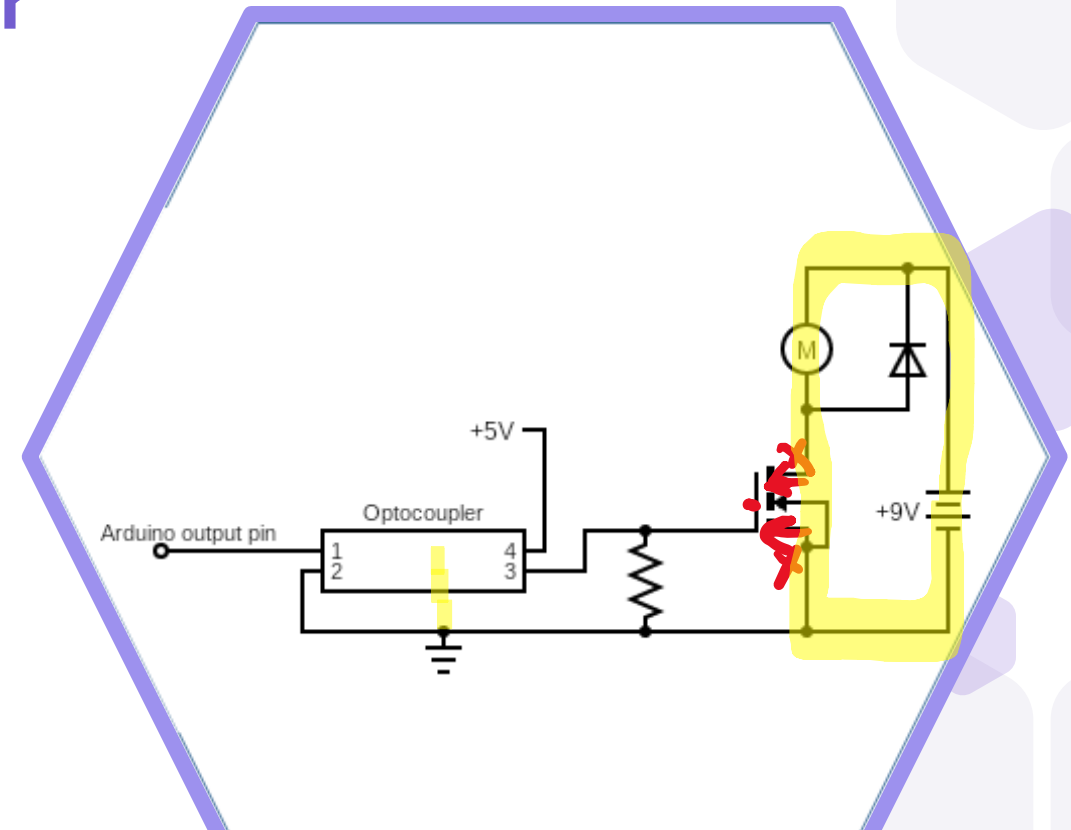
04: Analog and Digital signals

Motor driven using mosfet and optocoupler



Analog ↔ Digital I/O

Microcontrollers are digital

How do we read in analog input?

How do we produce or simulate analog output?

Modeling analog devices

Useful to have a mathematical model of sensors/actuators for verification and understanding

Linear and Affine Models, Range

Map *physical* value at time t *x(t)* to sensor input

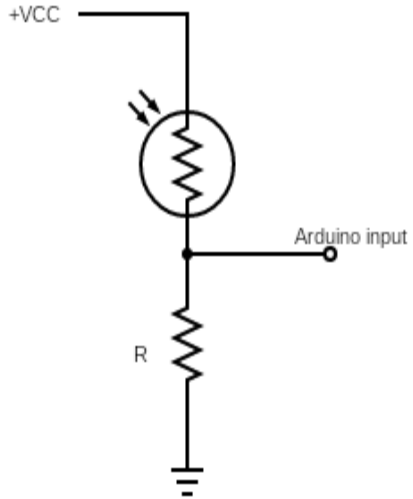
Approximate with linear affine function,
saturate if outside range [L,H]:

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

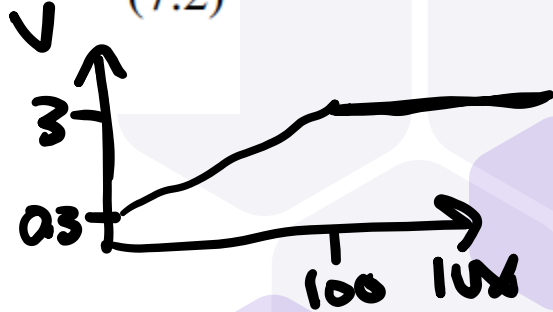
Example: photoresistor

Physical value *x(t)* is measure of light (e.g. illuminance)

MCU input voltage depends on physical properties of the sensor and how it's wired

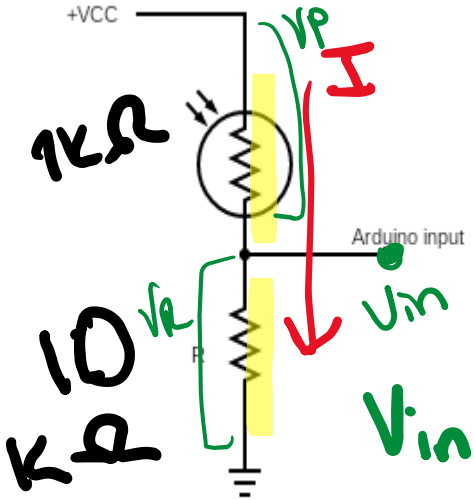


$$f(x(t)) = \begin{cases} ax(t)+b & \text{if } L \le x(t) \le H \\ aH+b & \text{if } x(t) > H \\ aL+b & \text{if } x(t) < L, \end{cases} \tag{7.2}$$



