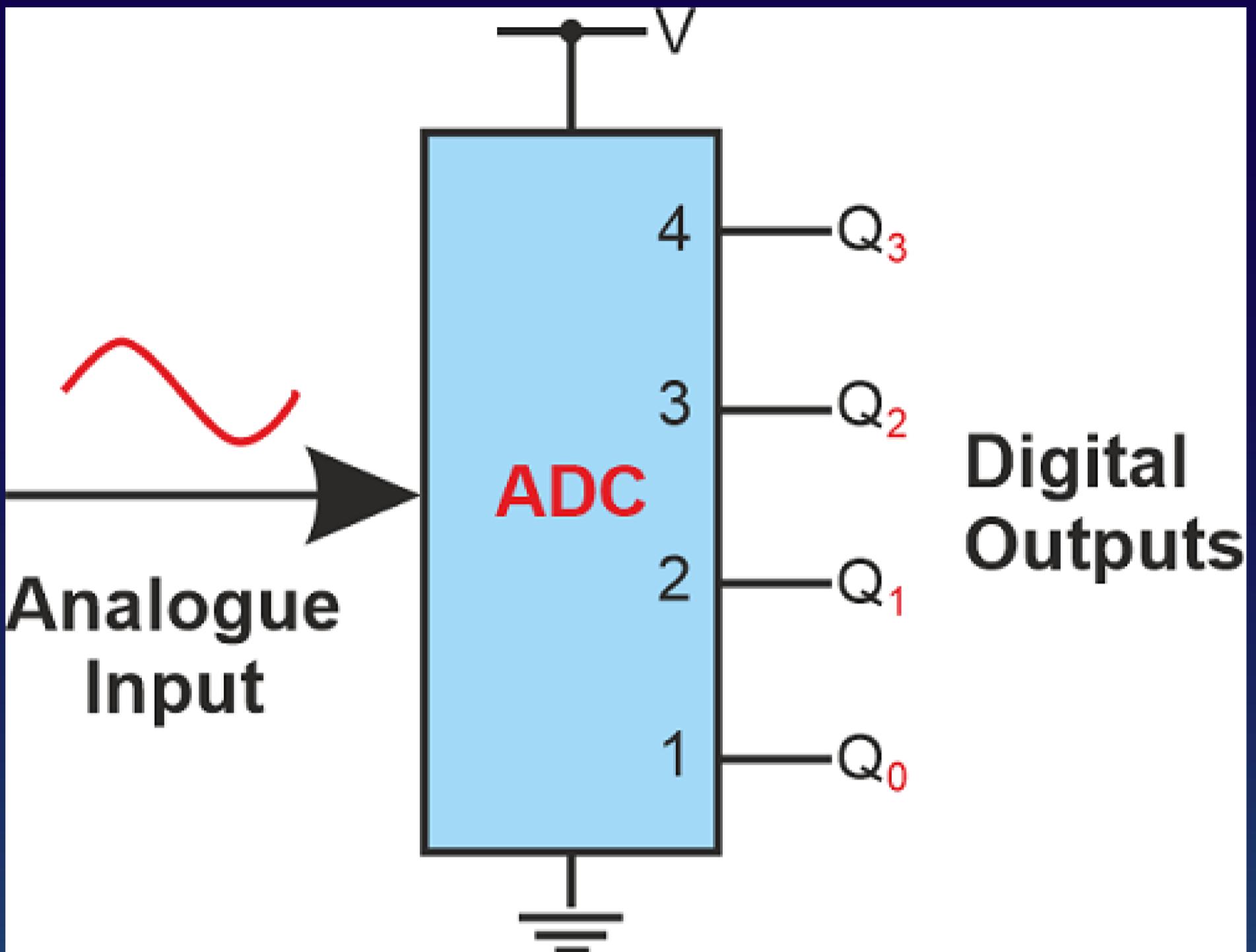


# #EmbeddedEdge



## UNDERSTANDING ADC IN MICROCONTROLLERS



nabiloulhaj@outlook.com



# #EmbeddedEdge

ADC, or **Analog-to-Digital Conversion**, is a fundamental process in microcontrollers that enables the conversion of **analog signals** into **digital values**. In the world of electronics, many real-world phenomena, such as temperature, light intensity, and sound, are analog in nature, meaning they vary continuously. However, microcontrollers operate on digital signals, which are **discrete** and can only represent a finite set of values.

