Analog Interfacing

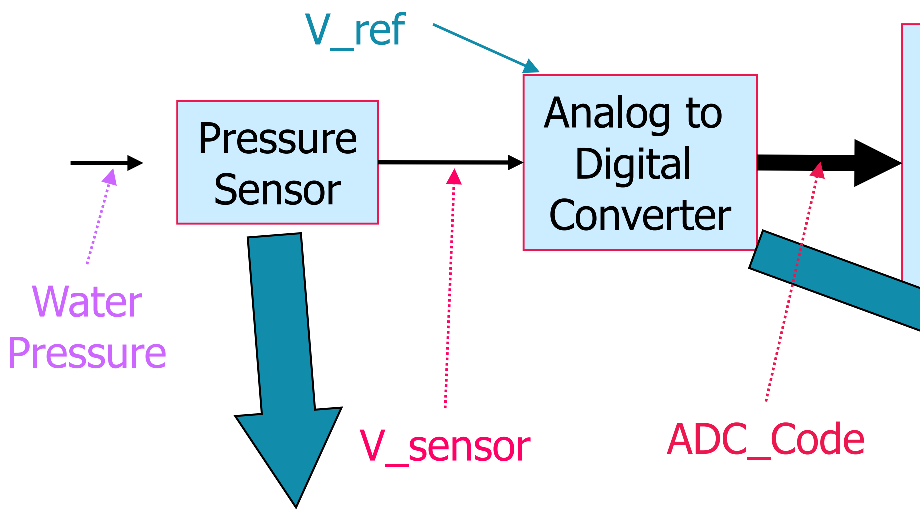
Why It's Needed

- Embedded systems often need to measure values of physical parameters
- These parameters are usually continuous (analog) and not in a digital form which computers (which operate on discrete data values) can process

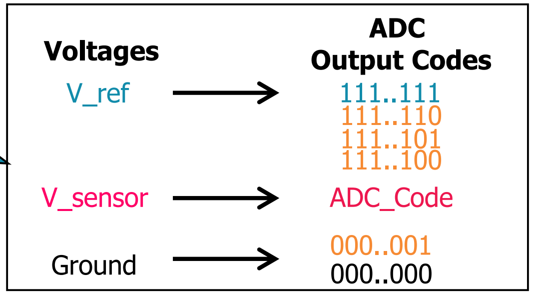
- Temperature
  - Thermometer (do you have a fever?)
  - Thermostat for building, fridge, freezer
  - Car engine controller
  - Chemical reaction monitor
  - Safety (e.g. microprocessor processor thermal management)
- Light (or infrared or ultraviolet) intensity
  - Digital camera
  - IR remote control receiver
  - Tanning bed
  - UV monitor
- Rotary position
  - Wind gauge
  - Knobs
- Pressure
  - Blood pressure monitor
  - Altimeter
  - Car engine controller
  - Scuba dive computer
  - Tsunami detector
- Acceleration
  - Air bag controller
  - Vehicle stability
  - Video game remote
- Mechanical strain
- Other
  - Touch screen controller
  - EKG, EEG
  - Breathalyzer

CONVERTING BETWEEN ANALOG AND DIGITAL VALUES

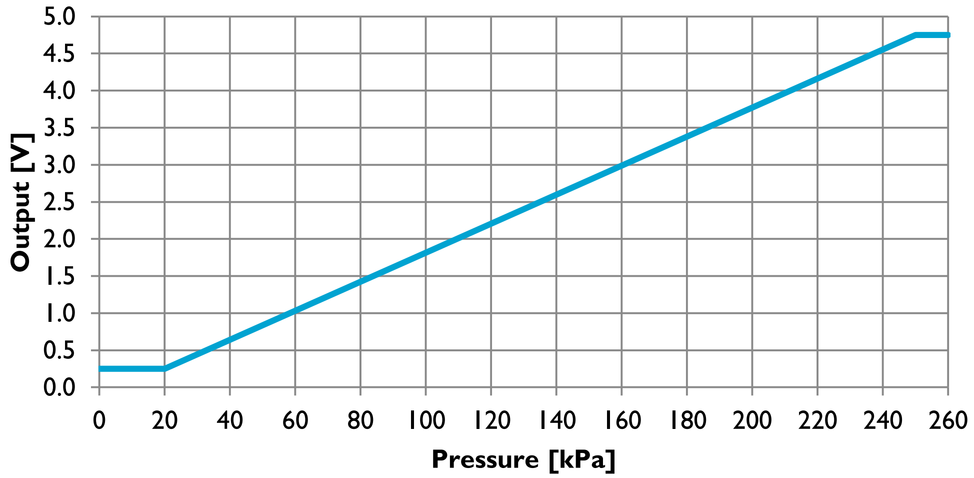
Example Analog Sensor - Depth Gauge




```
// Your software
ADC_Code = adc_read();
V_sensor = ADC_code * V_ref / ADC_MASK;
Pressure_kPa = 250 * (V_sensor / V_supply + 0.04);
Depth_ft = 33 * (Pressure_kPa – Atmos_Press_kPa) / 101.3;
```



