Simplicity studio user guide

Start your IoT development with Simplicity Studio and discover the latest content here. This is where you'll find the latest way content, including training modules, videos, and sample projects so you can reach the ground running with your project. Questions or feedback? Visit the Simplicity Studio Community. Simplicity Studio Training Curriculum: Getting Started 3. AN0822: Simplicity Studio User Guide Comprehensive user guide to getting started with wireless application development with EFM32, EFM8, 8051 Silicon Labs MCUs, wireless MCUs, and Zigbee SoCs. The 40-page document includes useful resources from interface navigation to project creation, debugging, development flows, Hardware Configurator, Network Analyzer, Application Builder, and relevant information about Eclipse for an easier transition to Simplicity Studio. Beginners | Prerequisite: none | Hardware Required: Every Silicon Labs Beginner Kit | Software Required: Simplicity Studio Installer | PDF 1. UG343: Multi-Node Energy Profiler User Guide User guide To understand the basics of Silicon Labs's Multi-Node Energy Profiler, including starting energy analysis sessions, interface navigation, energy statistics, data control, freezing and recording triggers, search capabilities, and profiling with code correlation. Beginners | Prerequisite: none | Requirements: none | PDF 1. Project Sharing Training on how to share project/code between team members. Medium | Prerequisite: none | Requirements: none | PDF ads ▼ Scroll to page 2 of 14 ads Thanks for your participation! * Your assessment is critical to improving the work of artificial intelligence, which forms the content of this Simplicity Studio 1 project greatly reducing development time and complexity with EFM32, EFM8, and 8051 Silicon Labs MCUs, wireless MCUs, and ZigBee SoCs. Simplicity Studio can create wireless applications and provide hardware configuration, network analysis, real-time energy debugging, high-powered IDE, and links to useful resources, all in one place. Download and install Simplicity Studio from: KEY POINTS Simplicity Studio makes the development process easier, faster, and more efficient. Integrated IDE and tools help optimize design. Get started with quick and easy development with Demo and Sample Software. Quickly find help and design resources. silabs.com Smart. Connected. Energy-friendly. Rev. 0.3 2 Relevant Resources 1. Relevant Resource Simplicity IDE Guide In the Simplicity Studio IDE, select [Help]>[Help Content] to display this guide as well as any documentation installed. AN0821: Simplicity Studio C8051F85x Walkthrough This document provides a step-by-step guide to using the Simplicity Configurator and Simplicity IDE tools with the C8051F85x. AN0823 family: Configurator User Simplicity In addition to the documentation in the tool itself, this document provides a discussion about Simplicity simplicity Tool. Ember UG102: Ember Application Framework Developer Guide This document explains how to use the Application Builder tool in detail and more information about the Application Framework. Ember UG104: Testing and Debugging Bucket Applications This document provides guidance for how to debug wireless networks. Silicon Labs Videos See Simplicity Studio tools in action! Access this video from the Silicon Labs website (Lizard Labs More information about certain tools included in Simplicity Studio and their use can be found at support/training/pages/online-training.aspx. silabs.com Smart. Connected. Energy-friendly. Rev 3 Overview 2. Overview Simplicity Studio is the launch pad for everything needed to evaluate, configure, and develop with EFM32, EFM8, and 8051 MCUs, Wireless Geckos and MCUs, ZigBee SoCs, and Wireless Modules. The software collects all these tools into four categories: Getting Started, Documentation, Compatible Tools, and Resources. The content of this section is conceptualized for the selected device or kit so that only relevant information and tools are displayed. 2.1 Selecting the Select device section will change the available tiles and the behavior of each tile in the launcher. The device can be selected by connecting a kit or board or by creating an empty solution. Examples and documentation can also be found by searching for devices or kits. Figure 2.1. Select Devices To automatically detect the appropriate device, connect the board to your PC. The board will automatically detect. Siimplicity Studio will scan any connected device and present a list. Select a device from the list. silabs.com Smart. Connected. Energy-friendly. Rev 4 Overview To see available collateral without connecting the board, click the [Solutions] tab, then click the [New Solution] button. Then, select [Blank Solution]>[Blank] and click [OK]. Figure 2.2. Create An Empty silabs.com Solution. Connected. Energy-friendly. Rev 5 Overview This will create an empty solution folder in the [Solutions] area. Click the [Add Device] button and find the device you want. Select a kit, board, or device from the resulting list to add it to the solution. Figure 2.3. Add a Device to an Empty silabs.com Solution. Connected. Energy-friendly. Rev 6 Overview Finally, click the board in the solution area to update all options on the Getting Started tab, Documentation, Compatible Tools, and Launcher Resources. Figure 2.4. Selecting Devices in the Blank Solutions section 2.2 Getting Started [Getting Started] provides a demo, which is a previously compiled example for use with starter or development kits, software examples, and application notes. These resources enable quick and easy development with supported Silicon Labs products. 