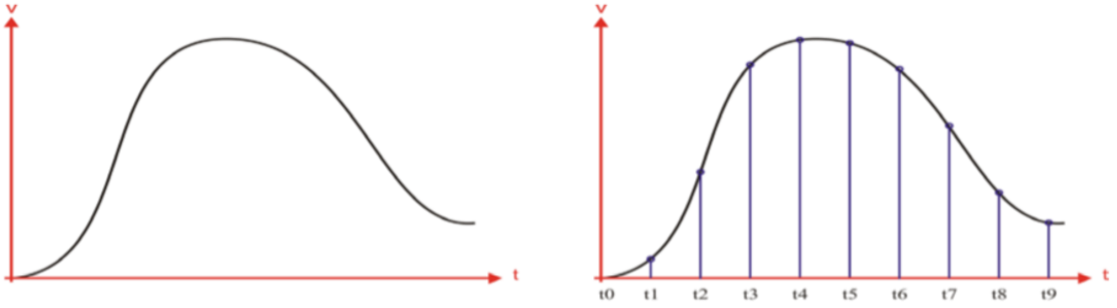
Analog-Digital Converter

Analog-digital conversion

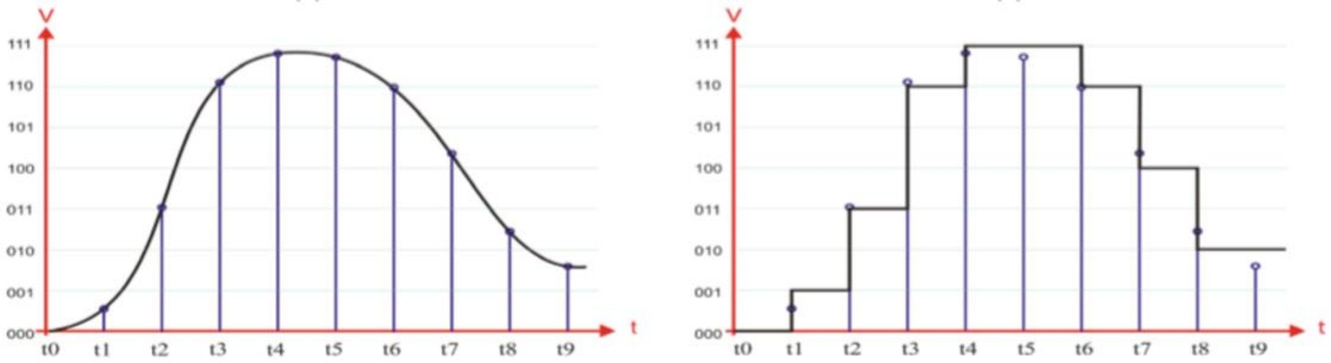
- Analog to digital process → Sampling and quantization
- Sampling → Read/Take input voltage values (samples)



Sampling frequency (Fs) = 2 x F (Vin) → Teorema de Nyquist

- Quantization → Assign a digital value to the sampled value (n bits)

  ADC with n= 3 bits → 2^3 = 8 digital values (0 to 7)



Analog-digital conversion

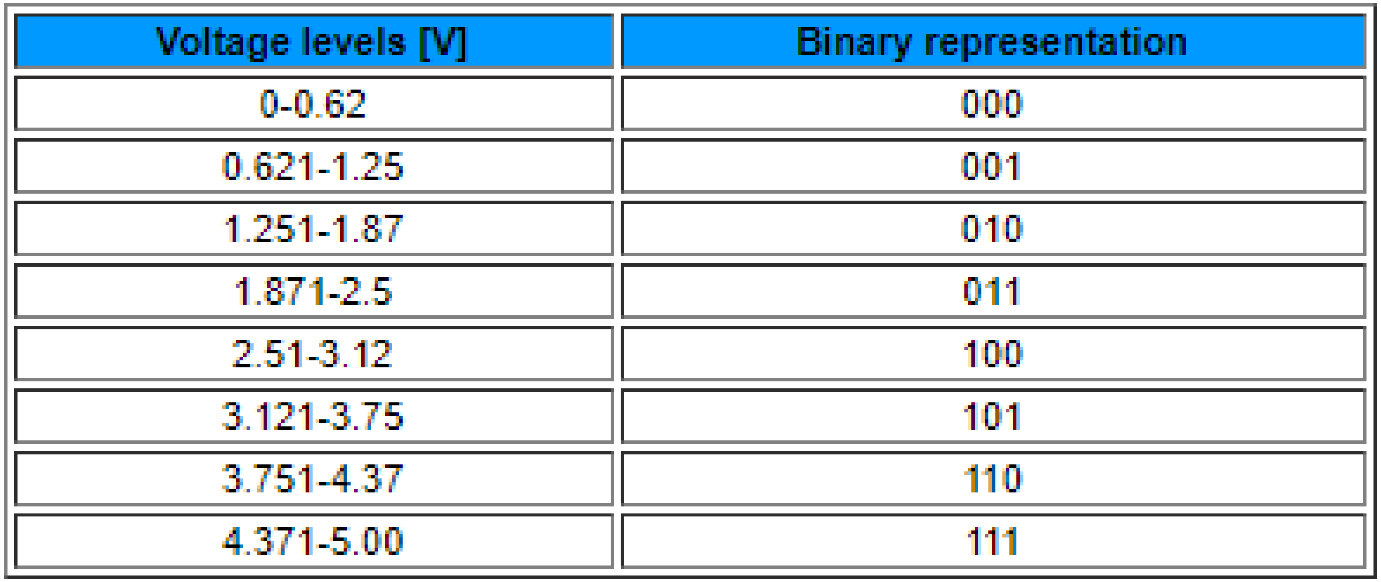
- ADC resolution is determined by:
  - Number of bits → number of steps
  - The V range
  - Both of them → Step size

Voltage range: Vref to GND, n=3 → Resolution = $(\overset{\text{Vref}}{\overset{\downarrow}{5V}}-0V)/\ \overset{\text{Nº steps}}{\overset{\downarrow}{2^3}} = 0{,}625V = \overset{\text{Step size}}{\overset{\downarrow}{625mV}}$

- Resolution definition → the smallest incremental voltage that can be recognized and causes a change in the digital output.

