Interfacing with Analogue Signals

Dongbing Gu

School of Computer Science and Electronic Engineering
University of Essex
UK

Spring 2018

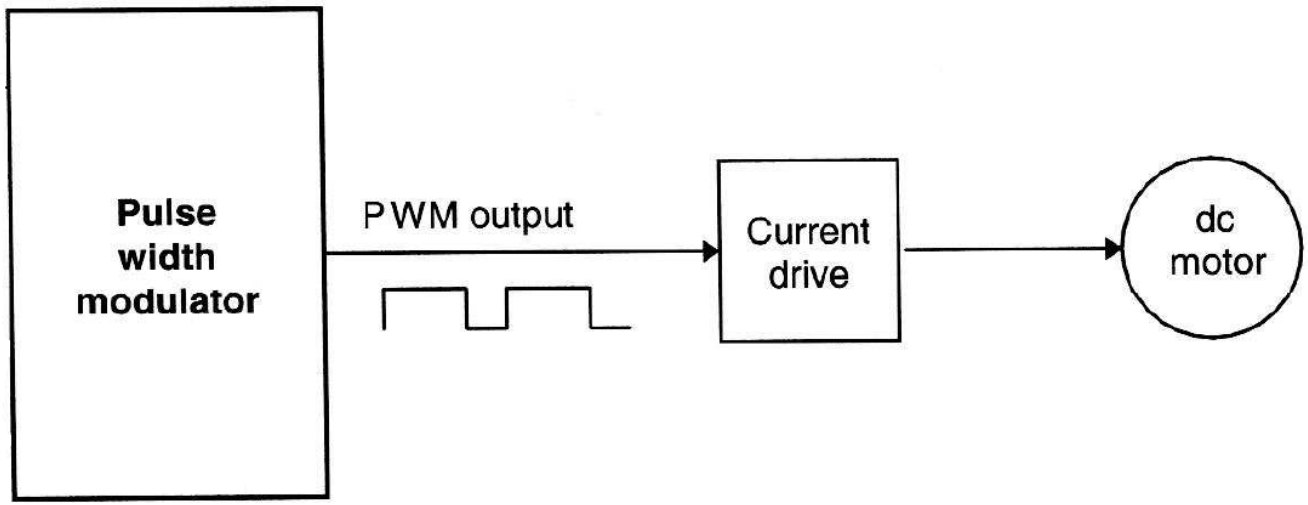
Outline

Section 1

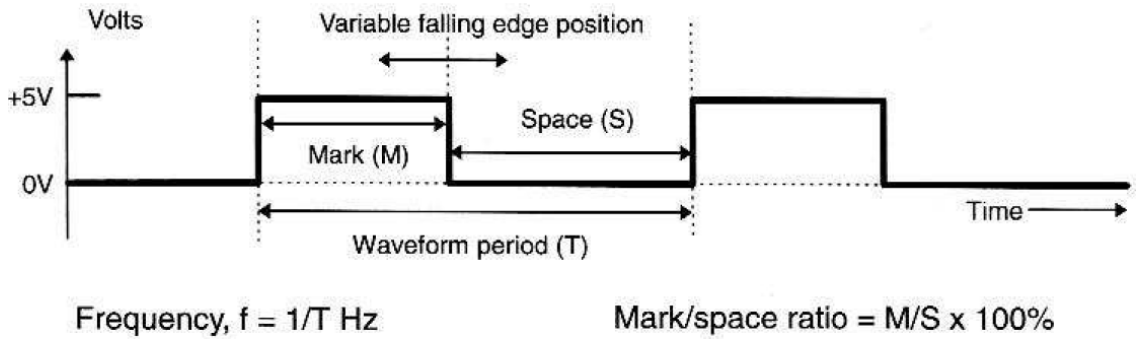
Pulse Width Modulation (PWM)

Pulse Width Modulation (PWM)

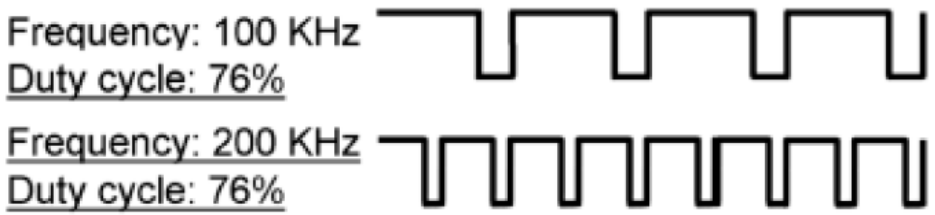
- This is a way to vary the average voltage applied to a DC motor.
- The speed of a DC motor depends on the voltage/current applied to it.
- You switch the power to the motor On and OFF and also vary the ratio of ON time to OFF time
- This has the effect of varying the average voltage applied to the motor and hence the average current through the motor.



Duty Cycles



- PWM controls DC motor speed by driving the motor with short pulses.
- These pulses vary in duration to change the speed of the motor. The longer the pulses, the faster the motor turns, and vice versa
- Duty Cycle is the ON time as a percentage of each cycle.



PWM Applications

- DC motors can be controlled by a variable resistor or variable resistor connected to a transistor. It generates heat and hence wastes power.
- PWM varies the power delivered to a load without the losses normally incurred when a power source drops its output voltage by using resistors.
- PWM has a number of uses: Motor speeds, Brightness of lights, Digital to Analogue conversion.

PWM on the mbed

- The mbed has up to six PWM outputs, on pins 21 to 26.
- They all share the same period timer, if you change the period of one output, you change them all.
- The pulsewidth can be set independently for each output
- Four of them are used for LED1-LED4 (p26,p25,p24,p23).

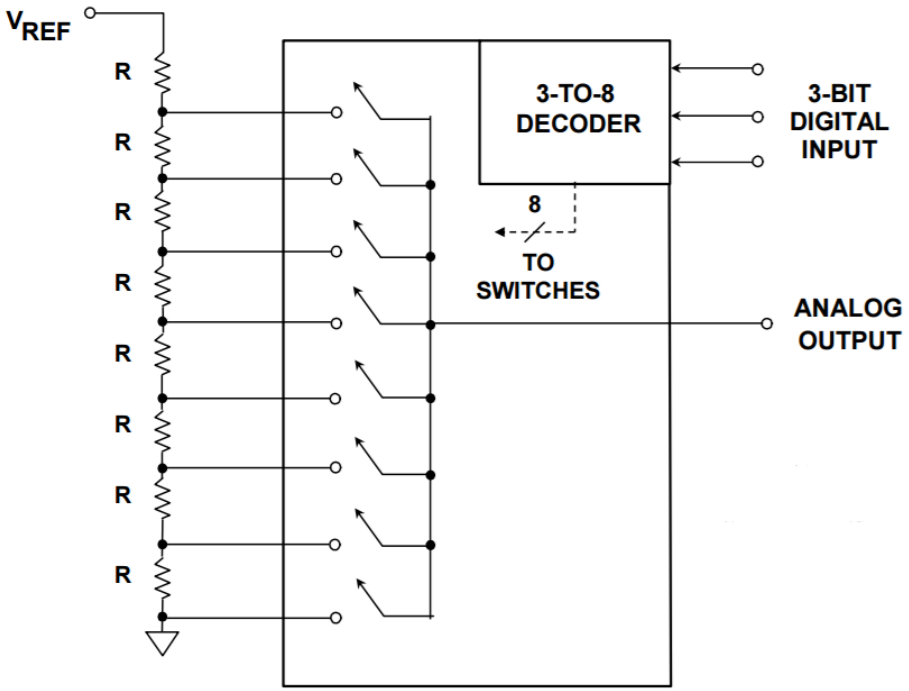
```
#include "mbed.h"

PwmOut led(LED1);

int main() {
    // specify period first
    led.period(4.0f);       // 4 second period
    led.write(0.50f);       // 50% duty cycle
    //led = 0.5f;           // shorthand for led.write()
    //led.pulsewidth(2);    // set duty cycle time in seconds
    while(1);
}
```

