04: Analog and Digital signals

Analog ↔ Digital I/O

Microcontrollers are digital

How do we read in analog input?

How do we produce or simulate analog output?

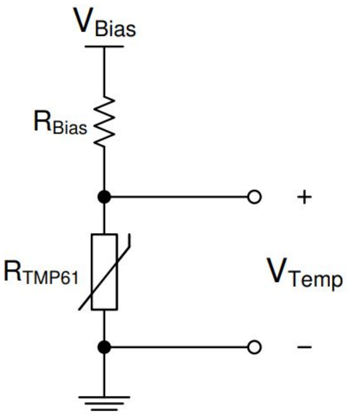
```
temp = readTemperature()
if (temp < 30) {
    … // good temperature
} else {
   … // too hot!
}
```

image source

Modeling thermistor input

Model how physical value maps to sensor value

- Physical value: temperature
- Sensor value: MCU input voltage
  - Converted to an N-bit quantity
  - Depends on [physical properties]() of the sensor and how it's wired



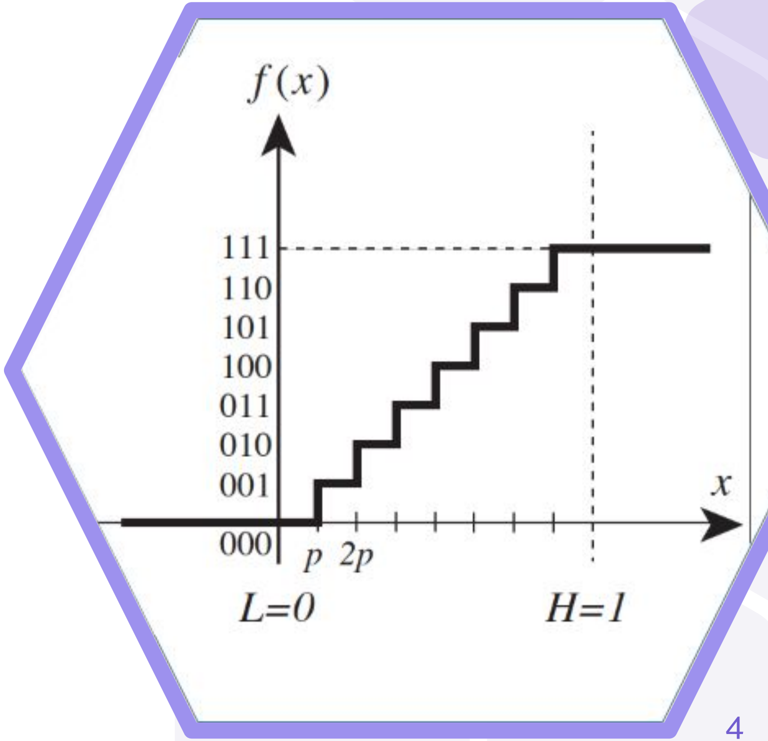
$$V_{Temp} = \frac{V_{Bias} * R_{TMP61}}{R_{Bias} + R_{TMP61}}$$

Quantization

Input to MCU is not a voltage, it is a number of some precision

For example, 10 bits precision: input is number from 0-1023

- 0V maps to 0
- 3.3V (or VCC) maps to 1023



If VCC = 3.3v and our precision is 10 bits, what is the smallest change in voltage that we can detect?

How do we find the smallest change in temperature we can detect for our example?

When doesn’t it make sense to increase the resolution of our microcontroller pin?

Linear and Affine Models

Approximate sensor value with linear affine function, saturate if outside range [L,H]

