ARM Microcontroller Internship Programme

Trainer: *Kiran Jaybhave*

Sunbeam – Industrial Training Programme

Sunbeam Pune (Hinjawadi)

ADC

- An **ADC (Analog to Digital Converter)** is a circuit/module that **converts real-world analog signals into digital numbers** that a **microcontroller or processor can understand**.
- Microcontrollers work only with **digital data (0s and 1s)**, but most sensors give **analog output (continuous voltage)**.
- **Sensor Examples**

Sensor	What it measures	Output type	Why ADC is needed
LM35	Temperature	Analog voltage (10 mV/°C)	Convert voltage to temperature value
MQ135	Air quality / Gas	Analog voltage	Convert gas concentration to number
Pressure sensor	Pressure	Analog voltage	Convert pressure to digital
Microphone	Sound	Analog waveform	Convert sound to digital
DHT11	Temperature & Humidity	Digital output	ADC not required (internal ADC present)

Types of ADC

1. Flash ADC

- A **Flash ADC** is the **fastest type of analog-to-digital converter** because it converts the input signal in a **single step**.
- It uses **a large number of comparators**, where each comparator compares the input voltage with a different reference level.
- Due to the use of many comparators, **hardware complexity and cost are very high**.
- Flash ADCs are mainly used in **high-speed applications** such as **video processing, radar systems, and digital oscilloscopes**.

2. Dual Slope ADC

- A **Dual Slope ADC** provides **very high accuracy** because it integrates the input signal over a fixed period of time.
- The conversion process is **slow**, as it requires two phases: integration and de-integration.
- Due to its high accuracy and noise rejection capability, it is commonly used in **digital multimeters (DMMs)**.

3. Successive Approximation ADC (SAR ADC)

- A **Successive Approximation ADC (SAR ADC)** provides the **best balance between conversion speed and accuracy**.
- It is **widely used in microcontrollers** such as **8051, AVR, ARM, and STM32**.
- SAR ADC works on the **binary search method**, where the input voltage is compared step-by-step to determine the final digital value.
- Because of its efficiency, moderate hardware cost, and reliable performance, **SAR ADC is the most commonly used ADC in embedded systems**.

