DTU

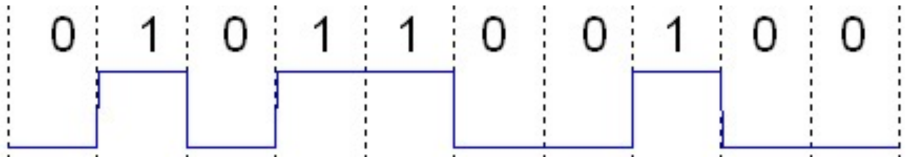
02226 – Networked Embedded Systems

Week 3: MCU Input/Output

Xenofon (Fontas) Fafoutis
Associate Professor
xefa@dtu.dk
www.compute.dtu.dk/~xefa

Digital Logic

- Two distinguishable voltage levels are mapped in two a binary digit
- **High voltage** (typically, the supply voltage) is ‘1’
- **Low voltage** (typically, the ground) is ‘0’
- Real world is noisy, but noise can be tolerated
  - E.g. if $V>V_S/2$, then ‘1’



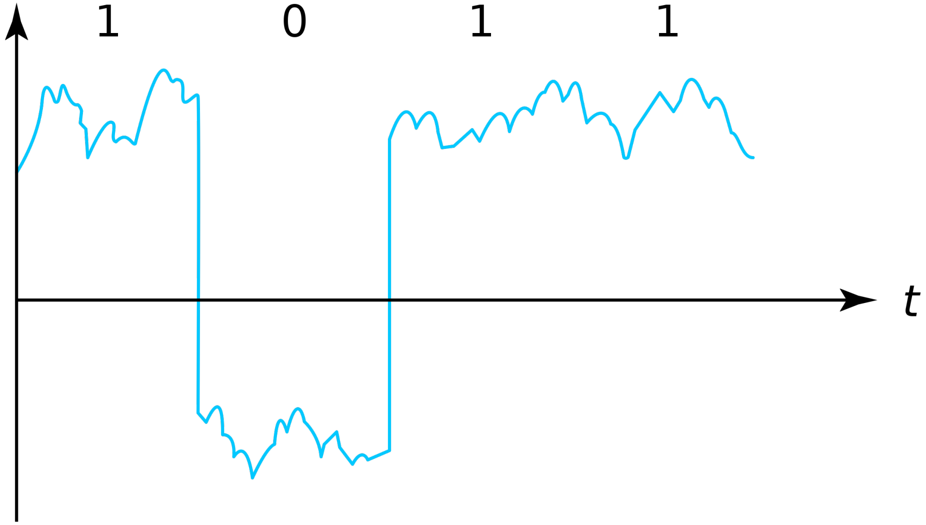


Image source: Wikipedia

General Purpose Input/Output (GPIO)

- A GPIO is controllable by software and can:
  - Set the logic level of a line to high or ‘1’
  - Set the logic level of a line to low or ‘0’
  - Read the logic level of the line

- Example: the nRF2832 has 32 GPIOs
  - Labelled as P0.00-P0.31 in the figure

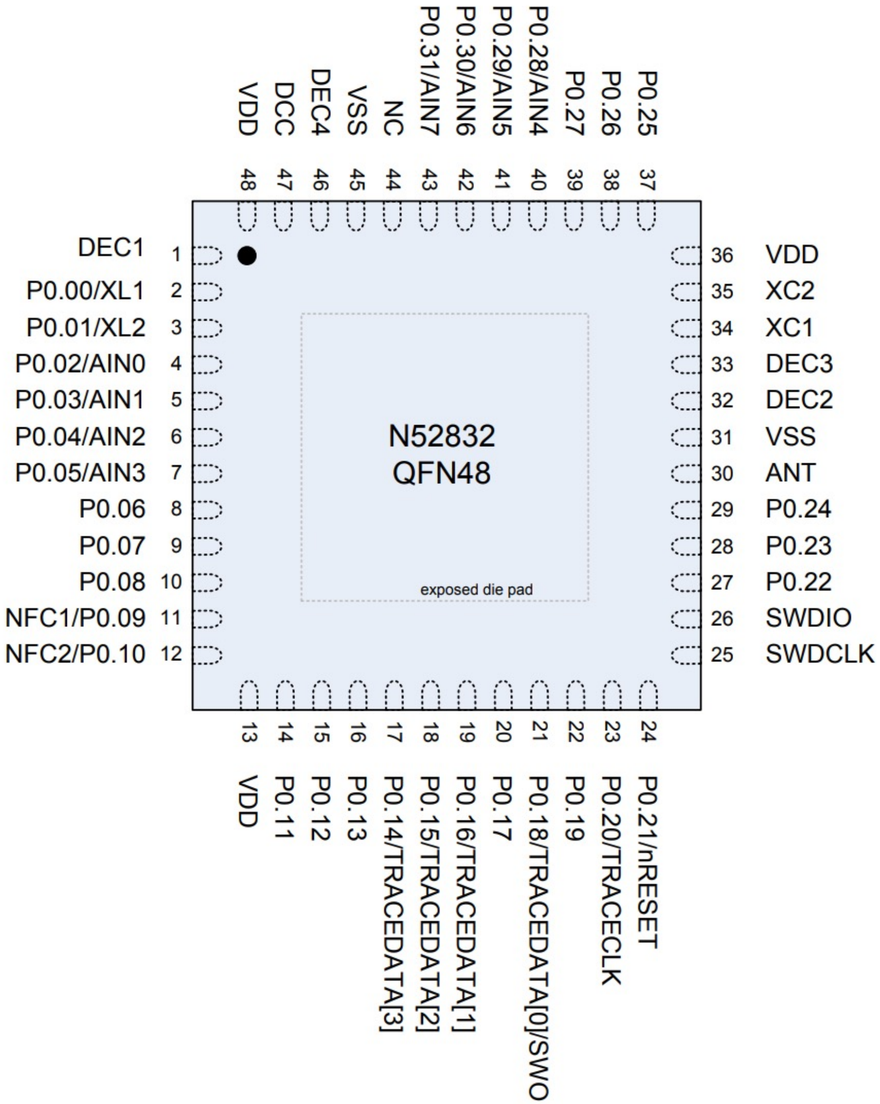
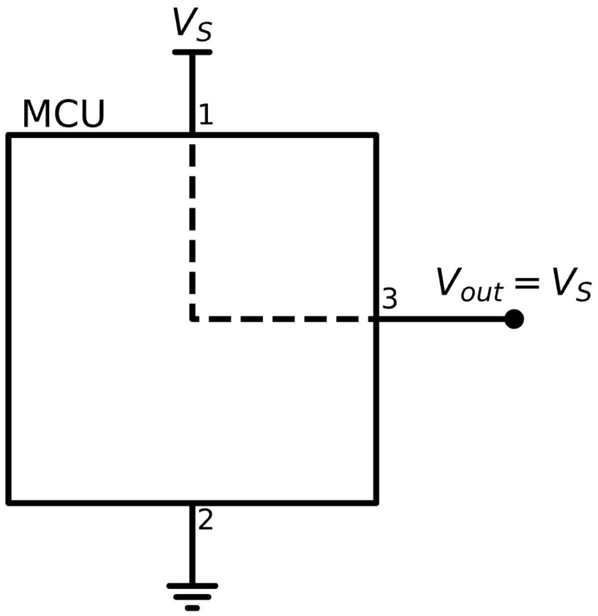
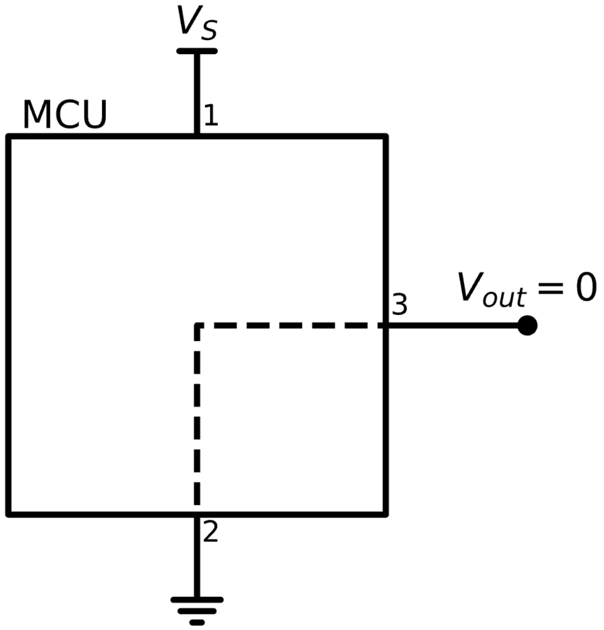


Image source: nRF52832 Datasheet by Nordic Semiconductors

GPIO Output: Push-Pull Mode

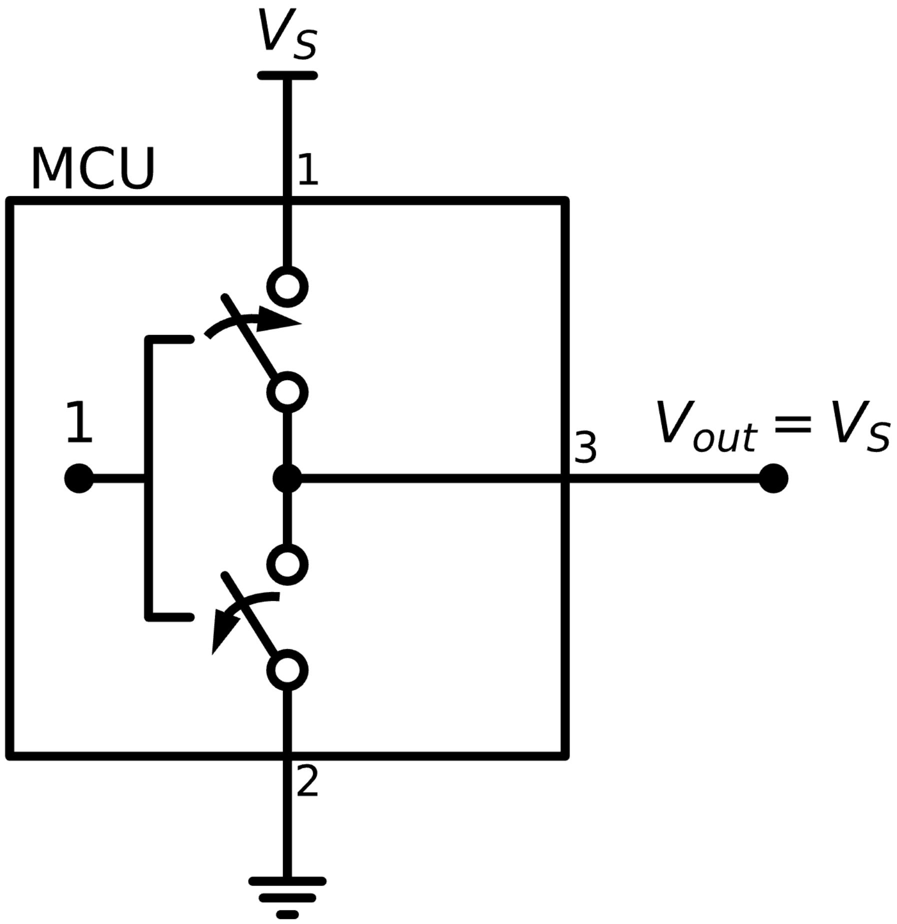
- The MCU **drives** the output by actively connecting it to the supply voltage or ground
- When the output voltage is high, the MCU **pushes** the output to V_S
- When the output voltage is low, the MCU **pulls** the output to the ground
- Often push-pull mode is the default output state of the GPIO
- Controlled in **software**





