Atmel SAM4L Xplained Pro – DAC and ADC

ELC 4438 Lab Manual

March 14th, 2016

Part 1. Digital to Analog Converter (DAC)

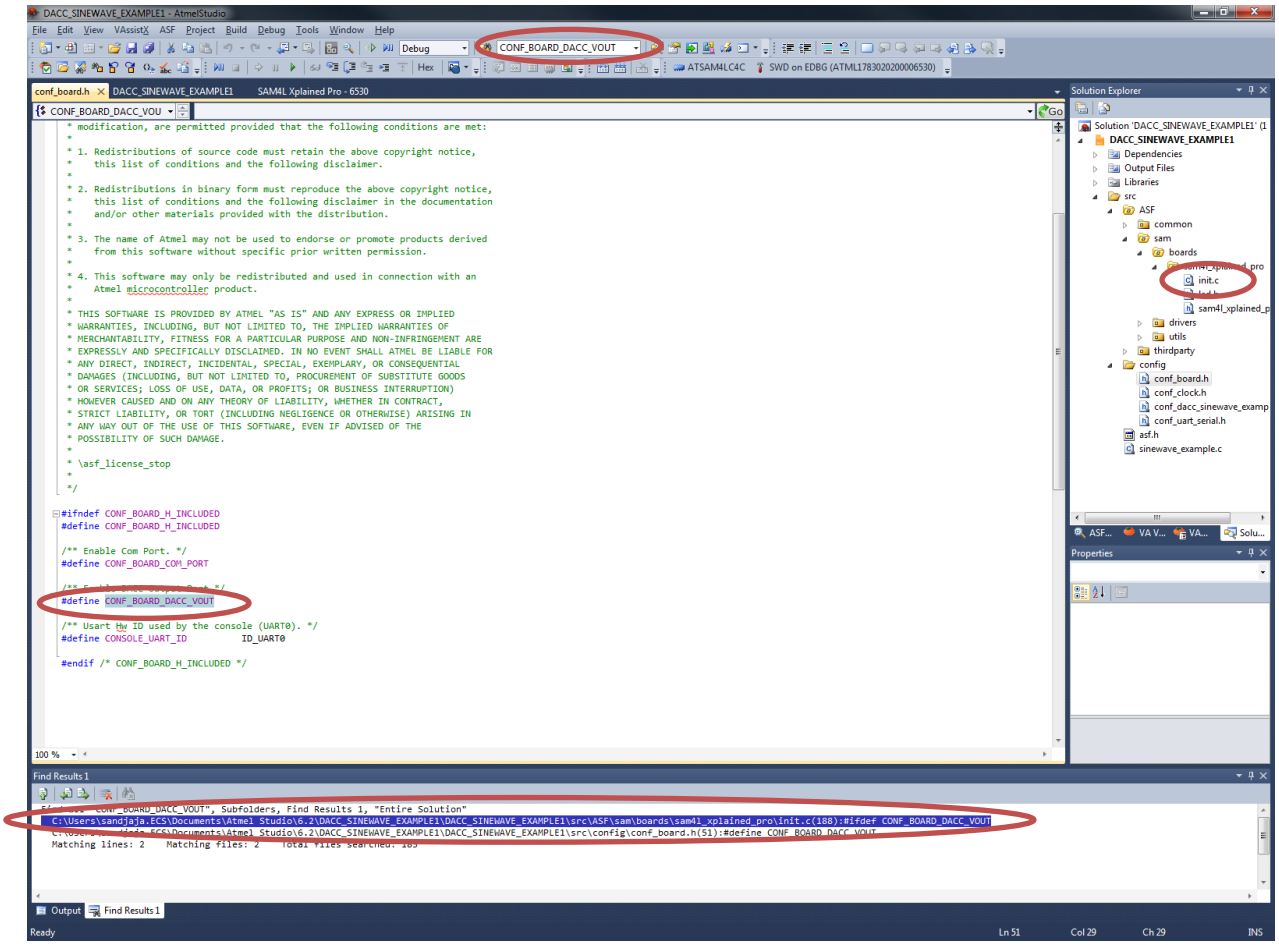
1. Download **ATSAM4L Datasheet**
   http://www.atmel.com/Images/Atmel-42023-ARM-Microcontroller-ATSAM4L-Low-Power-LCD_Datasheet.pdf

   Read Chapter **3.2** about **Peripheral Multiplexing on I/O lines** and Chapter **33** about **Digital to Analog Converter Controller (DACC)**.