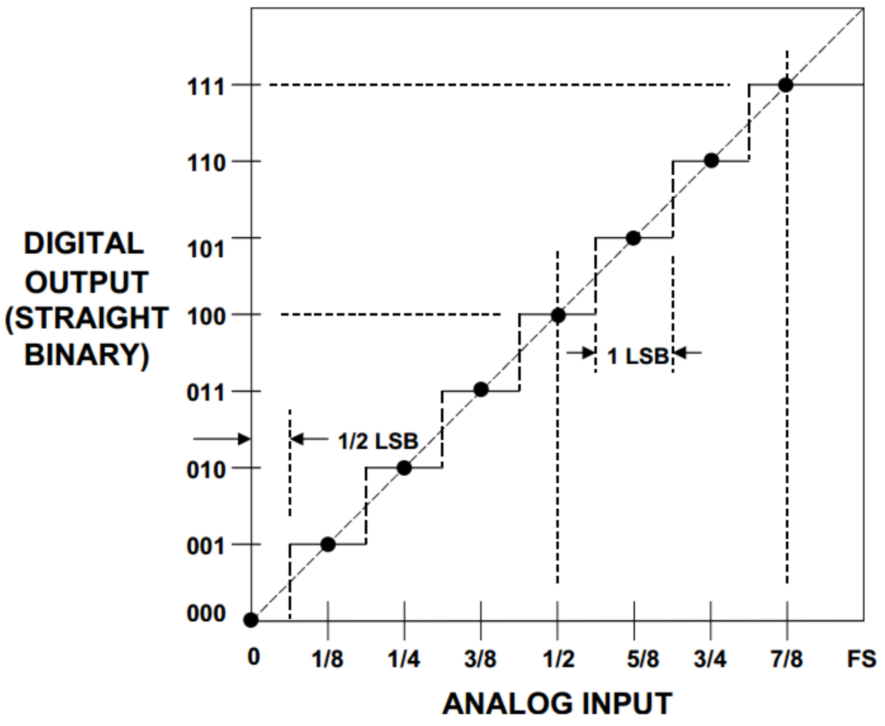
ADC characteristics

1. Resolution (ADC Resolution)

- **Resolution** of an ADC is defined as the **number of bits used to represent the analog input**.
- It decides **how many discrete digital levels (steps)** the ADC can generate between 0 V and reference voltage (**Vref**).
- Higher resolution means the ADC can detect **smaller changes in input voltage**, resulting in **better accuracy**.
- **Example: 8-bit ADC**
  - Resolution = **8 bits**
  - Number of digital levels (steps):
  - 2^8 = 256 steps
  - This means the analog input range is divided into **256 equal voltage levels**.
  - Each step represents a **small portion of the input voltage**.
- **Higher resolution → smaller step size → more accuracy**
- **Lower resolution → larger step size → less accuracy**
- ADC resolution is the number of bits used by the ADC, which determines the total number of digital levels and directly affects conversion accuracy.

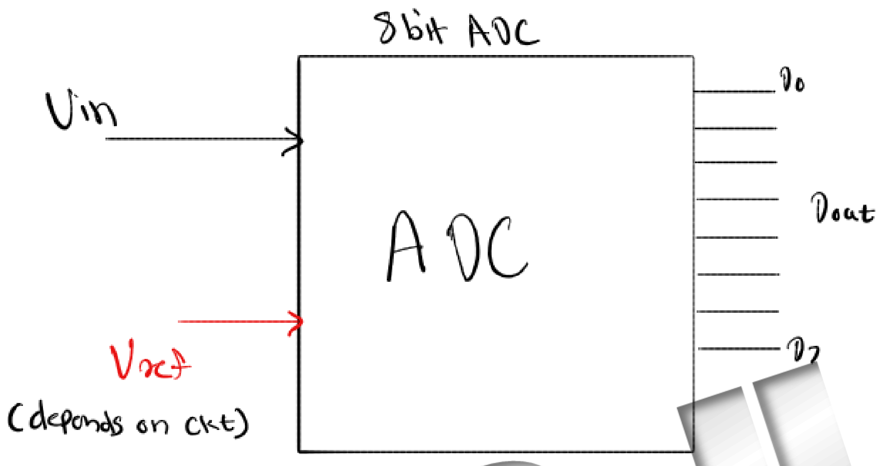
12

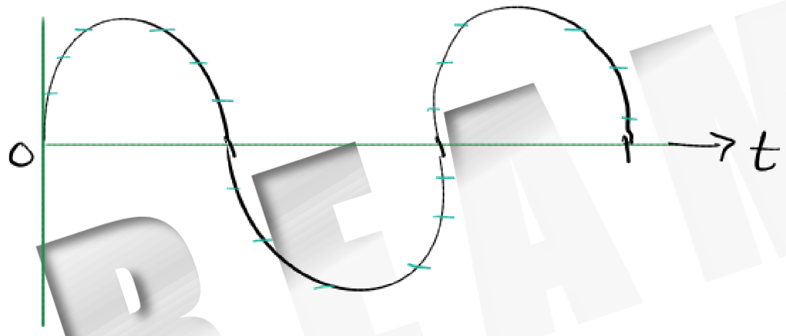
$2^{12} = 4096$


DIGITAL OUTPUT (STRAIGHT BINARY)
111
110
101
100
011
010
001
000
1/2 LSB
1 LSB
0
1/8
1/4
3/8
1/2
5/8
3/4
7/8
FS
ANALOG INPUT

2. Reference Voltage (Vref)

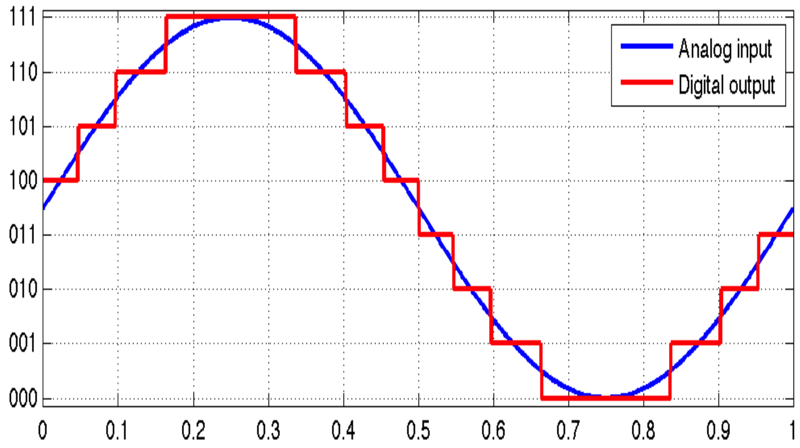
- **Reference Voltage (Vref)** is the **maximum voltage level** against which the ADC compares the input voltage (**Vin**).
- The ADC converts the analog input by **comparing Vin with Vref** and generating a corresponding digital value.
- If the ADC is designed to accept **only positive voltages**, then the **maximum allowable input voltage** is equal to **Vref**.
- **Mathematical Relation**
  - Maximum : Vin=Vref
- **Example**
  - Given:
    - Vref=2.56
    - Then the **valid input voltage range** is:
    - 0 V to 2.56
- Any input voltage **greater than Vref** will be treated as **maximum digital value (saturation)** by the ADC.
- Reference voltage (Vref) defines the maximum input voltage range of an ADC, and all analog inputs are compared with this voltage during conversion


