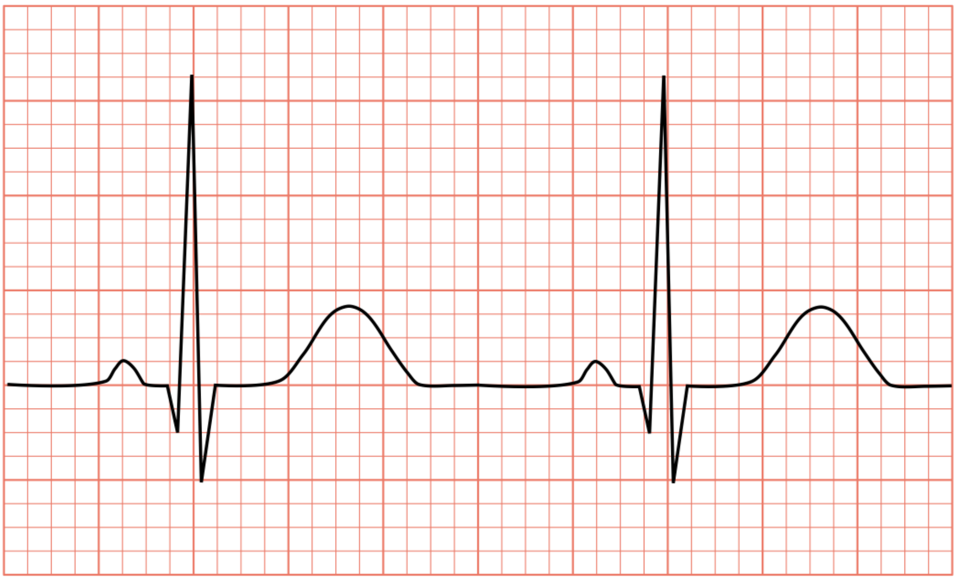
C9

Analogue Input

This exercise is an introduction to acquiring and measuring analogue signals using the `Il Matto` development board. The focus of the exercise is to learn to control the analogue-to-digital converter. You will discover how to measure voltages and then apply this to build a proximity detector using a reflective optical sensor. Finally, you will discover how to amplify very small variations in intensity from the optical sensor to convert it into a non-invasive system for measuring your heart rate, known as photoplethysmography.



Schedule

Preparation Time	:	3 hours
Lab Time	:	3 hours

Items provided

Tools	:	
Components	:	TCRT1010 Reflective Optical Sensor,
		MCP602 Dual Operational Amplifier IC,
		2×100nF capacitors, $2 \times 1\mu$F capacitors,
		100Ω resistor, $10k\Omega$ resistor, $10k\Omega$ potentiometer,
		$2 \times 6.8k\Omega$ resistors, $2 \times 68k\Omega$ resistors, $2 \times 680k\Omega$ resistors.
		[*Components to be retained by the student after the exercise.*]
Equipment	:	Oscilloscope [5]
Software	:	`avr-gcc`, `Eclipse`

Items to bring

Il Matto

c232hm cable

Identity card

Laboratory logbook

Toolkit

Version: May 12, 2013

©2012

Steve R. Gunn
Electronics and Computer Science
University of Southampton

Before entering the laboratory you should read through this document and complete the preparatory tasks detailed in section 2.

> **Academic Integrity** – *If you wish you may undertake the preparation jointly with other students. If you do so it is important that you acknowledge this fact in your logbook. Similarly, you will probably want to use sources from the internet to help answer some of the questions. Again, record any sources in your logbook.*

You will undertake the exercise working with your laboratory partner. During the exercise you should use your logbook to record your observations, which you can refer to in the future – perhaps to write a formal report on the exercise, or to remind you about the procedures. As such it should be legible, and observations should be clearly referenced to the appropriate part of the exercise. As a guide the ✎ symbol has been used to indicate a mandatory entry in your logbook. However, you should always record additional observations whenever something unexpected occurs, or when you discover something of interest.