2.3 The Documentation Section [Documentation] provides links to device documentation and kits. In addition, this section is connected to the software documentation available for 2.4 The Compatible Tools Section [Compatible Tools] of Simplicity Studio provides software to develop code, configure or download code to the device, and debug code on the hardware. There are also tools to measure energy consumption in real-time, estimate battery life, download demo codes to devices, analyze wireless device networks, and configure wireless applications. The default IDE option is Simplicity Idea, which is an Eclipse-based IDE provided with Simplicity Studio. To change IDE preferences and use another IDE, click the [Settings] button and select [Simplicity Studio]>[IDE Options]. All IDs detected on the system will be available as options. silabs.com Smart. Connected. Energy-friendly. Rev 7 Overview 2.5 Resource Section [Resources] provides links to useful resources, such as the Silicon Labs Community or Technical Support. silabs.com Smart. Connected. Energy-friendly. Rev 8 Energy Profiler 3. The [Energy Profiler] Energy Profiler is available from the Simplicity Studio Compatible Tools area and enables Advanced Energy Monitoring. This tool enables real-time creation of code power profiles by measuring power consumption. Some devices also support power linking used with related lines of code. Figure 3.1. Energy Profiler More information about this tool can be found in silabs.com Smart. Connected. Energy-friendly. Rev 9 Hardware Configurator 4. Hardware Configurator Hardware Configurator is part of Simplicity Studio and greatly simplifies peripheral initialization of EFM32, EFM8, and C8051 MCU by presenting peripherals and peripheral properties in the graphical user interface. Most initialization firmware can be generated by selecting peripherals and property values from combo boxes or entering register values in text boxes. Some peripherals provide calculators, such as the baud rate calculator, timer overflow rate calculator, and SPI clock rate calculator, which can be used to automatically confirm the refill register value required to generate the specified hour rate. Configurator also provides real-time property validation to ensure that the configuration is valid before downloading the code to the MCU. Figure 4.1. Simplicity Studio Hardware Configurator More information on how to use Simplicity Configurator can be found in AN0823: Simplicator Simplicity Configurator User Guide. Note Applications can be found on silicon labs webpages (or in Simplicity Studio using [Application Notes] is the silabs.com tab. Connected. Energy-friendly. Rev 10 Network Analyzer 5. Network Analyzer Network Analyzer Simplicity enables debugging of complex wireless systems. The tool captures traces of wireless network activity that can be checked directly in detail or at a other time. Figure 5.1. Simplicity The Simplicity Network Analyzer analyzer incorporates all the features of Silicon Labs Ember Desktop software. More than just a package tracker, Network Analyzer works The data tracker interface on silicon labs wireless chip to provide direct feedback from each device's baseband radio, allows supported radios to report detailed packet transmission and receiving data, such as timestamsp, link quality (or LQI), receiving sensitivity (or RSSI), and CRC pass/fail results, all without software overhead. With Simplicity Studio, the PTI-supporting Silicon Labs platform can record radio activity regardless of the firmware of the application being used, so there's no need to have a dedicated tracking device installed to capture traffic. Network Analyzer also allows capture of multiple sources simultaneously into the same log file without duplicating fake packages. This allows developers to compare how well different radios in the network hear the same transmission. In cases when details are unwanted, Network Analyzer makes it easy to understand how complex wireless protocols work. Related package events are automatically grouped into the [Transactions] panel in the capture view, allowing for faster parsing of what happens during that part of the traffic log. Access statistics quickly such as total duration, number of related packages, number of pointto-point and end-to-end attempts, and unexpected conditions such as requests with missing responses or submissions where expected recognition is lost. 5.1 Event Filtering and Data Filters can be applied to