Can be done experimentally, makes for a simpler conversion

$$f(x(t)) = \begin{cases} ax(t) + b & \text{if } L \leq x(t) \leq H \\ aH + b & \text{if } x(t) > H \\ aL + b & \text{if } x(t) < L, \end{cases} \tag{7.2}$$

is this a reasonable model? photoresistor datasheet

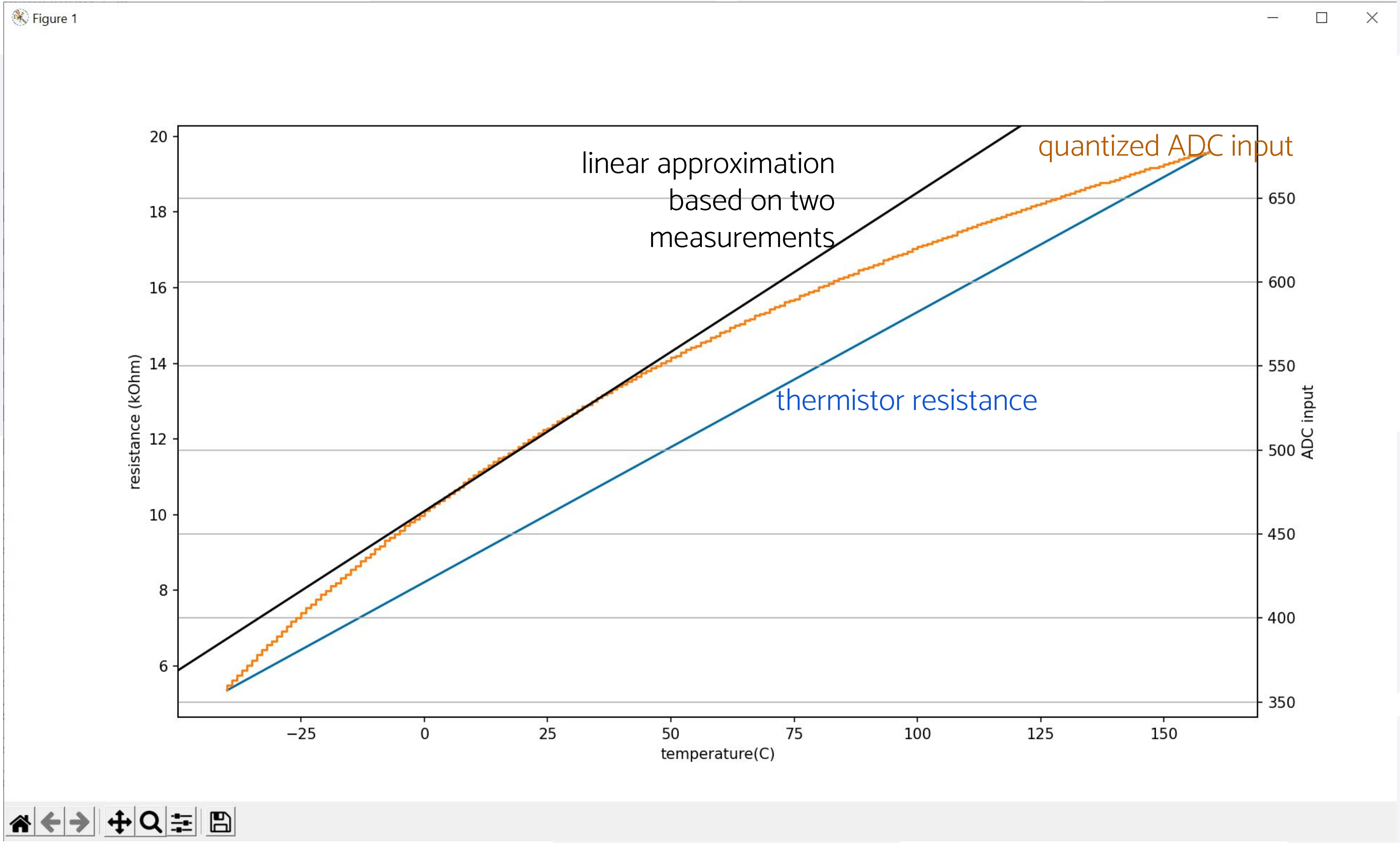


Figure 1
linear approximation
based on two
measurements
quantized ADC input
thermistor resistance
resistance (kOhm)
ADC input
temperature(C)
20
18
16
14
12
10
8
6
650
600
550
500
450
400
350
−25
0
25
50
75
100
125
150

Noise

Error of measured value comes from:

- Quantization
- Non-linearity
- Sensor **noise**

What are sources of noise of our thermistor?

