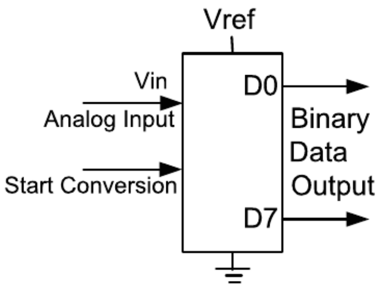
ADC INTERFACING

Analog to Digital Converter (ADC) is used to convert analog data (such as temperature, sound etc.) to digital form so that microcontroller can use the digital data. Usually the analog data is first converted into equivalent electrical form (voltage/current - again analog) using a transducer (also called sensor). This voltage or current is given to ADC as input and the output is the equivalent binary data.

Figure shows the block diagram of an 8-bit ADC:



Resolution of an ADC: It is the number of bits used to represent digital data

Step size of an ADC: An n-bit resolution ADC have 2^n steps. For an 8 bit ADC, total number of steps is 256. Step size is the voltage measured for each step based on the Reference voltage. For example, an 8-bit ADC with Vref=5V has a step size 19.53mv (5V/256 = 0.01953V = 19.53mV).

- It means that:

Input in mV	Output in binary
0	00000000
19.53	00000001
39.06	00000010
58.59	00000011

And so on. From the table, any input less than 19.53mV gives 00000000 in binary and any input voltage less than 39.06mV gives 00000001 in binary. So that a 5V input gives 11111111 in binary.

Step size is the smallest change in voltage that can be identified by the ADC. Higher-resolution ADCs provide a smaller step size.

- The resolution of an ADC chip is decided at the time of its design and cannot be changed, but we can control the step size using V_{ref}.

ATMega32 ADC Features
1. It is a 10-bit ADC.
2. It has 8 analog input channels
3. The converted output binary data is held by two special function registers called ADCL (A/D Result Low) and ADCH (A/D Result High).

4. Because the ADCH:ADCL registers give us 16 bits and the ADC data out is only 10 bits wide, 6 bits of the 16 are unused. We have the option of making either the upper 6 bits or the lower 6 bits unused.
5. We have three options for Vref. Vref can be connected to AVCC (Analog Vcc), internal 2.56V reference, or external AREF pin.

Programming:

In ADC programming involves the following five registers:

1. ADCH (to hold higher bits of digital data after conversion)
2. ADCL (to hold lower bits of digital data after conversion)
3. ADMUX (ADC multiplexer selection register)
4. ADCSRA (ADC Control and Status Register)
5. SPIOR (Special Function I/O Register)

ADMUX Register:

REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0

REFS1:0 (Bit 7:6)   : Select reference voltage. For example, to select internal 2.56V as reference, bit 1 is used at both position.

ADLAR (Bit 5) : ADC Left Adjust Results.

- The 10 bit digital output is in ADCH:ADCL, which is 16 bit and 6 bits are unused
- The ADLAR bit is used to justify the result bits. (ADLAR =1 left justified, ADLAR = 0 right justified) as shown in figure.

	ADCH								ADCL							
Left-Justified ADLAR = 1	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	UNUSED					
ADLAR = 0 Right-Justified	UNUSED						D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

MUX4:0 (Bit 4:0) :   Analog Channel selection bits

- For example, to select ADC0 input, use 00000. For ADC1, use 00001 and so on.

Example: If we want to connect analog input at ADC0 pin, with internal 2.56V reference voltage and left justified result, the ADMUX bits will be: 11100000 = 0xE0

ADCSRA (ADC Control and Status Register):

ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0

- Bits of this register is used to control and monitor the operation of ADC
  - ADEN (Bit 7)   : ADC Enable
  - ADSC (Bit 6)   : ADC Start Conversion

- ADATE (Bit 5) : ADC Auto Trigger Enable
- ADIF (Bit 4) : ADC Interrupt Flag (set when conversion is completed)
- ADIE (Bit 3) : ADC Interrupt Enable
- ADPS2:0 (Bit 2:0) : ADC Prescaler Select Bits

ADPS2	ADPS1	ADPS0	ADC Clock
0	0	0	Reserved
0	0	1	CK/2
0	1	0	CK/4
0	1	1	CK/8
1	0	0	CK/16
1	0	1	CK/32
1	1	0	CK/64
1	1	1	CK/128

For example,

- ADC can be enabled with 1:128 prescaler by assigning the byte 0x87 to ADCSRA register.
- To start the conversion, we set the ADSC bit by using the statement:
  ADCSRA |= (1<<ADSC)
- We can monitor the completion of conversion by looking in to the ADIF bit by using:
  while ((ADCSRA & (1<<ADIF))==0);

  or by enabling the interrupt and executing its ISR
- We can take the digital data from ADCH:ADCL register.

<u>Steps in ADC Programming:</u>

1. Make the pin as input for the selected ADC channel.
2. Enable the ADC
3. Select the conversion speed using prescaler
4. Select the voltage reference and ADC input channels
5. Activate the start conversion bit
6. Wait for the conversion to be completed by polling the ADIF
7. Read the ADCL and ADCH registers to get the digital data output