direct or previously captured traffic and can be entered through [Expression Builder] by selecting one of the suggested filters in the context menu of the [Transactions] or [Events] panel or by manually entering or editing text in the [Filter Bar]. Filters can be chained together using regular expressions to produce more specific results. Filters can also double as search criteria with other events around them for context. When developers find items of interest, add bookmarks for future reference. It's great for annotating catches during debugging or collaborating with other users. silabs.com Smart. Connected. Energy-friendly. Rev 11 Network Analyzer 5.2 Logging Results Simplicity Network Analyzer can record software-based debug activity from firmware on embedded network devices, allowing app developers to glimpse which firmware events are closely correlated in time with specific network-related events and transactions. Stepping through time-sensitive embedded network applications with a traditional debugger can interfere with the realtime behavior of network stacks, but with Network Analyzer, firmware engineers can get transcripts of all network stack API calls and stack callback functions as they happen, which can be a huge help in about why certain network activity occurs or why it has certain results. Silicon Labs even provides a Virtual UART interface to allow traditional printf output through serial wire debug lines to allow application printf statements API and callback activities. Other important software activities, such as resets, assert() violations, and startup stacks, are also captured in this [Events] list, so developers have the maximum amount of information about what happens at the time of the process. This recorded information can also be sent to Silicon Labs technical support to quickly resolve issues or answer questions. 5.3 More information about the Simplicity Network Analyzer tool can be found in UG104: Testing and Debugging Bucket Applications in Chapter 3. This document can be found on the Silicon Labs website: UG104.pdf. silabs.com Smart. Connected. Energy-friendly. Rev 12 App Maker 6. Application Builder Simplicity Studio shortens the software design cycle and reduces time to market with Application Builder, a graphical interface that works in harmony with silicon labs application frameworks to make important design choices in advance. Figure 6.1. The Silicon Labs Application Builder Simplicity App Builder, or AppBuilder, allows developers to start new projects based on existing frameworks of bestpractice app status engine code developed and tested by Silicon Labs. This framework sits a top of the wireless network stack to interact with the Hardware Abstraction Layer (or HAL) and provides application layer functionality, including the following: Start-up routines. Mechanisms for finding, joining, or forming networks. High-level APIs for creating, parsing, and handling message loads. Configure the network stack. Command-line interface for program control. Human readable debug output is tailored to the needs of developers. Merging customer-provided libraries and code modules. The AppBuilder tool is modular enough to accommodate different data sets for different versions of the application protocol, so you can choose the right version of the application-level specification when it's really important for interoperability. In certain application layers, such as ZigBee's ZCL or ZigBee Cluster Library, you can choose from one of the specified device types to get the desired mix of client and server functionality, or you can build your own custom device definitions from any of the available client and server clusters. Device-wide stack-level settings such as network node types and security methods can be easily set through the selector list in AppBuilder, and all relevant code is automatically included, with all unnecessary code automatically removed in the final output. You can even use Simplicity Studio Application Builder to configure two different networks on the same chip, such as ZigBee PRO and RF4CE. silabs.com Smart. Connected. Energy-friendly. Rev 13 Image App Maker 6.2. Stack Selections for Application 6.1 The AppBuilder Callback Interface is very versatile, with one example being the callback interface, which allows the user's application to be notified of important events or make decisions based on the process time Software designers can choose which callbacks are required for their application and can make the rest disabled so that only the selected callbacks appear in the user's Callbacks C file. The rest is safely defined as an empty stub in the framework until it is enabled. Callbacks are places where custom application code can be added above the existing Silicon Labs framework to provide the app's unique behavior and decide how it reacts. In callback implementations, developers can use the entire HAL API and stack as well as a series of Application Framework-specific APIs that often provide high-level wrappers around complex HAL functions or stacks. This