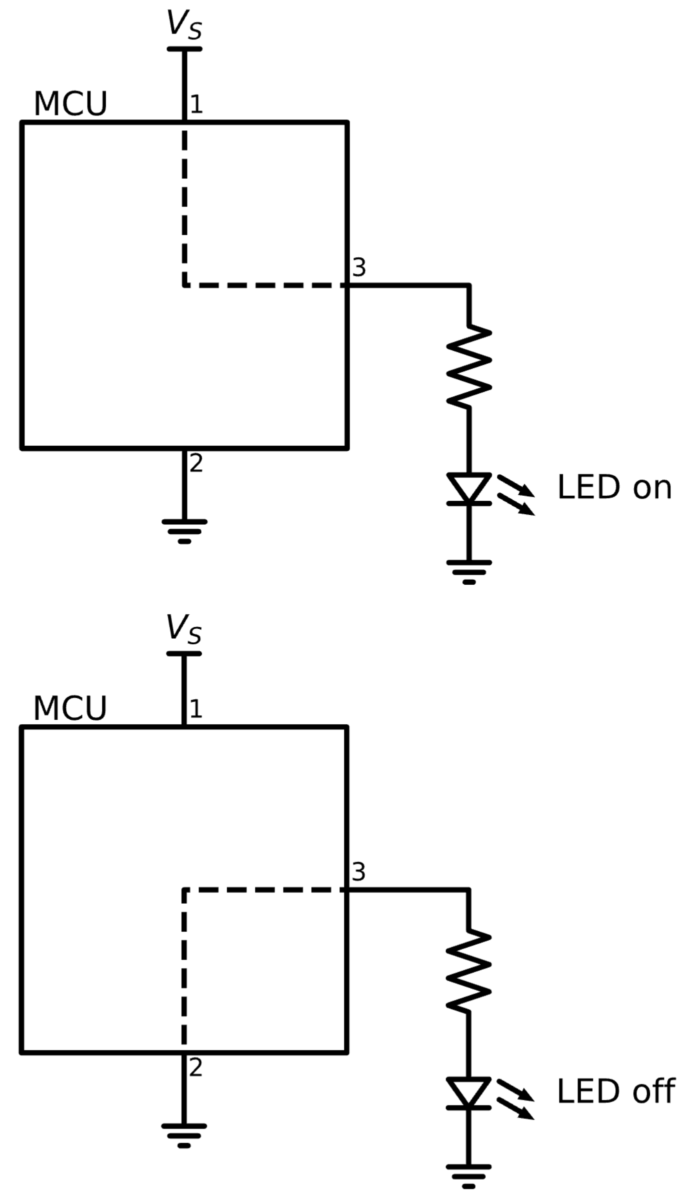
GPIO Output: Push-Pull Mode

- Implemented with MOSFET switches
- Input ’1’ connects output pin to V_S while simultaneously disconnects it from the ground



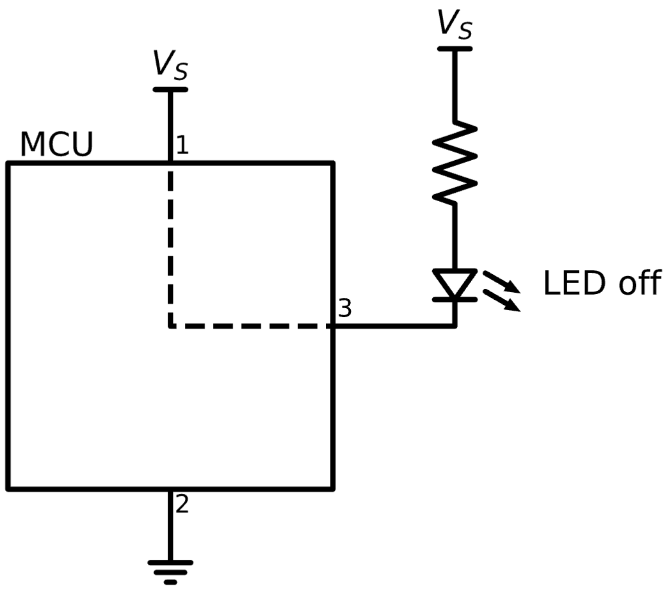
GPIO as a Current Source

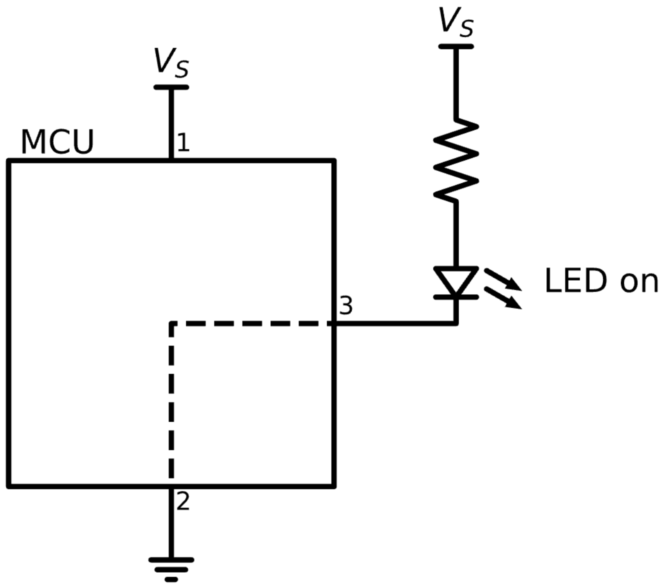
- Communication signals are low current signals
- GPIOs can also **source current** to power external components
- MCU can use GPIO as an **on/off switch**
- However, the **output current** of a GPIO is limited (typically, 2-12mA)
- MCU might have some **high drive** GPIOs that can output more current (e.g. 40mA)
- High-power components need to be connected directly to the power source



GPIO as a Current Sink

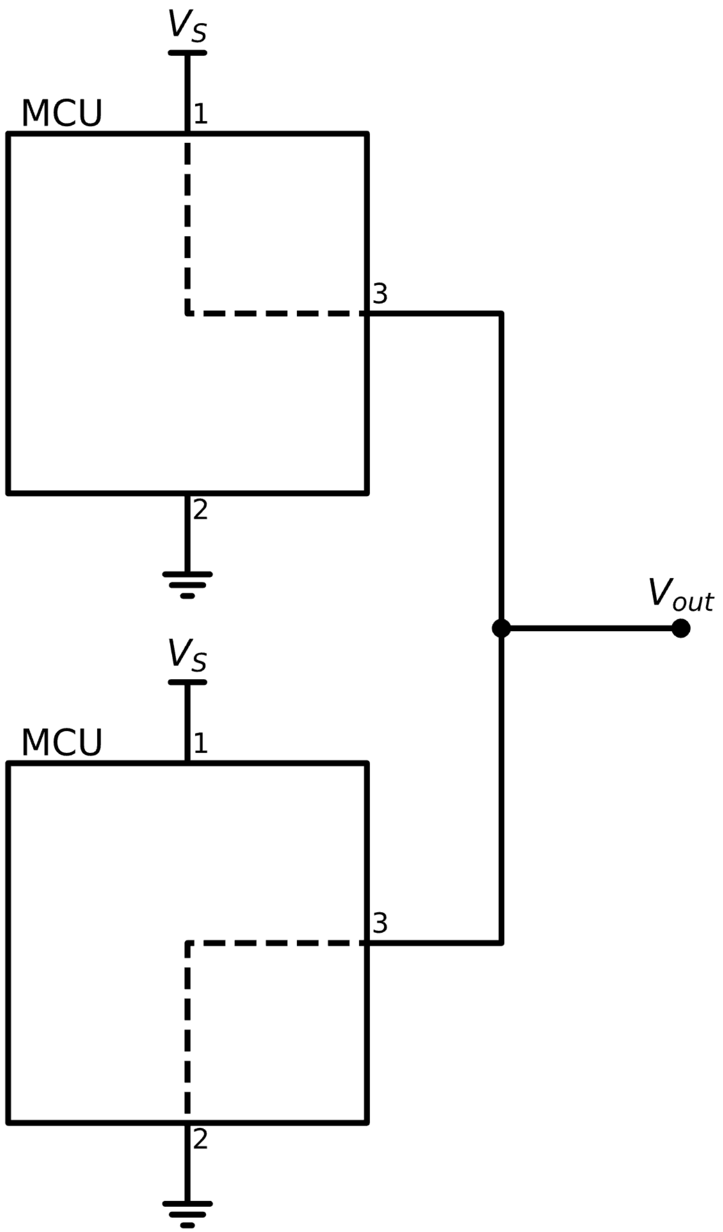
- GPIO can also **sink current**
- Current flows inwards
- Closes the circuit by connecting it to the ground
- Logic is reversed
  - ‘1’ turns off the LED
  - ‘0’ turns on the LED
- Sink current also limited





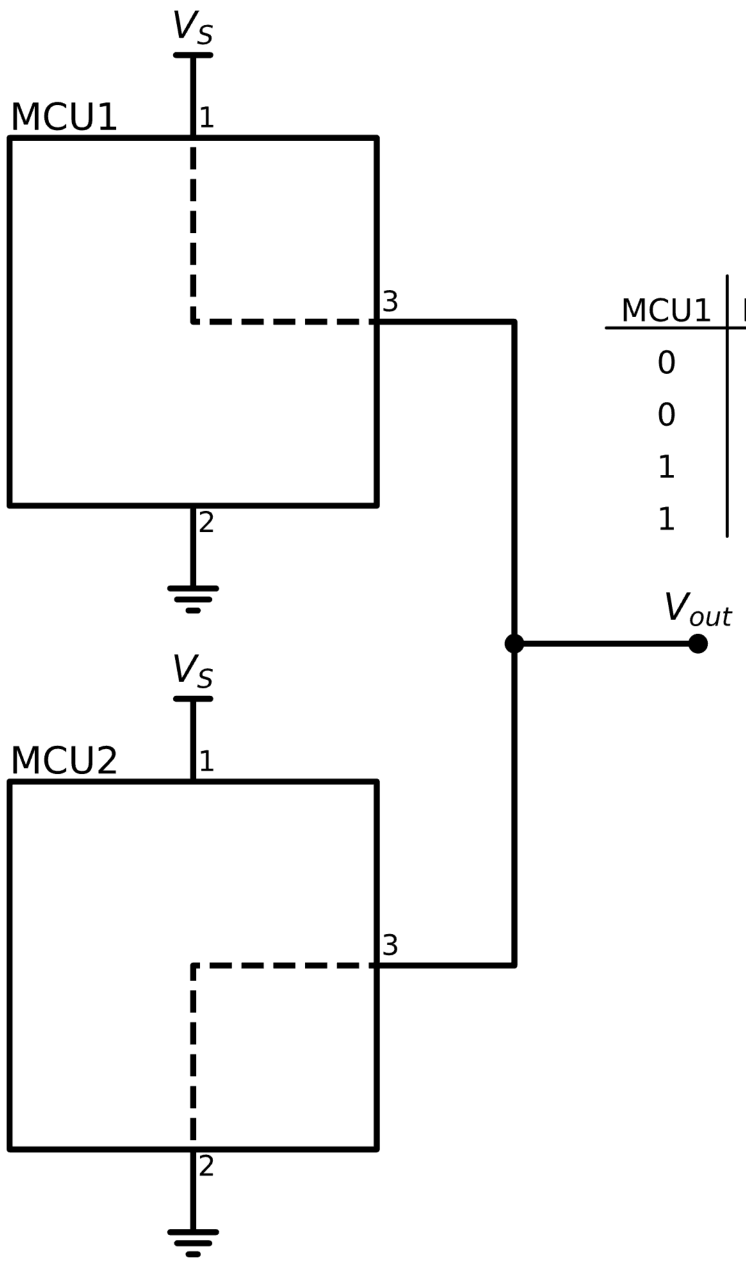
Two MCUs

- Multiple devices may share the same output line
- What happens if two MCUs drive the same output line?



