Creating the First Application on PIC32CM JH01 Microcontrollers Using MPLAB Harmony v3 with MPLAB Code Configurator (MCC)



TB3353

Introduction

MPLAB[®] Harmony v3 is a software development framework consisting of compatible and interoperable modules that include peripheral libraries (PLIBs), drivers, system services, middleware, and third-party libraries. The MPLAB Code Configurator (MCC) is a GUI-based tool that provides an easy way to enable and configure various MPLAB Harmony modules. The MCC is a plug-in to the MPLAB X Integrated Development Environment (IDE).

The PIC32CM JH family of microcontrollers (MCUs) is the next generation of the popular SAM C21 family of Arm[®] Cortex[®]-M0+ based MCUs. The PIC32CM JH family of MCUs delivers a variety of popular features plus extended memory options up to 512 KB of Flash and 64 KB to help create designs that need functional safety, enhanced touch, or security. This family pin is compatible with the SAM C21 family in the 32-pin, 48-pin and 64-pin packages.

This document explains how to create a simple application on a Cortex-M0+ based PIC32CM JH01 Microcontroller using the MCC with MPLAB Harmony v3 modules. The objective of this application is to toggle an LED on a timeout basis and print the LED toggling rate. For this demonstration, the following MPLAB Harmony v3 modules are used and configured using the MCC:

- The PORT pin to toggle LED.
- Real-Time Clock (RTC) PLIB to periodically sample LED toggling rate.
- External Interrupt Controller (EIC) PLIB to change the toggling rate when there is a switch press event.
- Serial Communication Interfaces (SERCOM (SERCOM configured as USART)) and DMA PLIBs to print LED toggling rate on COM port terminal application (Serial Console) running on a computer.

1. Creating the First Application on the PIC32CM JH01 MCU

The following software and hardware tools are used for this demonstration:

- MPLAB X IDE v6.20
- MPLAB Code Configurator Plug-in v5.5.0
- MPLAB XC32 Compiler v4.35
- MPLAB Harmony v3 repository: csp v3.18.2
- PIC32CM JH01 Curiosity Pro Evaluation Kit

Note: The latest versions of these tools can also be used to develop the application.

To create an MPLAB Harmony v3-based project, follow these steps:

- 1. On the **Start** menu, launch MPLAB X IDE.
- 2. On the **File** menu, click **New Project** or click the *New Project* icon.
- 3. In the New Project window, in the left Navigation bar, under Steps click **Choose Project**.

Figure 1-1. Choose Project

X New Project		×
Steps	Choose Project	
1. Choose Project 2	Q, Filter:	
	Categories: Microchip Embedded Other Embedded	Projects: Application Project(s) Prebuit (Hex, Loadable Image) Project User Makefile Project Library Project
	Description:	
	Creates a new application project. It uses	s an IDE-generated makefile to build your project.
	< Bade	Next > Finish Cancel Help

- 4. In the Choose Project property page:
 - a. Categories: Selelct Micorchip Embedded.
 - b. Project: Select Application Project(s).
- 5. Click Next.
- 6. In the left Navigation bar, under Steps click **Select Device**.



🔀 New Project		×
Steps	Select Device	
 Choose Project Select Device Select Plugin Board Select Compiler Select Compiler Select Project Name and Folder (Optional) Add Project 	Family: All Families Device: PIC32CM25323H01100 Tool: No Tool	
MPLAB X IDE		
< Back Add And	nother Project Next > Finish Cancel	Help

Figure 1-2. Selecting the Device

- 7. In the Select Device property page, in the **Device** box, type or select the device PIC32CM5164JH01100.
- 8. Click Next.
- 9. In the left Navigation bar, under Steps click **Select Compiler**.
- 10. In the Select Compiler property page, expand XC32 list of options, and then select **XC32 (v4.35)**.



Figure 1-3. Compiler Selection

🕱 New Project	×
Steps 1. Choose Project 2. Select Device 3. Select Header 4. Select Plugin Board 5. Select Compiler 6. Select Project Name and Folder 7. (Optional) Add Project	Select Compiler Compiler Toolchains C32 C32 (v4.35) [C:\Program Files\Microchip\xc32\v4.35\bin] XC32 (v4.30) [C:\Program Files\Microchip\xc32\v4.21\bin] XC32 (v4.21) [C:\Program Files\Microchip\xc32\v4.21\bin]
MPLAB X IDE	
< Back Add An	other Project Next > Finish Cancel Help

11. Click Next.

- 12. In the left Navigation bar, click **Select Project Name and Folder**.
- 13. In the Select Project Name and Folder property page:
 - a. Project Name: Enter getting_started_pic32cm_jh01_cpro.
 - b. Project Location: Click the **Browse** button and choose C:\microchip\h3\Tech_Brief.