API is documented in the Application Framework API Guide and found in existing sample code. Like all Silicon Labs Application Frameworks, the full source code in C is provided to help understand the implementation of this API. In all, there are more than 200 callbacks to choose from. Additionally, Application Builder automatically provides unique callbacks for each type of incoming message supported by the device's application protocol, allowing the application to handle each incoming command frame exactly as desired. silabs.com Smart. Connected. Energy-friendly. Rev 14 Application Builder 6.2 Customization To add ownership extensions to standard application protocols such as ZCL, simply describe the custom cluster, commands, attributes, and device types in the XML file that follow the Silicon Labs schema, and point AppBuilder to the XML file. The next time a stack-based configuration is created or edited, the manufacturer-specific upgrade will appear as a cluster and attributes that can be enabled, and callbacks will be automatically provided to handle any incoming commands declared in the XML data. App- Builder will even create a macro that firmware can use to quickly create outgoing messages in proprietary format, allowing the assembly of command frames and authentication arguments from the API to populate the parameters of that command. 6.3 Plugins Tab [Plugins] in AppBuilder also allows application acceleration. This tab has more than 100 plugins that each implement one or more callbacks above the Application Framework to provide advanced status engines for complex features such as message fragmentation, RF4CE device profiles, or external dataflash storage to download firmware updates. The source code for all plugins is available, so the code exists as a reference point for customized implementations. Many plugins implement functionality for certain ZigBee ZCL clusters, such as Level Control, Group, or Key Formation, so plugins can be enabled and, in many cases, provide all the messages needed to pass the ZigBee application layer compliance test. Other plugins implement device status engines and expects developers to integrate hardware behavior into the state's machines to get devices to react to changes in the status of the software. Still other plugins are provided as sample code that can be used such as a test harness to run functionality if it is not available in the system. 6.4 Generating a Project After making an option for any part of the application that the developer wants to customize, AppBuilder will generate a software project with a customized header file and an array definition in code C to represent the desired application behavior. The resulting project can be built from within the Simplicity Idea to produce a binary that can be loaded into the target wireless SoC. The creation of this project is specific to the target stack and chip versions in which the configuration is performed, but the Silicon Labs Application Framework abstracts many of these differences in its state APIs and engines, so that AppBuilder configurations can be reused across different versions of stacks or chip platforms. This makes it much easier to transition from one release version of the stack to the next. 6.5 More information about the Simplicity AppBuilder tool can be found in UG102: Ember Application Framework Developer's Guide. This document can be found on the Silicon Labs website: silabs.com Smart. Connected. Energy-friendly. Rev 15 Simplicity IDE 7. The Idea of Simplicity Idea simplicity is an Eclipse-based Integrated Development Environment (IDE) that enables code editing, downloading, and debugging for Wireless (including EM35xx), Wireless Gekes, Wireless Modules, EFM32, EFM8, and 8051 devices. Figure 7.1. Simplicity Ideas Launches the Simplicity IDE from the Simplicity Studio launcher by clicking the [Simplicity Idea] perspective, clicking the [Software Example] link, clicking the [New Project] button, or using the [Import Project] button in the [Application Notes] dialog>[See All]. 7.1 Workspace Workspaces in the Simplicity IDE are active project groupings that are displayed in [Project Explorer] view and bound to the physical location on the disk. Workspaces contain top-level IDE settings, including global defaults, window view positions, and projects in workspaces. Workspace information is contained in the metadata subdirectory (.metadata) in the workspace directory. Any projects added to the workspace will be copied to this location. The default workspace location can be viewed and modified by going to [File]>[Switch Workspace]>[Other]. 7.2 ProjectIng Simplicity Ideas Projects contains files, build options, and project settings. Projects generally exist as directories that contain sub-directories and files. The project structure that is visible in the IDE in the [Project Explorer] view is physically replicated on the disk. However, the project may also contain linked files or directories that are only pointers to files or folders outside the project directory. silabs.com Smart. Connected. Energy-friendly. Rev 16 Simplicity IDE 7.3 Importing Projects To import projects from µvision4, IAR Embedded Workbench, or 8-bit Silicon Labs IDE to IDE: 1. Open [File]>[Import]. 