Section 2

DA Convertors

String Resistor DACs

- DAC converts a binary integer to an analogue voltage.
- **String resistor** ladders are popular.
- An n-bit DAC consists of 2^n equal resistors and 2^n switches.
- The output is taken by closing just one of the switches.
- Expensive to make one with large number of bits n.



DAC Resolution

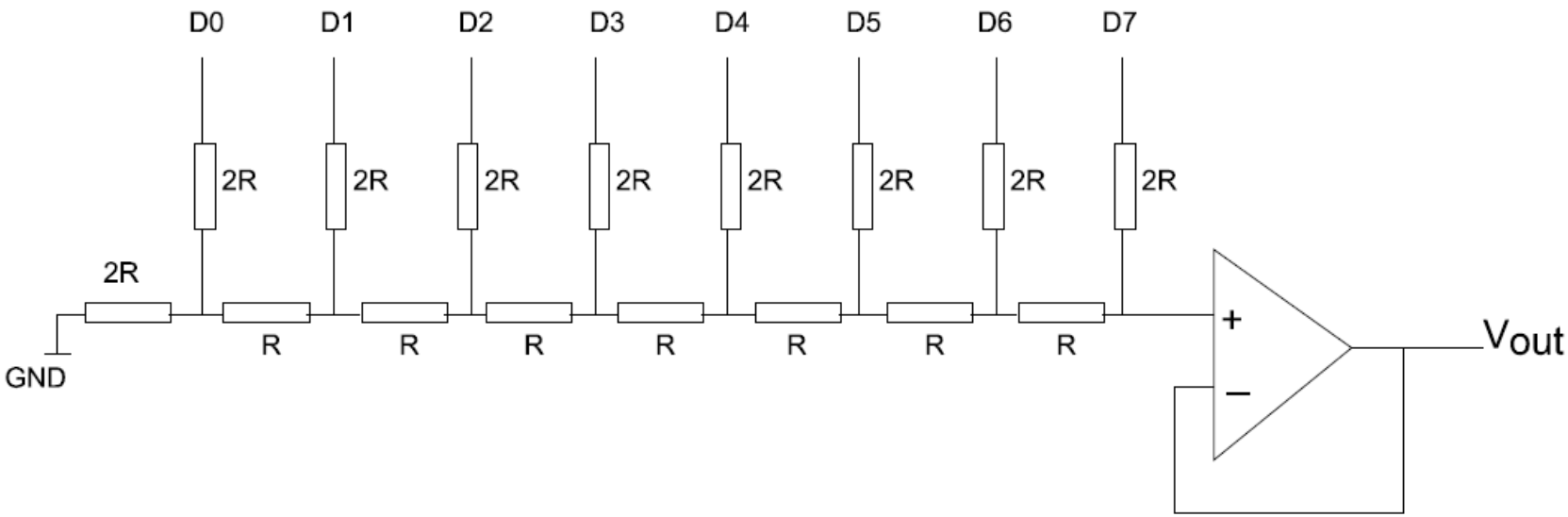
- The digital input is D.
- The number of bits in D is n.
- A voltage reference V_r is a precise and stable voltage.
- The analogue output is V_o

$$V_o = \frac{D}{2^n} V_r$$

- The number of possible values is 2^n and the step size or the **resolution** is $V_r/2^n$.
- An 6 bit DAC has 64 possible output values and the resolution is $50mV$ if the voltage reference is $3.2V$.

R-2R Resistor Ladders

- The **R-2R resistor** ladder involves two resistor values, but one must be exactly double the other.
- For an n bit conversion, the R-2R ladder needs $2n$ resistors.



DAC Problems

- **Offset error**: the actual output voltage differs from the ideal by some constant amount (an added or subtracted amount).
- **Gain error**: the actual output voltage differs from the ideal by some constant factor (a multiplier).
- **Non-linearity**: the input/output curve is not a straight line.
- The first two problems can be solved using adjuster circuits. To vary the gain or offset in the output signal.
- The last one is impossible to fix. Avoid them by buying more expensive DACs.

Applications

- The DAC is at the heart of every audio CD and MP3 player. They commonly use 24-bit DACs.
- The DAC in a CD player will be capable of around 96,000 samples per second.
- Also it does two separate channels at the same time for stereo sound, i.e. two DAC streams.

DAC on the mbed

- The LPC1768 chip has a 10-bit resistor string DAC on pin 18.
- Its own 3.3 V power supply as voltage reference.
- AnalogOut class member functions include write(), write_u16, read, and operator =

```
#include "mbed.h"

AnalogOut Aout(p18);
float i;

int main() {
  while(1){
    for (i=0;i<1;i=i+0.1){
    Aout=i; //or Aout.write(i);
    wait(0.001);
    }
  }
}
```

Section 3

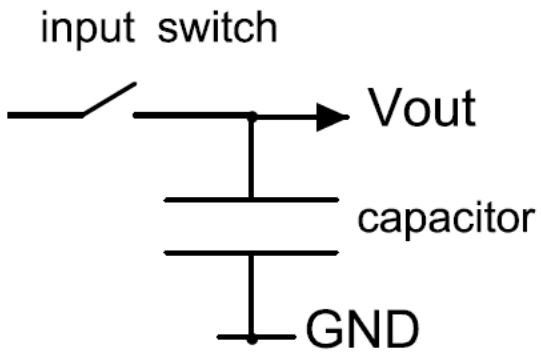
AD Converters

Signal Conditioning

- This is much widely used than DAC in embedded systems because monitoring the analogue environment requires ADC.
- ADC is much more difficult than DAC.
- For a start, the nature of the real world means that real quantities are never static - they vary constantly.
- A real world quantity is converted into a voltage by a suitable transducer (sensor).
- An amplifier processes this input to make it the right size for the chosen ADC, and removes (filters) all the random noise and other unwanted high frequencies from the input (signal conditioning).

