CG2271 Real Time Operating Systems

Lab 3 – Analog Inputs

1. Introduction

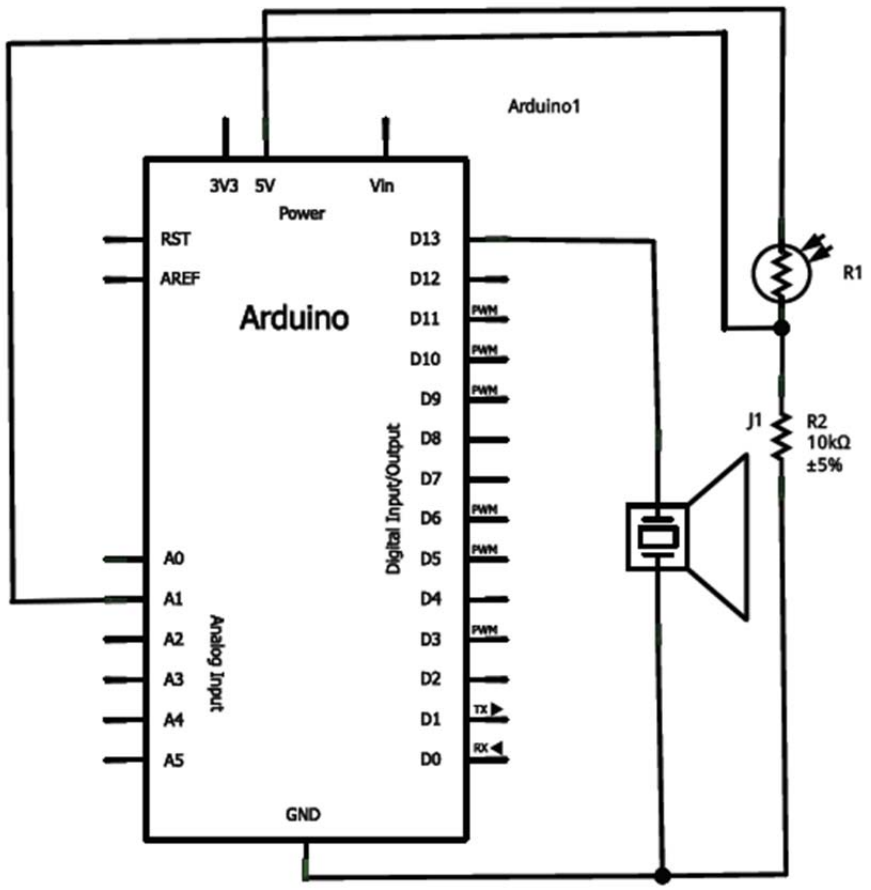
In this lab we will look at programming the analog to digital converters (ADCs) on the Atmega328. There are simple functions available in the Arduino library for doing this, but as in Lab 2 we will program the ADCs "bare-metal", i.e. by accessing the hardware registers directly. You will not receive any credits if you use any Arduino function calls.

2. Submission Instructions

You will submit a PRINTED HARD COPY of your answer book to your lab tutor at the next lab sesion in the week of 26 September 2011 (One week break starts on 19 September). You will also demo your completed assignment to your lab tutor in the first 20 minutes of the following lab session, so please be punctual or you will miss your demo slot.

3. Questions

Assemble the following circuit and answer the questions below:



The parts you require are:

Part Label	Part	Quantity
R1	Photoresistor 	1
R2	Resistor 10k (brown-black-orange)	1
J1	Piezo-electric buzzer 	1
	Arduino Uno Board	1
	Breadboard	1
	Wires	As required

In a nutshell, connect R1 and R2 to form a voltage divider with a centre tap connecting to pin A1 (analog input 1), and connect the buzzer between digital pin 13 and ground.

We will now write a program that varies the tone of the buzzer depending on how much light falls on the photoresistor.

The programming portion consists of two parts. In the first part we will explore polling the ADC to see when to read in the results of the conversion, and in the second part we will use an interrupt based approach.

Create a new project called "lab3prog1", and insert a file called "lab3prog1.c", with all the proper pre-processor directives, then answer the following questions.

Question 1 (3 marks)

The photo-resistor R1 has a resistance range of 1.5 to 6.5 ohms. The voltage going to analog input 1 (A1) is given by:

$$V = \frac{R2}{(R1 + R2)} \times 5$$

What are the minimum and maximum voltages that A1 would expect to see?

Question 2 (4 marks)

The ADC produces a value of 0 when the input is 0v, and 1023 when the input is 5v. Given the circuit above, what are the minimum and maximum values that we can expect from the ADC?

Question 3 (4 marks)

Given your answer to question 2, write a function called "remap" that takes in the value read in from the ADC and returns a value from between 0 and 255, using the formula:

$$val = \frac{input - \min\_val}{\max\_val - \min\_val} \times 255$$

Here min_val and max_val are the minimum and maximum values you calculated from question 2. Use the following prototype, and cut and paste the code into your answer book.

```
int remap(int val, int min_val, int max_val);
```

Since the ADC may not perfectly give values between min_val and max_val as found in Question 2, make sure that remap returns a 0 if *val* is negative and 255 if *val* is more than 255.

Question 4 (6 marks)

We will now create one-second tones of between 100 Hz and 500 Hz. To produce a 100 Hz tone, we need to generate a square wave 100 times in one second. I.e. we need to write a "1" to the buzzer for 5 ms, then a "0" for 5 ms, and repeat this 100 times to produce a 1 second tone. Fill in the blanks in the following function to generate a 100 Hz tone when the input parameter *input* is 0, and a 500 Hz tone when the input is 255, and intermediate tones for other values. You may add any other necessary code not shown here to make tone work.

```
void  tone(int input)
{
        int i;
        … Other necessary code…
        for(i=0; i<_____; i++)
        {
                // Write a 1 to digital pin 13

                _____
                _delay_ms(_____);

                // Write a 0 to digital pin 13

                _____
                _delay_ms(_____);
        }
}
```

Implement this function in lab3prog1.c.

Question 5 (10 marks)

Now fully implement lab3prog1.c. The steps you need to take include:

- Setting up pin 13 for output.
- Setting up ADC channel 1
- Polling the ADC to see if it has completed conversion.
- Reading the converted value.
- Producing a tone of between 100 and 500 Hz depending on the value read.

Cut and paste your code into the answer book.

Now create a new project called lab3prog2, and insert a new C file called lab3prog2.c. Ensure you include all the necessary headers and set up the project properly. Then answer the question below:

Question 6 (10 marks)

Cut and paste your program lab3prog1.c into lab3prog2.c, then modify it to use interrupts instead of polling. Detail the changes you have made in your answer book.

You will demo both parts of your assignment to your lab tutor at the start of the next lab session. Please be punctual.

4. Summary

In this lab you implemented two programs to read analog data from a photoresistor, the first using polling I/O, and the second using interrupt drive I/O.