2. Select [Simplicity Studio]>[MCU Project]. 3. Follow the steps in the guide to complete the project import. The SDK is a collection of header files and examples for the device family. The header file defines the register and enumeration for each bitfield in the register. 7.4 Exporting Projects To Share Ide Projects Simplicity: 1. Go to [File]>[Export]. 2. Select [Simplicity Studio]>[MCU Projects to Files]. 3. Follow the steps in the guide to complete the project export. This will allow Simplicity Studio to re-import the project at a future time. silabs.com Smart. Connected. Energy-friendly. Rev 17 Simplicity IDE 7.5 Creating a New Project There are four types of projects in The Simplicity Studio: Sample Program C Blank This type of project is based on a previously existing example. The Library A Library project enables the creation of libraries, which can then be used as part of other projects. Simplicity Configurator Program This type of project enables the configuration of pins, hours, and peripherals of supported devices. This option may not exist if it is not already installed. To create a new project in Simplicity Ideas: 1. Click the [Get Started]>[New Project] button from the launcher. 2. Select the desired [Kit], [Part], and [SDK] from the drop-down menu and click [Next]. 3. Select the project type and click [Next]. For the [Example] project, select the instance and click [Next]. 4. Name the project and click [Next] or [Finish]. 5. After clicking [Next], the [Build Configurations] dialog allows setting up the build project options. The defaults are [Debug] and [Release] configurations, which are different from debug symbols and optimization settings. Alternatively, create a project directly from the Simplicity Studio launcher by clicking the [Getting Started]>[Software Example]>[See All] link and follow the guidelines. The simplicity of the IDE can detect whether the Keil C51 toolchain is not enabled. If [License Helper] is displayed, perform the following steps to enable toolchain: 1. Click the [this form] link to open the activation form in a web browser. 2. Complete and submit the form. 3. Once the registration key is received, copy the key into the text box in the [License Helper] dialog. 4. Click [OK] License Helper can also be accessed using the [Help]>[License] menu option. silabs.com Smart. Connected. Energy-friendly. Rev 18 Simplicity IDE 7.6 Views and Perspectives Perspectives is a series and configuration of saved views. Each view in perspective provides specific information for active perspective tasks. For example, when debugging a project, the [Debug] perspective displays the [Debug] view with stack tracking information, the [Registers] view, the [Breakpoints] view, the [Expression] view, and the [Uninstall] view. Switch between perspectives using the buttons in the upper-right corner of the IDE. The perspectives available by default are as follows: The [Launcher] Perspective Clicking Launcher opens Ultimate simplicity launcher page. Click the Simplicity IDE tile to return to the Development perspective. Simplicity Perspective IDE simplicity to switch to the default perspective designed for code editing and development. Configurator is used by the Simplicity Configurator tool, which enables peripheral and pin configurations for devices. Debug Starting a debug session will automatically redirect the IDE to the [Debug] perspective, which includes useful debugging tools such as the [Registers] and [Disassembly] views. The Energy Profiler will automatically switch to the Energy Profiler tool, which uses the AEM circuit in the Starter Kit to measure power consumption. Figure 7.2. Perspective In Simplicity IDE 7.7 Code Editing and Debugging Features The Simplicity of the IDE is a code editing and development environment. Editors include context highlighting, reference search, and standard features found in any modern editor. silabs.com Smart. Connected. Energy-friendly. Rev 19 Simplicity IDE Open Declaration In addition to basic features, Simplicity IDE supports many advanced code editing features. For example, the IDE automatically indexes all the code in the project to support symbol search. Code doesn't have to be fully created for the indexer to work, although certain features may not be available if, for example, the main routine() is not declared. In this example, the [Open Declaration] feature (shortcut F3) quickly finds the symbol declaration. 1. Open the interesting file by double clicking in [Project Explorer]. 2. Right click on the desired symbol to display the context menu. 3. Click [Open Declaration] to quickly navigate to the symbol definition (for example, [TMR2CN_TF2H]). 