Figure 1-4. Project Creation

🔀 New Project		×
Steps	Select Project Name a	nd Folder
 Choose Project Select Device Select Header Select Plugin Board Select Compiler Select Project Name and Folder (Optional) Add Project 	Project Name: Project Location: Project Folder: Overwrite existing p Also delete sources Set as main project Set as main project Use project location	getting_started_pic32cm_jh01_cpro C: \microchip \h3\Technical_Brief Browse \Technical_Brief\getting_started_pic32cm_jh01_cpro.X project. s. n Finish
MPLAB X IDE	Encoding: ISO-	8859-1
< Back Add And	other Project	Next > Finish Cancel Help

- 14. Click Finish.
- 15. Before launching the MCC, the Configuration Database Setup window will be displayed, where the Device Family Pack (DFP) and Cortex[®] Microcontroller Software Interface Standard (CMSIS) path can be changed if required. For this demonstration, the default settings are used.
- 16. The MCC plug-in will open in a new window as shown in the following figure.

Figure 1-5. MPLAB Code Configurator Window

Edit View Navigate Source Refactor Production Debu			Q Search (Ctrl+I)
) 🥙 🎦 🔛 🍓 🔤 default 🛛 🦻	🗸 🌬 - 🎼 - 😼 - 🕨 - 👱 - 👱	- 🏹 🌆 - 🞯 - 🚥 🧶 🛒 - 🔜 PC:	0x0 How do I? Keyword(s)
Resource Management [MCC] × •	Kit Window 🗴 Start Page 🗴 🐏 main.c	x 🖭 initialization.c x Project Graph x	
MCC v5.3.7	🗙 🗔 📖 🖲 📖 📾 🛧 🏟 Plugins	· ~	View: Root 🗸
Project Res Gen Im E 😮 🚇			
► Libraries	NVMCTRL		
▼ System	Peripheral Library		Configuration
🛞 🔵 System	MEMORY 🔶		Options
		Project	options
	EVSYS Peripheral Library	Graph	
Project Resources	Perpheral Cibrary	Старн	
	Device Family Pack (DFP)		
	System		
Device Resources	CMSIS Pack		
Libraries	CHOID F JUK		
Harmony			

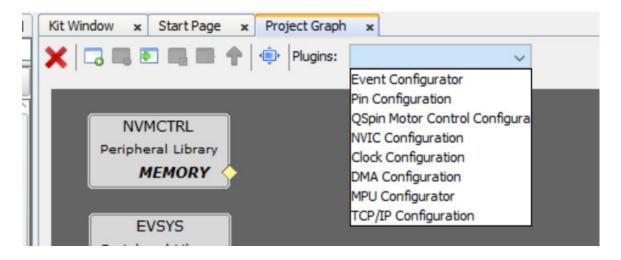
1.1 Adding and Configuring the MPLAB Harmony Components

To add and configure MPLAB Harmony components using the MCC, follow these steps:

1. In the MCC window, click **Project Graph**.



Figure 1-6. MPLAB Code Configurator



In the **Plugins** drop-down list, select **Clock Configuration**. The **Clock Easy View** window will be displayed, verify that the Main Clock is set to 48 MHz.
 Note: Make sure to make the following modification for the GCLK Generator 1.

S Clock Easy View - Editor ٥ Clock Easy View × Oscillators Controller xosc GCLK Generator 0 Main Clock Internal 🗹 OSC48M OSC48M -CPU 48,000,000 Hz FDPLL CFD 48,000,000 Hz OSCULP32K VDD = 3.6V to 5.5V 🗸 48 MHz ~ DIV1 ~ GCLKO ge Calibration OSC32K 48,000,000 Hz хозсак xosc . GCLK_IN[0] 1÷ eripheral Clock Selectio GCLK1 GCLKO CRYSTA GCLK1 ~ GCLK2 Fractional Digital Phase Locked Loop GCLK3 GCLK Generator 1 GCLK4 Peripherals XOSC FDPLL GCLK5 £63 OSC48M GCLK6 Peripheral Clock Configuration FDPLL GCLK7 0 🗘 0 🗘 DIV1 V OSCULP32K GCLK1 GCLK8 OSC32K Auto Calculate ~ хозсак EIC £63 1 🗘 32,768 Hz GCLK_IN[1] WDT 1,024 Hz 32 KHz Os OSCULP1K OSC32K 32 KH OSCULP32K OSC1K OSC1K GCLK Generator x (x=2-8) GCLK 2 RTC OSC32K 1 024 H

Figure 1-7. MPLAB Code Configurator - GCLK Generator One

- 3. Under **Device Resources**, expand the list of options *Harmony > Peripherals > SERCOM*.
- 4. Click **SERCOM4** and observe that the SERCOM4 Peripheral Library block is added in the Project Graph window.



