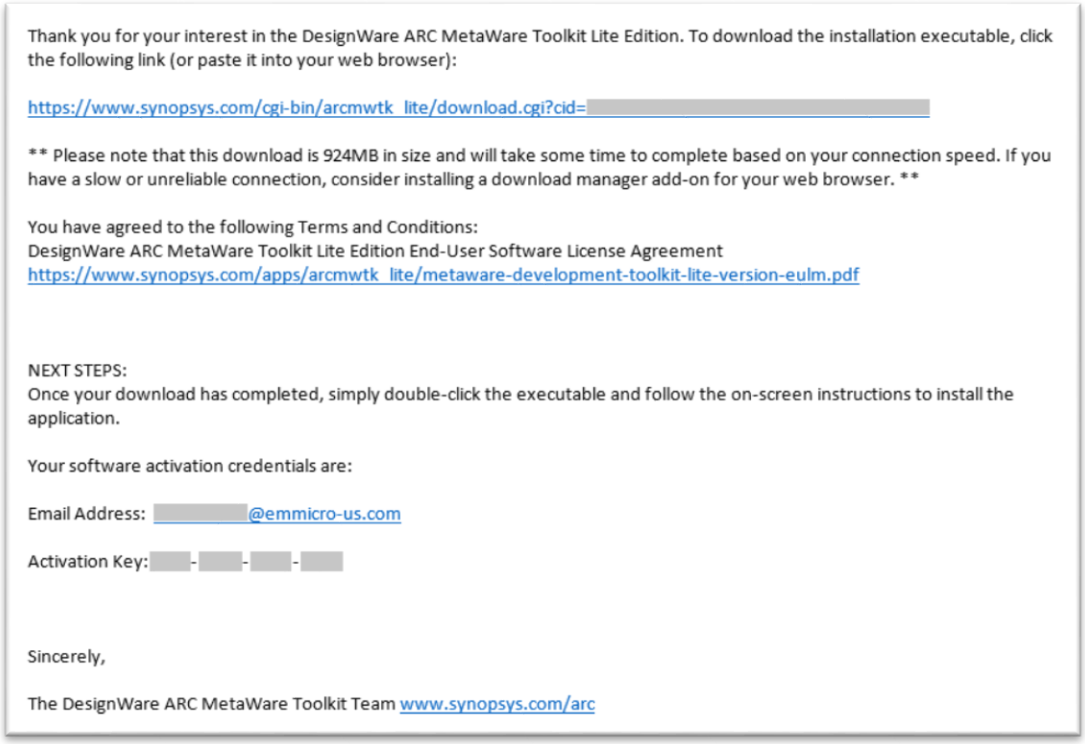
EM Microelectronic may at its discretion issue firmware updates to improve and correct the ROM firmware. The EM9304 meta SOC patch is critical to the proper operation of the EM9304 when operating in SOC mode. It is strongly recommended that the meta SOC patch be programmed into the EM9304 prior to beginning application development as described in the instructions below. Please see the release notes for a description of EM9304 meta SOC patch.

4. DEVELOPMENT TOOL INSTALLATION

The complete development environment for the EM9304 includes the Metaware Toolkit Lite integrated development environment and the EM9304 SOC SDK.

4.1 INSTALLING METAWARE LITE

All EM9304 software and tools are included in the EM9304 SOC SDK, but the Metaware Toolkit Lite software must be downloaded from the Synopsys website and installed separately. The Metaware Lite software can be downloaded at https://www.synopsys.com/cgi-bin/arcmwtk_lite/reg1.cgi. You will be prompted to register and you will receive a registration code and download link through email. The email message from Synopsys should arrive within minutes of registering and will look similar to the Figure 1 below.



Thank you for your interest in the DesignWare ARC MetaWare Toolkit Lite Edition. To download the installation executable, click the following link (or paste it into your web browser):

https://www.synopsys.com/cgi-bin/arcmwtk_lite/download.cgi?cid=

** Please note that this download is 924MB in size and will take some time to complete based on your connection speed. If you have a slow or unreliable connection, consider installing a download manager add-on for your web browser. **

You have agreed to the following Terms and Conditions:
DesignWare ARC MetaWare Toolkit Lite Edition End-User Software License Agreement
https://www.synopsys.com/apps/arcmwtk_lite/metaware-development-toolkit-lite-version-eulm.pdf

NEXT STEPS:
Once your download has completed, simply double-click the executable and follow the on-screen instructions to install the application.

Your software activation credentials are:

Email Address: @emmicro-us.com

Activation Key: - - -

Sincerely,

The DesignWare ARC MetaWare Toolkit Team www.synopsys.com/arc

Figure 1 Registration email message Metaware Toolkit Lite

Once the download is complete, execute the executable and following the instructions. Note that the Metaware Toolkit Lite must be installed in a folder where the path does not contain any spaces. For example, Metaware can be installed at “C:\Tools\MetawareLite”, but cannot be installed in “C:\Program Files (x86)\MetawareLite”. Metaware Lite should be installed with user level permissions rather than administrator level permissions.

4.1 INSTALLING THE EM9304 SOC SDK

The EM9304 SOC SDK can be downloaded from the EM Developer website forum section at https://forums.emdeveloper.com/index.php. The EM Developer website contains a support forum as well as links to applications notes and software tools for the EM9304 DVK. Registration is simple and is required to access the EM9304 BLE 4.2 Transceiver discussion thread.

Double-click the EM9304 SOC SDK installation executable to start the installation. The Welcome page of the installation setup wizard will be displayed. Click next to continue to the Component Selection page as shown in Figure 2.

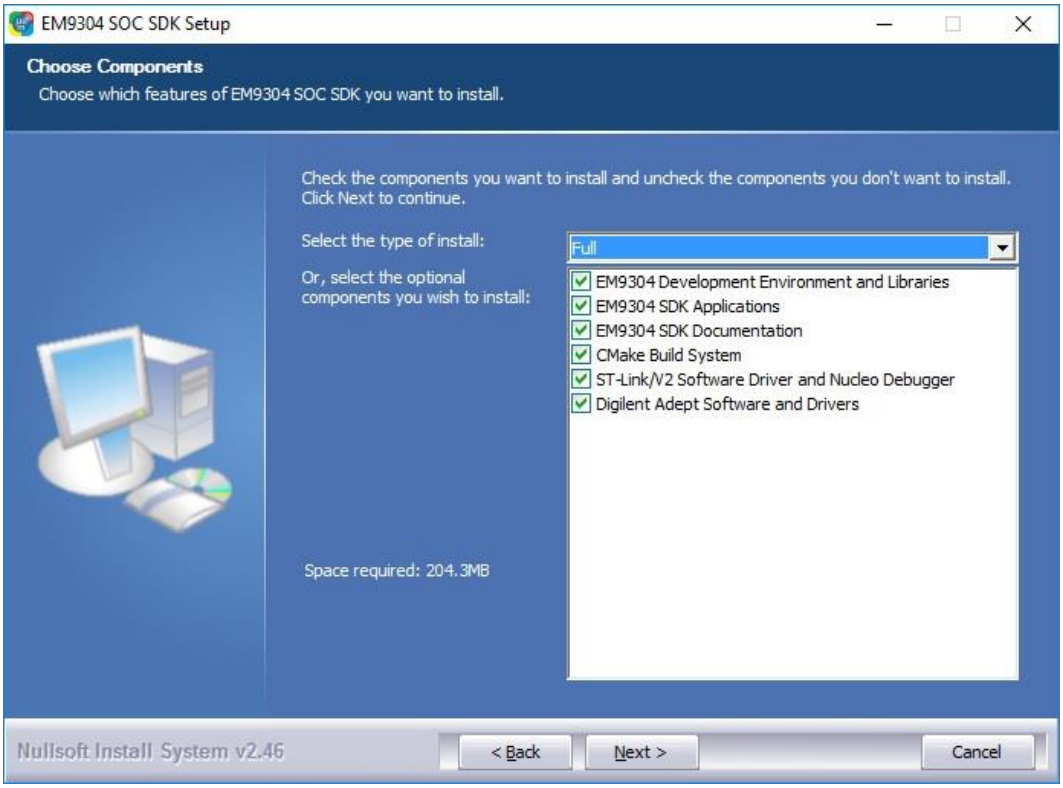


Figure 2 Installation component selection page.

The Component Selection page allows you to customize the development environment installation, which is generally only used to deselect components that may already be installed on your PC. For instance, if you have already installed the Digilent Adept driver software, you could deselect that item from the installation list. In most cases, the Full installation type should be chosen. If the installation detects that CMake is already installed, the CMake Build System will not be selected.

This tutorial assumes the Full installation is installed.

The EM9304 Development Environment and Libraries are required for application development. The component contains the build environment, software libraries, and example projects. The EM9304 SDK applications are utilities developed for the DVK and allow you to issue HCI or ACI commands to the EM9304 with the EM9304 Connector application and to configure the EM9304 with the EM9304 Configuration Editor. The EM9304 SDK documentation contains the EM9304 software API documentation, the DVK schematic, the implementers guide for the Bluetooth stack, and other reference documentation. The CMake Build System is required for the software build environment. The ST Link Software and Drivers are required to allow your PC to communicate with the DVK through a virtual COM port, and the Digilent Adapt Software and Drivers are required to use the Metaware Lite debugger.

Click Next to continue to the Application Installation Location page. This page allows you to select the destination location for the SDK applications.

Click Next to continue to the Project Installation Location page. This page allows you to select the destination location for the SDK development environment and libraries, and the SDK documentation.

Click Next to continue to continue to the Start Menu Location page where you can specify where the application and documentation shortcut links will be placed.

Click Next to begin the SDK installation. After the EM9304 components have been installed, separate installers will be launched to install the following third-party applications

- CMake
- ST Link software and drivers
- Diligent Adept software and drivers.

Follow the instructions provide for each installation procedure.

Note, during the CMake installation process, select "Add CMake to the system PATH for all users" when prompted in the Install Options screen. Prior to launching the CMake Setup Wizard, you will be reminded to set the option.

Once the setup process is complete, and all required third party software has been installed, the Setup Complete page will be displayed as shown in Figure 3.

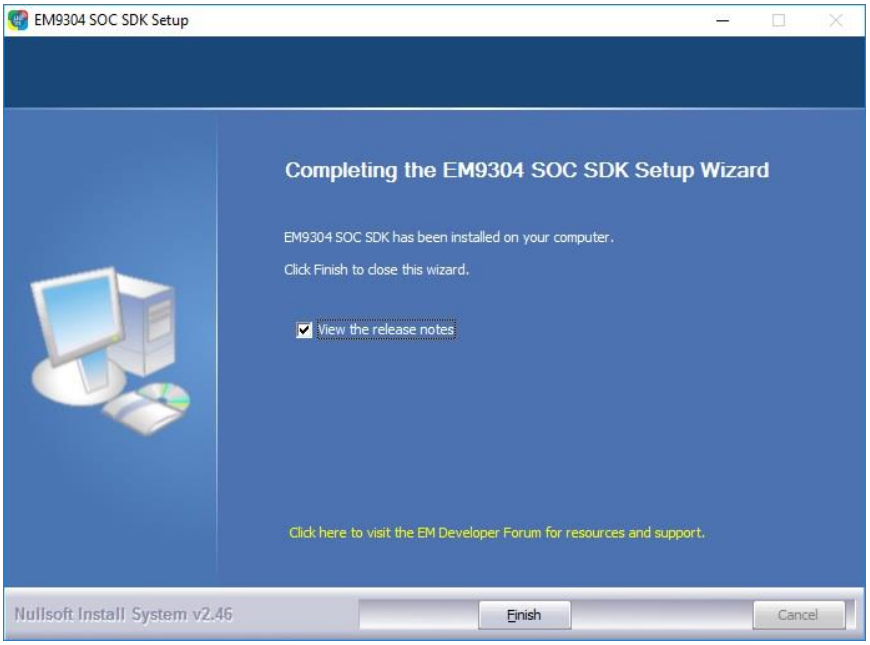


Figure 3 Setup complete page.

5. SOC SDK ORGANIZATION

The SDK contains documentation, reference material for the Metaware development tools, EM9304 Bluetooth utilities, example applications, and other material in addition to the development files and libraries. The SDK is structured as described below. The description does not list all material, but identifies those items that are particularly helpful when beginning development with the EM9304.

- documents
  - Alpwise
    - Bluetooth stack and profile documentation
    - Core stack (Bluetooth low energy stack) implementer's guide
  - EM Microelectonics
    - EM9304 software development API in HTML format
    - EM9304 DVK layout images and schematic
- sw
  - ide_project.bat (DOS batch file to setup the example project files)
  - ide_project_clean.bat (DOS batch file to clean the build environment)
  - CMakeLists.txt (build file where application directories are listed)
  - cmake (build environment)
  - common (common include files)
  - libs (project include files and libraries)
  - projects (example applications)
    - See the ReadMe.txt file for a description of example applications
  - rom
    - library and symbol files for the ROM software

6. SETUP THE METAWARE LITE WORKSPACE

Metaware Lite is based on the Eclipse IDE platform, which organizes application development into Workspaces and Projects. Typically, a Project is setup for each application. The project contains the source files and build settings create the application program file that is executed on the EM9304. The EM9304 SOC SDK contains four example projects. A workspace provides a way group one or more projects. This tutorial groups the four examples into projects and the projects into a single workspace.

Create the Metaware workspace with the following steps.

1. To minimize the length of this tutorial, the project files are created for you by going to the Windows Start menu, "All Apps" ("All Programs" in Windows 7), and clicking **EM Microelectronic EM9304 SOC SDK→Setup Project Files**. This Start menu shortcut executes a batch file that can also be executed from a DOS command prompt. The batch file is located at C:\[project-install-dir]\sw\ide_project.bat. A DOS command window will be displayed as shown in Figure 4. Close the DOS command window.

   Anytime a new project is added to the Metaware IDE Workspace, rerun the Setup Projects Files batch file to add EM-specific development files to the new project.