Example: 625mV

Analog-digital conversion



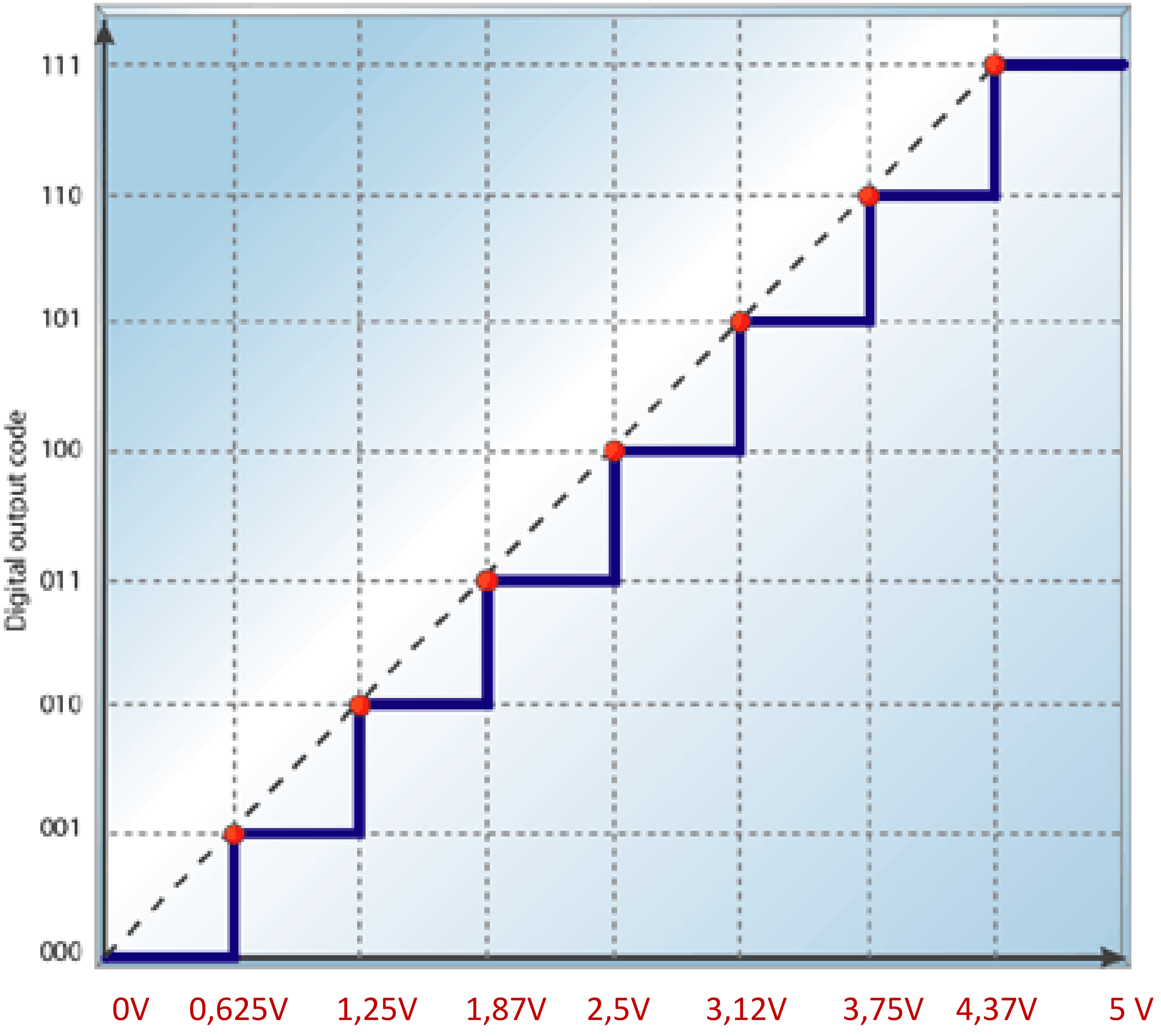
Voltage levels [V]	Binary representation
0-0.62	000
0.621-1.25	001
1.251-1.87	010
1.871-2.5	011
2.51-3.12	100
3.121-3.75	101
3.751-4.37	110
4.371-5.00	111

Range = 5V-0V = 5V N = 3 bits

Resolution = $5/2^3$ = 0,62

http://www.microcontrollerboard.com/analog-to-digital-converter.html

Analog-digital conversion



Source: https://www.embedded.com/understanding-analog-to-digital-converter-specifications/

