

## MPLAB<sup>®</sup> Harmony 3 之基础篇(09)

一 如何使用 Harmony I2C 驱动开发应用程序

Microchip Technology Inc. MCU32 产品部

一、 简介

MPLAB<sup>®</sup> Harmony 是 MPLAB 开发工具生态系统的重要组成部分, MPLAB<sup>®</sup> Harmony 3 适用于是 Harmony 系列开发工具的换代升级,增加了对 SAM<sup>®</sup>系列微处理器的 支持,是 Microchip<sup>®</sup>32 位 SAM<sup>®</sup>和 PIC<sup>®</sup>微控制器的嵌入式系统的重要软件方案。本文主要介绍如何利用 MPLAB X IDE 创建一个工程,利用 MPLAB Harmony 3 Configurator (MHC) 添加 IC 外设驱动到工程文件,并利用 MHC 的配置工具 (CLOCK, PIN 等) 完成 IC 外设的配置。通过调用 I2C PLIB API 实现对 AT24 系列 EEPROM 的读写操作。

## 二、 硬件工具和软件平台

硬件: SAM E70 Xplained Board

http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-44050-Cortex-M7-Micr ocontroller-SAM-E70-XPLD-Xplained User-guide.pdf





软件(开发工具和环境的安装和使用,见 "MPLAB<sup>®</sup> Harmony 3 之基础篇(01) -- Harmony 3 开发环境搭建" "MPLAB<sup>®</sup> Harmony 3 之基础篇(02) -- 了解 MHC"

MPLAB <sup>®</sup> X IDE:	v5.10 或者更新
XC32:	v2.10 或者更新
Harmony 3:	v3.10 或者更新

#### 详细步骤 三、

接下来我们就可以用 MPLAB X IDE 和 MHC 一步步地创建和配置 I2C 外设驱动的程 序。

注:以下MHC 配置里没有特别标注出来的地方,说明使用的是默认选项。

(一)在 MPLAB X 里新建一个 Harmony 3 项目 在 MPLAB X IDE 里点击 File > New Project:



😢 New Project			
Steps	Choose Project		
1. Choose Project	Q Filter:		
	Categories: Microchip Embedded Other Embedded Categories: Other Embedded Samples	Projects:	
	Description:		
	MPLAB® Harmony Project Wizard		
	< Back	Next > Finish Cancel Help	

# 选择"32-bit MPLAB Harmony Porject",然后点击"Next"按钮。

😰 New Project	×	
Steps	Manage Framework	_
<ol> <li>Choose Project</li> <li>Framework Selection</li> <li>Project Settings</li> <li>Configuration Settings</li> </ol>	Use the Framework Downloader tool to download or configure a local framework. Launch Framework Downloader Harmony3的安装路径 Framework Path: C:\microchip\harmony\H3 C:\microchip\harmony\H3 C:\microchip\harmony\H3 C:\microchip\harmony\H3 C:\microchip\harmony\H3	
	< Back Next > Finish Cancel Help	



			×
Steps	Name and Lo	cation	
1. Choose Project 2. Framework Selection 3. Project Settings	Location:	C:\Users\HarmonyProjects	1
<ol> <li>Configuration Settings</li> </ol>	Folder:	E70_I2d	
	Name:	E70_I2C	
	Path:	C:\Users\HarmonyProjects\firmware\E70_12C.X	

### 选择"Harmony Framework"路径,然后点击"Next"按钮。

填写项目名称,本示例使用"E70\_I2C",然后点击"Next"按钮。



😵 New Project	×
Steps	Configuration Settings
<ol> <li>Choose Project</li> <li>Framework Selection</li> <li>Project Settings</li> <li>Configuration</li> </ol>	Name: default 配置名称,可用默认设置
Settings	Device Family: AISAM V Iarget Device: AISAME70Q21B
	Device Filter: 选择你的开发板MCU芯片的类型
	Show Visual Help
1	
	< Back     Wext >     Finish     Cancel     Help

