

MIV_ESS Design Guide

Introduction

The MIV_ESS is a Mi-V ecosystem IP core available for the Microchip FPGA and System-on-Chip (SoC) FPGA device families. This design guide describes four different Libero[®] designs highlighting various features of the MIV_ESS IP core. The following table lists the Libero designs and the corresponding features.

Table 1. MIV Design Options and Configurations

Configuration Argument	Feature	GitHub Repository
DGC1	SPI Flash Bootstrap	PolarFire [®] Evaluation Kit
DGC2	I ² C Flash Bootstrap ¹	PolarFire [®] Avalanche Kit
DGC3	PolarFire [®] µPROM Bootstrap	PolarFire [®] Evaluation Kit
DGC4	MIV_RV32 basic design using MIV_ESS ²	PolarFire [®] Evaluation Kit

Notes:

- 1. Mikroe Bus Dual EE Click board is available from https://www.mikroe.com/dual-ee-click
- 2. This is a basic design, which allows you to debug the software and use external peripherals such as UART, GPIO, and CoreTimers.

This guide provides quick-start steps to set up the MIV_ESS IP core in a Libero SoC SmartDesign environment, generate a bitstream, and program the FPGA data to a device. It also describes how to debug the software and use a serial terminal to capture the UART output, and how to copy a program from LSRAM to off-chip memory devices.

Note: All the designs discussed in this document are available as FlashPro Express Project job files in the "FlashPro_Express_Projects" folder, under the <code>BoardName_MIV_RV32_DGCX_BaseDesign.job</code>. BoardName is the target FPGA board name and DGCX is the configuration parameter listed in the preceding table.

Table of Contents

Intr	oductio	n	1
1.	Desig	ns	3
	1.1.	SPI Flash Bootstrap	3
	1.2.	I ² C Bootstrap	5
	1.3.	μPROM Bootstrap	7
	1.4.	Basic Example	8
2.	Apper	ndix: Configuring LSRAM	. 10
	2.1.	Example Firmware	. 10
	2.2.	Configuring LSRAM Using the LSRAM Configurator	. 11
	2.3.	Configuring LSRAM Using the Configure Design Initialization Data and Memories Configurat	
			11
3.	Apper	ndix: Running a Project from SoftConsole	.14
	3.1.	Setting the System Clock Frequency	. 14
	3.2.	Setting the Peripheral Base Addresses	.15
	3.3.	Building a SoftConsole Project	
	3.4.	Creating a Deployable Hex File	
	3.5.	Communication Through UART	. 18
4.	Apper	ndix: Design Constraints	.21
5.	Apper	ndix: Running an Elf file	. 22
6.	Revis	on History	. 25
The	Micro	chip Website	.26
Pro	duct C	nange Notification Service	26
Cus	stomer	Support	. 26
Mic	rochip	Devices Code Protection Feature	. 26
Leg	al Noti	ce	. 26
Tra	demark	is	. 27
Qua	ality Ma	inagement System	. 28
	-	Sales and Service	
**0			.20

1. Designs

This section describes the four Libero design configurations DGC1 to DGC4 listed in Table 1. In addition, it also describes the example firmware and the software configuration changes that are required to make the firmware compatible with the design configurations.

1.1 SPI Flash Bootstrap

This design demonstrates the SPI bootstrap feature of the MIV_ESS IP core on the PolarFire Evaluation Kit board. The following features are enabled on the MIV_ESS core in this design.

- SPI Flash
- Bootstrap
- UART
- GPIO
 - GPIO inputs connected to Push Buttons (SW7 and SW8 on the PolarFire Evaluation Kit), as shown in the following figure.

Figure 1-1. PolarFire Evaluation Kit



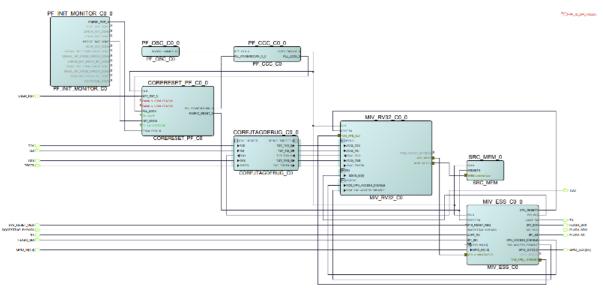
- GPIO outputs connected to user LEDs, as shown in the following figure.



Figure 1-2. LEDs on PolarFire Evaluation Kit

The following figure illustrates the SPI Flash bootstrap design configuration.

Figure 1-3. SPI Flash Boot



The Libero SoC project is available at: github.com/Mi-V-Soft-RISC-V/PolarFire-Eval-Kit. For more information on how to run this configuration, see the README.md file in the GitHub project. The DGC1 design is used for this configuration.

1.1.1 Operation

On power ON, the MIV_ESS copies a program from a Flash device to the MIV_RV32 Tightly-Coupled Memory (TCM). When the MIV_ESS releases the MIV_RV32 reset, the MIV_RV32 will boot the application. The following steps describe how this is accomplished.

1. Download the Libero SoC programming files from the GitHub link provided.

Note: This design is set up to disable the BOOTSTRAP_BYPASS, by default.

- 2. Program the bitstream to the development kit.
- 3. Open SoftConsole.
 - a. To run the provided Elf project, see 5. Appendix: Running an Elf file.
 - b. To run the provided SoftConsole project, see 3. Appendix: Running a Project from SoftConsole.
- 4. Open a serial terminal session, see the 3.5. Communication Through UART section.
- 5. Launch the debug session in SoftConsole. The program is built to copy a program from volatile to nonvolatile memory. The MIV_RV32 TCM is the target nonvolatile memory in this application design.
- 6. The serial terminal window is shown in the following figure. Select **1** to initiate copying the example firmware to SPI Flash from the LSRAM.

