ELEC240 Lab7

Analogue-to-Digital Conversion (ADC) and Digital-to-Analogue Conversion (DAC) on the STM32F429 Nucleo-144 Development Board

1 Introduction

In this lab you will configure the on-board ADC and DAC modules to sample an analogue signal and reproduce it using the DAC. Refer to Table 2 for information.

1.1 Learning Outcomes

- By the end of this lab exercise you should be able to:
  1. Demonstrate an understanding of Analogue-to-Digital and Digital-to-Analogue conversion
  2. Produce code to enable and use the on-board ADC and DAC modules

2 Analogue and Digital signal Conversion

Digital systems operate using **binary** meaning **two' states** which are ***0*** and ***1***.
In order for digital signals to propagate through the real-world, which is an **analogue** domain (**infinite states**), they are converted to analogue voltages. In this state a digital ***1*** or `High' signal is represented by the pin being pulled up to the positive supply rail (VDD) and a digital ***0*** or `Low' signal is represented by the pin being pulled down to the 0V supply rail (GND).
Although not obvious, a digital output pin actually performs a Digital-to-Analogue conversion, converting binary High and Low states inside the micro-controller into real-world High and Low voltage levels. Also a digital input pin is actually performing a Analogue-to-Digital conversion, converting real-world High and Low voltage levels into binary High and Low states inside the micro-controller.
The difference between a digital input pin and an Analogue-to-Digital converter (ADC) is that instead of simply detecting whether the voltage is high or low and converting it to a binary ***0*** or ***1*** it can detect any voltage level on the pin and converts it to a binary number.

2.1 ADC Upper and Lower Limits

The ADC operates between ***VDD*** and ***GND***, meaning it can read any voltage between these levels, and convert it to a binary number which represents that level (see Table 1). An ADC of width ***b*** bits will convert a voltage on the pin of 0 ⇨VDD to a binary value of 0 ⇨val_{upper}, where

$$val_{upper} = 2^b - 1$$

2.2 ADC Resolution

The resolution of an ADC determines the amount by which the voltage on the pin must change before the ADC can detect it, given by

$$res = VDD/2^b$$

where ***b*** is the ADC width in ***bits***. When an analogue voltage is digitised by the ADC it effectively divides the ADC voltage range (0V ⇨ VDD) into ***2^b Quantisation Levels***. Each quantisation level represents the voltage difference required to change the converted value by 1.

The more bits in the ADC means more quantisation levels thus higher resolution as a much smaller voltage difference will be required to change the converted value by 1.

ADC Width (bits)	Lower Voltage Limit	Lower Numeric Value	Upper Voltage Limit	Upper Numeric Value	Resolution (VDD=3.3V)
6	0V	0	VDD	2^6-1 (63)	51.5 mV
8	0V	0	VDD	2^8-1 (255)	12.89 mV
10	0V	0	VDD	2^{10}-1 (1,023)	3.22 mV
12	0V	0	VDD	2^{12}-1 (4,095)	805.67 µV
16	0V	0	VDD	2^{16}-1 (65,535)	50.3 µV
24	0V	0	VDD	2^{24}-1 (16,777,215)	196.6 nV

Table 1: ADC limits and resolution vs width (in bits)

Task 1

a. Download and run the ADC/DAC example code from the DLE.
   This program reads the ADC and outputs the converted value straight to the DAC. Therefore, the voltage on the DAC pin should replicate the voltage on the ADC pin.

b. Identify the ADC and DAC pins using either the code or the datasheets available from Table 2. Connect a potentiometer to the ADC pin as shown in Figure 1 OR use a signal generator to input a waveform. Use the oscilloscope to verify that the ADC and DAC pins are doing the same thing.

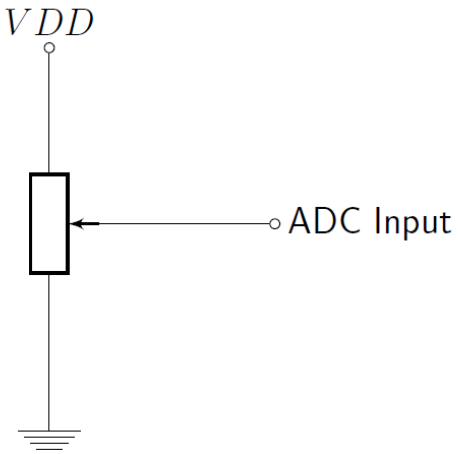

Figure 1: ADC Potentiometer Connection

Task 2

Modify the code to convert the ADC converted value to actual volts and display them on the LCD.

Task 3

Use the DAC to generate the following waveforms with a precise frequency of 500Hz

a. A sawtooth wave,
b. A triangle wave,
c. A sine wave

Hint: Refer to Lab 3

Task 4

Write code which will determine the frequency and peak amplitude of a waveform being fed into the ADC and display it on the LCD and/or PC terminal.
Hint: Using Timers here is a good idea.

Task 5

Modify the code from the previous task to reproduce the digitised waveform using ASCII characters on the PC terminal.

Task 6

Connect the Light Dependant Resistor (**LDR**) to the ADC input pin as shown in Figure 2 and produce code that will turn the three external LEDs on in the sequence ***RED, YELLOW, GREEN*** as the light level drops.
Hint: You will need to measure the resistance of the **LDR** under various light levels and apply Ohms Law to determine an appropriate value for the resistor ***R***.

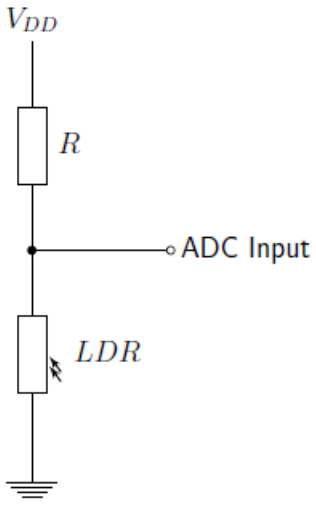

Figure 2: LDR arrangement

3 Support Documentation

Document Name	Contained Information
UM1974 User manual	• Pin identification and the supported special functions • Circuit schematics • Jumper and component identification • Header pinouts
RM0090 Reference manual	• MCU memory and peripherals architecture • Peripheral control registers, addresses and bit-fields

Table 2: Table of relevant support documentation for Nucleo-144 development boards (The document names are hyperlinks, please click on them to access the documents)