Lab Sheet 5: Analogue Input

See deadlines on QMPlus

1 Aims

The aim of this lab is to explore the use the Analog to Digital Convertor (ADC). When you have completed this lab you should:

- Know how to use an ADC in a program.
- Understand the principles of configuring the ADC.

1.1 Overview of Activities

Perform the following activities and answer the questions given in section 6.

1. Activity 1: Download, read, compile and run the sample program
2. Activity 2: Use the ADC with a voltage divider
3. Activity 3: Multiple measurements
4. Activity 4: Differential measurement.

You must also answer the questions (see Section 6) on the QMPlus quiz and demonstrate your program in the scheduled lab time. *You are recommended to look at the questions as you go along: some are about the given code and answering them will help you complete the rest of the lab.*

1.2 Other Information

Additional information is in the following appendices:

- Appendix A: has notes on the ADC operation.
- Appendix B: recaps the operation of a voltage divider.

2 Activity 1: Download, Read, Compile and Run the Sample Project

An example project is available on the course QMPlus web page as a zip file. The sample project:

- Contains code to initialise the ADC and use it in two modes:
  - Single-ended mode
  - Differential mode.
- Implements a simple cyclic system to take a reading with the ADC when the button is pressed, using the single-ended mode.

Review the provide code carefully and connect the button (see lab 2). You may need to review the notes in the Appendices. Note that the code is divided into 3 files, plus headers:

1. main.c – contains the code to initialise GPIOs[1] and for the cyclic system
2. adc.c – contains 4 functions for using the ADC
3. SysTick.c – contains 2 functions for using the SysTick counter.

[1] It would be better if the GPIO initialization were moved to a separate file. Try it!

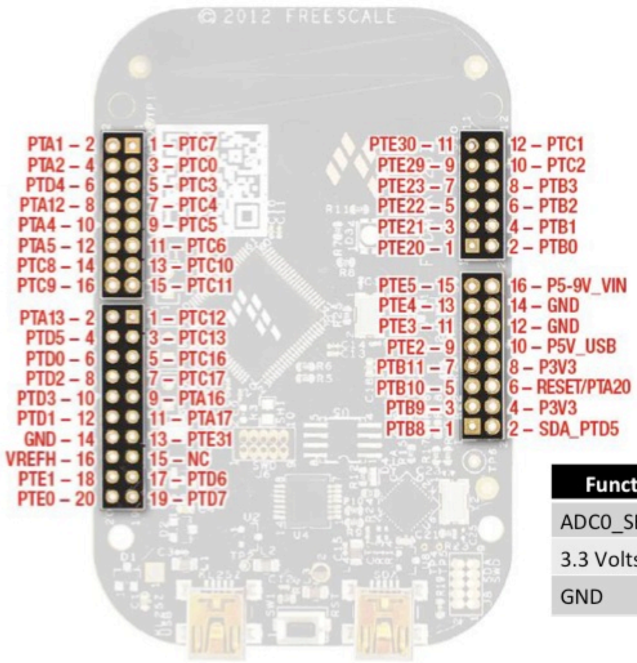
3 Activity 2: Use the ADC with a Voltage Divider

This section explains how you can use an ADC to measure the voltage between two resisters. All the cod is provided. You a required to complete a table of measurements, comparing the result of the ADC with a measurement made using the multimeter.

3.1 Connections

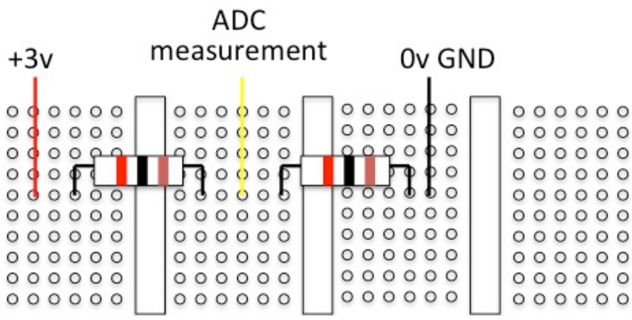
The ADC measurement is on PTB0, which on J10 pin 2. Connect this to the measured volts point of the voltage divider.

The following picture (from the user manual) shows the position of the 3.3v power supply output; there are several GND (ground) connections: one is 2 rows above the 3.3v power supply output.



Function	Pin / Header
ADC0_SE8	PTB0 / J10, 2
3.3 Volts	J9, 8 or 4
GND	J9, 12 or 14

The voltage divider is crated in the breadboard with two resistors, as follows. See Appendix B for more details, including a formula for calculating the voltage.



3.2 Using Multimeter

The meter should be set to measure voltage in the range 0-10v. **Be sure that the setting is correct before proceeding.**

Measure the supply voltage, which has a nominal value of 3.3v.

Record the measured values on the answer sheet on QMplus.

3.3 Accuracy of the ADC

Obtain the following resistors: 1KΩ, 2.2 KΩ, 4.7 KΩ and 7.5K. (If these are not available, get similar ones.) Record the nominal and measured voltages on the answer sheet.

The provided code uses the ADC to measure the voltage when the button is pressed. The measured value is a 16-bit unsigned integer. You need to convert this 16-bit integer to a voltage reading. You could implement this in the software or you could use a spreadsheet or calculator.

4 Activity 3: Multiple Measurement

When measuring real values, small variations in successive measurements are often observed. In this activity, you are required to enhance the code to take multiple measurements.