Figure 1-4. Bootload Menu

```
MIV_ESS Bootstrap support utility to load binary executable from LSRRAM to Non-Volatile memory

This program supports writing HEX data from Source LSRAM (@ Address 0x80000000) into a Non-Volatile memory

Choose the destination Non-Volatile memory:

Type 0 to show this menu

Type 1 copy .hex from LSRAM to SPI Flash

Type 2 copy .hex from LSRAM to MikroBus EEPROM
```

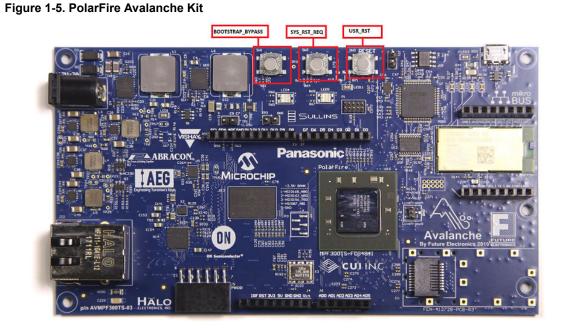
- 7. Once the program is copied, end the debug session.
- 8. Push and hold SW8 and press and release SW6 or SW7.
- 9. Release SW8 when the LEDs on the board are actively blinking.

For more information on how to update a hex file in the LSRAM, see 2. Appendix: Configuring LSRAM.

1.2 I²C Bootstrap

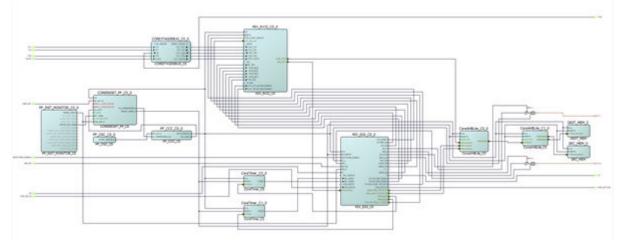
This design demonstrates the I²C bootstrap feature of the MIV_ESS core on the Avalanche Development Kit board. This design requires a Dual EE Click board from mikroBUS inserted into the mikroBUS header on the Avalanche Development Kit board. The Dual EE Click is available at Mikroe.com. The following features are enabled on the MIV_ESS core in this design.

- I²C
- Bootstrap
- UART
- GPIOs
 - GPIO inputs connected to Push Buttons (SW1 and SW2 on the PolarFire Avalanche Kit), as shown in the following figure.



The following figure illustrates the I²C bootstrap design configuration.

Figure 1-6. I²C Bootstrap



The Libero SoC project for this configuration is available at: github.com/Mi-V-Soft-RISC-V/Future-Avalanche-Board. For more information on how to run a configuration, see the README.md file in the GitHub project. The DGC2 design is used for this configuration.

1.2.1 Operation

On power ON, the MIV_ESS copies a program from the I²C Flash device to the MIV_RV32 Tightly-Coupled Memory (TCM). When the MIV_ESS releases the MIV_RV32 reset, the MIV_RV32 will boot the application. The following steps describe how this is accomplished.

- 1. Download the Libero SoC programing files from the GitHub link provided.
- 2. Connect the Dual EE Click board to the Avalanche Board mikroBUS header.
- 3. Program the bitstream to the development kit. **Note:** This design is set up to disable the BOOTSTRAP_BYPASS, by default.
- 4. Open SoftConsole.

- a. To run the provided Elf project, see 5. Appendix: Running an Elf file.
- b. To run the provided SoftConsole project, see 3. Appendix: Running a Project from SoftConsole.
- 5. Open a serial terminal session, see 3.5. Communication Through UART section.
- 6. Launch the debug session. The application is built to copy a program from volatile to nonvolatile memory. The MIV_RV32 TCM is the target memory in this software project.
- 7. A menu is displayed in the terminal window, as shown in the following figure. Select **2** to initiate a copy operation to I²C Flash from LSRAM.

Figure 1-7. Bootload Menu

```
MIV_ESS Bootstrap support utility to load binary executable from LSRRAM to Non-Volatile memory

This program supports writing HEX data from Source LSRAM (@ Address 0x800000000) into a Non-Volatile memory

Choose the destination Non-Volatile memory:

Type 0 to show this menu

Type 1 copy .hex from LSRAM to SPI Flash

Type 2 copy .hex from LSRAM to MikroBus EEPROM
```

- 8. Once the program is copied, end the debug session.
- 9. Push and hold SW1 to disable BOOTSTRAP_BYPASS.
- 10. Press and release SW2 to perform a system reset request.
- 11. Release SW1 when the LEDs on the board are actively blinking. The program that is stored in I²C Flash is copied by the bootstrap into the MIV_R32 TCM, and the MIV_RV32 starts executing the example firmware.

For more information on how to update a hex file in the LSRAM, see 2. Appendix: Configuring LSRAM.

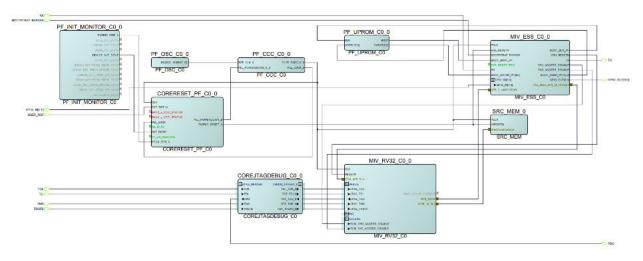
1.3 µPROM Bootstrap

This design demonstrates the µPROM bootstrap feature of the MIV_ESS core on the PolarFire Evaluation Kit board. The following features are enabled on the MIV_ESS core in this design.

- PF_uPROM
- Bootstrap
- UART
- GPIOs

The following figure illustrates the µPROM bootstrap design configuration.

Figure 1-8. µPROM Boot Load



The Libero SoC project is available at: github.com/Mi-V-Soft-RISC-V/PolarFire-Eval-Kit. The DGC3 design is used for this configuration.

1.3.1 Operation

On power ON, the MIV_ESS copies a program from the µPROM device to the MIV_RV32 Tightly-Coupled Memory (TCM). When the MIV_ESS releases the MIV_RV32 reset, the MIV_RV32 will boot the application. The following steps describe how this is accomplished.

- 1. Download the Libero SoC programing files from the GitHub link provided.
 - Note: This design is set up to disable the BOOTSTRAP_BYPASS, by default.
- 2. Program the bitstream to the development kit.
- 3. Open a serial terminal session, see 3.5. Communication Through UART section.
- 4. Press the push button reset SW6 on the PolarFire Evaluation Kit.
- 5. The bootstrap copies a program from μ PROM to MIV_RV32 TCM. When the program is copied, MIV_RV32 can execute the program in TCM.

