UNIVERSITEIT
iYUNIVESITHI
STELLENBOSCH
UNIVERSITY

100

1918 · 2018

forward together · saam vorentoe · masiye phambili

Computer Systems / Rekenaarstelsels 245

Lecture 24

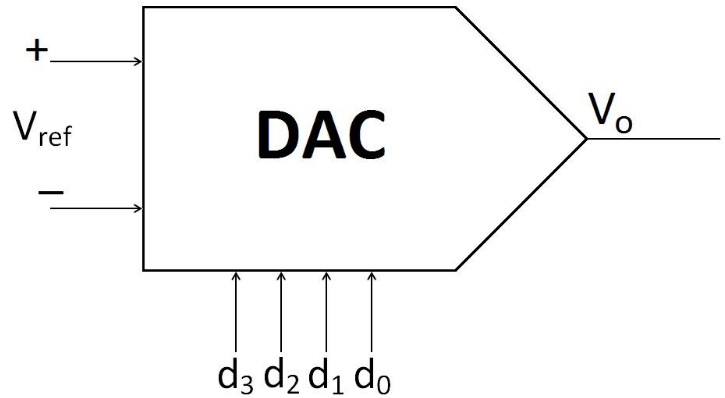
Digital-to-Analog Converter (DAC) and signal conditioning/ Digitaal-na-Analoog Omsetter en sein aanpassing

Dr Rensu Theart & Dr Lourens Visagie

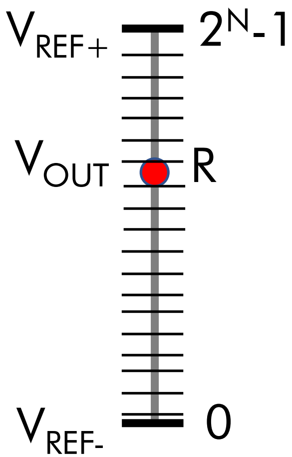
Digital-to-Analog Conversion
Digitaal-na-Analoog omskakeling

- Digital-to-Analog Conversion is the opposite process of ADC: Convert a digital value (number) to an analog voltage
- DAC is more often used to convert a stream of digital values to time-varying analog signal
- DAC applications include
  - Outputting audio signals (from digital source)
  - Software-defined Radio
  - Motor control (variable speed)
- Also has different methods to generate
  - Filtering of PWM signal
  - Resistor tree (binary weighted resistor DAC)
  - R-2R ladder

Digital-to-Analog Conversion
Digitaal-na-Analoog omskakeling



- The ideal DAC also has the same linear relationship between output analog signal and digital input:

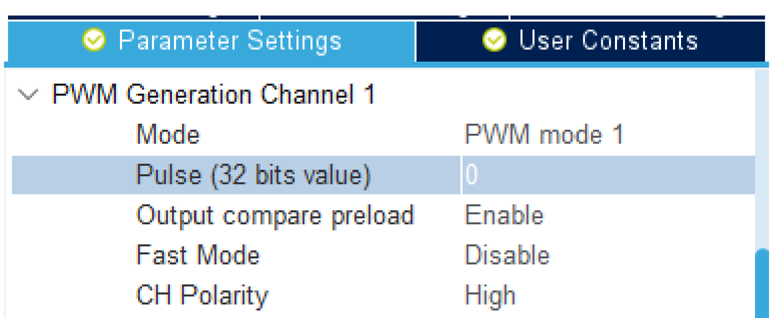


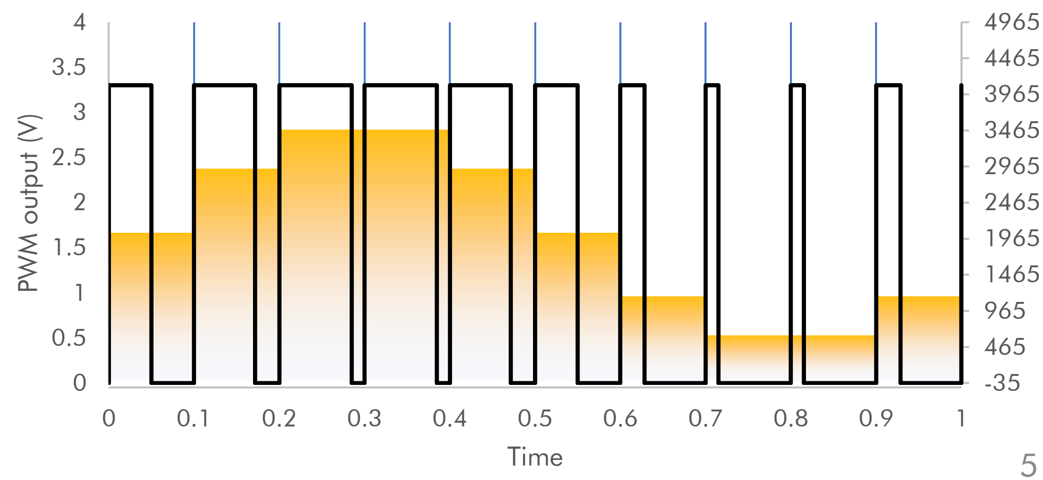
$$V_{OUT} = V_{REF-} + \frac{R(V_{REF+} - V_{REF-})}{(2^N - 1)}$$

DAC Implementations
DAC Implementerings

PWM (Pulse Width Modulation) DAC

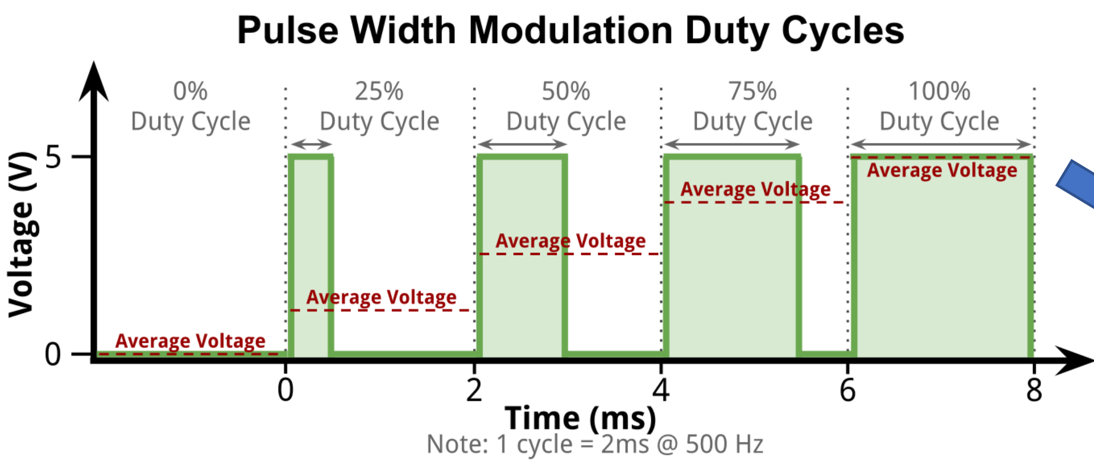
- We can create PWM signals using timer modules, remember? - Review lecture 14
- Not all microcontrollers have DAC peripheral, but most have timers with PWM capability
- To create an analog output, vary the PWM duty cycle to correspond to the desired analog value
- (PWM "pulse" register value = digital value to convert)
- Eventual analog signal will equal the average signal level over the PWM period
- For a changing analog signal, every time the timer overflows, a new value is loaded into the pulse register (use DMA)