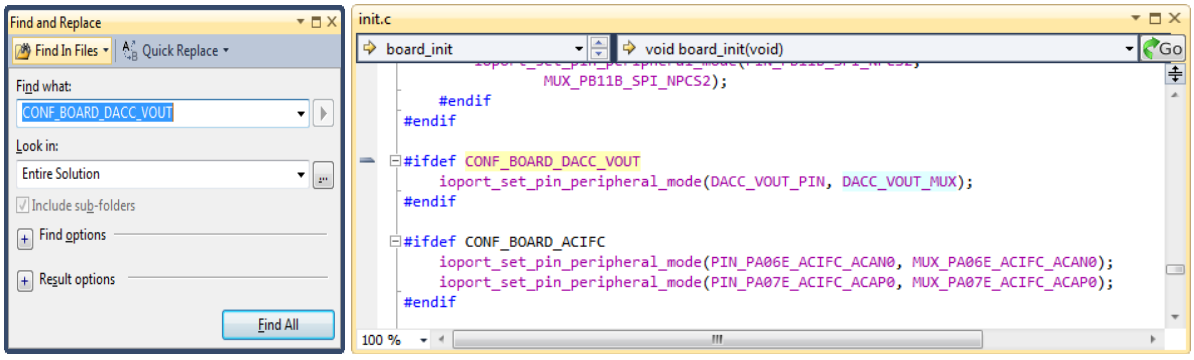
2. Connect the Atmel board without any extension board with the PC through DEBUG USB.

3. Open **Atmel Studio 6.2**.
   Click New Example Project in the Atmel Studio 6.2 – SAM4L Xplained Pro tab.
   Click on the small triangle before Atmel-Atmel Corp to see all available examples.
   Choose **DAC Sinewave Example – SAM4L Xplained Pro.**
   Click **OK** to create this example. Accept the license agreement and wait patiently.

4. Click **No tool** on the menu toolbar and select EDBG for Selected debugger/programmer in the Tool Tab.
   Click Start without debugging (green triangle) in the menu toolbar.
   Open **config_board.h** in the config directory.
   Use **Find in Files** (Ctrl+Shift+F) to find where **CONF_BOARD_DACC_VOUT** is used.