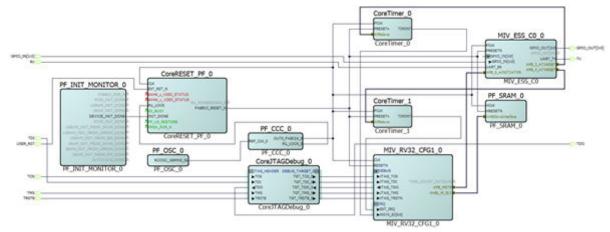
1.4 Basic Example

This section demonstrates how to set up a basic design with MIV_ESS and MIV_RV32 cores for a PolarFire Evaluation Kit. It is based on the MIV_RV32 Design Guide with the DirectCore peripherals, CoreGPIO and CoreUART, replaced with the MIV_ESS UART and GPIO. The following features are enabled on the MIV_ESS core in this design.

- UART
- GPIOs
- APB Target interfaces (used by CoreTimers)

The following figure illustrates a sample design with MIV_ESS and MIV_RV32 cores in the SmartDesign.

Figure 1-9. Basic Example



The Libero SoC project is available at: github.com/Mi-V-Soft-RISC-V/PolarFire-Eval-Kit. The DGC4 design is used for this configuration.

1.4.1 Operation

The following steps describe how to set up a design.

- 1. Download the Libero SoC programing files from the GitHub link provided.
- 2. Open a terminal session, see 3.5. Communication Through UART
- 3. Program the bitstream to the development kit.
- 4. On a reset SW6, the LEDs start blinking and a welcome message appears on the terminal.

You can run a debug example, see 3. Appendix: Running a Project from SoftConsole. For more information on how to update a hex file in the LSRAM, see 2. Appendix: Configuring LSRAM.

2. Appendix: Configuring LSRAM

To boot a program on power-on-reset, you can configure the PolarFire LSRAM using one of the following two methods.

- Configuring the LSRAM memory using the LSRAM configurator.
- Configuring the LSRAM memory using the Configure Design Initialization Data and Memories configurator.

2.1 Example Firmware

The example firmware included in the PF_SRAM in the Libero SoC design is built from the MIV_RV32 HAL example in the Firmware Catalog bundled with Libero. The fpga_design_config.h file from the Firmware Catalog is modified to use the MIV_ESS components and the default linker script is modified to use MIV_RV32 TCM.

The following changes are required in the fpga_design_config.h file.

Figure 2-1. Changes in fpga_design_config.h

Original fpga_design_config.h

44 * Non-memory Peripheral base addresses 45 * Format of define is: * <corename>_<instance>_BASE_ADDR 46 47 * The <instance> field is optional if there is only one instance of the core 48 * in the design 49 */ 50 #define COREUARTAPB0 BASE ADDR 0x70001000UL 51 #define COREGPIO_IN_BASE_ADDR 0x70002000UL 52 #define CORETIMER0 BASE ADDR 0x70003000UL 53 #define CORETIMER1 BASE ADDR 0x70004000UL 54 #define COREGPIO OUT BASE ADDR 0x70005000UL 55 #define FLASH CORE SPI BASE 0x70006000UL 56 #define CORE16550_BASE_ADDR 0x70007000UL Updated fpga_design_config.h 44 * Non-memory Peripheral base addresses 45 * Format of define is: 46 * <corename> <instance> BASE ADDR 47 * The <instance> field is optional if there is only one instance of the core 48 * in the design 49 */ 50 **#define** MIV_PLIC_BASE_ADDR 0x70000000UL 51 #define COREUARTAP0 BASE ADDR 0x71000000UL 52 #define MIV MTIMER BASE ADDR 0x72000000UL 53 #define CORETIMER0 BASE ADDR 0x73000000UL 54 #define CORETIMER1 BASE ADDR 0x74000000UL 55 #define COREGPIO INOUT BASE ADDR 0x75000000UL 56 #define FLASH CORE SPI BASE ADDR 0x76000000UL 57 #define PF UPROM BASE ADDR 0x77000000UL 58 #define uDMA BASE ADDR 0x78000000UL 59 #define MIV WDT BASE ADDR 0x79000000UL 60 #define MIV I2C BASE ADDR 0x7A000000UL

The following changes are required to the linker script (miv-rv32-ram-imc.ld). The RAM ORIGIN needs to be updated from 0x80000000 to 0x40000000, and the RAM_START_ADDRESS needs to be updated from 0x80000000 to 0x40000000, as shown in the following figure.

Figure 2-2. Changes to Linker Script

```
MEMORY
{
   ram (rwx) : ORIGIN = 0x80000000, LENGTH = 16k
}
RAM START ADDRESS
                   = 0 \times 80000000;
                                        /* Must be the same value MEMORY region ram ORIGIN above. */
                                        /* Must be the same value MEMORY region ram LENGTH above. */
RAM SIZE
                    = 16k;
                    = 1k;
                                        /* needs to be calculated for your application */
STACK SIZE
HEAP SIZE
                    = 0k;
                                        /* needs to be calculated for your application */
```

2.2 Configuring LSRAM Using the LSRAM Configurator

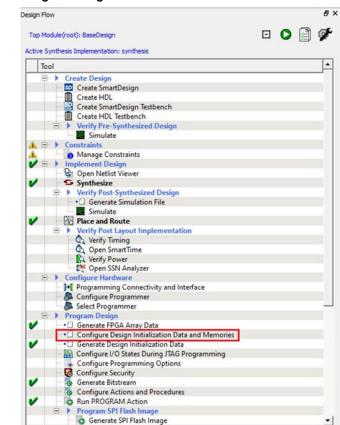
The following steps describe how to configure LSRAM using the LSRAM Configurator.

- 1. Right click on the design hierarchy and select **Configure**.
- 2. Click on the Memory Initialization Settings tab.
- 3. Select the Initialize RAM at Power up check box.
- 4. Select the hex file created by SoftConsole while building a project, see Creating a Deployable Hex File.

Note: The maximum size of the hex file is 32 KB.

2.3 Configuring LSRAM Using the Configure Design Initialization Data and Memories Configurator

If you have run the bitstream generation and want to add or change the client in an LSRAM, you can configure it using the **Configure Design Initialization Data and Memories** configurator as shown in the following figure.