- Take a measurement on successive cycle for a fixed number of cycles (e.g. 5 to 10)
- Collect the measured values in an array. (Note that MasureVoltage writes the result to a global variable, '`sres`').
- Calculate the average value

Either in the code or otherwise (for example, in a spreadsheet), calculate a measure of the amount of variation between measurements (for example, the statistical variance could be used).

Viva Required [8 marks]

> You should demonstrate the array of measurements (using the debugger).
>
> A demonstrator will witness the viva. Ensure that both you and the demonstrator sign the viva sheet.

The code should be uploaded – see instructions on QMPlus.

5 Activity 4: Differential Measurement

The ADC can operate in many modes. One alternative is to compare two voltages and give the difference, which can either be positive or negative.

Modify the software so that:

- The difference between two voltages can be measured.
- The on-board LEDs are used to show both which voltage is larger and whether the difference is large or small. You can choose how to do this

The file `adc.c` contains a function MeasureVioltageDiff that can be used for this; the result is in the variable '`dres`'. The pins used are:

Connection	Pin	Use
ADC0_DP0	PTE20	Larger voltage (for positive result)
ADC0_DM0	PTE21	Smaller voltage

Note that if the voltage on DP0 > voltage on DM0, then the difference is positive. If it is the other way round, the difference is negative. The results is a 16 bit signed number, in the range $\pm 2^{15}$. The two voltages are in the range 0v to 3.3v (as before) but the difference can be between -3.3v and +3.3v, so that the resolution is halved compared to a single ended conversion.

Viva Required [8 marks]

Demonstrate your software working in at least three configurations a) first voltage larger b) two voltages (nearly) equal and c) second voltage larger.

A demonstrator will witness the viva. Ensure that both you and the demonstrator sign the viva sheet.

6 Question Sheet

Answers to these questions should be completed on the QMplus quiz.

<table>
<tr><th>Relates to</th><th>Questions</th></tr>
<tr><td>Section 2
Activity 1</td><td>The provided code contains a loop that is executed after the ADC is asked to take a reading. Explain:
a) What is the loop doing and when does it terminate?
b) Why is this loop needed?</td></tr>
<tr><td>Section 3
Activity 2</td><td>Copy and complete the following table of values, showing at least 8 different resistor combinations (Section 3.3):
<table>
<tr><th>R1 (Ω)</th><th>R2 (Ω)</th><th>Calculated Volts (V)</th><th>Meter reading (V)</th><th>ADC Hex Value</th><th>ADC Voltage (V)</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table></td></tr>
<tr><td>Section 4
Activity 3</td><td>When multiple measurement were made, you may have noticed some variation:
a) How did you quantify ‘the amount of variation’?
b) How much variation, if any, did you observe (using your chosen method of quantification)?</td></tr>
</table>

6.1 Question about concepts

Answer the following questions concisely:

1. A 12-bit ADC is measuring a voltage with a reference of 3.3v. Answer the following questions, explaining your working:

   a. What value will the ADC give if the voltage is 1v (assuming it is accurate)?

   b. What is the smallest difference between two voltages that can be distinguished?

2. An ADC is used to make several reading in quick succession. The time between the readings is very much shorter than the minimum time needed for a real change in the measurement (for example, the temperature of a large object changes slowly). It is observed that successive measurements vary slightly. Explain why this might occur.

[8 marks]

7 Appendix A: Principles of the Analog to Digital Converter (ADC)

An ADC is a way for the MCU to read (input) a value that varies continuously over some range. This value – for example an angular position, or a temperature – must be converted to a voltage, which is read by the ADC. The ADC works by comparing the voltage with a reference voltage. The MCU uses a reference voltage of 3.3v so the converted voltage is in the range 0 to 3.3v. In comparison, a GPIO is a digital input: any voltage over the threshold counts as '1', any value below as '0'.

7.1 Accuracy, Time and Configuration

To do the conversion, the ADC must draw some current and take some time. The electrical characteristics of the ADC, including its input impedance, are given in the MCU datasheet. We must take care that the current drawn by the ADC does not significantly change the voltage we are trying to measure, which would occur if we used very large resistors. The conversion time depends on the accuracy in bits and the speed at which the ADC operates.

The software provided configures the ADC to operate with 16-bit accuracy.

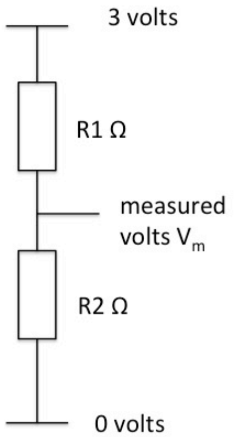
7.2 Modes of Operation

The ADC has many modes of operation.

- *What triggers the conversion?* We will use a software trigger – i.e. the conversion starts when the program sets the necessary bit; the program then waits for the conversion to complete.
- *What is measured?* We will use the ADC is two modes a) measuring the voltage on a pin relative to the supply voltage (3.3v) b) measure the difference between two voltages.

8 Appendix B: Voltage Divider

The ADC can be tested with a simple voltage divider.



$$i = \frac{3}{R1 + R2}$$

$$V_m = i \times R2 = \frac{R2}{R1 + R2} \times 3 \text{ volts}$$

If the supply voltage is V (shown as 3v in the diagram), the measured voltage should then be R2 / (R1 + R2) x V volts. The total resistance R1 + R2 should ensure that the total current is below 1 mA. Given a supply voltage of approximately 3v, this requires 3 KΩ.