Institute for Cognitive Systems
Technische Universität München
Prof. Gordon Cheng

Humanoid Sensors and Actuators - Tutorial 2

Microcontroller & Electronic Basics (128 points)

In this tutorial we learn:

- How to use the ADC
- How to create and simulate circuits in LTSpice
- How to simulate and calculate filter circuits
- How to use Operation Amplifier circuits
- How to design and use a Wheatstone Bridge

Setup

If you want to use your own PC for this tutorial please follow the instructions of
`http://ics.ei.tum.de/~flo/hsa-lecture/Tutorials/T2/Setup/setup.txt`

1. The ADC of the Microcontroller (50 points)

- Download the template project for micro-controller C programs
  `http://ics.ei.tum.de/~flo/hsa-lecture/Tutorials/T1/Material/Atmega32Template.tar.gz`
- Download the template project for the PC program which uses the FtdiUart Library and creates CSV files
  `http://ics.ei.tum.de/~flo/hsa-lecture/Tutorials/T2/Material/FtdiAdcTemplate.tar.gz`
- Use the AVR UART library introduced in the 'Atmega32Uart512BitAdderTemplate' project
- Download the documentation of the micro-controller
- `http://ics.ei.tum.de/~flo/hsa-lecture/Tutorials/T1/Material/atmega32.pdf`

a) Setup the ADC block (16 points)

T1.1: Consult the documentation of the ATmega32 microcontroller and find out how you need to setup the ADC block of the microcontroller. Set the bits in the appropriate registers to acquire the following setup:

- The ADC uses the AVCC voltage (5V) and an external decoupling capacitor is connected to the AREF pin (in our case the capacitor is not strictly needed) **(2 points)**

- The ADC uses a prescaler of 2 (the clock frequency of the ADC is half of the CPU frequency) **(2 points)**
- The ADC uses the free running mode **(2 points)**
- The ADC runs in 8 bit mode **(2 points)**

T1.2: Use the AVR UART library introduced in the 'Atmega32Uart512BitAdderTemplate' project to communicate with the PC. **(2 points)**

T1.3: Implement a function to sample and read the 8bit ADC value for a specified channel using the following function declaration:

```
void adc_readBlocking(uint8_t* b, uint8_t ch);
```

- The function uses the "ch" argument and samples the correct channel **(2 points)**
- The function samples and returns the correct 8bits of the sample **(2 points)**

T1.4: Send the samples of pin ADC2 in binary format to the PC. **(2 points)**

b) Test the ADC with a potentiometer (4 points)

T1.5: Connect a potentiometer to the ADC1 pin, such that you can use the potentiometer to set voltages between 0V and 5V. Run your MCU program and capture the data with the PC program while turning the potentiometer. The PC program generates a CSV file. Save this file and plot it (e.g. in Matlab). Hand in the plot **(4 points)**

c) Measure the capacitance of a capacitor using the ADC (10 points)

- Download the template project for this task which uses the ADC library
  `http://ics.ei.tum.de/~flo/hsa-lecture/Tutorials/T2/Material/Atmega32AdcTemplate.tar.gz`

T1.6: Use pin PC1 to load/unload a 1 uF capacitor via a 1 kOhm resistor. **(2 points)**

T1.7: Use the ADC0 pin to measure the loading curve with maximum ADC resolution. To minimize any delays between samples store 1024 samples into the SRAM. **(4 points)**

T1.8: Send the 1024 samples you stored in to the SRAM to the PC using the `uart_writeBlocking()` function. **(2 points)**

T1.9: Generate a CSV file of the 1024 samples and plot it. NOTE: The CSV file and plot are needed to answer some questions (**R1.3 - R1.7**) in the report. Hand in the CSV file and your plot. **(2 points)**

d) Report (20 points)

R1.1: In **T1.1** you use the ADC with specific settings and a CPU frequency of 1 MHz. How many CPU cycles pass between the acquisitions of two ADC samples? Elaborate and explain. **(4 points)**