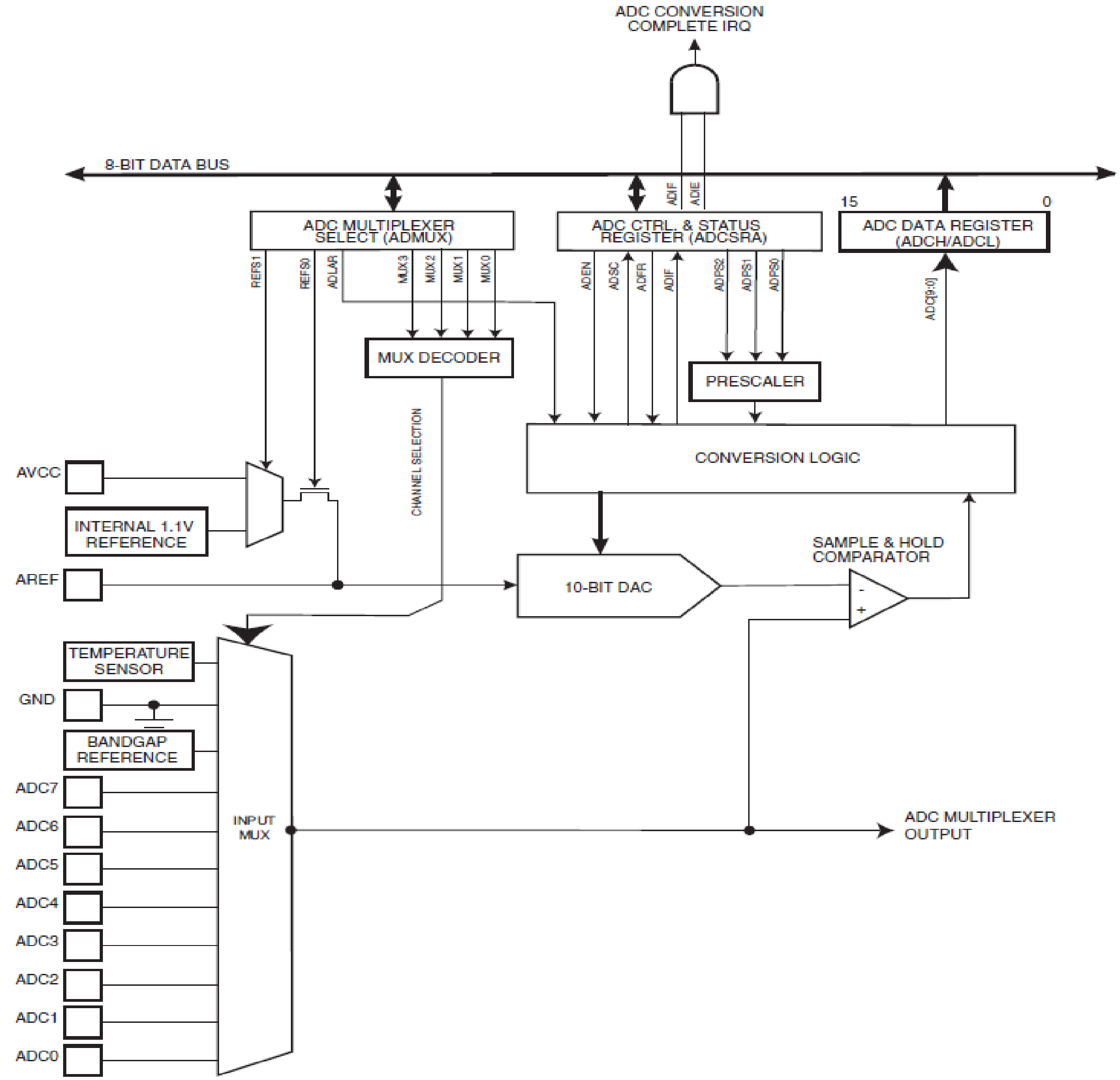
Analog-Digital Converter (ADC ATMega328)

Analog-digital converter (ADC)-ATMEGA328

- 10-bit successive approximation ADC
- 8-channel Analog Multiplexer in PORTC (ADC0-ADC5)
- ADC has a separate analog supply voltage pin, AVCC.
- Internal reference voltages of 1.1V or AVCC are provided On-chip and an external Aref
- The 10-bit result is stored in →

  ADC Data Register: ADCH and ADCL.
- The ADC triggers an interrupt when a conversion completes.
- Free Running or Single Conversion Mode

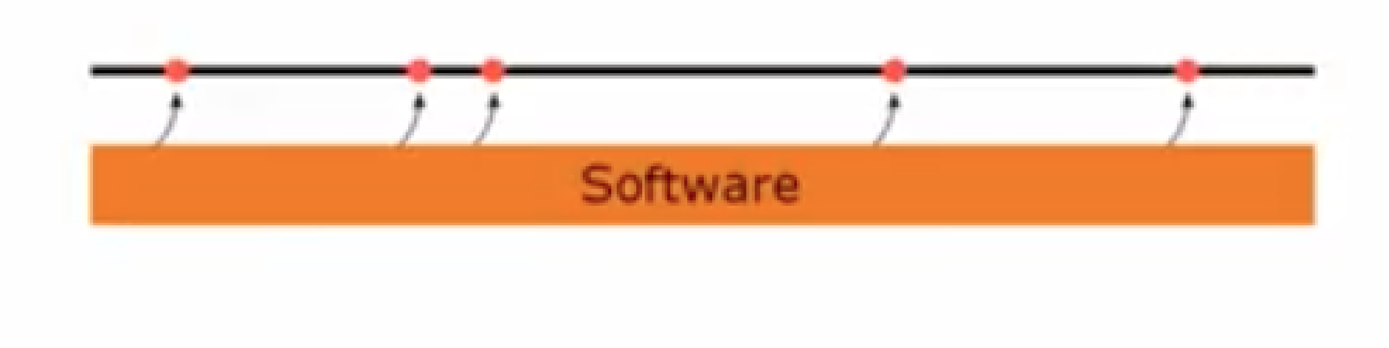
Analog-digital converter (ADC)



Analog-digital converter (ADC): Modes

Single Conversion Mode (Polling or Interrupt)

- You have to initiate each conversion (setting bit ADSC in ADCSRA).
- When the conversion is finished the flag ADIF is set and ADSC is cleared by hardware.
- Result is placed in the ADC Data register pair (16-bit) and no new conversion is started.
- Previously to start a new conversion ADIF must be clear by software (writting '1').

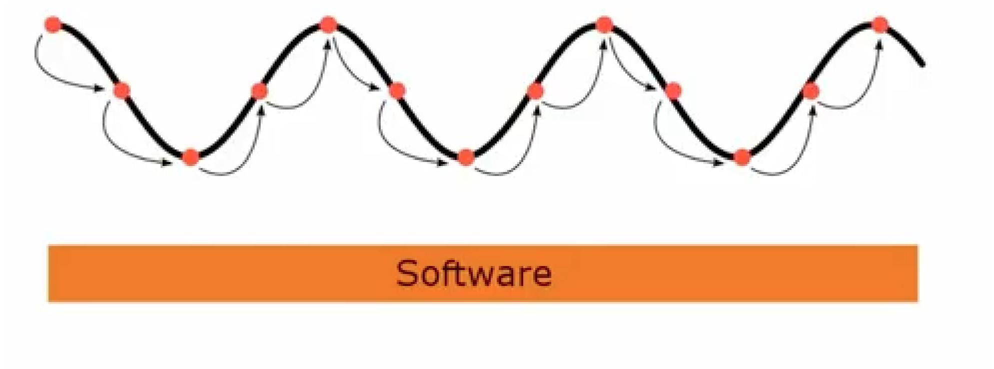


Source: https://microchipdeveloper.com/8avr:adcopmodes

Analog-digital converter (ADC): Modes

Free Running Mode (Interrupt)