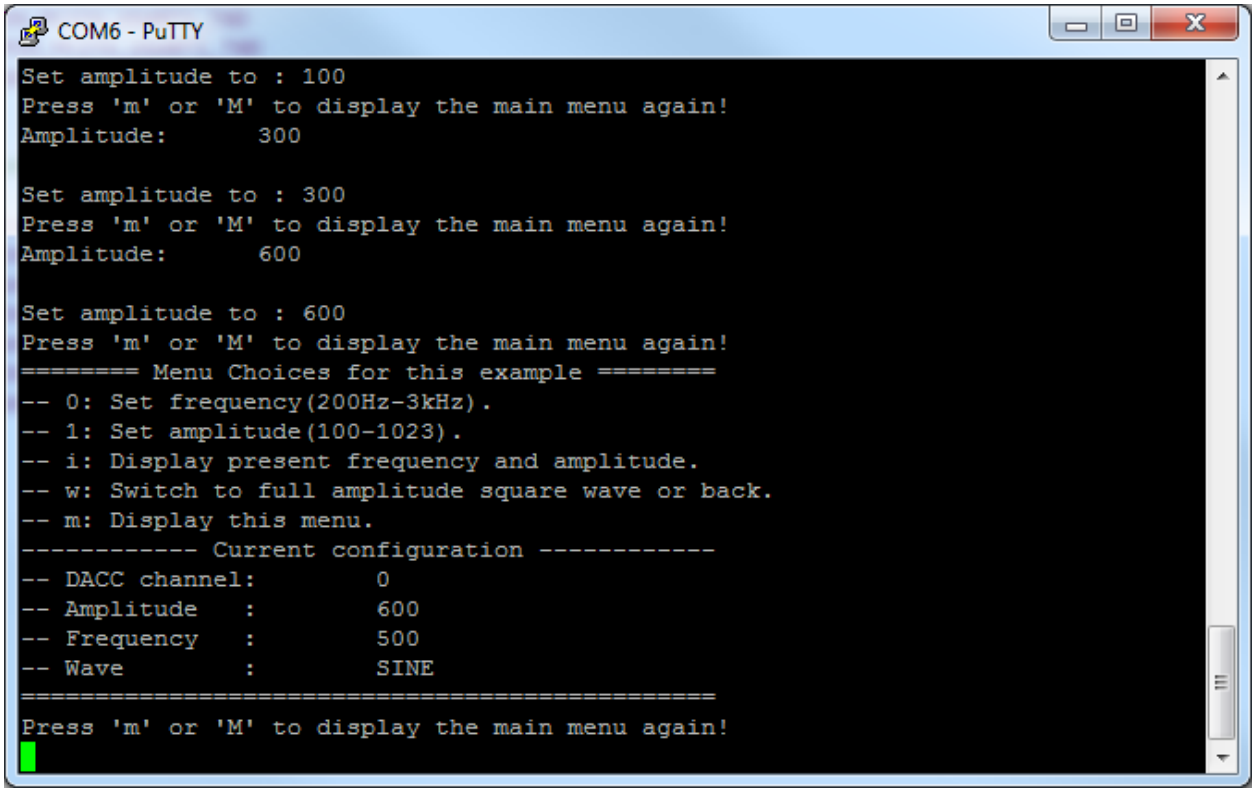


5. In the **init.c**, **DACC_VOUT_PIN** is set to be in **DACC_VOUT_MUX**.
   Find out what is the implementation of **DACC_VOUT_MUX** by using right click – Goto Implementation (Alt-G). Keep doing it until you find that **VOUT** is on **PA06** mux A



6. Connect oscilloscope probe from Agilent MSO-X 3024A to **PA06 (EXT 3 pin 9).**
   Do not forget to connect the ground too.
   You will see the sinewave when using **autoscale** button on Agilent MSO-X 3024A oscilloscope.

   Try to change the amplitude, frequency and shape (to a square wave) by typing the value into the PuTTY terminal.
   Try to understand the program flow by comparing with DACC API Documentation (see ASF Explorer to find DACC API Documentation).



7. Modify the program to show a triangle wave and a sawtooth wave in addition to the sine wave.

   Show the results and explain your program to your TA.

   Congratulations! You have made a function/arbitrary waveform generator.

Part 2. Analog to Digital Converter (ADC)

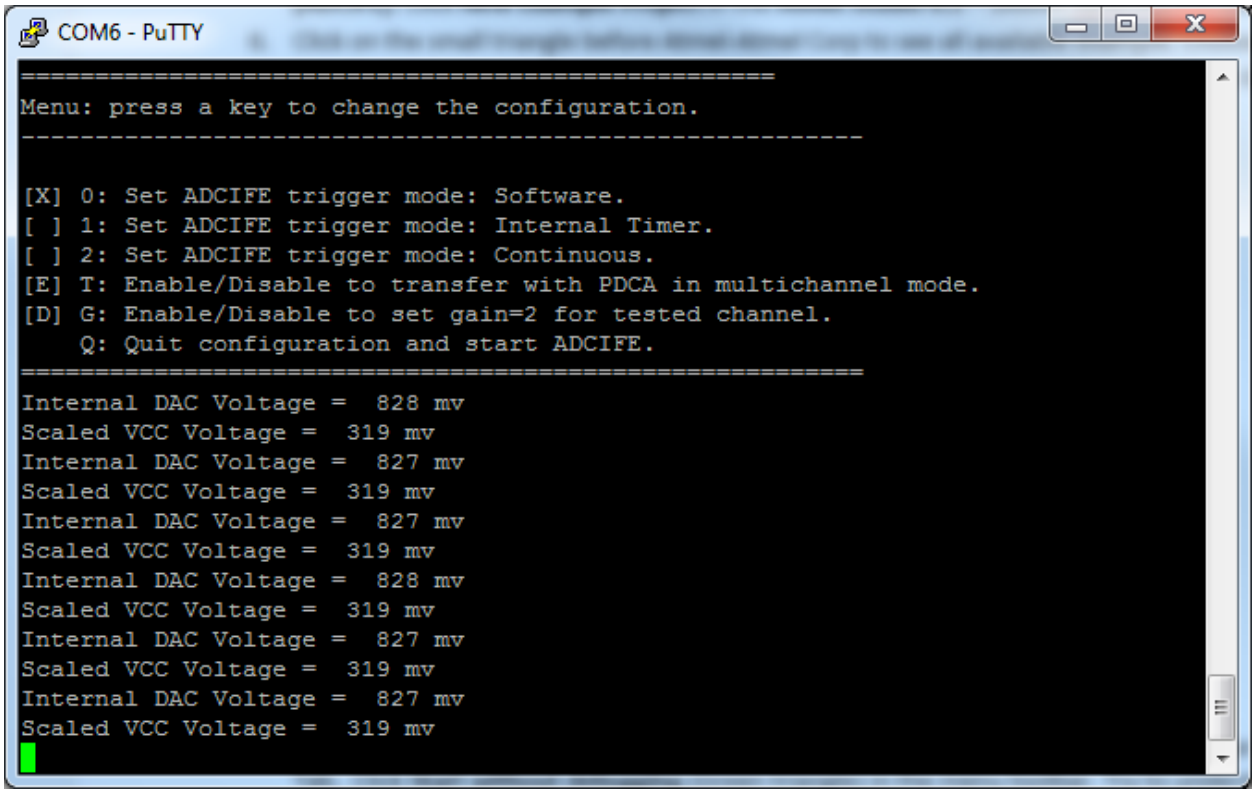
1. Download **ATSAM4L Datasheet**
   http://www.atmel.com/Images/Atmel-42023-ARM-Microcontroller-ATSAM4L-Low-Power-LCD_Datasheet.pdf