R1.2: What is then the sampling frequency of the ADC? **(2 points)**

R1.3: Use the data and plot of **T1.9** and measure the 3τ (τ = RC) value. Remember the capacitance is not known and will be determined using the τ value. **(2 points)**

R1.4: What ADC value did you use for measuring 3τ? Explain. **(2 points)**

R1.5: What voltage level does this ADC value represent? Explain. **(2 points)**

R1.6: Use the τ value you determined in **R1.3** and the given resistance R=1.0 kOhm and calculate the capacitance of the capacitor. **(2 points)**

R1.7: Compare the calculated capacitance of **R1.6** with the capacitance of the capacitor you used in your circuit.

- How big is the capacitance error **(1 point)**, and
- does the error lie within the tolerance of the used capacitor? **(1 point)**

R1.8: We discussed the operation of the ADC in the lecture. Let's assume that the conversion logic uses a digital counter with a DAC to generate a voltage ramp. Considering the Sample&Hold time (see datasheet) and the total conversion time of the ADC (see datasheet, **R1.1**) with the settings used in **T1.1**, what is the minimal clock frequency of the clock which drives the digital counter generating the voltage ramp? **(4 points)**

2. Getting started with LTSpice (8 points)

You find a short crash course on LTSpice at "http://denethor.wlu.ca/ltspice/". Familiarize yourself with the basic editing, simulation and evaluation capabilities of LTSpice in this first example.

T2.1: Create a new schematic in LTSpice. Add a 5V voltage source and serially connect it to a 10k Ohm resistor and a 1uF Capacitor. Hand in the circuit. **(2 points)**

T2.2: Start a DC operation point (.op) simulation and have a look at the currents and voltages. Do they make sense? Elaborate. **(2 points)**

T2.3: Start a transient simulation (.tran) and measure the voltage and current on the capacitor.

- Hand in the circuit, and the plot (screenshot) of the measured voltage and current. **(2 points)**
- Compare the calculated (τ = RC) and measured time constant (~63.2% signal reached). Specify both time constants. Do they match? **(2 points)**
- Plot the power consumption of the circuit over time (P=U*I) in LTSpice.
  Hand in the plot. **(2 points)**

3. Filter circuits (20 points)

We use the LTSpice circuit of Task 2.

T3.1: Conduct a small signal AC behavior analysis (.ac) with an amplitude of 1V from 0.01 Hz to 100 Hz. Hand in the circuit, and the plot of the phase and amplitude. **(2 points)**

Report (18 points)

R3.1: Derive the complex transfer function $H(j\omega) = \frac{U_{output}}{U_{input}}$ of the filter. Use symbolic expressions and explain step by step. **(4 points)**

R3.2: Derive the gain $G(\omega) = |H(j\omega)|$ of the filter. Use symbolic expressions and explain step by step. **(4 points)**

R3.3: Derive the phase shift $\phi(\omega) = arg(x + jy) = atan2(y, x) \approx arctan\left(\frac{y}{x}\right)$ of the filter. Use symbolic expressions and explain step by step. **(4 points)**

R3.4: Derive the -3dB cutoff frequency $f_g = \frac{\omega_g}{2\pi}$ with $G(\omega_g) = \frac{1}{\sqrt{2}}$ of the low pass filter. Use symbolic expressions and explain step by step. **(4 points)**

R3.5: Compare your results with the result of your LTSpice simulation. Elaborate and explain. **(2 points)**

4. Operational Amplifier (OpAmp) circuits (18 points)

a) Voltage divider with a resistive output load (6 points)

T4.1: Create a voltage divider LTSpice circuit with a 5V voltage source. Set both resistor values to 200k Ohm and use a 500k Ohm load resistance on the output. Hand in the circuit. **(1 point)**

T4.2: Increase the cross-current of the divider by lowering the divider resistor values. How does the load resistance change the output voltage? Explain, elaborate and hand in the circuit. **(1 point)**

