Indian Institute of Technology Bombay
CS684 Autumn Semester 2016

Analog to digital Conversion (ADC) and Serial Communication

E.R.T.S. Lab

July 9, 2016

1 Lab Objective

This lab will introduce you to the use of the analog to digital conversion (ADC) peripheral and Universal Asynchronous Receiver/Transmitter (UART) on the TM4C123GH6PM.

2 Pre-requisite

Please go through the reference documents given in the relevant theory section.

3 Problem Statement

1. Use the inbuilt ADC to interface a joystick with Tiva Board. The analog values read from the joystick has to be converted to digital and displayed on the watch window of the CCS IDE.For information about the joystick pinout refer to the figure below:

2. Now, use the inbuilt UART to communicate the digital values (i.e. both the X axis and the Y axis) to the computer. Use a terminal emulation software like Serial Terminal or Real Term to view the data being sent by the Tiva Board to your computer.
   The data sent should have the following syntax :
   "X: Digital equivalent of the read value"
   "Y: Digital equivalent of the read value"

3. Once you complete the above mentioned problem statements, you have to depict the values received in a graphical representation. In short, create a GUI which tracks the real time movements of the joystick.
   Refer to the Figure 1, the circle marker should move to the left if the joystick button

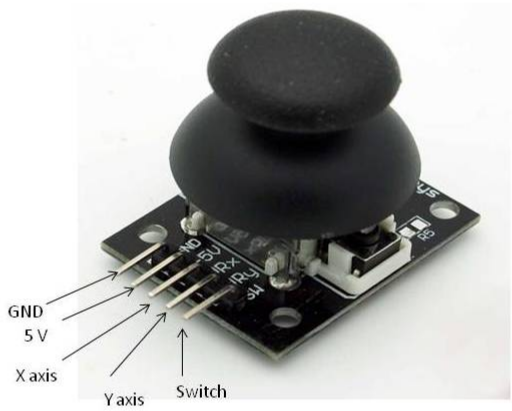

Figure 1: Joystick Pinout

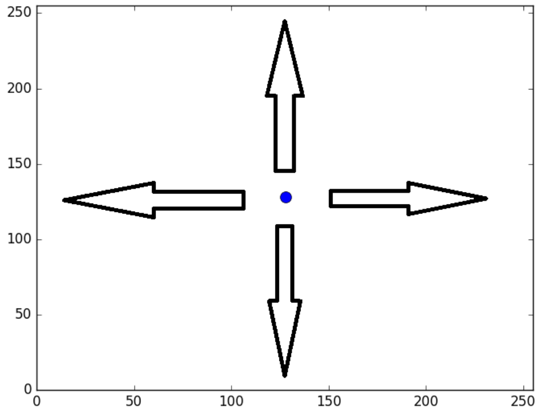

Figure 2: Joystick Graphical interface

is tilted towards left and right if tilted towards right. The marker should remain at the centre when joystick is stationary.

4 Relevant Theory

Some general terms related to ADC. *

1. **Sample Sequencer** The sampling control and data capture is handled by the sample sequencers. The sample sequencers can be assumed to be a module with memory, which can sample different analog sources with a single trigger event without processor

*EE712 Embedded Systems, presentation from WEL Lab intervention. All of the sequencers are identical in implementation except for the number of samples that can be captured and the depth of the FIFO.

2. **Trigger Source** ADC can be triggered by various sources like processor, external GPIO, PWM, internal comparators, timer etc.

3. **Relative Priority** There are 4 sample sequencers. Hence, we need to specify the relative priority of a sequencer in case we are using multiple sample sequencers.

4. **GPIO Pins** There are 12 analog input channels shared by two Analog-to-Digital Converter modules. Certain GPIO pins can act as input channels for these ADC modules. For information about these pins, refer to section 12, pages 799 - 801 of this **link**. Use GPIOPinTypeADC() function to configure these pins.Refer to page 266 of this **link** detailed description of this function.

Detailed description of theory and code examples are available in the following links:

1. Reference material 1: For ADC12 please go through **Resources7-Chapter 5**.

2. Reference material 2:For UART please go through **Resources7-Chapter 12**.

Joystick Graphical User Interface:
You can use any open source programming language to build your GUI. Below, we have listed down some pointers to assist you, if you choose python

1. The first step is to read serial data coming on your COM port. Use pyserial library to do so.

2. If you notice carefully, GUI only requires five distinct values, you can avoid sending other values in your embedded C code. Also the rate at which Joystick data is sampled can also be reduced.

3. Once you start receiving your required data, you can use matplotlib and drawnow library to get a real time Graphical representation. There are several other libraries that can be used to get a real time representation, you are free to use any of them.

5 Procedure

1. For ADC and UART code :
   Include ”adc.h” and ”uart.h” header files.
   Enable ADC0 and ADC1 peripherals. Enable and configure the UART pins. Set the required Baud rate.
   Wait till ADC conversion is complete. Average the data obtained. This is then converted according to requirements. Send the ASCII value of this data to PC using UART.

2. To create GUI
   Import serial library, matplotlib, numpy.
   Read the serial data.
   Enable the interactive plot using plt.ion() function.
   Split the received co-ordinates at comma and convert it into float.
   Plot these co-ordinates using ”drawnow()” function.

6 Demo and Submissions

You have to shoot two individual videos demonstrating the output of the problem statement. Your codes for each of the problem statement has to be uploaded in Github repository.