ADC programming

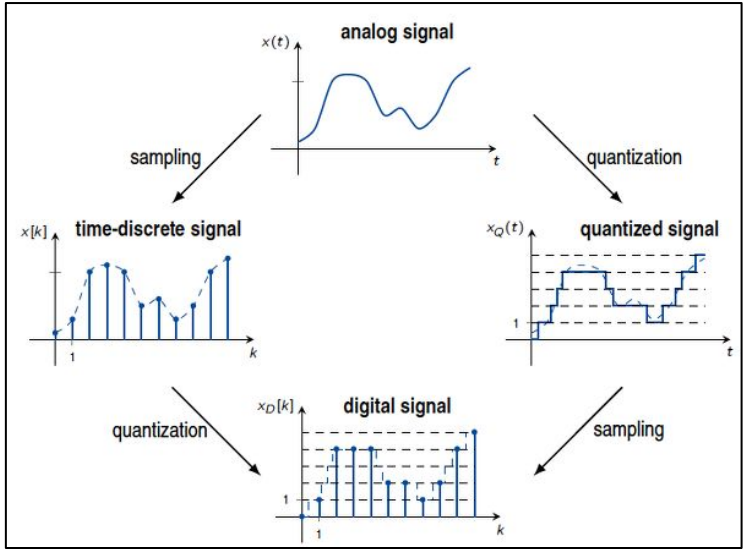
By: Yehia M. Abu Eita

Outlines

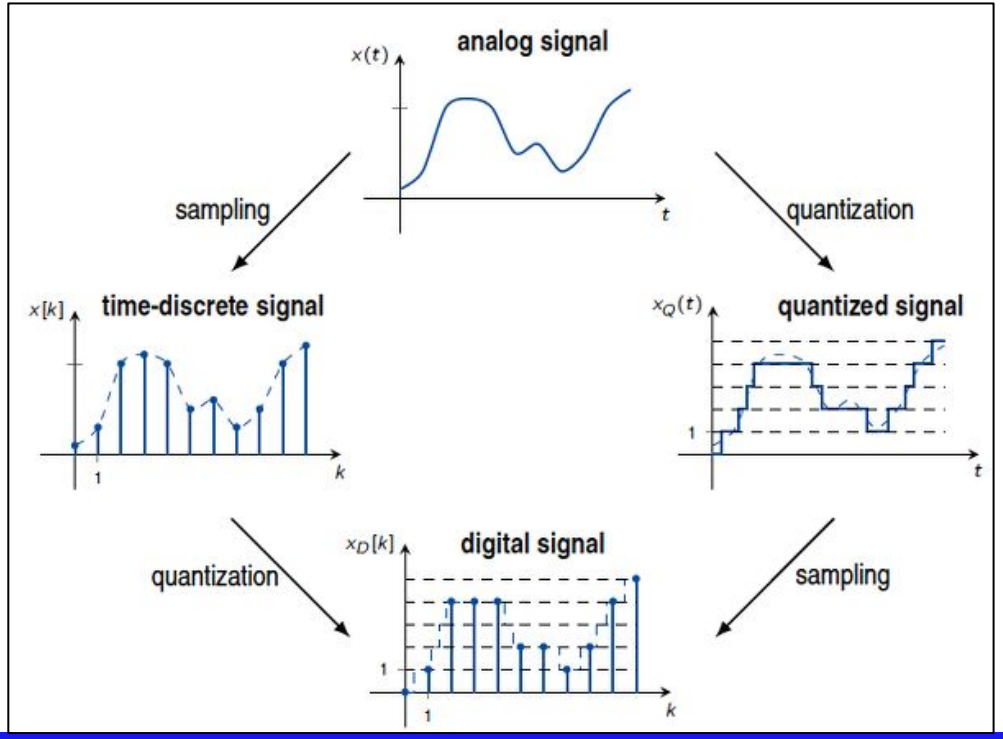
- Introduction
- ADC calculations
- Types of ADC
- Block diagram
- ATmega32 ADC registers
- Steps to program ATmega32 ADC

Introduction

- **ADC** is acronym for **A**nalog to **D**igital **C**onverter.
- It is used to **convert analog** signals into **digital** signals.
- **To convert analog signal to digital**:
  - **Sample and Hold**
  - **Quantization**
  - **Encoding**
- **Sampling rate:**
  - It is number of samples per second, **$F_s >= 2F_m$ (Nyquist Criteria)**
- **ADC resolution:**
  - It is how much **precision** can an ADC convert