Analog to Digital Converter

Analog signal gets discretized to some number

Done via an *Analog to Digital Converter* (ADC)

MCUs have ADC built in

- Different implementations - idea is to compare to reference voltages

Why it matters to embedded software developers: cost, timing, precision

- Find this in the data sheet!

32.2 Features

- 8-, 10- or 12-bit resolution
- Up to 350,000 samples per second (350ksps)

From the SAMD 21 family datasheet

ADC sharing

Mutex

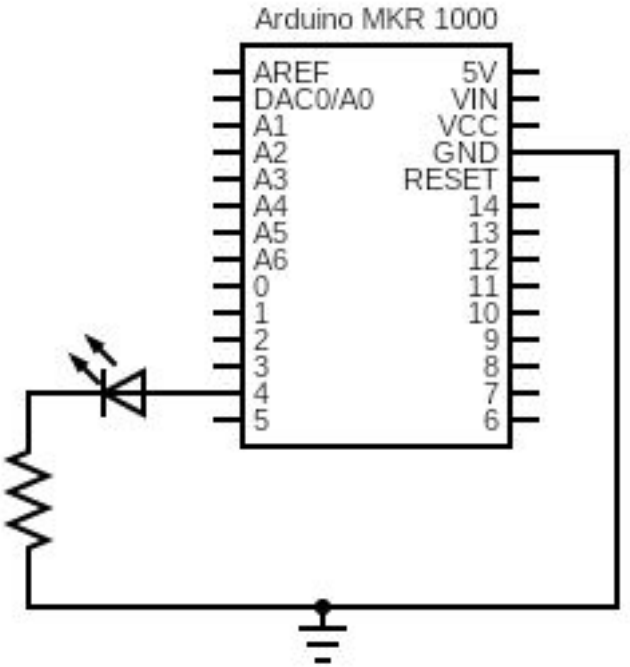
Up to individual task to check for ADC to be free and then ask for sample

Time-triggered

ADC runs in background and converts for all possible sources, storing the latest for each in a buffer

What tradeoffs do you see between mutex and time-triggered ADC sharing?

Reading and producing signals on an MCU



What about outputs?

Digital to analog converter - pin produces voltage from 0 to VCC

Pulse Width Modulation - simulates analog output

Digital to Analog Converter (DAC)

Divide voltage based on digital number

Actuator driven by DAC has similar quantization error to ADC

Some MCUs don't have DACs

- Expensive
- Fewer applications
- Unlike ADC, cannot share

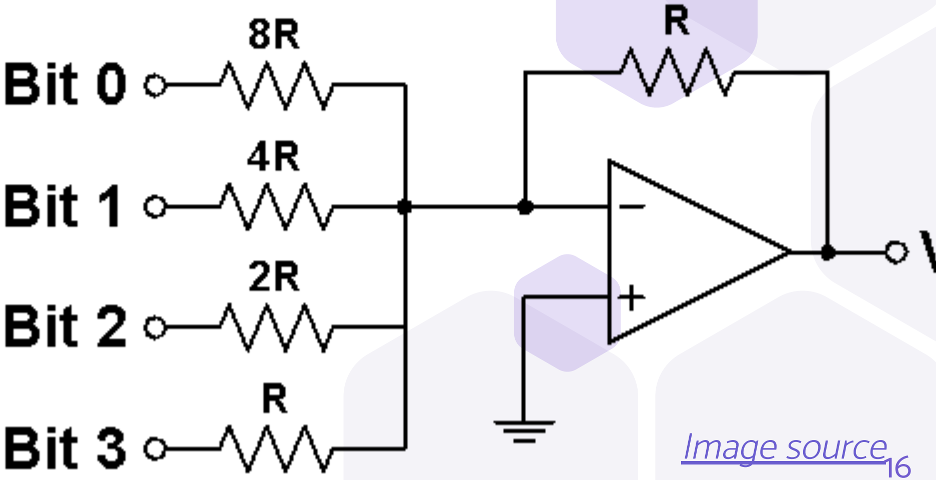


Image source

Pulse Width Modulation (PWM)

Rapidly switch digital pin on and off

Creates perception of analog output

Increasing/decreasing duty cycle increases/decreases perception of power output level