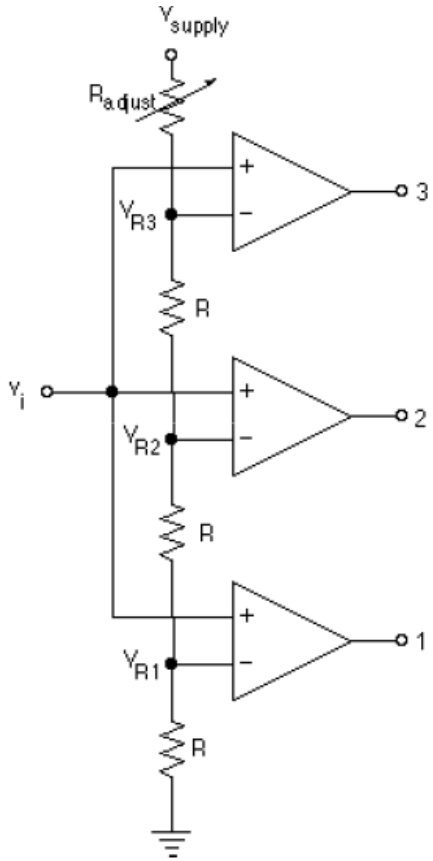
Keep Still While Converting

- Accurate analogue to digital conversion takes time, and it also requires an input that stays constant during the conversion.
- You can sample the input voltage at a certain moment, and hold it steady during the conversion
- This is done by a sample and hold circuit, which is a little capacitor and an input switch.
  - Close the switch - charge the capacitor
  - Open the switch - use ADC to measure the voltage on the capacitor



A two bit flash ADC

- R_{adjust} is set so that $V_{R3} = 3V$, $V_{R2} = 2V$, and $V_{R1} = 1V$.
- A comparator gives a HIGH output if the input voltage is greater than its reference voltage.
- If a comparator reads HIGH, then all those below it must also be HIGH. The number of comparators reading high will be 0, 1, 2, or 3.
- Additionally a combinational logic circuit is used to convert the comparator outputs to binary: (00), (01), (10), and (11).
- A voltage between 1V and 2V will be represented as (01).



Flash ADC

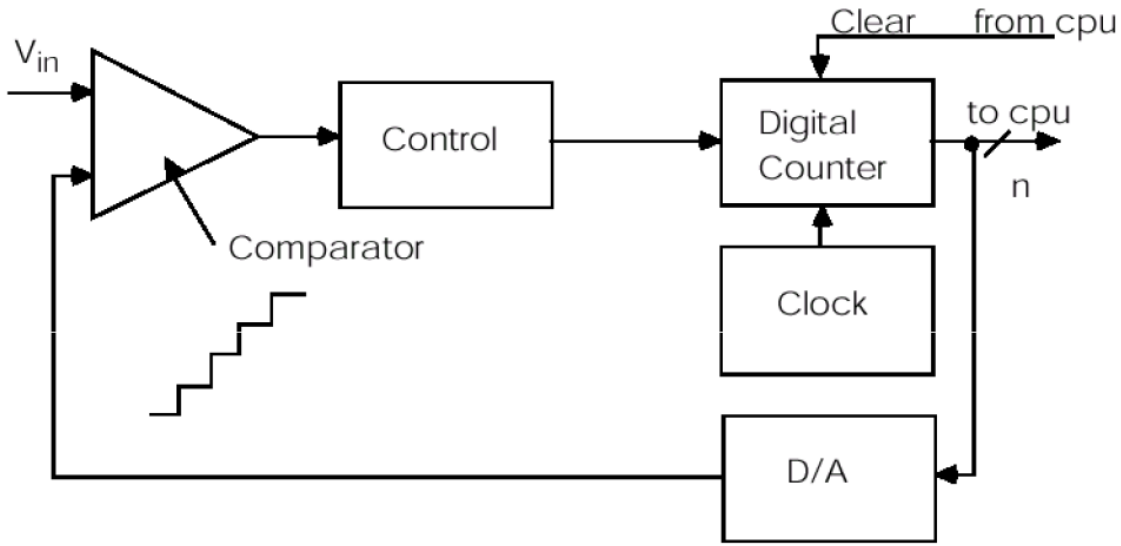
- A standard reference voltage is divided into a large number of equal voltage steps, and each of these voltages is supplied to a voltage comparator.
- The other input of each voltage comparator is connected to the input voltage from the sample.
- In parallel, each comparator yields a 0 or 1, depending on whether the input voltage is greater than their individual reference voltage.
- The outputs from the comparator go to a combinational logic circuit, which counts the 1s, and outputs that number in binary.

Flash ADC

- This type of ADC is simple in concept and extremely fast - up to 30 million samples per second.
- But every added bit means doubling the number of comparators, so they're rarely seen over 8 or 10 bits, and even 6 bit flash converters are not cheap.

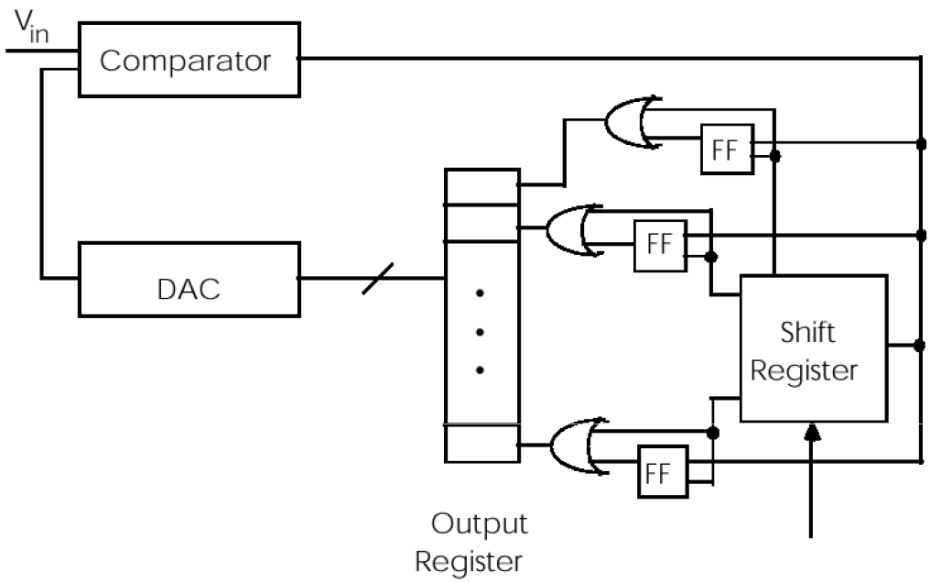
Sequential Counter (binary ramp) ADC

- The simplest type using a DAC is the sequential counter.
- A counter generates an increasing binary number, which is converted by a DAC
- And compared with the input voltage.
- When the comparator switches, the input voltage corresponds to the count.
- Easy! Cheap! But can take 2^n time steps. why?



Successive Approximation ADC

- Instead of a binary count, set one bit at a time and compare the voltage it produces. The result is a BINARY SEARCH for the unknown voltage level.
- The logic circuit arrangement on the right is called the SAR, or successive approximation register.
- The output register is converted to a voltage by the DAC, and tested against the input voltage by the comparator.