8bit ADC
Vin
ADC
Vref
(depends on ckt)
D0
Dout
D7


0
t

3. Step Size (ADC Resolution in Voltage)

- **Step size** is the **smallest change in input voltage** that an ADC can detect and convert into a different digital value.
- It represents the **voltage width of one digital step** in the ADC output.
- Step size depends on:
  - **Reference voltage (Vref)**
  - **ADC resolution (number of steps)**
  - **Formula**
    - Step Size = Vref / Number of steps
  - **Example**
    - **Given:**
      - **Vref = 2.56** 5
      - **ADC resolution = 8-bit**
      - **Number of steps = 2^8 = 256** (0 - 255)
        - **Step Size = 2.56 / 256**
        - **=0.01 V i.e. 10 mv**
      - **ADC cannot detect voltage changes smaller than 10 mV**
      - If the input voltage changes by **less than 10 mV**, the digital output **will not change**.



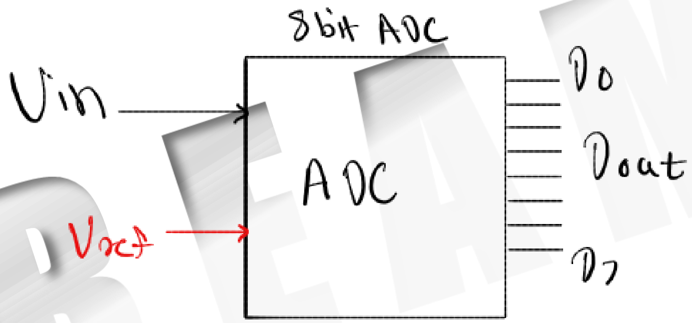
- Smaller step size means:
  - **Better accuracy**
  - **Higher resolution ADC**
- Step size is the minimum voltage difference that an ADC can resolve, calculated as the ratio of reference voltage to the total number of ADC steps.

4. Conversion Time

- **Conversion time** is the **time required by an ADC to convert an analog input voltage into its digital equivalent**.
- It depends mainly on:
  - **ADC clock frequency (Fadc)**
  - **ADC resolution (number of bits)**
- **Successive Approximation ADC (SAR ADC)**
  - In a **Successive Approximation ADC**, the conversion is performed **bit by bit**.
  - Therefore, the **number of clock cycles required** is:
    - Number of clock cycles=Resolution+1
- **Example**
  - Resolution = **8-bit**
  - Clock cycles required = **8 + 1 = 9**
    - Conversion Time = 9 / Fadc
  - **Higher ADC clock (Fadc) → Faster conversion**
  - **Higher resolution → More clock cycles → Longer conversion time**
- ADC conversion time is the time taken to convert an analog input into a digital value and, in a SAR ADC, it equals (Resolution + 1) clock cycles divided by the ADC clock frequency.