- First conversion starts by writing a 1 to ADCSRA.ADSC
- A new conversion starts when the previous has completed (Interrupt Flag is used as a trigger source)
- The ADC register must be read as son as posible, before a new conversion starts

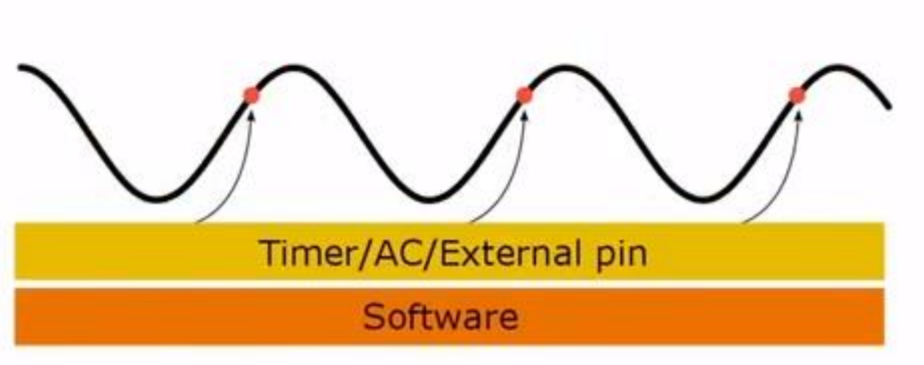


Source: https://microchipdeveloper.com/8avr:adcopmodes

Analog-digital converter (ADC): Modes

Auto triggering Mode

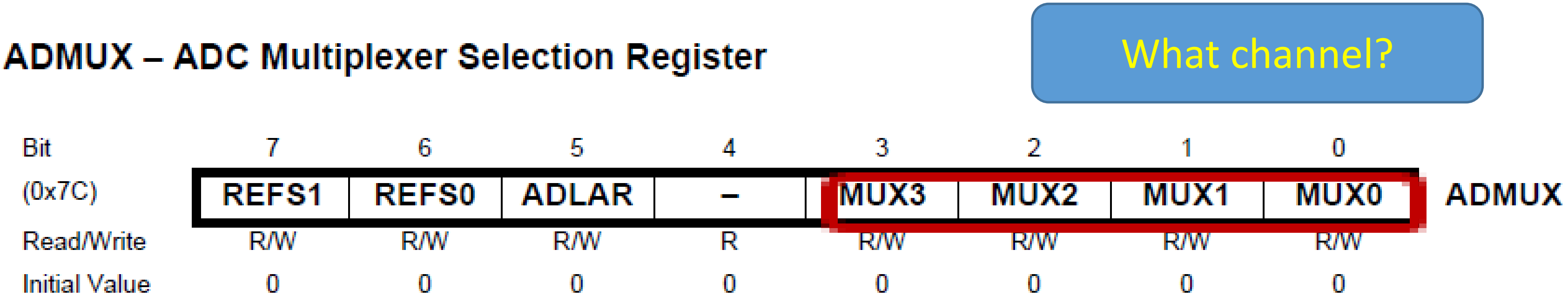
- Controlled by a peripheral: Timer, comparator, external interrupt
- Start a new conversion when the previous has completed (Interrupt Flag is used as a trigger source)
- The ADC register must be read as son as posible, before a new conversion starts



Source: https://microchipdeveloper.com/8avr:adcopmodes

Analog-digital converter (ADC): Register ADMUX

What channel?



ADMUX – ADC Multiplexer Selection Register

Bit	7	6	5	4	3	2	1	0	
(0x7C)	REFS1	REFS0	ADLAR	–	MUX3	MUX2	MUX1	MUX0	ADMUX
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- MUX2-MUX0: selects which analog inputs are connected to the ADC

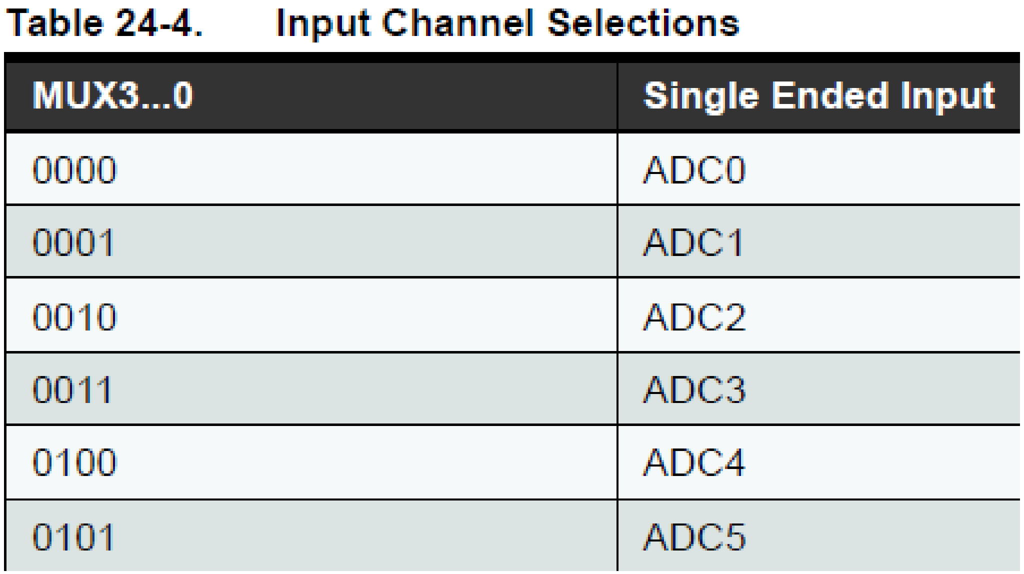
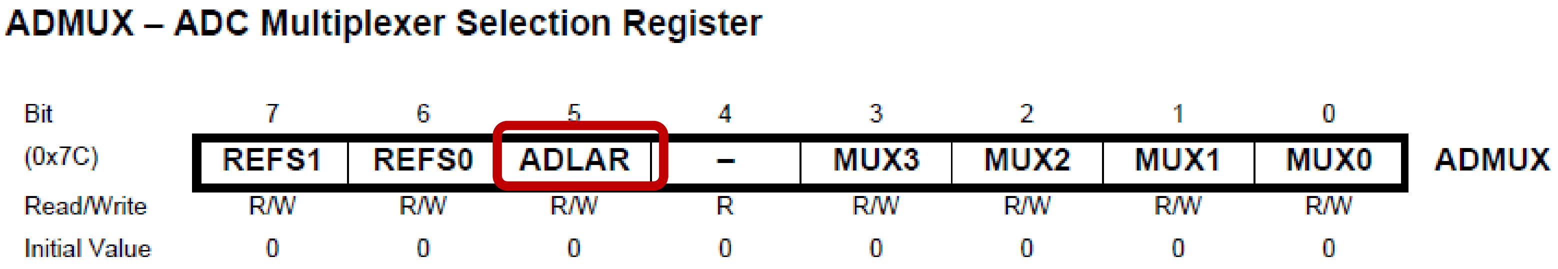