Successive Approximation ADC

- If the input voltage is higher, the bit is left set. If not, it is cleared.
- This is repeated for each bit, starting at the MS-bit end of the n-bit register.
- The conversion is thus guaranteed in n time steps. Successive approximation converters are therefore fairly fast.
- They are also reasonably cheap, and reasonably accurate. They are available up to 16 bits.
- A drawback is their susceptibility to noise.

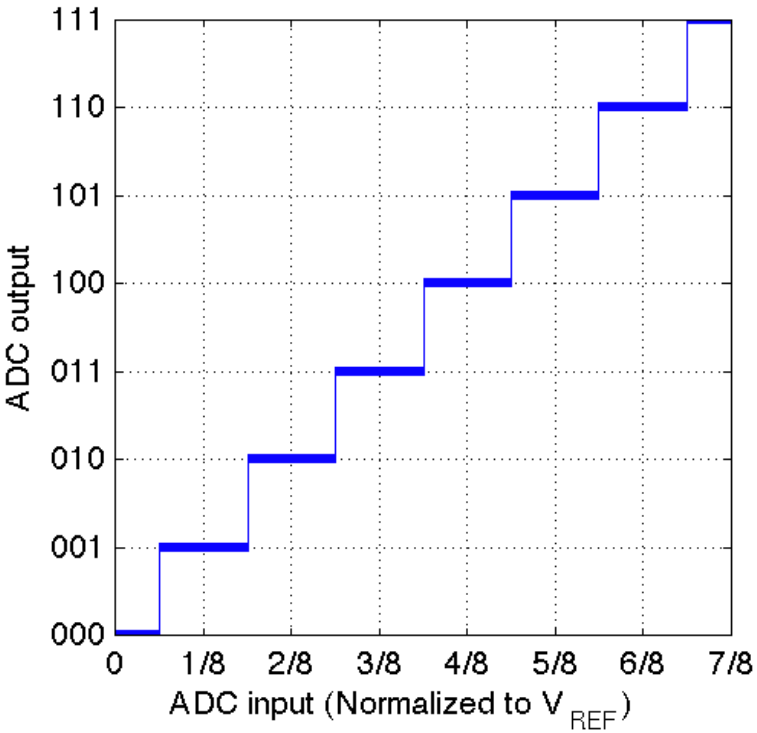
ADC Range and Quantisation

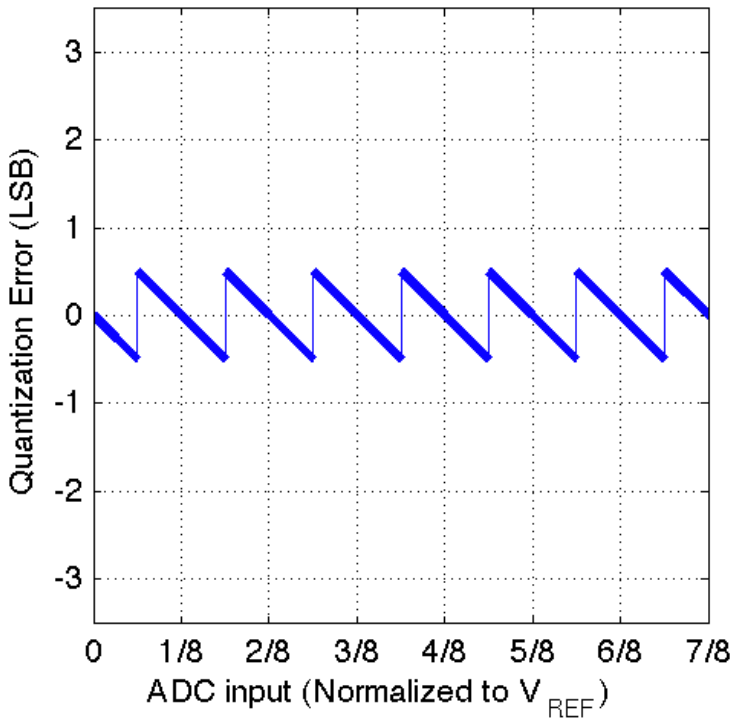
- The analogue input voltage is V_i
- The voltage reference is V_o
- The digital output is D
- The number of bits in D is n
- D can take any value from 0 to $(2^n - 1)$

$$D = \frac{V_i}{V_r} 2^n$$

- The converted digital output is an approximation to the analogue voltage.

ADC Outputs


111
110
101
100
011
010
001
000
ADC output
0
1/8
2/8
3/8
4/8
5/8
6/8
7/8
ADC input (Normalized to V REF)


3
2
1
0
-1
-2
-3
Quantization Error (LSB)
0
1/8
2/8
3/8
4/8
5/8
6/8
7/8
ADC input (Normalized to V REF)

ADC Quantisation

- The least significant bit (LSB) voltage or the voltage resolution is

$$V_{lsb} = \frac{V_{REF} - V_{AGND}}{2^n}$$

- The maximum quantisation error is defined as half of V_{lsb}. Increase the number of bits n can reduce this error.