Figure 1-8. MPLAB Code Configurator - Selection of Peripherals

Resource Management [MCC v5.5.0	[MCC]				× •	
Project Resources	Generate	Import	Export	•		
Device Resources						
 Peripherals 					^	
► AC						
ADC						
CAN						
► CCL						
► DAC						
DIVAS						
DSU						
FREQM						
PDEC						
► PM						
RAM						
RSTC						
▼ SERCOM						
🛞 🛨 🔿 s	SERCOM0					
🛞 🛨 🔿 s	SERCOM1					
🛞 🛨 🔿 s	SERCOM2					
🕜 🕂 🔿 s	SERCOM3					
8 🛨 🕓 s	SERCOM4					
⊕ + ○ ≤	SERCOM5					

Note: Users can also select other peripherals under Device Resources, Harmony > Peripherals.

5. In the Project Graph window, in the left Navigation bar, select **SERCOM4 Peripheral Library**. In the right **Configuration Options** property page, configure it as follows to print the data on the Serial Console at 115200 baud rate.



c pin_configurations.csv x Proconfiguration Options x of	•
NVMCTRL Peripheral Library MEMORY	

Figure 1-9. MPLAB Code Configurator - SERCOM4 Configuration

- 6. Under **Device Resources** expand the list of options *Harmony* > *Peripherals* > *EIC*. Click **EIC** and observe that the EIC Peripheral Library block is added in the Project Graph window.
- 7. In the right Configuration Options property page, select the **Enable Interrupt** check box and the **Enable EIC Channel3** check box for the switch press event.

Figure 1-10. MPLAB Code Configurator - EIC Configuration

Kit Window 🗴 Start Page 🗴 Project Graph 🗴	Configuration Options ×						
🗙 🗔 🛤 🖲 📾 🛖 🏟 Plugins:	- +						
	B-EIC						
NVMCTRL	EIC Clock Source Selection Clocked by ULP32K V						
Peripheral Library	Non Maskable Interrupt Control 🗌						
	Enable EIC Channel0						
	Enable EIC Channel1						
EVSYS	Enable EIC Channel2						
Peripheral Library Peripheral Library	🖻 <mark>Enable EIC Channel3 🛛 🗹</mark>						
	🖻 EIC Channel3 Configuration						
Device Family Pack (DFP)	Enable Interrupt						
	Enable Event Output						
System	-External Interrupt3 Detection Clock Edge detection is dock synchronously operated $$						
CMSIS Pack	External Interrupt3 Edge Selection \sim Rising edge detection \sim						
CMSIS Pack	Enable Debounce						
SERCOM4	Enable filter						
Peripheral Library	Enable EIC Channel4						
12C 🔶	Enable EIC Channel5						
SPI 🔷	-Enable EIC Channel6						
UART	-Enable EIC Channel7						
	-Enable EIC Channel8						

8. Under **Device Resources** expand the list of options *Harmony > Peripherals > RTC*. Click **RTC** and observe that the RTC Peripheral Library block is added in the Project Graph window to generate a compare interrupt every 500 milliseconds. Slect the **Compate 0 Interrupt Enable** check box.



Note: The Compare Value is set as 0x200. This compare value generates an RTC compare interrupt every 500 milliseconds. RTC clock = 1024 Hz RTC Prescaler = 1 Required Interrupt rate = 500 ms

Hence, Compare Value = (500/1000) x 1024 = 512 (i.e., 0x200)

