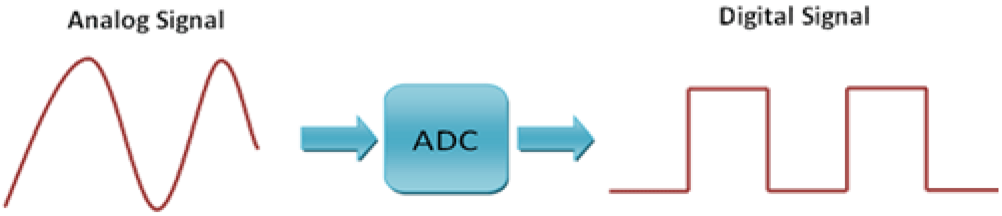
Microcontroller understands only digital language. However, the inputs available from the environment to the microcontroller are mostly analog in nature, i.e., they vary continuously with time. In order to understand the inputs by the digital processor, a device called Analog to Digital Converter (ADC) is used. As the name suggests this peripheral gathers the analog information supplied from the environment and converts it to the controller understandable digital format, microcontroller then processes the information and provides the desired result at the output end.



ATmega16 has an inbuilt 10 bit, 8-channel ADC system. Some of the basic features of Armega16 ADC are:

- 8 Channels.
- 10-bit Resolution.
- Input voltage range of 0 to Vcc.
- Selectable 2.56V of internal Reference voltage source.
- AREF pin for External Reference voltage.
- ADC Conversion Complete Interrupt.

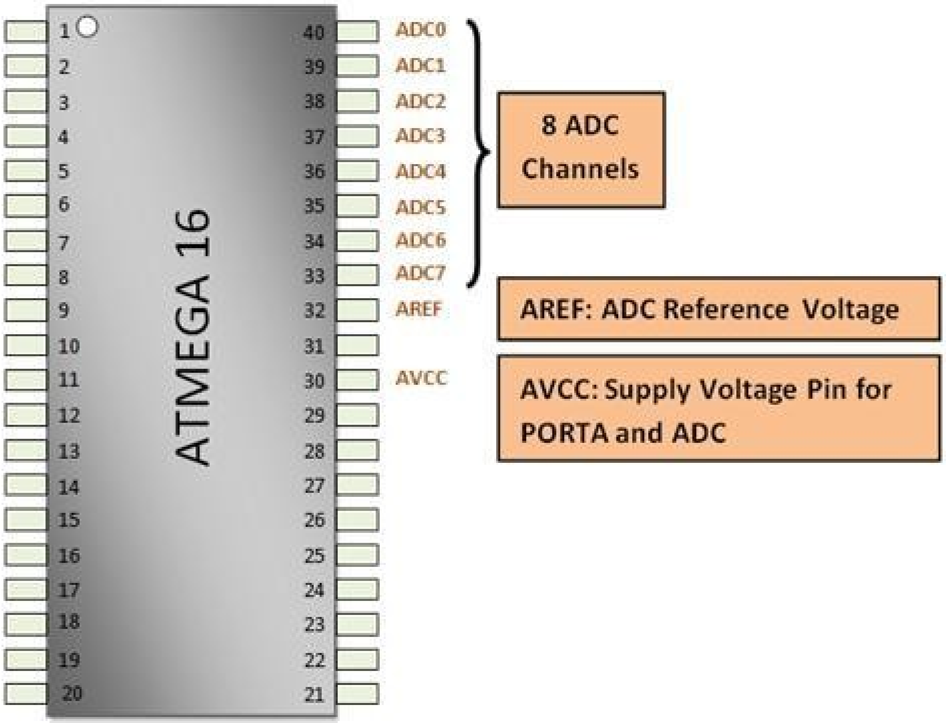
ADC channels in Atmega16 are multiplexed with PORTA and use the common pins (pin33 to pin40) with PORTA. ADC system of Atmega16 microcontroller consists of following pins:

i. **ADC0-ADC7:** 8 Channels from Pin 40 to Pin 33 of Atmega16 ADC peripheral.

ii. **AREF:** Pin32 of Atmega16 microcontroller, the voltage on AREF pin acts as the reference voltage for ADC conversion, reference voltage is always less than or equal to the supply voltage, i.e., Vcc.

iii. **AVCC:** Pin30, this pin is the supply voltage pin for using PORTA and the ADC; AVCC pin must be connected to Vcc (microcontroller supply voltage) to use PORTA and ADC.