Elaura 0.0 Dec	Inn Flaur Canfle	www. Deelaw Indial	Instan Data and	Manaaniaa
Fluure 2-3. Des	ian riow: Coniid	gure Design Initial	ization Data and	wemories

The following steps describe how to configure LSRAM using the Configure Design Initialization Data and Memories Configurator.

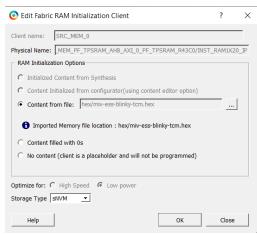
- 1. Ensure to run the Generate FPGA Array Data step of the design flow.
- 2. Run the design flow to complete the "Generate FPGA Array Data" step in the design flow.
- 3. Click on the Fabric RAMs tab.
- 4. Double click on the RAM to be initialized to open its configurator.

Figure 2-4. RAM Initialization GUI

Design Initialization uPROM sNVM SPI Flash Fabric RAMs Discard Apph Help Usage statistics Clients LSRAM Memory Initialize all clients from: Initialize all Clients from sWM Edit Load design configuration • Available Memory(Bytes): 2437120 Filter out Inferred RAMs Used Memory(Bytes): 204800 PORTA Free Memory(Bytes) : 2232320 Logical Instance Name Depth * Width 8192x32 1 MIV_RV32_C0_0/MIV_RV32_C0_0/u_opsrv_0/gen_tcm0.u_opsrv_TCM_0/tcm_ram.u_ram_0/mem[31:0] 2 MIV_RV32_C0_0/MIV_RV32_C0_0/u_opsrv_0/u_core_0/u_expipe_0/gen_gpr_ram.u_gpr_0/gen_gpr.u_gpr_array_0/mem(31:0) 32x32 3 MIV_RV32_C0_0/MIV_RV32_C0_0/u_opsrv_0/u_core_0/u_expipe_0/gen_gpr_ram.u_gpr_0/gen_gpr.u_gpr_array_0/mem_1[31:0] 32x32 4 SRC_MEM_0 32768x40 Used space Free space **USRAM Memory** Available Memory(Bytes): 266112 Used Memory(Bytes): 576 Free Memory(Bytes) : 265536

5. Under **Ram Initialization Options**, select **Content from file** option and select the hex file that SoftConsole generates while building a project. See 3.4. Creating a Deployable Hex File

6. Choose the appropriate storage location, as shown in the following figure. For example, **sNVM**. **Figure 2-5. RAM Initialization Client**



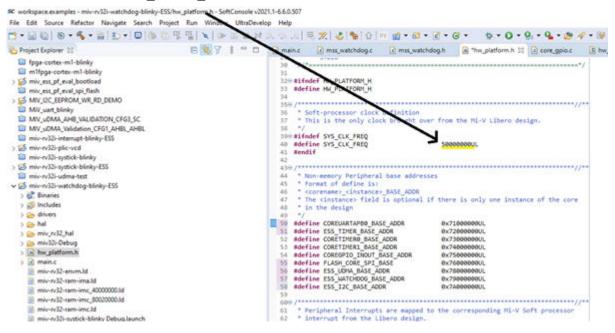
- 7. Click OK.
- 8. Click Apply in the Fabric RAMs tab.

3. Appendix: Running a Project from SoftConsole

This chapter describes how to build and run a project in SoftConsole for the MIV_RV32 core. Open SoftConsole and an example project.

The system clock frequency and the peripheral base addresses of the MIV_ESS components are set in the fpga_design_config.h file, as shown in the following figure.

Figure 3-1. Update to fpga_design_config.h System Clock and Peripheral Addresses

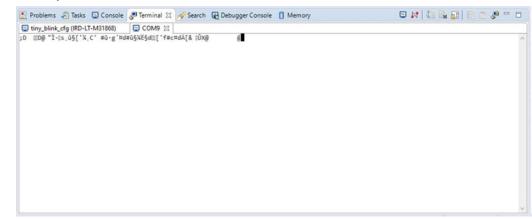


3.1 Setting the System Clock Frequency

If UART is used in the design, the system clock frequency must be updated in the fpga_design_config.h file by changing the #define SYS_CLK_FREQ to the clock frequency. An example of unexpected behavior is shown the following figure.

Note: The system clock frequency value must be in hertz.

Figure 3-2. Unexpected Behavior on the Terminal



3.2 Setting the Peripheral Base Addresses

The fpga_design_config.h file is used to set the base address for peripherals. You can find the base address of a peripheral in the Libero generated project memory map view, as shown in the following figure.

Figure 3-3. Generated Memory Map in Libero SoC

×	Reports 🗗 🗙 🛛 Constraint Manager	8	× spi_constraints.pdc	TOP & X	io_const	traints.pdc 🗗	× Star
	🔂 🗸 🔂 📭 🖻 🚃 🕅 🤅	С) 🔍 🔂 🤇	₽	🛛 🛛 A 🛛 🛃	Filter Ne
		1	SO View Memory Map				
			🔍 🛨 🖃 Show Target	is 🗆 🕴 🔺	0		
			Initiator/Bus/Bridge/Peripheral	Offset Address	Range	High Address	DRC
J		С	MIV_ESS_C0_0/CoreGPIO_0:APB_bif	0x0100_0000 0x0500_0000 0x0600_0000	16MB 16MB 16MB	0x01FF_FFFF 0x05FF_FFFF 0x06FF_FFFF	

The peripheral module address [27:0] in the fpga_design_config.h file must match the address in Libero for the peripheral to function correctly. This upper nibble [31:28] is determined by the APB Initiator connected to the MIV_ESS core. In MIV_RV32, the default APB Initiator address is 0x7000_0000, which corresponds to the address shown in the following figure.

