Introduction to Atmel Software Framework

ELC 4438 Lab Manual

February 1st, 2016



1. Please watch http://goo.gl/jeikPh and have a quick look at http://goo.gl/ILTQuB before you start.

Select File - New Project, GCC ASF Board Project. Give a name to your project then click OK.



 Choose Select By Board and BoardTypes Atmel. Type SAM4L in the Search for Board window. Choose SAM4L Xplained Pro – ATSAM4LC4C. Download and read the datasheets. Click OK. (Please be patient as it may take a few minutes.)

Board Selection					×
○ Select By Dev	ric	tensions Atmel ASF	3.19.0) 🔻	Device	Board
BoardTypes Atm SAM4E-XPR SAM4E XPIa SAM45 XpIai SAM633 XpI XMEGA-ALU SAM D11 Xp SAM D21 Xp	nel D - ATSAM4EIGE inted Pro - ATSAM4LC8C inted Pro - ATSAM4LC8C inted Pro - ATSAM45D32C ainted Pro - ATSAM53N19 X plainted Pro - ATSAM511014AM lainted Pro - ATSAM511014AM lainted Pro - ATSAM52018	Sea	ch for Board	Device Name: App./Boot Memo Data Memory (byt EEPROM (bytes): Speed: Vcc: Family: Datasheets Supported Tools	ATSAM4LC4C ry (Kbytes): 256 tes): 24816 N/A 0 0/0 SAM4
SAM R21 Xpl	App./Boot Memory (Kbytes)	Data Memory (bytes) 34816	EEPROM (bytes) N/A	 <u>Atmel-ICE</u> <u>EDBG</u> <u>J-Link</u> J-LinkARM-Pr 	0
				J <u>-Link Ultra</u> SAM-ICE JTAGICE3	
					Ok Cancel

3. Open the source in **Solution Explorer – src** directory – main.c.

You can use left-click on any function (ioport_get_pin_level/ ioport_set_pin_level) and choose **Goto Implementation** (ALT-G) to see its implementation.

Click "**No tool**" on the menu toolbar and select "**EDBG**" for debugger/ programmer in the Tool Tab.

Click "Start without debugging" (Green triangle) in the menu toolbar.

Verify that the code is running on the board by using SW0 to turn on LED0.

4. Click **ASF Explorer** and the white triangle before the module name to see **API documentation** for each module and **Quick Start Guide** for several modules.

Read IOPORT API documentation and Quick Start Guide.





5. You can add modules (driver/component/service) through menu bar ASF - ASF Wizard.

Add Standard serial I/O (stdio) (driver) and System Clock Control (service). Click Apply then OK.

Please read their API documentation and Quick Start Guide. Do not forget to adjust the configuration files in **config** directory within **Solution Explorer**.

Compare the configuration files with the ones of the previous lab (Introduction to Atmel SAM4L Xplained Pro).

MyAnger-Atmethodo								
En Est Vew Vacinty AS Breject Build Debug Tools Window Help								
SAM4L Xplained Pro - 6530 ASF Wizard X MyProject* main.c	Solution Explorer + 4 X							
Project MyProject Device: ATSAM4LC4C	Solution 'MyProject' (1 project)							
	MyProject							
Extensions Version	 Dependencies Defendencies 							
Available Modules	Selected Modules	> 🖼 Libraries						
Extensions: Atmel ASE(3.19.0) Show: All Search for modules	> 🐻 Generic board support (driver)	b 🛄 src						
	DPORT - General purpose I/O service (service)							
Image of the sensor with I2C interface (component)	WDT - Watchdog Timer (driver)							
Image and components.display.s42412a (component)								
5 E SD/MMC stack (component) spi								
Serial I/O - Host (component) uart *								
Serial I/O - NCP (component) uset								
Image: Sleep manager (service)								
Image:								
p 10 SPL - Serial Peripheral Interface Master (Common API) (service) standard_spi •								
Image: SSD1305 OLED controller (component)								
Image: Standard serial I/O (stdio) (driver)								
To The Timer Counter Mitherit								
TML_Two-Wire Interface (Common API) (service)								
Im TWI - Two-Wire Master Interface (driver)	🔍 ASF 👄 VA.V 🌴 VA 🙉 Solu							
TWIS - Two-Wire Slave Interface (driver)	Properties + Ŧ ×							
Unit test framework (driver) E ISART - Serial interface (ramina)	MyProject Project Properties •							
Im USART - Univ. Syn Aoync Rec/Trans (driver)		22 24 33						
USB Device (service) cdc •		 Misc 						
USB Host (service) cdc •		Project File MyProject.cproj						
Info Antino Datalia		Fight Folder C. (SHITGERSPIRECTO						
Inty ALSHIS VESHS								
TC - Timer Counter								
Driver for the TC (Timer Counter). The Timer Counter includes three identical 36-bit Timer Counter channels. Each channel can be independently programmed to perform a wide range of functions including frequency measurement, event counting, interving measurement, but ends expension development and the modulation.								
		Project File						
Add>> Apply Rever	t << Remove Summary	The name of the file containing build, configuration, and other information about						
Outout		- 8 X						
Show output from: Build								
Tome suitant target "coreculo" in project myrolycet (proj. Tomes "Bortel (Balant" collegation dis target in a collegation of the suitant is a collegation of the suitant in a collegation of the suitant is a collegation of t								
Target "Bulld" in file "C:\Atmel\Atm								
Done building target "Build" in project "MyProject.cproj". Done building project "MyProject.cproj".								
Build surgeaded								
Bulls 1 successed or up-to-date, 0 failed, 0 skipped								
4								