For each Task you should create a new directory so that you have a working version of every program at the end of the lab. Remember to place comments in your code.

You will be marked using the standard laboratory marking scheme; at the beginning of the exercise one of the laboratory demonstrators will mark your preparatory work and at the end of the exercise you will be marked on your progress, understanding and logbook.

> *The last part of this exercise uses a reflective optical sensor to measure tiny variations in the intensity in your finger caused by your heart pumping blood around your body. There is no requirement to have (or marks associated with having) your heart rate measured. If you do not wish to perform this part you can ask your partner to do it, or ask one of the demonstrators to do so.*

Notation

This document uses the following conventions:

✎	An entry should be made in your logbook
`command input`	Command to be entered at the command line

1 Introduction

This lab introduces you to acquiring analogue signals with an embedded platform.

1.1 Outcomes

At the end of the exercise you should be able to:

- Configure the ADC to sample from a chosen input.
- Capture a single measurement from the chosen input channel.
- Configure the ADC to capture multiple samples at a specified sampling rate.
- Apply processing to the acquired samples to extract the signal of interest.

2 Preparation

> *The `Il Matto` board has the analogue reference (`AREF`) tied to* VCC*, so all measurements should be taken with `REFS1` and `REFS0` set to zero in the `ADMUX` register. You may also assume that the `ADLAR` bit is set to zero for this exercise.*

Study and familiarise yourself with the provided `C` source code files. Answer the following questions with reference to the `Il Matto` board running at 3.3V:

1. What is the resolution of the ADC?
2. How many input channels are available for single ended input, and where can they be found?
3. How is a channel selected for single ended input?
4. If the ADC is configured for single ended input and returns a result of `0x00FF` what is the voltage that was measured?
5. What clock pre-scaler settings are supported by the ADC? What is the fastest pre-scaler setting that can be used whilst maintaining full precision?
6. How many clock cycles are required to capture a measurement on a single ended input?
7. Write a `C` statement to configure the pre-scaler setting calculated above and enable the ADC.
8. Write a `C` statement that starts a single conversion.
9. Write a `C` statement that waits until a conversion is complete.
10. What is auto triggering, and what events are supported?
11. What is free running mode and how is it enabled?

3 Laboratory Work

Open the Eclipse application on the computer you are using and when asked for the workspace you wish to use, make sure the checkbox for *use as default* is not ticked and type `H:\ELEC1201\Labs\C9`.

Remember to create a new C project (File -> New -> New project.. -> C/C++ -> C Project), for each task of the exercise and name it with the appropriate part of the exercise for easy future reference, e.g. `C9_3.1`. This time you should select "AVR Cross Target Application" -> "Empty Project" as "Project type" and add in a new C source file.

3.1 Measuring voltages

To begin, you will connect up a simple test circuit to verify correct operation of the ADC. Connect up the 10kΩ potentiometer according to figure 1. Note that the wiper should be connected to the first single ended input of the ADC (`PA0`).

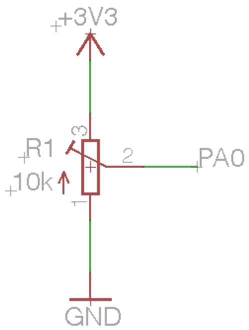


FIGURE 1: ADC test schematic

Implement the functions in the program `voltage.c` to initialise the ADC for single conversion mode on channel zero, and to capture a new measurement. Implement the main function to take a new measurement on channel zero every second and write the result to the debug UART as you did in lab C7. Write out both the raw 10-bit value and the corresponding voltage measured. In order to make `printf` work for floating point numbers[1] you will need to link with additional libraries which can be done with `-Wl,-u,vfprintf -lprintf_flt -lm`. Hence, the build command becomes

```
avr-gcc -mmcu=atmega644p -DF_CPU=12000000 -Wall -Os -Wl,-u,vfprintf -lprintf_flt -lm
    voltage.c -o voltage.elf
```

Run the program and record what range of values you are able to measure when varying the potentiometer. If you cannot quite get the full range from `0x0000` to `0x03FF` what might be the reason and how could you test it?