Figure 3-4. MIV_RV32 Peripheral Port Base Addresses

Configurator		- 0
Mi-V RV32IMC Configuresemi:HIV:HIV_RV32IMC:2.1.100	jurator	
Configuration Memory Map		
AHB Master Address Start Address: Upper 16bits (Hex):	000 Lower 16bits (Hex)	
End Address: Upper 16bits (Hex):		
APB Master Address		
Start Address: Upper 16bits (Hex):	000 Lower 16bts (Hex): 0x0	
End Address: Upper 16bits (Hex): 0x	fff Lower 16bits (Hex): Ouffff	

3.3 Building a SoftConsole Project

1. Right click on the project and select (for example) **Build Configurations > Set Active > miv32i-Debug**, as shown in the following figure.

i ig	Jule 5-5. Setting the Bullu	-			
	New Go Into	>			
	Open in New Window				
	Show In	Alt+Shift+W >			
	Show in Local Terminal	>			
D	Сору	Ctrl+C			
Ē	Paste	Ctrl+V			
×	Delete	Delete			
	Source	>			
	Move				
	Rename	F2			
è	Import				
4	Export				
8	Robot Framework	>			
	Build Project				
	Clean Project				
8	Refresh	F5			
	Close Project				
	Close Unrelated Project				
	Build Configurations	>	Set Active >	~	1 miv32i-Debug
	Build Targets	>	Manage		2 miv32i-Release
	Index	>	Build All		3 miv32ima-Debug
	Profiling Tools	>	Clean All		4 miv32ima-Release
0	Run As	>	Build Selected		5 miv32imc-Debug
*	Debug As	>			6 miv32imc-Release

Figure 3-5. Setting the Build Configurations

- 2. Click the **Build** icon. In this case, the project is built for miv32i-Debug.
- 3. Click the **Debug** icon.
- 4. If you are debugging this project for the first time, you must select a debug configuration as shown in the following figure.

The state of the second state of the	on prins prins prins Gibble di S-O-S-S-S-S-S-S-S-S-S-S-S-S-S-S-S-S-S-S				0 10 200
the second se					
Prestation II C C I * C		- 0	All Reposition 11.1	Courses to a	
, 🛃 min-in (D-autholog string, 115	int i int i int i int i inter all the state of the st	í	Add and signs	-	tas wa
	(in anis) Parties				
	the set of				
	2004 () 2004 () 200				
	tet alette, tarvet, valan, et alette, tarvet,				
	ant_intro_ent() total Execution(_ent()				
	and pulled to ovringing own, coment wheth a vig builts mught				
	101 // Installating With 1/ 101 // Installating With 1/ 101 // Philiphic Sect. Constraints, Society, Mar. 2010, 497, 10, 2017, 2011.				
	Color (* The reference strategies provided on a BBDg have the Class configuration for the configuration of the Strategies of the Class Configuration a configuration and the theory of the configuration can replies (changed by the Classica configuration). The Configuration can replies (changed by the Classica configuration) with the OPD configurations. Taked, then you will need to restrike the theory of the OPD configurations. Taked to restl. Strend to restrike the strend strends, and the Constant?				
	1011 (* sel the setue vilae *) 1011 (* sel the setue vilae *) 1011 (*)				
	26 /* gat the Sounds */ read rol = 970, get Inputsing gain, Inputsi- 1		re-lacable 12		Dimetrie I *
O Debug SI to home - D			Name	her	Take
E Statistical	and Arry 10 percent as one, why 10 had and any, manading other special in their		in tout	Citree	1540
 Provid #1 (Supported: Swatpoort) main() at makes (Int SetEntitie) 	the 24 July Law the sampling theor 17				
all operations all received and places	and a starting the second sector of				
	In house of theirs O'cause of a house Quantum house	**************************************			
	me-mill-watching-blog-blog-blog-blog-plot-(percicit desaging) (2006) Namination (201)				
	Linear Street and Linear Street				
	(2011) execute (4 (211) (2014) execute (201) (2014) simple (201) (2014) execute (201)				
	(1912)				

Figure 3-6. Selecting Debug Configuration

Once you select a debug configuration, click **Debug**. This downloads the program to your device and take you to the debug perspective. The debug session gets launched and connected to the target device.
 Figure 3-7. Downloading Program

<pre>Classes to the classes to the c</pre>	W sortspannanger, municipi automogenisky 2020 Die Sale Jaurie Refector Sarigete Segrit D	ant St Den Handen Data			and the second
<pre>bit multiple mul</pre>					
<pre>i la manufe and generality i la manufe and general i la manufe and generality i la manufe and genera</pre>			- 5	Whereastern 11 March 1998	· 티블루(우 프 노) 전 전 트 트)
<pre></pre>	→ Lat monolity and help the property (15)	<pre>cont staty * "Productional further form the relation of the second function of the second state of the second state is set in the second state of the second state of the second state state is set in the second state of the second state of the second state state is set in the second state of the second state of the second state state is set in the second state of the second state of the second state of second state is set in the second state of the second state of the second state of the second state of the second state of the second state of the second state of the second state of the second state of the second state of the second state of the second state of the second state of the second state of the second state is the second state of the second state of the second state of the second state of the second state of the second state of the second state of the second state of the second state of the second state of the second state of the second state of the second state of the second state is the second state of the second state of the second state of the second state is the second state of the second state of the second state of the second state of the second state of the second state of the second state of the second state of the second state of the second state of the second state of the second state of the second state of the second st</pre>			Yas Was
• Divery II: Subscreen * Diversity in and the Static					
¹ O (1) ¹		The of gat the Seals Tr			
Improvide stategy sta	O Dates II to hanne - D	11 read_cal = 9/20_get_linexts(%g.geis_linext));			
Internet of the set of the s	 The scale of the set of the set	(iv) and participation of the second state			
ee - chi-setting talay (Bis will fam) (Mit Speel)(20 Settinged) (1981) second to (10) (1981) second to (10) (1981) second (10) (1983) second (10) (1983) second (10) (1983) second (10)	2 reciti antress et phase				
		Here All Constrainty Briefs (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998) (1998)	• X S (4 5 = 10 1 4 0 - 13 - 1 5		
	Reno selected				

3.4 Creating a Deployable Hex File

The following steps describe how to create a hex file in SoftConsole.

- 1. Right click and open the Project Properties and select Project Properties.
- 2. Select the C/C++ Build > Settings > GNU RISC-V Cross Create Flash Image > General.

3. In the **Other Flags** window, add where the MEMORY_ADDRESS is the base address in the linker. For example,

0x4000000

--change-section-lma *-MEMORY_ADDRESS

- 4. Build the project.
- 5. Click Apply and Close.

A build configuration is created, which allows you to create deployable builds without repeating this step. For more information, see the *SoftConsole Release Notes*.