Typical Absolute Pressure vs. Ouput

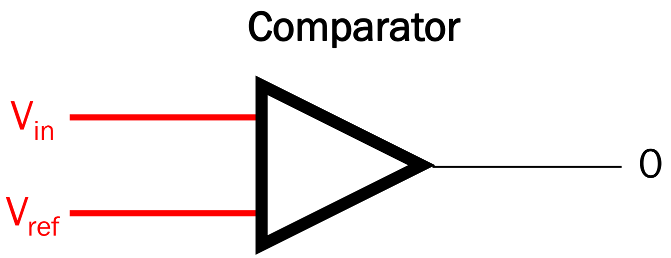


- Sensor detects pressure and generates a proportional output voltage V_sensor
- ADC generates a proportional digital integer (code) based on V_sensor and V_ref
- Code can convert that integer to a something more useful
  - first a float representing the voltage,
  - then another float representing pressure,
  - finally another float representing depth

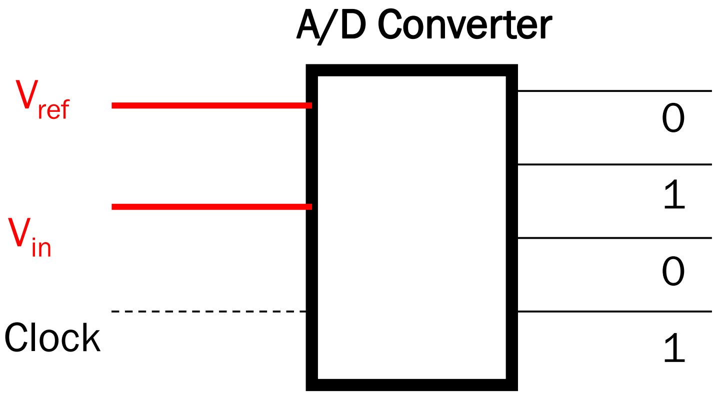
ARM

Getting From Analog to Digital

- A Comparator tells us "Is Vin > Vref?"
  - Compares an analog input voltage with an analog reference voltage and determines which is larger, returning a 1-bit number
  - E.g. Indicate if depth > 100 ft
  - Set Vref to voltage pressure sensor returns with 100 ft depth.

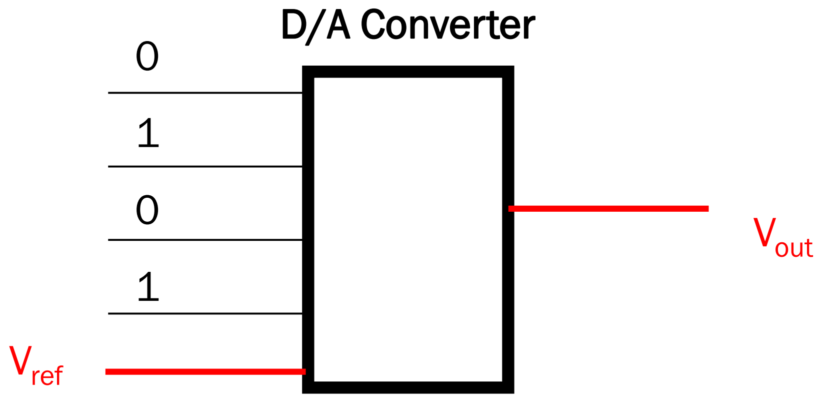


- An Analog to Digital converter [AD or ADC] tells us how large Vin is as a fraction of Vref.
  - Reads an analog input signal (usually a voltage) and produces a corresponding multi-bit number at the output.
  - E.g. calculate the depth

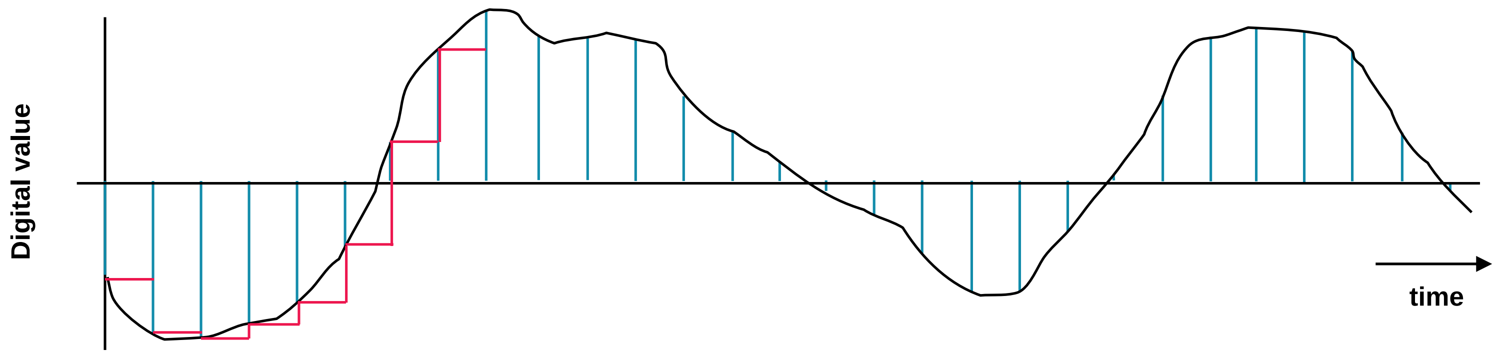


Digital to Analog Conversion

- May need to generate an analog voltage or current as an output signal
  - E.g. audio signal, video signal brightness.
- DAC: "Generate the analog voltage which is this fraction of Vref"
- Digital to Analog Converter equation
  - n = input code
  - N = number of bits of resolution of converter
  - Vref = reference voltage
  - Vout = output voltage. Either
    - Vout = Vref * n/(2N) or
    - Vout = Vref * (n+1)/(2N)
    - The offset +1 term depends on the internal tap configuration of the DAC – check the datasheet to be sure



