# <u>Lab 01</u>

Developed by: Muhammad Imran and Peng Cheng

Date : 2015-02-06

Course : Programming Embedded Systems (ET014G) Mid Sweden University, Sweden.

This lab has three tasks which will enable you to develop understanding of the AVR32 tools i.e., EVK1100 evaluation kit, its software tools including AVR32 studio and framework. Task1 is preparatory task and is required to be done before coming to lab.

### <u>Task 1</u>

#### How to find pin mapping between microprocessor and evaluation board:

In the EVK1100 evaluation board's <u>schematic</u>, you can find different peripherals such as LEDs, push buttons, potentiometer, LDR, etc. These peripherals are connected to microcontroller through GPIOs. In order to use these peripherals, you have to map the pin of the microprocessor to the required peripheral because some pins are multiplexed for three different functions, discussed in later section.

**Example:** if you want to turn on LED0 (It is LED1 on physical layout of the board), you can find in the schematic that it is PIN27 which connects LED0 (Figure 1), through port PB to the microcontroller PIN 15 of the package (Figure 3). PIN27 is also shared with expansion header J26 (Figure 2). The GPIO number for this pin is 59.

In AVR studio, you can find this mapping in "evk1100.h"

• #define LED0\_GPIO AVR32\_PIN\_PB27

And its low level mapping on actual hardware can be seen in "uc3a0512.h"

• #define AVR32\_PIN\_PB27 59

#### Finding port and Pin configuration

Each GPIO line can be assigned to one of 3 peripheral functions; A, B or C as shown in Table 1 and each GPIO line has a unique number ports PA, PB, PC and PX which corresponds to the GPIO pins. For simplicity, the GPIO pins are grouped in different ports. However, the numbering on ports is not directly translated to GPIO numbering. You can use the following formula to find the port and pin number or easy way is to see the AVR datasheet (Table 1) for the pin mapping. NOTE : 32 in the formula corresponds to the fact that "The pins are managed as 32-bit ports" (section 21.5 in datasheet)

Port = floor((GPIO number) / 32), example: floor((59)/32) = 1 (Note: 0 corresponds to A, 1 to B, and so on) Pin = GPIO number mod 32, example: 59 mod 32 = 27

#### For example for LED0

Component	PIN	GPIO PIN	<b>GPIO</b> port/PIN
LEDO	PB27	GPIO 59	1/27

Table 1. (Table 12.9 in datasheet) GPIO controller function multiplexing

TQFP100	VQFP144	PIN	GPIO Pin	Function A	Function B	Function C
9	14	PB26	GPIO 58	TC – B1	USART1 - RI	
10	15	PB27	GPIO 59	TC –A2	PWM – PWM [4]	



Figure 1. LEDS



Figure 3. Microcontroller

#### Show to the lab demonstrator following sub-tasks,

Which port and PIN does GPIO 61 corresponds to?. PORT\_\_\_\_\_, PIN \_\_\_\_\_. What are three functions that GPIO 61 can be associated to.

- (1) \_\_\_\_\_
- (2) \_\_\_\_\_
- (3) \_\_\_\_\_

### Task 2

You have developed understanding of using the datasheet and schematic of the board and microcontroller. Next, step is to use the software for programming the microcontroller.

#### Start AVR32 studio

- In the lab computer, AVR studio is located in C:\Program Files\Atmel\AVR Tools\AVR32 Studio.
- Click on the windows start button and locate AVR32 studio under the *Program* menu.
- The first time you start the program you might need to select a location of you workspace.
- If not, click on the File menu and select switch workspace. Select a location for you workspace and create folder at that location called workspace (important)
- Create new project by opening *File/New/Project/AVR32 Project from template* as shown in **Figure 5**Figure 4. Give authors information and click finish.

CDT Project	- • ×
New AVR C Project	
Create C project of selected type	
Project name: lab01	
Vert Use default location	
Location: C:\atmel\Workspace_AT32UC3_ET	014G\lab01 Browse
Choose file system: default	
Project type:	Toolchains:
Erensty Project     Empty Project     Star STK1000 application     Star EVK1100 application     Star EVK1104 application     Star EVK1104 application     Star EVK1104 application     Star EVK1101 application     Star EVK1101 application     Star EVK1010 application     Star STK600 RCU310 application     Star STK600 RCU310 application     Start Library     Start Library     Show project types and toolchains only if	bey are supported on the platform
? < Back Ne	xt > Finish Cancel

Figure 4. Project template

• Go to the menu bar and click Framework as shown in Figure 5 for selecting required drivers/components/service from the framework. The drivers' window is shown in Figure 6. Similarly select components and services and click finish.



Figure 5. Framework

dd, remove, update AT32UC3Ax software framework drivers (v	ersion 1.7.0) to the selected project.	L.C.
is creates a 'src/SOFTWARE_FRAMEWORK/COMPONENTS' dir Ided to the project.	ectory with the selected services. Related drivers, compo	ments and services are automatically
CPU - Cycle Counter	<ul> <li>This is a software API for the General</li> </ul>	I Purpose Input/Output registers.
CPU - MPU	View context help	
EBI - External Bus Interface SDRAMC	The sources	
EBI - External Bus Interface SMC		
EIC - External Interrupt Controller		
FLASHC - Flash Controller	-	
GPIO - General Purpose I/O Controller	*	
INTC - Interrupt Controller		
MACB - Ethernet MAC 10/100		
PDCA - Peripheral DMA controller		
PM - Power Manager		
PWM - Pulse Width Modulation		
RTC - Real Time Counter		
SPI - Serial Peripheral Interface	-	

Figure 6. Selecting drivers/components/services

• Now you add/write your required source code/header files in the main as shown in Figure 7.