The importance of ADC lies in its ability to bridge the gap between the analog and digital domains. By converting analog signals into digital values, microcontrollers can process, analyze, and make decisions based on these signals. This opens up a world of possibilities for applications in **various fields**, including industrial automation, sensor networks, medical devices, and more.



nabiloulhaj@outlook.com



# #EmbeddedEdge

**ADC resolution and sampling rate are two critical parameters that significantly impact the accuracy of analog-to-digital conversion in microcontrollers.**



nabiloulhaj@outlook.com



# #EmbeddedEdge

**Resolution** refers to the number of discrete levels or bits that an ADC can represent. It determines the smallest change in voltage that the ADC can detect and convert into a digital value. Higher resolution means more precise conversion and the ability to capture smaller variations in the analog signal. For example, an ADC with a resolution of 12 bits can represent  $2^{12}$  (4096) different voltage levels, providing finer granularity compared to an 8-bit ADC with only  $2^8$  (256) levels.



nabiloulhaj@outlook.com



# #EmbeddedEdge

The sampling rate, on the other hand, refers to the number of samples an ADC takes per second. It defines how frequently the ADC captures the analog signal and converts it into a digital value. A higher sampling rate allows for a more accurate representation of rapidly changing or time-sensitive signals. It is particularly crucial when dealing with dynamic signals, such as audio or fast-changing sensor data.



nabiloulhaj@outlook.com



# #EmbeddedEdge

ADC inputs can be classified into three main types: **single-ended**, **differential**, and **pseudo-differential**. Each type offers different advantages and is suitable for specific applications in microcontroller-based systems.



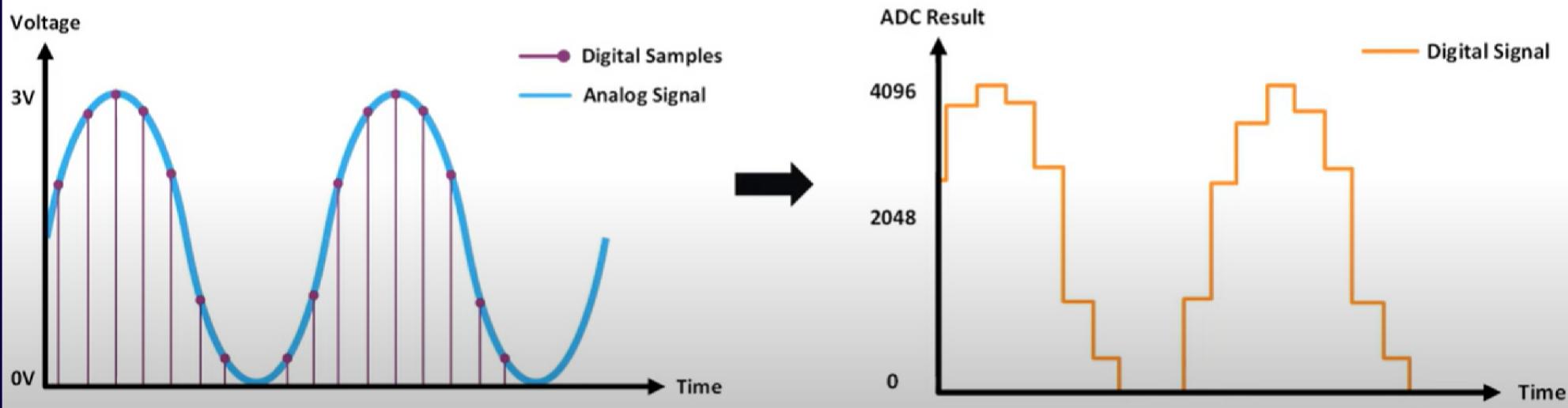
nabiloulhaj@outlook.com



# #EmbeddedEdge

## Single-Ended ADC

- 12-bit ADC with a voltage reference of 3 volts will result in the 4096 distinct values spread over 3 volts, which is about 0.73mV of precision.