Figure 1-11. MPLAB Code Configurator - RTC PLIB Configuration

Projects × Files	Kit Window x Start Page x Project Graph x	Configuration Options	×
pic32cm_jh01_cpro_getting_started	🗙 🗔 🛤 💽 🖏 📾 🔶 Plugins:		
Device actions Header Files		⊟-RTC	
Important Files	NVMCTRL	-Hardware Settings	
🗈 🛅 Linker Files	NVMCTRL Peripheral Library	-Generate Frequency Correction API	
⊕-/@ Source Files ⊕-/@ Libraries	MEMORY	-RTC Count Sync Enable	
⊕- 🙀 Loadables		E-RTC Operation Mode 32-bit Counter with Single 32-bit Compare 🗸	
		EIC B-RTC MODE 0 Configuration	
	Peripheral Library Periphera	Inal Library	
	Device Family Pack (DFP)	- Periodic Interval O Interrupt Enable	
		Periodic Interval 1 Interrupt Enable Periodic Interval 2 Interrupt Enable	
	System	RTC — Periodic Interval 2 Interrupt Enable	
		TMR Periodic Interval 4 Interrupt Enable	
	CMSIS Pack	- Periodic Interval 5 Interrupt Enable	
	SERCOM4	-Periodic Interval 6 Interrupt Enable 🗌	
	Peripheral Library	- Periodic Interval 7 Interrupt Enable	
	12C 🔶	– Compare O Interrupt Enable 🛛	
	SPI 🔶	Overflow Interrupt Enable	
	UART 🔶	RTC Prescaler DIV1 V	
		Compare Value 0x 200 🜩	
		Clear on compare Match 🗹	

9. From the Plugins drop-down list, select **Add Channel** and then select **DMA Configuration**. Configure the DMA Channel 0 to transmit application buffer to the USART TX register. The DMA transfers one byte from the user buffer to the USART transmit buffer on each trigger.

Figure 1-12	. MPLAB Code	Configurator -	DMA Configuration
-------------	--------------	----------------	-------------------

Configuration				-	٥
Active	Channels List	Use Linked List Mode	ę		
Channel Number	Trigger]	DMA Channel 0 Settings		
DMAC Channel 0	SERCOM4_Transmit 🗸	Enable Interrupt			
		Trigger Action	One Beat Transfer per DMA Request		~
Add Channel	Remove Selected Channel	Source Address Mode	Increment Address After Every Transfer		~
		Destination Address Mode	Fixed Address Mode		~
		Beat Size	8-bit bus transfer		~

- 10. From the Plugins drop-down list select **Pin Configuration** and then click **Pin Settings**.
- 11. In the **Order** box, type or select Ports. Build configurations according to the application as indicated below. Change the Custom Name of the pin IDs, PC05 and PB19, as shown in the following figure.



Kit Window 🗙	Start Page ×	Packs x Project Gra	ph 🗴 Pin Diagram	× Pin Tabl	e 🗴 Pin Settings	×			
Order: Ports	~	Table View 🗹 Easy	View						
Pin Number	Pin ID	Custom Name	Function	Mode	Direction	Latch	Pull Up	Pull Down	Drive Strength
32	PB10		SERCOM4_PAD2 ~	Digital	High Impedance \lor	n/a			NORMAL V
33	PB11		SERCOM4_PAD3 ~	Digital	High Impedance \lor	n/a			NORMAL V
67	PB19	SWO	EIC_EXTINT3 ~	Digital	High Impedance $$	n/a			STRONG 🗸
21	PC05	LED0	GPIO 🗸	Digital	Out 🗸	Low			NORMAL 🗸

Figure 1-13. Pin Settings Window - SERCOM, EIC and LED Pin Configuration

1.2 Generating the Code

After configuring the peripherals, as shown in the following figure click **Generate** under **Resource Management** [MCC].

Figure 1-14. Code Generation

WPLAB X IDE v6.20 - pic32cm_jh01_cpro_getting_started : pic32cm_jh01_cpro

File	Edit View Navigate Sour	ce Refactor	Production	Debug	Team	Tools	1
P) 🛅 🔛 🖣 🦻	C pic32	cm_jh01_cpro	~	SRC -		÷
or D	Resource Management [MCC] MCC v5.5.0						
Navigator	Project Resources	Generate	Import	Export		0	
ard 📎 I	 Libraries Harmony 						

Notes:

- The generated code will add files and folders to the 32-bit MCC Harmony project. In the generated code, notice the Peripheral Library files generated for the Real-Time Clock (RTC), External Interrupt Controller (EIC), PORT peripherals, SERCOM4 (as Universal Synchronous Asynchronous Receiver Transmitter (USART)), and the Direct Memory Access (DMA) peripherals. The MCC also generates the main.c file.
- 2. The MCC provides an option to change the generated file name. By default, the file name main.c is generated if a name is not assigned.