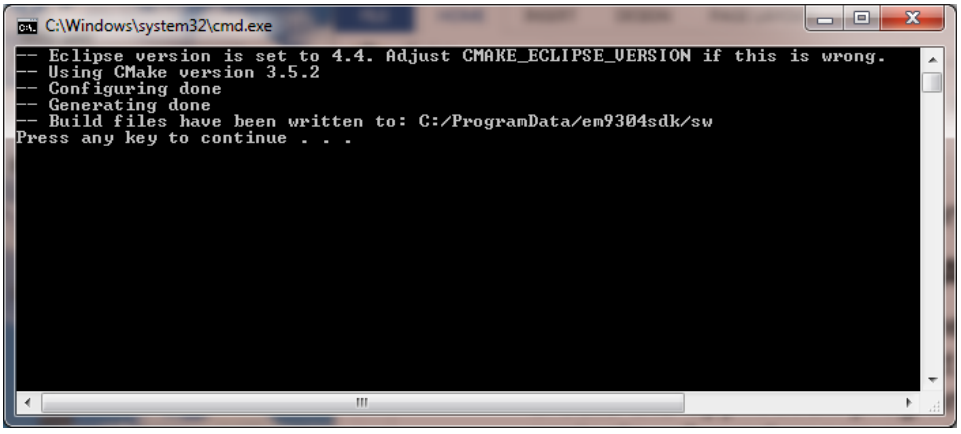
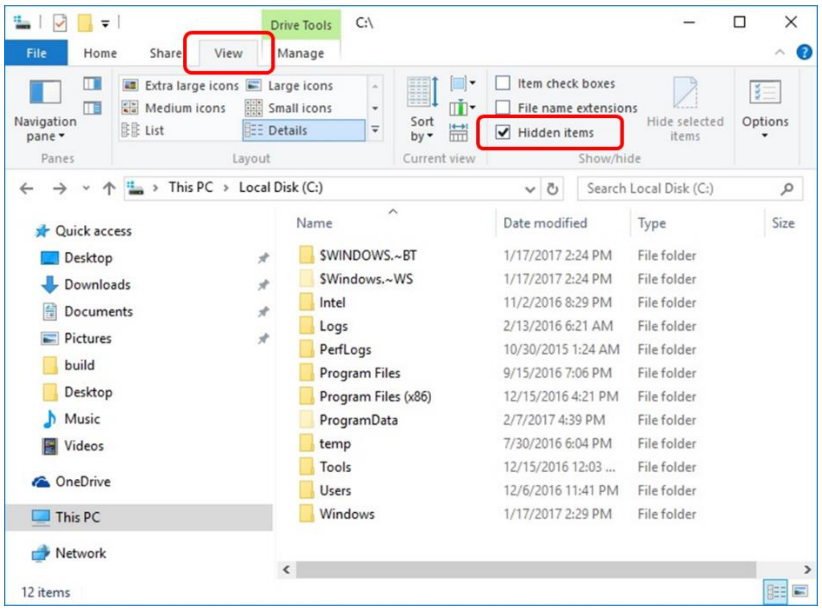


Figure 4 DOS command window after project creation.

2. If you installed the SOC SDK project files in the ProgramData folder, ensure you are able to view this directory because some versions of Windows hide this folder. If you did not install the project files into the ProgramData folder, skip this step.
   a. Under Windows 8 and 10, check the **Hidden Items** checkbox located on the **View** tab of the ribbon menu.



b. Under Windows 7, make this folder visible by select **Tools** then **Folder options…** from the main menu of a Windows Explore window.

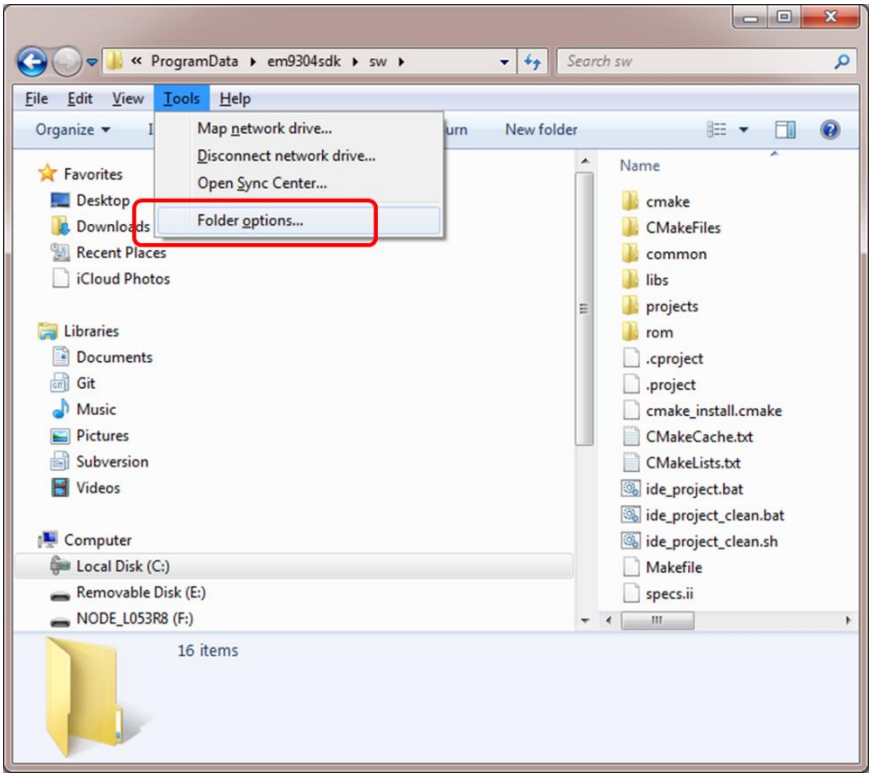


Figure 5 Windows Explorer folder options.

c. In the Folder Options dialog window, select Show hidden files, folders, or drivers, and click OK. Now, the ProgramData folder will be visible.

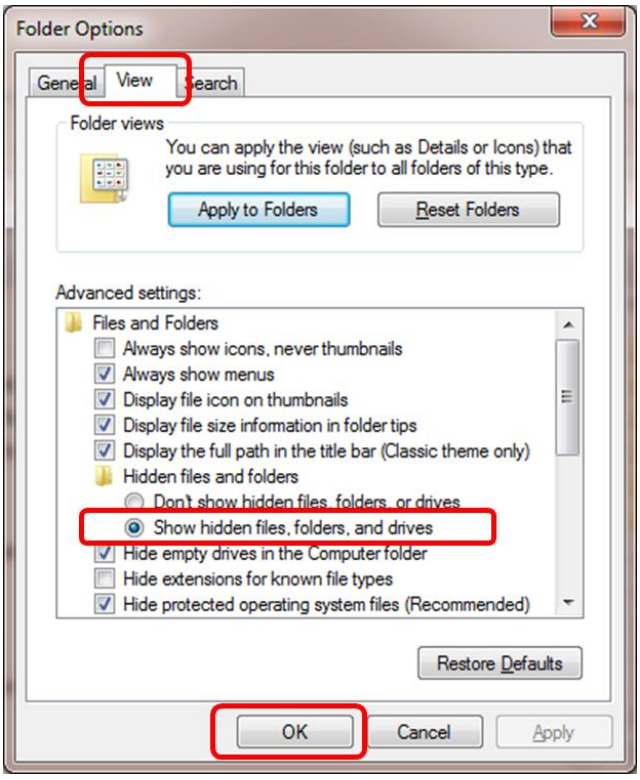


Figure 6 Show hidden files and folders.

3. Launch Metaware IDE Lite from the Windows Start menu.
4. Once Metaware IDE Lite completes the loading process, the Metaware IDE Lite Welcome window will be displayed as shown in Figure 7. Click **Start Working**.

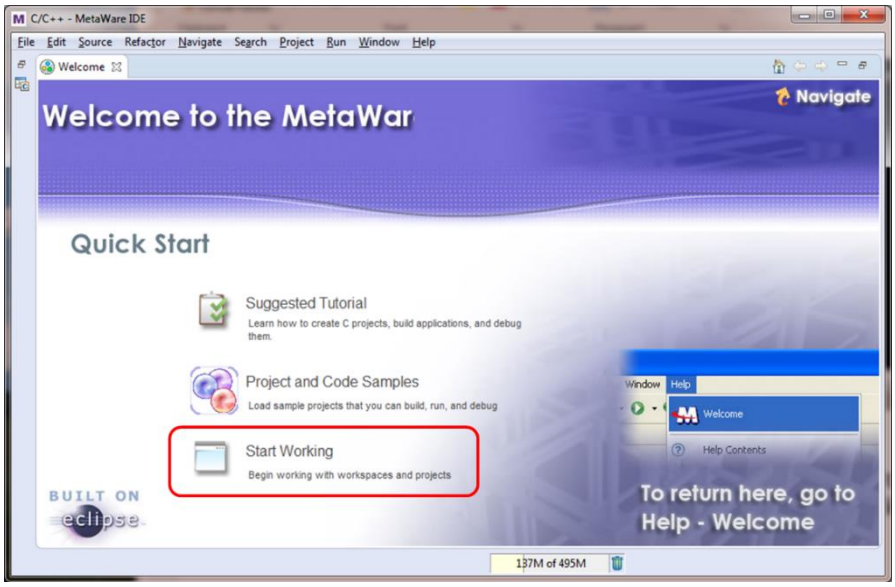


Figure 7 Metaware IDE Lite Welcome window.

5. Click **Create a New Workspace**.

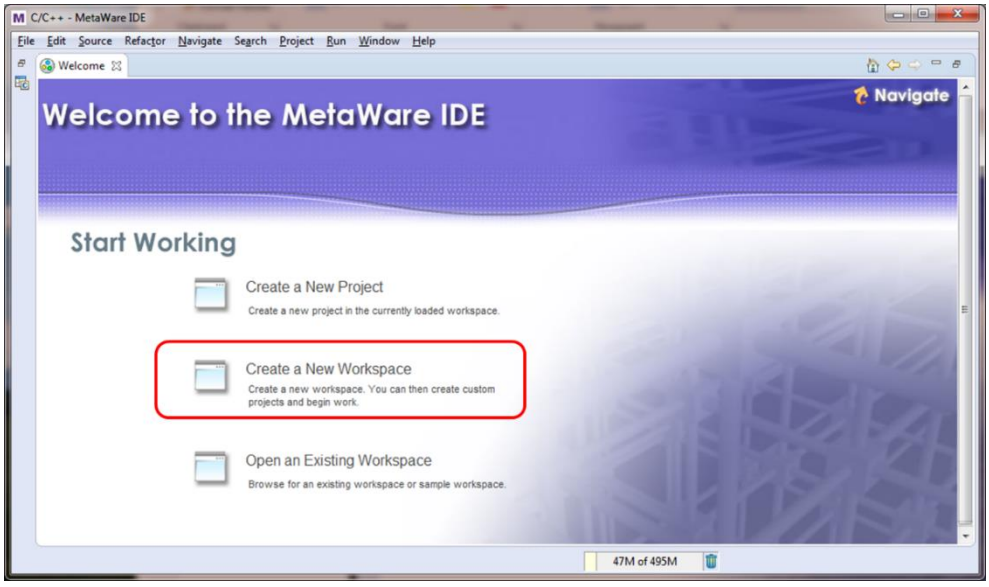


Figure 8 Create a New Workspace.

6. Click **Browse** in the Workspace Launcher window.

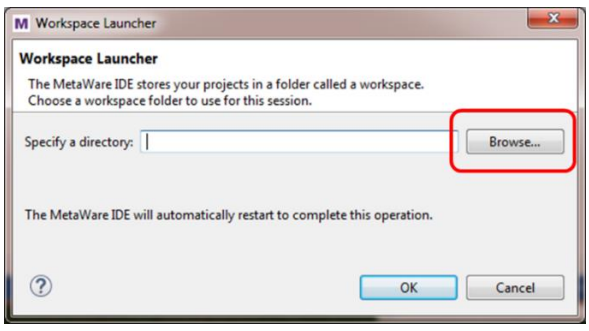


Figure 9 Workspace Launcher window.

7. Browse to the **em9304sdk** folder of the project files destination folder. The default location of sw folder is C:\ProgramData\em9304sdk. Select the **em9304sdk** folder and click **OK**.

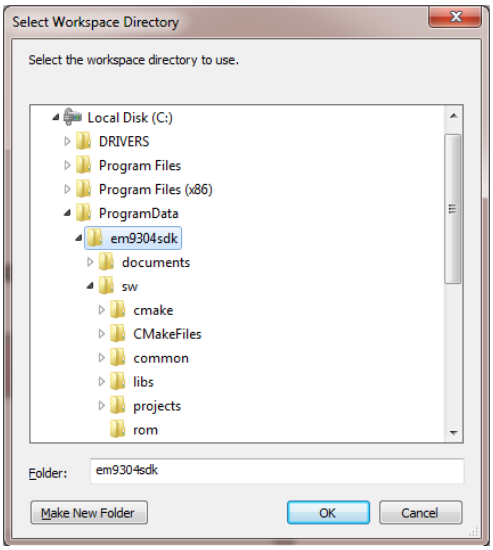


*Figure 10 Select the **em9304sdk** folder in the Workspace Directory window.*

8. Click **OK** in the Workspace Launcher window. Note the **Specify a directory** entry box shows the **em9304sdk** folder. Note that the workspace destination folder cannot be in the same folder as the projects folder.

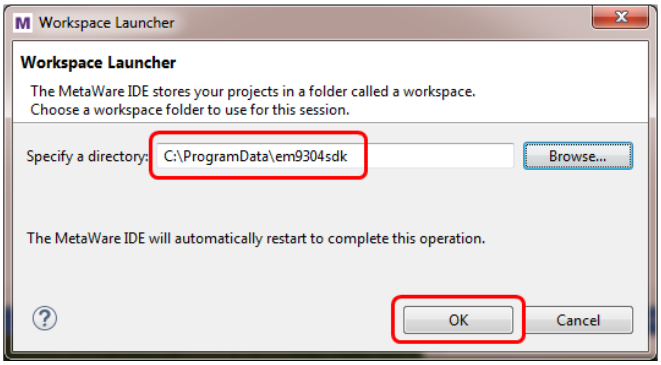


*Figure 11 Workspace Launcher with **em9304sdk** folder specified.*

9. Metaware IDE Lite will restart, and display an empty workspace. The next step is to import the project files into the newly created workspace. Begin by clicking **File** and **Import…**

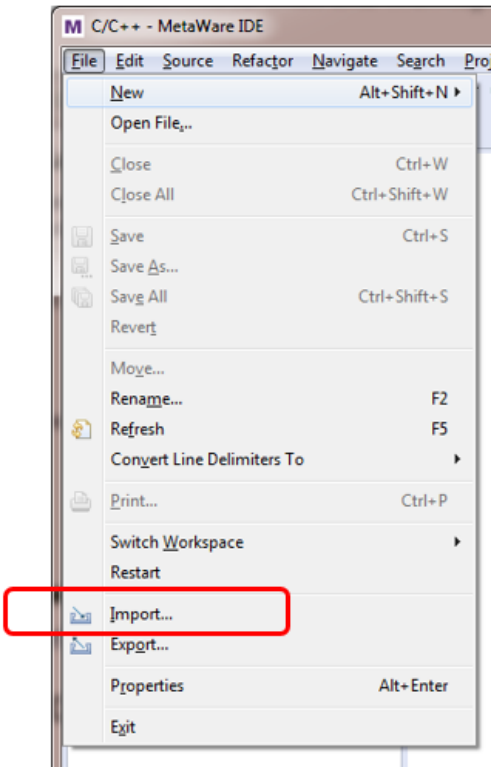


Figure 12 Import the project files.