$$\frac{1}{2}\frac{V_{REF} - V_{AGND}}{2^n}$$

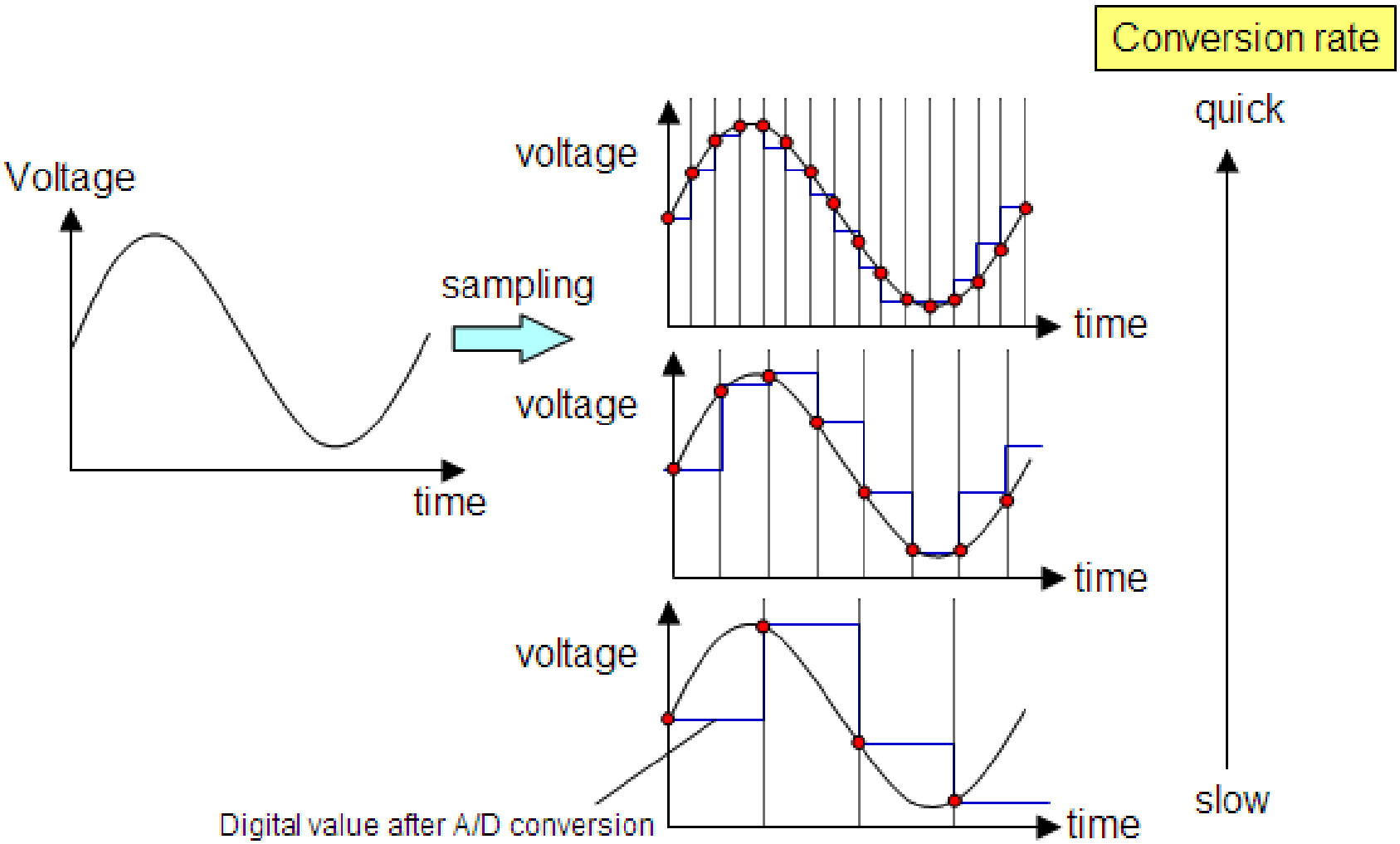
- An analogue signal has a range of 0-3.3V. For an 8-bit ADC, the maximum quantisation error is

$$\frac{1}{2} \times \frac{3.3}{2^8} = 6.45mV$$

ADC Sampling frequency

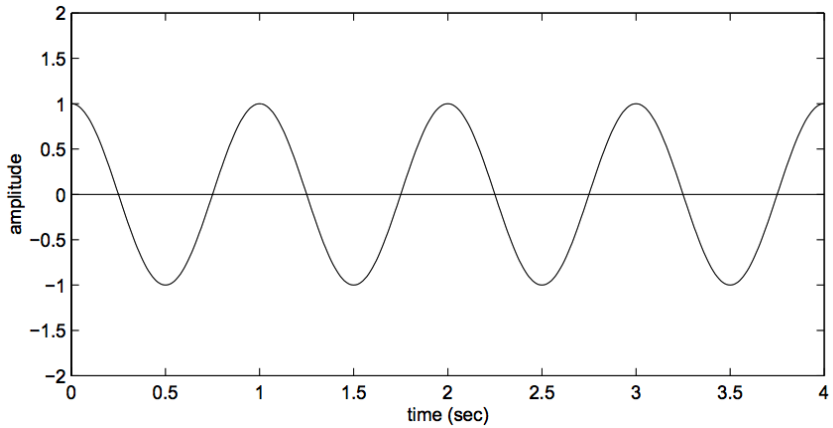
- When converting an analogue signal to digital, the ADC repeatedly takes a "sample" and convert this to the digital data.
- The more samples taken, the more accurate the digital data will be. Samples are normally taken at fixed periods and define the rate of sampling by the sampling frequency (the number of samples taken per second).
- If the sample frequency is too low then rapid changes in the analogue signal may not be obvious in the resulting digital data.

8-bit ADC Outputs

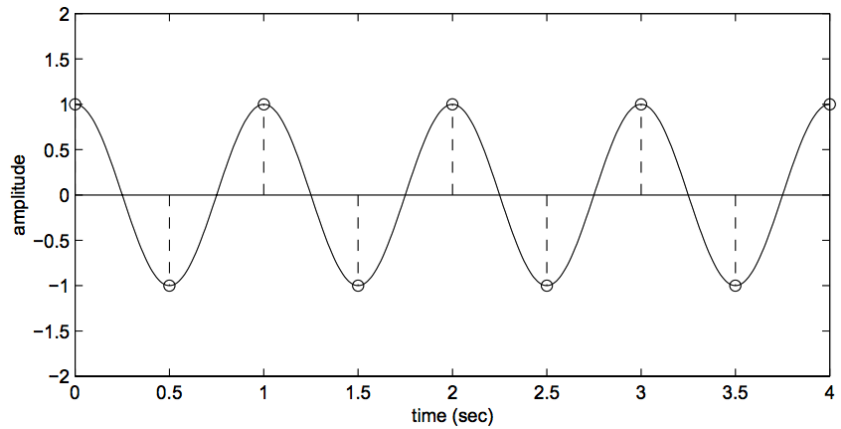


Nyquist Sampling Theorem

- The Nyquist sampling theorem states that the sampling frequency must be at least twice the highest frequency contained in the signal.
- A single sinewave at a frequency of 1 Hz:

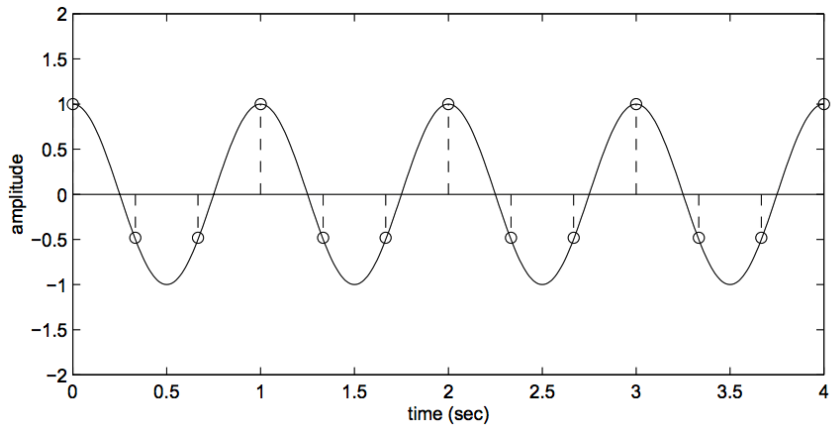


- If sampling it at 2 Hz, that is to capture each peak and trough:

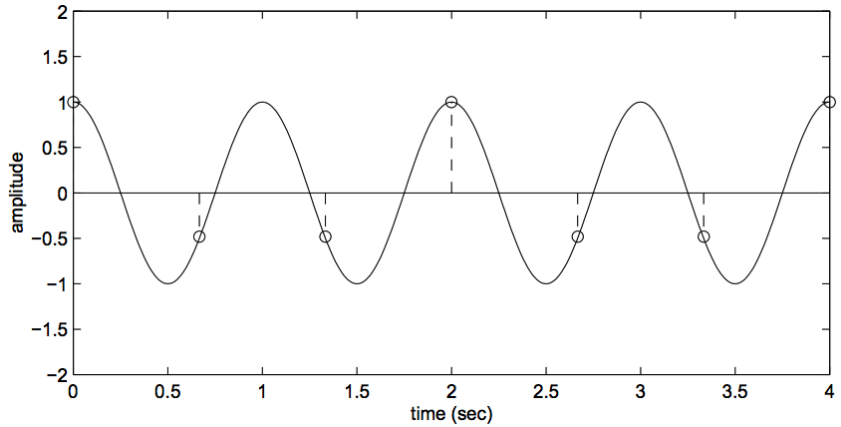


Nyquist sampling theorem

- If sampling it at a frequency higher than 2 Hz, e.g. 3 Hz , then there are more than enough samples to capture the variations:

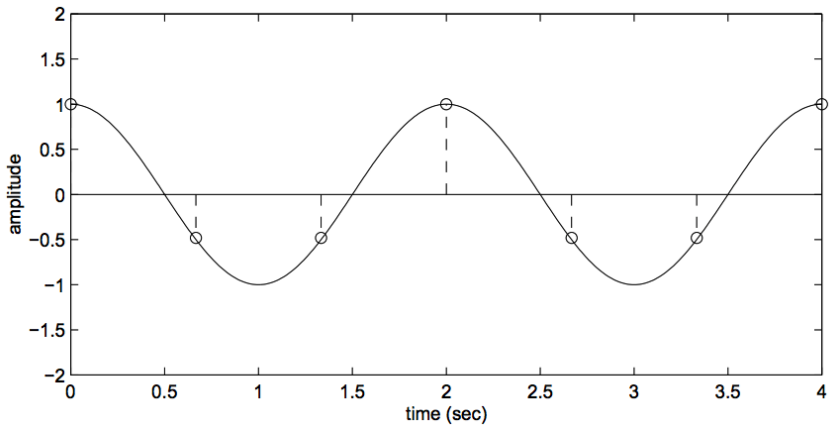


- If sampling it at a frequency lower than 2 Hz, e.g. 1.5 Hz , then there are now not enough samples to capture the variations:

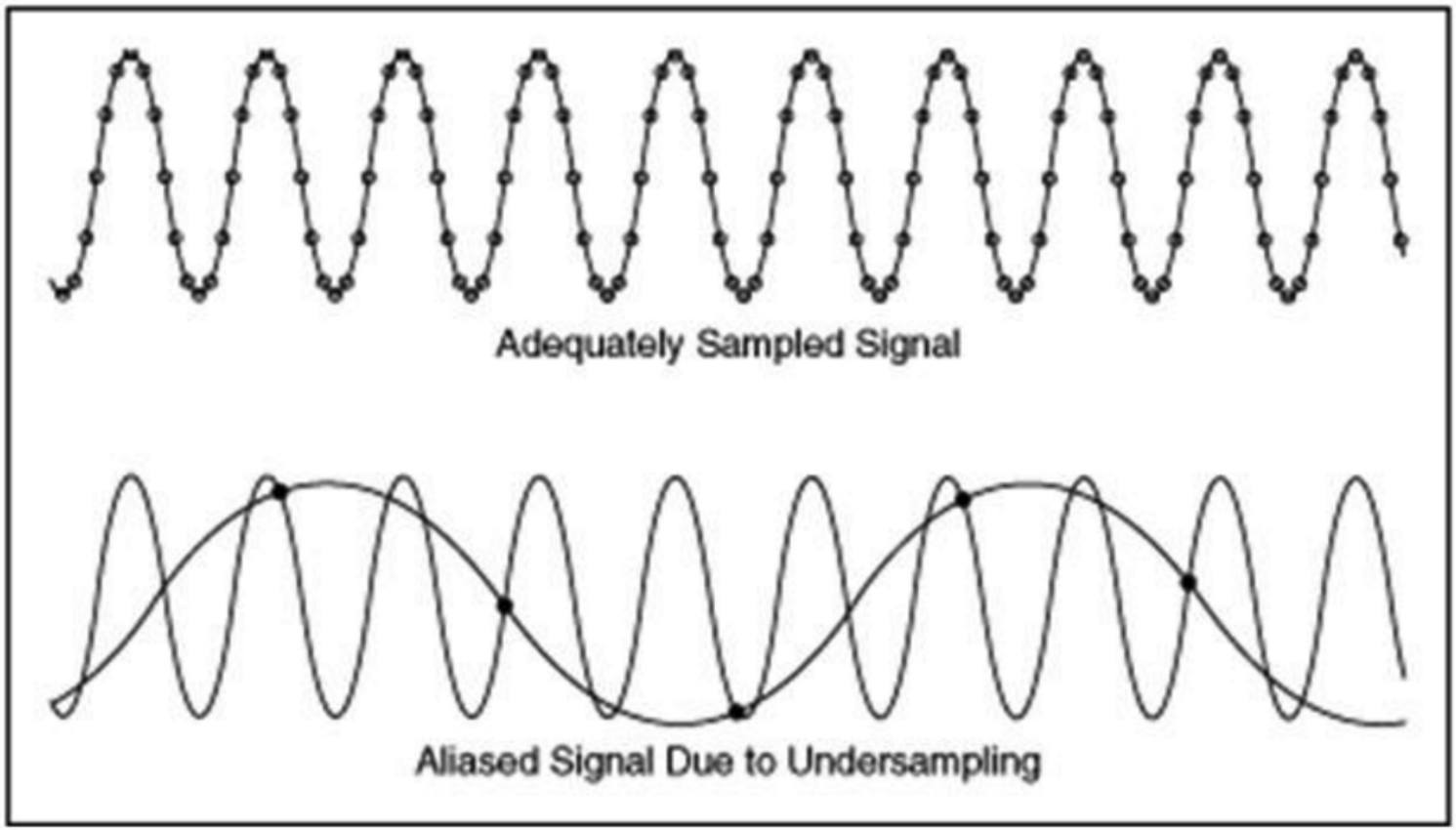


Aliasing

- Aliasing arises when a signal is discretely sampled at a rate that is insufficient to capture the changes in the signal.
- The misleading signal in a different form:

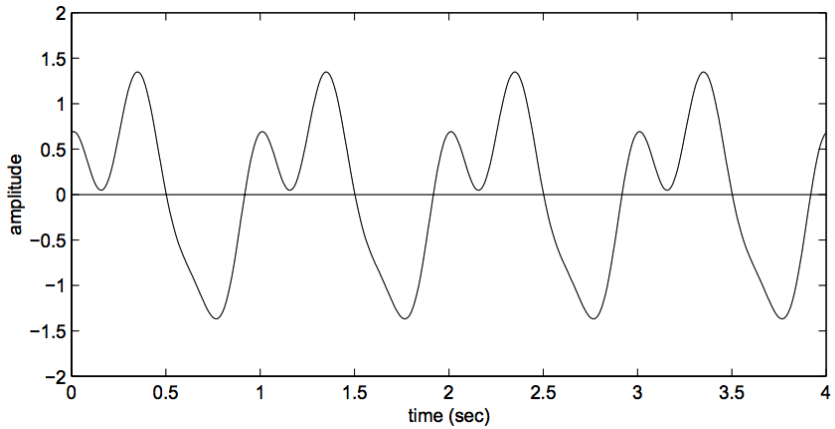


Aliasing

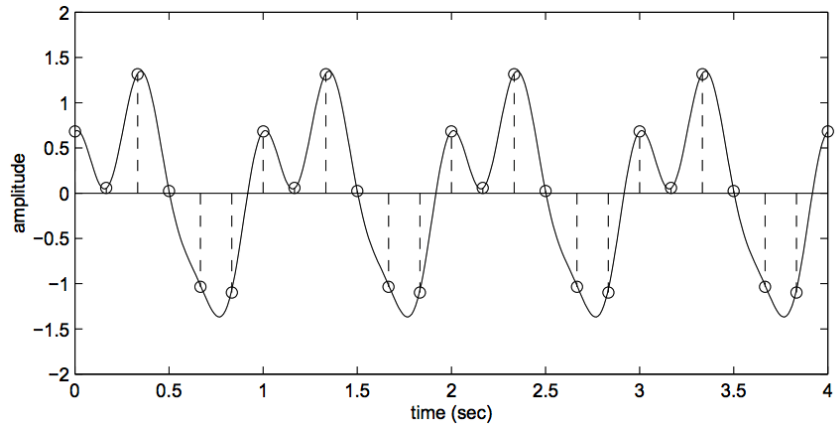

Adequately Sampled Signal
Aliased Signal Due to Undersampling

Aliasing

- Any continuous signal may be decomposed in terms of a sum of sines and cosines at different frequencies.
- Adding together sinewaves at frequencies of 1 Hz, 2 Hz, and 3 Hz:



- The signal must be sampled at least 6 Hz.



ADC on the mbed

- The mbed has up to six analogue inputs, on pins 15 to 20.
- Actually one successive approximation ADC plus one multiplexer.
- AnalogIn class member functions include read(), read_u16, and operator =

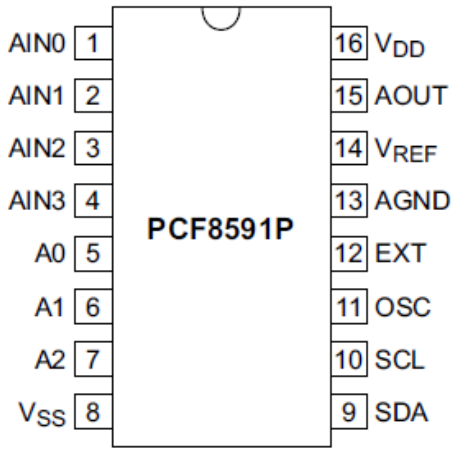
```
#include "mbed.h"

Serial pc(USBTX, USBRX);
AnalogIn Ain(p20);
float ADCdata;

int main() {
  pc.printf("ADC Data Values... \n\r");
  while (1) {
     ADCdata=Ain; //Ain.read()
     pc.printf("%f \n\r",ADCdata);
     wait (0.5);
  }
}
```

Section 4

PCF8591

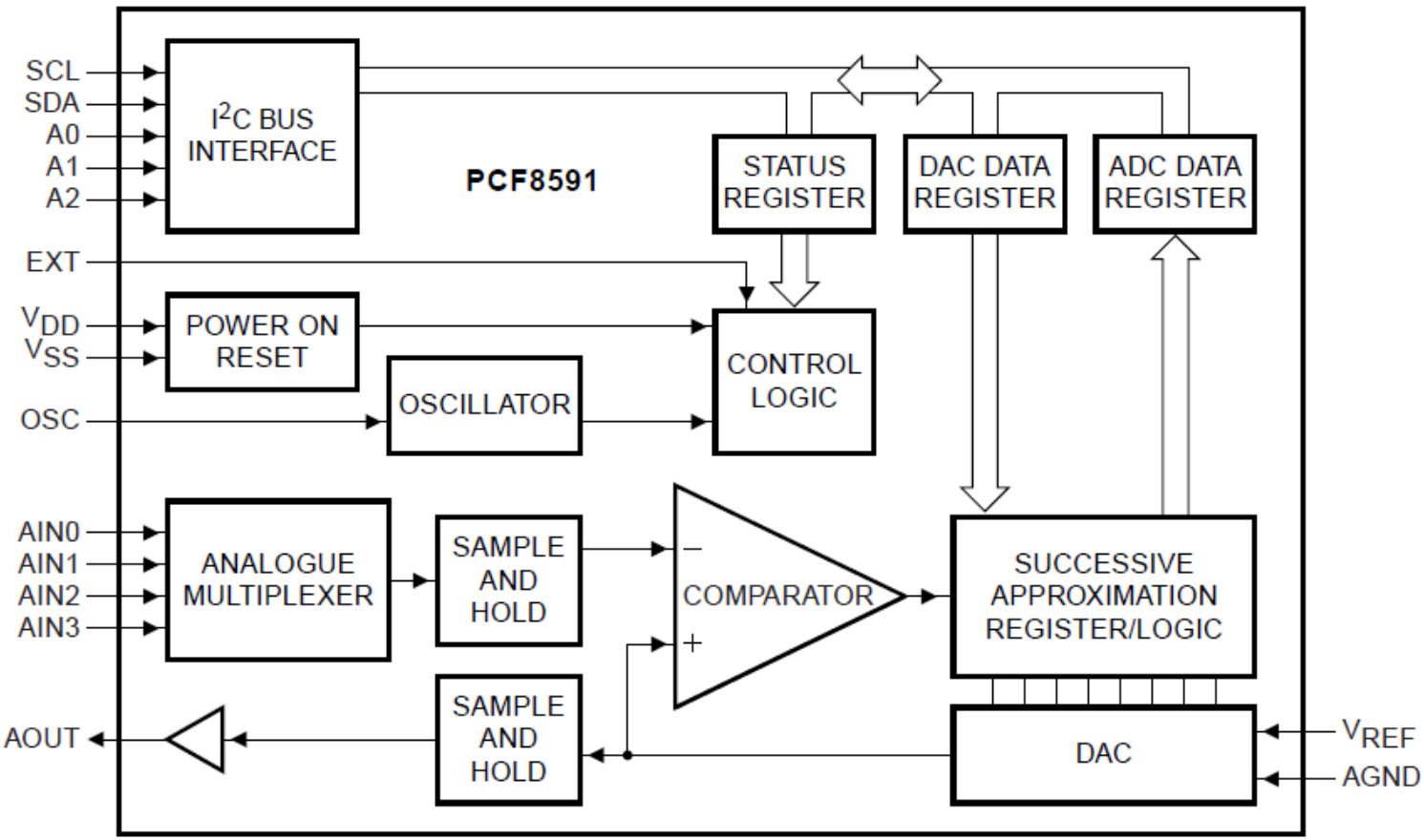
PCF8591

- The PCF8591 is a single-chip, single-supply low-power 8-bit device
- Four analogue inputs, one analogue output and a serial I2C-bus interface.
- Three address pins A0, A1 and A2 are used for programming the hardware address, allowing the use of up to eight devices connected to the I2C-bus.
- The maximum conversion rate is given by the maximum speed of the I2C-bus.



