Exercise 9: Analog Signal Acquisition

Overview

Internally, microcontrollers work with digital signals only. However, in many cases the physical input and output signals require to be analog, e.g., a voltage corresponding to temperature, sound or force. For this reason, converters are needed to transform the physical valued into a digital form. An *analog-to-digital converter* (ADC) is a device which converts continuous (analog) signals to discrete (digital) numbers. The reverse operation is performed by *digital-to-analog converters* (DAC).

Figure 1 shows a typical sensor-actuator scenario, in which ADCs and DACs serve as an interface between the microcontroller and the real world. Signals from sensors are first sampled to digital values, from which the response is calculated by the microcontroller. This response signal is then converted to proper analog signals to drive the actuators, e.g., motors or speakers.

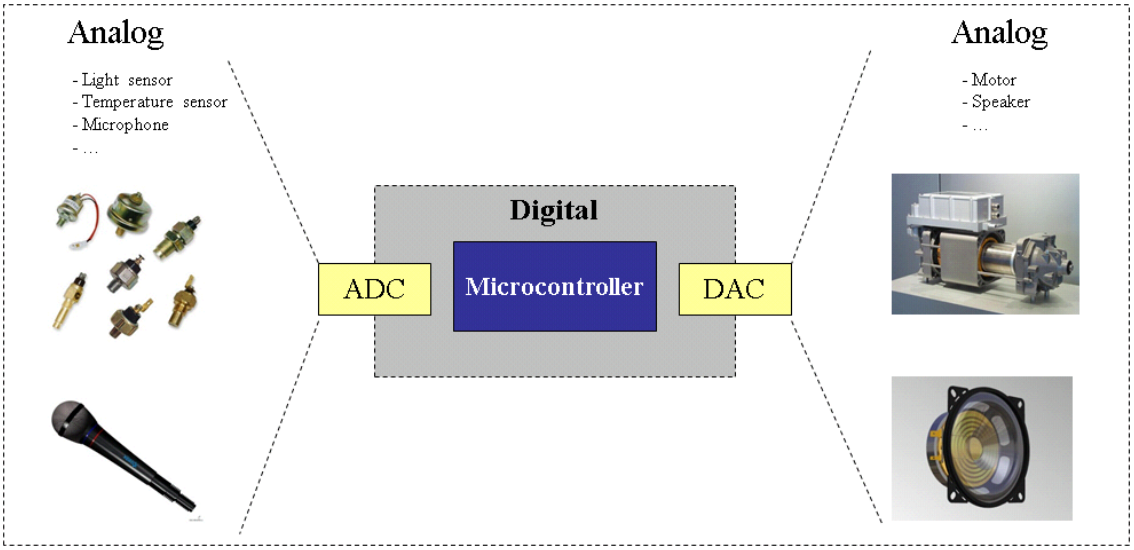

Figure 1: Sensor Actuator Scenario

When transforming continuous values to a discrete domain, the number of bits used to encode the digital values determines the *resolution* of the result. Using a finite number of bits, only a certain number of discrete levels can be represented.

Assume we use a 12-bit ADC to convert a voltage between 0 V and 10 V into a digital value:

$$\begin{aligned} \text{ADC resolution is 12 bits} \quad &\Rightarrow \quad 2^{12} &&= \text{4096 quantization levels (codes)} \\ \text{ADC voltage resolution is} \quad V_{\text{res}} &= (10\,\text{V} - 0\,\text{V})/4096 \text{ codes} &&= 10\,\text{V}/4096 \text{ codes} \\ &&&\approx 0.00244\,\text{V/code} \\ &&&\approx 2.44\,\text{mV/code} \end{aligned}$$

Assume that $d_1 =$ `0x000` is used to represent 0 mV and $d_2 =$ `0x001` is used to represent 2.44 mV, then all the values in between 0 mV and 2.44 mV have to be mapped to either d_1 or d_2. A typical solution is to map voltages between 0 mV and 2.44 mV/2 = 1.22 mV to d_1 and to map voltages between 1.22 mV and 2.44 mV to d_2. In this example, a worst-case *quantization error* of $V_{\text{err}} =$ 1.22 mV is introduced when the analog value is 1.22 mV. To visualize the difference, Figure 2 shows the original analog signal (in gray) and the corresponding digital signal sampled in a time-discrete manner (red).