10. Click **General**, **Existing Projects into Workspace**, and click **Next**.

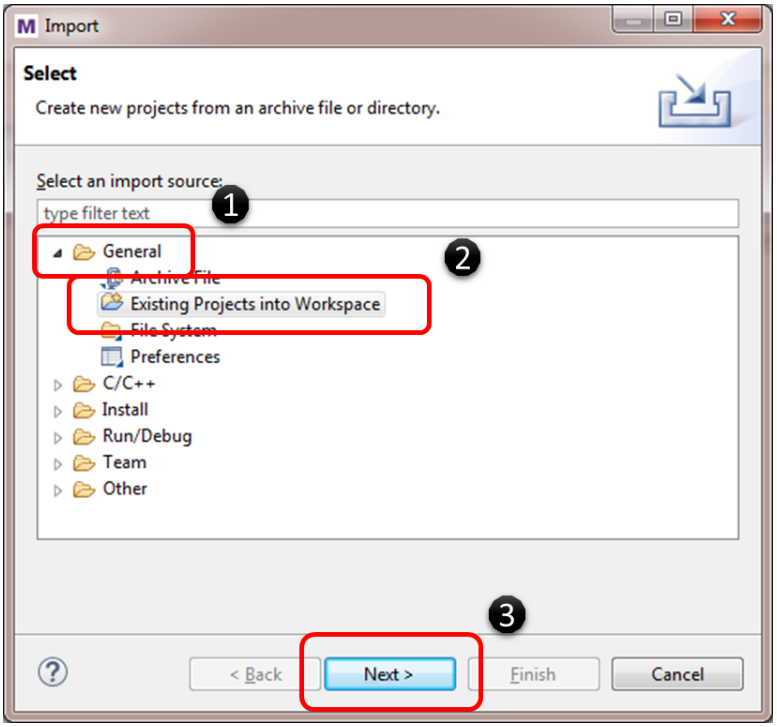


Figure 13 Adding existing projects into the workspace.

11. Click Browse to select the **sw** folder containing the project files.

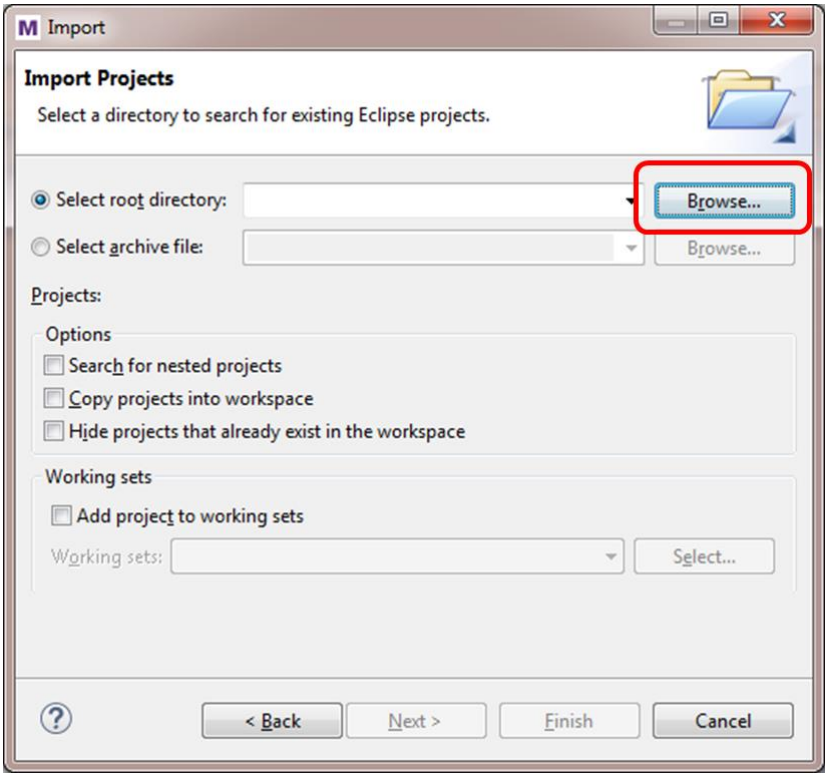


Figure 14 Import project files.

12. Navigate to the **em9304sdk** folder, select the **sw** folder, and click **OK**. The **sw** folder contains the project files for the example applications.

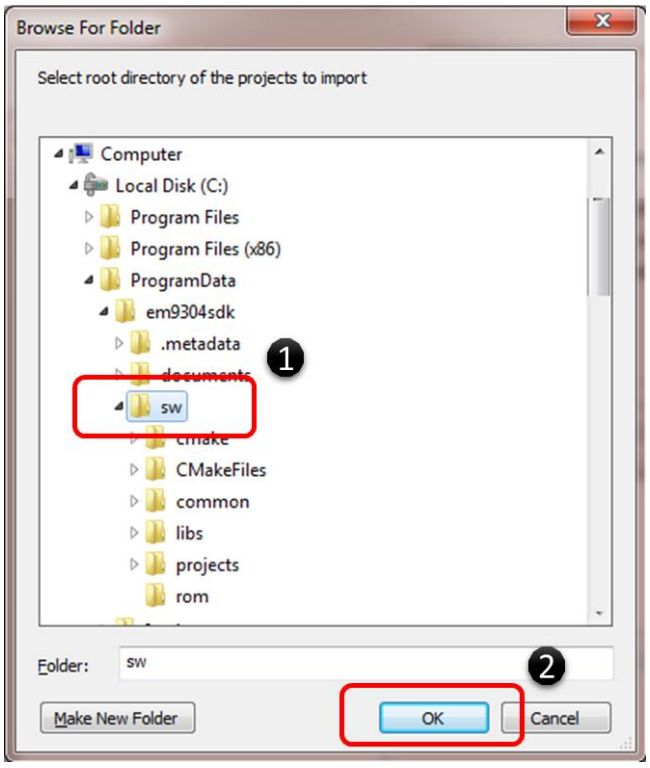


Figure 15 Selecting the sw folder containing the project files.

13. Click **Finish** in the Import Projects window.

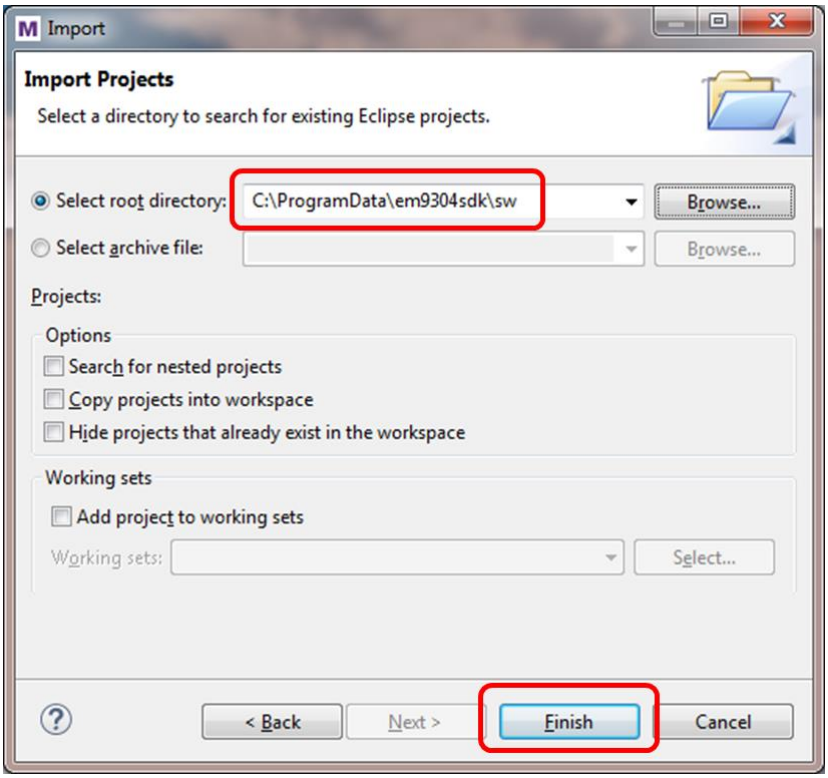


Figure 16 Finishing the import process.

14. The workspace and projects setup is now complete, and the main Metaware IDE Lite window will be displayed. In the Project Explorer pane, expand the **sw** folder then the projects folder to reveal example projects. The FindMe example application is shown in Figure 17.

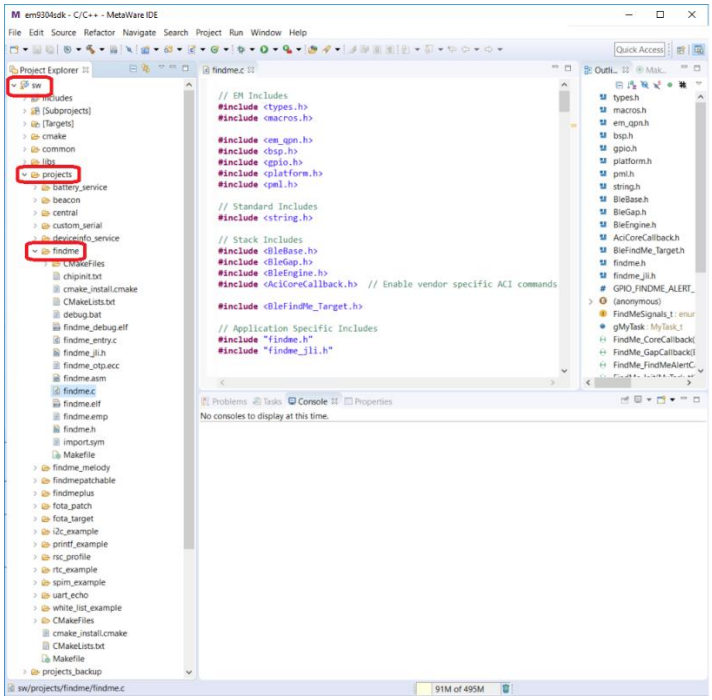


Figure 17 Main window with workspace files shown.

7. BUILDING THE EXAMPLE APPLICATIONS

The example applications can be built individual or together. To build the all of the example applications, click **Project** then **Build All**. A single project can be built by selecting the project in the Project Explorer pane and the clicking **Project** and then **Build Project**. Select **Build All** for this tutorial or use the keyboard shortcut, Ctrl-B.

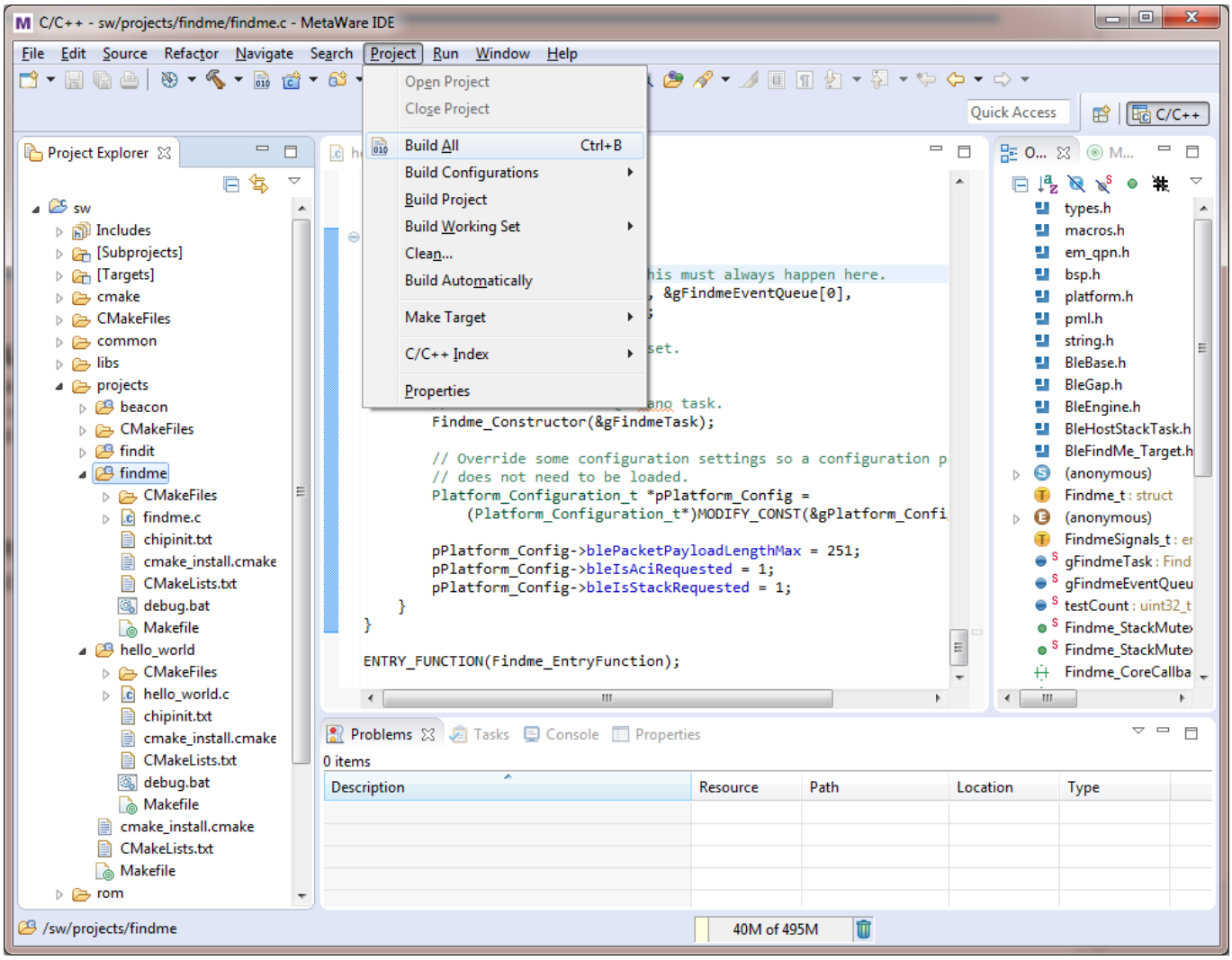


Figure 18 Build All to build all of the example applications.

After a few seconds, the build will complete, and the bottom right pane will show zero problems or errors. Clicking the Console tab will show something similar to output in Figure 19.