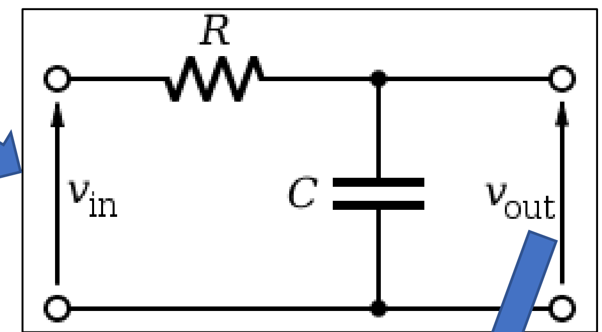


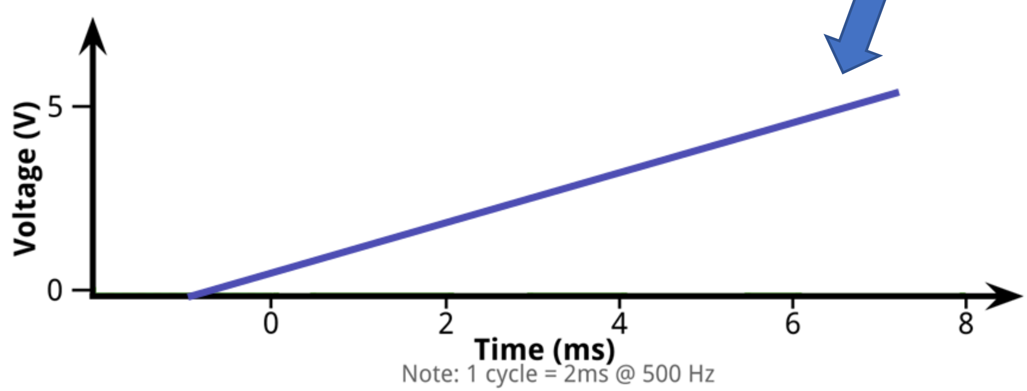


DAC Implementations
DAC Implementerings

- Filter (average) PWM signal to convert to analog signal
- Simple low-pass RC filter is ok, but ideally you need a higher order filter





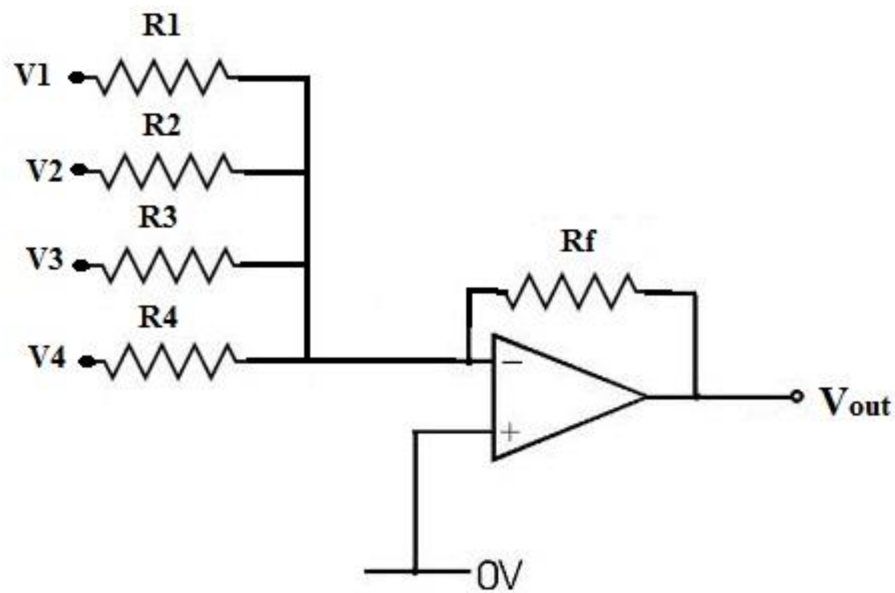


DAC Implementations
DAC Implementerings

Binary weighted resistor DAC

- Uses an inverting summing op-amp
- Inverted summing-op amp: Adds different input signals, each with a gain depending on resistor values ($A_I = R_f/R_I$)

$$V_{OUT} = -\left[\frac{R_f}{R_1}V_1 + \frac{R_f}{R_2}V_2 + \frac{R_f}{R_3}V_3 + \frac{R_f}{R_4}V_4\right]$$