Two MCUs

- Multiple devices may share the same output line
- What happens if two MCUs drive the same output line?
- If one device pushes the output high while the other pulls it low, we have **a short circuit**
- In general, **only one** device should drive a line
- Two output states are not enough, we need to have the option to **disconnect** the output



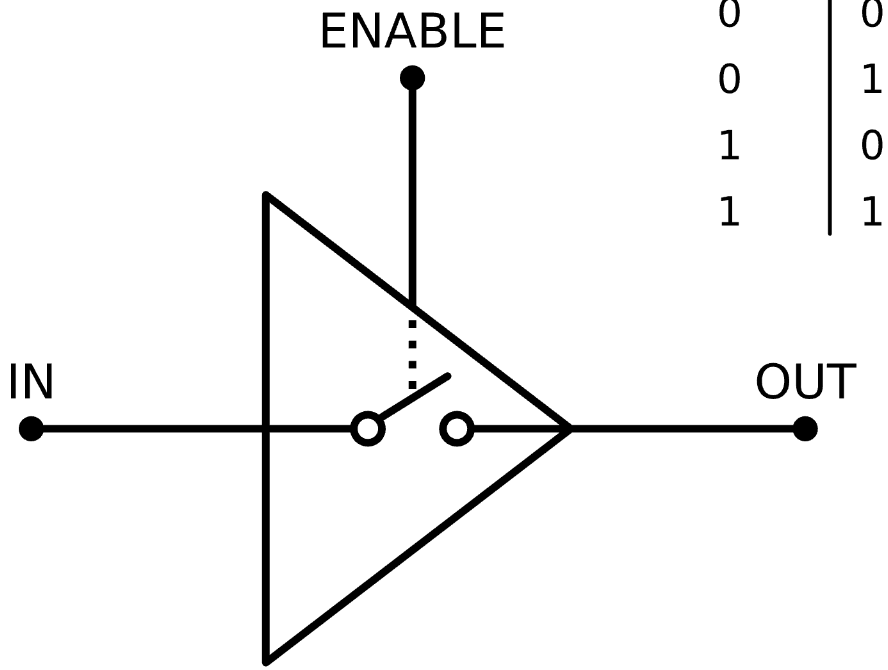
MCU1	MCU2	V_{out}
0	0	0
0	1	SHORT
1	0	SHORT
1	1	1

Tri-State Logic

- Three States
  - Logical ‘0’ (i.e. low voltage)
  - Logical ‘1’ (i.e. high voltage)
  - High Impedance or Hi-Z (disconnected)

- Tri-state Buffer
  - Implemented with MOSFET switches
  - At open state the resistance is so high that no current can go through
  - Equivalent to open circuit

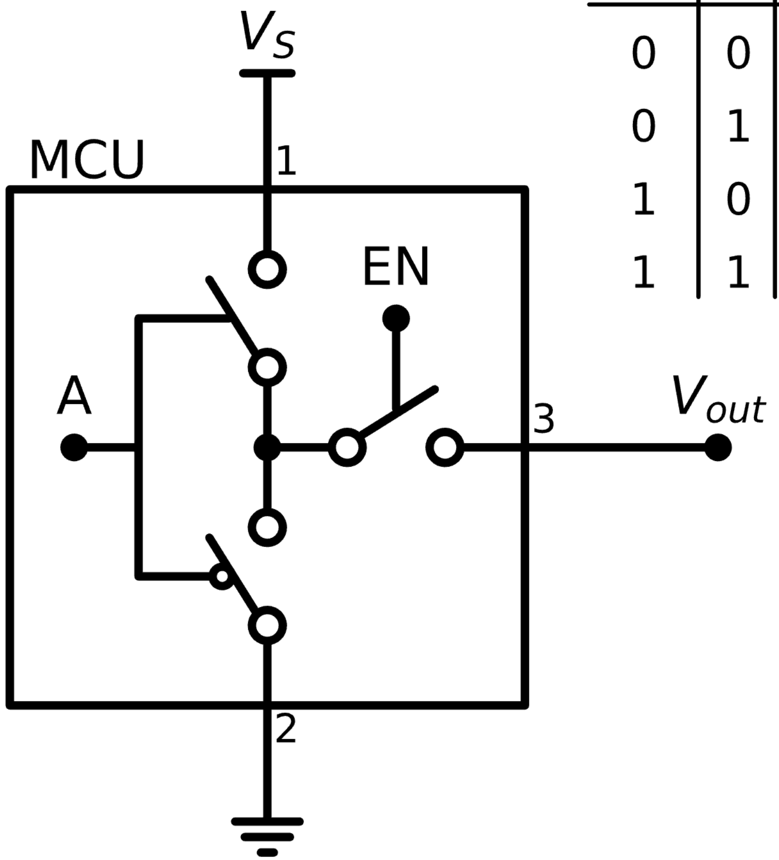
ENABLE	IN	OUT
0	0	Hi-Z
0	1	Hi-Z
1	0	0
1	1	1



GPIO Output States

- With two controls:
  - Output value (A)
  - Enable (EN)

- GPIO can be:
  - Actively pushed to ‘1’
  - Actively pulled to ‘0’
  - Disabled (Hi-Z)



EN	A	OUT
0	0	Hi-Z
0	1	Hi-Z
1	0	0
1	1	1

GPIO Input and Input Hysteresis

- **Comparator** is a component that compares two (analogue) input signals and outputs '0' or '1' depending which is larger
- A GPIO input can be implemented by comparing the input signal to $V_S/2$
- Noise close to the threshold creates quick changes successive changes between '0' and '1'
- **Input Hysteresis** mitigates this problem by applying two different thresholds
  - To go from '0' to '1', $V > (V_S/2) + V_H$
  - To go from '1' to '0', $V < (V_S/2) - V_H$

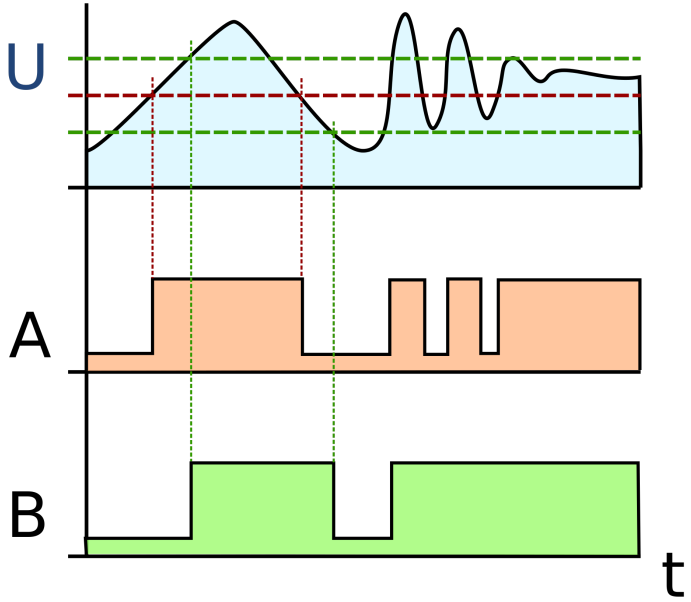
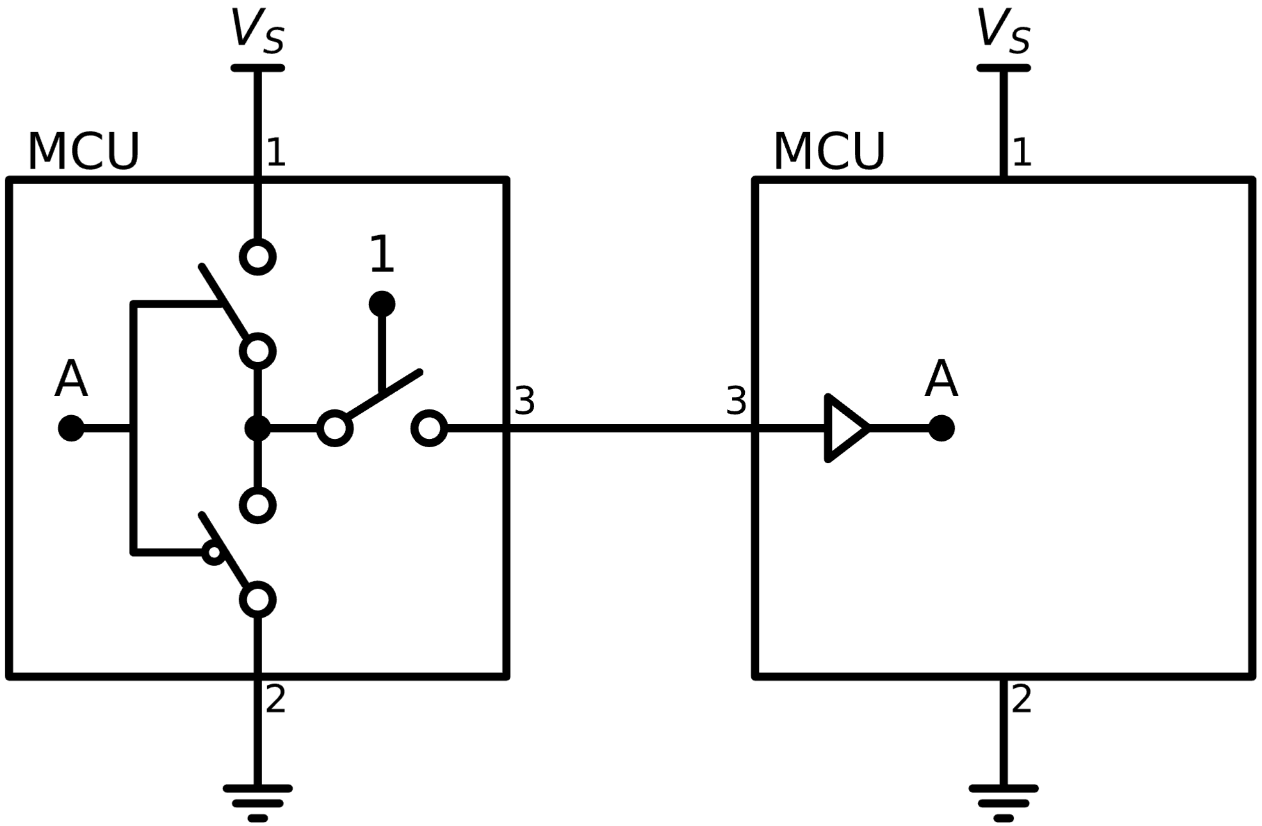