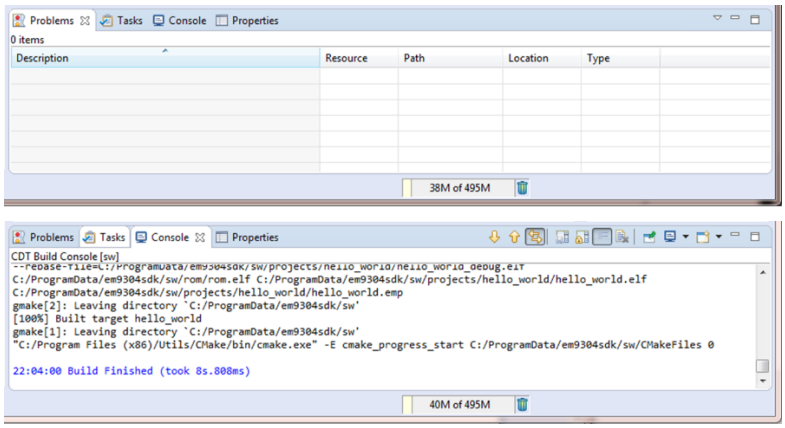


Figure 19 Completed build with 0 problems.

The FindMe project is compiled and ready to send to the EM9304 SOC SDK.

8. PREPARING THE DVK FOR THE FINDME EXAMPLE APPLICATION (OPTIONAL)

The FindMe example application toggles LED1 on the EM9304 to demonstrate issuing a FindMe alert from a smartphone to the EM9304. In order to drive the LED, a minor modification must be made to the EM9304 DVK board. The sample application can be demonstrated without modifying the board to drive LED1, so the instructions in this section can be skipped. In other words, the sample application will still operate, but LED1 will not change state when an alert is issued.

The EM9304 DVK board is designed to allow GPIO1 to GPIO7 to be easily connected to LED1, LED2, button 1 (B1), or button 2 (B2) by applying a small amount of solder to the two adjacent pads. For more information regarding the configuration options, see the EM9304 schematic provided in the SDK at

documents\EM Microelectronic\dvk

To modify the DVK to connect GPIO6 to LED1, perform the following steps.

1. Separate the EM9304 DVK board from the Nucleo board, and turn the board over so the underside of the board is facing up.

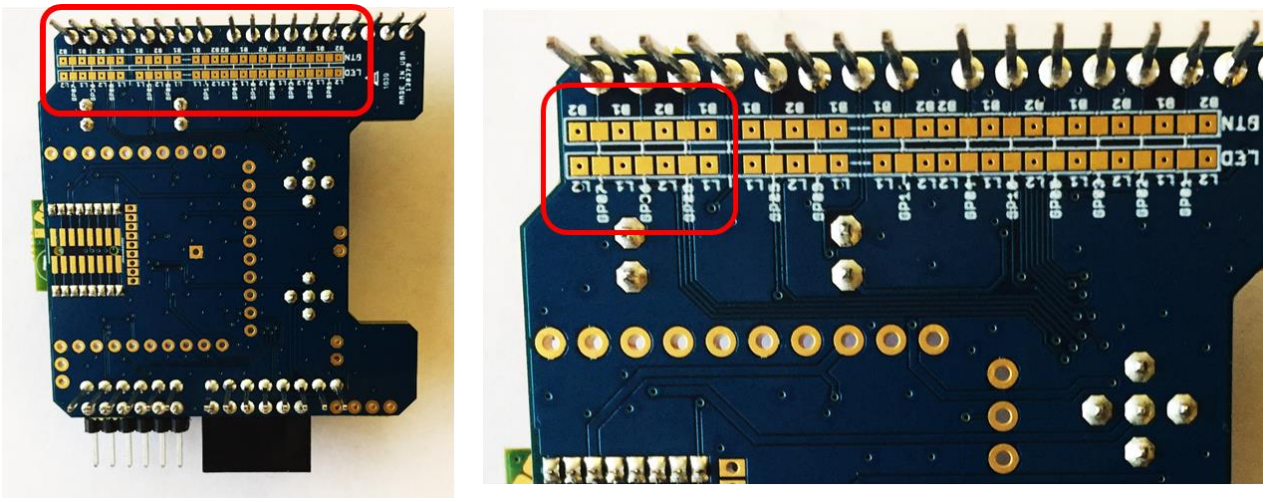


Figure 20 Underside of EM9304 DVK board (DVK Revision 4.0)

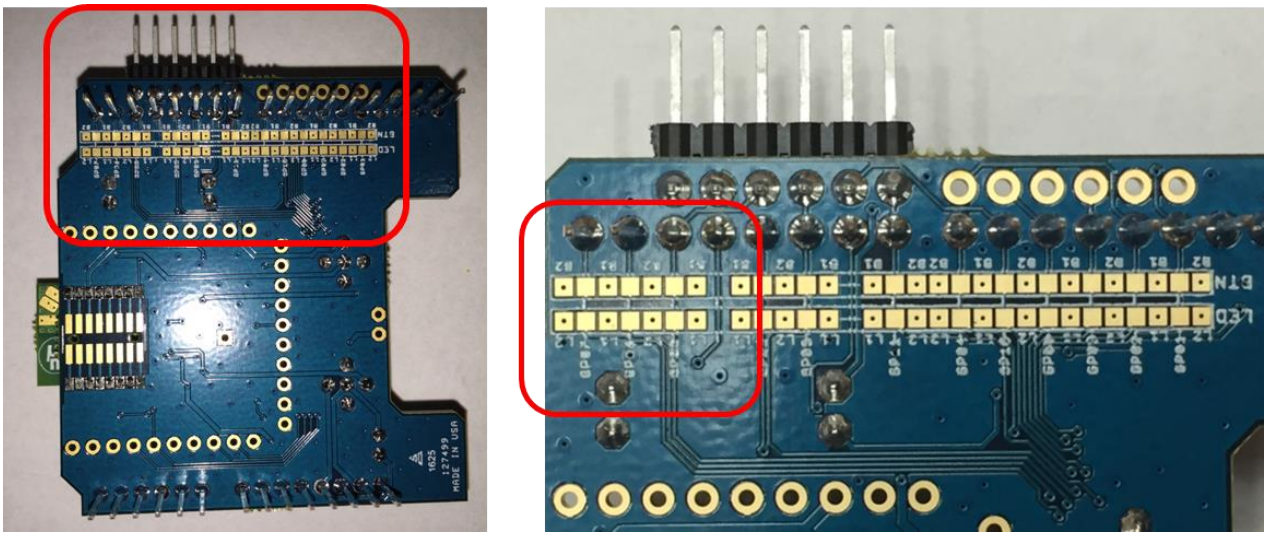


Figure 21 Underside of EM9304 DVK board (DVK Revision 3.0).

2. Locate the pad location marked GPIO6 near the JTAG connector as shown in Figure 22. Solder the pad labelled GPIO6 to the adjacent pad labelled L1. The pads are located close together to allow for a solder bridge to be easily formed. Ensure that only these two pads are connected.

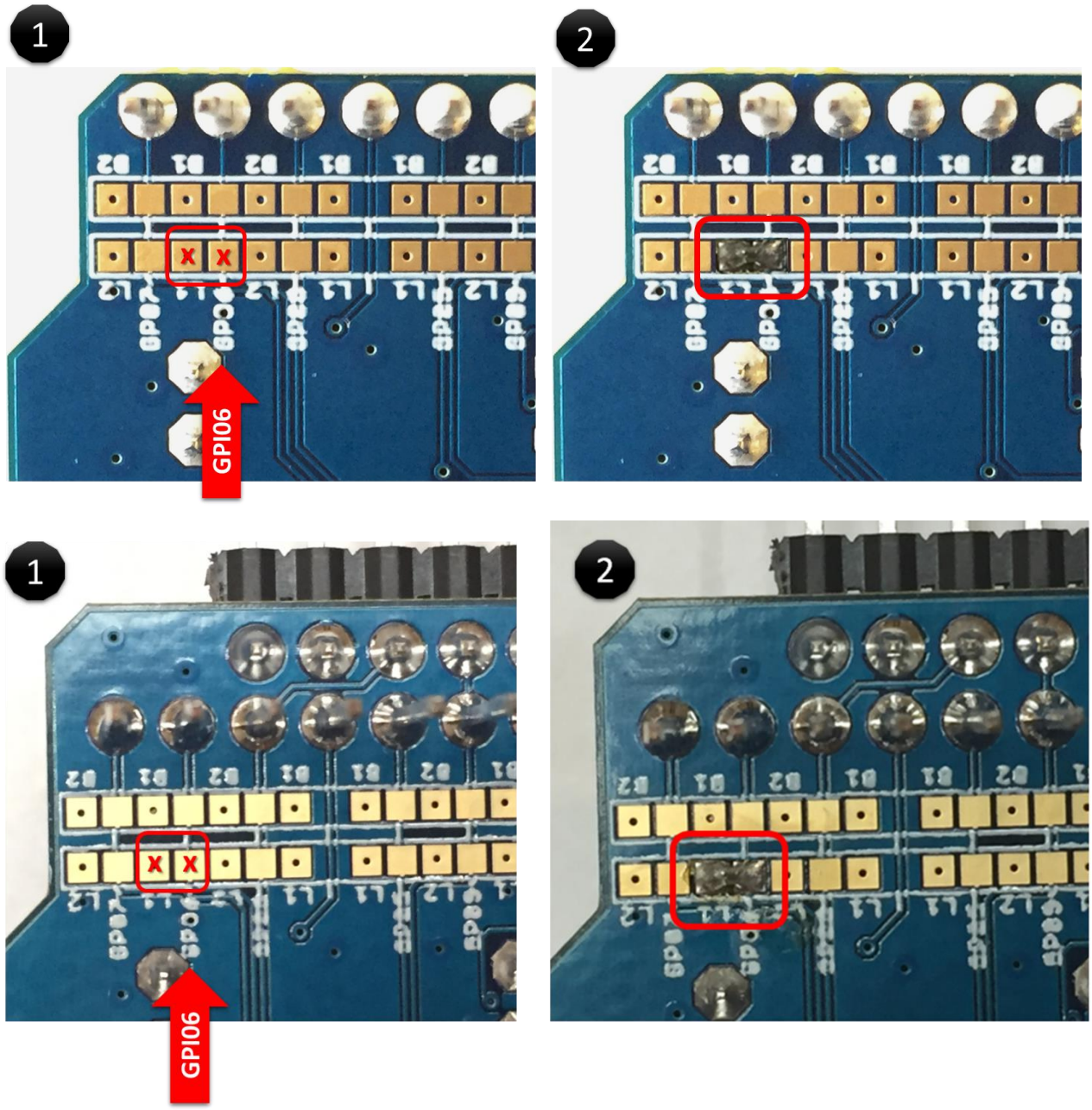


Figure 22 Connecting GPIO6 pad to LED1 (L1) pad (DVK Revision 3.0).

3. This completes the steps necessary to connect GPIO6 to LED1. Reattach the EM9304 DVK board to the Nucleo board.

The FindMe example application requires GPIO6 to be connected to LED1.

9. CONNECTING THE EM9304 DVK TO THE PC

The EM9304 DVK version 4.0 consists of three boards; white ST Nucleo board, blue EM Microelectronic Arduino board, and the green EM Microelectronic EM9304 module.

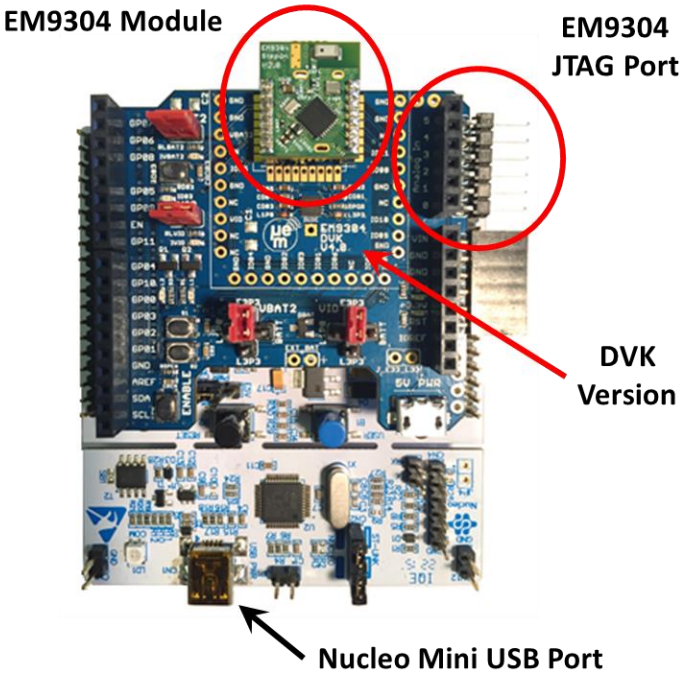


Figure 23 EM9304 DVK with JTAG and Nucleo Mini USB port identified.

Connect the Digilent HS2 JTAG cable and the mini USB cable as shown in Figure 24.

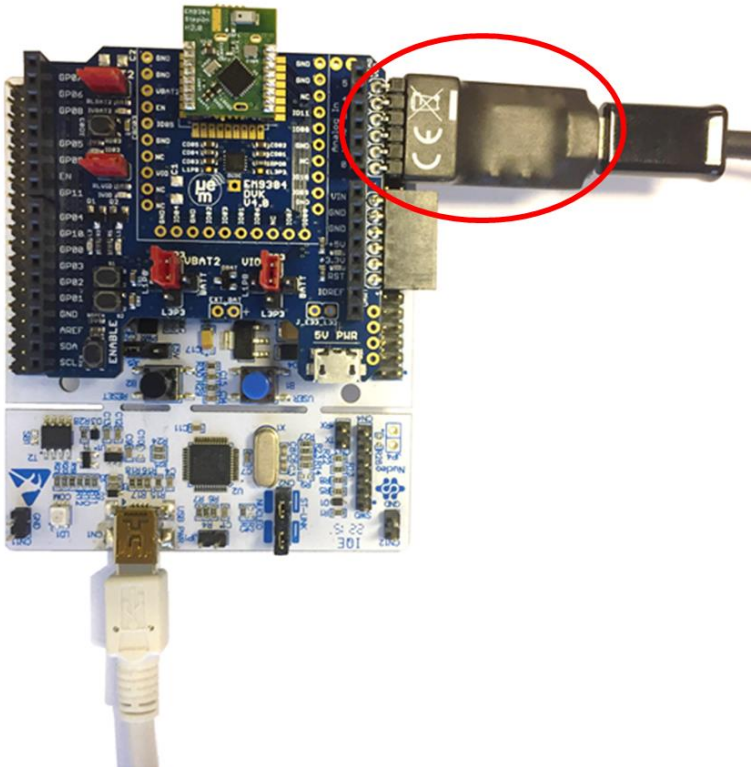


Figure 24 JTAG cable and mini USB cable connected to the EM9304 DVK.

Connect the USB end of the JTAG cable to the PC then connect the unattached end of the mini USB cable to the PC.

