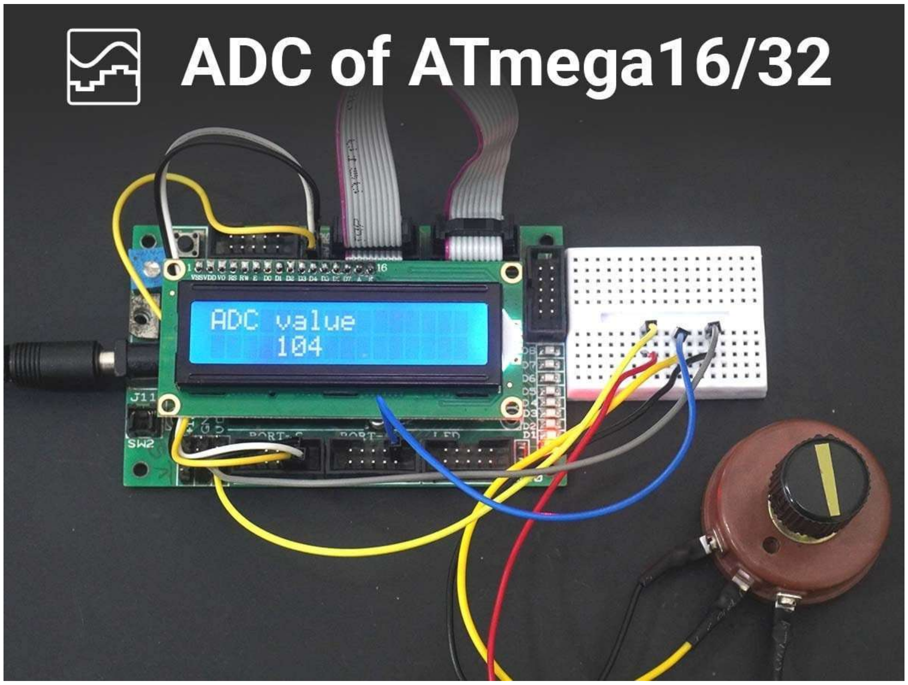
ADC in AVR ATmega16/ATmega32

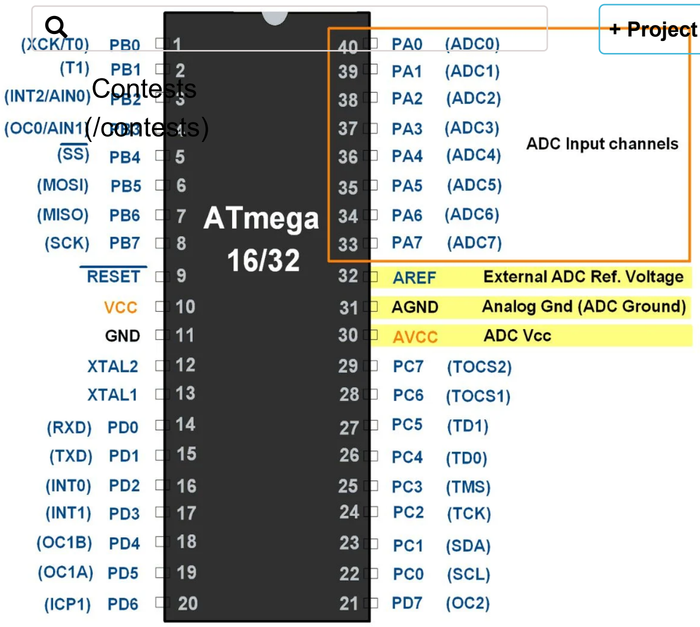


Introduction to ATmega ADC

ADC (Analog to Digital converter) is the most widely used device in embedded systems which is designed especially for data acquisition. In the AVR ATmega series normally 10-bit ADC is inbuilt in the controller.

Let us see how to use the ADC of AVR ATmega16 / ATmega32.

ATmega16/32 supports eight ADC channels, which means we can connect eight analog inputs at a time. ADC channel 0 to channel 7 are present on PORTA. i.e. Pin no.33 to 40.