Waveform Sampling and Quantization



- A waveform is sampled at a constant rate – every Δt
  - Each such sample represents the instantaneous amplitude at the instant of sampling
  - "At 37 ms, the input is 1.91341914513451451234311… V"
  - Sampling converts a continuous time signal to a discrete time signal
- The sample can now be quantized (converted) into a digital value
  - Quantization represents a continuous (analog) value with the closest discrete (digital) value
  - "The sampled input voltage of 1.91341914513451451234311… V is best represented by the code 0x018, since it is in the range of 1.901 to 1.9980 V which corresponds to code 0x018."

ARM

Forward Transfer Function Equations

What code n will the ADC use to represent voltage V_{in}?

General Equation

n = converted code

Vin = sampled input voltage

V+ref = upper voltage reference

V-ref = lower voltage reference

N = number of bits of resolution in ADC

$$n = \left\lfloor \frac{V_{in} - V_{-ref}}{V_{+ref} - V_{-ref}} 2^N + 1/2 \right\rfloor$$

Simplification with V_{-ref} = 0 V

$$n = \left\lfloor \frac{V_{in}}{V_{+ref}} 2^N + 1/2 \right\rfloor$$

$$n = \left\lfloor \frac{3.3V}{5V} 2^{10} + 1/2 \right\rfloor = 388$$

$$\lfloor X \rfloor = \mathrm{floor}(X)$$

floor(X) nearest integer I such that I <= X

floor(x+0.5) rounds x to the nearest integer

Inverse Transfer Function

What range of voltages V_{in_min} to V_{in_max} does code n represent?

General Equation

n = converted code

Vin_min = minimum input voltage for code n

Vin_max = maximum input voltage for code n

V+ref = upper voltage reference

V-ref = lower voltage reference

N = number of bits of resolution in ADC

$$V_{in\_min} = \frac{n - \frac{1}{2}}{2^N}\left(V_{+ref} - V_{-ref}\right) + V_{-ref}$$

$$V_{in\_max} = \frac{n + \frac{1}{2}}{2^N}\left(V_{+ref} - V_{-ref}\right) + V_{-ref}$$

Simplification with V_{-ref} = 0 V

$$V_{in\_min} = \frac{n - \frac{1}{2}}{2^N}\left(V_{+ref}\right)$$

$$V_{in\_max} = \frac{n + \frac{1}{2}}{2^N}\left(V_{+ref}\right)$$

What if the Reference Voltage is not known?

- Example - running off an unregulated battery (to save power)
- Measure a known voltage and an unknown voltage

$$V_{unknown} = V_{known} \frac{n_{unknown}}{n_{known}}$$

- Many MCUs include an internal fixed voltage source which ADC can measure for this purpose
- Can also solve for Vref