10. UPDATING THE EM9304 FIRMWARE

The SDK contains a firmware update that must be programmed into the EM9304 prior to programming to beginning development with the DVK. The firmware update provides critical updates and improvements required to successfully program and execute applications on the EM9304. Firmware updates for the EM9304 are packaged in EM patch (EMP) files. The latest firmware update is included in the SOC SDK meta-patch file located in the following subfolder of the SDK installation folder. The firmware is updated using the STLink Virtual COM Port using the ST Microelectronics Nucleo board.

sw\cmake\patches

Follow these steps to update or patch the firmware on the EM9304 by programming the EM9304 meta SOC patch to the OTP.

1. Launch the EM9304 Configuration Editor by selecting **EM Microelectronic EM9304 SOC SDK** and then **EM9304 Config Editor** from the Start menu.
2. From the **Select Firmware** Dialog Window, click **Browse**.

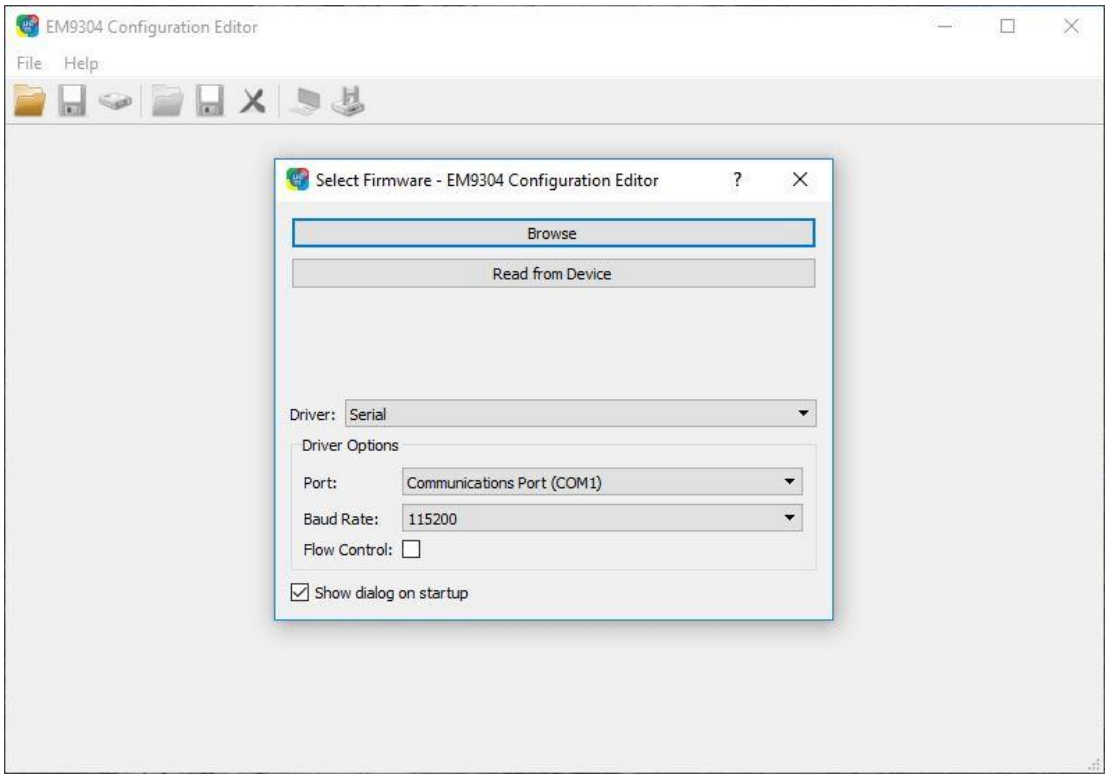


Figure 25 EM9304 configuration editor.

3. Select the **rom_revB.elf** file and click **Open**. Selecting the ROM ELF files is necessary so that the Configuration Editor retrieves the correct default configuration settings for the EM9304 ROM version.

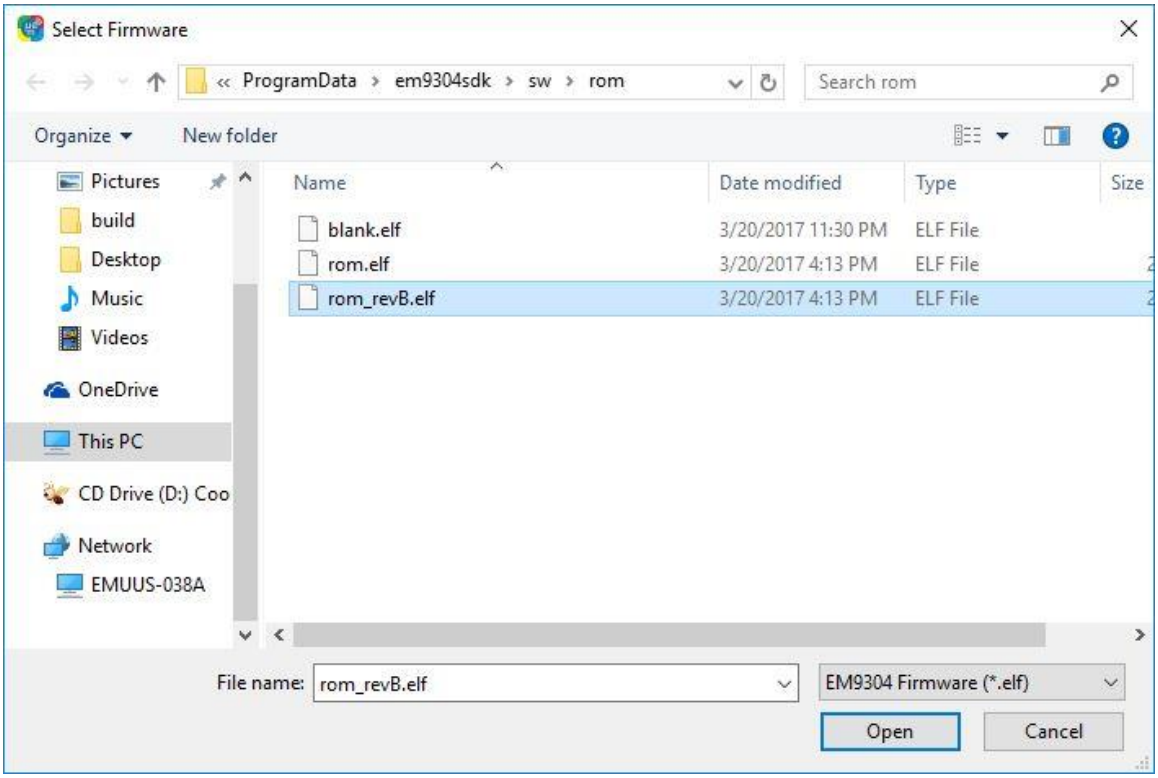


Figure 26 Selecting the rom_revB.elf file.

4. From the main menu, select **File**, and then **Upload Patch...**

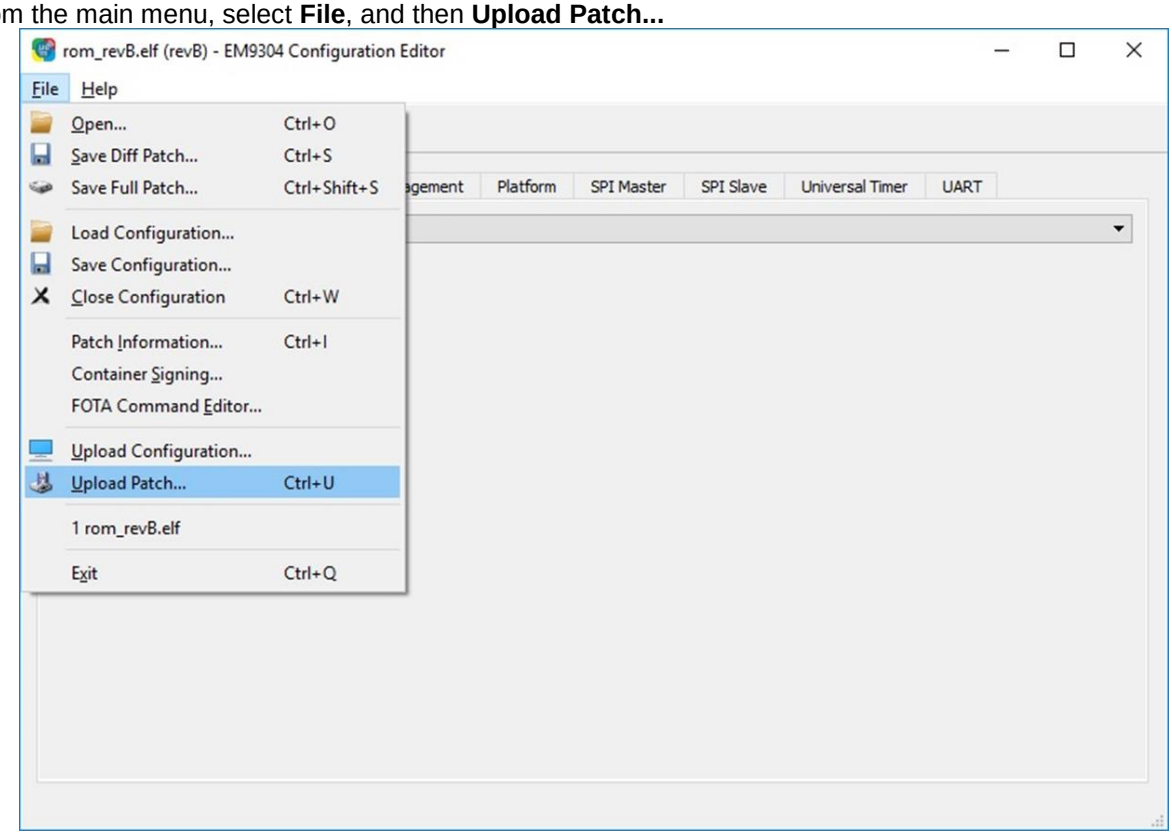


Figure 27 Program or upload the firmware update.

5. Select the SOC SDK meta-patch by browsing to the SDK installation location, and then the sw\cmake\patches folder. Select the SOC SDK meta-patch, **0000000_META_soc_patches_v8.emp**, and click Open.

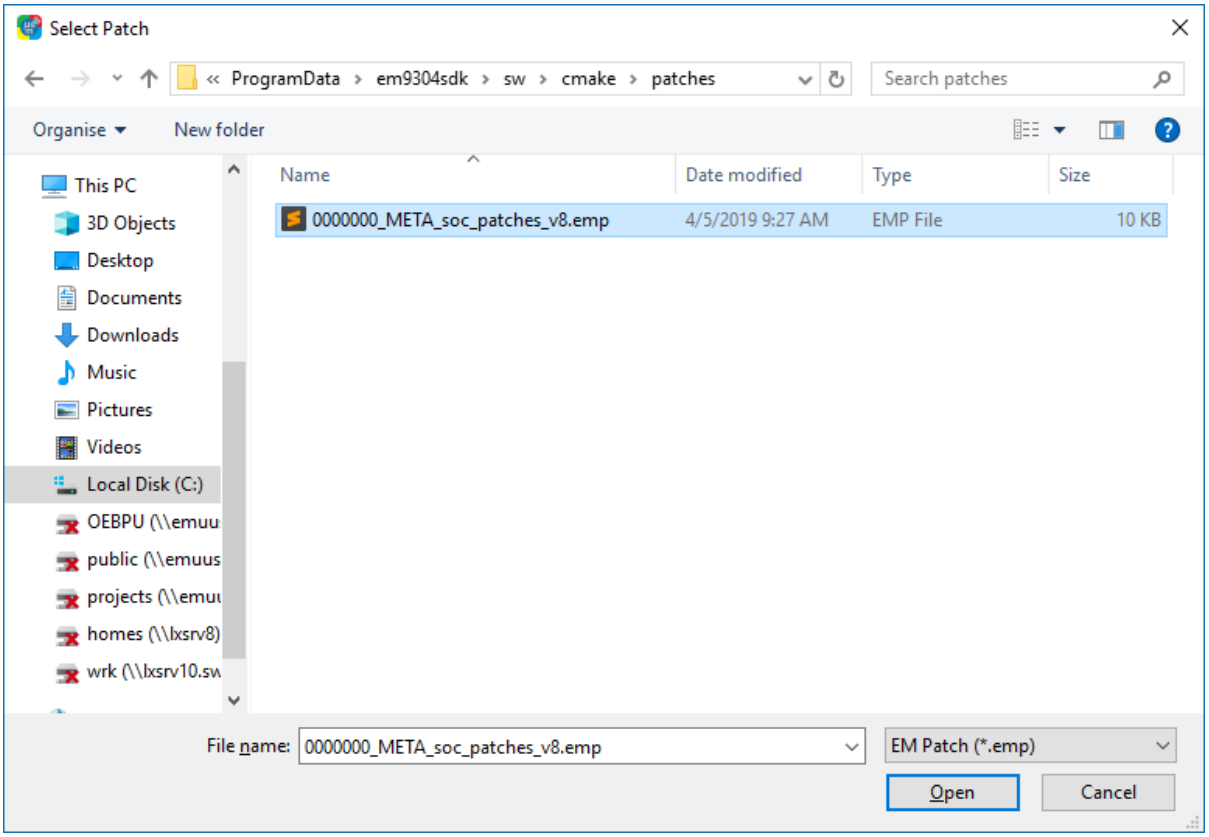


Figure 28 Select the EM9304 meta SOC patch.

6. Under **Destination**, select **OTP**, and verify the other settings are set as shown in Figure 29 Upload patch dialog. Ensure the **Check SDK Patch** checkbox is unchecked. The Driver entry allows the user to program the DVK through the serial, JTAG, or USB interface, or save the code patch to a file. The port entry will display "**STMicroelectronics STLink Virtual COM Port (COM *X*)**" where COM *X* is the port attached to the Nucleo board The **Reset After Upload** checkbox forces a reset of the EM9304 after the patch has been programmed into the OTP. The programmed patch does not take effect until the EM9304 is reset.
7. **Click Upload**.