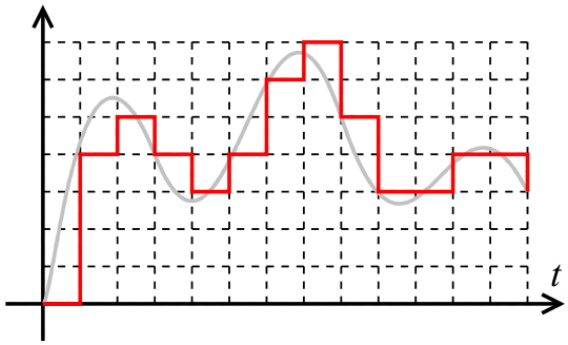

Figure 2: Analog Signal (gray) and Time-Discrete Sampled Digital Signal (red)

On the other hand, a DAC converts a finite-precision number into a concrete physical quantity (e.g., voltage). A DAC may convert the abstract numbers into a concrete sequence of impulses that are then processed by a reconstruction filter using some form of interpolation to fill the gaps in between the impulses. A low-pass filter can then be attached after the DAC to improve the signal quality. Alternatively, a network of resistors may be used of which only a subset is connected depending on the abstract input value.

Analog-to-Digital Conversion and Analog Comparator in ATmega168

Exercise 9.1

a) Read the ATmega168 documentation on the analog comparator and answer the following questions:

- **i)** How does it work (1 sentence)?
- **ii)** Which concrete pins are involved and what is their role?
- **iii)** How is a certain channel selected?
- **iv)** Which types of interrupts can be generated and when will they fire?

b) Read the ATmega168 documentation on the analog-to-digital converter and answer the following questions:

- **i)** How does it work (1 sentence)?
- **ii)** What is the maximum resolution and what is the range of digital values?
- **iii)** Which concrete pins are involved and what is their role?
- **iv)** How is a certain channel selected?
- **v)** Which types of interrupts can be generated and when will they fire?
- **vi)** Which additional measures should be taken to ensure maximum quantization accuracy?

c) What is the maximum quantization error of the ADC in ATmega168 given a reference voltage of 5 V and an input voltage range of 0 V to 5 V?

For the following exercises, you are given a *potentiometer* (compare Figure 3). From an electrical point of view, a potentiometer is a variable resistor. As you might know, resistors have the property that they cause a certain voltage drop when used in an electrical circuit. The voltage drop depends on the electrical resistance (in Ohms, Ω). By adjusting the resistance of a potentiometer, we can

control the voltage. In general, higher resistance values will cause a higher voltage drop and hence a lower voltage.

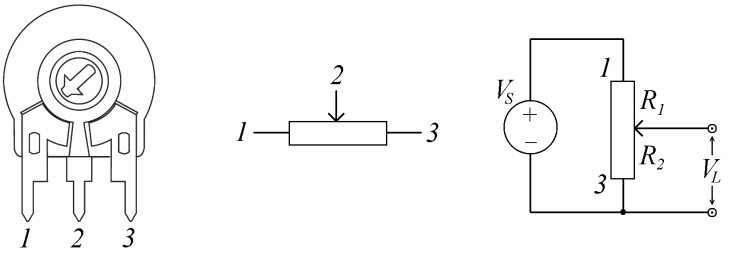

Figure 3: A Potentiometer, its electronic Symbol and typical Usage as Voltage Divider

A potentiometer is characterized by the total resistance value and its linearity. Our potentiometer has a total resistance of $R = 10\,\text{k}\Omega$ (10,000 Ω) and has a linear scale (in contrast to logarithmic).

A typical use case of a potentiometer is to use it as a voltage divider by attaching a well-known source voltage V_S to both ends of the potentiometer (pin 1 and 3 in Figure 3) and retrieving the adjusted voltage V_L between ground and pin 2. This means that the total resistance of the potentiometer is divided into two distinct resistances (e.g., R_1 between pin 1 and 2 and R_2 between pin 2 and pin 3 with $R = R_1 + R_2$). In this case, V_L can be calculated with the following equation:

$$V_L \quad = \quad \frac{R_2}{R} \cdot V_S \quad = \quad \frac{R_2}{R_1 + R_2} \cdot V_S$$

Exercise 9.2

a) What is the minimum and maximum value for V_L we can obtain in the described setup?

b) Implement a program that will continuously sample the voltage adjusted by the potentiometer according to the specified setup. Document which potentiometer pins are connected to which pins on STK500. You can adjust the `AREF` voltage in the STK500 programming dialog at the *HW Settings* tab (make sure the `AREF` jumper on STK500 is in place).

Exercise 9.3

For the following exercises, keep the potentiometer attached and connect the motor board known from Exercise 5 to STK500 in the following configuration:

- Connect the power supply pins (`GND`, `+5V`) to the respective pins on STK500 (`GND`, `VTG`).
- Connect the `Verstärker OUT` output to the `AIN0` pin of ATmega168 *and* a free analog input channel pin of your choice.
- Make sure jumper `JP2` is set to position `2-3` so that the photo transistor is selected.

a) Implement a program that displays the voltage from the photo transistor *and* the potentiometer on the debug console or the LCD.

b) Implement an application that continuously shows both voltages *and* compares the signal from the photo transistor on the motor board to a threshold voltage set up by the potentiometer using the analog comparator. Raise an interrupt whenever the voltage value from the photo transistor becomes higher or lower than the threshold value and show the current status on the debug console or the LCD.