Single-ended inputs are the most straightforward type of ADC input. In this configuration, the analog input signal is measured with respect to a single reference voltage. The voltage level of the signal is compared to the reference voltage, and the resulting digital value represents the magnitude of the input signal. Single-ended inputs are commonly used when measuring signals that are referenced to ground or have a known voltage level.

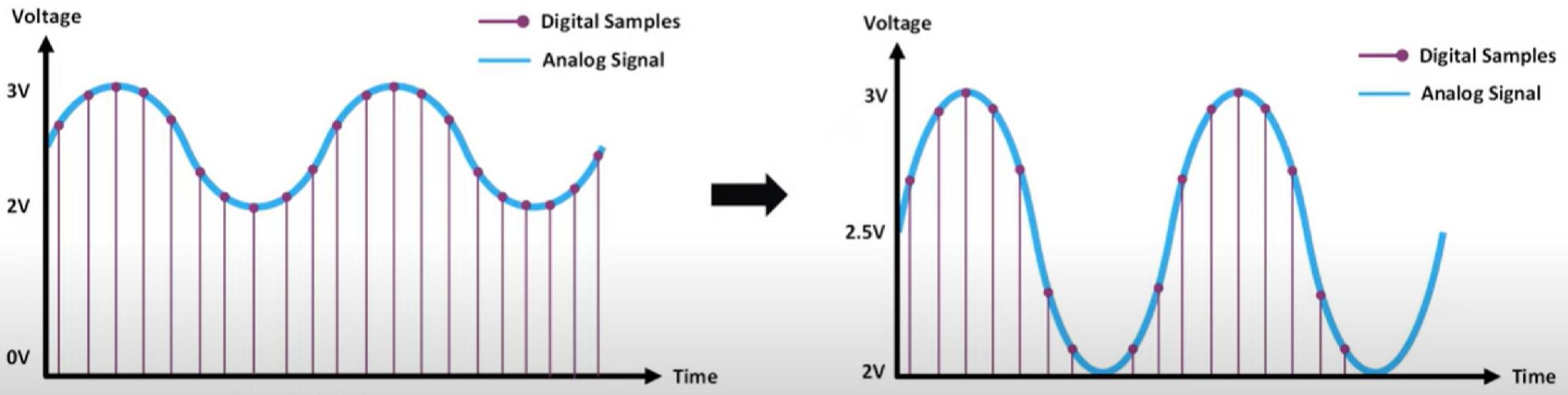


nabiloulhaj@outlook.com



# #EmbeddedEdge

## Differential ADC



**Differential inputs** involve measuring the voltage difference between two input signals. Instead of referencing the signal to a single voltage, a differential ADC compares the voltage between two input pins. This configuration provides several advantages, including improved noise immunity, common-mode rejection, and the ability to measure signals that are not referenced to the ground.



nabiloulhaj@outlook.com



# #EmbeddedEdge

**Pseudo-differential inputs** are a variation of differential inputs, where one input is a true differential voltage, and the other input is a single-ended voltage referenced to a common point, typically ground. The pseudo-differential configuration allows for improved noise rejection compared to single-ended inputs but with reduced complexity compared to a fully differential setup. It offers a balance between simplicity and performance, making it a popular choice in many microcontroller applications.



nabiloulhaj@outlook.com



# #EmbeddedEdge

**ADCs play a vital role in microcontrollers, enabling a wide range of practical applications that rely on precise measurement and conversion of analog signals into digital values. Here are some common practical applications of ADC in microcontroller-based systems:**



nabiloulhaj@outlook.com



# #EmbeddedEdge

**Sensor Measurements:** Microcontrollers often interface with various sensors, such as temperature sensors, light sensors, humidity sensors, and more. ADCs allow microcontrollers to accurately measure the analog output from these sensors, converting physical quantities into digital values for further processing and analysis. This enables applications such as environmental monitoring, industrial automation, and smart home systems.