Set the potentiometer approximately to its mid-position. By observing the recorded values write down the minimum ($V_{\min}$) and maximum ($V_{\max}$) returned from the ADC. Calculate the noise-free code resolution of the ADC in bits which is given by $\log_2 \frac{3.3}{V_{\max}-V_{\min}}$.

[1] See http://www.avrfreaks.net/index.php?name=PNphpBB2&file=viewtopic&t=56098

3.2 Building a proximity detector

In this section you will build a proximity detector using the TCRT1010 reflective optical sensor [6]. The device incorporates an infra-red emitting diode and a phototransistor for detection. The two components are shielded from each other and only when an object is brought into the immediate vicinity will light from the diode be reflected back to the phototransistor changing the current flowing though it. Construct the circuit shown in figure 2 on the `Il Matto` breadboard. Take care to place the sensor near an outer corner of the breadboard so that you can place your finger on it without obstruction from other components. You will need to carefully spread pins 1 and 4 of the sensor to enable it to fit into four adjacent holes in the breadboard.

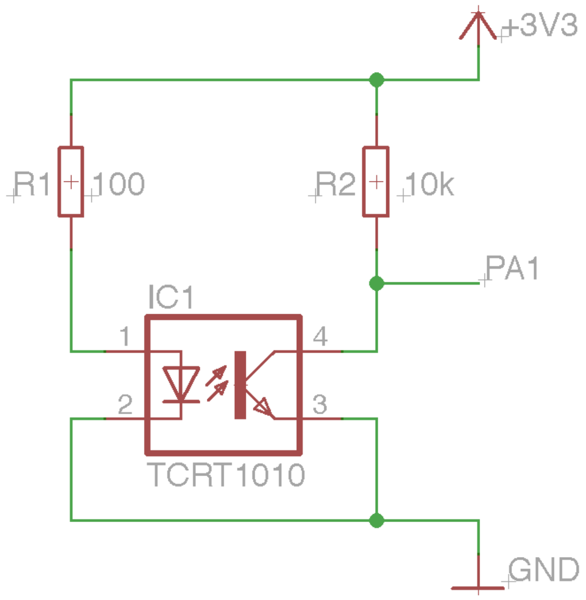


FIGURE 2: Proximity detector schematic

Connect `PA1` to the oscilloscope and observe how the voltage level changes as you move your finger nearer to the sensor and then away from it. Determine suitable voltage thresholds $V_p < V_a$ such that when $V > V_a$ you are confident that there is nothing in proximity to the sensor, and when $V < V_p$ you are confident that there is an object in proximity to the sensor. Your program can now operate using hysteresis, by only performing an action when one of these thresholds is true.

Take a copy of your `voltage.c` program and add a new function `void channel_adc(uint8_t n)` to select channel `n` for measurement. Modify the main program to select the correct channel for measurement and to illuminate the LED (`PB7`) when an object is detected in close proximity. The LED should be turned off when there is no object in proximity. Verify correct operation at a 5Hz sampling frequency.

3.3 Measuring your heart rate

So far you have investigated the DC behaviour of the reflective optical sensor. It transpires that very small fluctuations are present in the proximity detector signal when your finger is placed on the sensor. These fluctuations are due to the reflectance changing as the blood is pumped around your finger. In this part of the exercise you will build a band-pass filter [1] to amplify these very small changes in the frequencies of interest enabling you to build a device which can measure your heart rate.

The schematic for the band-pass filter you will be using is shown in figure 3 along with its frequency response. The circuit is composed of a passive high-pass filter (`C1`, `R3`) and an active low-pass filter (`R4`, `R5`, `C2`, `IC2A` [4]). The filter has a voltage gain of around 50 at the centre frequency (2.3Hz). Construct the circuit on the `Il Matto` breadboard and connect the output from the phototransistor (`PA1`) to the filter input (`FIN`). Do not forget to connect pins 8 and 4 of IC2 to VCC and GND respectively[2].