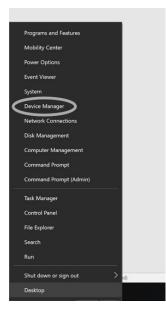
3.5 Communication Through UART

To communicate through UART, you need the following.

- The number of the COM port you are using and its baud rate.
- · A serial communication client (the built-in terminal in SoftConsole is used in this example).

For example, you can use **Device Manager** on Windows to find your device's COM port. To open it, right click the **Start** button and select **Device Manager**.

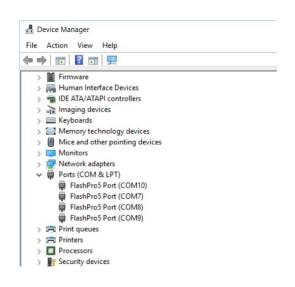
Figure 3-8. Opening Device Manager



Note:

Your device's COM port is one of the ports listed under **Ports (COM & LPT)**.

Figure 3-9. COM Ports



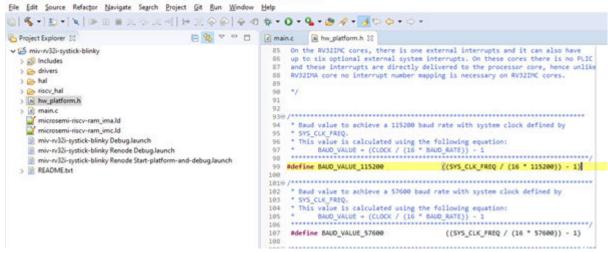
Note:

You might have to try them all to find your device's port.

You can find the baud rate for the UART connection in the fpga_design_config.h file within the SoftConsole project, see Setting the System Clock Frequency.

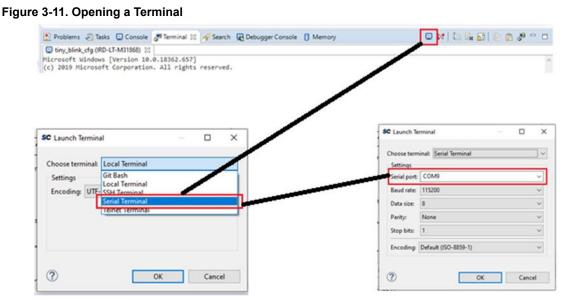
Figure 3-10. Baud Rate Location

SC workspace.rv32imc-hal-v3 - miv-rv32i-systick-blinky/hw_platform.h - Microsemi SoftConsole v6.2.0.251



The following steps describe how to communicate through UART.

- 1. Open the serial communication client within SoftConsole by navigating to **Window > Show View > Terminal**.
- 2. Input the values for your COM port and baud rate, as shown in the following figure.



3. Run your program and the UART output will be displayed in the communication client's terminal, as shown in the following figure.

Figure 3-12. FPGA UART Output

🖹 Problems 🖉 Tasks 📮 Console	🖉 Terminal 🕴 🛷 Search	Debugger Console 🚺 Memory	🔍 🕺 🖄 🕼 🖓 👘 🕅 🖓 🔍 🗆
Univ_blink_cfg (IRD-LT-M31868)	COM9 🖾		
HelloWorld from Microsemi HelloWorld from Microsemi HelloWorld from Microsemi HelloWorld from Microsemi	i PolarFire FPGA i PolarFire FPGA		^

4. Appendix: Design Constraints

The following are constraints for the PolarFire Evaluation Kit design. The pin constraints can be modified for other development kits. See the example Libero SoC designs on GitHub for more example design constraints.

```
# -- User PushButtons I/O -
set io -port name {GPIO IN[1]} \
-pin_name B27 \
-fixed true \
 -DIRECTION INPUT
set io -port name {GPIO IN[0]} \
 -pin name C21 \
 -fixed true \
-DIRECTION INPUT
set io -port name DEVRST N \setminus
-pin name K22 \
-fixed true \
-DIRECTION INPUT
# -- LEDs I/O --#
set io -port name {GPIO OUT[0]} \
-pin_name F22 \
 -fixed true \
-DIRECTION OUTPUT
set io -port name {GPIO OUT[1]} \
-pin name B26 \
 -fixed true \
-DIRECTION OUTPUT
set io -port name {GPIO OUT[2]} \
 -pin name C26 \
 -fixed true \
-DIRECTION OUTPUT
set_io -port_name {GPIO_OUT[3]} \
-pin name D25 \
 -fixed true \
-DIRECTION OUTPUT
# -- UART RX/TX --
set io -port name RX \setminus
-pin name H18 \
 -fixed true \
-DIRECTION INPUT
set io -port name TX \
 -pin name G17
 -fixed true \
 -DIRECTION OUTPUT
```

The following steps describe how to add constraints.

- 1. Add Timing Constraints to the Design Flow tab and select Manage Constraints.
- 2. Under the Timing tab, select New and name the file as io_jtag_constraints.
- 3. Copy the following constraints into this file. #Constraining the JTAG clock to 6 MHz

```
create_clock -name {TCK} -period 166.67 -waveform {0 83.33} [ get_ports { TCK } ]
set_clock_groups -name {async1} -asynchronous -group [ get_clocks { PF_CCC_C0_0/
PF_CCC_C0_0/pll_inst_0/OUT0 } ] -group [ get_clocks { TCK } ]
```

- 4. Save the constraints file, and associate it with Synthesis, Place-and-Route, and Timing Verification by selecting each of the check boxes.
- 5. To add the derived constraints, select **Derive Constraints** in the **Timing** tab.
- 6. Save the **Timing Constraints**.

5. Appendix: Running an Elf file

Use the following steps to run the provided ${\tt Elf}$ file.

- 1. Create an empty workspace by extracting the workspace.empty.zip file from <SoftConsole-installdirectory>\extras.
- 2. Open SoftConsole and locate workspace.empty, which is extracted in the previous step.
 - Figure 5-1. Creating a New Workspace

sc SoftConsole v2021.1-6.6.0.507 Launcher		×
Select a directory as workspace		
SoftConsole v2021.1-6.6.0.507 uses the workspace directory to store its preferences an	nd development artifacts.	
Workspace: C:\Microchip\SoftConsole-v2021.1\extras\workspace.empty	✓ Browse	
Use this as the default and do not ask again		
▼ Recent Workspaces		
/extras/workspace.examples		
workspace.temp		
workspace2021p1		
	Launch Cancel	

- 3. Select Launch.
- 4. From the empty workspace, create a new Debug Configuration.
 - a. From **Run > Debug Configurations**, double click **GDB OpenOCD Debugging**. This creates a new debug configuration called New_configuration.