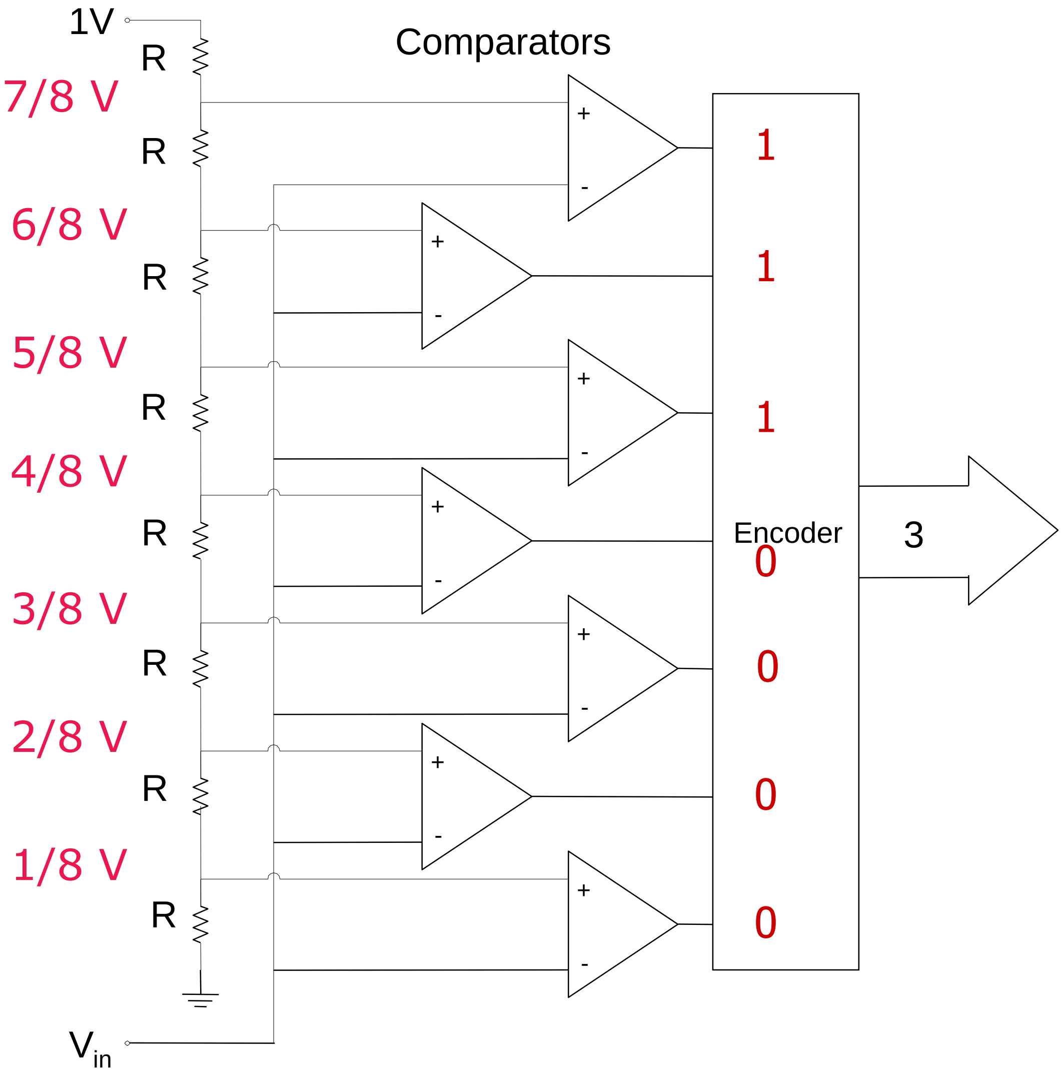
$$V_{ref} = V_{known} \frac{2^N}{n}$$

"My ADC tells me that channel 27 returns a code of 0x6543, so I can calculate that V_{REFSH} = 1.0V * 2^{16}/0x6543 = …