ADC calculations



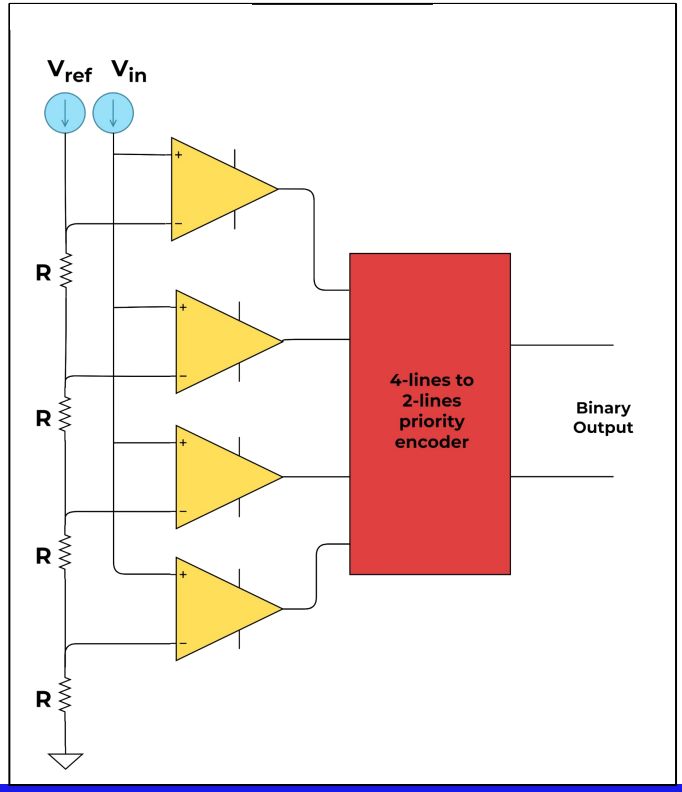
$$\Delta = \frac{V_{ref}}{2^n} \ (Step\ size)$$

$$V_{digital} = number\ of\ steps$$

$$V_{Analog} = V_{digital} \times \Delta$$

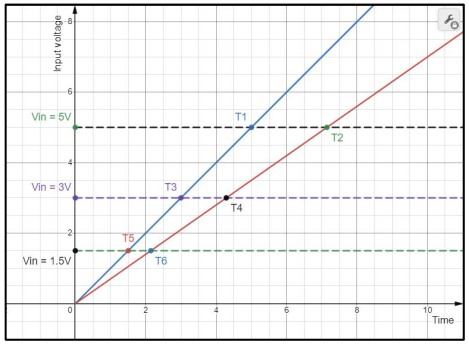
Types of ADC

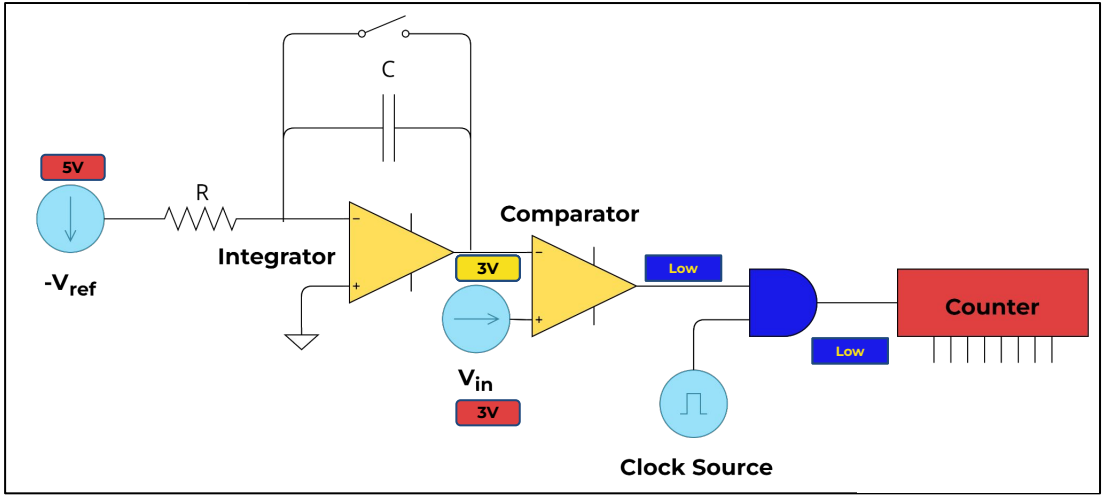
- **Flash ADC**
  - The fastest ADC
  - It has constant conversion time
  - Expensive
  - Large size



