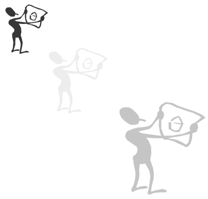
Chapter 6

Analog / Digital Converters

Process Control

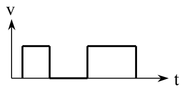
Outline

- ➡ **Digital vs. Analog**
- **Analog/Digital Conversion**
  - Resolotion & Error
  - Unipolar & Bipolar Code
  - Circuit Diagrams
- **Programming**
  - DAC Example.
  - ADC Example

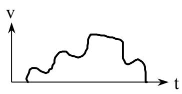


Digital vs. Analog

- Digital characteristics
  - Discrete signal levels (voltage usually)
  - Two levels: on/off, high/low 1/0 (binary)
  - Disjoint or quantized level changes



- Analog characteristics
  - Continuous signal levels
  - Very small, smooth level changes



Digital vs. Analog

- Natural world is analog
- Many devices better when digital:

<u>Digital</u>	<u>Analog</u>
Compact Disc	Magnetic Tape
Microcomputer-controlled Engine	Mechanically-controlled Engine
Telephone System	Telephone System
Movie Special Effects	Movie Special Effects
Digital Computers:	Analog Computer:
PC, Mainframe, Supercomputer	OpAmp, Res, Cap

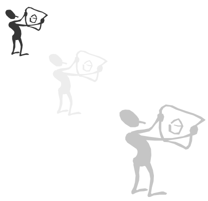
Digital vs. Analog

- Advantages of each technology:

<u>Digital</u>	<u>Analog</u>
Reproducible results	Less complex ?
Ease of design	
Flexible and function	Higher speed ?
Programmable	
High speed	
Economical	

Outline

- **Digital vs. Analog**
- ➡ **Analog/Digital Conversion**
  - Resolution & Error
  - Unipolar & Bipolar Code
  - Circuit Diagrams
- **Programming**
  - DAC Example.
  - ADC Example



Analog/Digital Conversion

A/D conversion is the process of sampling a continuous signal

Two significant implications

- **The information content of the sampled signal is less than the continuous signal**

  The continuous signal contains an infinite number of independent samples, the sampling process reduces that to a finite number of independent samples

- **Uncertainty is added to the sampled data.**

  Quantization error is part of the sampling process since the number of intervals is finite. This is analogous to truncating a number after a specific number of places

Resolution and Error

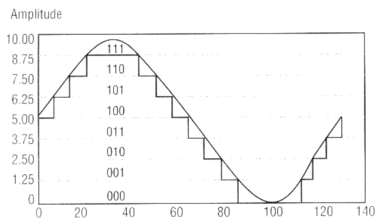
Quantization error

Quantization error is defined as +/- ½ LSB (Least Significant Bit) = +/- ½ the resolution (see definition below)
Variance of the quantization error = resolution^2/12 (variance of a uniform distribution)

Resolution

Resolution = 1 LSB = $V_{full\ scale}/2^n$

Simple Example



Number of bits = 3
Number of intervals = 2^3
Range = 0- 10 volts
Resolution= 1.25 volts

Quantization error= +/- .625 volts
Variance =$(1.25)^2/12$=.130 volts^2

Unipolar & Bipolar Binary Code

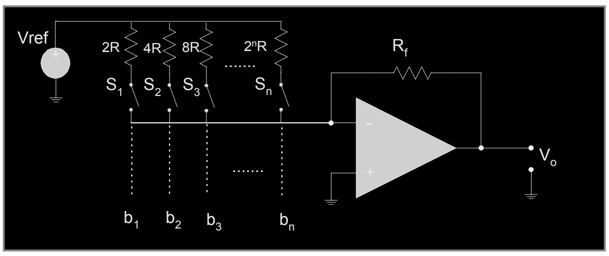
- Unipolar Straight Binary (**USB**) for unipolar analog signals. For example: 0 to 5V, 0 to 10V.
- Bipolar Offset Binary (**BOB**) used for bipolar analog signals. For example: +/-5V, +/-10V.
- Bipolar Two's Complement (**BTC**) also used for bipolar analog signals like the BOB.