1.3 Adding Application Logic to the Project

 To develop and run the Application, use the following steps: Open the main.c file of the project and add the following application logic. Add the following code in the main() function:

```
uint8_t uartLocalTxBuffer[100] = {0};
    /* Initialize all modules */
    DMAC_ChannelCallbackRegister(DMAC_CHANNEL_0, uartDmaChannelHandler_Tx, 0);
    RTC_Timer32CallbackRegister(rtcEventHandler, 0);
    EIC_CallbackRegister(EIC_PIN_3, SW0_userHandler, 0);
    sprintf((char*)uartTxBuffer, "*********** Printing Toggling LED rate
    **********\r\n");
    DMAC_ChannelTransfer(DMAC_CHANNEL_0, uartTxBuffer, \
        (const void *)&(SERCOM4_REGS->USART_INT.SERCOM_DATA), \
        strlen((const char*)uartTxBuffer));
```



```
/* Start the timer */
RTC_Timer32Start();
```

Figure 1-15. Adding Application Logic to Register Callback Event Handlers



2. Implement the registered callback event handlers for the peripherals by adding the following code before the main() function.

```
static void SW0_userHandler(uintptr_t context)
{
    changeTempSamplingRate = true;
}
static void rtcEventHandler (RTC_TIMER32_INT_MASK intCause, uintptr_t context)
{
    if (intCause & RTC_MODE0_INTENSET_CMP0_Msk)
    {
        isRTCExpired = true;
    }
}
static void uartDmaChannelHandler_Tx(DMAC_TRANSFER_EVENT event, uintptr_t contextHandle)
{
    if (event == DMAC_TRANSFER_EVENT_COMPLETE)
    {
        isUARTTxComplete = true;
    }
}
```

3. According to the status of the isRTCExpired and isUARTTxComplete flags (These flags are handled by rtcEventHandler and uartDmaChannelHandler_Tx event handlers when RTC Timer expires and when UART completes the transfer of data.), the LEDO is toggled at a default rate of 500 ms. To change the toggling rate, if the user presses the SWO switch, the toggling rate changes to 1s, 2s, and 4s and back to 500 ms with subsequent switch press events. SWO_userHandler will be responsible for changing the toggling rate when the user presses the SWO switch on the board.



Inside the while loop, delete $SYS_Tasks()$ and add the following code to toggle the LED at a default rate of 500 ms.

4. Add the following code immediately after adding the above code to change the toggling rate when there is a switch press event.

```
if(changeTempSamplingRate == true)
            {
                changeTempSamplingRate = false;
                if (tempSampleRate == TEMP SAMPLING RATE 500MS)
                    tempSampleRate = TEMP SAMPLING RATE 1S;
                    RTC Timer32CompareSet(PERIOD_1S);
                else if(tempSampleRate == TEMP SAMPLING RATE 1S)
                    tempSampleRate = TEMP SAMPLING RATE 2S;
                    RTC_Timer32CompareSet(PERIOD_2S);
                else if (tempSampleRate == TEMP SAMPLING RATE 2S)
                    tempSampleRate = TEMP SAMPLING_RATE_4S;
                    RTC Timer32CompareSet(PERIOD_4S);
                else if(tempSampleRate == TEMP SAMPLING RATE 4S)
                   tempSampleRate = TEMP SAMPLING RATE 500MS;
                   RTC Timer32CompareSet(PERIOD 500MS);
                else
                {
                    ;
                RTC Timer32CounterSet(0);
                sprintf((char*)uartLocalTxBuffer, "LED Toggling rate is changed to
%s\r\n", &timeouts[(uint8_t)tempSampleRate][0]);
                DMAC ChannelTransfer(DMAC CHANNEL 0, uartLocalTxBuffer, \
                     \overline{(\text{const void }^*)} (SERCOM4 REGS->USART INT.SERCOM DATA), \
                    strlen((const char*)uartLocalTxBuffer));
            }
```

5. Add the following code to include the necessary header files, and define the macros for different RTC compare values.

#define TX_BUFFER_SIZE 100

This code declares the various flags whose status is monitored and changed by the event handlers in the application. It has various declarations and definitions of arrays used to print the LED toggling rate on the console.

static volatile bool isRTCExpired = false; static volatile bool changeTempSamplingRate = false; static volatile bool isUARTTxComplete = true;



```
static volatile bool isUARTRxComplete = false;
static uint8_t uartTxBuffer[TX_BUFFER_SIZE] = {0};
typedef enum
{
    TEMP_SAMPLING_RATE_500MS = 0,
    TEMP_SAMPLING_RATE_1S = 1,
    TEMP_SAMPLING_RATE_2S = 2,
    TEMP_SAMPLING_RATE_4S = 3,
} TEMP_SAMPLING_RATE_4S = 3,
} TEMP_SAMPLING_RATE;
static TEMP_SAMPLING_RATE tempSampleRate = TEMP_SAMPLING_RATE_500MS;
static const char timeouts[4][20] = {"500 milliSeconds", "1 Second", "2 Seconds", "4
```