L = 100 lux → 1 kΩ → 3V

H = 100 k lux → 0 lux → 0.3V



$V_{in} = V_R$

$V_P + V_R = V_{cc} = 3.3V$

$1k\Omega \cdot I + 10k\Omega \cdot I = 3.3V$

(resistors in series)

$V_{in} = 10k\Omega \cdot I = 10k\Omega \cdot \frac{V_{cc}}{R_c} = 10k\Omega \cdot \frac{3.3V}{11k\Omega}$

$= 3V$

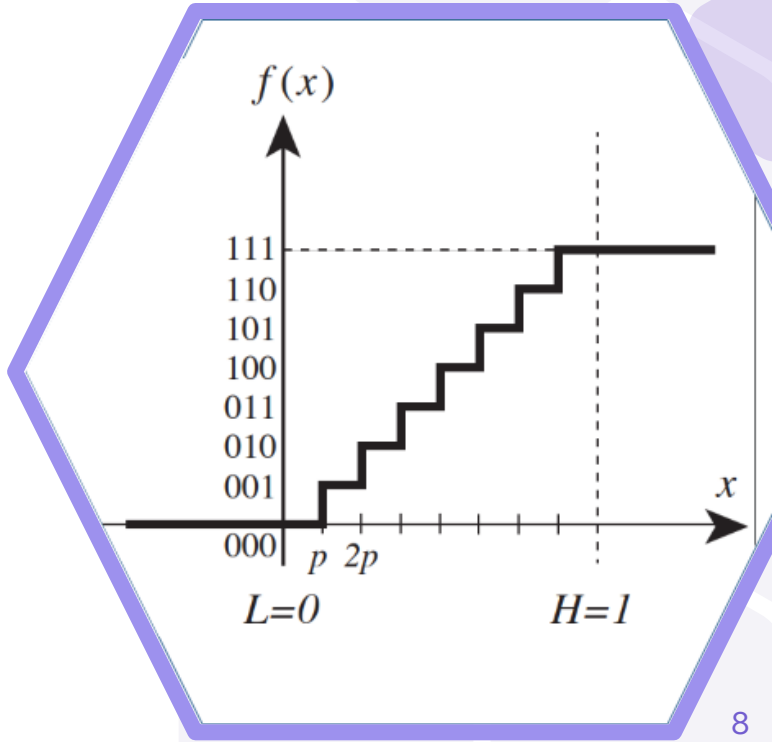
Quantization

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

For our example,

*f(v(x(t)) = 1023 * v(x(t)) / VCC*



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

What is the smallest change in lux we can detect for our example?

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

Noise

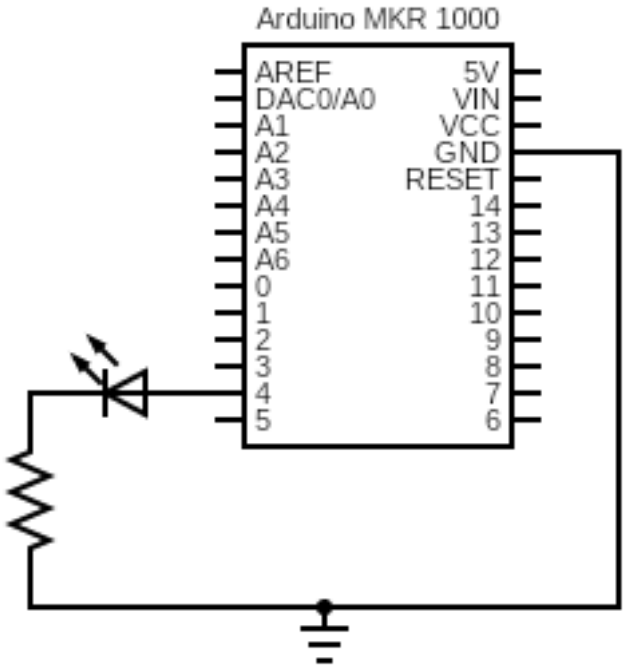
Error of measured value

Comes from:

- Quantization
- Actual non-linearity
- Sensor imperfections

What is the noise of our photoresistor?

Reading and producing signals on an MCU



Analog to Digital Converter

Analog signal gets discretized to some number

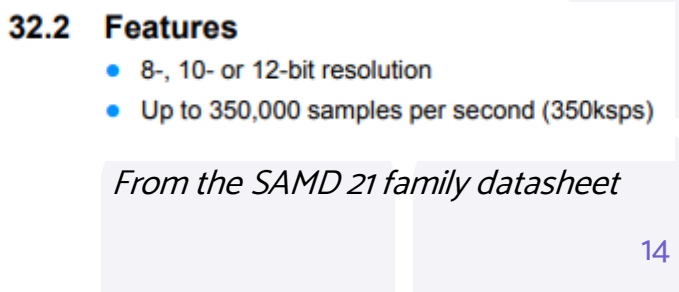
Done via an *Analog to Digital Converter* (ADC)

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!



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?

ADC sharing

Mutex

Probably scales better

Different resolutions for different pins?

Greedy pin ☹

Time-triggered

Data might grow slightly stale between readings

A bit wasteful (requires buffer)

More precise guarantees on timing