ANALOG TO DIGITAL CONVERSION CONCEPTS

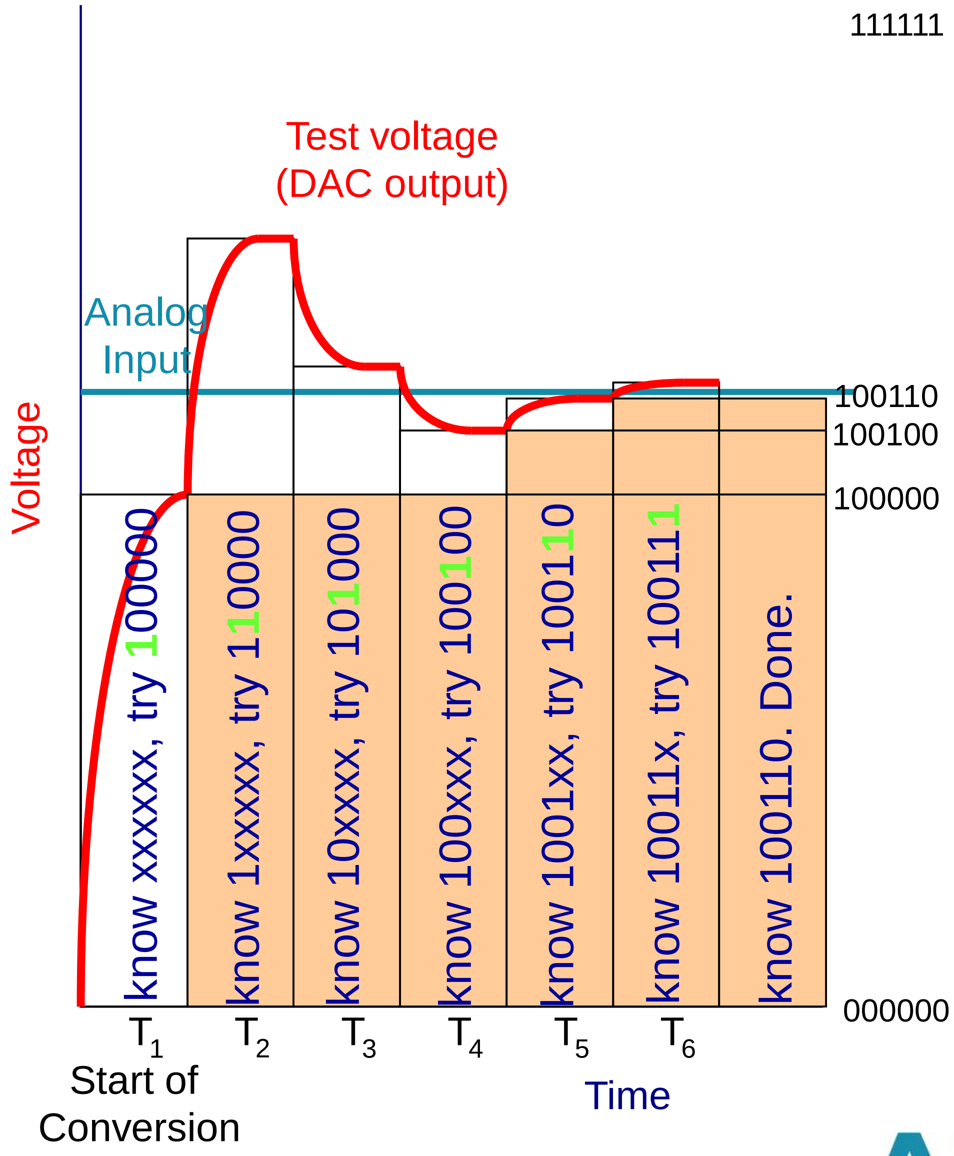
A/D – Flash Conversion

- A multi-level voltage divider is used to set voltage levels over the complete range of conversion.
- A comparator is used at each level to determine whether the voltage is lower or higher than the level.
- The series of comparator outputs are encoded to a binary number in digital logic (a priority encoder)
- Components used
  - 2N resistors
  - 2N-1 comparators
- Note
  - This particular resistor divider generates voltages which are not offset by ½ bit, so maximum error is 1 bit
  - We could change this offset voltage by using resistors of values R, 2R, 2R ... 2R, 3R (starting at bottom)



ADC - Successive Approximation Conversion

- Successively approximate input voltage by using a binary search and a DAC
- SA Register holds current approximation of result
- Set all DAC input bits to 0
- Start with DAC's most significant bit
- Repeat
  - Set next input bit for DAC to 1
  - Wait for DAC and comparator to stabilize
  - If the DAC output (test voltage) is smaller than the input then set the current bit to 1, else clear the current bit to 0



ADC Performance Metrics

- Linearity measures how well the transition voltages lie on a straight line.
- Differential linearity measure the equality of the step size.
- Conversion time: between start of conversion and generation of result
- Conversion rate = inverse of conversion time

Sampling Problems

- Nyquist criterion
  - Fsample >= 2 * Fmax frequency component
  - Frequency components above ½ Fsample are aliased, distort measured signal

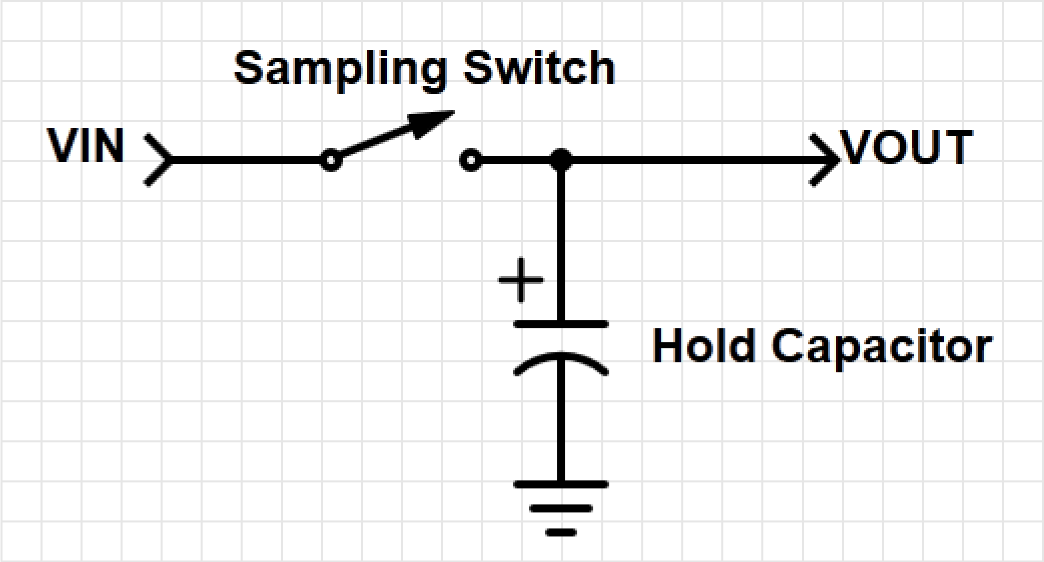
- Nyquist and the real world
  - This theorem assumes we have a perfect filter with "brick wall" roll-off
  - Real world filters have more gentle roll-off
  - Inexpensive filters are even worse (e.g. first order filter is 20 dB/decade, aka 6 dB/octave)
  - So we have to choose a sampling frequency high enough that our filter attenuates aliasing components adequately

Inputs

- Differential
  - Use two channels, and compute difference between them
  - Very good noise immunity
  - Some sensors offer differential outputs (e.g. Wheatstone Bridge)
- Multiplexing
  - Typically share a single ADC among multiple inputs
  - Need to select an input, allow time to settle before sampling
- Signal Conditioning
  - Amplify and filter input signal
  - Protect against out-of-range inputs with clamping diodes