ADC Pins of ATmega16/32

The controller has 10 bit ADC, which means we will get digital output 0 to 1023.

i.e. When the input is 0V, the digital output will be 0V & when input is 5V (and Vref=5V), we will get the highest digital output corresponding to 1023 steps, which is 5V.

So controller ADC has 1023 steps and

- Step size with Vref=5V : 5/1023 = 4.88 mV.
- Step size with Vref=2.56 : 2.56/1023 = 2.5 mV.

So Digital data output will be Dout = Vin / step size.

ATmega16/32 ADC Features

- It is 10-bit ADC
- Converted output binary data is held in two special functions 8-bit register ADCL (result Low) and ADCH (result in High).
- ADC gives 10-bit output, so (ADCH: ADCL) only 10-bits are useful out of 16-bits.
- We have options to use this 10-bits as upper bits or lower bits.
- We also have three options for Vref. 1. AVcc (analog Vcc), 2. Internal 2.56 v3. External Aref. Pin.
- The total conversion time depends on crystal frequency and ADPS0: 2 (frequency devisor)
- If you decided to use AVcc or Vref pin as ADC voltage reference, you can make it more stable and increase the precision of ADC **by connecting a capacitor between that pin and GND**.

ATmega16/32 ADC Registers

In AVR ADC, we need to understand four main register -

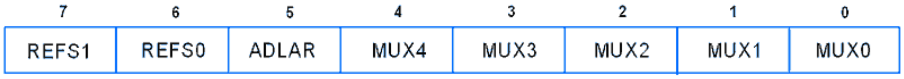
1. **ADCH:** Holds digital converted data higher byte
2. **ADCL:** Holds digital converted data lower byte
3. **ADMUX:** ADC Multiplexer selection register

4. **ADCSRA:** ADC Control and status register

ADCH: ADCL register

First, two-register holds the digital converted data, which is 10-bit.

ADMUX Register



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

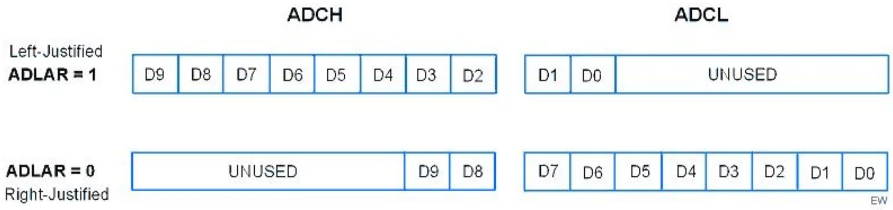
Bit 7: 6 – REFS1 : 0: Reference Selection Bits

Reference voltage selection for ADC

REFS1	REFS0	Vref to ADC
0	0	AREF pin
0	1	AVCC pin i.e. Vcc 5 V
1	0	Reserved
1	1	Internal 2

Bit 5 – ADLAR: ADC Left Adjust Result

Use 10-bits output as upper bits or lower bits in ADCH & ADCL.



	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

Bits 4 : 0 – MUX4 : 0: Analog Channel and Gain Selection Bits

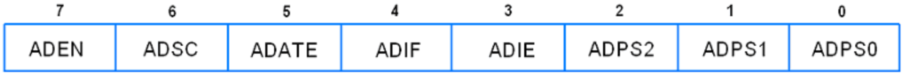
We can select input channel ADC0 to ADC7 by using these bits. These bits are also used to select comparator (inbuilt in AVR) inputs with various gain. We will cover these comparator operations in another part.

Selecting a channel is very easy, just put the channel number in MUX4 : 0.

Suppose you are connecting the input to ADC channel 2 then put 00010 in MUX4 : 0.

Suppose you are connecting the input to ADC channel 5 then put 00101 in MUX4 : 0.

ADCSRA Register:



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

- **Bit 7 – ADEN: ADC Enable**

Writing one to this bit enables the ADC. By writing it to zero, the ADC is turned off. Turning the ADC off while a conversion is in progress, will terrninate this conversion.

- **Bit 6 – ADSC: ADC Start Conversion**

Writing one to this bit starts the conversion.