4. Studio will automatically open the file and highlight the line containing the declaration of the Symbol Of Figure 7.3. Find the Smart silabs.com Declaration. Connected. Energy-friendly. Rev 20 Simplicity IDE Content Assist Simplicity Studio also supports code completion, a feature called [Content Assist]. Content Help requires that the appropriate header file be included in the file so that the symbol is available. To use [Content Help], type the first few letters of the symbol or include a file and press [Ctrl+Space] to display a list of matching symbols. For example, to use [Content Help] to display a list of symbols starting with the character [P1]: 1. Open the file and type the desired character in the file (for example, [P1]). 2. Press [Ctrl+Space] to display the [Content Help] list. 3. Use the arrow keys or page up and down keys to see a list of matching symbols. 4. Pressing [Enter] replaces the typed character with the selected symbol. 12 Figure 7.4. Use Content Assist silabs.com Smart. Connected. Energy-friendly. Rev 21 Simplicity IDE Link with Button Editor [Link with Editor] synchronizes editor with view [Project Explorer], selected in the editor. Figure 7.5. Links with Symbol Expansion Editor Hovering over a function or macro in the editor will make the window hover with expanded information on the symbol. Figure 7.6. Expansion of silabs.com Symbols. Connected. Energy-friendly. Task List Rev 22 Idea Simplicity Task list will automatically take any comment with [TODO] in the line and consolidate it into the [Tasks] view. To do this: 1. Type [TODO] in the comment line with the desired text. 2. Go to [Window]>[Show View]>[Other], type [Tasks], and press [OK]. 3. The line [TODO] is highlighted by the clipboard in the left blue margin in the editor, and clicking the line in the [Tasks] list will jump to that place in the project. 1 Figure 7.7. Use the Smart silabs.com List. Connected. Energy-friendly. Quick Access Console IDE Simplicity Rev 23 Pressing [Ctrl+3] in Simplicity Studio brings up the quick access console to find any menu or view within the IDE. For example, press [Ctrl+3] and type [Preferences]. This lists all the Preferences menus available in Simplicity Studio. Then, select the option to open the menu. Figure 7.8. The Smart Silabs.com Quick Access Console. Connected. Energy-friendly. Rev 24 Simplicity IDE Call Hierarchy IDE includes a call stack and call hierarchy that can help debug and find where functions are called. To find the call hierarchy for a function in debug or development mode, right-click on it and select [Open Call Hierarchy]. Figure 7.9. Use The Call Stack and Call Hierarchy silabs.com Hierarchy. Connected. Energy-friendly. Rev 25 Simplicity IDE Snapshots Feature [Snapshot] is available when debugging store register values on the device at a certain point in time. Take add snapshot: 1. Click the [Snapshot] button when the core is stopped and the register value is constant and accessible to debug hardware. 2. Each [Snapshot] is listed in the [Snapshot Albums] view This feature is useful when comparing the state of two systems next to each other or viewing the state of the hardware at separate time points. To view the snapshot, switch to the [Development] perspective and go to [Window]>[Show View]>[Other] and select [Snapshot Albums]. In the list, right-click on [Snapshot] and select [Launch Snapshot]. This will open the [Debug] perspective as it was at the time of the [Snapshot], but this is not an active debug session on the hardware. To switch back to the debug session on the hardware, go to [Run]>[Debug]. 1 Images Using the Snapshot Feature silabs.com Smart. Connected. Energy-friendly. Rev 26 Simplicity IDE 7.8 General Action Common Actions when developing is described in Table 7.1 of the Simplicity IDE Development Toolbar on page 25. Table 7.1. Simplicity IDE Development Toolbar Icon Command Description Debug Build Button [Debug] starts a new debug session. Active debug session must be disconnected before starting using the same debug adapter. Tje Tje the selected project. In debug sessions, the IDE switches to the [Debug] perspective with the general actions shown in the Debug Toolbar of the IDE Simplicity Table 7.2 on page 25. If one debug adapter is available, the IDE will automatically download the code to the MCU. If more than one debug adapter is available, the IDE will ask to select a debugger. Table 7.2. Simplicity IDE Debug Toolbar Icon Command Description Debug Continue To Deable Disconnect Reset Device Step To Step Up Step Back Step Step Step [Debug] Button start a new debug session. The active debug session must be disconnected before starting the new session using the same debug adapter. The [Resume] button runs the MCU after a reset or after pressing a breakpoint. The [Suspend] button stops the MCU. The [Disconnect] button terminates the current debug session and disconnects the debug adapter. The IDE will automatically switch back to the Development perspective. The [Reset Device] button performs a hardware reset on the MCU. The [Step To] button is one step to the first line of the function. The [Top Step] button is one step above the function, executing the entire function. The [Step Return] button exits the function, performs