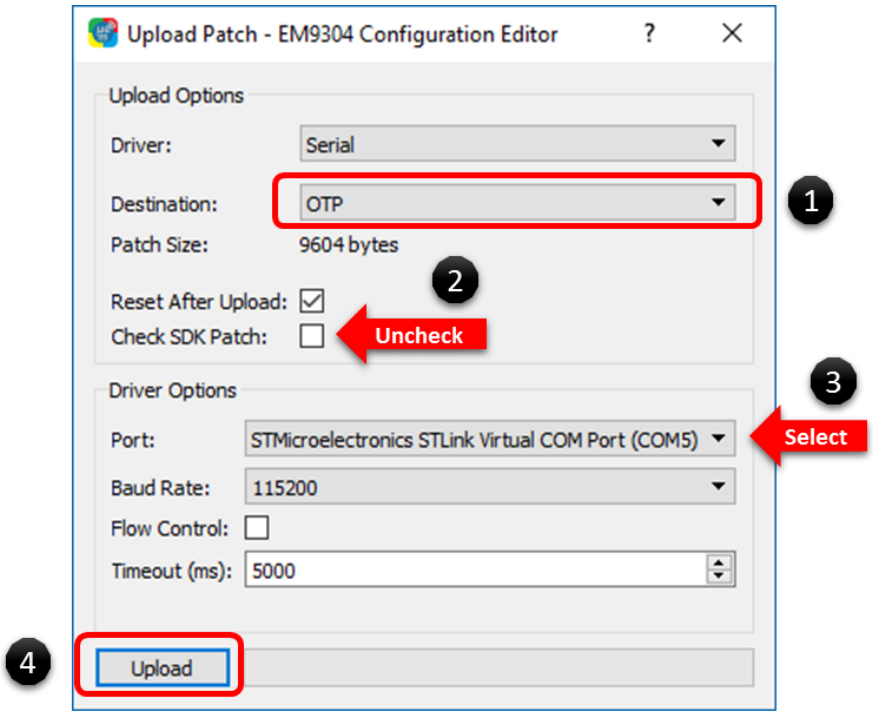


Figure 29 Upload patch dialog.

8. Programming the EM9304 meta SOC patch should take less than a few seconds. The message box will show the status of the programming process.

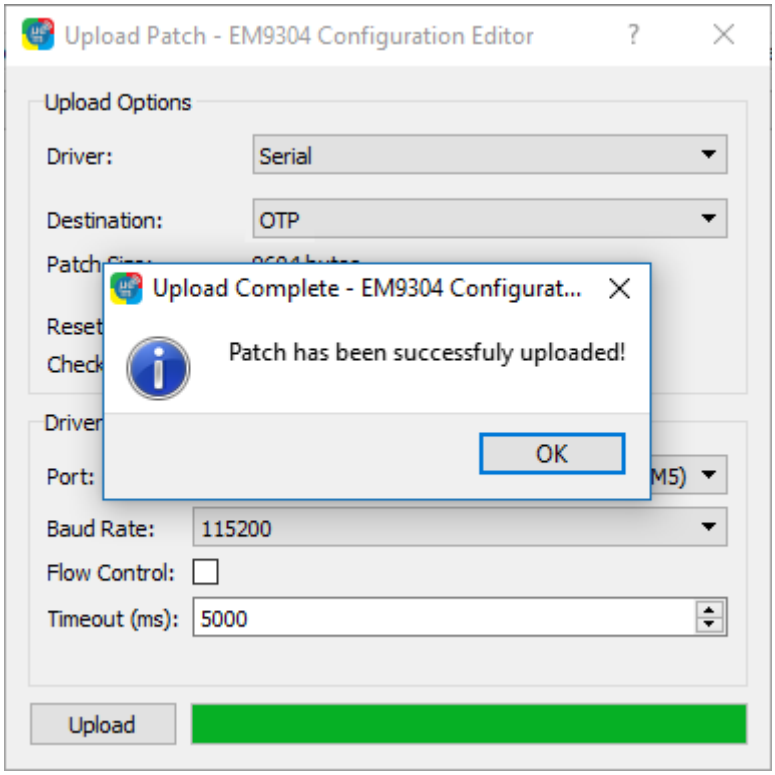


Figure 30 Firmware update successfully applied.

The EM9304 on the DVK is now ready for application development.

11. CONFIGURING THE EM9304 TO ENABLE JTAG DEBUGGING

The EM9304 supports debugging through the JTAG port with the Diligent HS2 cable. To minimize power consumption of the EM9304, system sleep mode disables most of the EM9304 subsystems including the JTAG port. When the EM9304 enters system sleep mode, the debugger is disconnected and cannot be reconnected without resetting the debugger. In other words, *the debugger cannot be used when system sleep mode is enabled*, so a configuration option is provided to disable system sleep mode.

When system sleep mode is disabled, the embedded software substitutes CPU sleep mode for system sleep mode. CPU sleep mode simulates the operation of system sleep mode without adversely affecting debug capability of the EM9304. The power consumption when CPU sleep mode is active is significantly higher than system sleep mode, so accurate power consumption measurements cannot be made with system sleep mode disabled.

To disable system sleep mode, the EM9304 Configuration Editor application is used to create a configuration patch that disables system sleep mode. The configuration patch is then programmed into the OTP memory using the configuration editor. This change is permanent until a new configuration patch that enables system sleep mode is programmed into OTP memory. In this context a patch is defined as a program and is uploaded to the EM9304 via the Diligent HS2 cable.

After connecting the DVK to the PC, follow these steps to disable system sleep mode by programming a configuration patch to the OTP.

1. Launch the EM9304 Configuration Editor by selecting **EM Microelectronic EM9304 SOC SDK→EM9304 Config Editor** from the Start menu.
2. From the **Select Firmware** Dialog Window, select **rom_revB.elf**. Since the ELF was already selected in the previous section, the Configuration Editor provides the option of selecting the last two selected ELF files.

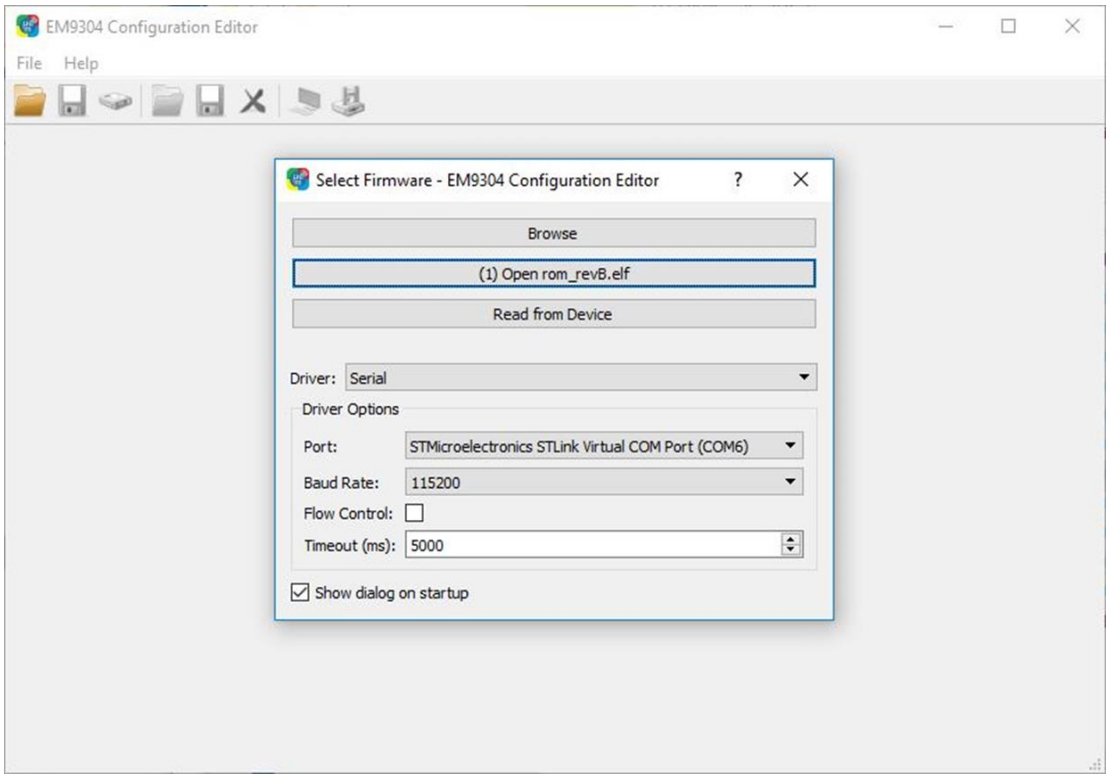


Figure 31 EM9304 Configuration Editor.

3. All of the available configuration options will be shown by the Configuration Editor organized under several tabs. Click the **Power Management** tab and check the **Disable Sleep Mode** checkbox.

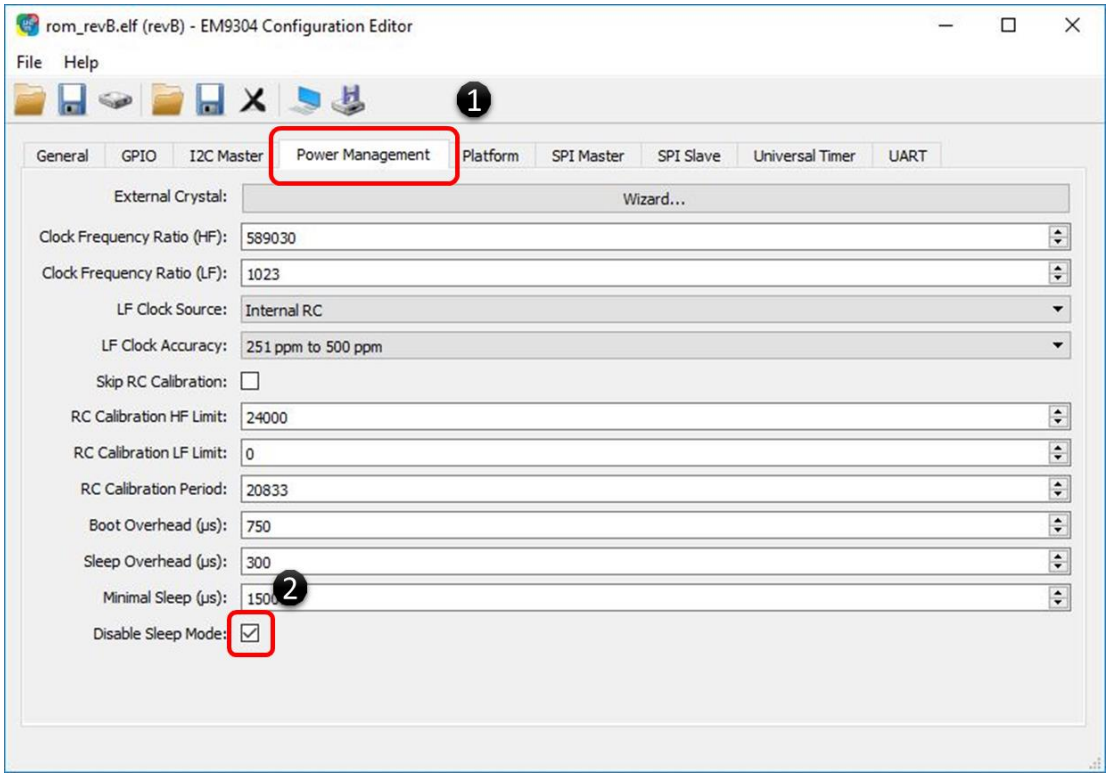


Figure 32 Disable sleep mode from the Power Management tab.

4. Select **File** and **Upload Configuration…** to bring up the **Upload Patch** window.

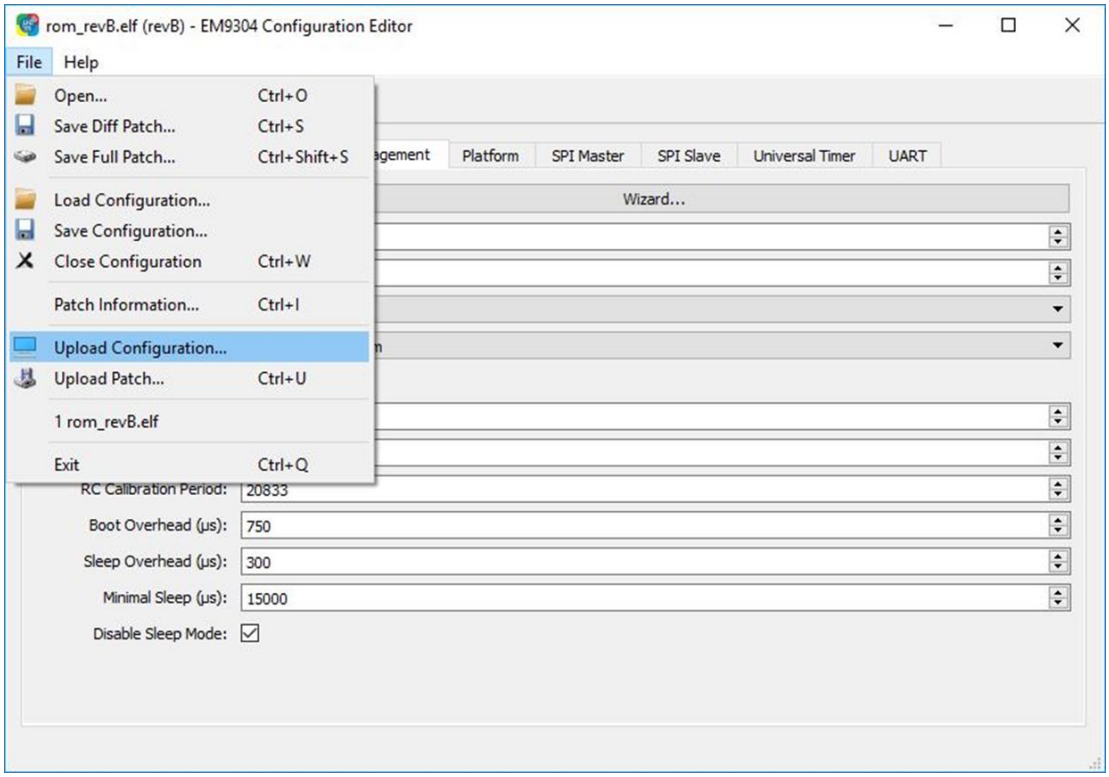


Figure 33 Prepare to upload the configuration change.

5. Under **Destination**, select **OTP**, and verify the other settings are set as shown in Figure 34 Upload patch dialog. The Driver entry allows the user to program the DVK through the serial interface or save the configuration patch to a file. The port entry will display "**STMicroelectronics STLink Virtual COM Port (COM *X*)**" where COM *X* is the port attached to the Nucleo board. The **Patch Type** specifies whether all configuration options will be written or whether only those configuration options that have changed will be

written. Since only a single option is changed, a Diff patch saves significant space and some processing time during the boot process. The **Patch Size** is 32 bytes. The **Reset After Upload** checkbox forces a reset of the EM9304 after the patch has been programmed into the OTP. The programmed patch does not take effect until the EM9304 is reset.

