Exercise 0    Analog digital conversion

In the following exercise, we want to use the ADC of the AVR to convert analog values into digital values.

- Read chapter 24 in the ATMega168 manual.
- What does it mean that the AVR has a 10bit ADC?
- Why is the conversion speed important? How can you set it?
- Try to find some widely available analog sensors on the internet and discuss other possible use cases of the ADC.

Exercise 1    ADC AVR

In this exercise, we will connect a potentiometer to the AVR and measure the current voltage. For displaying the result, we will use the SPI display from the previous sheet.

- Connect the display as in the previous sheet and make sure that everything works as expected.
- Connect the potentiometer to the AVR. As shown in the image below, pin 1 of the poti should be connected to $V_TG$ and pin 3 to $GND$. Connect pin 2 of the poti to any $ADCX$ of your choice (e.g. $ADC0$ on pin $PC0$).

- In the following, we want to implement two functions. Function $adc\_init$ initializes the ADC. The second function $adc\_read( $uint8_t$ channel )$ is then used to read a value from the ADC input $channel$.
  - $adc\_init$: Use $AVCC$ as voltage reference. Further set a prescaler division factor of 2 and perform single conversion without any interrupts. Without setting any specific ADC input in the $ADMUX$ register, we need to perform an initial conversion to set everything up.
  - $adc\_read( $channel )$: Shall perform a single conversion and return the resulting value. The function shall block until the conversion is finished.
- Visualize the conversion result on the display.