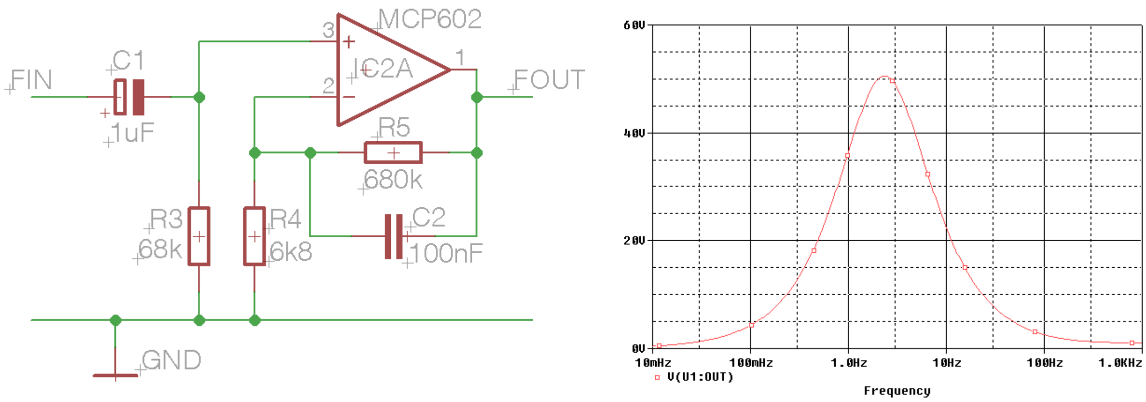


FIGURE 3: Band-pass filter schematic and frequency response plot

Connect the output from the filter (`FOUT`) to `PA2` and also to the oscilloscope. Place your index finger lightly over the sensor and observe the waveform on the oscilloscope. The quality of the signal will vary between individuals and will depend upon their physiology. Warm fingers will probably provide a better signal. Wait a few seconds before assessing the output. If there is no signal try varying the position and pressure of your finger slightly. If all goes well you should see a signal with a peak of around 1 volt varying at a frequency between 1-2Hz (figure 4). If not, ask one of the demonstrators to take a look at your circuit. You have been supplied with enough components to build two stages of the bandpass filter which can be cascaded to provide you with greater sensitivity if you think this is necessary.

Take a copy of the program that you wrote in section 3.2 and modify it to calculate the time interval between successive peaks. You can do this by identifying the first time the rising edges pass your

[2] In normal operation, operational amplifiers require a positive and negative supply, or a virtual ground at VCC/2, to accommodate bipolar signals. Here however, we only require the positive going pulse and hence we may use a single supply from the `Il Matto` board.