nabiloulhaj@outlook.com



# #EmbeddedEdge

**Data Acquisition Systems:** ADCs are a key component in data acquisition systems, which are used to collect and process data from various sources in real time. These systems often involve measuring multiple analog signals simultaneously. ADCs with multiple channels enable microcontrollers to acquire data from multiple sensors or sources simultaneously, facilitating applications such as data logging, scientific instrumentation, and monitoring systems.



nabiloulhaj@outlook.com



# #EmbeddedEdge

**Motor Control:** ADCs are integral to motor control applications, where precise measurement of sensor feedback signals is necessary for accurate motor control algorithms. By converting analog sensor signals, such as position, speed, or torque, into digital values, microcontrollers can implement closed-loop control systems, ensuring precise motor control. Motor control applications are prevalent in robotics, industrial automation, and electric vehicle systems.



nabiloulhaj@outlook.com



# #EmbeddedEdge

Configuring and initializing the ADC is a crucial step in using its capabilities effectively in microcontroller-based systems. This process involves setting up the ADC parameters, such as **reference voltage**, **resolution**, **sampling rate**, **input channel selection**, and **conversion triggering**. The proper configuration ensures accurate and reliable analog-to-digital conversion. Here are the key steps involved in ADC configuration and initialization:



nabiloulhaj@outlook.com



# #EmbeddedEdge

**Reference Voltage Selection:** The reference voltage determines the voltage range against which the analog input signal is measured. Microcontrollers typically provide options for internal or external reference voltages. Selecting the appropriate reference voltage ensures that the desired input signal range is properly converted into the available digital range.



nabiloulhaj@outlook.com



# #EmbeddedEdge

## Resolution and Sampling Rate Configuration:

The ADC resolution determines the number of bits used to represent the converted analog signal. Higher resolution provides finer granularity but requires more processing power and memory. The sampling rate determines how frequently the ADC samples the analog input. Setting an appropriate resolution and sampling rate depends on the accuracy requirements and the characteristics of the input signal.



nabiloulhaj@outlook.com



# #EmbeddedEdge

**Input Channel Selection:** Microcontrollers often have multiple ADC input channels, allowing for the measurement of various analog signals. Selecting the appropriate input channel(s) enables the ADC to measure the desired signal. The channel selection process depends on the specific application requirements and the connected sensors or external devices.



nabiloulhaj@outlook.com



# #EmbeddedEdge

**Conversion Triggering:** ADC conversion can be triggered in different ways, such as software triggering or hardware triggering through events or timers. Selecting the proper triggering method ensures that the ADC samples the analog signal at the desired timing or in response to specific events. This is particularly important when precise timing or synchronization is required.



nabiloulhaj@outlook.com



# #EmbeddedEdge

**Interrupt or Polling-Based Handling:** After configuring the ADC, it is essential to decide how to handle the conversion results. This can be done through interrupt-driven or polling-based approaches. Interrupts provide a way to asynchronously handle the completion of ADC conversions, allowing the microcontroller to perform other tasks while waiting for the ADC to finish. Polling, on the other hand, involves periodically checking the conversion status in a loop. The choice depends on the system requirements, real-time constraints, and overall design considerations.



nabiloulhaj@outlook.com



# #EmbeddedEdge

In summary, ADC configuration and initialization are crucial for accurate and reliable analog-to-digital conversion. Proper setup of parameters such as reference voltage, resolution, sampling rate, input channels, and conversion triggering ensures optimal performance. By configuring the ADC correctly, microcontrollers can effectively measure sensors, process audio, monitor power, control motors, and acquire data. Careful consideration of application requirements and consulting the microcontroller's documentation are essential for successful ADC implementation.



nabiloulhaj@outlook.com