Image source: Wikipedia

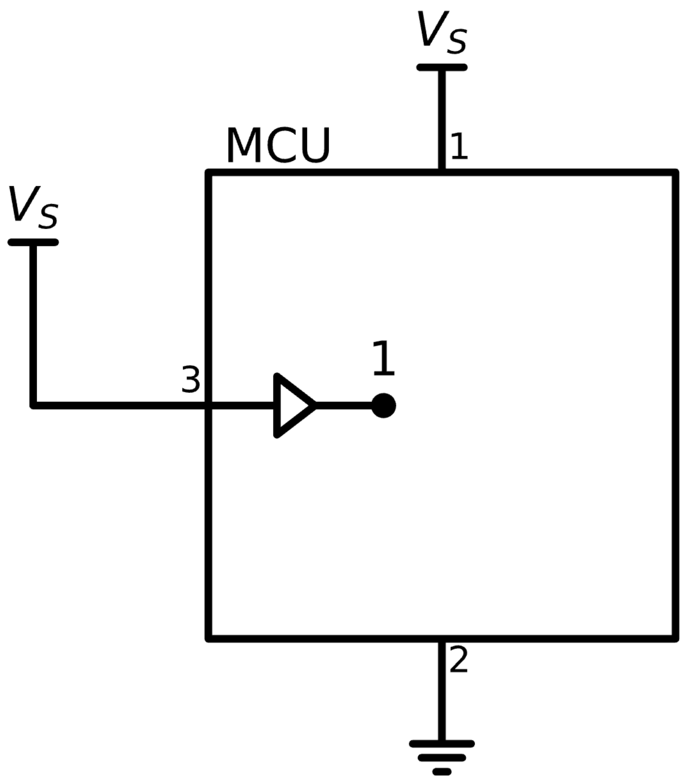
GPIO Input Mode

- When EN=1, the input IN equals A



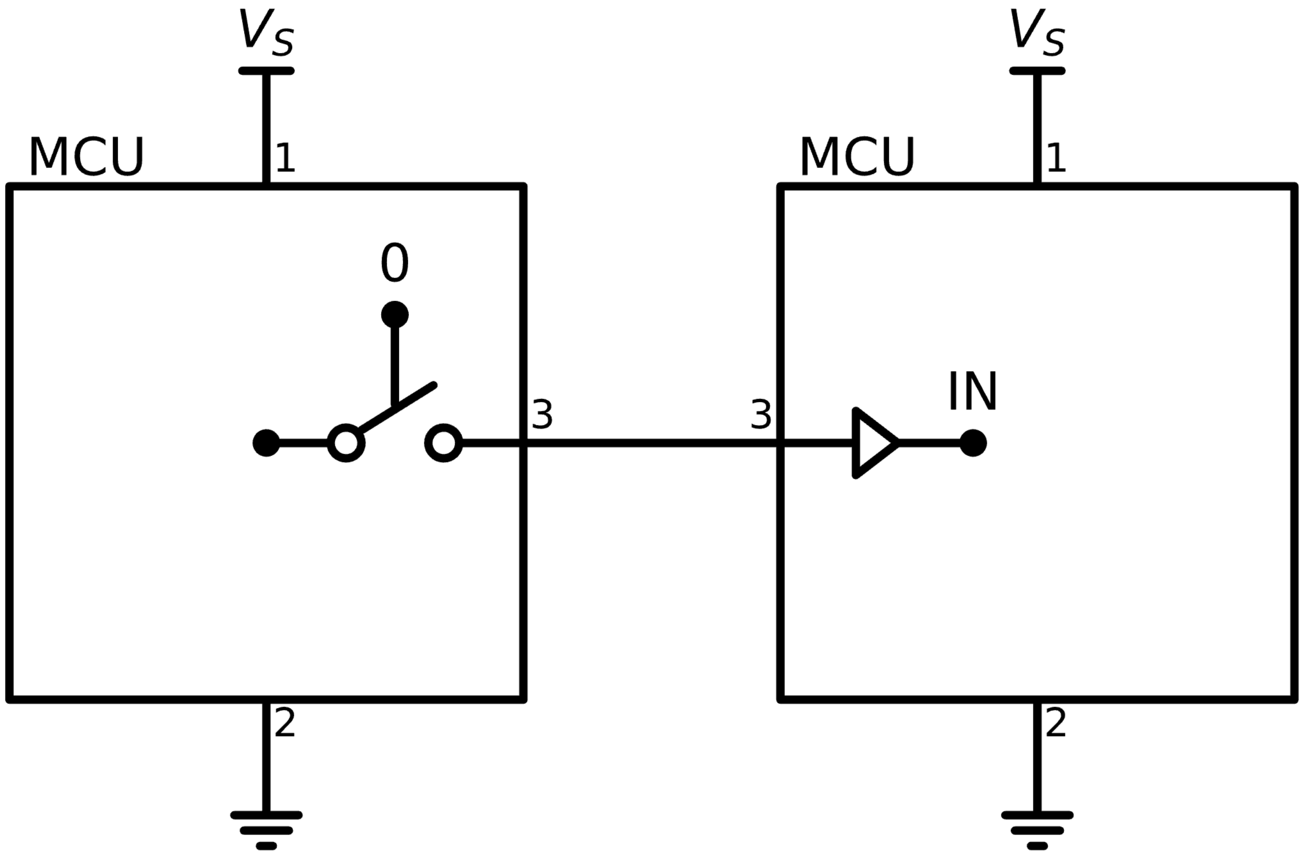
Hardwired Input

- Input can also be hardwired
  - Fixed to have **constant value** in the circuit
- Permanent configuration
  - Assuming input controls some configuration
- As a form of ID
  - Consider a circuit with two MCUs running the same software, but one has Pin3 hardwired to '0' and the other has Pin3 hardwired to '1'
  - Reading Pin3 is a form of self-identification



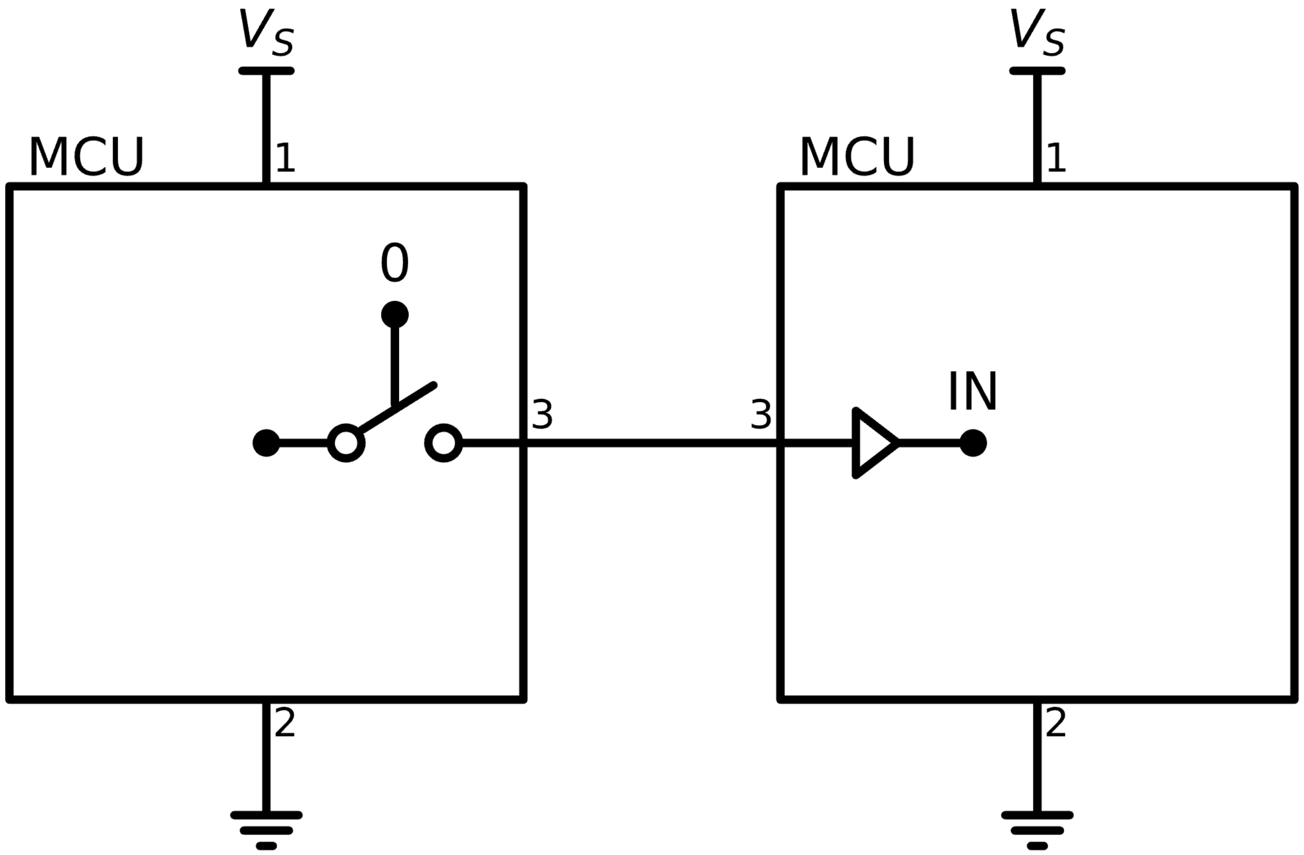
Reading a Hi-Z output

- What value will we get if we try to read a disabled output?



Reading a Hi-Z output

- What value will we get if we try to read a disabled output?
- The input is undefined
  - It can be 0 or 1 or can switch between them
- It essentially acts as a tiny antenna, capturing noise from its surroundings
- A Hi-Z pin is **floating**