Table 24-4. Input Channel Selections

MUX3...0	Single Ended Input
0000	ADC0
0001	ADC1
0010	ADC2
0011	ADC3
0100	ADC4
0101	ADC5

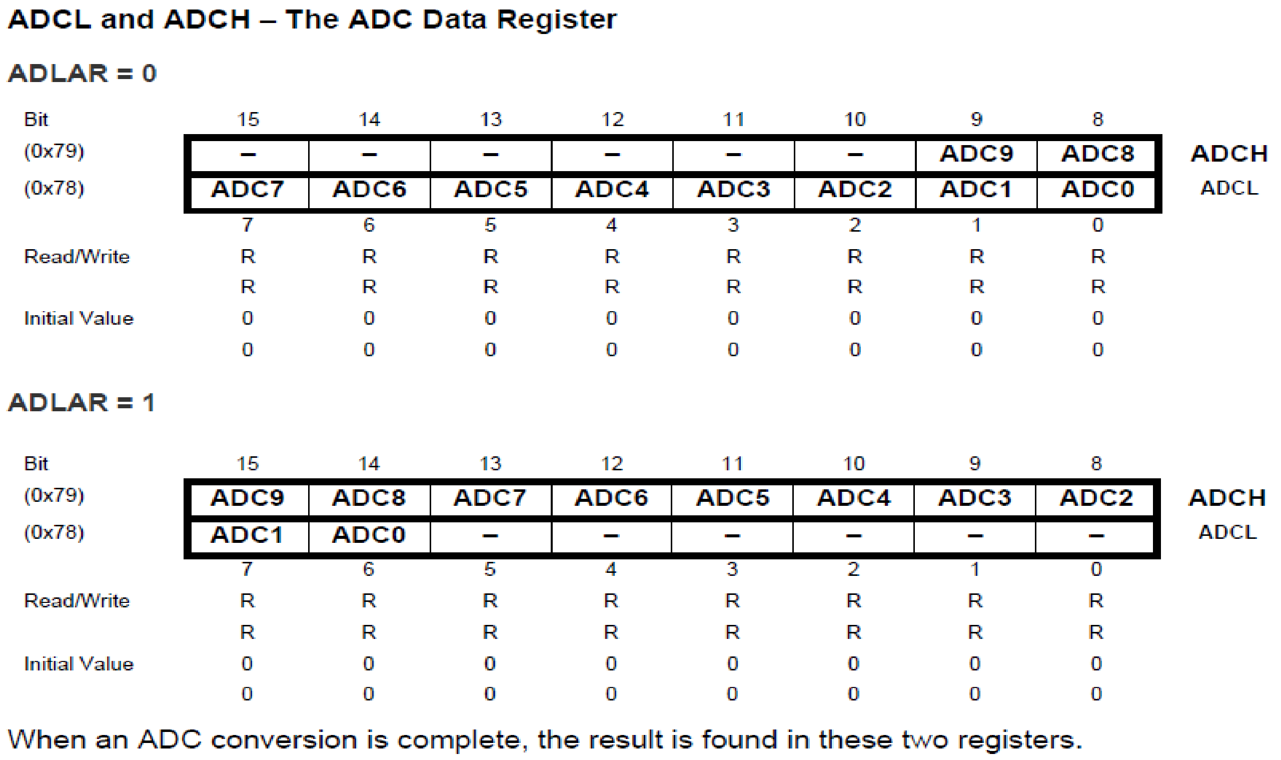
Analog-digital converter (ADC): Register ADMUX



ADMUX – ADC Multiplexer Selection Register

Bit	7	6	5	4	3	2	1	0	
(0x7C)	REFS1	REFS0	ADLAR	–	MUX3	MUX2	MUX1	MUX0	ADMUX
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- ➢ ADLAR: Write one to ADLAR to left adjust the result. Otherwise, the result is right adjusted.