- **Bit 5 – ADATE: ADC Auto Trigger Enable**

Writing one to this bit, results in Auto Triggering of the ADC is enabled.

- **Bit 4 – ADIF: ADC Interrupt Flag**

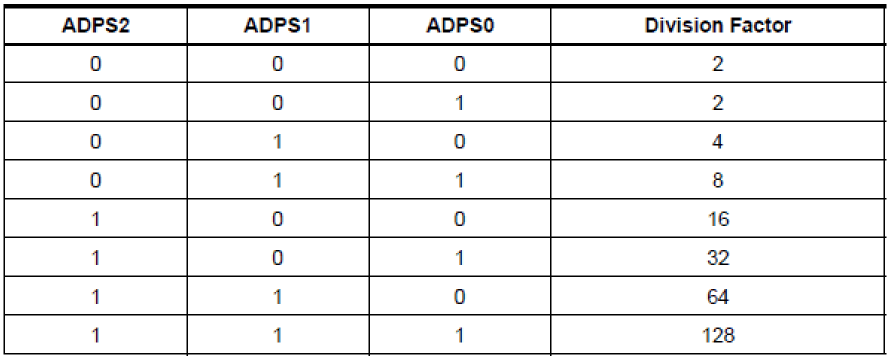
This bit is set when an ADC conversion completes and the Data Registers are updated.

- **Bit 3 – ADIE: ADC Interrupt Enable**

Writing one to this bit, the ADC Conversion Complete Interrupt is activated.

- **Bits 2 : 0 – ADPS2 : 0: ADC Prescaler Select Bits**

These bits determine the division factor between the XTAL frequency and the input clock to the ADC



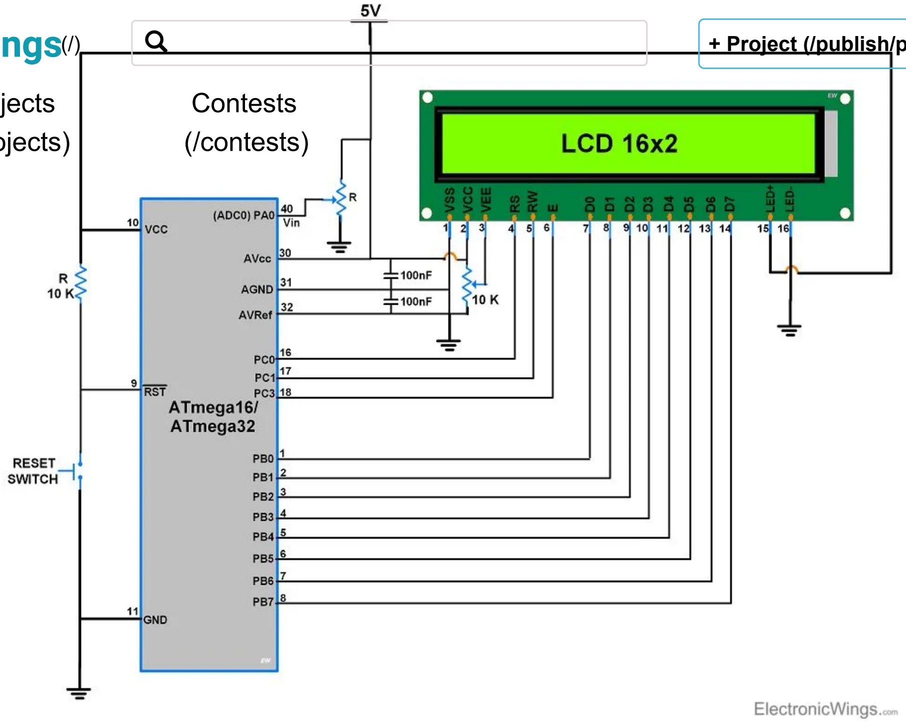
ADPS2	ADPS1	ADPS0	Division Factor
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

We can select any divisor and set frequency Fosc/2, Fosc/4, etc. for ADC, But in AVR, ADC requires an input clock frequency less than 200KHz for max. accuracy. So we have to always take care of not exceeding ADC frequency more than 200KHz.