   Read Chapter **16** about **Peripheral DMA Controller (PDCA)** and Chapter **38** about **ADC Interface (ADCIFE)**.

2. Connect the Atmel SAM4L Xplained board with Atmel SEGMENT LCD1 boards.
   Take off VLCD A and VLCD B jumper.
   Connect the Atmel board with the PC through DEBUG USB.

3. Open **Atmel Studio 6.2**.
   Click New Example Project in the Atmel Studio 6.2 – SAM4L Xplained Pro tab.
   Click on the small triangle before Atmel-Atmel Corp to see all available examples.
   Choose **ADC Interface Example – SAM4L Xplained Pro.**
   Click **OK** to create this example.
   Accept the license agreement and wait patiently.

4. Click **No tool** on the menu toolbar and select EDBG for Selected debugger/programmer in the Tool Tab.
   Click Start without debugging (green triangle) in the menu toolbar.
   Understand the program and explore all the options.
   ASF Explorer has a good summary on how to use **PDCA** and **ADCIFE** driver in the Quickstart guide.



5. Combine the program with LCD (**sam.components.display.c42412a**).
   Program it so that LCD will always show Internal DAC Voltage with its unit (mV icon).
   When SW0 is pressed or light is detected, change DAC Voltage into the value in the table below.

	Light is detected	Light is not detected
SW0 is pressed	467-468 mv	827-828 mv
SW0 is not pressed	263-263 mv	672-673 mv

Show the results and explain your program to your TA.

6. Change the `uc_pdc_en` into 0 in the main(void) function to disable Peripheral DMA.

```
/* Set default ADCIFE test mode. */
g_adc_test_mode.uc_trigger_mode = TRIGGER_MODE_SOFTWARE;
g_adc_test_mode.uc_pdc_en = 1;
g_adc_test_mode.uc_gain_en = 0;
```

7. In the start_adc(void) function, configure the `.muxpos` into `ADC_MUXPOS_1` and `.internal` into `ADC_INTERNAL_2`.
   Based on **ATSAM4L Datasheet (Chapter 38, p. 1014-1015)**, the most significant bit (MSB) of `.internal` and `.muxneg` will determine the MUX selection on Negative ADC. The least significant bit (LSB) of .internal and .muxpos will determine the MUX selection on Positive ADC. `ADC_MUXPOS_1` and `ADC_INTERNAL_2` will choose **AD1** as the positive input to ADC. Negative input to ADC is set to Pad ground because of `ADC_MUXNEG_1` and `ADC_INTERNAL_2`

```
struct adc_seq_config adc_seq_cfg = {
        /* Select Vref for shift cycle */
        .zoomrange = ADC_ZOOMRANGE_0,
        /* Pad Ground */
        .muxneg = ADC_MUXNEG_1,
        /* DAC internal */
        .muxpos = ADC_MUXPOS_3,
        /* Enables the internal voltage sources */
        .internal = ADC_INTERNAL_3,
        /* Disables the ADC gain error reduction */
        .gcomp = ADC_GCOMP_DIS,
        /* Disables the HWLA mode */
        .hwla = ADC_HWLA_DIS,
        /* 12-bits resolution */
        .res = ADC_RES_12_BIT,
        /* Enables the single-ended mode */
        .bipolar = ADC_BIPOLAR_SINGLEENDED
};
```

8. Add this line to init.c in ASF – boards – sam4l_xplained_pro folder:

```
ioport_set_pin_peripheral_mode(PIN_PA05A_ADCIFE_AD1, MUX_PA05A_ADCIFE_AD1);
```

   This will cause the mux to connect pin PA05 (EXT1_PIN_4) to AD1 pin in the ADC.
   See **ATSAM4L Datasheet (Chapter 3.2.1, p. 19)**

9. Lower the delay in the main loop to make it more responsive.
   Run the program and connect the **Agilent E3647A** Power Supply positive output to **EXT1_PIN_4** and the negative output to **GND**.
   Change the voltage and see it change also in the LCD.

   Show the results and explain your program to your TA.

   Congratulations! You have made a digital multimeter.