ADCL and ADCH – The ADC Data Register

ADLAR = 0

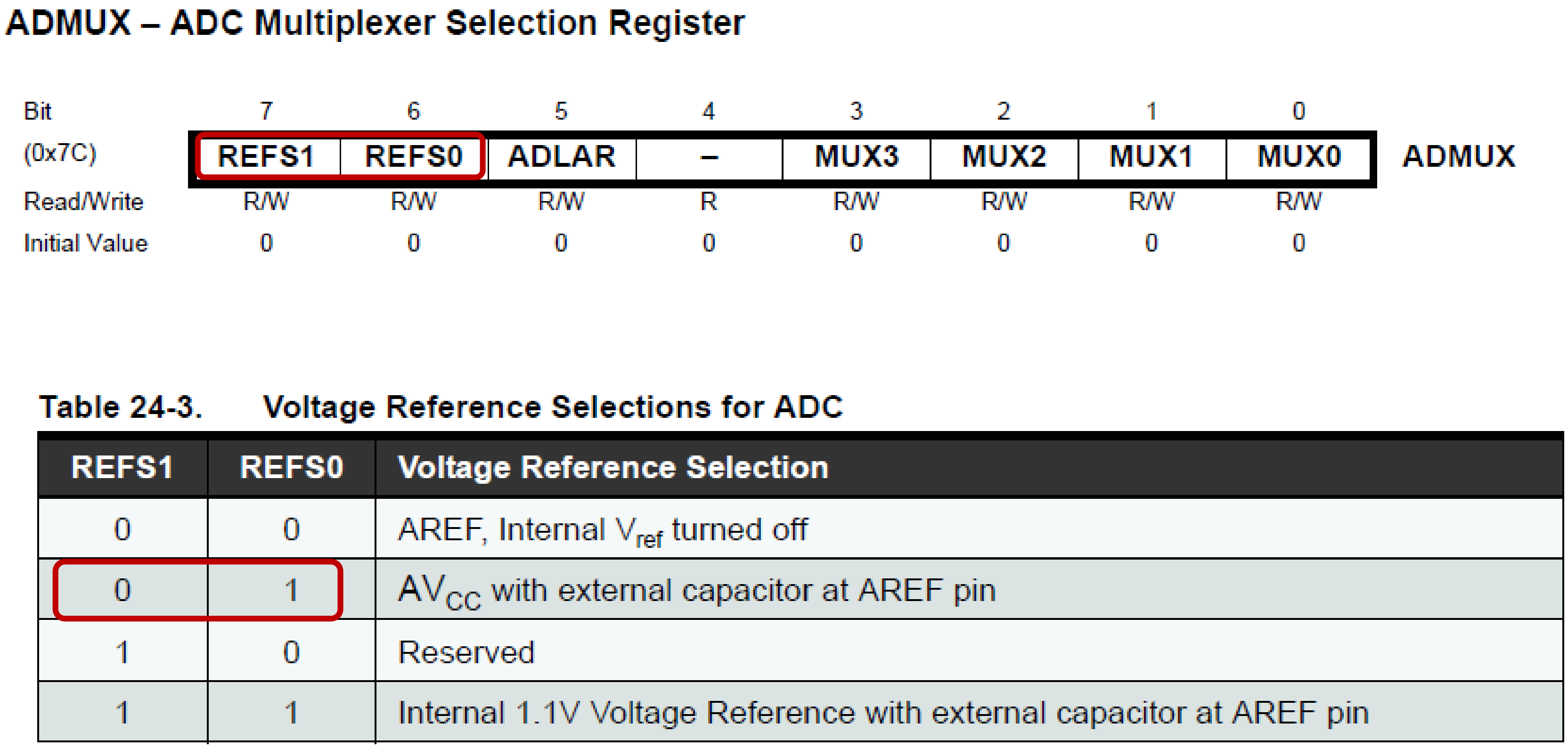
Bit	15	14	13	12	11	10	9	8	
(0x79)	–	–	–	–	–	–	ADC9	ADC8	ADCH
(0x78)	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	ADCL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R	
	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

ADLAR = 1

Bit	15	14	13	12	11	10	9	8	
(0x79)	ADC9	ADC8	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADCH
(0x78)	ADC1	ADC0	–	–	–	–	–	–	ADCL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R	
	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

When an ADC conversion is complete, the result is found in these two registers.

Analog-digital converter (ADC): Register ADMUX



ADMUX – ADC Multiplexer Selection Register

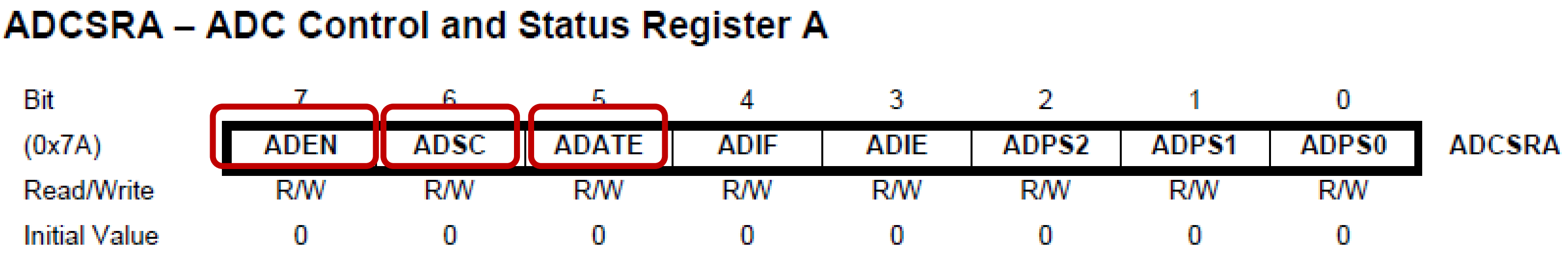
Bit	7	6	5	4	3	2	1	0	
(0x7C)	REFS1	REFS0	ADLAR	–	MUX3	MUX2	MUX1	MUX0	ADMUX
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Table 24-3. Voltage Reference Selections for ADC

REFS1	REFS0	Voltage Reference Selection
0	0	AREF, Internal V_{ref} turned off
0	1	AV_{CC} with external capacitor at AREF pin
1	0	Reserved
1	1	Internal 1.1V Voltage Reference with external capacitor at AREF pin

Arduino connection

Analog-digital converter (ADC): Register ADCSRA



ADCSRA – ADC Control and Status Register A

Bit	7	6	5	4	3	2	1	0	
(0x7A)	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ADCSRA
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	

- **ADEN**: ADC Enable. Writing this bit to one enables the ADC.

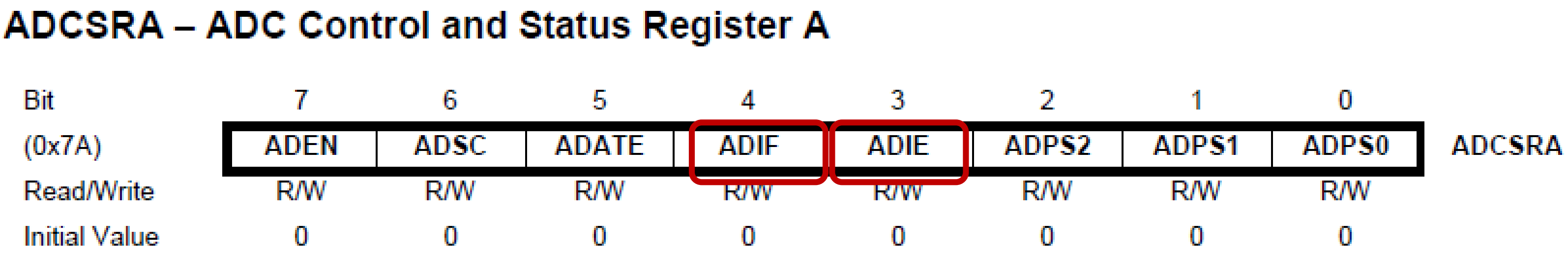
- **ADSC:** ADC Start Conversion.
  - Single conversión: ‘1’ starts a new conversion
  - Free running: ‘1’ starts the first conversion

First conversión (initial) → 25 ADC clock cycles
Normal conversion → 13 ADC clock cycles

- **ADATE: ADC Auto Trigger Enable.** The ADC will start a conversion on a positive edge of the selected trigger signal. The trigger source is selected by setting the ADC Trigger Select bits, ADTS in ADCSRB.