Sample and Hold Devices

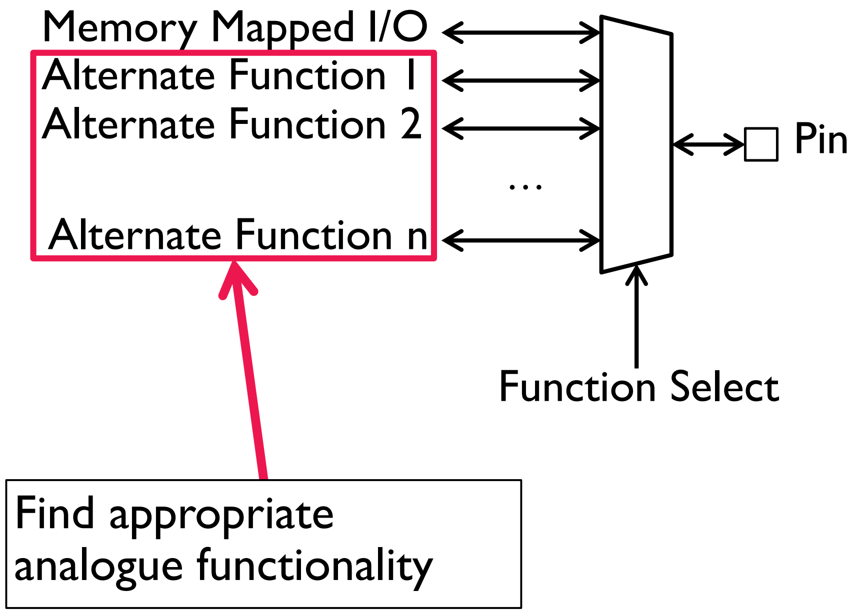
- Some A/D converters require the input analog signal to be held constant during conversion, (e.g. successive approximation devices)
- In other cases, peak capture or sampling at a specific point in time necessitates a sampling device.
- This function is accomplished by a sample and hold device as shown to the right
- These devices are incorporated into some A/D converters



ANALOG INTERFACING PERIPHERALS

GPIO Alternative Functions

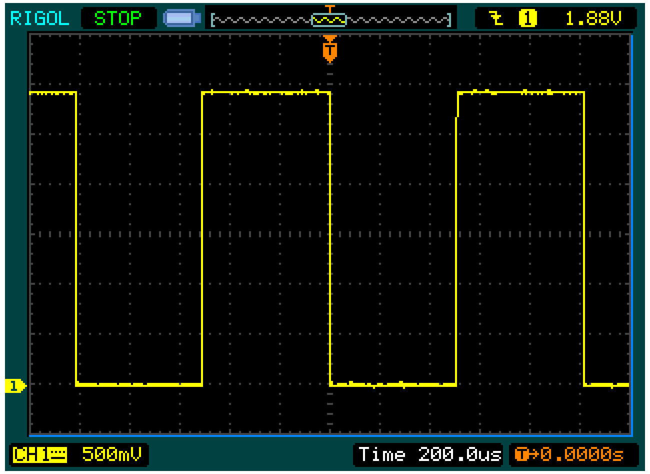
- Pins may have different features
- To enable an alternative function, set up the appropriate register
- May also have analogue paths for ADC / DAC etc.
- Advantages:
  - Saves space on the package
  - Improves flexibility

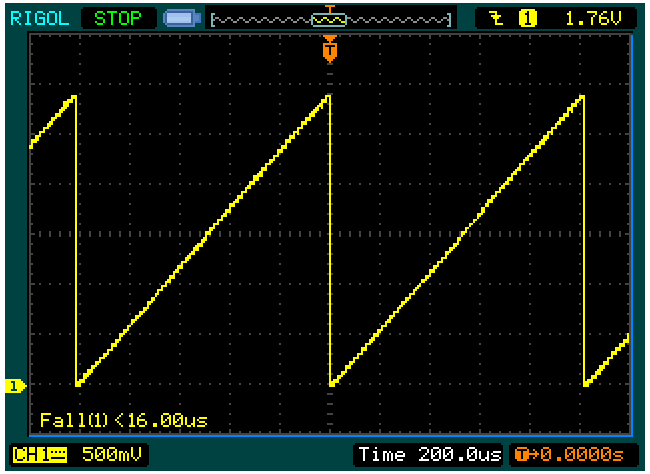


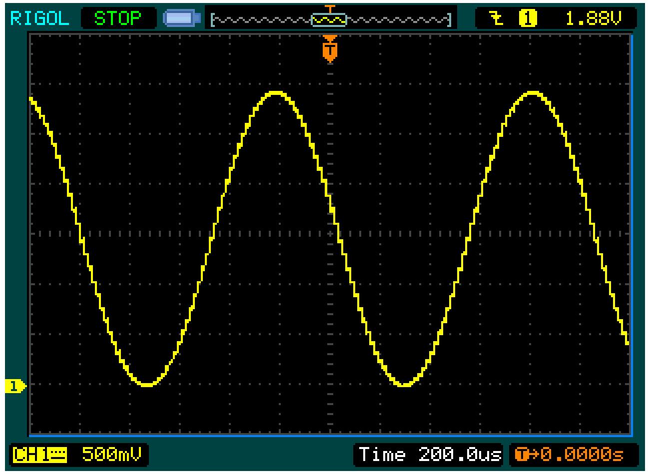
ARM

DIGITAL TO ANALOG CONVERTER

Example: Waveform Generation







- DAC can be used to generate arbitrary waveforms
  - Pregenerate lookup table
  - Update DAC output value
  - Delay
  - Repeat