6. Click **Upload**.

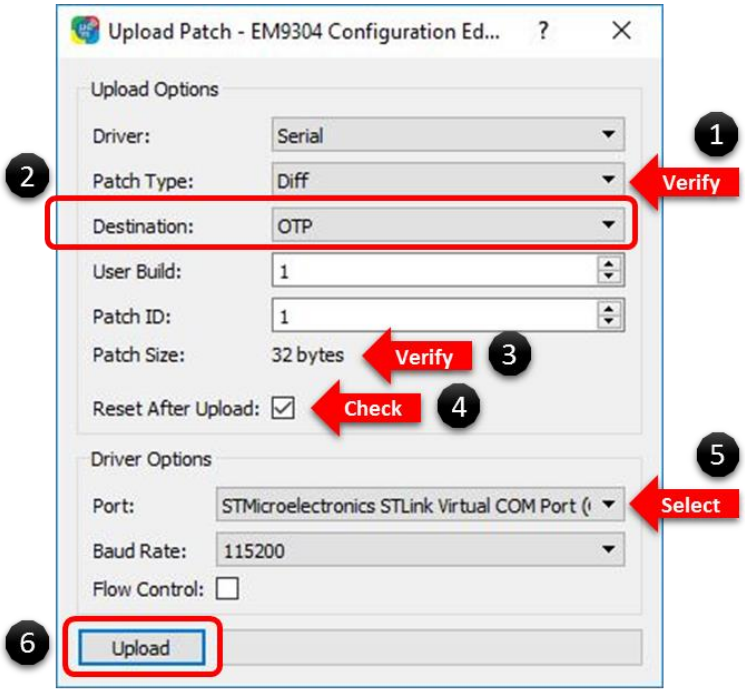


Figure 34 Upload patch dialog.

7. The message box will indicate the patch uploaded successfully as shown in Figure 35. Click **OK** in the **Upload Complete** message box.

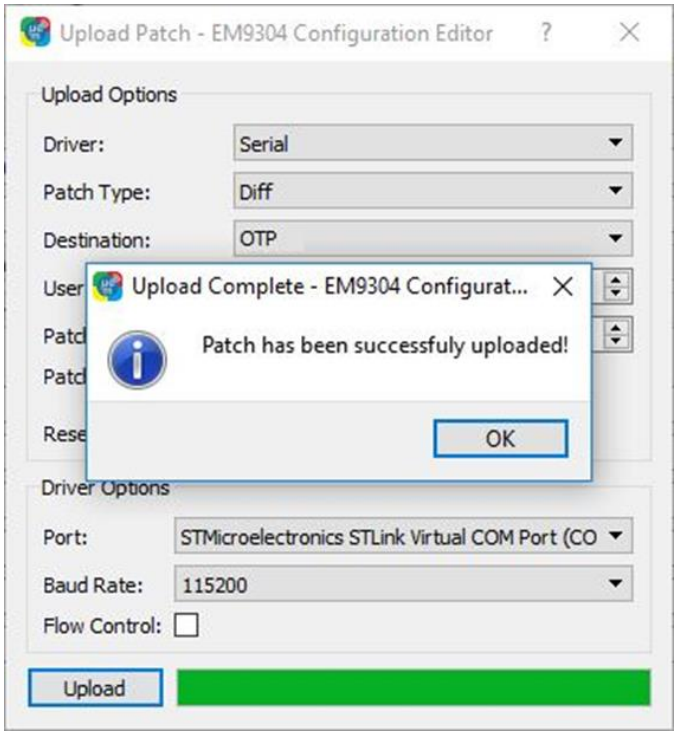


Figure 35 Patch upload successful.

8. Close the upload patch dialog, and **Exit** the EM9304 Configuration Editor. There is no need to save the configuration patch upon exit.

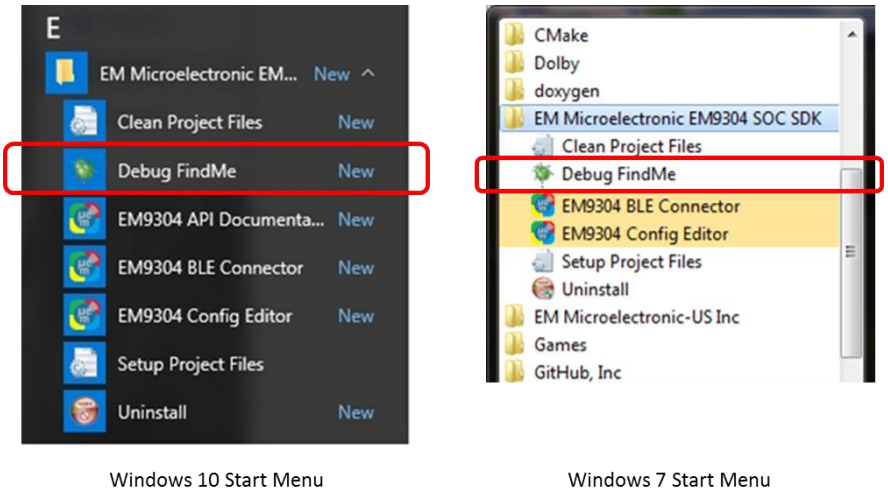
12. STARTING THE METWARE DEBUGGER

Although the Metaware IDE contains integrated debug functionality, it is recommended that you use the standalone Metaware Debugger. A short batch file (debug.bat) is used to launch the debugger with the required debugger command line options. The batch file can be run from a DOS command prompt from one of the example project folders. Each example program comes with a debugger.bat file. Alternatively, a Windows Start menu shortcut is provided to launch the debugger for a given example application.

When the batch file is executed, a small number of patches are programmed into the IRAM of EM9304 through the JTAG debugger. These patches are required to enable the internal programming process through JTAG. Following the patches, the user application is programmed in to the IRAM of the EM9304 through the JTAG debugger. Once the programming is complete, the debugger executes the EM9304 internal firmware until the entry function of the user application is reached, and execution is halted.

To launch the debugger and run the example FindMe example application, complete the following steps.

1. From the Start menu, click EM Microelectronic EM9304 SOC SDK ⟶ FindMe.



Windows 10 Start Menu Windows 7 Start Menu

Figure 36 Launch the standalone debugger for the FindMe example application.

2. Once the Metware Debugger is launched and the debugger splash screen disappears, the main debugger window will show the **Debug a process or processes** dialog window. Click **OK** to proceed as the appropriate settings are already set.

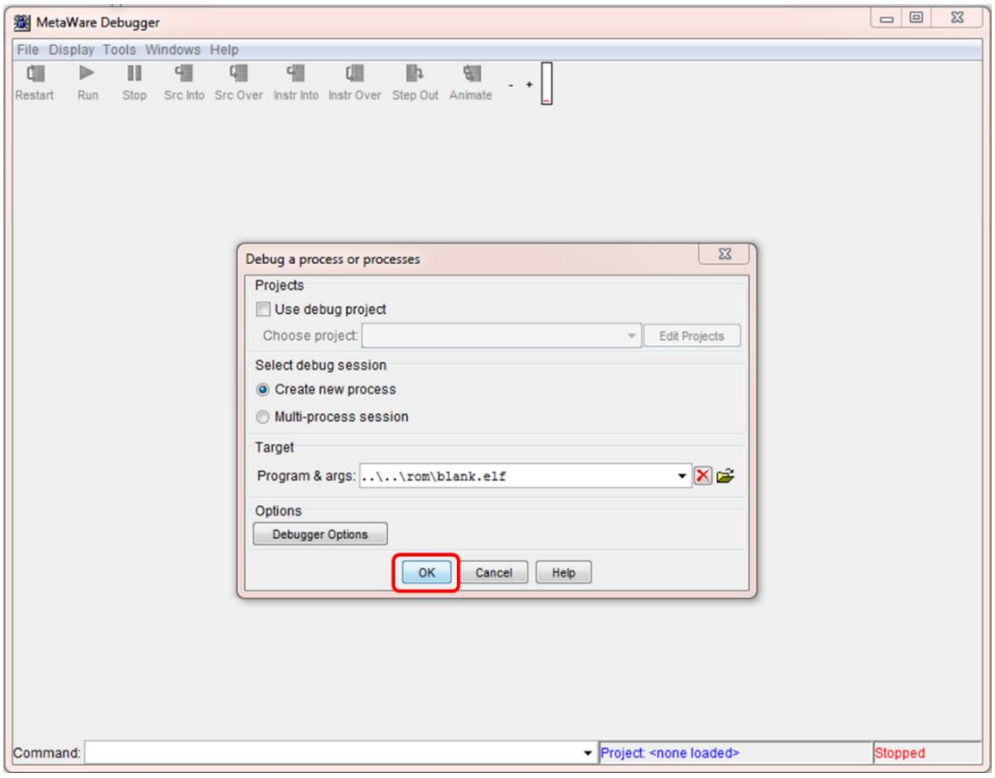


Figure 37 Initial Metaware Debugger window after launch.

3. It may take approximately 30 seconds for the debugger to load. The bottom right corner of the main debugger window will blink the word, "Busy…" while the loading process is underway. At the same time, the DOS command window that was launched when the Metaware Debugger was launched will show activity as well. Once loaded, the debugger will halt at the entry function of the FindMe example application as shown in Figure 36.

   The entry function of an application is an application specific function that is called early in the boot process when the content of the OTP is scanned for program code and other content. The entry function should contain minimal functionality such as creating the QP-nano task and setting the Bluetooth low energy options. Most application initialization functionality should be completed in the tasks initialization function.

   The default debugger view contains a **Local Variables** pane, a **Source** pane, a **Call Stack** pane and a **Debug Console** pane. The bottom right corner of the window shows the current activity status. Additional information panes are added by going to **Display** under the main menu. Figure 38 highlights some of the main components of the main debugger window.

   Clicking the **Run** icon in the main toolbar will start the example application. Clicking **Stop** will halt the application, and **Restart** will restart the application, and halt execution at the entry function. Restarting the application may take up to 30 seconds depending on a number of application specific factors. Break points are added by right clicking on a source code line, and selecting **Hardware Breakpoint**. A maximum of four breakpoints can be active. Software breakpoints are not supported.

   The **Debug Console** pane can also list **Watchpoints** and **Breakpoints** when the corresponding tab is selected. Watchpoints provide a means of halting the debugger when a specified variable changes state or is assigned a value or assigned a specific value. Additional information can be found by clicking **Help** from the main menu.

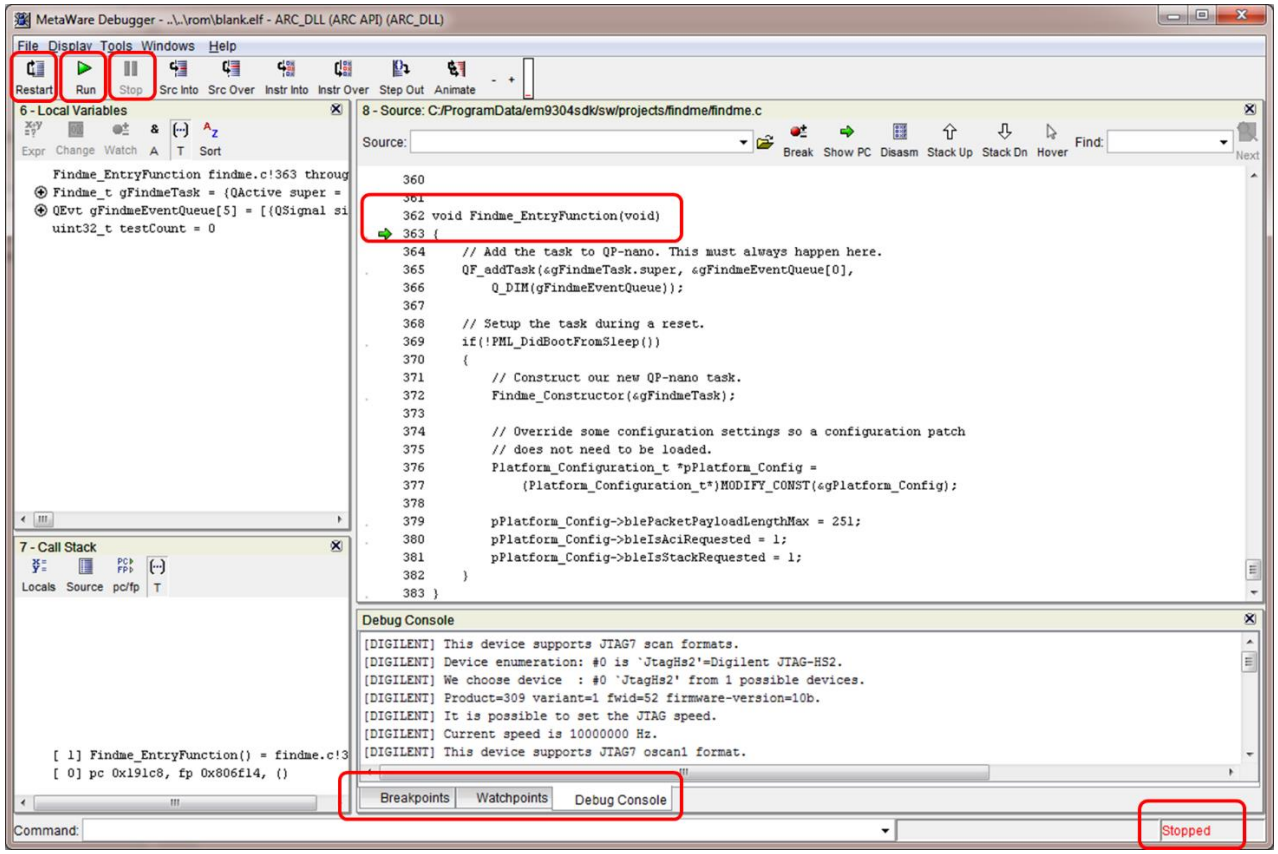


Figure 38 Main debugger window highlights.

4. Start the example application by clicking the **Run** button on the toolbar. When the debugger is running, the bottom right corner of the main debugger window will display “Run #” where the “#” will move across the bottom of the status area.

13. TESTING THE FINDME EXAMPLE APPLICATION WITH A SMARTPHONE

The FindMe example application presents the BLE FindMe profile to Bluetooth enabled devices such as a smartphone. After connecting with a smartphone, alerts can be sent from the smartphone to the EM9304, which turns the LED on when a high alert is issued, and turns the LED off when the alert is cleared. A simple way to interact with the FindMe application running on the EM9304 is with a smartphone and the Alpwise iBLE app for iOS or the Alpwise BLE app for Android. Complete the following steps to download and run the Alpwise app.

1. Download and install the Alpwise iBLE app for iOS or the Alpwise BLE app for Android. You can find the Alpwise mobil app by searching for “Alpwise” in the Apple App Store or in Google Play.
2. Install and open the Alpwise app.
3. Tap the Welcome screen to view a list of the available BLE devices in the immediate area. The EM9304 will be listed as **EM9304 Findme**.
4. If **EM9304 Findme** is not listed on the Android smartphone with Android OS 6.0.1 or higher, the application permissions for the Alpwise App must be adjusted. To add storage and location permissions to the Alpwise app on an Android smartphone:
    a. Open the **Settings** screen.
    b. Tap **Apps**.
    c. From the list of installed apps, tap **Alpwise iBLE**.
    d. Tap **Permissions**.
    e. Allow **Location** and **Storage** permissions.
    f. Exit the **Settings**.