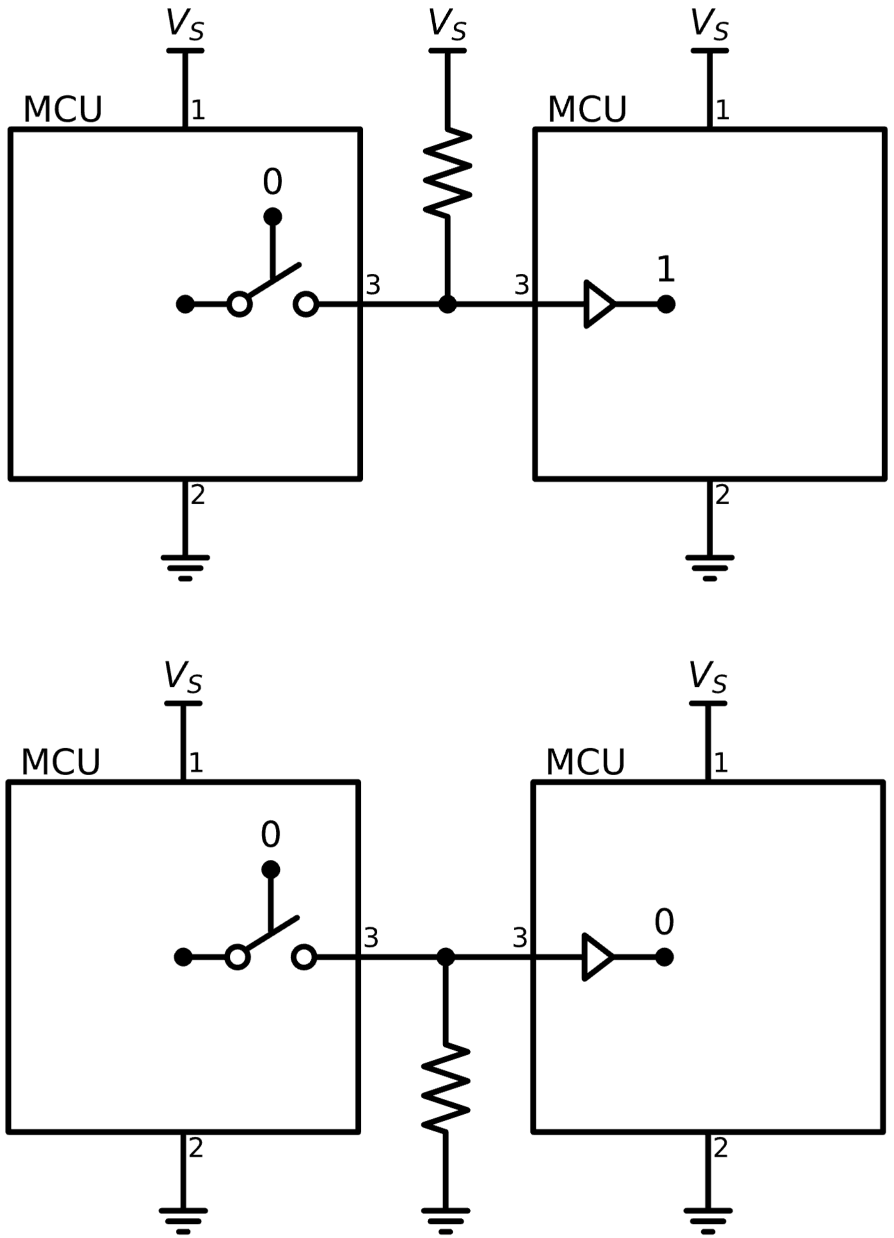
Pullup and Pulldown Resistors

- They provide **a default state**

- If nobody actively drives the line
  - The pullup resistor will raise the line to '1'
- If the line is actively pulled to '1'
  - No voltage difference across the pullup resistor (no current), line is '1'
- If the line is actively pushed to '0'
  - There is voltage difference across the pullup resistor (current goes through), line is '0'

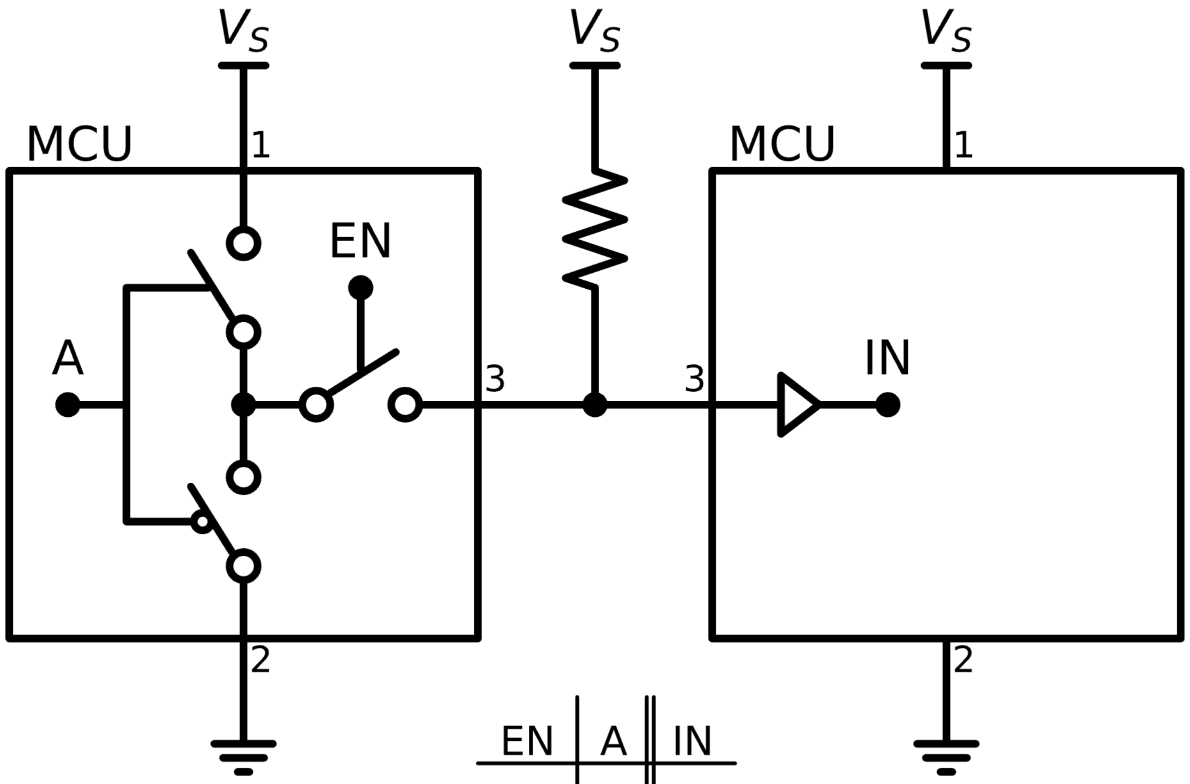
- Similarly, the pulldown resistor defaults to '0'

- MCU typically supply internal pullup/pushdown resistors



GPIO Output with Pullup

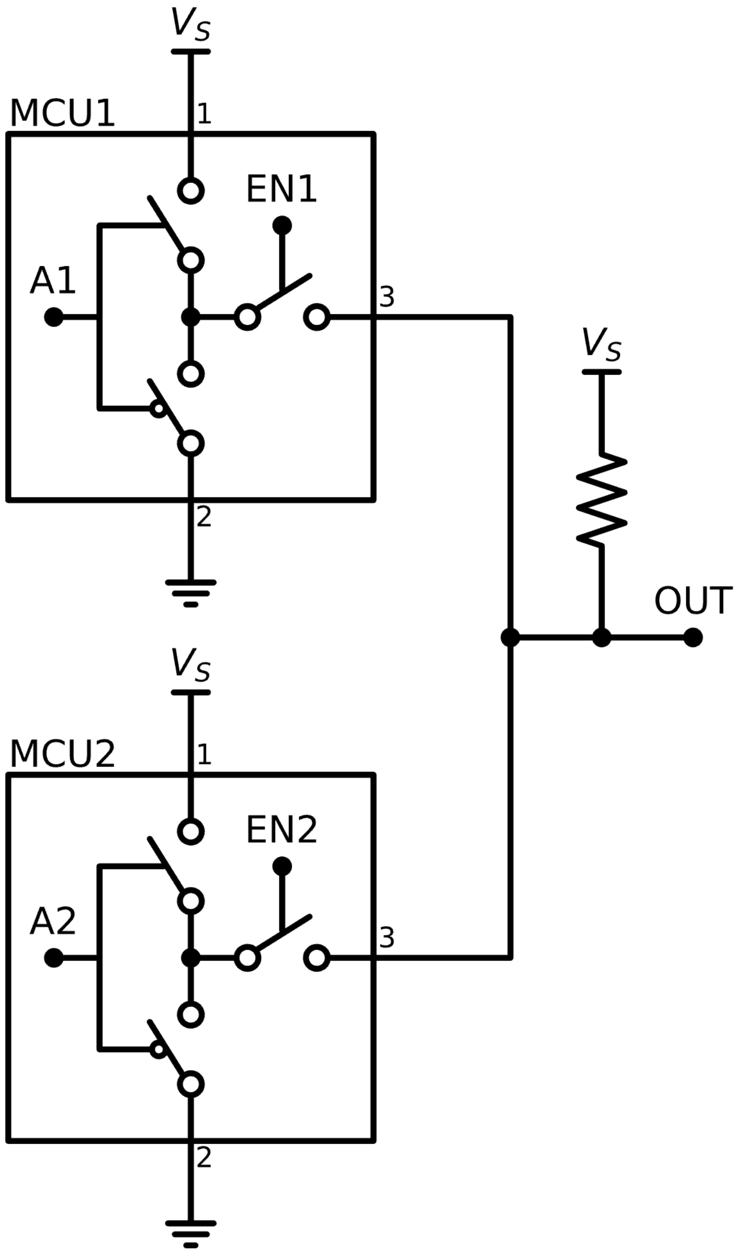
- If output is disabled (EN=‘0’)
  - Pullup resistor determines the input (IN=‘1’)
- If output is enabled (EN=‘1’)
  - Output determines the input (IN=A)



EN	A	IN
0	0	1
0	1	1
1	0	0
1	1	1

Multiple MCUs

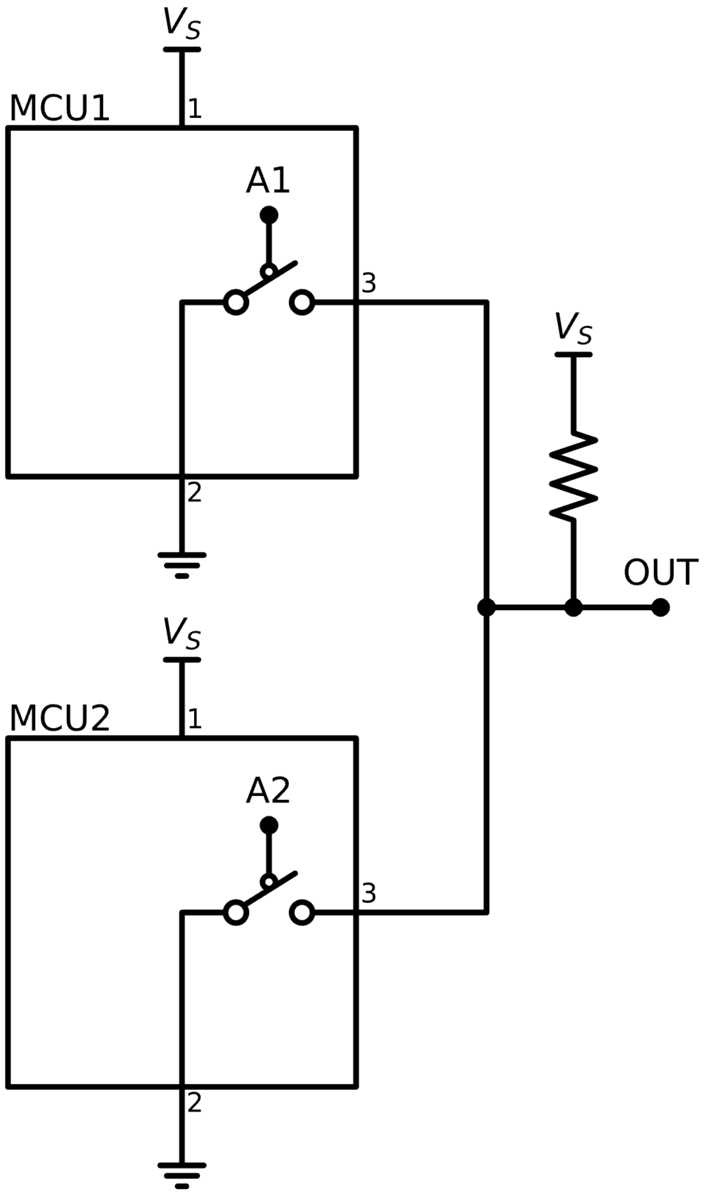
- OK if one output is enabled at a time
- Short circuits are still possible with push-pull outputs
- Is this safe enough?