Types of ADC

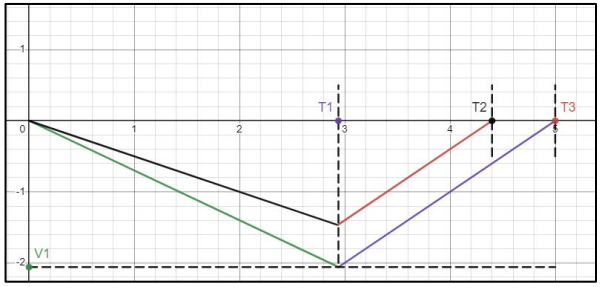
- **Single-slope ADC**
  - Fast conversion time
  - Conversion time directly proportional with V_{in}

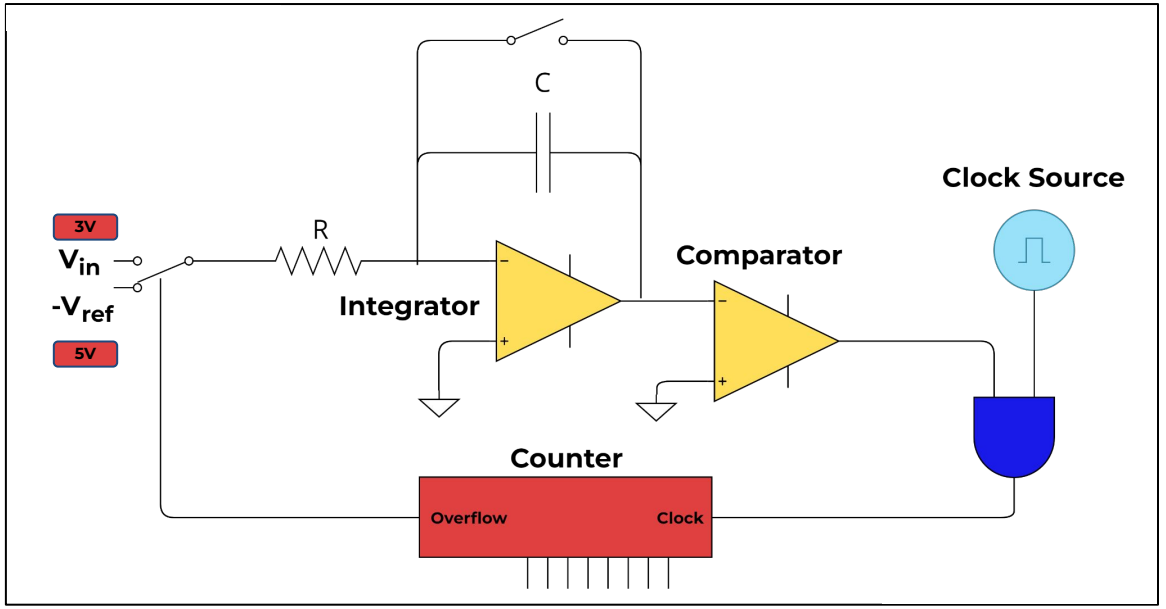




Types of ADC

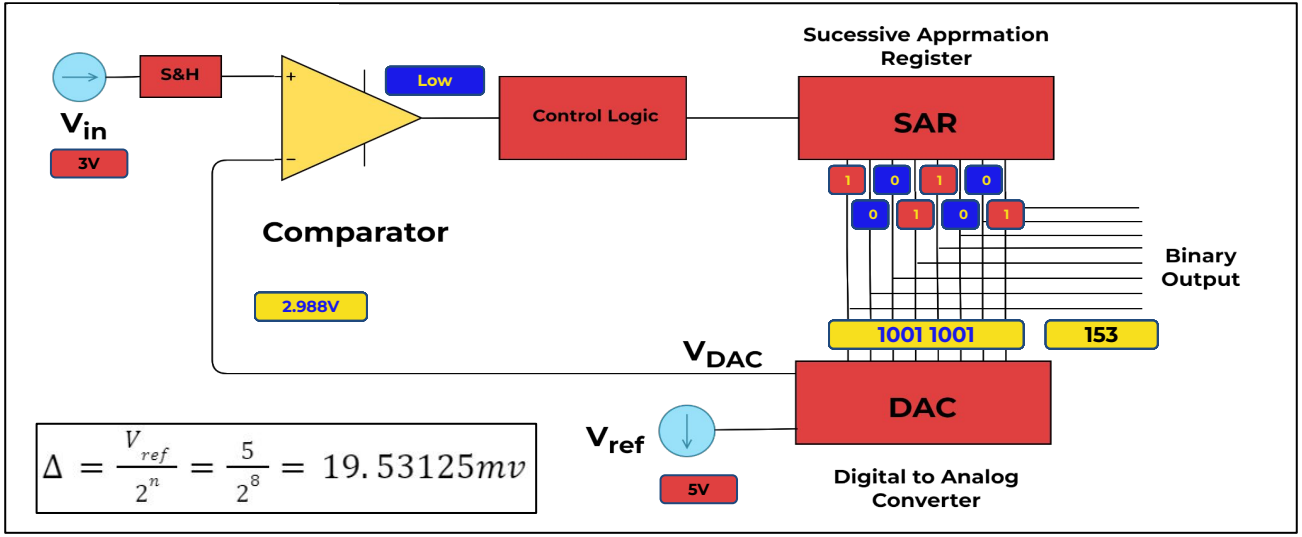
- **Dual-slope ADC**
  - Slow conversion time
  - Conversion time directly proportional with V_{in}





