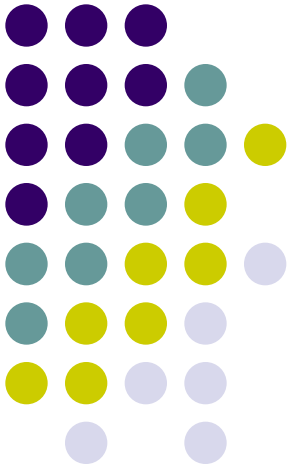
Chapter 5: Analog Input

EE2405

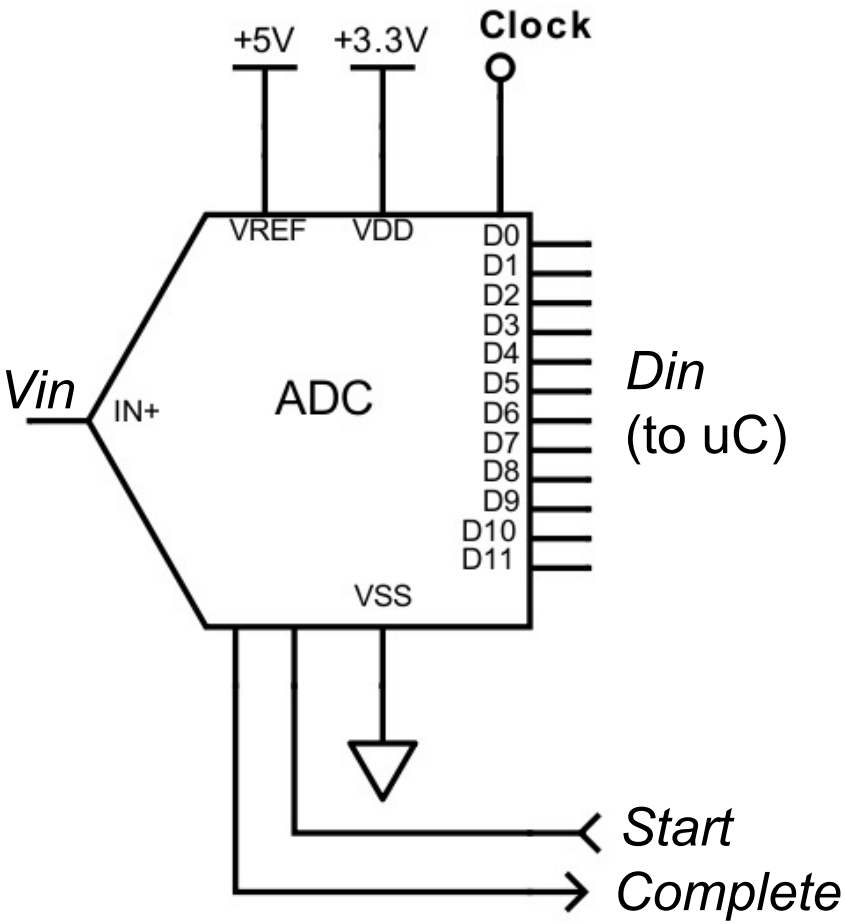
嵌入式系統與實驗

Embedded System Lab

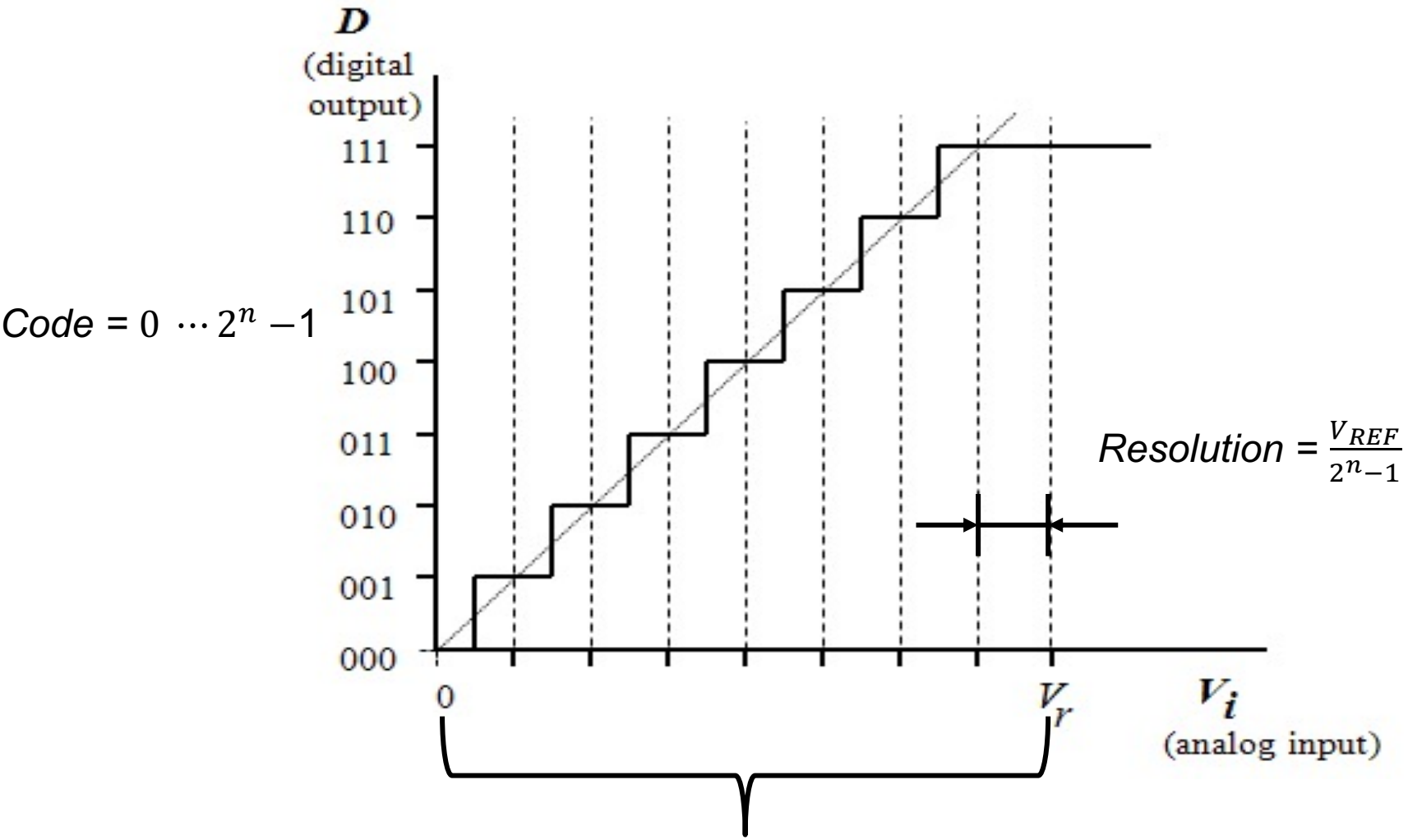


Analog to Digital Converter (ADC)

- An ADC can be used to **sample** and convert an analog input signal to binary numbers (**quantization**).
  - $D_{in} = round\left(\frac{V_{in}}{V_{REF}} 2^n\right)$
- Conversion is compared with V_{REF} to define range
- Conversion is triggered by *start* signal and return completion
  - Conversion takes time!
  - Usually sync by a clock