选择芯片类型"ATSAME70Q21B",	最后点击"Finish	n"按钮启动 MHC 配置界面。
-----------------------	-------------	------------------

### (二)启动 MHC

第一次创建项目时,MHC配置界面会自动启动。或者手动在MPLABX里点击Tools> Embeded > MPLAB Harmony 3 Configurator 启动 MHC:

samd21_	debug_uart : default				
e Source	Refactor Production De	ebug Team To	ools Window Help		
5 6	default 👻	r • 78	Embedded	٠.	MPLAB® Harmony 3 Framework Downloader
sses	Services		Licenses Apply Diff Patch		MPLAB® Harmony 3 Configurator
			Templates DTDs and XML Schemas		
			Plugins Plugins Download		
			Options		



启动完成后的, 主界面如下图:

MPLAB X IDE v5.10.04 - E70_USART : default	Mail Investigation - State	the set of the set of the set of the	
File Edit MHC View Navigate Source Refactor Production Debug Team Tools Window Help			
👚 🚰 🚰 🦉 🧶 🖗 (default 💦 🖓 - 🔖 - 🔽 - 🏠 🖏 - 1 🖹 🗇 🔊 silvered (s)			
Proj % Files Classes Services Active C	🔁 Output 🛎 Xplained Window 🕮 Project Graph*	_	
E ETO_USANT	▓╔┇┇┇┇┇╋┝╠	View:	
Linker Files	Device Family Pack (DFP) System CMSIS Pack		
Libraries Losdables			
空的项目工程			
		•	
Available Components*	最小的工程组件		
Board Support Packages (BSPs)			
Graphics			
Harmony			
🖶 Input			
Libraries			
Peripherals	Variables 🕺 Console		
Third Party Libraries	Ulanary Detabase [Tafa], CCD, such assessed, Deviabara] UADIA (TD, CA10)		
• Tools	<pre><harmony database="">[Info]: CSP: create component: Peripheral USARTO (ID = 6089) </harmony></pre>		
	<pre><harmony database="">[Info]: CSP: create component: Peripheral USARII (ID = 6089) <harmony database="">[Info]: CSP: create component: Peripheral USARI2 (ID = 6089)</harmony></harmony></pre>		
可选的H3组件	<pre><harmony database="">[Info]: CSP: Peripheral [USBHS id=11292] is not supported in MCC <harmony database="">[Info]: CSP: Peripheral [UTMI id=11300] is not supported in MCC</harmony></harmony></pre>	The company manager of the	
	<pre><harmony database="">[Info]: CSP: System Peripheral [WDT id=6080] <harmony database="">[Info]: CSP: System Peripheral [XDMAC id=11161]</harmony></harmony></pre>		
	<harmony database="">[Info]: CSP: Peripheral [FUSES id=1] is not supported in MCC</harmony>		

## (三)添加 Harmony Core, I2C0 外设驱动, AT24 EEPROM 驱动和 E70 Xplained Board (BSP)支持

从 Harmony 中选择 Core,并选择合理的 Core 配置。

Available Components	-	7
+		
🖶 Audio		
🖶 Bluetooth		
Board Support Packages (BSPs)		
🖶 Graphics		
Harmony		
Core		
Drivers		
Harmony Networking		
System Services		
Libraries		
HSMCI		
I2SC		
⊕ PWM		
🖶 QSPI		
RSTC		
I RTC		
I RTT		
■ SMC		
SPI SPI		



Configuration Options	
■ €	
⊡Core	
🕂 Generate Harmony Application File	es 🗸
Application Configuration	
	1 🛋
-Application 0 Configuration	
Application Name app	
**** Application name m	ust be valid C-Language identifier and sho
Enable System Interrupt	
Enable System Ports	
- Enable System Cache	
Enable System DMA	
Enable OSAL	