6. Add procedure configure_console (void) from the previous lab (Introduction to Atmel SAM4L Xplained Pro).

Call it from main (void) after board init() and sysclk init().

Print something in PC terminal such as "Hello World" after the initialization. When SWO is pressed, change the output and print "ON".

7. Modify the program to use **SysTick_Config** and **SysTick_Handler**.

Refer to the code on the following page.

```
#include <asf.h>
/** Global g_ul_ms_ticks in milliseconds since start of application */
volatile uint32_t g_ul_ms_ticks = 0;
volatile uint32_t time_delay = 500;
#define BLINK PERIOD
                            1000
static void configure console(void)
{
   const usart_serial_options_t uart_serial_options = {
          .baudrate = (115200UL),
          .charlength = US MR CHRL 8 BIT,
          .paritytype = US MR PAR NO,
          .stopbits = US_MR_NBSTOP_1_BIT,
   };
   /* Configure console UART. */
   stdio_serial_init(COM_PORT_USART, &uart_serial_options);
}
void SysTick_Handler(void)
{
   g_ul_ms_ticks++;
   //printf("Ticks");
}
int main (void)
{
   board_init();
   sysclk_init();
   configure console();
   /* Configure systick for 1 ms */
   puts("Configure system tick to get 1ms tick period.\r");
   if (SysTick_Config(sysclk_get_cpu_hz() / BLINK_PERIOD)) {
          puts("-F- Systick configuration error\r");
          while (1);
   }
   while (1) {
          // Is button pressed?
          if (ioport_get_pin_level(BUTTON_0_PIN) == BUTTON_0_ACTIVE) {
                 // Yes, so turn LED on.
                 //ioport_set_pin_level(LED_0_PIN, LED_0_ACTIVE);
                 time delay = 500;
          } else {
                 // No, so turn LED off.
                 //ioport_set_pin_level(LED_0_PIN, !LED_0_ACTIVE);
                 time delay = 2000;
          }
          if (g_ul_ms_ticks>=time_delay)
          {
                 g ul ms ticks = 0;
                 LED Toggle(LED0);
          }
   }
}
```

8. Load new example project: SAM4L QTouch Example – SAM4L Xplained Pro

Read the code main.c in Atmel Studio to understand how it works.

Refer to the **SAM4L Datasheet** (Chapter 19) to understand the Asynchronous Timer (AST). http://www.atmel.com/Images/Atmel-42023-ARM-Microcontroller-ATSAM4L-Low-Power-LCD_Datasheet.pdf

Using the same procedure, run the program and use the onboard Qtouch button.

9. Referring to both of the sample programs above, modify and write a program that blinks a LED at different frequencies.

You have two buttons (SWO and Qtouch) to control the blinking frequency. If no button is pressed, the LED blinks at 0.5 Hz. When the SWO button is pressed, the LED blinks at 1 Hz. When the QTouch button is pressed, the LED blinks faster at 2 Hz. If both buttons are pressed, the LED blinks at 4 Hz.

Save your code and show your experiment result to your TA.

10. Extra: Create the same program using timer counter (TC) instead of system clock tick.

Refer to the SAM4L Datasheet (Chapter 30) to understand the Timer Counter. http://www.atmel.com/Images/Atmel-42023-ARM-Microcontroller-ATSAM4L-Low-Power-LCD_Datasheet.pdf