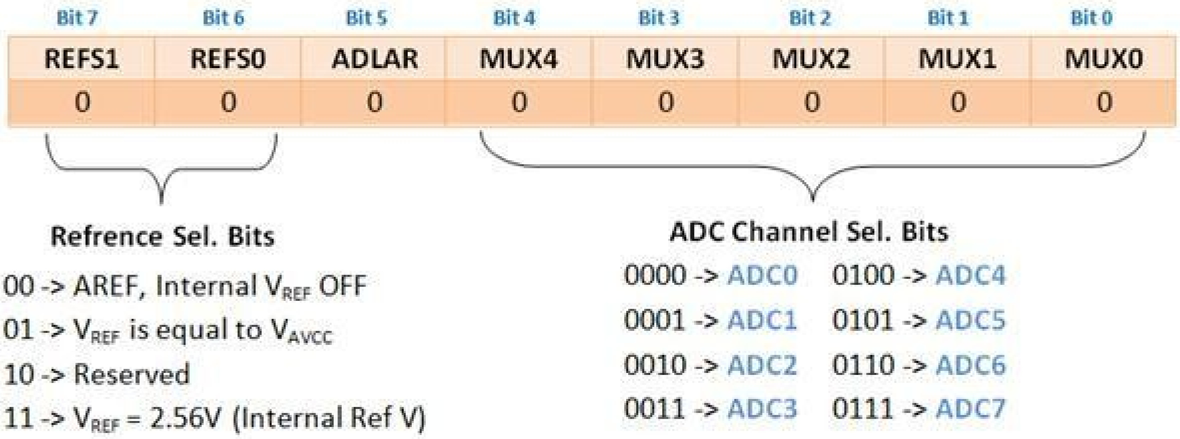
Note: External reference voltage source can be used at AREF pin. However, Atmega16 also has internal reference voltage options of 2.56V and Vref = Vcc.



ADC Registers

To use the ADC peripheral of Atmega16, certain registers need to be configured.

i. ADMUX (ADC Multiplexer And Selection Register)



Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0
0	0	0	0	0	0	0	0

Refrence Sel. Bits

00 -> AREF, Internal V_{REF} OFF

01 -> V_{REF} is equal to V_{AVCC}

10 -> Reserved

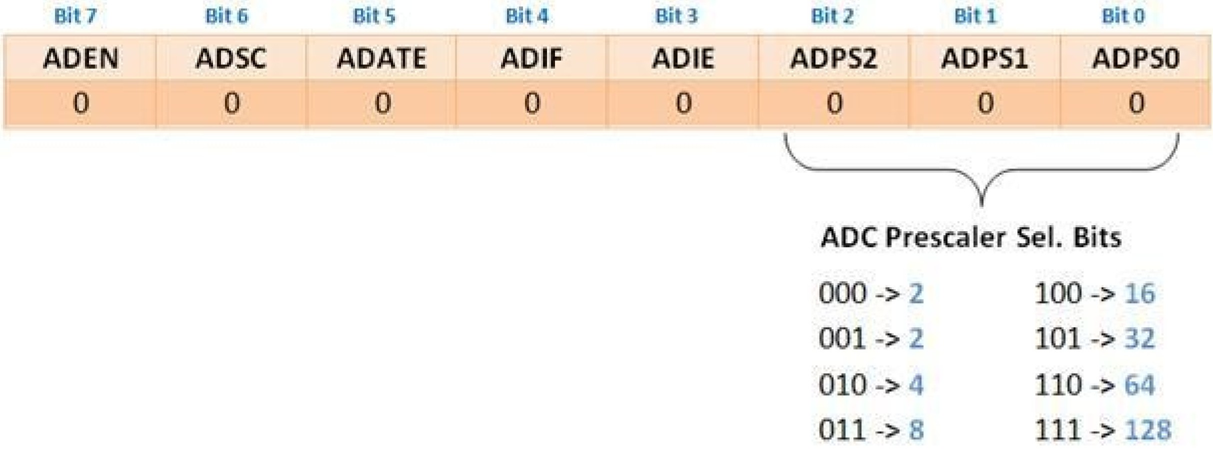
11 -> V_{REF} = 2.56V (Internal Ref V)

ADC Channel Sel. Bits

0000 -> ADC0	0100 -> ADC4
0001 -> ADC1	0101 -> ADC5
0010 -> ADC2	0110 -> ADC6
0011 -> ADC3	0111 -> ADC7

REFS[0:1] bits determine the source of reference voltage whether it is internal or the external voltage source connected to AREF pin. MUX[4:0] bits are used to select between the channels which will provide data to ADC for conversion. ADLAR bit when set to 1 gives the left adjusted result in data registers ADCH and ADCL.