5. ADC Working Formula

- The **digital output (Dout)** of an ADC is obtained by dividing the **input voltage (Vin)** by the **step size**.
- Digital Output (Dout) = Vin / Step Size
- Since:
    - Step Size = Vref / Number of steps
- The formula can also be written as:
    - Dout=Vin / (Vref/Steps)



  - Example :
    - Vin=1V
    - Vref=2.56
    - ADC resolution = **8-bit**
    - Number of steps = 2^8 = 256

$\frac{2V}{2.56}$

$\frac{1}{(2.56/256)} \Rightarrow \frac{1}{0.01} \Rightarrow 100$

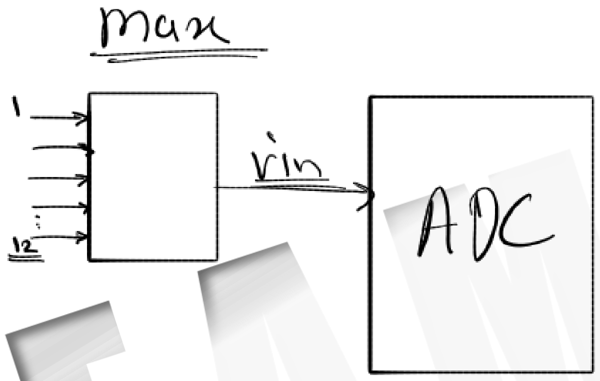
- The digital output of an ADC is calculated by dividing the input voltage by the step size, which depends on reference voltage and ADC resolution

STM32 ADC

mux

- **Overview of STM32 ADC**
  - The ADC used in **STM32 microcontrollers** is a **Successive Approximation Register (SAR) ADC**.
  - It converts **analog voltage signals** into **digital values** that the CPU can process.
  - STM32 ADCs are designed for **high speed, good accuracy, and flexibility**, making them suitable for embedded and real-time applications.
- **ADC Resolution (Configurable)**
  - STM32 ADC supports **multiple resolutions**:
    - **12-bit** , **10-bit** ,**8-bit** ,**6-bit**
  - Resolution can be selected based on:
    - Required accuracy
    - Speed of conversion
    - Application constraints
  - **Higher resolution → Better accuracy but slower conversion**
  - **Lower resolution → Faster conversion but less accuracy**

- **ADC Instances in STM32**
  - STM32 provides **multiple ADC peripherals**:
    - **ADC1** , **ADC2** ,**ADC3**
  - These ADCs can work:
    - Independently
    - In parallel (for higher sampling rate)
    - In interleaved modes (advanced applications)
      - **Interleaved mode** is an advanced ADC operation in STM32 where **two or more ADCs (ADC1, ADC2, ADC3)** work together



In STM 32 have 4 ADC Conversion Modes

- **Single Conversion Mode**
- **Continuous Conversion Mode**
- **Scan Conversion Mode**
- **Discontinuous Mode**

A. Single Conversion Mode

- Converts **one sample at a time**.
- Working sequence:
  - Start ADC
  - Wait for conversion
  - Read ADC value
  - Stop ADC
- Suitable for:
  - **Slow applications** and **Non time-critical measurements**
- ADC completion can be detected using:
  - **Polling** Or **Interrupt**
- **Example**: Periodic temperature reading

B. Continuous Conversion Mode

- ADC continuously converts input at **maximum possible speed**.
- Working sequence:
  - Start ADC once
  - ADC keeps converting in a loop
  - Stop ADC manually
- Also known as **BURST mode**.
- Interrupt can be used for conversion completion.
- Used in **real-time and fast signal monitoring**

C. Scan Conversion Mode

- Used to read **multiple ADC channels** sequentially.
- Allows sampling from:
  - Multiple sensors
  - Multiple analog inputs
- ADC scans channels in a predefined order.
- Example: Reading **temperature, gas, and light sensors together**

D. Discontinuous Mode

- Conversion is split into **small groups** instead of continuous scanning.
- Mainly used in **advanced and power-sensitive applications**.
- Detailed operation is **referenced from STM32 Reference Manual**

- **STM 32 ADC Input Voltage Range**
  - VREF − ≤ VIN ≤ VREF+
  - Input voltage must always remain **within reference limits**.
  - Exceeding this range may:
    - Saturate ADC output
    - Damage ADC input (if absolute max ratings violated)
- **ADC Conversion Rate**
  - Maximum ADC conversion rate:
    - **2.4 Mega Samples per Second (MSPS)**
  - Actual conversion speed depends on:
    - ADC clock
    - Resolution
    - Sampling time

- **STM32 ADC Clock (ADCCLK)**
  - ADC clock is derived from the system clock using a **prescaler**.
  - Operating voltage range:
    - **2.4 V to 3.6 V**
  - ADC clock frequency range:
    - **0.6 MHz to 36 MHz**
  - Higher ADC clock → Faster conversion

Thank you!

Kiran Jaybhave

email – kiran.jaybhave@sunbeaminfo.com