ARM

C Code – Initialisation

```
void tone_init(void) {
	dac_init();
	sinewave_init();
}
void sinewave_init(void) {
	int n;
	for (n = 0; n < NUM_STEPS; n++) {
		sine_table[n] = MAX_DAC_CODE * (1 + sin(n*2*PI/NUM_STEPS)) / 2;
	}
}
```

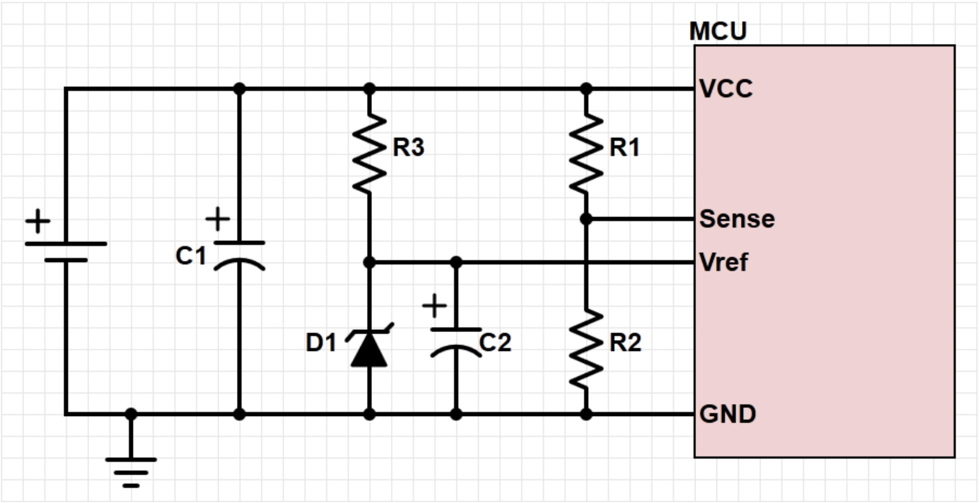
C Code – Playback

```
void tone_play(int period_us, int num_cycles, wavetype wave) {
        int sample, step;
        while(num_cycles-- > 0) {
                for (step = 0; step < NUM_STEPS; step++) {
                        switch(wave) {
                                case SINE: sample = sine_table[step]; break;
                                case SQUARE: sample = step < NUM_STEPS / 2 ? 0 : MAX_DAC_CODE;
                                        break;
                                case RAMP: sample = (step * MAX_DAC_CODE) / NUM_STEPS; break;
                        }
                        dac_set(sample);
                        delay_us(period_us);
                }
        }
}
```


ARM

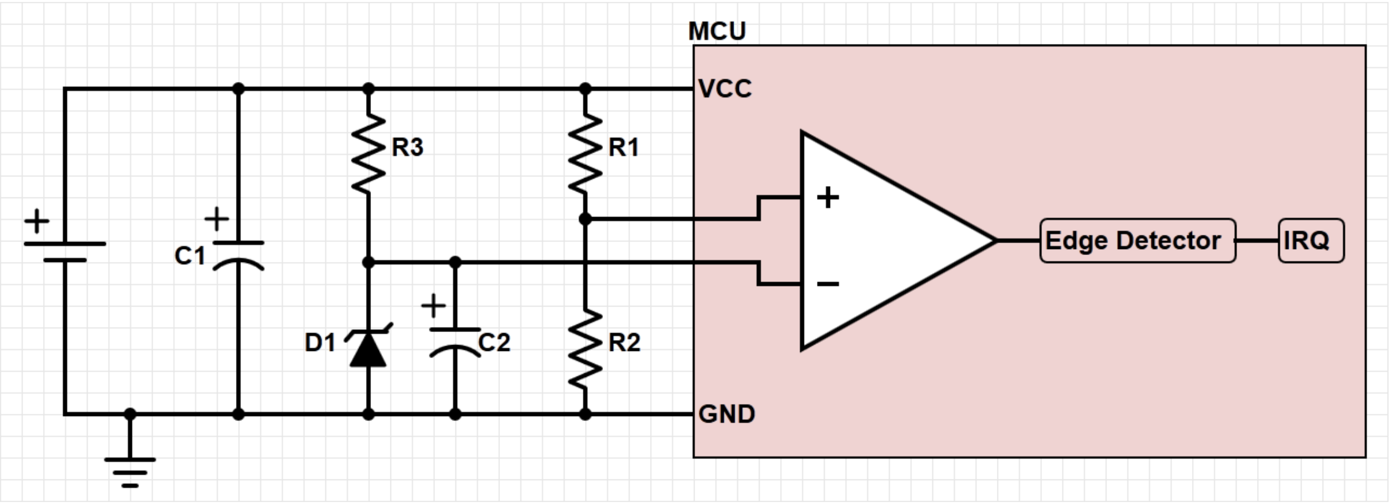
ANALOG COMPARATOR

Example: Power Failure Detection



- Need warning of when power has failed
  - Very limited amount of time before capacitor C1 discharges
  - Save critical information
  - Turn off output devices
  - Put system into safe mode
- Can use a comparator to compare Vcc against a fixed reference voltage Vref