A1	EN1	A2	EN2	OUT
0	0	0	0	1 (pullup)
0	0	0	1	0 (MCU2)
0	0	1	0	1 (pullup)
0	0	1	1	1 (MCU2)
0	1	0	0	0 (MCU1)
0	1	0	1	0 (both)
0	1	1	0	0 (MCU1)
0	1	1	1	SHORT
1	0	0	0	1 (pullup)
1	0	0	1	0 (MCU2)
1	0	1	0	1 (pullup)
1	0	1	1	1 (MCU2)
1	1	0	0	1 (MCU1)
1	1	0	1	SHORT
1	1	1	0	1 (MCU1)
1	1	1	1	1 (both)

GPIO Output: Open-Drain Mode

- Open-Drain mode has two states:
  - Logical ‘0’ (i.e. low voltage)
  - Hi-Z (i.e. disconnected)
- Sets the line to ‘0’ **actively**
- Sets the line to ‘1’ **indirectly** by disconnecting the output and letting the pullup resistor raise it to ‘1’
- The pullup resistor can also be internal
- Open-Drain is used when multiple devices share the same line (safe from shorts)



A1	A2	OUT
0	0	0 (both)
0	1	0 (MCU1)
1	0	0 (MCU2)
1	1	1 (pullup)

Analog-to-Digital Converter (ADC)

- An ADC converts an analogue signal (voltage signal) into a digital signal (series of numbers)

- An ADC essentially measures the voltage of the input and maps it to a number

- The Bit Resolution (M) defines the number of discrete values the ADC can produce

- E.g.: A 12-bit ADC maps the input to a number in [0,4095], i.e., 2^{12} values

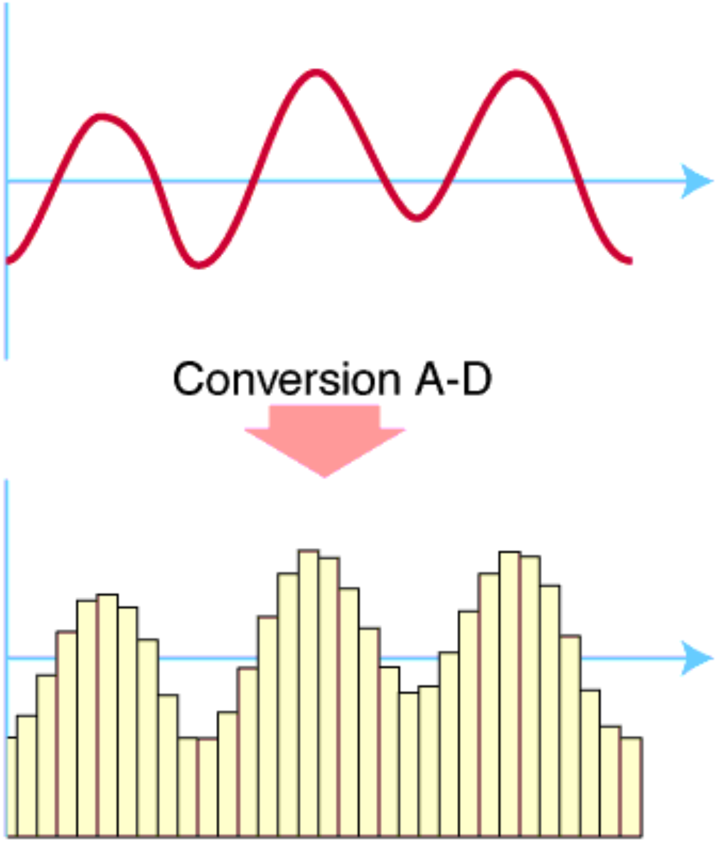
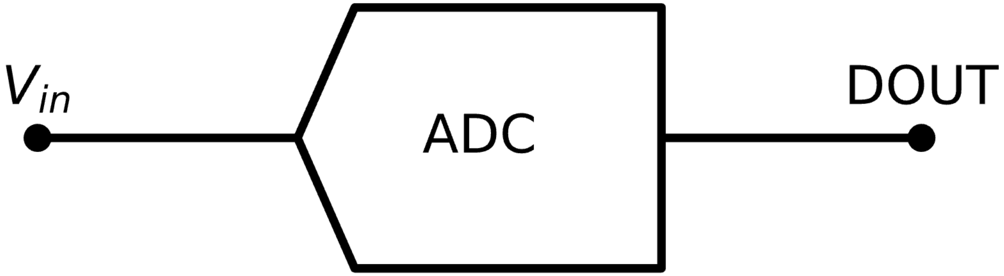


Image source: Wikipedia

Single-Ended ADC

- Measures input voltage (V_{in}) referenced to the ground
  - Input and ADC have common ground
- The reference voltage (V_{ref}) defines the maximum value that can be measured
  - V_{ref} can be equal to the supply voltage or less
  - Voltages in $[0, V_{ref}]$ are linearly mapped in $[0, 2^M-1]$
  - If $V_{in} > V_{ref}$, DOUT=2^M (saturation)
- Examples:
  - A 12-bit ADC with V_{ref}=5V will map:
    - V_{in}=0V to 0
    - V_{in}=2.5V to 2048
    - V_{in}=5V to 4095
    - V_{in}=6V to 4095



$$\mathrm{DOUT} = \frac{V_{in}}{V_{ref}} \cdot 2^M$$

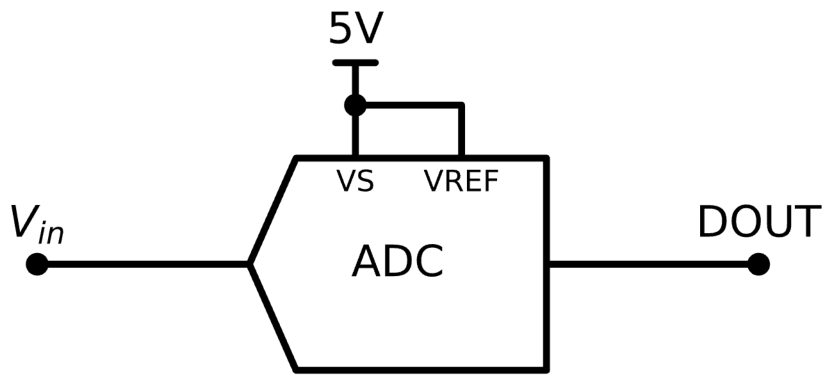
Single-Ended ADC

- Quantisation error
  - The continuous analogue values are mapped to the closest integer
  - The reference voltage (V_{ref}) together with the bit resolution (M) define the voltage resolution (Q)
  - In other words, measurements are in steps or quanta of $Q = V_{ref} / 2^M$
- Examples:
  - A 12-bit ADC with V_{ref}=5 V, Q= 1.22 mV
  - A 12-bit ADC with V_{ref}=2.5 V, Q= 0.6 mV
  - A 8-bit ADC with V_{ref}=5 V, Q= 19.5 mV
  - A 8-bit ADC with V_{ref}=2.5 V, Q= 9.7 mV
- Trade-off: range vs resolution

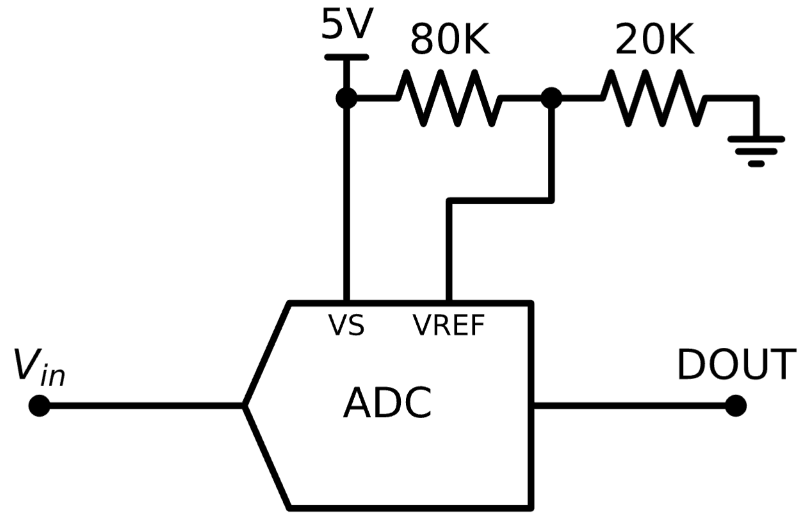
$$DOUT = \frac{V_{in}}{V_{ref}} \cdot 2^M$$

ADC: Range vs Resolution

- Assuming an 8-bit ADC and supply voltage 5V
- If I want to measure a signal in [0, 5] Volts
  - I can do it with a 19.5 mV resolution
- If I want to measure a smaller signal in [0,1] Volts with more subtle variations
  - With V_{ref}=5V, 19.5 mV resolution may not be enough and DOUT values [52,255] will never be used!
  - Instead with V_{ref}=1V, I have 3.9 mV resolution and I use all DOUT values [0,255]



$V_{ref} = 5\,V, \quad Q = 19.5\,mV$

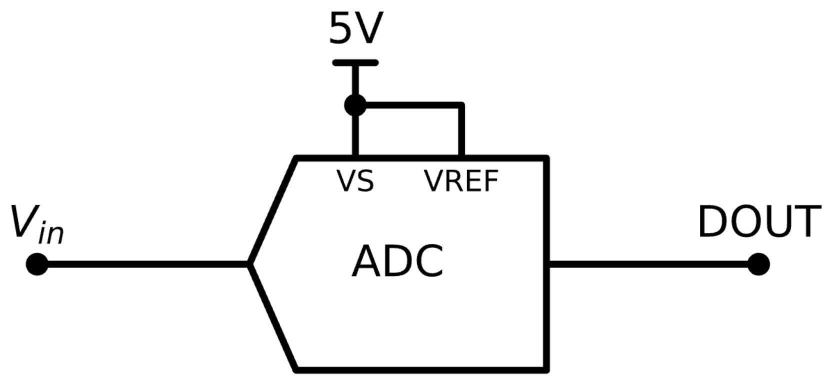


$V_{ref} = 1\,V, \quad Q = 3.9\,mV$

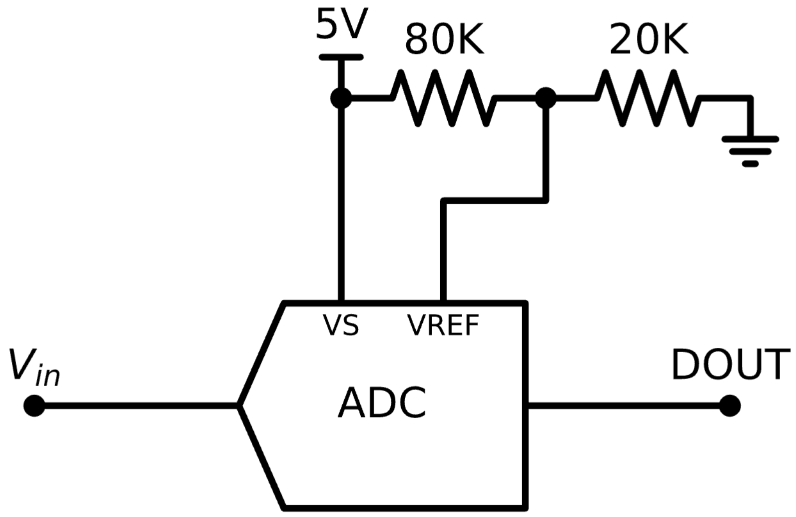
ADC: Range vs Resolution

- Assuming an 8-bit ADC and supply voltage 5V
- If I want to measure a signal in [0, 5] Volts
  - I can do it with a 19.5 mV resolution
- If I want to measure a smaller signal in [0,1] Volts with more subtle variations
  - With V_{ref}=5V, 19.5 mV resolution may not be enough and DOUT values [52,255] will never be used!
  - Instead with V_{ref}=1V, I have 3.9 mV resolution and I use all DOUT values [0,255]

How can I measure a signal in [4,5] Volts with resolution 3.9mV?