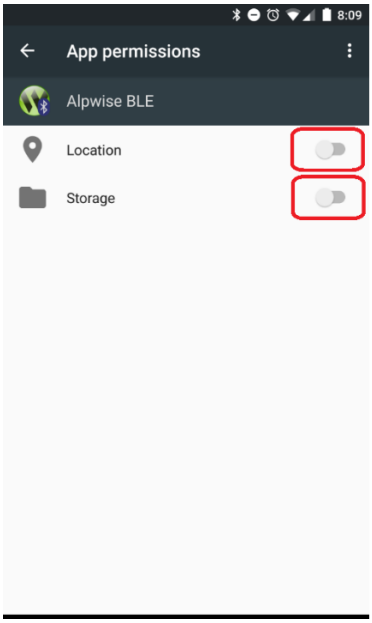


Figure 39 Android Smartphone Alpwise App Permissions.

5. Tap the **EM9304 Findme** entry in the list to display the example applications supported profile.
6. Tap the **Find Me** profile to display the **Select the alert level** screen.

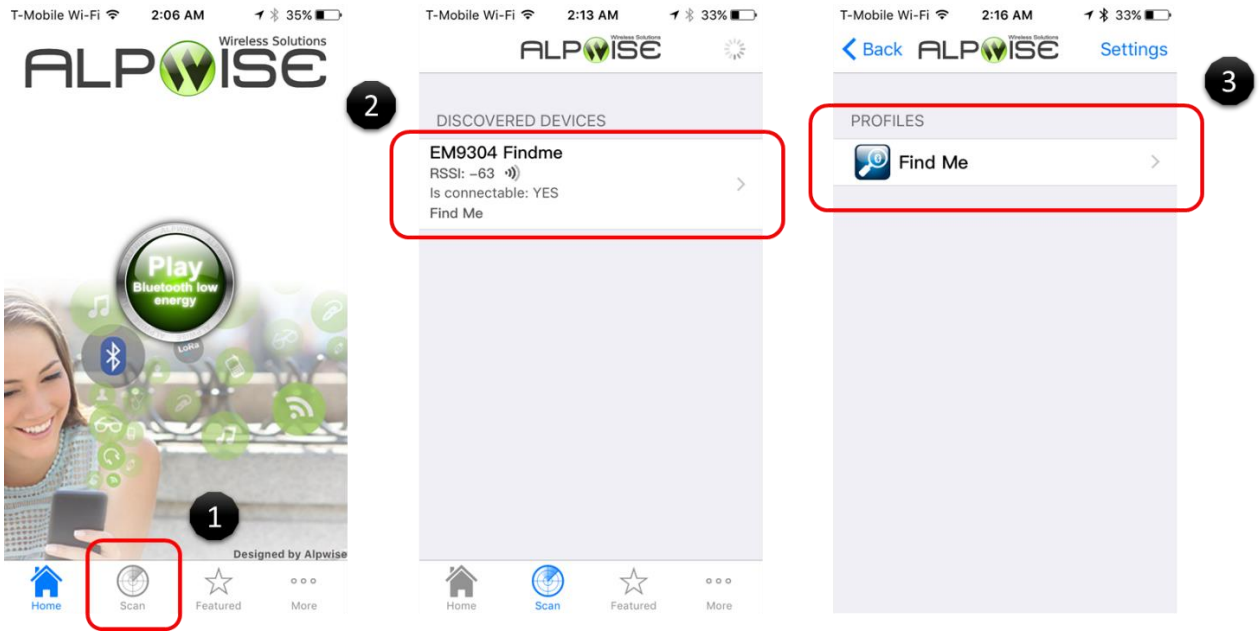


Figure 40 Alpwise BLE mobile app connecting to the EM9304 (iPhone).

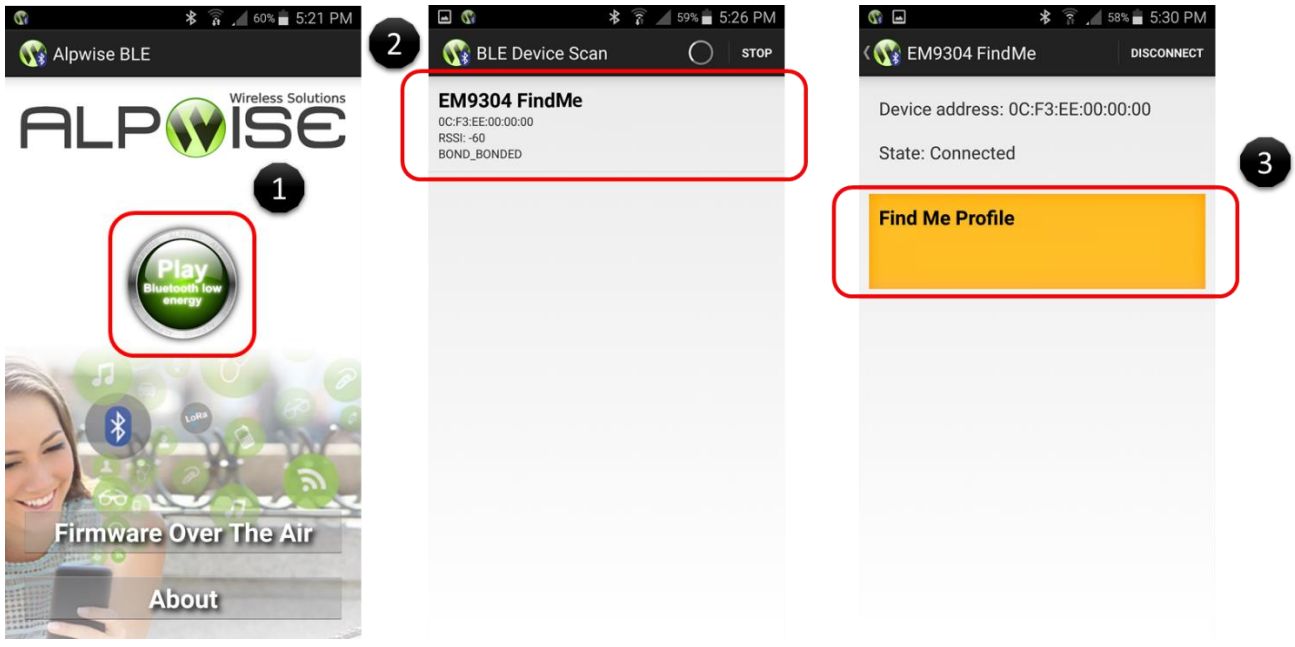


Figure 41 Alpwise BLE mobile app connecting to the EM9304 (Android).

7. The Select the alert level screen allows you to send and clear alerts. Tap the High Alert button to send the alert to the EM9304. LED1 on the DVK will be lit once the alert is received by the EM9304.

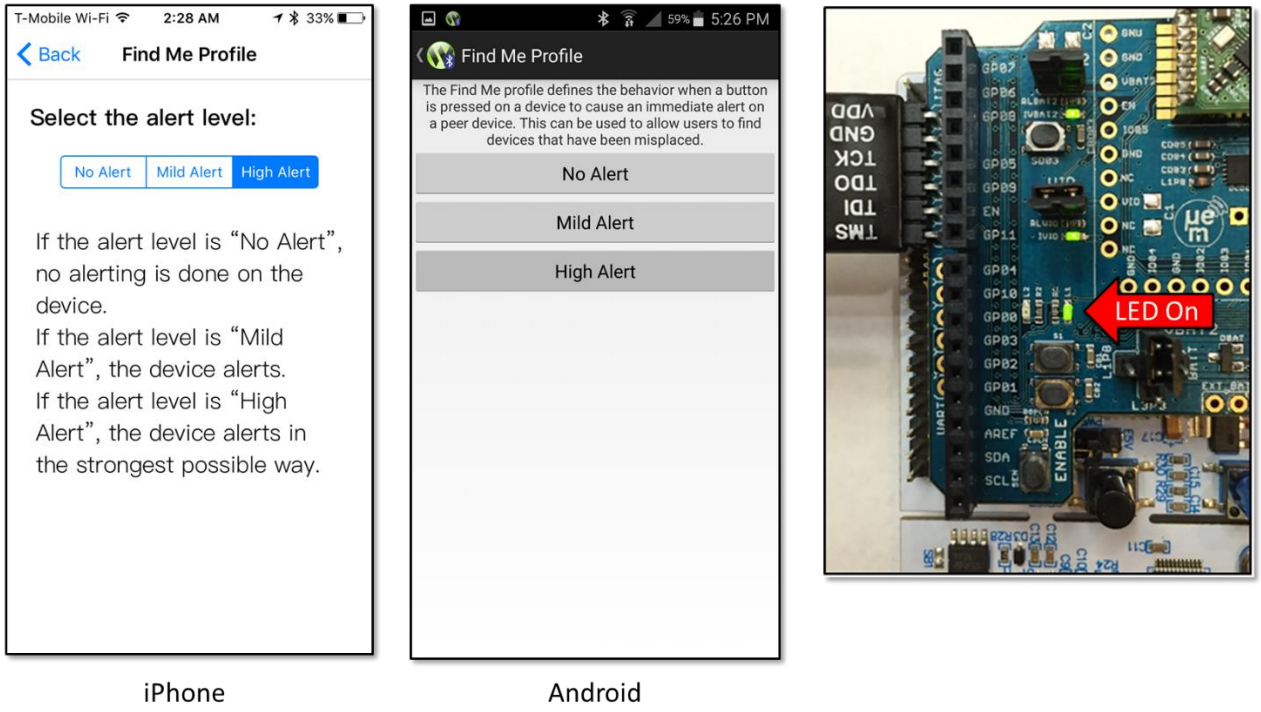


Figure 42 Find Me high alert issued and received on an iPhone and Android phone.

8. Tap the No Alert button to clear the alert and LED1 on the DVK will turn off.

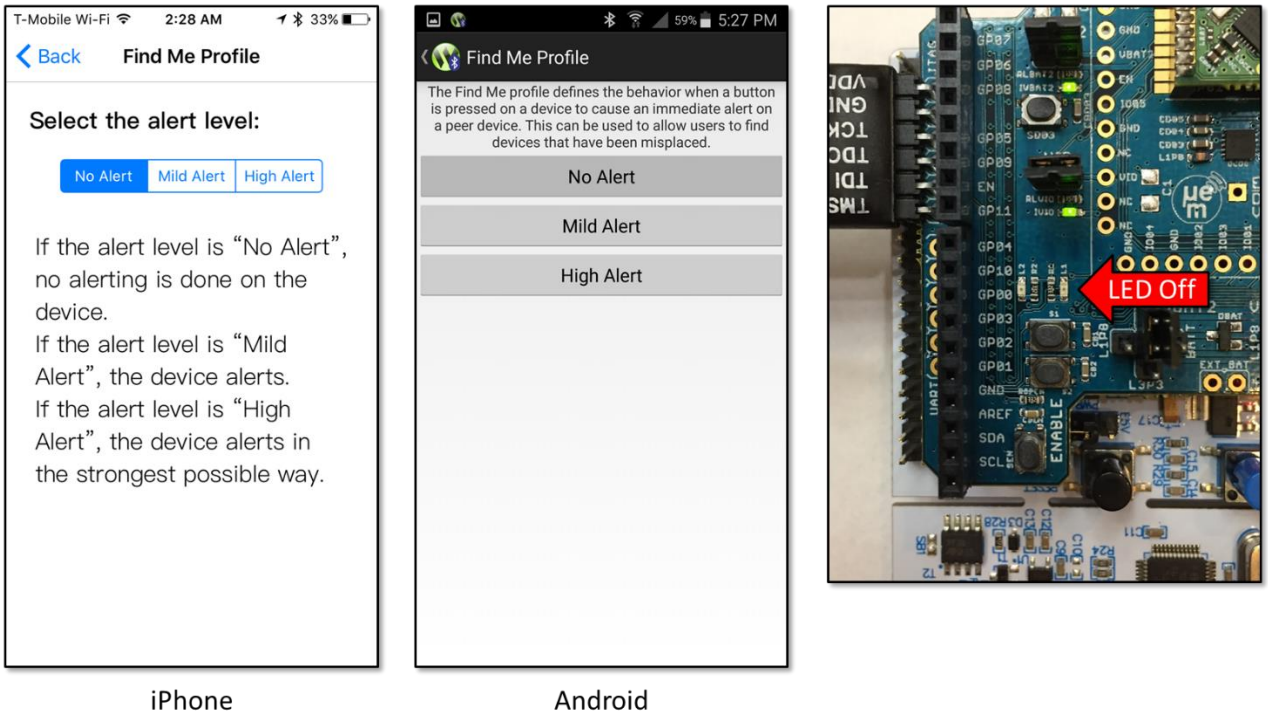


Figure 43 Find Me alert cleared on an iPhone and Android phone.

This completes the tutorial.

14. NEXT STEPS

This tutorial covered many topics, but was only able to provide a glimpse of the things possible with the EM9304 DVK and SOC SDK. To find out more, the following list provides some ideas of what to look at next.

- Visit www.emdeveloper.com and register for a free account
- Review the EM9304 data sheet and the EM9304 DVK schematic file located in the documents folder
- Read the EM9304 software API documentation in the documents folder of the SDK
- Examine the include files provided in the **sw\libs** folder of the SDK
- Review the example applications
- Read the Bluetooth© Low Energy SDK Core stack Implementer's Guide located in the **documents\Alpwise** folder
- Explore the information contained in the documents folder of the SDK
- Experiment with the sample applications. The FindMePlus example also uses the FindMe Bluetooth Stack profile, but incorporates more functionality including the Alpwise data exchange profile, the battery profile, reading the sleep timer, using a random Bluetooth address, and pairing and bonding.
- Gain an understanding of the requirements for making an application patchable when it is programmed to the OTP. The FindMePatchable example shows how the FindMe example was made patchable.

Copyright @ 2019

EM Microelectronic-Marin SA (“EM”) makes no warranties for the use of EM products, other than those expressly contained in EM's applicable General Terms of Sale, located at http://www.emmicroelectronic.com. EM assumes no responsibility for any errors which may have crept into this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein.

No licenses to patents or other intellectual property rights of EM are granted in connection with the sale of EM products, neither expressly nor implicitly.

In respect of the intended use of EM products by customer, customer is solely responsible for observing existing patents and other intellectual property rights of third parties and for obtaining, as the case may be, the necessary licenses.

Important note: The use of EM products as components in medical devices and/or medical applications, including but not limited to, safety and life supporting systems, where malfunction of such EM products might result in damage to and/or injury or death of persons is expressly prohibited, as EM products are neither destined nor qualified for use as components in such medical devices and/or medical applications. The prohibited use of EM products in such medical devices and/or medical applications is exclusively at the risk of the customer