
Workshop 3

Lecture Material Covered : Topic 3 UART

Required Documents:

Register Reference.pdf (from the Course Resources Page link on the Canvas homepage)

Course Reference notes (from the Course Resources Page Link on the Canvas homepage)

TM4C1294NCPDT Datasheet from the Course Resources Page Link on the Canvas homepage)

The aim of this workshop is to :

1. Familiarise yourself with the PCTL register and setting the GPIO pin to an alternate function. (GPIO configuration)
2. Write code to initialise and configure a UART. (UART configuration)
3. Write code to transmit a character. (UART Use)
4. Write code to transmit a string. (UART Use)

1 Setting the GPIO PCTL Register for the Alternate Function

Use the TM4C1294NCPDT datasheet (section 26.4 *GPIO Pins & Alternate Functions*, Table 26.5, pg 1808) to find the Port, Pin and PCTL values associated with the alternate functions listed in the table. Fill in the following table accordingly:

Alt Function	Pin	PCTL Value	Embedded C Code
U0RX			
M0PWM7			
I2C7SCL			
U0Tx U0Rx (both)			
U1Tx, U1Rx (both)			

Question 2 on the next page

2 Initialising and Configuring the UART

Background

- UART0 (U0) uses an onboard FTDI chip to translate your serial signals to USB signals for communication between the Tiva and Teraterm via the PC's USB. We will be using UART 0 in this tutorial.
- The GPIO A registers can be accessed using `GPIOA_AHB->register_name`
- The UART 0 registers can be accessed using `UART0->register_name`
- The C code for setting up the GPIO and UART **must** be performed according to the steps set out in Appendix A
- The configuration takes two parts:
 1. *Configure the GPIO for an Alternate Function (UART)*
 2. *Configure the UART communication parameters*
- The System Clock speed is **16MHz**.
- Use the main.c file in the template to write your code. The steps required are given as comments in this file. You will need to add your own code to the provide functions.

Task

Write the code to initialise and configure the GPIO requires for the UART 0 peripheral.

Write code to initialise and configure UART 0 for the following communication parameters.:

- a baud rate of 115200
- a word length of 8 bits
- No parity
- 1 stop bit
- System clock $\div 8$

Test the code by checking it with the debugger.

Note: You will need to calculate the **IBRD** and **FBRD** register values to set the baud rate to **115200**.

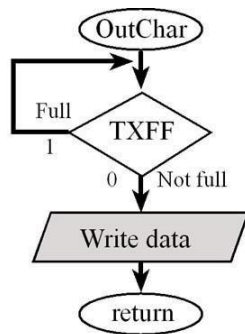
You will need to determine the **LCRH** register value for the communication mode of **8 bits, NO parity, 1 stop bit**

You will need to determine the **CTL** register values required to enable and disable the UART and a ($\div 8$) prescaler

3 Transmitting a Character

Modify the code in Question 2 to output a character to the terminal using the UART0 Tx pin (U0Tx). The function will use the UART0->FR and UART0->DR registers. You will need Teraterm to be connected and set up with same communication parameters as Question 2.

(See the flow chart below for Transmitting a character)



Bit	Name	Cleared (=0)	Set (=1)
5	TxFF	Tx Empty (ready for data)	Tx Full (not ready for data)
4	RxFE	Rx Full (data available)	Rx Empty (no data received)

The code in the template will continually print the letter 'U' when you have successfully written the function to transmit a character to teraterm.

<Quiz Questions supplied after you successfully complete section 3>

4 A serial message.

Use the program from Question 3 and modify it to print out a message. After the message is printed then print out the Carriage Return character and the Line Feed character which is equivalent to a '\n'.

The message can be string which is initialised when it is declared.

Appendix A Configuration Procedure

GPIO

Set	Action	Register
1	Enable clock on required GPIO Port Wait for it to stabilise and reset	SYSCTL->RCGCGPIO SYSCTL->PRGPIO
2	Enable alternate function for I/O pin(s)	GPIOX->AFSEL
3	Select type of alternate function required (ie UART) for I/O pin	GPIOX->PCTL
4	Enable digital access for I/O pin	GPIOX->DEN

Where *X* is the port letter.

UART

Step	Action	Register
1	Enable clock on required UART Wait for it to stabilise and reset	SYSCTL->RCGCUART SYSCTL->PRUART
3	Deactivate/disable UART	UART#->CTL
4	Set baud rate (Integer and Fractional registers)	UART#->IBRD UART#->FBRD
5	Set line control parameters	UART#->LCRH
6	Set the clock source	UART#->CC
7	Set prescalar Reactivate/enable UART	UART#->CTL UART#->CTL

Where *#* is the Uart number.