If ADATE=‘0’ → Single mode
If ADATE=‘1’ → (trigger default) Free running mode source

Analog-digital converter (ADC): Register ADCSRA



ADCSRA – ADC Control and Status Register A

Bit	7	6	5	4	3	2	1	0	
(0x7A)	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ADCSRA
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	

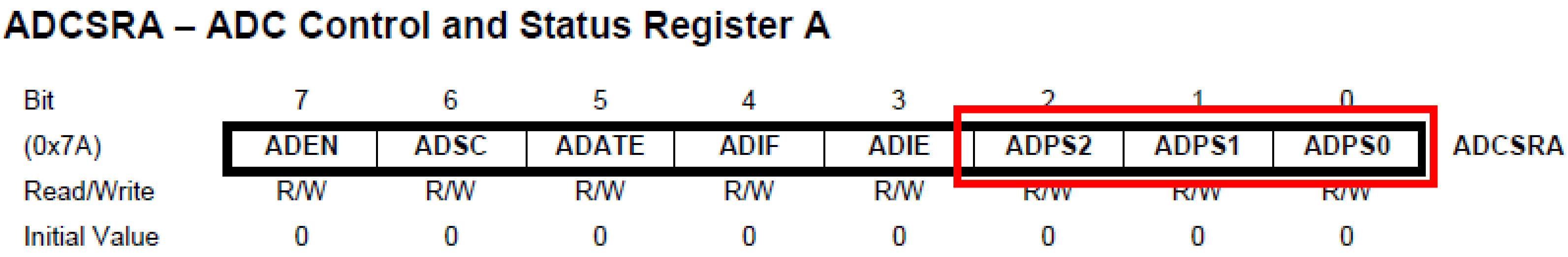
- **ADIF: ADC Interrupt Flag**

This bit is set when an ADC conversion completes and the Data Registers are updated. Clearing by software previous to a new conversion in “Single conversion mode”

- **ADIE: ADC Interrupt Enable**

When this bit is written to one and the I-bit in SREG is set, the ADC Conversion Complete Interrupt is activated.

Analog-digital converter (ADC)



ADCSRA – ADC Control and Status Register A

Bit	7	6	5	4	3	2	1	0	
(0x7A)	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ADCSRA
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	

- **ADPS[2:0]: ADC Prescaler Select Bits.** These bits determine the division factor between the system clock frequency and the input clock to the ADC.