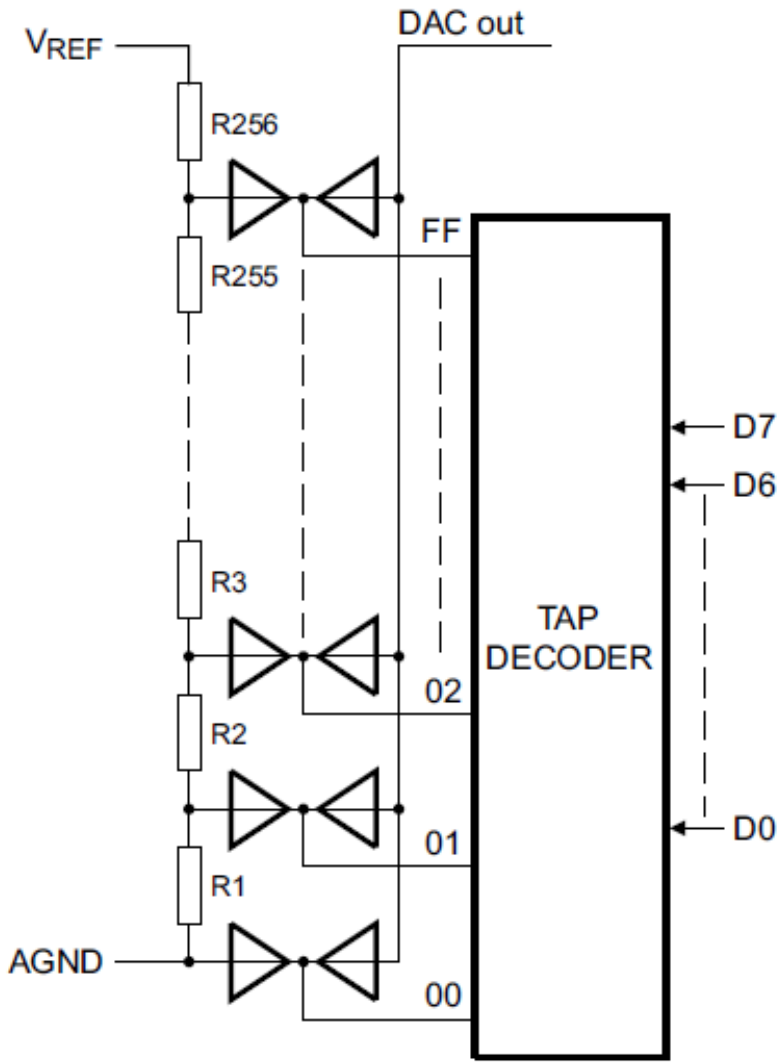
PCF8591 AD DA Convertor

- The A/D converter uses the successive approximation conversion technique. The on-chip D/A converter and a high-gain comparator are used.



PCF8591 AD DA Convertor

- The data sent to a PCF8591 device is stored in the DAC data register.
- This D/A converter consists of a resistor divider chain connected to the external reference voltage with 256 switches.
- The tap-decoder switches one of these taps to the DAC output line.



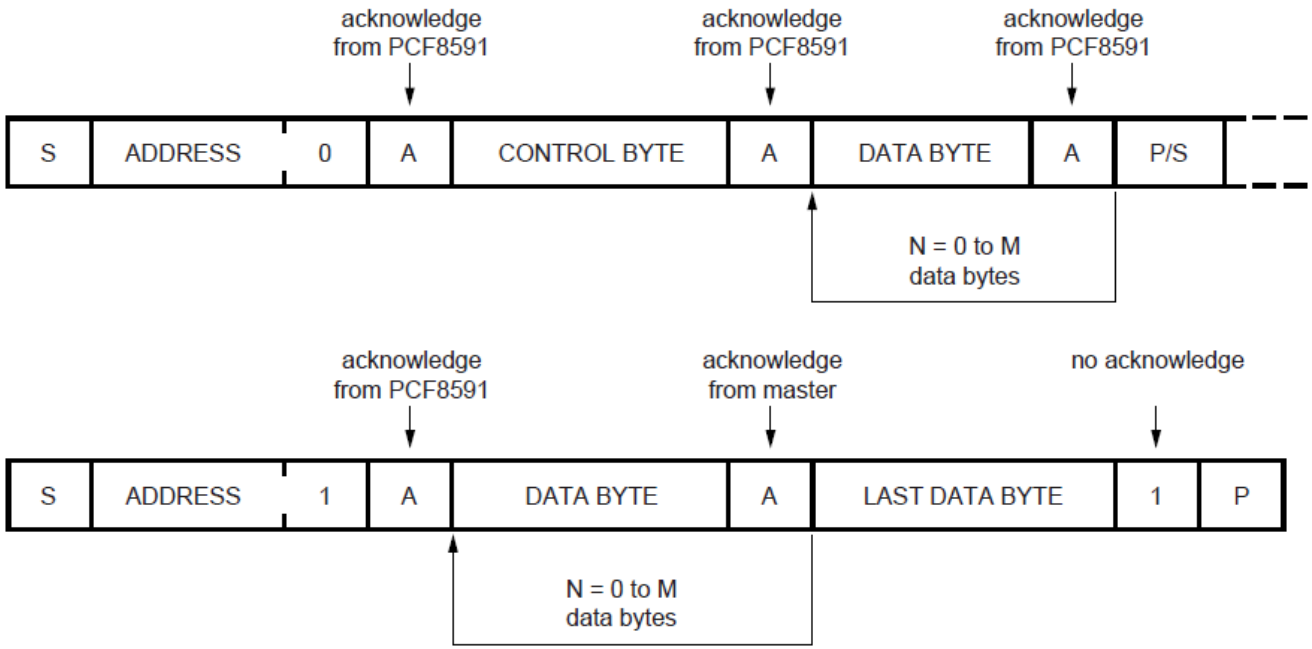
PCF8591 Addressing

- The address consists of a fixed part (1001) and a programmable part (A2 A1 A0 R/W).
- The address is always sent as the first byte after the start condition in the I2C-bus protocol.
- The last bit of the address byte is the read/write-bit.

Bit	Slave address							
	7 MSB	6	5	4	3	2	1	0 LSB
slave address	1	0	0	1	A2	A1	A0	R/$\overline{W}$

PCF8591 Control and Data Bytes

- The second byte sent to a PCF8591 device is stored in its control register (enabling the analogue output,selecting the analogue input channel,etc.)
- For DA, the third byte is the data.
- For AD, the third byte is the address again with read bit, followed by data bytes.



Connection Example