T4.3: How does the output voltage error (in %) change for higher divider currents? Provide 2 examples and describe the tendency of the error. Explain and elaborate. **(2 points)**

T4.4: What is the draw-back of higher divider currents? **(2 points)**

b) Voltage divider with a resistive output load and a voltage follower (8 points)

T4.5: Add an OpAmp based voltage follower circuit between the voltage divider and the load resistor. Load the “opamp” component library using the spice directive “.lib opamp.sub” and hand in the circuit. **(2 points)**

T4.6: How does the output voltage change in comparison to the setup without the voltage follower? Explain and elaborate. **(2 points)**

T4.5: How much can the cross current of the voltage divider be reduced without affecting the desired output voltage? Explain and elaborate. **(4 points)**

c) Report (4 points)

R4.1: What are the advantages and disadvantages of very small divider currents? Elaborate and explain. **(4 points)**

5. Wheatstone Bridge (32 points)

Now let’s design a Wheatstone bridge in combination with an instrumental amplifier to measure small resistance changes of a strain gauge (e.g. 1 µm change results in a change of resistance of 57 mOhm).

a) Wheatstone Bridge without OpAmp (10 points)

T5.1: Create a new LTSpice schematic and design a Wheatstone bridge with 320 Ohm resistors and a 5V voltage source. Copy the provided strain-gauge.LIB and strain-gauge.asy files next to your schematic file. Replace the upper left resistor by the strain-gauge component which models the strain gauge. Setup the strain-gauge model for the strain-gauge symbol and add the spice directive “.lib strain-gauge.LIB”. **(2 points)**

T5.2: Set the strain gauge properties of the model to $R = 320\Omega, L = 0.32m, k = 1.76, n = 32$ where R is the resistance, L is the length, k is the gauge factor and n is the number of sensitive tracks of the strain gauge. **(2 points)**

T5.3: For simulating the change of length, connect a sine voltage source to the strain gauge part with 1µV amplitude and 5 Hz; this corresponds to length changes of 1 µm with the same frequency. Perform a transient simulation for 1s and plot the bridge voltage. Hand in the circuit and the plot. **(2 points)**

T5.4: Lower the maximum simulation time step to 10us and compare it to e.g. 100us. What is the difference? Elaborate and explain. **(2 points)**

T5.5: Discuss the simulation results. What do you conclude? **(2 points)**

b) Wheatstone Bridge with OpAmp (10 points)

T5.6: Copy the schematic of a), import the INA122 instrumental amplifier and add it to the circuit. Adjust your circuit such that the INA:

- uses a bias voltage of 2.5V, **(2 points)**
- and a gain of 10000, and **(2 points)**
- connect the INA correctly to the bridge. **(2 points)**

T5.7: Conduct a transient simulation ".tran 0 1 0 10u" with a bias of 2.5V and 0V. What is the difference? Hand in the circuit and the plots. Elaborate and explain. **(4 points)**

c) Power Line Noise Simulation (6 points)

T5.8: Copy the schematic of b) and simulate power line noise and high frequency noise from actuators. Add the following voltage sources in series on top to the 5V DC voltage source:

- a sine, 200 mV amplitude, 50 Hz
- a sine, 200 mV amplitude, 100 Hz
- a sine, 200 mV amplitude, 1kHz

Hand in the circuit. **(2 points)**

T5.9: Which noise sources of T5.8 simulate power line noise and which actuator noise. Elaborate and explain. **(2 points)**

T5.10: How do these noise sources influence the output of the OpAmp? Elaborate and explain. Hand in the plot. **(2 points)**

d) Power Line Noise Filtering (6 points)

T5.11: Copy the schematic of c) and add a LC low pass filter between the voltage sources and the circuit. Use a 10 µH inductor with an ESR of 10 Ohm and a 1000 µF capacitor with an ESR of 20 mOhm. Hand in the circuit. **(4 points)**

T5.12: How does the filter influence the output of the OpAmp? Elaborate and explain. Hand in the plot. **(2 points)**