从 Peripherals 选择 TWIHSO 外设驱动,双击,添加到 Project Graph,USART 采用 默认的设置。

Available Components	- 19
+	
⊕- Audio	
Bluetooth	
Board Support Packages (BSPs)	
Graphics	
🖶 Harmony	
🖶 Input	
🖶 Libraries	
🖻 Peripherals	
■ EFC	
⊕ GFX2D	
HSMCI	
# I2SC	
B SMC	
BPI	
I SSC	
⊕ TC	
TWIHSO	
- TWIHS1	
TWIHS2	



从 Harmony 中选择 AT24 的驱动程序。



选好 I2C 和 AT24 的驱动以后,在 Project Graph 中用直线连接连接两者的 I2C 接口。

📋 Project Graph		- 🗗 🗖
	¢ ا	View: Root ~
Device Family Pack (D	FP) System CMSIS Pack	
	TWIHS0 Peripheral Library I2C	AT24 Driver Core Service I2C MEMORY



从 Board Support Packages 选择 E70 Xplained BSP 的支持,双击添加到 Project Graph.

E70_USART - Dashboard	Available Components*
+	
Board Suppor	t Packages (BSPs)
Default Boa	rd (BSP)
SAM E70 Xp	lained BSP
SAM E70 Xp	lained Ultra BSP
Graphics	
Harmony	
🗉 Input	
Libraries	
Peripherals	
ACC	
I AFEC	
DACC	
🖶 EFC	

📋 Project Graph		- 20
		View: Root ~
Device Family Pack (DFP)	System CMSIS Pack	SAM E70 Xplained BSP
Core Harmony Core Service • RTOS Core Service •	TWIHS0 Peripheral Library I2C	AT24 Driver Core Service I2C MEMORY



## (四)选择 MHC>Tools>Clock Configuration 菜单,启动时钟配置

E70 的时钟配置界面,确认时钟源和 I2C 的时钟是我们需要的配置。

MPLAB X IDE v5.10.04 - E70\_USART : default

-	_	
File Edit MH	C view Navigate	Source Refactor Production Debug Team Tools Window Help
🔁 🖆	Save State	default 🕞 🚏 * 🎇 * 🕨 * 🔽 * 🖏 🖏 * 🕨 🖱 🛙
Proj 8	Load State	Services Active C   Services Active
	Import	
÷6	Export	Device Family Pack (DFP) System
	Preferences	
	Generate Code	
	Tools	Clock Configuration
F70 USAF	Utilities 🛛	DMA Configuration
+	Exit	NVIC Configuration
🖻 Librari	ies	Pin Configuration
Periph	erals	



这里我们使用默认配置即可:







MPLAB X IDE v5.10.04 - E70_USART : default							
File Edit MHC View Navigate Source Refactor Production Debug Team Tools Window Help							
P 2	Save State Save State As	default	•	°° • 👸 • 🕨 • 💺 • 靠 • 🎧 🚯 • 🕨 🖻 🎐			
Proj 8	Load State	Services Active C 🖃 🖾 Output 📽 Xplained Window 🕸 Project Graph*					
E E	Import	-		Order: Fins 💌 Table View			
	Export			Pin Number	Pin ID	Custom Name	
	Preferences			1	PD0		
	Generate Code	-		2	PD31		
· 📶	Concrate code			3	VDDOUT		
	Tools	Clock Configuration		4	PE0		
	Utilities 0	DMA Configuration		5	VDDIN		
E70_USAF		MPU Configuration		6	PE1		
+	Exit	NVIC Configuration		7	PE2		
🖭 Librar	ries	Pin Configuration		8	VREFN		
Periph	herals			9	VREFP		
				10	PE3		
				11	PC0		
					0000		

### (五)使用 MHC>Tools>Pin Configuration, 配置 I2C 和 LED0 的引脚分配

## I2C 的管脚分配可以参考以下 E70 Xplained Board User guide 文档 (Atmel-44050-Cortex-M7-Microcontroller-SAM-E70-XPLD-Xplained\_User-guide.pdf)