Suppose your clock frequency of AVR is 8MHz, then we must have to use devisor 64 or 128. Because it gives 8MHz/64 = 125KHz, which is lesser than 200KHz.

ATmega16/32 ADC Interfacing Diagram

Here we are displaying ADC channel 0 values on the 16x2 LCD using ATmega16/32 Microcontroller



Circuit Diagram For Using ADC0 Of ATmega16/32

Steps to Program ATmega16/32 ADC

1. Make the ADC channel pin as an input.
2. Set ADC enable bit in ADCSRA, select the conversion speed using ADPS2 : 0. For example, we will select devisor 128.
3. Select ADC reference voltage using REFS1: REFS0 in ADMUX register, for example, we will use AVcc as a reference voltage.
4. Select the ADC input channel using MUX4 : 0 in ADMUX, for example, we will use channel 0.
5. So our value in register ADCSRA = 0x87 and ADMUX = 0x40.
6. Start conversion by setting bit ADSC in ADCSRA. E.g.ADCSRA |= (1<<ADSC);
7. Wait for conversion to complete by polling ADIF bit in ADCSRA register.
8. After the ADIF bit gone high, read ADCL and ADCH register to get digital output.
9. Notice that read ADCL before ADCH; otherwise result will not be valid.

ATmega16/32 ADC Code

```
#define F_CPU 8000000UL
#include <avr/io.h>
#include <util/delay.h>
#include <stdlib.h>
#include "LCD_16x2_H.h"

void ADC_Init()
{
        DDRA=0x0;                       /* Make ADC port as input */
        ADCSRA = 0x87;                  /* Enable ADC, fr/128  */
        ADMUX = 0x40;                   /* Vref: Avcc, ADC channel: 0 */

}

int ADC_Read(char channel)
{
        int Ain,AinLow;

        ADMUX=ADMUX|(channel & 0x0f);   /* Set input channel to read */

        ADCSRA |= (1<<ADSC);            /* Start conversion */
        while((ADCSRA&(1<<ADIF))==0);   /* Monitor end of conversion inte
```

Video

Components Used

Powered By (https://www.mouser.in?utm_source=electronicswings&utm_medium=display&utm_campaign=mouser-componentslisting&utm_content=0x0)

ATmega 16 ATmega 16	X 1	(https://www.mouser.in/ProductDetail/Microchip-Technology-Atmel/ATMEGA16L-8PU?qs=%2Fha2pyFaduiGCJtTvs2wv8fVZbVAalLu7Iq%2FglTS0tALAx6fMenLvg%3D%3D&utm_source=electronicswings&utm_medium=display&utm_campaign=mouser-componentslisting&utm_content=0x0) Datasheet (/components/atmega-16/1/datasheet)
Atmega32 Atmega32	X 1	(https://www.mouser.in/ProductDetail/Microchip-Technology-Atmel/ATMEGA32-16PU?qs=aqrrBurbvGdpkmgj7RWmsQ%3D%3D&utm_source=electronicswings&utm_medium=display&utm_campaign=mouser-componentslisting&utm_content=0x0) Datasheet (/components/atmega32/1/datasheet)

Components Used

LCD16x2 Display LCD16x2 Display	X 1	(https://www.mouser.com/ProductDetail/Adafruit/1447?qs=XAKIUOoRPe6AClmsjw7y7g%3D%3D&utm_source=electronicswings&utm_medium=display&utm_campaign=mouser-componentslisting&utm_content=0x0)

Downloads

Proteus Simulation file	Download (/api/download/platform-attachment/19)
ATmega16 ADC project file	Download (/api/download/platform-attachment/290)

Comments

Comment

sridhar (/users/sridhar/profile)
2018-10-11 01:43:34

Above code was checked with Proteus, It's working.

Reply Like 1

bolajinathan
(/users/bolajinathan/profile)
2018-11-04 00:51:17

Please which software do you use in drawing your diagrams shown above.

Reply Like

365ngay0khoc
(/users/365ngay0khoc/profile)
2018-11-12 19:32:16 • Edited

You can give me libraly LCD 16x2 , please!