$V_{ref} = 5\,V, \quad Q = 19.5\,mV$



$V_{ref} = 1\,V, \quad Q = 3.9\,mV$

Differential ADC

- Measures the voltage difference between two input signals (V_A - V_B)
  - A differential voltage can be positive or negative depending which signal is higher
  - Better at rejecting noise

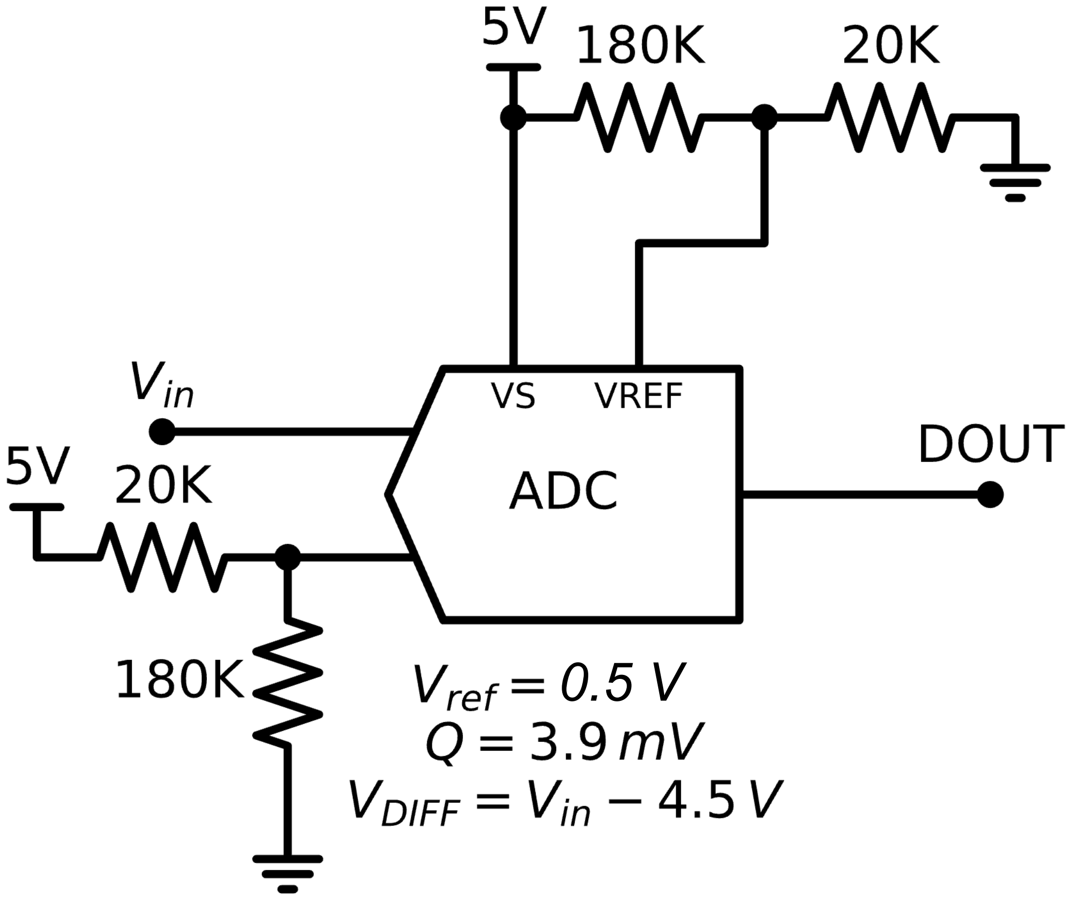
- The reference voltage (V_{ref}) defines the maximum value that can be measured
  - V_{ref} can be equal to the supply voltage or less
  - Voltages in [$-V_{ref}$, V_{ref}] are linearly mapped in [-2^{M-1}, $2^{M-1}-1$]
  - If $|V_P - V_N| > V_{ref}$, DOUT=2^{M-1} (saturation)
  - The resolution = Q = $V_{ref} / 2^{M-1}$

- Examples:
  - A 12-bit ADC with V_{ref}=5V will map:
    - V_A=0V and V_B=5V to -2048
    - V_A=5V and V_B=0V to 2047
    - V_A=5V and V_B=5V to 0
    - V_A=3V and V_B=2V to 410

$$\mathrm{DOUT} = \frac{V_A - V_B}{V_{ref}} \cdot 2^{M-1}$$

ADC: Removing a DC Offset

- Measure a small single-ended signal with a large DC-offset
- Assuming an 8-bit ADC and supply voltage 5V
- I want to measure a signal in [4, 5] Volts with 3.9mV resolution
- Apply to the negative input a fixed 4.5V signal using a voltage divider
  - The differential signal (V_{DIFF}=V_{in}-4.5V) is in [-0.5, 0.5] Volts
- With V_{ref}=0.5V, I can measure the differential signal with Q = V_{ref}/128 = 3.9 mV resolution



$$V_{ref} = 0.5\ V$$
$$Q = 3.9\, mV$$
$$V_{DIFF} = V_{in} - 4.5\, V$$

ADC Channels

- MCUs typically have internal ADCs
- Typically, the input(s) of the ADC can be internally connected to multiple external pins
- Example: the nRF2832 has 8 input channels
  - Labelled as AIN0-AIN7 in the figure
- Only one channel can be used at a time

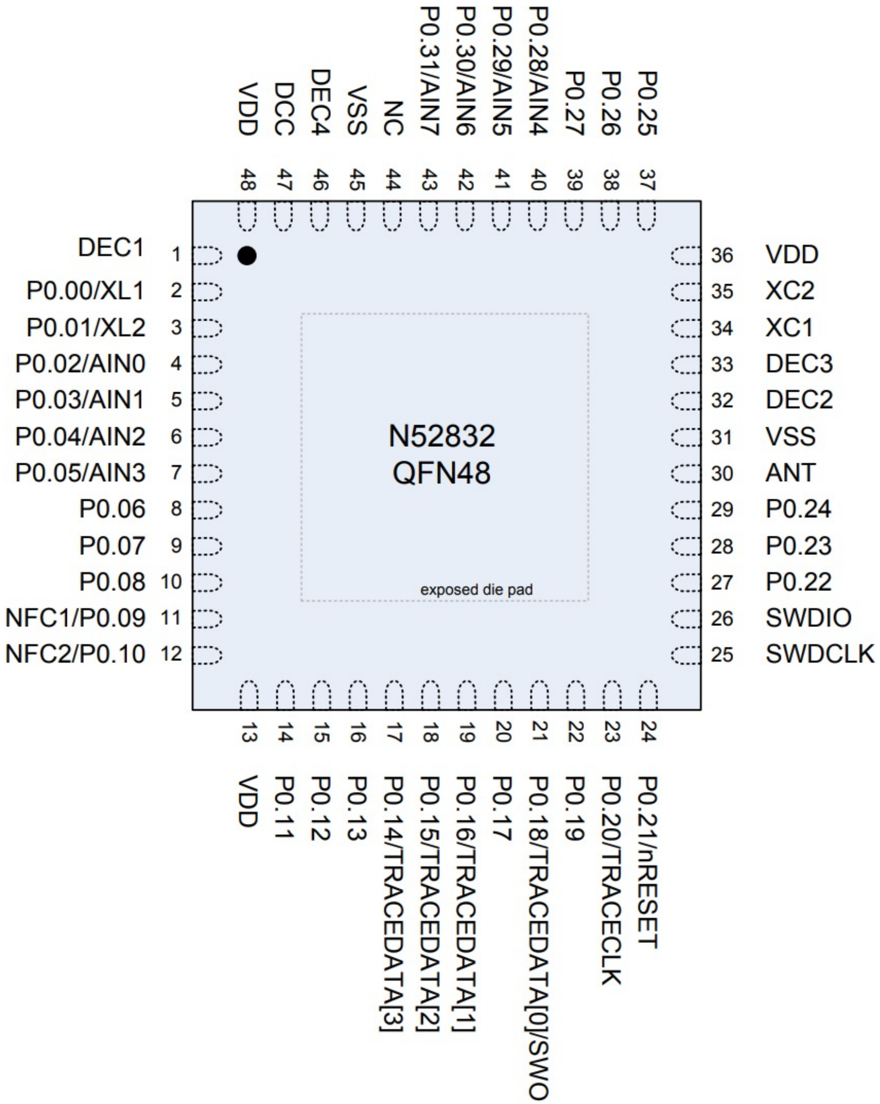


Image source: nRF52832 Datasheet by Nordic Semiconductors

ADC Modes of operation

- **One-shot mode:** a single sample is taken
- **Continuous mode:** samples are taken at a specified sampling frequency
  - Nyquist Theorem: to accurately represent a signal the sampling frequency should be double the highest frequency component of the input signal
- **Oversampling mode:** samples are taken at a higher than Nyquist frequency and then averaged out to reduce noise
- **Scan mode:** cycles through multiple input channels to measure multiple inputs in “parallel”

Digital-to-Analog Converter (DAC)

- A DAC converts a digital signal (series of numbers) into an analogue signal (voltage signal)
- An DAC essentially continuous signal out of discreet time data
- Applications
  - Audio
  - Video
  - Communications (wireless/optical)
  - Digitally-controlled potentiometer
  - …

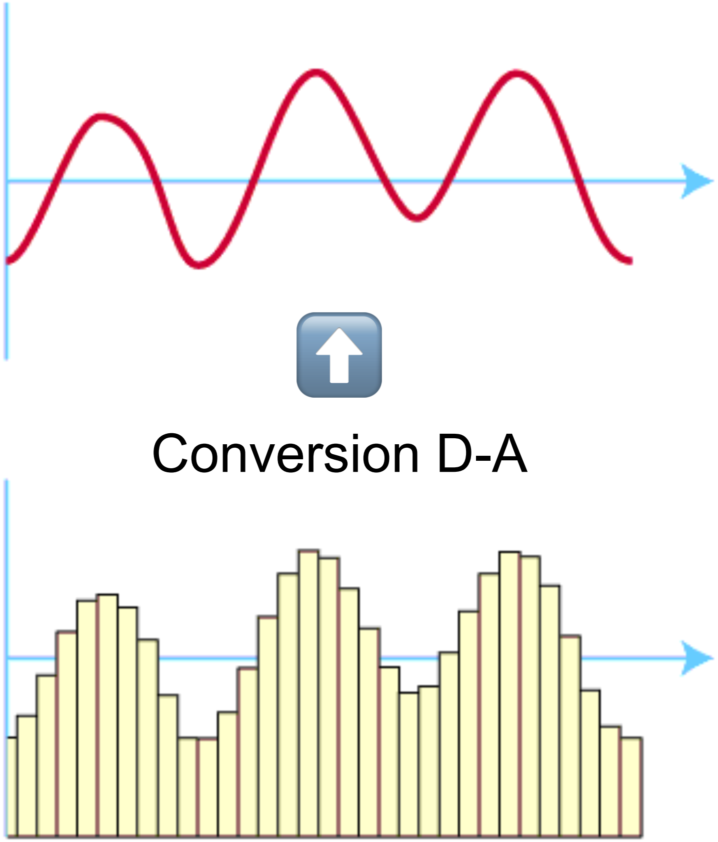
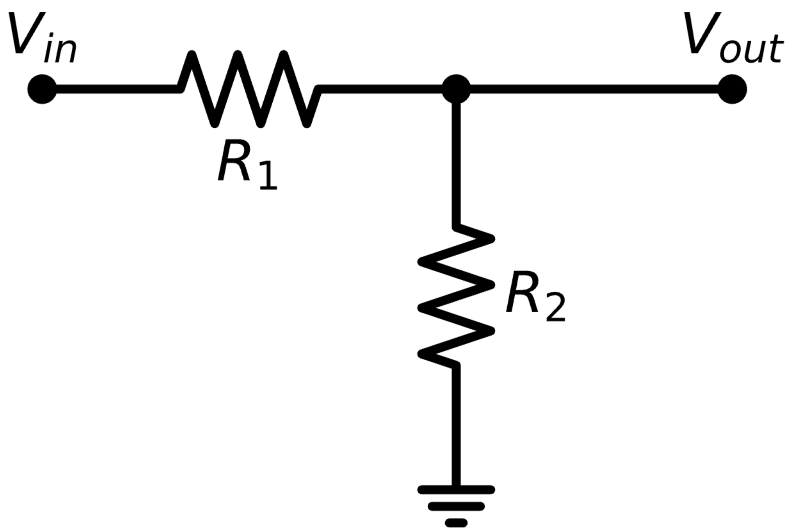