11 [I2C_SDA]	PA3	TWD0	Camera Connector, EXT2 Header, J500 Header (Arduino Shield), AT24MAC402, Embedded Debugger
12 [I2C_SCL]	PA4	ТWCК0	Camera Connector, EXT2 Header, J500 Header (Arduino Shield), AT24MAC402, Embedded Debugger

#### Table 4-25 Virtual COM Port Connections

SAM E70 Pin	Function	Shared Functionality
PB4	TXD1 (SAM E70 UART TX Line)	EXT2 Header, J507 Header, Embedded Debugger
PA21	RXD1 (SAM E70 UART RX Line)	EXT2 Header, J507 Header, Embedded Debugger

## E70 XPlained board 上分配 PA3 和 PA4 到 TWD0 和 TWCK0。

Pin Settings					
Order: Ports	∼ Table View				
Pin Number	Pin ID	Custom Name	Function	Direction	Latch
102	PAO		Available $\vee$	In	n/a
99	PA1		Available $\vee$	In	n/a
93	PA2		Available $\vee$	In	n/a
91	PA3	TWIHS0_TWD0	TWIHS0_TWD0 V	n/a	n/a
77	PA4	TWIHS0_TWCK0	TWIHS0_TWCK0 V	n/a	n/a



#### E70 XPlained board 上分配 PA23 到 LED0。

46 PA23 LED0 LED\_AL Out High

## (六)"保存"项目配置和"代码生成"

在任务栏点击"保存"和"代码生成"按钮



展开左侧的工程项目管理目录树,可以看到相关的 I2C 头文件和源代码已经生成了。





### (七)设置项目的调试接口和编译器

选择项目 E70\_I2C, 右键选择 Properties



Proj··· % Files	Classes	Services	Active C····	
ETO_USAPT	New			+
C C C C C C C C C C C C C C C C C C C	Add Existing Add Existing New Logical Locate Head Add Item to Export Hex Build Clean and Bi Clean Batch Build	Item Items from Folder ers Important I uild	ı Folders Files	
6	Set Configur Run Debug Step into Make and Pi	ation rogram Dev	ice	•
- Ge Libr	Unset as Ma Open Requir Close Rename	in Project ed Projects		Þ
E70_USART - Da	Move Copy Delete			Delete
	Code Assista Find Versioning History	ince		+
<b>R</b>	Properties			



\*\*\*\*\*



// Section: Type Definitions #define BUFFER SIZE 256 /\* Application states Summary: Application states enumeration Description: This enumeration defines the valid application states. These states determine the behavior of the application at various times. \*/ typedef enum { /\* Application's state machine's initial state. \*/ APP STATE INIT=0, APP\_STATE\_WRITE, APP\_STATE\_WAIT\_WRITE\_COMPLETE, APP\_STATE\_READ, APP\_STATE\_WAIT\_READ\_COMPLETE, APP STATE VERIFY, APP\_STATE\_ERROR, APP\_STATE\_IDLE, /\* TODO: Define states used by the application state machine. \*/ } APP\_STATES;

/\* Application Data

Summary: Holds application data

Description:

This structure holds the application's data.



Remarks:

Application strings and buffers are be defined outside this structure.

\*/

typedef struct

```
{
```

/\* The application's current state \*/
APP\_STATES state;

DRV\_HANDLE drvHandle;

uint8\_t writeBuffer[BUFFER\_SIZE];

uint8\_t readBuffer[BUFFER\_SIZE];

volatile bool isTransferDone;

/\* TODO: Define any additional data used by the application. \*/

} APP\_DATA;



#### (九) app.c 里增加如下测试代码

// ************************************	***********
// ************************************	**********
// Section: Global Data Definitions	
// ************************************	***********
// ************************************	************
#define AT24_EEPROM_MEM_ADDR	0x00
// ************************************	************

/\* Application Data

Summary:

Holds application data

Description:

This structure holds the application's data.

#### Remarks:

This structure should be initialized by the APP\_Initialize function.

Application strings and buffers are be defined outside this structure.