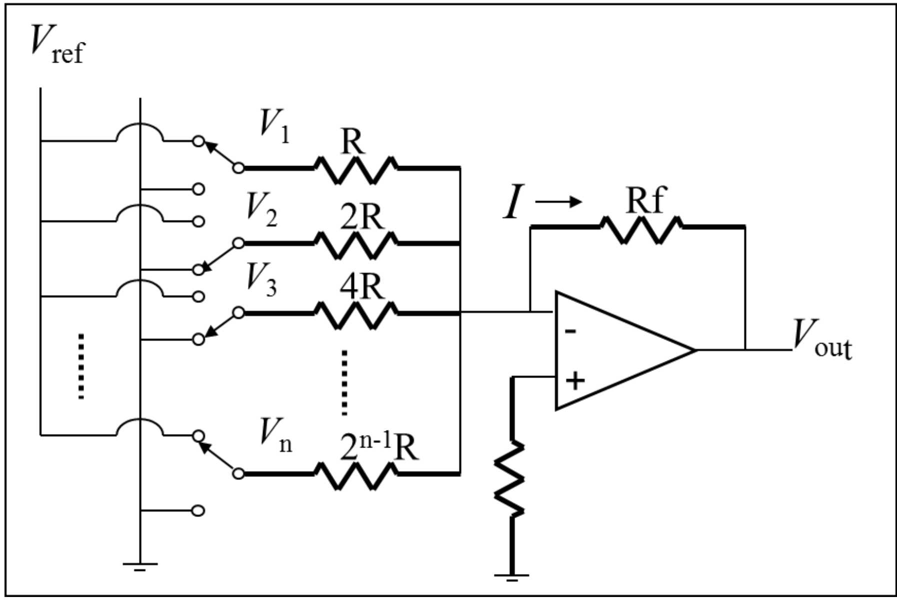
DAC Implementations
DAC Implementerings

Binary weighted resistor DAC

- Uses an inverting summing op-amp
- Each bit of binary number (that has to be converted to analog) is used to control a switch (either 0V or V_{ref})
- Resistors correspond to weight of binary digit

$$V_{out} = -IR_f = -R_f\left(\frac{V_1}{R} + \frac{V_2}{2R} + \frac{V_3}{4R} + \cdots \frac{V_n}{2^{n-1}R}\right)$$

Example:
4-bit binary DAC output: $1010_2 = 10$
V1 = 3.3V, V2 = 0V, V3 = 3.3V, V4 = 0V
Choose R_f = 0.5*R
V_{OUT} = -0.5*R(3.3V/R + 3.3V/4R)
V_{OUT} = -2.0625V
(= 3.3V * 10/16 = $-V_{ref}$*dac_val/2^N)



DAC Implementations
DAC Implementerings

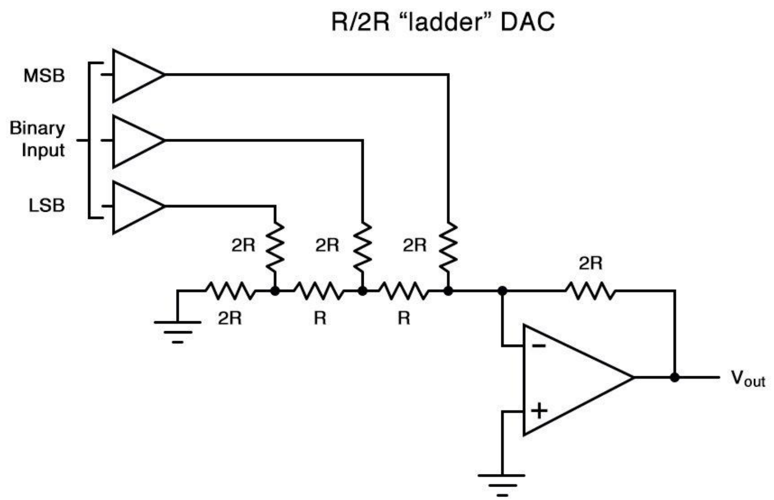
Binary weighted resistor DAC

- Drawbacks of binary weighted resistor DAC:
  - Different resistor values needed, and large range of resistance values (each resistor has double the resistance of previous one)
  - Resistors have errors, and can lead to inaccuracies