ii. ADCSRA (ADC Control and Status Regis

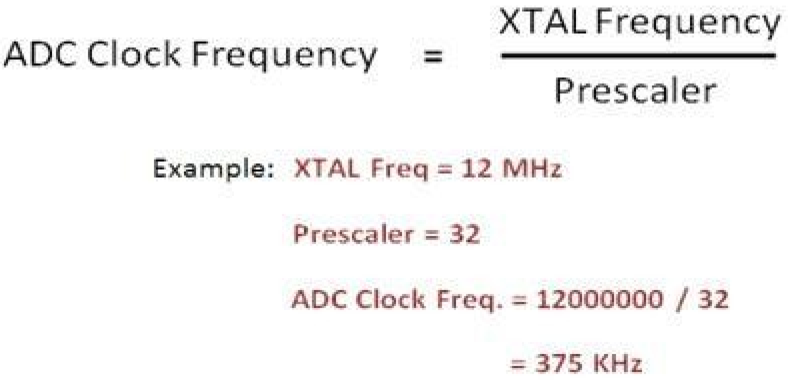


Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0
0	0	0	0	0	0	0	0

ADC Prescaler Sel. Bits

000 -> 2	100 -> 16
001 -> 2	101 -> 32
010 -> 4	110 -> 64
011 -> 8	111 -> 128

ADEN: ADC Enable bit, this bit must be set to 1 for turning ADC on.
ADSC: ADC Start Conversion bit, this bit is set to 1 to start ADC conversion, as soon as conversion is completed this bit is set back to 0 by the hardware.
ADATE: ADC Auto Trigger Enable, this bit is set to 1 to enable auto triggering of ADC conversion.
ADIF: ADC Interrupt Flag, this bit is set to 1 when ADC conversion gets complete.
ADIE: ADC Interrupt Enable, this bit is set to 1 if we want to activate the ADC conversion complete interrupt.
ADPS[0:2]: ADC Prescaler bits, these bits are used to set the ADC clock frequency, the configuration of these bits determine the division factor by which the microcontroller clock frequency is divided to get the ADC clock frequency. The figure above shows the prescaler bit values for respective division factor.



$$\text{ADC Clock Frequency} = \frac{\text{XTAL Frequency}}{\text{Prescaler}}$$

Example: XTAL Freq = 12 MHz

Prescaler = 32

ADC Clock Freq. = 12000000 / 32

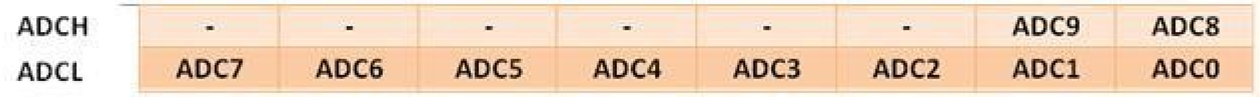
= 375 KHz

The ADC clock frequency must lie somewhere between 50 KHz to 200 KHz.

iii. ADCH & ADCL (ADC Data Registers)

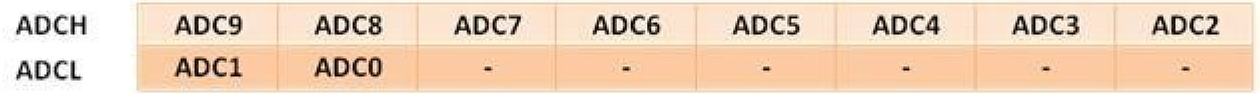
When the ADC conversion is complete the data is stored in these two registers. The data configuration depends on the ADLAR bit value of ADMUX register. If ADLAR=0, data is right adjusted and if ADLAR=1, data is left adjusted. Always read ADCL first and then ADCH. In cases where the 8-bit precision is enough set the ADLAR bit to 1 to left adjust the data and read only the ADCH data register.

When ADLAR = 0,



ADCH	-	-	-	-	-	-	ADC9	ADC8
ADCL	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0

When ADLAR = 1,



ADCH	ADC9	ADC8	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2
ADCL	ADC1	ADC0	-	-	-	-	-	-

Circuit description

Connect the circuit as shown in the circuit diagram. A ceramic capacitor 104 is connected in between AVcc (pin 30) and Aref (pin 32). AVcc (pin 30) is connected to external supply +5V.

Code explanation

To interface analog device with AVR microcontroller, follow the following steps for programming it.

Step1: To initialize ADC

i. Set the value in ADMUX register according to the ADC channel and the reference voltage.
ii. Set the Prescaler bits accordingly in ADCSRA register.
iii. Set the ADEN bit to enable the ADC.

Step2: To read the analog value

i. Put the channel value in ADMUX
ii. Start the conversion by setting the ADSC bit.
iii. Monitor the ADIF bit for conversion complete. Remember all interrupt flags either in ADC or UART or TIMER needs to be reset (1) after interrupt occurs.
iv. Clear the conversion bit ADIF. By writing it 1.
v. Digital converted result is now available in ADCH and ADCL registers.