Electronics II
exercise 8

Deadline: Thu 26.3.2026 14:00

Fast processing

Task 1 – USB oscilloscope (3p)

Make a crude USB oscilloscope. The oscilloscope should have the following functionalities:

- Sampling frequency as high as possible. Take note that sending values over a serial connection is slow. Collect the samples to a buffer and transmit them at once to get better sampling rates.
- Communication with a host PC (Serial or wireless).
- An analog trigger level can configured by host PC.
- Data is plotted on the PC in SI units (e.g., μs and mV).
- Measure and plot: maximum, minimum, mean and peak-to-peak voltages of the signal.

Arduino: Use the avdweb_AnalogReadFast library (github.com) or (arduino.cc) to get faster sampling rates.

ESP32: This task can be completed with the ESP32 as well but you'll have to figure out a way to increase sampling frequency yourself. This discussion (esp32.com) or this library (github.com) may be of use. The goal is to have a sampling frequency comparable to the Arduino library.

Tips:

You can use the Arduino IDE serial plotter before you implement plotting on the host PC.

Make a Python (other languages ok) loop which asks for user input and sends a command to the microcontroller. The microcontroller then sends data to the PC for processing and plotting. The loop then starts again and updates the plot when new data is received. It's okay to ask user input only once and just kill the code when you want to change input.

You can use the microphone, or a PWM output of another pin (attached to an RC filter), as a waveform source. You can use the other microcontroller as the source to reduce processing load. **Remember to use a voltage divider if you output a signal from the Arduino to the ESP32.**

Task 2 – Music visualizer (3p) [Arduino]

Make a music visualizer. Listen to audio through a microphone*, compute FFT and display the frequency content on the dot matrix display.

*You will need to adjust the potentiometer. The DC level of the analog microphone output should be around 4 V (for Arduino). Note that it's still not very sensitive but you should be able to pick up audio if you place it next to a speaker. For testing you can find e.g. a sine wave generator from the internet.