1.4 Building and Programming the Application

- 1. The PIC32CM JH01 Curiosity Pro Evaluation Kit supports debugging using an Embedded Debugger (EDBG). Connect the Type-A male to micro-B USB cable to the micro-B debug USB port on the PIC32CM JH01 Curiosity Pro Evaluation Kit to power and debug the PIC32CM JH01 Curiosity Pro Evaluation Kit.
- 2. Ensure that the compiler optimization is set to 1, to check that follow these steps:
 - a. Right-click on the project *getting_started_pic32cm_jh01_cpro*, a shortcut menu appears. Click **Properties.**

e Edit View Navigate Source Refactor	r Production Debug Team Tools V	/indow Help Q Search (Ctrl+I)
🤊 🥙 🎦 🎦 😼 🔤	ault 🗸 🕅 🗸 🖓 🗸	🌇 + 🜔 + 🖳 + 🏪 + 🎧 🖏 + PC: 0x0 How do 12 (Zeyword(s) 👘 🌇 🦉 🔟 + 🔜 + 🛒 + 🕮 🚥
rojects × Files		Page x 🛒 MPLAB X Store x 🐻 Output x 🥪 Variables x 🗐 Call Stack x 🖬 Breakpoints x 🖷 main.c x 🖷 initialization.c 🚺 🛡
getting_started_pic32cm_jh01	Build	_pic32ck_sg01_cult × EDBG × EDBG-getting_started_pic32cm_jh01_cpro × getting_started_pic32cm_jh01_cpro (Build, Load,) ×
Device actions Header Files	Clean and Build	
Important Files	Clean	
🕀 🛅 Linker Files	Batch Build	
Plugins	Set Configuration	
Gource Files Joraries	ser comgaration	
Eloranes	New	
	Add	lons:
	Unset as Main Project	
	Find	val
	Locate Headers	164JH01100 found.
	-	0x1
	Run	
	Debug	lges for operation
	Step into	
	Make and Program Device	rea(s) will be programmed:
	Projects	rea(s) will be programmed: address = 0x0, end address = 0x1dff
avigator × ed_pic32cm_jh01_c	Close	
PERIOD_1S	Open Required Projects	, y ranges on this device, only the areas of memory that have been loaded with code (via the build process or loadin
PERIOD_2S	Code Assistance	read by default. If you wish to read custom ranges, please go to the Memories to Program property page and specify
PERIOD_4S		o read.
PERIOD_500MS TEMP_SAMPLING_RATE	Analysis	
TX_BUFFER_SIZE	Show Code Coverage Summary	
whangeTempSamplingRate	Versioning	
isRTCExpired	History	
	Properties	

Figure 1-16. Project Properties

- b. In the **Project Properties** window, for Option categores select **Optimization** and for the Optimation level from the item list select **1**.
- c. Click **OK** to close the **Project Properties** window.



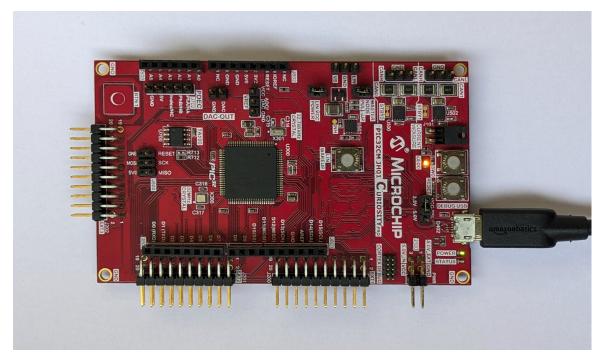
X

Figure 1-17. Optimization level

🔯 Project Properties - getting_started_pic32cm_jh01_cpro

Categories: Categories: Categories: File Inclusion/Exclusion File Inclusion/Exclusion Conf: [default] Conf: [default] Categories: Conf: [default] Conf: [default] Categories: Conf: [default] Categories: Conf: [default] Categories: Conf: [default] Conf: [default]	Options for xc32-gcc (v4.35) Option categories: Optimization optimization-level Unroll loops Omit frame pointer Pre-optimization instruction scheduling Post-optimization instruction scheduling	Reset				
 	Use common tentative definitions Disable Additional options: Option Description Generated Command Line					
Manage Configurations Manage Network Tools	 O - Do not optimize. The compiler's goal is to reduce the cost of compilation and to make debugging produce the expected results. 1 - Optimize. Optimizing compilation takes somewhat longer, and a lot more host memory for a large function. The compiler tries to reduce code size and execution time. 2 - Optimize even more. The compiler performs nearly all supported optimizations that do not involve a space-speed trade-off. 2 - Optimize uset more function speed (superset of O2). 					
Manage Network Tools		K Cancel Apply Unlock Help				

Figure 1-18. Hardware Setup



3. Set getting_started_pic32cm_jh01_cpro as the main project, and from Project Properties,
 select the latest compiler version (v4.35). Clean and build your project by clicking the highlighted
 icon.