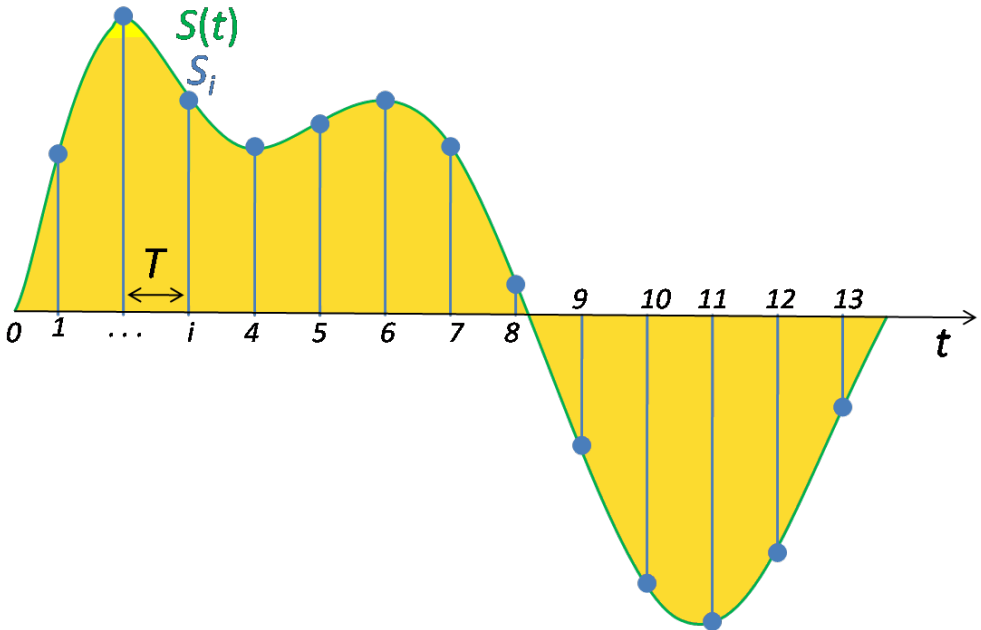


ADC Conversion Curve



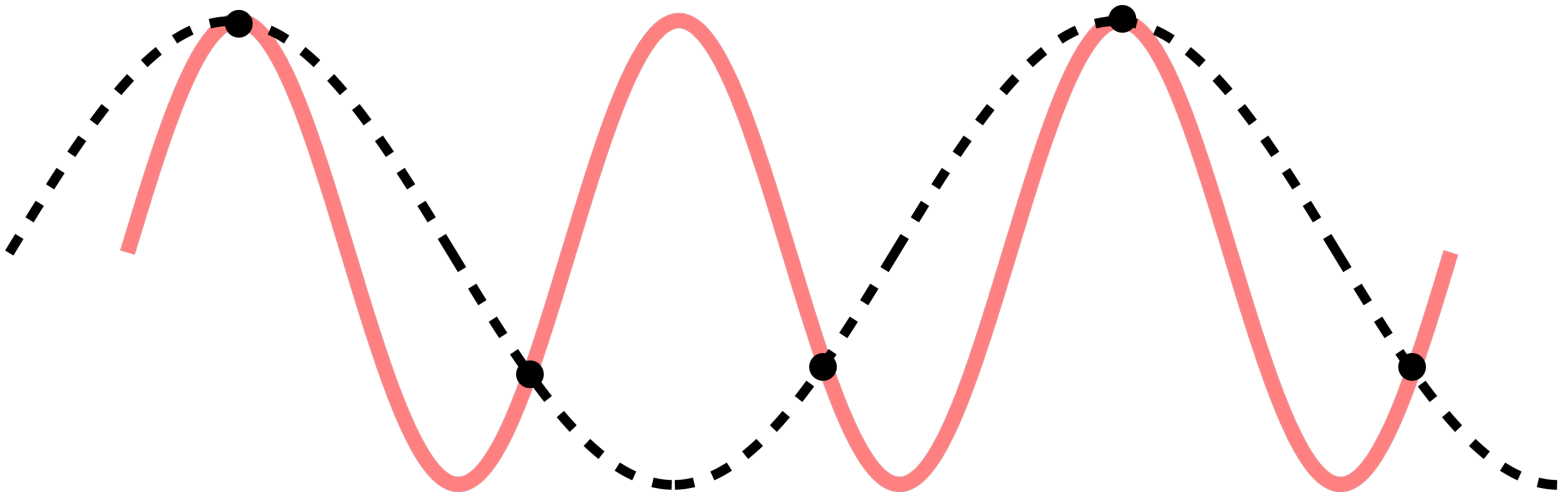
Sampling Frequency

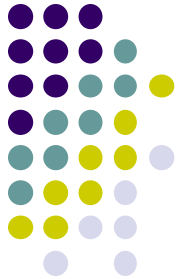
- We can use samples to represent the original signal.
- The more samples we take within a period (higher sampling frequency), the more accurate the sampled data will be.
- In order for a bandlimited signal (one with a frequency spectrum between 0 and f_{max}) to be reconstructed fully, it must be sampled at a rate of $f_s > 2 f_{max}$,
  - f_s is the Nyquist frequency.



Aliasing

- When sampled at lower then f_s
  - Any frequency component above $f_s/2$ is indistinguishable from a lower-frequency component
- For example,
  - The samples of the following two sine waves can be identical when at least one of them is at a frequency above half the sample rate





- The conversion time of an ADC is found to be 5.0 us. The ADC is set to convert repeatedly, with no other programming requirements. What is the maximum frequency signal it can digitize?