Unipolar & Bipolar Binary Code

	USB			BOB			BTC	
+V		256			256			
V - bit	11111111	255	V- bit	11111111	255	V- bit	01111111	127
	10000001	129		10000001	129		00000001	1
V/2	10000000	128	0	10000000	128	0	00000000	0
	01111111	127		01111111	127		11111111	-1
0	00000000	0	-V	00000000	0	-V	10000000	-128

DAC Diagram

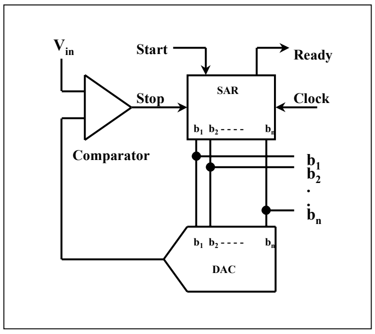
The converting is by writing a binary word to the digital switches



ACD Diagram

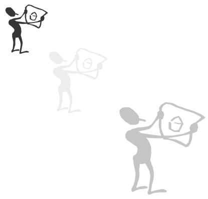
Configuration of a SAR A/D converter

The converting is by:
1) Writing **start**.
2) Waiting for **ready.**
3) Reading binary word.



Outline

- **Digital vs. Analog**
- **Analog/Digital Conversion**
  - Resolotion & Error
  - Unipolar & Bipolar Code
  - Circuit Diagrams
- ➡ **Programming**
  - DAC Example.
  - ADC Example



Programming Example

```
/**************************************************/
/*                   EFDAC.C                      */
/*------------------------------------------------*/
/*   Task      : Drivers for PC_CARD I/O Interface */
**************************************************
        Digital Parameters:
            Chan  = (0, 1)
            Data  = (0..255)

        Analog Parameters:
            Chan  = (0, 1, 2, 3)
            Gain  = (1, 2, 3, 4)
            Rang  = (1.25, 2.50)
            Pol   = (UNIPOLAR, BIPOLAR)

        Counter Parameters:
            Chan  = (0, 1, 2)
            Mode  = (0, 1, 2, 3, 4, 5)
            Count = (BINARY, BCD)
            Format= (LATCH, MSB, LSB, LMSB)
  *************************************************/
```


Card Addresses

Base + 8 = DAC0d (Read / Write data to dac channel)
Base + 9 = DAC0s (Write=ST start conversion, Read=BUSY end of conversion)
Base + A = DAC1d (Read / Write data to dac channel)
Base + B = DAC1s (Write=ST start conversion, Read=BUSY end of conversion)
Base + C = DAC2d (Read / Write data to dac channel)
Base + D = DAC2s (Write=ST start conversion, Read=BUSY end of conversion)
Base + E = DAC3d (Read / Write data to dac channel)
Base + F = DAC3s (Write=ST start conversion, Read=BUSY end of conversion)

Write to DACxd is digital to analog channel x
Read from DACxd is analog to digital channel x
Write to DACxs (any data) is ST- start conversion A/D
Read from DACxs (bit 1) is BUSY- end of conversion A/D

DAC Example

```
/*********************************************************/
void AnalogOut(byte Chan, double Rang, byte Pol, double Volt)
 {
  double Temp = (Volt / Rang * 256.0);

  if (Pol == 0)
     outp(Base + 8 + Chan * 2, (byte)Temp);     // unsigned char
  else
     outp(Base + 8 + Chan * 2, (char)(Temp / 2)); // signed char
 }
/*********************************************************/
```


ADC Example

```
double AnalogIn(byte Chan, double Rang, byte Pol)
 {
  double Temp;
  byte Data1;
  char Data2;
  int Address = Base + 8 + Chan * 2;
  outp(Address + 1, BIT0);                  // Start convertion
  while (!(inp(Address+1) & BIT1));      // Wait until BIT1 is Set
  if (Pol == 0) {
    Data1 = inp(Address);
    Temp = Data1 / 256.0 * Rang;
    }
  else {
    Data2 = inp(Address);
    Temp = Data2 / 256.0 * Rang * 2;
    }
  return(Temp);
  }
```