4. Program the application by clicking the highlighted icon.

Figure 1-20. Program the Device



1.5 Observing the Output on the Board and Serial Terminal

- 1. Press the **Start** button to open Tera Term or PuTTY terminal application on the computer from the **Start** menu.
- 2. Select the required Serial Port and then click **OK**.

Figure 1-21. Selection of Serial COM Port

Tera Term: New connection	nc		×
O TCP/IP	Host: myhost.exa	mple.com	7
	✓ History		
	Service: \bigcirc Telnet	TCP port#: 22	
	⊚ S\$H	SSH version: SSH2 ~~	
	○ Other	IP version: AUTO	~
● Serial	Port: COM27: ED	BG Virtual COM Port (COM2	~
	0K Cancel	Help	

3. In the **Tera Term Serial port setup and connection** dialog box, in the **Speed** box type or select the baud rate to **115200**.



Figure 1-22. Setting the Baud Rate

 \times Tera Term: Serial port setup and connection COM27 Port: New setting 115200 Speed: 110 Data: 300 Cancel 600 Parity: 1200 2400 Help Stop bits: 4800 9600 Flow control: 14400 19200 Transmit 38400 57600 0 b msec/line 115200 230400 460800 Device Friendly Nam 921600 COM Port (COM27) Device Instance ID: USDIVIU_USED&PID_2111&MI_01\6&189E889; Device Manufacturer: Microchip Technology, Inc. Provider Name: Microchip Technology, Inc. Driver Date: 10-15-2020 Driver Version: 7.5.0.0 < >

- 4. An LED (LED0) on the PIC32CM JH01 Curiosity Pro Development Evaluation Kit toggles on a timeout basis and the default periodicity of the timeout is 500 milliseconds.
- 5. The LED toggling rate is displayed on the Serial Terminal.
- 6. Press the switch SW0 on the PIC32CM JH01 Curiosity Pro Evaluation Kit to change the default periodicity of the timeout to one second.
- 7. Every subsequent pressing of the switch SW0 on the PIC32CM JH01 Curiosity Pro Evaluation Kit changes the periodicity of the timeout to 2 seconds, 4 seconds, 500 milliseconds, and back to 1 second in cyclic order.



Figure 1-23. LED Toggling Rate on a Serial Terminal

💆 COM27 - Tera Term VT	—	×
File Edit Setup Control Window Help		
***********************************		*

While the LED toggling rate on the Serial Terminal changes with every subsequent switch press, observe the same change in the toggling rate of the LED0 on the evaluation kit.



TB3353 Resources

2. Resources

- PIC32CM JH01 Curiosity Pro Evaluation Kit
- For additional information on MPLAB Harmony v3, refer to the Microchip web site: https://www.microchip.com/mplab/mplab-harmony and

microchipdeveloper.com/harmony3:start

- For more information on various applications, refer to: github.com/Microchip-MPLAB-Harmony/reference_apps
- For the example application, refer to "Getting Started Application with PIC32CM JH01 Curiosity Pro Development Board" under the "Software" heading. www.microchip.com/en-us/development-tool/ev81x90a
- For additional info about 32-bit Microcontroller Collaterals and Solutions, refer to: ww1.microchip.com/downloads/aemDocuments/documents/MCU32/ProductDocuments/ ReferenceManuals/32-bit-Microcontroller-Collateral-and-Solutions-Reference-Guide-DS70005534.pdf



Microchip Information

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, 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, ClockWorks, The Embedded Control Solutions Company, EtherSynch, Flashtec, Hyper Speed Control, HyperLight Load, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet-Wire, SmartFusion, SyncWorld, TimeCesium, TimeHub, TimePictra, TimeProvider, 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, Clockstudio, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, Espresso T1S, EtherGREEN, EyeOpen, GridTime, IdealBridge, IGaT, In-Circuit Serial Programming, ICSP, INICnet, Intelligent Paralleling, IntelliMOS, Inter-Chip Connectivity, JitterBlocker, Knob-on-Display, MarginLink, maxCrypto, maxView, memBrain, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, mSiC, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, Power MOS IV, Power MOS 7, 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, Trusted Time, TSHARC, Turing, 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, and Symmcom 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.

© 2024, Microchip Technology Incorporated and its subsidiaries. All Rights Reserved.

