Custom Search Search

- home
- electronics
- howto
- tools
- knowledge
- forum

ADC in Atmel microcontrollers (using Atmega8)

This tutorial is strictly with Atmega8 as a reference. However, the same concept can be used across most AVR devices. Atmega8 features 10 bit ADC and provides 6 analog pins; meaning you can connect up to 6 different analog devices to microcontroller (TQFP and QFN/MLF package has additional 2 input pins). Separate reference voltage pins AVcc and Aref are provided and voltagein AVcc must not differ more than ± 0.3V from Vcc.Aref is voltage reference pin which can be optionally used if we need external voltage reference. AVROutput from ADC can either be left adjusted, or right adjusted (we will see what it is in later discussions). ADC can be set to either free running mode where it continuously takes samples and provides conversion or just a single convert & stop mode.

Atmel microcontrollers are register based microcontrollers. Atmega8 ADC is also controlled by registers ADCSRA, ADMUX, ADC data registers; ADCL and ADCH. We will understand the characteristics of each register, one at a time.

ADCSRA Register

This register is responsible for enabling ADC, start ADC converting, prescaler selection and interrupt control.

ADCSRA bit definition:

Bit	7	6	5	4	3	2	1	0
ADCSRA	ADEN	ADSC	ADFR	ADIF	ADIE	ADPS2	ADPS1	ADPS0
Read / Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

We will see what each of these bits means:

ADC Enable (ADEN): ADC is turned on by setting this bit to logic 1 and turned off by writing logic 0.

ADC Start Conversion (ADSC): Setting this bit to logic 1 tells ADC to start the conversion. However, if Single conversion mode is selected, then we need to set this to logic 1 each time we need a conversion to start. Once the conversion is complete, the hardware sets it to zero. This can be considered as a flag to check if the conversion is complete.

ADC Free Running Select (ADFR): This bit can be used to toggle ADC into free running mode or single conversion mode. Setting it to 1 activates ADC in free running mode and clearing it sets it to single conversion mode.

Note: Some AVR microcontrollers have ADATE instead of ADFR. ADATE serves more complex purpose and used to enable Auto trigger by setting trigger select bits. Best way to know if your microcontroller uses ADFR or ADATE is through the datasheet, although you can verify the same from particular device header file.Higher end AVR versions may have additional ADCSRB register along with ADCSRA register

ADC Interrupt Flag (ADIF): This is an interrupt bit which is set to 1 by hardware once conversion is complete and Data registers are updated.ADC conversion interrupt is executed if ADIE bit and I-bit in SREG is set. ADIE bit is Bit 3 in ADCSRA register and SREG is the status register. To avoid further confusion, status register contains result of the latest arithmetic instruction executed. I-bit is bit 7 in Status register and known as Global interrupt enable.
Note: ADIFbit can be manually cleared by setting a 1 to the bit. Read it again, you are clearing this bit by writing 1 and not 0 as in the usual case. This is not an error in datasheet, but a well planned approach to clear a bit without disturbing other bits.

Bit 3 – ADIE: ADC Interrupt Enable: When this bit is written a 1 and the I-bit in SREG is set, the ADC Conversion Complete Interrupt is activated.

ADC Prescaler Select Bits (ADPS): **ADPS2**, **ADPS1** and **ADPS0** bits are used to set circuit clock frequency. For ADC circuitry to work at its maximum resolution needs to be supplied with 50 kHz to 200 kHz frequency as per the datasheet; but the system clock will be generally higher (8 MHz, 10 MHz, 16 MHz etc). To reduce it to required frequency we use ADC prescaler bits. Suppose we have system clock with frequency 10Mhz (10000000 Hz) and set division factor to 64, then ADC clock frequency is 10000000/64 = 156250Hz = 156.25 KHz, which is between 50 to 200 KHz as mentioned in the datasheet.

The table below is the complete prescaler setting for these 3 bits **ADPS2**, **ADPS1** and **ADPS0**

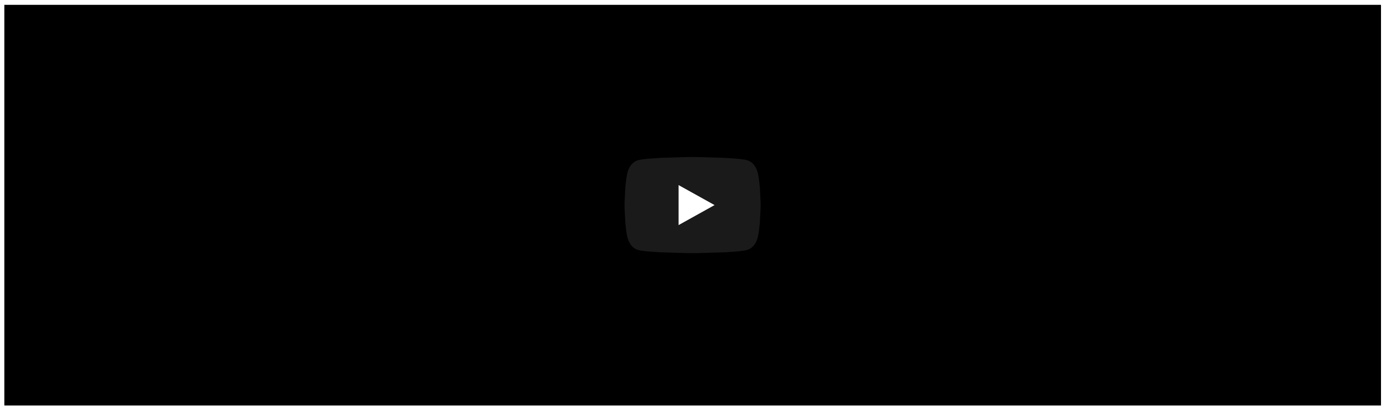
ADPS2	ADPS1	ADPS0	Division Factor
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

Tutorial index:

«Previous - § - Next»

G+

Featured Videos



Advertisements

Recent Articles

Atmega8 Development Board

A great step-by-step tutorial on building your own Atmel AVR based Atmega8 development board. The board is ideal for beginners with detailed explanation and pictures More...

L293D Motor Driver

For robots to do work, you need to know how to control a motor. L293D is a cleverly packed IC which can control two DC motors in both directions: forwards and reverse. Here is a detailed explanation of building a board based on L293D ICMore...

Hobby Servo Tutorial

Servo Motor is a device which uses error-sensing feedback signals to determine and control the position of a motor shaft. The term "servomechanism" closely relates to servo motors..More...

Blinking LED Tutorial

This is similar to what we achieve in any "Hello World" program. However, it is not just limited to blinking LED but scratches the surface of AVR-GCC programming... More...

Kindly Donate

If this site has helped you, then kindly consider a Donation to say "Thank You!!". Donation might help us keep all this information available for free and also pay for the resources.

If that is difficult, then a simple "Hi" in the forum would still do good :)

Five US Dollars $5.00 USD ▼

Pay Now

HOME | ELECTRONICS | HOWTO | KNOWLDGE | TOOLS | FORUM
Contact Us | Disclaimer & Privacy | Sitemap