Many microcontrollers provide PWM peripherals

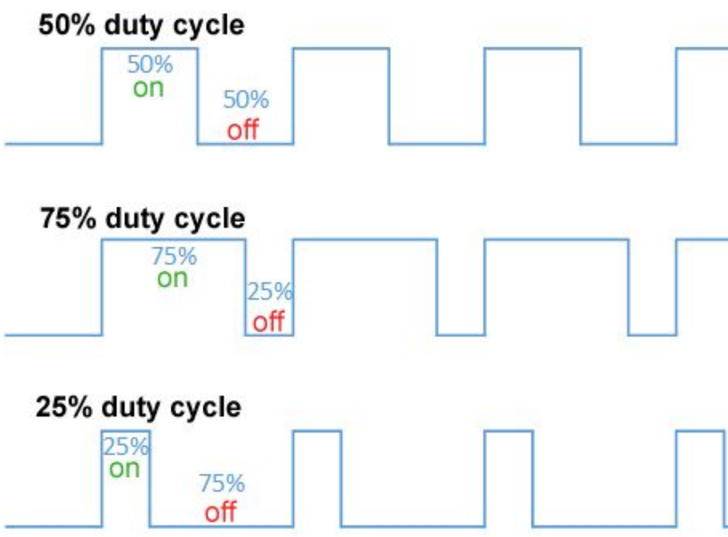


Image source

Digital Signal Processing (DSP)

ADC/DAC/PWM combined with computational power of an MCU has enabled the explosion of digital applications

- Audio, video, robotics, medical…

MCU lets you take in an analog signal, do computations on it, and produce a new analog signal

DSP is a cool area but (mostly) beyond the scope of this course

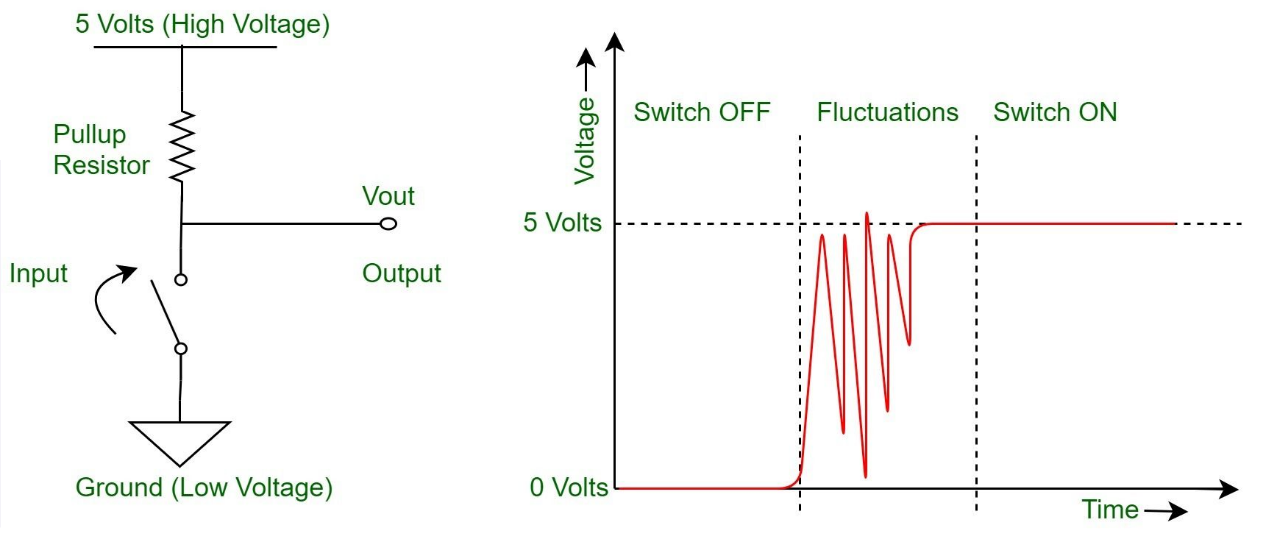
These things are not perfect

Quantization, non-linearity, error in components all contribute to imprecision

DSP can help alleviate some sources of error

Design and models that take sources of error into account are vital for some applications

Example of very simple DSP: debouncing



What do we do here?

Image source