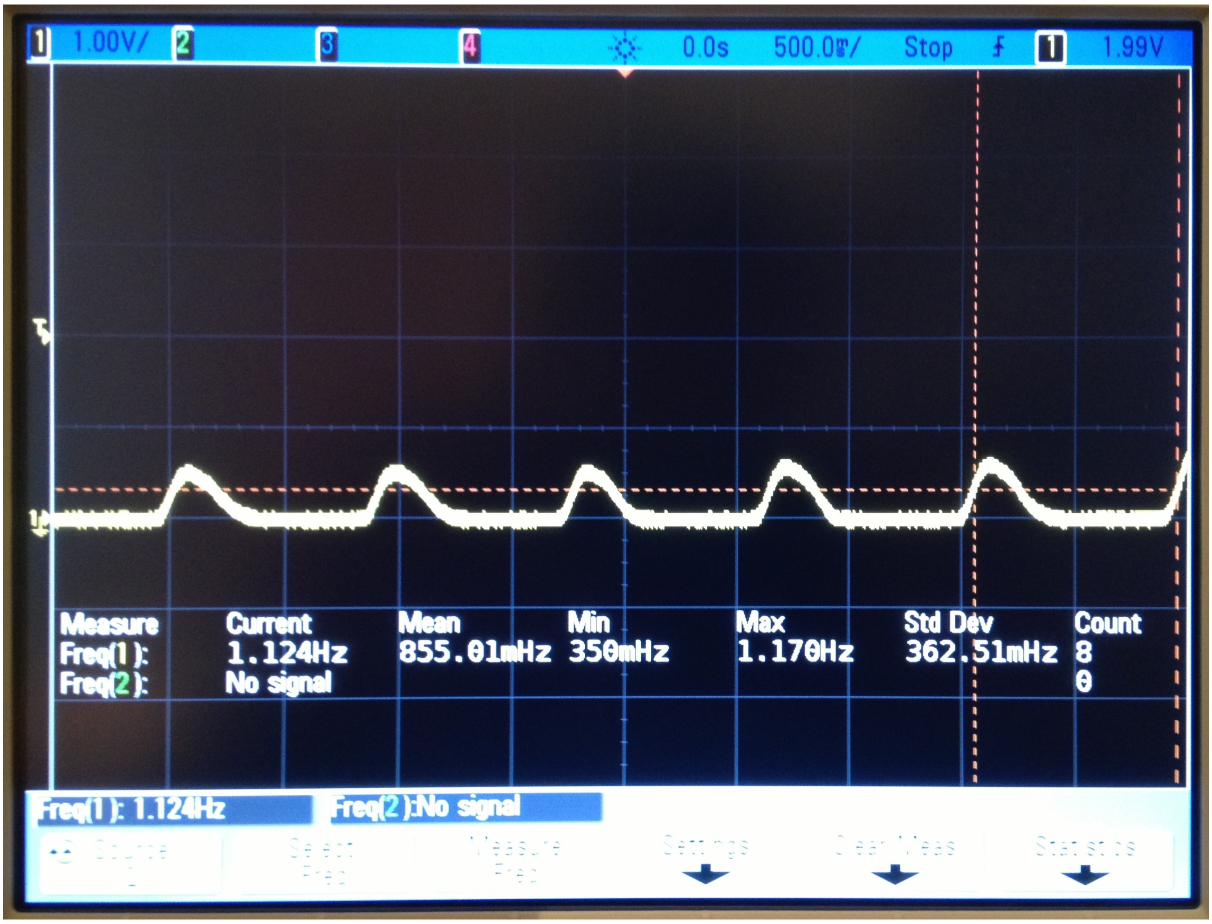


FIGURE 4: Heart beat signal from band-pass filter

threshold. You will need to change the thresholds from those that you used in section 3.2. Output the corresponding frequency in beats per minute (BPM) on the debug UART.

Modify your program to illuminate the LED on PB7 for 0.1s every time a pulse is detected.

4 Optional Additional Work

Modify your heart rate measuring program to sense when a finger is placed in the vicinity using the code you developed in section 3.2 and only generate the BPM output when this occurs. You will need to dynamically switch between ADC channels to implement this.

References

[1] Rajendra Bhatt. Heart rate measurement from fingertip, 2011. URL http://embedded-lab.com/blog/?p=1671.

[2] Atmel Corporation. ATMEGA164PA/324PA/644PA/1284P Datasheet. Datasheet 8152G-AVR-11/09, 2011. URL http://www.atmel.com/Images/doc8152.pdf.

[3] Madhero88. Normal ECG Picture, 2012. URL https://commons.wikimedia.org/w/index.php?title=File:ECGbasic.svg.

[4] Microchip. MCP602 Operational Amplifier Datasheet, 2012. URL http://www.microchip.com/S/?q=DS21314G.

[5] Tektronix. Digital Storage Oscilloscope (TDS1000C). User Manual Rev. A, 2006. URL https://secure.ecs.soton.ac.uk/notes/ellabs/reference/equipment/std-bench/Tektronix_TDS1000C_2000C_User_Manual.pdf.

[6] Vishay. TCRT1010 Reflective Optical Sensor with Transistor Output Datasheet, 2012. URL http://www.vishay.com/docs/83752/tcrt1000.pdf.