#### \*/

#### APP\_DATA appData;

```
// Section: Application Callback Functions
/* TODO: Add any necessary callback functions.
*/
void APP_EEPROM_EventHandler(DRV_AT24_TRANSFER_STATUS event, uintptr_t context)
{
  switch(event)
  {
    case DRV_AT24_TRANSFER_STATUS_COMPLETED:
      appData.isTransferDone = true;
      break;
    case DRV_AT24_TRANSFER_STATUS_ERROR:
    default:
```



```
appData.isTransferDone = false;
    appData.state = APP_STATE_ERROR;
    break;
 }
}
// Section: Application Local Functions
/* TODO: Add any necessary local functions.
*/
// Section: Application Initialization and State Machine Functions
Function:
 void APP_Initialize (void)
Remarks:
 See prototype in app.h.
*/
void APP Initialize (void)
{
 /* Place the App state machine in its initial state. */
 appData.state = APP_STATE_INIT;
 appData.isTransferDone = false;
}
Function:
 void APP_Tasks (void)
```



Remarks: See prototype in app.h. \*/ void APP\_Tasks (void) { /\* Check the application's current state. \*/ switch (appData.state) { /\* Application's initial state. \*/ case APP\_STATE\_INIT: appData.drvHandle = DRV\_AT24\_Open(DRV\_AT24\_INDEX, DRV\_IO\_INTENT\_READWRITE); if (appData.drvHandle != DRV\_HANDLE\_INVALID) { DRV\_AT24\_EventHandlerSet(appData.drvHandle, APP\_EEPROM\_EventHandler, 0); appData.state = APP\_STATE\_WRITE; } else { appData.state = APP\_STATE\_ERROR; } break; case APP\_STATE\_WRITE: /\* Fill up the write buffer \*/ for (uint32 t i = 0; i < BUFFER SIZE; i++) { appData.writeBuffer[i] = i; } appData.state = APP\_STATE\_WAIT\_WRITE\_COMPLETE; if (DRV\_AT24\_Write(appData.drvHandle, appData.writeBuffer, BUFFER\_SIZE, AT24\_EEPROM\_MEM\_ADDR) == false) { appData.state = APP\_STATE\_ERROR; 20



#### }

{

break;

```
case APP_STATE_WAIT_WRITE_COMPLETE:
    if (appData.isTransferDone == true)
    {
        appData.isTransferDone = false;
        appData.state = APP_STATE_READ;
    }
    break;
case APP_STATE_READ:
    appData.state = APP_STATE_WAIT_READ_COMPLETE;
    if (DRV_AT24_Read(appData.drvHandle,
             appData.readBuffer,
             BUFFER_SIZE,
             AT24_EEPROM_MEM_ADDR) == false)
    {
        appData.state = APP_STATE_ERROR;
    }
    break;
case APP_STATE_WAIT_READ_COMPLETE:
    if (appData.isTransferDone == true)
    {
        appData.isTransferDone = false;
        appData.state = APP_STATE_VERIFY;
    }
    break;
case APP_STATE_VERIFY:
    if (memcmp(appData.writeBuffer, appData.readBuffer, BUFFER_SIZE ) == 0)
    {
        appData.state = APP_STATE_IDLE;
    }
    else
```

appData.state = APP\_STATE\_ERROR;



#### } break;

case APP\_STATE\_IDLE: /\* Turn on the LED to indicate success \*/ LED\_On(); break;

case APP\_STATE\_ERROR: default: LED\_Off(); break;

}

}

#### (十)编译下载测试

用 Micro USB 线通过 EDBG 调试口将 SAM E70 Xplained Board 开发板连接到电脑 编译并下载程序:



如果程序正常运行,你将看到 LEDO 指示灯被点亮。

## 四、 总结

本文展示了如何通过 MPLAB X IDE 和 MHC 一步步完成了一个通过 I2C 的程序,开发人员可以从这个过程了解到 Harmony 配置开发的全过程。