other functions. The [Instruction Stepping Mode] button switches assembly one step. When enabled, one step will run one assembly instruction at a time. See the [Uninstall] view for assembly code that corresponds to the source code on the current execution line. The [Debug] view in the upper-left corner displays any active debug sessions. Debug adapters can only support one debug session at a time. The active debug session must be disconnected before the code can be reworked and a new debug session starts. To set a breakpoint, double-click on the blue bar to the left of the code editor or right-click on the line of code and select [Add Breakpoint]. Breakpoints can be managed using the [Breakpoints] view in the [Debug] perspective. The contents of the register can be viewed and edited using the [Registers] view. Memory can be accessed using the [Memory] view. silabs.com Smart. Connected. Energy-friendly. Image IDE Simplicity Rev 27 Navigate ide while Debugging 7.9 Running Blinky with EFM32 Device To create a project based on Blinky example using starter kit Zero Gecko: 1. From the main Simplicity Studio launcher, click the [New Project] button. 2. On the first [Project Settings] page of the project creation wizard: a. Select [Zero Gecko 3200 Starter Kit] in the [Kit] drop-down menu. B. The wizard automatically selects the device [EFM32ZG222F32] in the [Section] drop-down menu. c. Select SDK in the [SDK] drop-down menu. d. Click [Next]. 3. Select the [STK3200_blinky] example from the list and click [Done]. 4. Click The project in the [Project Explorer] view and click the [Debug] button to build and download the code to the hardware. 5. Click the [Resume] button to start running. Example. The LED should flash. 6. Press the [Disconnect] button to return to the [Development] perspective. silabs.com Smart. Connected. Energy-friendly. Rev 28 Simplicity IDE 7.10 Running Blinky with EFM8 or 8-bit Devices To create a project based on Blinky's example using the starter kit EFM8BB1:1. From the Simplicity Studio Main launcher, click the [New Project] button. 2. On the first [Project setup] page of the project creation wizard: a. Select [EFM8BB1 Starter Kit] in the [Kit] drop-down menu. B. The wizard automatically selects the device [EFM8BB10F8G] in the [Section] drop-down menu. c. Select SDK in the [SDK] drop-down menu. d. Click [Next]. 3. Select the example [EFM8BB1 Blinky] from the list and click [Finish]. 4. Click The project in the [Project Explorer] view and click the [Debug] button to build and download the code to the hardware. 5. Click the [Resume] button to start the example. The LED should flash. 6. Press the [Disconnect] button to return to perspective [Development] More Information Simplicity Studio includes detailed help information and device documentation in the tool. Help contains a description for each dialog window. To display the documentation for the dialog, click the question mark icon in the window: This will open a special panel for the dialog with additional details. Documentation in the tool can also be viewed by going to [Help]>[Help Contents] or [Help]>[Search]. Simplicity Studio also contains some interactive guides called [Cheat Sheets]. This tutorial provides guidance through the basic usage scenarios in the IDE. Access this tutorial by going to [Help]>[Cheat Sheets]. silabs.com Smart. Connected. Energy-friendly. Rev 29 Other Tools 8. Other Tools More information about other tools available in Simplicity Studio can be found on silabs.com Smart. Connected. Energy-friendly. Pdt 30 Historical Revision 9. Revision History 9.1 Revision 0.3 June 13, 2016 Updated screenshots and procedures for Simplicity Studio v Revision 0.2 February 13th, 2015 Updated formatting. Updated screenshot for Simplicity Studio v Revision 0.1 February 2014 Initial revision. silabs.com Smart. Connected. Energy-friendly. Rev 31 Simplicity Studio One-click access to MCU and wireless tools, documentation, software, source code libraries & more. Available for Windows, Mac, and Linux! IoT Portfolio SW/HW Quality Support and Community community.silabs.com Disclaimer Silicon Laboratories intends to provide customers with the latest, accurate, and in-depth documentation of all peripherals and modules available for implementing systems and software that use or intend to use Silicon Laboratory products. Characterization data, available modules and peripherals, memory size, and memory address refer to each particular device, and the Typical parameters provided may and may vary in different applications. The sample application described here is for illustrative purposes only. Silicon Silicone the right to make changes without further notice and limitation to the product information, specifications and descriptions herein, and make no warranty as to the accuracy or completeness of the information included. Silicon Laboratories will not be responsible for the consequences of using the information provided herein. This document does not imply or express the copyright licenses granted below to design or create any