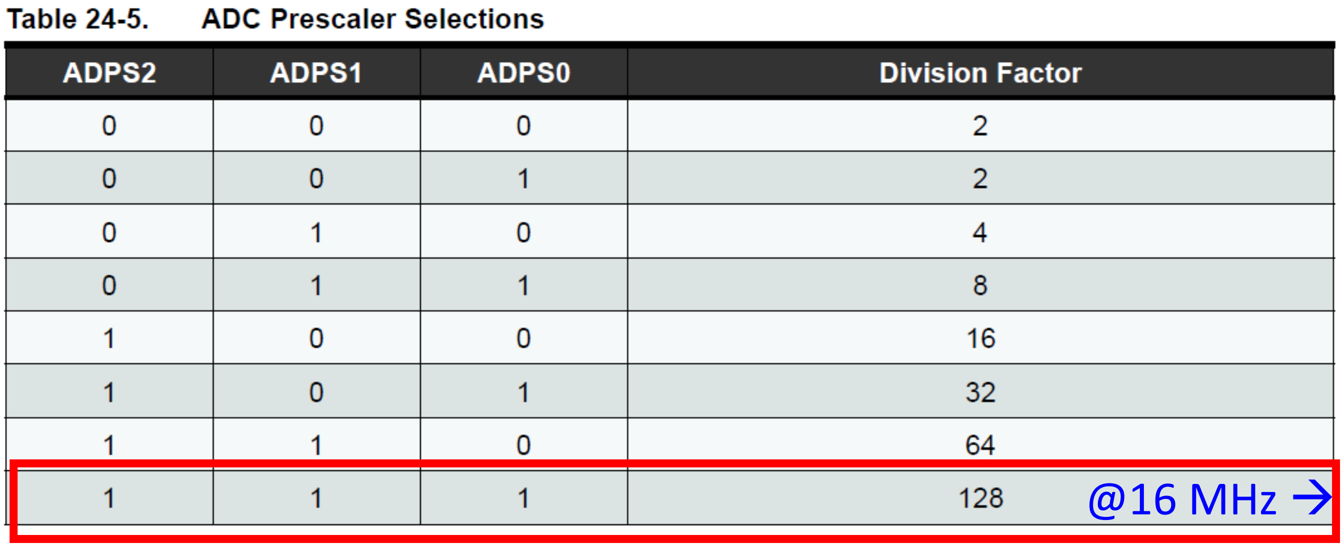


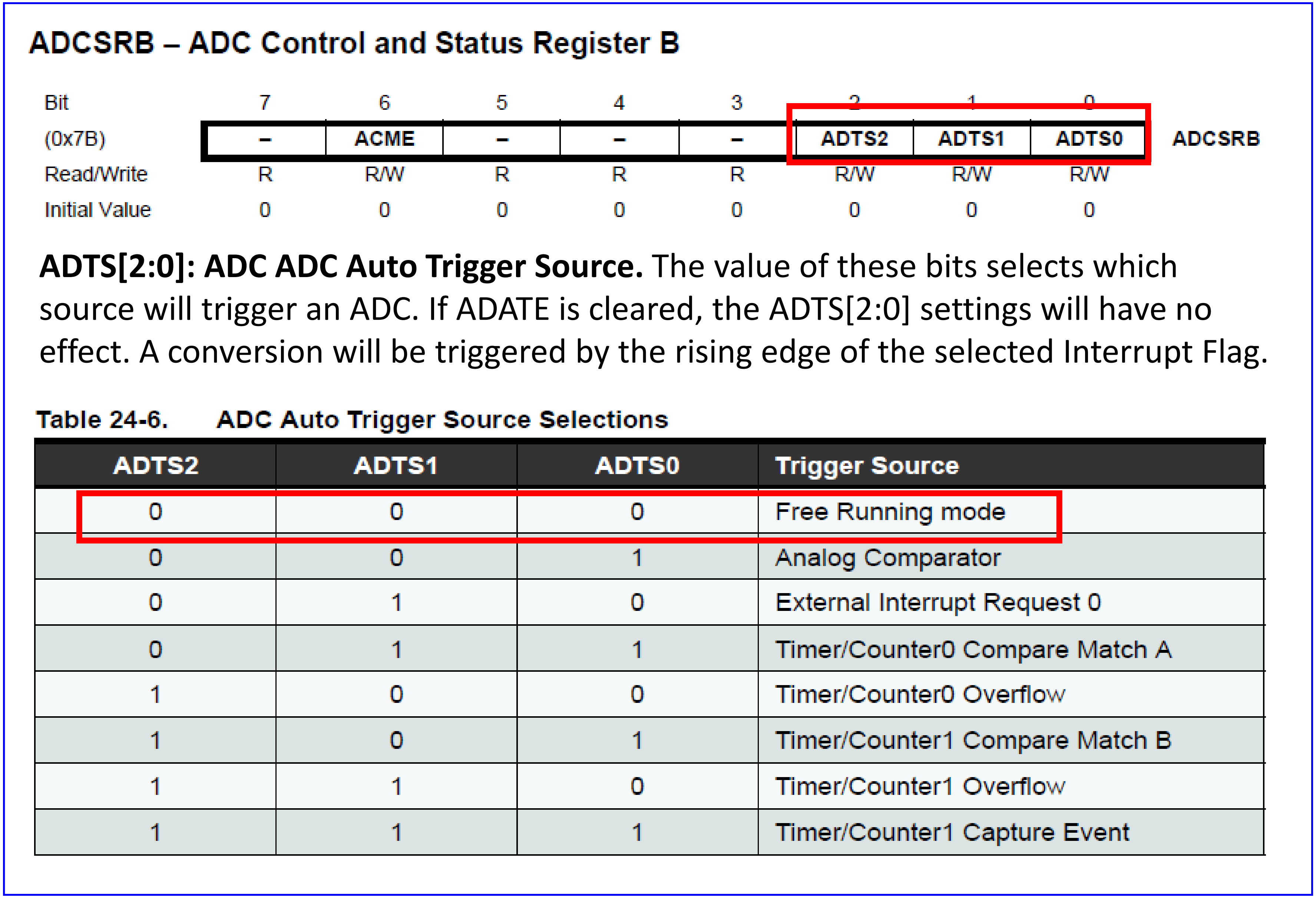
Table 24-5. ADC Prescaler Selections

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

@16 MHz → 125KHz

- The successive approximation circuitry requires an input clock frequency between 50kHz and 200kHz to get maximum resolution

Analog-digital converter (ADC)



ADCSRB – ADC Control and Status Register B

Bit	7	6	5	4	3	2	1	0	
(0x7B)	–	ACME	–	–	–	ADTS2	ADTS1	ADTS0	ADCSRB
Read/Write	R	R/W	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

ADTS[2:0]: ADC ADC Auto Trigger Source. The value of these bits selects which source will trigger an ADC. If ADATE is cleared, the ADTS[2:0] settings will have no effect. A conversion will be triggered by the rising edge of the selected Interrupt Flag.

Table 24-6. ADC Auto Trigger Source Selections

ADTS2	ADTS1	ADTS0	Trigger Source
0	0	0	Free Running mode
0	0	1	Analog Comparator
0	1	0	External Interrupt Request 0
0	1	1	Timer/Counter0 Compare Match A
1	0	0	Timer/Counter0 Overflow
1	0	1	Timer/Counter1 Compare Match B
1	1	0	Timer/Counter1 Overflow
1	1	1	Timer/Counter1 Capture Event

ADC Steps: Single mode (Polling)

1. Configure the A/D module:
    - Clear a previous setup
    - Configure voltage reference
    - Set a aligment format for the value
    - Select A/D conversion clock (Prescaler)
    - Turn on (Enable) the A/D module
    - Set the single mode

A function is recommended

2. Read the conversion:
    - Clear flag ADIF writting ‘1’ .
    - Clear a previous selected channel.
    - Select A/D input channel
    - Starts a new conversion, ADSC=‘1’.
    - Wait for A/D conversion to complete polling the bit ADIF
    - Read A/D result register ADC.

A function is recommended

3. For a new conversion go to step 2.

ADC steps: Single mode (Interrupt)

1. Configure the A/D module:
   - Clear a revious setup
   - Configure voltage reference
   - Set the aligment
   - Select A/D conversion clock
   - Turn on A/D module

A function is recommended

2. Enable A/D interrupt + sei()
3. Start a conversion:
   - Select A/D input channel
   - Start the conversión, set the ADSC bit
4. Add the ISR function
   - Read the value in ADC register
   - (Flag ADIF is cleared by the ISR)
   - (Optional) starts a new conversion or disable interrupt

Waiting for the A/D interrupt

ADC steps: Free run mode (Interrupt)

1. Configure the A/D module:
   - Configure voltage reference
   - Select A/D conversion clock
   - Select Free run:
     - Enable interrupt (ADIE)
     - Free run (ADATE)
     - Auto-trigger mode (ADTS)
   - Turn on A/D module

A function is recommended

2. Global interrupt → Sei()
3. Select A/D input channel
4. Trigger only the first conversion → ADSC
5. Add the ISR
   - Read the value