Reply Like 1

emanueltiec
(/users/emanueltiec/profile)
2018-11-27 10:06:46

how you do this schematics with avr?

Reply Like

shuklabhinav1011
(/users/shuklabhinav1011/profile)
2018-12-26 18:18:52

Hello sir , i have one query , how to convert float value to string by using dtostrf function , i follwed the syntax properly but still the flost value was not displayed

Reply Like

authorized
(/users/authorized/profile)
2018-12-31 19:33:46

char * dtostrf(double __val, signed char __width, unsigned char __prec, char * __s)

__width and __prec decides the output size and dot(.) position e.g. if __width is 7 and __prec is 3 then o/p will come as "000.000"
__s is result buffer we have to pass to function. Note: keep always result buffer size > __width

definition (referred from microchip website) :
The dtostrf() function converts the double value passed in __val into an ASCII representation that will be stored under __s. The caller is responsible for providing sufficient storage in __s.

Conversion is done in the format "[-]d.ddd". The minimum field __width of the output string (including the possible dot(.) and the possible sign for negative values) is given in __width, and __prec determines the number of digits after the decimal sign. __width is signed value, negative for left adjustment.

The dtostrf() function returns the pointer to the converted string __s.

Reply Like 1

vivekkaundal
(/users/vivekkaundal/profile)
2019-02-28 12:03:31

The code is not working with Simullide. Pls help

Reply Like

chandannegich
(/users/chandannegich/profile)
2019-03-03 18:04:18

sir how a port start d7 pin our first pin and d0 the last pin.
i want lcd data pin d7-d0 connect the pcb board
plz tell me a example number send to the port and binary start from d7 pin

Reply Like

vikaskumarsah23
(/users/vikaskumarsah23/profile)

2019-08-14 12:12:07

sir, in adxl335 sensor , how to use three channel??

Reply Like

wbarbew (/users/wbarbew/profile)
2019-10-04 23:50:55

a mistake Step size with Vref=5V : 5/1023 = 4.88 mV.
5/1024 is right

Reply Like 1 👍

jony071054 (/users/jony071054/profile)
2019-12-09 21:18:24

how can i show factorial number like 230.5v in lcd using ur code ? pls help

Reply Like

rontojoy (/users/rontojoy/profile)
2020-05-12 17:12:41

Sir can you share multiple adc interrupt code

Reply Like

hiteshpatidar20 (/users/hiteshpatidar20/profile)
2020-05-15 09:26:36 • Edited

LCD_String(" ");
how am i getting wrong output without using this function above
what is the pupose of this function here?

Reply Like

mcu2010 (/users/mcu2010/profile)
2021-02-14 13:46:55

Thank you for sharing.

Reply Like

AndreyPustoshkin (/users/AndreyPustoshkin/profile)
2023-02-02 18:25:10

```
int ADC_Read(char channel)
{
   ADMUX |= channel & 0x0f;

ADCSRA |= 1<<ADSC;
while((ADCSRA&(1<<ADIF))==0);

_delay_us(10);

//return((ADCH<<8)+ADCL); // doesn't work right
return(ADCL+(ADCH<<8)); // works fine
}
```

Could anybody answer why? Thanks!

Reply Like

authorized (/users/authorized/profile)
2023-07-01 11:31:30

in 9th step of above program, it is given that
read ADCL before ADCH; otherwise result will not be valid.

Reply Like

About Us (/about)

Connect On:

Facebook(https://www.facebook.com/electronicwings)

EW ElectronicWings(/)

+ Project (/publish/project)

Platforms (/explore)

Projects (/projects)

Contests (/contests)

Business Offering (/business-services)

Host Platform (/launch-platform)

Contact Us (/contactus)

LinkedIn(https://www.linkedin.com/company/electronicwin

Youtube(https://www.youtube.com/channel/UCNdqkukBtk4

Instagram(https://www.instagram.com/electronicwings_co
igshid=1cip10jijttko)

Terms of Service (/terms-of-service)
Cookies Policy (/cookie-policy)
Privacy Policy (/privacy-policy)

ElectronicWings
© 2023