AVK - IADUL/SIC/MAIN.C - AVK32 STUDIO	
File Edit Source Refactor Navigate Search Proje	t Run Framework Window Help
📬 🕶 📄 💼 🔸 🔷 🛪 🖗 🗸 🖓	• 🔗 • 🚺 🕹 🗐 👖 🧆 🗢 • 🔿 •
Project Explorer 🛛 📄 🔄 🔽 🗖	main.c 🗵
🗁 UTILS 🔺	/**************************************
readme.html	Name : main.c
🔁 uc3softwareframework-releasenotes.pd	Author : MIUN
😝 EVK1100 - APPLICATIONS - Control Panel Dem 🖊	Copyright : Not really
😂 lab01	Description : EVK1100 template
Software Libraries	***************************************
🔊 Includes	
🖓 src	// Include Files
SOFTWARE FRAMEWORK	#include "board.h"
ASM	#include "gpio.h"
BOARDS	
	Y TODO: Add software framework include drivers below */
C DRIVERS	//#include "gplo.n"
CPU CPU	//#include "adc.n"
EL ASHC	internation (and d) (
GPIO	inc main(vola) {
Serie c	/* TOPO: replace this comment with your code */
Se gpione	/ TODO, replace this comment with your code "/
INTC	return 0.
	3
CDI	
	<
D main a	
C main.c	🛣 Problems 🕴 🖳 Console 🕮 Properties

Figure 7. Adding header files in main

Write a program that sets the CPU frequency to 12 MHZ from OSC0. The program gets value from push button0 and turn on LED6 by using polling method. The power manage drivers in the framwork can be used to set the frequency.

You can find functions to LED and push buttons in "gpio.h" and "evk1100.h".

#### **Programming the device**

#### Create new programming target as shown in Figure 8

- Connect the USB cable between the EVK1100 and the computer
- In the AVR32 target window create a new target as shown in Figure 8
- Right click on the target and select properties
- Select USB DFU programmer
- Select Microcontroller UC3A0512

🖹 Problems 🗍	Console Droperties	5 23			2 -						
🗖 New Tar	get										_
General	Details					Â		AVR Targets 🙁	🐑 Error Log	·	
Details	Debugger/programmer:	USB DFU	Device:	AT32UC3A0512	Select			Name	Adapter	Board	м
Daisy Chain	Clock source:	Internal RC oscillator	▼ Board:	EVK1100	~		۲	AVR32 Simulat	AVR32 Simulat	AVR32 Simulat	UC
Information						Ξ	不	New Target	USB DFU	EVK1100	UC
	▼ Connection										

Figure 8. Target selection

#### Download the program to the processor

- The AVR32 is pre-programmed with a bootloader in order for your program to be downloaded to the correct starting address.
- Set the main power switch to the USB position
- To activate the bootloader, Push and hold the Joystick
- While pushing the joystick, push and release the reset button
- Right click on the USB programming target
  - o Select program
  - Locate the .elf file in you Debug folder of you project as shown in Figure 9.
  - Mark all options and press OK
  - Your program will know download to the AVR32

🙆 Progra	m target		23
JTAGIC	E mkli		AVR32
Program	n target		
File path:	NTROL-PANEL\AT32UC3A05	512_EVK1100\GCC\uc3a0512-ctrlpanel.elf	Browse
Offset:	0x0		
Length:			<b>V</b> Entire file
Options			
Verify	memory after programming	📝 Erase flash before programming	
📃 Unloc	k flash before erasing	Reset MCU after programming	
V Start	executing after programming		
?		ОК	Cancel

Figure 9. Elf file location

## Task 3

Write a C program which calculates the following numbers. Select suitable datatypes for these numbers.

x = 12345678;y = 87654321;

a = 1234.5678; b = 8765.4321; Measurement start here z = x \* y; Measurement stop here

Measurement start here c = a \* b; Measurement stop here

- Now, measure their calculation speed by using the CPU cycle counter and use the LCD display of the EVK1100 board to show the measured results.
- Display the time in micro second after converting the CPU cycles with following formula (you can see this formulas in cycle\_counter.h).

fcpu\_hz = 12000000; time in us = (CPU cycles\* 1000000 + fcpu\_hz-1) / fcpu\_hz;

The program needs to be analysed for the two cases: Case1 : (set optimization level to 0) Case2 : (set optimization level to 1)

Motivate from the ATMEL instruction manuals, what these optimizations are used for.

Optimization	0:
Optimization	1:

#### TIPS:

You will need these drivers, components and service in this lab

- CPU cycle counter driver: CYCLE COUNTER".
- LCD display driver : DIP204".
- Power manager driver :PM-power manager

#### How to set optimization level

You can either set it using graphical way as shown in Figure 10 or changing it in the config.mk file as shown in Figure 11.



Figure 10. Setting optimization level



Figure 11. Setting optimization level in config files