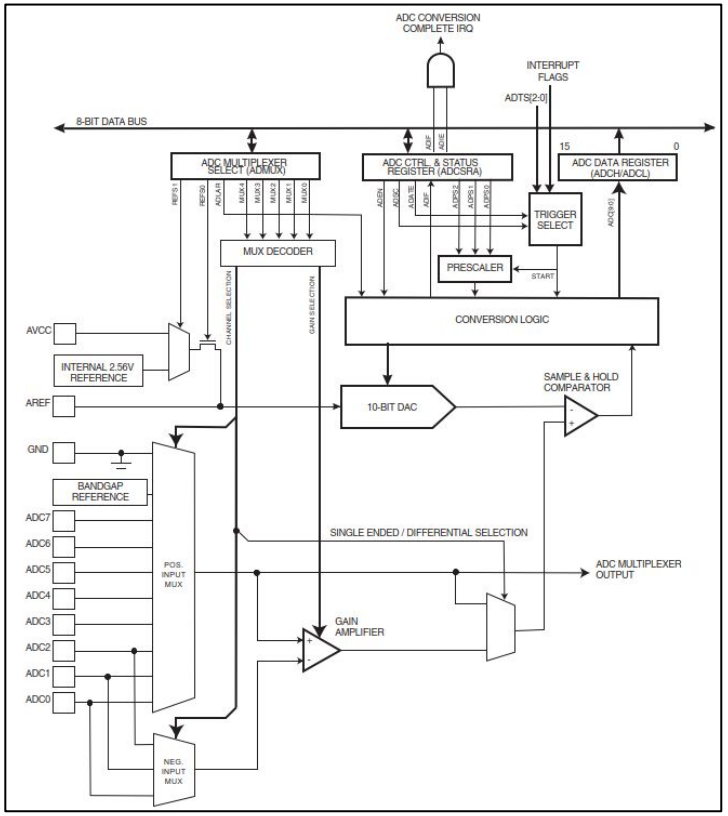
Types of ADC

- **Successive approximation ADC**



Block diagram

- Multi-channel ADC
- 10-bit ADC
- Differential inputs
- Internal 2.5 Volts



ATmega32 ADC registers

ADMUX – ADC Multiplexer Selection Register

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

ADCSRA – ADC Control and Status Register A

Bit	7	6	5	4	3	2	1	0	
	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	

SFIOR – Special FunctionIO Register

Bit	7	6	5	4	3	2	1	0	
	ADTS2	ADTS1	ADTS0	–	ACME	PUD	PSR2	PSR10	SFIOR
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	

ADCL and ADCH – The ADC Data Register

ADLAR = 0

Bit	15	14	13	12	11	10	9	8	
	–	–	–	–	–	–	ADC9	ADC8	ADCH
	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	
	ADC9	ADC8	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADCH
	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	

Steps to program ATmega32 ADC

- **Initialization**
  - Set ADC pin to be input, **DDRA** register
  - Choose ADC channel, Vref, mode, and left or right adjusted register, **ADMUX** register
  - Enable the ADC, choose prescaler, and enable or disable interrupts, **ADCSRA** register
- **Read ADC channel:**
  - Choose channel to read, **ADMUX** register
  - Start conversion, **ADCSRA** register
  - Wait for conversion to complete, **ADCSRA** register
  - Read the digital value and convert to analog, **ADCH** and **ADCL** registers

Summary

- You are now familiar with the ADC
- You must remember that the digital value may have error, it is called quantization error, that can not be recovered
- The most commonly used ACD is the successive approximation ADC
- ADCs are used to get data from the analog sensors so your code can take decisions according to these readings