Comparator Overview



- Comparator compares Sense and Vref
- Comparator output indicates if Sense > Vref (1) or Sense < Vref (0)
- Can generate an interrupt request (+, -, or +- edges)
- If the Sense input drops below Vref, fire an interrupt

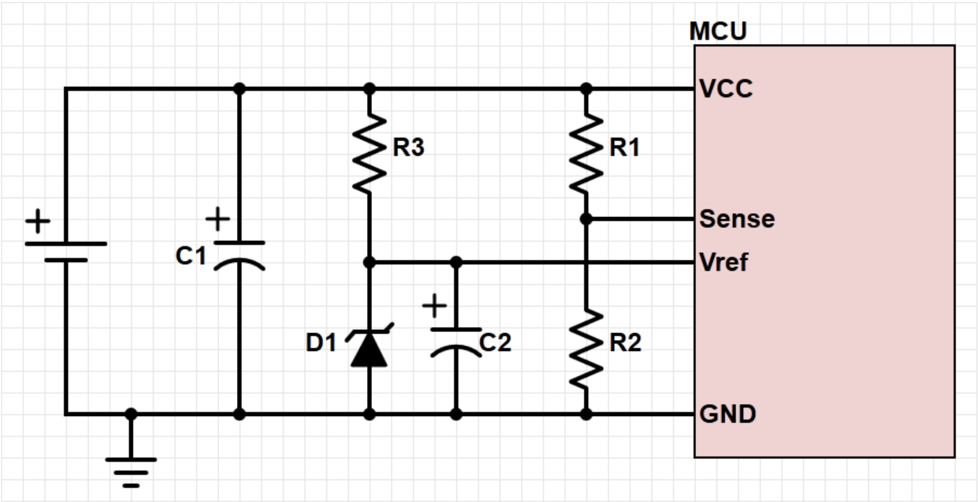
C Code – Comparator

```
void comparator_isr(int state) {
        if (state) {
                // Sense > Vref, turn off LEDs.
                leds_set(0, 0, 0);
        } else {
                // Sense < Vref, turn on red LED.
                leds_set(1, 0, 0);
        }
}

int main(void) {
        comparator_init();
        comparator_set_trigger(CompBoth); // ISR on both rising and falling edges.
        comparator_set_callback(comparator_isr);
        …
}
```

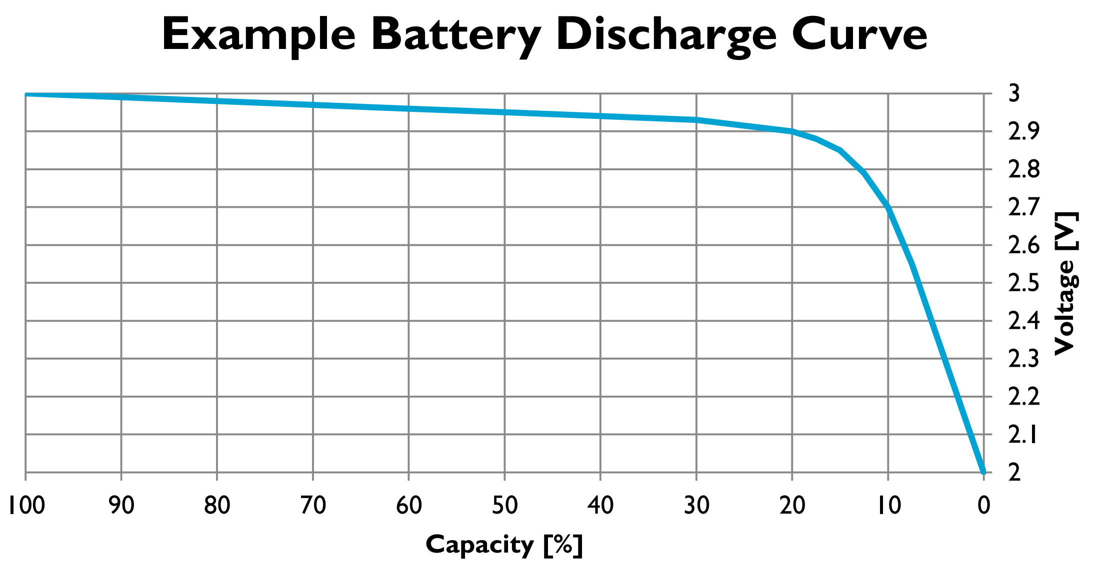
ANALOG TO DIGITAL CONVERTER

Example: Battery Monitoring



- Operates similar to comparator based system
- Measures battery voltage, better indication of battery life than comparator
- Can provide information about battery discharge rate

Battery Discharge



- As the battery discharges, the voltage decreases
- Measure with respect to Vref (fixed voltage reference)
- Convert to capacity with look-up table and interpolation