DAC Implementations
DAC Implementerings

R-2R Ladder Circuit

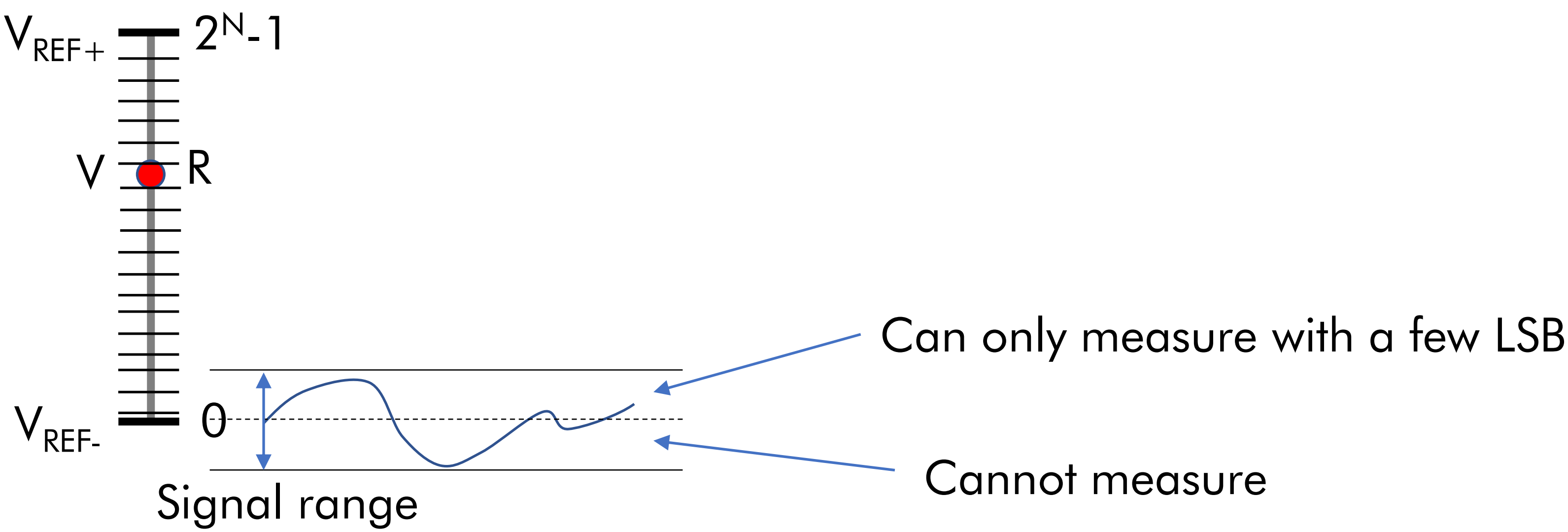
- Still uses an inverting summing op-amp, but different resistor network
- Requires only R and 2R resistors
- End effect is the same



https://www.allaboutcircuits.com/textbook/digital/chpt-13/r-2r-dac/

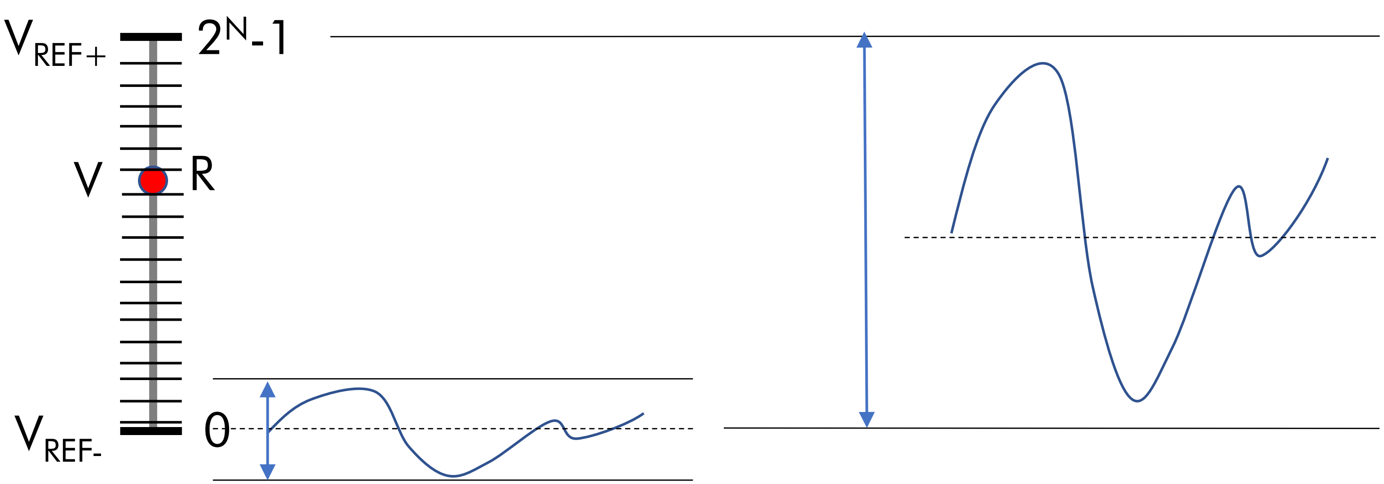
Signal Conditioning
Sein Aanpassing

- Microcontrollers typically only operate in the range 0V to VDD.
- They cannot convert negative voltages, or generate negative voltages
- Sometimes, analog signals have only a small range – using the full range ADC is not efficient and results in poor precision (and same for DAC signal generation)