Image source: Wikipedia

Power Control

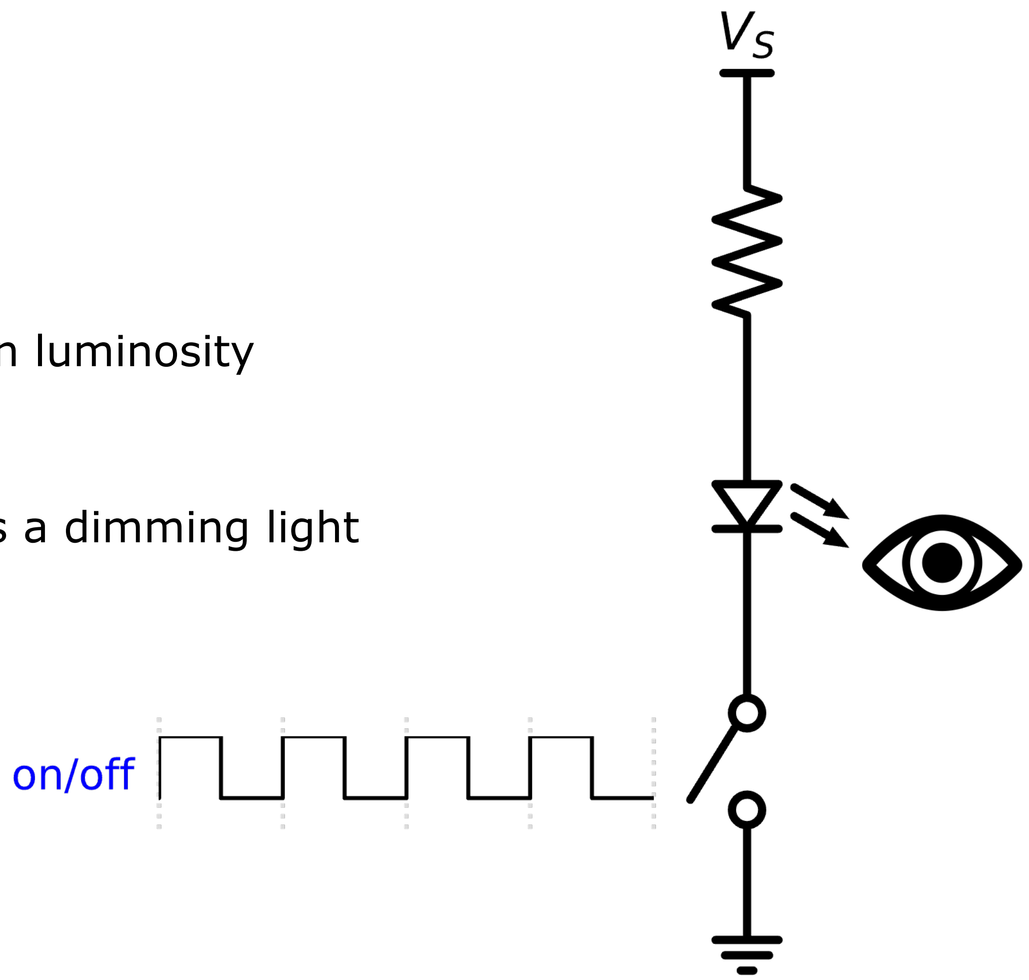
- Imagine R_2 is the load
- Imagine R_1 is a variable resistor that we use to control the power that goes to the load
- If R_1 << R_2, R_1 voltage drop negligible, power consumption of R_1 also negligible
- If R_1 >> R_2, R_1 current negligible, power consumption of R_1 also negligible
- If R_1 == R_2, R_1 consumes as much as power as the load

$$V_{out} = V_{in} \frac{R_2}{R_1 + R_2}$$



Dimming a LED

- Turn LED on/off at a high frequency
- Human eye is not able to capture frequent changes in luminosity
  - Has inertia or short-term memory
- Instead, it averages out the luminosity and perceives a dimming light
  - This is a low pass filter!



Pulse Width Modulation (PWM)

- An on/off signal with two properties
  - Frequency: repetitions per second
  - Duty cycle: on time over period
- MCUs can generate PWM signals using the GPIOs

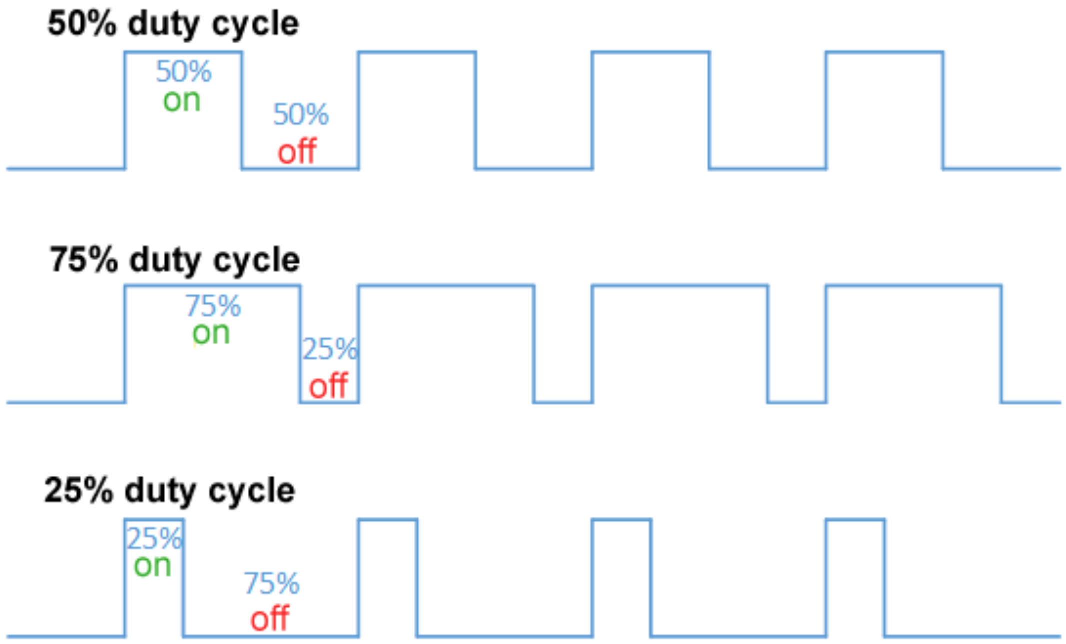
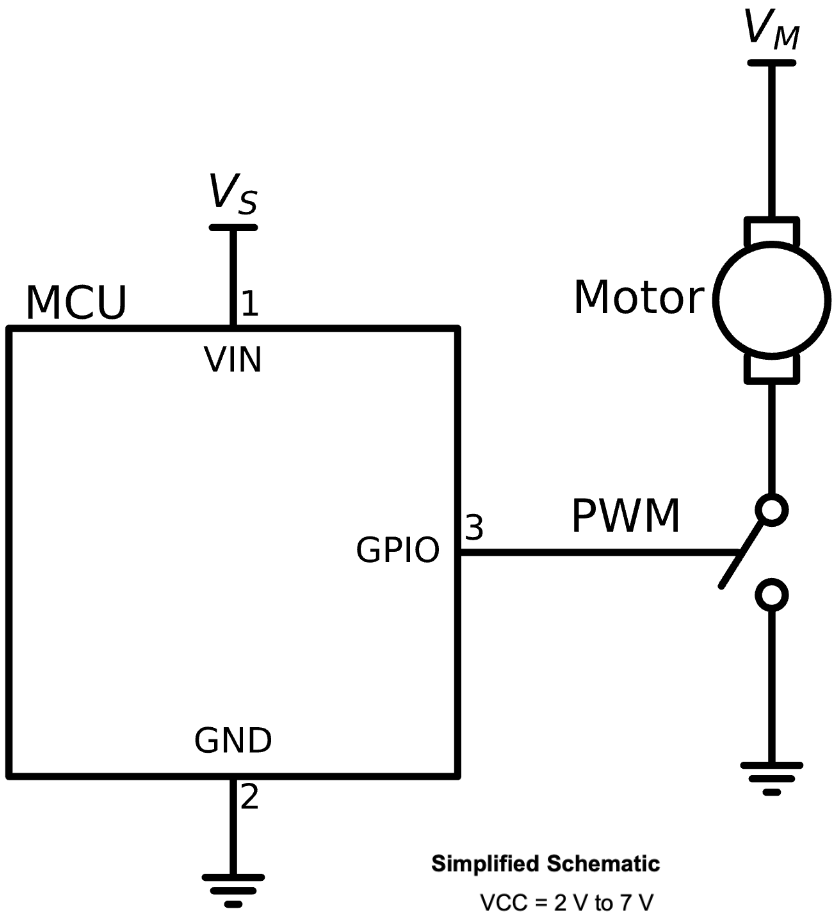


Image source: Wikipedia

PWM Motor Control

- The PWM duty cycle controls the power that goes to the motor
- Motors are inductive
  - They react slowly on changes in current
  - Current (and thus power) through is averaged out
- Why not power control with variable series resistor (transistor)?
  - Inefficient as resistor would consume energy
  - Less power would go to the motor
  - It is easier/cheaper to generate digital signals (PWM) than analogue signals



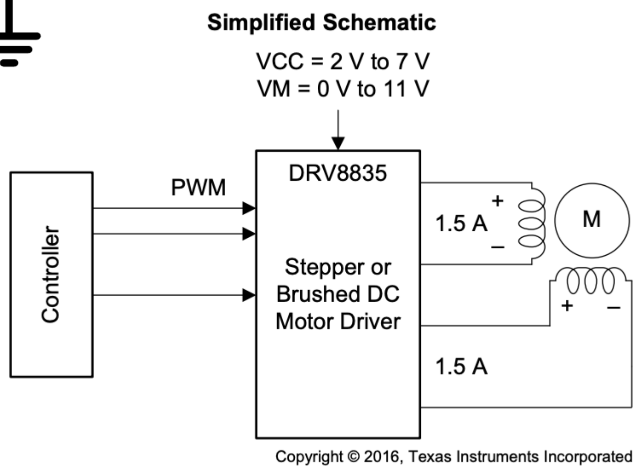


Image source: DRV8835 Datasheet from Texas Instruments