Laboratory 4

COMSYS 301 : Design : Hardware Software Systems

Objectives:

- To set up serial communication between a PSoC and a computer
- To set up a PWM signal
- To configure a PWM signal from a PC terminal

1. Abstract of the tasks

- Add an existing project to the current workspace
- Using a terminal program to send data to the PSoC
- Program a UART to receive a serial stream

2. Tasks

In this lab, instructions will be minimal. Refer to lab1 if you don't remember how to achieve a task.

2.1. Opening the workspace

As in lab 1, begin by opening the psoc_intro.cywrk workspace.

2.2. Download "PSoC Code Base" from CANVAS

Extract it in the folder containing the previous lab experiments.

Add the project to the workspace

Make the 'psoc code base' project active.

2.3. Viewing and exploring the schematic

Open TopDesign.cysch, look at all the components. Look at their configuration parameters.

The schematic contains two sheets. One contains a UART and a USBUART. The second contains PWM and quadrature decoder blocks.

The UART is a general purpose serial communication block while the USBUART is a bridge between a USB device and a UART. The USBUART can be used to transfer data between the PSoC and a PC. On examining the code, you will find that a simple but sufficient software interface is provided. You could build on this interface to create a custom interface.

Note: The USBUART bridge requires specific settings for the CPU clock. These have been set up in "PSoC Code Base".

2.4. Electrical Connections

Until now you have been using only one USB port – Port J10 ("USB Finger Connector"). In this exercise, the USBUART Bridge uses the USB pins of the PSoC (Target device on the Kit-059). These are connected the other USB port – Port J6 (USB Micro-B connector). Hence, you will need make use of both the USB ports; one for programming and the other for data exchange between the PsoC and a PC

2.5. Test with terminal

Before you begin, it is advisable to check that your PSoC is communicating with a terminal program (PuTTy for example).

1. Start a terminal program.
2. Configure it using the following settings: 57.6k baud, disable flow control, one stop bit, echo off.
3. Connect both the USB cables and "PSoC code base" programmed into the PsoC.
4. Establish a connection between the terminal and the Psoc.

5. Type “qwerty34” on the keyboard while the terminal window is selected. Note that what you are typing will not be displayed directly on the screen but it is sent nonetheless. The PSoC should echo “you entered: qwerty34”.

2.6. Changing the main.c file

Examine the datasheet of the PWM block. You will write code in main.c to output a specific duty cycle at any one of the 4 PWM output pins.

1. Select a suitable PWM pin and connect a scope probe to view the signal. The ground terminal of the probe must be connected to the ground of the PSoC board
2. Add code to main.c to activate and hence produce a PWM signal on the selected pin. Try 20%, 50%, 80%
3. What is the expected frequency that of the PWM signal? Use the scope to measure the frequency and the dutycyle
4. In main.c change the “Period” and observe the changes. How does the clock source and Period influence the frequency of the observed PWM signal.
5. Write a simple command interpreter for the PSoC so that you can alter the pulse width and/or frequency from the terminal. e.g. entering “p 20” on the terminal should change the duty cycle to 20%. You may choose your own syntax. This kind of capability is invaluable to a Computer Systems Engineer.
6. If the duty cycle is changed in the schematic (i.e. by editing the PWM symbol in the schematic), the recompilation time increases. Why does this happen?

3. Your notes

You are encouraged to take notes of the quick questions and remarks made in the lab because some of them may be in the quiz.