Signal Conditioning
Sein Aanpassing

- Use signal conditioning to scale and shift signal into desired range
- Typically accomplished with an operational amplifier – gain and offset

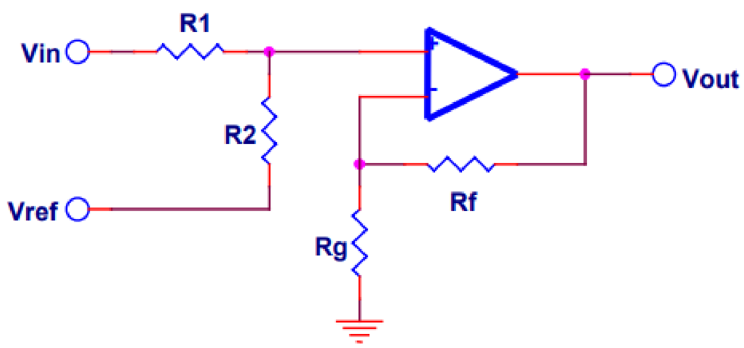


Signal range

Signal range after scaling and shifting

Signal Conditioning
Sein Aanpassing

- Texas Instruments SLOA097: Designing Gain and Offset in Thirty Seconds: http://www.ti.com/lit/pdf/sloa097
- m is the desired gain, b is the desired offset



Choose R1 = _____

Calculate $R2 = \frac{Vref \times R1 \times m}{b}$ = _____

Select Rf (may be suggested by data sheet) = _____

Calculate $Rg = \frac{R2 \times Rf}{m \times (R1 + R2) - R2}$ = _____

Figure 1. Schematic Diagram for Positive m and Positive b

- Different configurations and formula for negative gain (inverting) and negative offset

Signal Conditioning
Sein Aanpassing

- Not all op-amps are created equal
- To get a negative output voltage, the op-amp needs a negative (dual-rail) supply
- Op-amps typically become less linear the closer the output gets to the supply rail. Obviously you want to avoid that, so design the gain and offset to stay in the linear region
- Read the datasheet!
- Simulate the circuit in SPICE

ADC Calibration
ADC Kalibrasie

- We can use the linear relationship formula and sensor transfer function to determine a formula that relates sampled ADC value (R), to the quantity being measured.
- For the MCP9700 temperature sensor:

EQUATION 4-1: SENSOR TRANSFER FUNCTION

$$V_{OUT} = T_C \times T_A + V_{0°C}$$

Where:

T_A	=	Ambient Temperature
V_{OUT}	=	Sensor Output Voltage
$V_{0°C}$	=	Sensor Output Voltage at 0°C (see **DC Electrical Characteristics** table)
T_C	=	Temperature Coefficient (see **DC Electrical Characteristics** table)

$$R = (2^N - 1)\frac{(V_{IN} - V_{REF-})}{(V_{REF+} - V_{REF-})}$$

$$T_A = \frac{(V_{OUT} - V_{0°C})}{T_C} = \frac{R(V_{REF+} - V_{REF-}) + V_{REF-}}{(2^N - 1)T_C} - \frac{V_{0°C}}{T_C}$$