ISBN: 978-1-6683-4212-1

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
2355 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
Fel: 480-792-7200	Tel: 86-10-8569-7000	Tel: 91-11-4160-8631	Denmark - Copenhagen
ax: 480-792-7277	China - Chengdu	India - Pune	Tel: 45-4485-5910
Technical Support:	Tel: 86-28-8665-5511	Tel: 91-20-4121-0141	Fax: 45-4485-2829
www.microchip.com/support	China - Chongqing	Japan - Osaka	Finland - Espoo
Web Address:	Tel: 86-23-8980-9588	Tel: 81-6-6152-7160	Tel: 358-9-4520-820
www.microchip.com	China - Dongguan	Japan - Tokyo	France - Paris
Atlanta	Tel: 86-769-8702-9880	Tel: 81-3-6880- 3770	Tel: 33-1-69-53-63-20
Duluth, GA	China - Guangzhou	Korea - Daegu	Fax: 33-1-69-30-90-79
Гel: 678-957-9614	Tel: 86-20-8755-8029	Tel: 82-53-744-4301	Germany - Garching
ax: 678-957-1455	China - Hangzhou	Korea - Seoul	Tel: 49-8931-9700
Austin, TX	Tel: 86-571-8792-8115	Tel: 82-2-554-7200	Germany - Haan
Tel: 512-257-3370	China - Hong Kong SAR	Malaysia - Kuala Lumpur	Tel: 49-2129-3766400
Boston	Tel: 852-2943-5100	Tel: 60-3-7651-7906	Germany - Heilbronn
Westborough, MA	China - Nanjing	Malaysia - Penang	Tel: 49-7131-72400
Tel: 774-760-0087	Tel: 86-25-8473-2460	Tel: 60-4-227-8870	Germany - Karlsruhe
- Fax: 774-760-0088	China - Qingdao	Philippines - Manila	Tel: 49-721-625370
Chicago	Tel: 86-532-8502-7355	Tel: 63-2-634-9065	Germany - Munich
tasca, IL	China - Shanghai	Singapore	Tel: 49-89-627-144-0
Tel: 630-285-0071	Tel: 86-21-3326-8000	Tel: 65-6334-8870	Fax: 49-89-627-144-44
Fax: 630-285-0075	China - Shenyang		Germany - Rosenheim
	Tel: 86-24-2334-2829		Tel: 49-8031-354-560
Dallas	China - Shenzhen	Taiwan - Hsin Chu	Israel - Ra'anana
Addison, TX	Tel: 86-755-8864-2200	Tel: 886-3-577-8366	Tel: 972-9-744-7705
Tel: 972-818-7423	China - Suzhou	Taiwan - Kaohsiung	Italy - Milan
Fax: 972-818-2924	Tel: 86-186-6233-1526	Tel: 886-7-213-7830	Tel: 39-0331-742611
Detroit	China - Wuhan	Taiwan - Taipei	Fax: 39-0331-466781
Novi, MI	Tel: 86-27-5980-5300	Tel: 886-2-2508-8600	Italy - Padova
Fel: 248-848-4000	China - Xian	Thailand - Bangkok	Tel: 39-049-7625286
Houston, TX	Tel: 86-29-8833-7252	Tel: 66-2-694-1351	Netherlands - Drunen
Tel: 281-894-5983	China - Xiamen	Vietnam - Ho Chi Minh	Tel: 31-416-690399
Indianapolis	Tel: 86-592-2388138	Tel: 84-28-5448-2100	Fax: 31-416-690340
Noblesville, IN	China - Zhuhai		Norway - Trondheim
Геl: 317-773-8323	Tel: 86-756-3210040		Tel: 47-72884388
Fax: 317-773-5453			Poland - Warsaw
rel: 317-536-2380			Tel: 48-22-3325737
.os Angeles			Romania - Bucharest
Mission Viejo, CA			Tel: 40-21-407-87-50
Fel: 949-462-9523			Spain - Madrid
Fax: 949-462-9608			Tel: 34-91-708-08-90
Tel: 951-273-7800			Fax: 34-91-708-08-91
Raleigh, NC			Sweden - Gothenberg
Tel: 919-844-7510			Tel: 46-31-704-60-40
New York, NY			Sweden - Stockholm
Tel: 631-435-6000			Tel: 46-8-5090-4654

Worldwide Sales and Service

continued						
AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE			
San Jose, CA			UK - Wokingham			
Tel: 408-735-9110			Tel: 44-118-921-5800			
Tel: 408-436-4270			Fax: 44-118-921-5820			
Canada - Toronto						
Tel: 905-695-1980						
Fax: 905-695-2078						