1/5.0us=200k Hz

f_s =200k > 2 f_{max}

f_{max} < 100k

AnalogIn

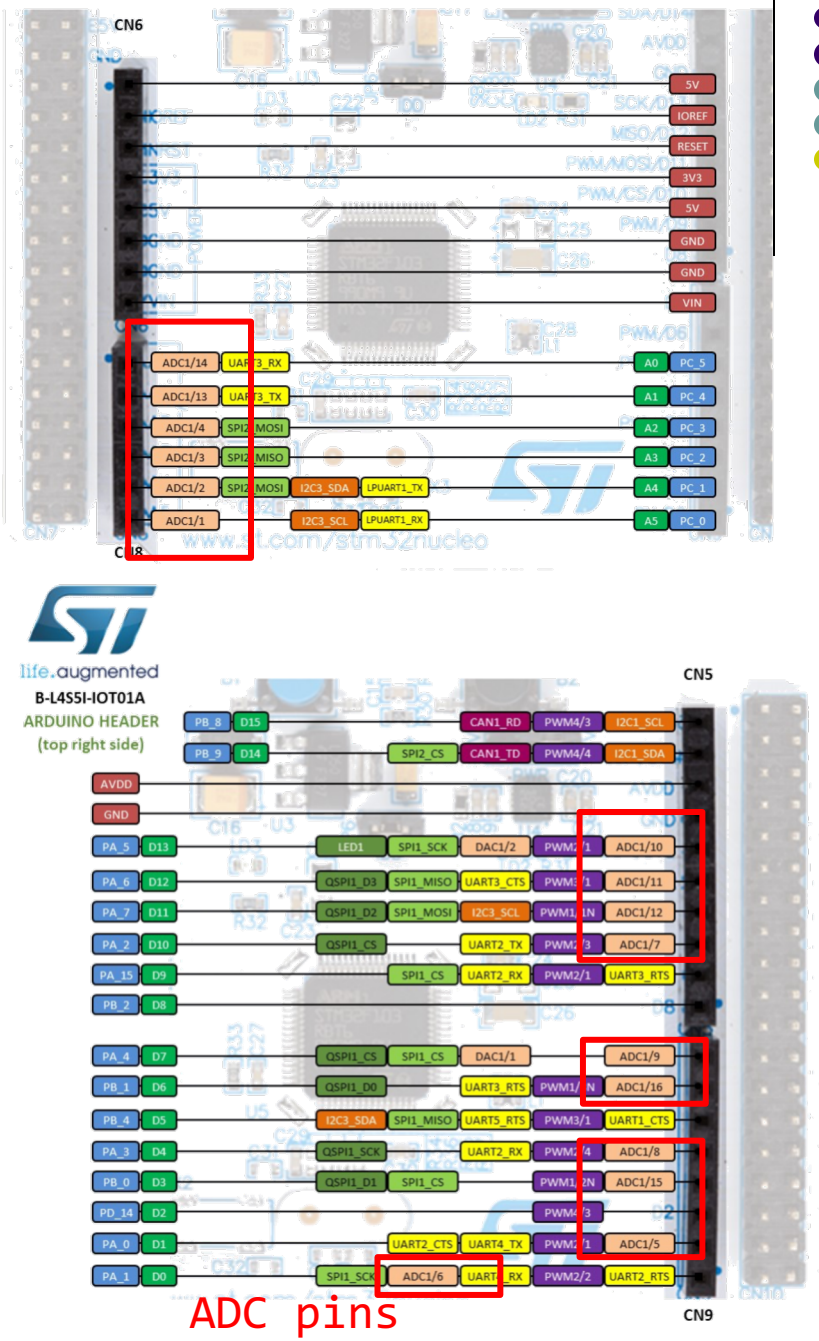
- AnalogIn API

Functions	Usage
AnalogIn	Create an AnalogIn object, connected to a specified pin
read	Read the input voltage, represented as a float in the range (0.0 - 1.0)
read_u16	Read the input voltage, represented as an unsigned short in the range (0x0 - 0xFFFF)

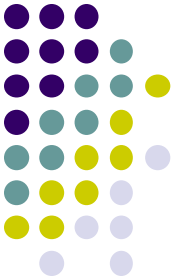
ADC Pins

All pink background labels are ADC pins.

A 12-bit 5Msps ADCs is multiplexed with 16 channels. Reference voltage is 3.3V.



Displaying ADC values on Computer Screen



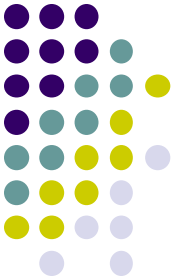
```
#include "mbed.h"

AnalogIn Ain(A0);
float ADCdata;

int main(){
    while(1){
        ADCdata = Ain;
        ThisThread::sleep_for(500ms);
        printf("%1.3f\r\n", ADCdata);
        //send data to the terminal
    }
}
```

printf( ) function sends data to the remote PC terminal connected by USB.

Averaging ADC Data to Filter Noise



- A very simple example of digital signal processing.
- Other filters can be applied

```
for (int i=0;i<=9;i++) {
    ADCdata=ADCdata+Ain*3.3;//sum 10 samples
}
ADCdata=ADCdata/10;//divide by 10
```

Chapter Review

- An ADC can be used to digitize analog input signals.
- It is important to understand ADC characteristics, in terms of input range, resolution, and conversion time.
- Nyquist's sampling theorem: sampling frequency must be at least twice that of the highest frequency component in the sampled analog signal.
- Aliasing occurs when the Nyquist criterion is not met, this can introduce false frequencies to the data. Aliasing can be avoided by introducing an anti-aliasing filter to the analog signal before it is sampled.
- There are numerous sensors available which have an analog output.