- V_{REF+} = 3.3V, V_{REF-} = 0V, $V_{0°C}$ = 0.5V, T_C = 0.01V/°C, N=12

$$T_A = 0.0806R - 50$$

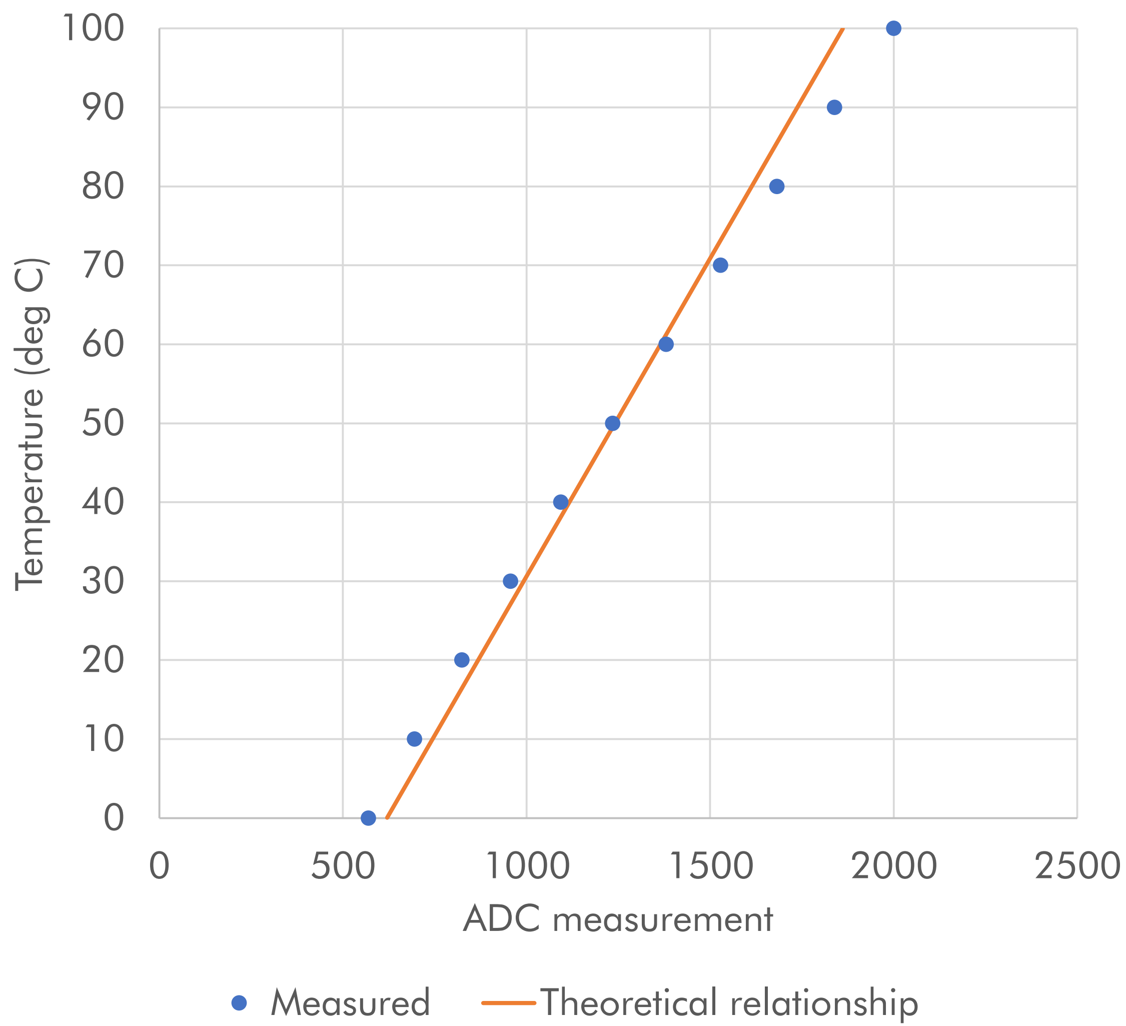
ADC Calibration
ADC Kalibrasie

- The overall measurement accuracy is affected not just by ADC error sources, but also extrinsic parameters:
  - $V_{REF} = V_{DD}$ might not be exactly 3.3V
  - Resistor values (voltage divider for instance) have tolerances
  - Non-zero offset
  - Measuring element itself might have some variation from one device to the next

- To get a more accurate measurement, you have to calibrate the measurement
- Calibration: come up with an appropriate equation and parameters to relate the sampled ADC value to the quantity being measured
- How?
- You can measure V_{ADC}, and then sample the ADC for that V_{ADC}, but this only solves half the problem (need an accurate Voltmeter!)
- Ideally, obtain a number of accurate measurements of the quantity you are measuring, and matching ADC sample – over the entire range
- Perform a "best fit" of the measurements to the model equation

ADC Calibration
ADC Kalibrasie

- Example measurements:



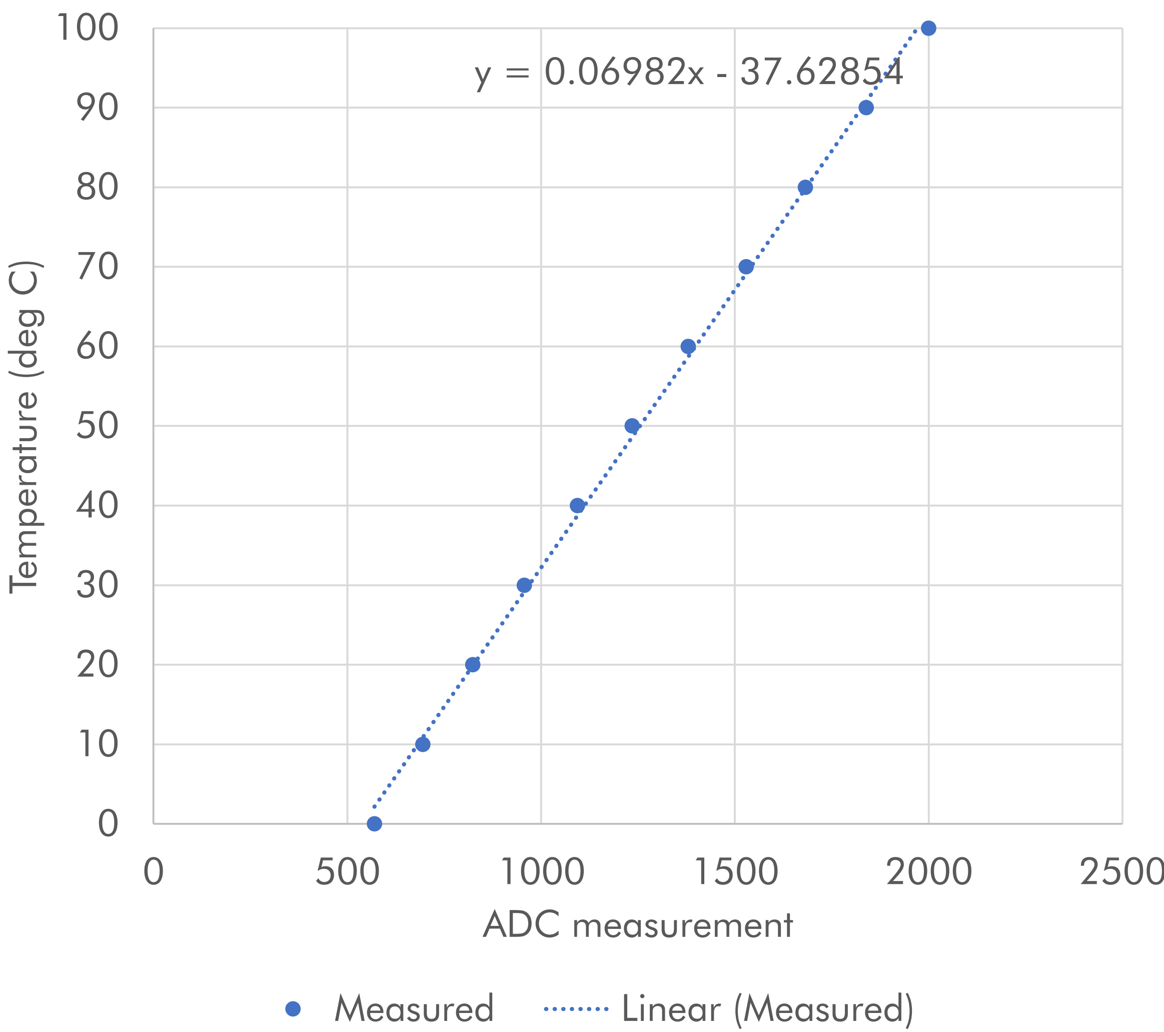
Measurement error using 'theoretical' relationship: 5.15 deg C (1-σ)

'theoretical' relationship:

$$T_A = 0.0806R - 50$$

ADC Calibration
ADC Kalibrasie

- Least squares fit to 1st order polynomial:



- After calibration error: 1.23 deg C (1-σ)
- To do better than this: Use other model equation – i.e. higher order polynomial
- In source code, use the "updated" equation instead:

$$T_A = 0.06982R - 37.62854$$

- Sometimes, the calibration equation coefficients will have to be determined uniquely for every device