Figure 5-2. Creating a New Debug Configuration

2 2 3 B X B 7 .	Name New_configuration				
type like text	🔝 Main 🔷 Debugger 🕨 Startup 🧤 Sox	era 🔲 Common 😤 SVD Path			
- GDB OpenOCD Debugging	Property				
New configuration Leunch Group	L	1	Browse_		
> 🍯 UltraDevelop Agent	C/C=+ Application		0.0		
		juriables	Browse		
	Build (if required) before launching Build Configuration, Select Automatically				
	C Enable auto build	O Disable auto-build			
	Use workspace settings	Configure Workspace Settings.			

5. In the **Main** window, select **C/C++ Applications** using the **Browse** button, then select the provided .elf file as shown in the following figure.

Figure 5-3. Selecting the elf File

eate, manage, and run configurations			-			
			N.			
] 🖻 🏚 📓 🗙 🖻 🖓 🔹	Name: New_configuration (1)					
ype filter text	📄 Main 🚿 Debugger 🕨 Startup 🧤 So	urce 🔲 Common 😤 SVD Path				
GDB OpenOCD Debugging	Project:					
Launch Group			Browse			
New_configuration	C/C++ Application:					
Olitabevelop Agent	C:\Users\ciaran.lappin\Downloads\12C_writin	g_and_booting_design_proj\DGC2_PF_I2C_BOOT_EEPROM_WRITE	.elf			
		<u>V</u> ariables Searc <u>h</u> Project	ct B <u>r</u> owse			
	Build (if required) before launching					
	Build Configuration: Select Automatically		~			
	O Enable auto build	Disable auto build				
	O Use workspace settings	Configure Workspace Settings				
ter matched 6 of 23 items		Revert	Apply			
ter matched 6 of 23 items			J U. Shekara			

- 6. Select the **Debugger** tab to set up OpenOCD and GDB.
 - a. To set up OpenOCD, check the Start OpenOCD locally check box and browse to the OpenOCD path <SoftConsole-install-directory>\openocd\bin\openocd.exe as shown in the following figure.

Figure 5-4. Selecting OpenOCD

	Name New_configur	ration (1)		
type filter text ~ [C] GDB OpenOCD Debugging	Main 🍄 Debugs OpenOCD Setup	ger 🔪 🐌 Startup 🏪 Source 🛄 Common 🐕 SVD Path		
C New_configuration (1) C Launch Group New_configuration	Start OpenOCD Executable path:	locally C/\Microchip\SoftConsole-v2021.1\openocd\bin\openocd.exe	Browse	Variables
S UltraDevelop Agent	Actual executable:	C:\Microchip\SoftConsole-v2021.1\openocd\bin\openocd.exe		
	GDB port: Telnet port: Tcl port:	(to change it use the <u>plobal</u> or <u>workspace</u> preferences pages or the <u>project</u> 3333 4444 6666	(properties page)	
	Config options:	file board/microsemi-riscv.cfg		¢
	2 Allocate consol	e for OpenOCD Allocate console for the te	inet connection	
	GDB Client Setup			
	Executable name:	S(cross_prefix)gdbS(cross_suffix)	Browse	Variables
				1
		\$(cross_prefix)gdb\$(cross_suffix)		

b. To set up GDB, browse to the GDB path <SoftConsole-install-directory>\riscv-unknown-elf-gcc\bin\riscv64-unknown-elf-gdb.exe.

C Debug Configurations		
reate, manage, and run configurations		1
1 2 2 1 1 X B 7 •	Name New_configuration (1)	,
GOB OpenOCD Debugging C New_configuration (1) G Learch Group Sev_configuration S Vev_configuration S UtraDevelop Agent	Main @ Debugger Startup & Source Common & SVD Path OpenOCD Setup Start OpenOCD locally	
	Executable path: C:\Microchip\SoftConsole-v2021.1\openocd\bin\openocd.exe Browse	Variables
	Actual executable: Cl/Microchip/SoftConsole-v2021.1\openocd\bin\openocd.exe	
	(to change it use the <u>global</u> or <u>workspace</u> preferences pages or the <u>project</u> properties page) GDB port: <u>3333</u> Telnet port: <u>444</u>	
	Tcl port: 6666	
	Config options:file board/microsemi-riscv.cfg	Q
	Allocate console for OpenOCD Allocate console for the telnet connection	
	GDB Client Setup	
	Start GDB session	
	Executable name: C:\Microchip\SoftConsole-v2021.1'u/scv-unknown-elf-gcc\bin\v/scv64-unknown-el Browse	
	Actual executable: C!\Microchip\SoftConsole-v2021.1\riscv-unknown-elf-gcc\bin\riscv64-unknown-elf-gdb.exe	
	Other options:	
	Commands: set Starget_riscv = 1 set menn inaccessible-by-default off set arch hiscxvn32 set remotetimeout 7	Ĵ
	Remote Target	
	Host name or IP address: localhost	
	Port number: 3333	
	Force thread list update on suspend	Restore defa
ter matched 6 of 23 items	Revert	Apply
2	Debug	Close

Figure 5-5. Selecting the GDB Location

- 7. Click Apply.
- 8. Click **Debug** to launch the debug session.

6. Revision History

Revision	Date	Description
A	01/2022	Initial Revision

The Microchip Website

Microchip provides online support via our website at www.microchip.com/. This website is used to make files and information easily available to customers. Some of the content available includes:

- **Product Support** Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip design partner program member listing
- **Business of Microchip** Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Product Change Notification Service

Microchip's product change notification service helps keep customers current on Microchip products. Subscribers will receive email notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, go to www.microchip.com/pcn and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: www.microchip.com/support

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip products:

- · Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip product is strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is "unbreakable". Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.

Legal Notice

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. Contact your local Microchip sales office for additional support or, obtain additional support at www.microchip.com/en-us/support/design-help/client-support-services.

THIS INFORMATION IS PROVIDED BY MICROCHIP "AS IS". MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION.

Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, Adaptec, AnyRate, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, CryptoMemory, CryptoRF, dsPIC, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Kleer, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AgileSwitch, APT, ClockWorks, The Embedded Control Solutions Company, EtherSynch, Flashtec, Hyper Speed Control, HyperLight Load, IntelliMOS, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet- Wire, SmartFusion, SyncWorld, Temux, TimeCesium, TimeHub, TimePictra, TimeProvider, TrueTime, WinPath, and ZL are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, Augmented Switching, BlueSky, BodyCom, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, Espresso T1S, EtherGREEN, GridTime, IdealBridge, In-Circuit Serial Programming, ICSP, INICnet, Intelligent Paralleling, Inter-Chip Connectivity, JitterBlocker, Knob-on-Display, maxCrypto, maxView, memBrain, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, NVM Express, NVMe, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, RTAX, RTG4, SAM-ICE, Serial Quad I/O, simpleMAP, SimpliPHY, SmartBuffer, SmartHLS, SMART-I.S., storClad, SQI, SuperSwitcher, SuperSwitcher II, Switchtec, SynchroPHY, Total Endurance, TSHARC, USBCheck, VariSense, VectorBlox, VeriPHY, ViewSpan, WiperLock, XpressConnect, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

The Adaptec logo, Frequency on Demand, Silicon Storage Technology, Symmcom, and Trusted Time are registered trademarks of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

[©] 2022, Microchip Technology Incorporated and its subsidiaries. All Rights Reserved.

ISBN: 978-1-5224-9661-8

Quality Management System

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.



Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office	Australia - Sydney	India - Bangalore	Austria - Wels
355 West Chandler Blvd.	Tel: 61-2-9868-6733	Tel: 91-80-3090-4444	Tel: 43-7242-2244-39
Chandler, AZ 85224-6199	China - Beijing	India - New Delhi	Fax: 43-7242-2244-393
el: 480-792-7200	Tel: 86-10-8569-7000	Tel: 91-11-4160-8631	Denmark - Copenhager
ax: 480-792-7277	China - Chengdu	India - Pune	Tel: 45-4485-5910
echnical Support:	Tel: 86-28-8665-5511	Tel: 91-20-4121-0141	Fax: 45-4485-2829
ww.microchip.com/support	China - Chongqing	Japan - Osaka	Finland - Espoo
Veb Address:	Tel: 86-23-8980-9588	Tel: 81-6-6152-7160	Tel: 358-9-4520-820
ww.microchip.com	China - Dongguan	Japan - Tokyo	France - Paris
tlanta	Tel: 86-769-8702-9880	Tel: 81-3-6880- 3770	Tel: 33-1-69-53-63-20
uluth, GA	China - Guangzhou	Korea - Daegu	Fax: 33-1-69-30-90-79
el: 678-957-9614	Tel: 86-20-8755-8029	Tel: 82-53-744-4301	Germany - Garching
ax: 678-957-1455		Korea - Seoul	Tel: 49-8931-9700
	China - Hangzhou Tel: 86-571-8792-8115	Tel: 82-2-554-7200	
ustin, TX			Germany - Haan
el: 512-257-3370	China - Hong Kong SAR	Malaysia - Kuala Lumpur	Tel: 49-2129-3766400
oston	Tel: 852-2943-5100	Tel: 60-3-7651-7906	Germany - Heilbronn Tel: 49-7131-72400
estborough, MA	China - Nanjing	Malaysia - Penang	
el: 774-760-0087	Tel: 86-25-8473-2460	Tel: 60-4-227-8870	Germany - Karlsruhe
ax: 774-760-0088	China - Qingdao	Philippines - Manila	Tel: 49-721-625370
hicago	Tel: 86-532-8502-7355	Tel: 63-2-634-9065	Germany - Munich
asca, IL	China - Shanghai	Singapore	Tel: 49-89-627-144-0
el: 630-285-0071	Tel: 86-21-3326-8000	Tel: 65-6334-8870	Fax: 49-89-627-144-44
ax: 630-285-0075	China - Shenyang	Taiwan - Hsin Chu	Germany - Rosenheim
allas	Tel: 86-24-2334-2829	Tel: 886-3-577-8366	Tel: 49-8031-354-560
ldison, TX	China - Shenzhen	Taiwan - Kaohsiung	Israel - Ra'anana
el: 972-818-7423	Tel: 86-755-8864-2200	Tel: 886-7-213-7830	Tel: 972-9-744-7705
ax: 972-818-2924	China - Suzhou	Taiwan - Taipei	Italy - Milan
etroit	Tel: 86-186-6233-1526	Tel: 886-2-2508-8600	Tel: 39-0331-742611
ovi, MI	China - Wuhan	Thailand - Bangkok	Fax: 39-0331-466781
el: 248-848-4000	Tel: 86-27-5980-5300	Tel: 66-2-694-1351	Italy - Padova
ouston, TX	China - Xian	Vietnam - Ho Chi Minh	Tel: 39-049-7625286
l: 281-894-5983	Tel: 86-29-8833-7252	Tel: 84-28-5448-2100	Netherlands - Drunen
dianapolis	China - Xiamen		Tel: 31-416-690399
oblesville, IN	Tel: 86-592-2388138		Fax: 31-416-690340
el: 317-773-8323	China - Zhuhai		Norway - Trondheim
ax: 317-773-5453	Tel: 86-756-3210040		Tel: 47-72884388
el: 317-536-2380			Poland - Warsaw
os Angeles			Tel: 48-22-3325737
ission Viejo, CA			Romania - Bucharest
el: 949-462-9523			Tel: 40-21-407-87-50
ax: 949-462-9608			Spain - Madrid
el: 951-273-7800			Tel: 34-91-708-08-90
aleigh, NC			Fax: 34-91-708-08-91
el: 919-844-7510			Sweden - Gothenberg
ew York, NY			Tel: 46-31-704-60-40
el: 631-435-6000			Sweden - Stockholm
an Jose, CA			Tel: 46-8-5090-4654
el: 408-735-9110			UK - Wokingham
el: 408-436-4270			Tel: 44-118-921-5800
anada - Toronto			Fax: 44-118-921-5820
			1 an. 44-110-921-9020
I: 905-695-1980			
ax: 905-695-2078			