ADC PROTOCOL

OVERVIEW

- The ADC protocol begins when the input analog voltage is sampled by the Sample & Hold circuit.
- Next, the ADC starts the conversion sequence controlled by an internal clock.
- Each bit of the digital result is determined step-by-step using the SAR algorithm until the final 10-bit binary value is produced.
- Once the conversion is complete, the ADC makes the result available to the CPU through ADRESH/ADRESL, and the microcontroller can then perform data comparison, computation, or threshold-based decision making.

- The protocol repeats continuously as the microcontroller triggers another conversion, enabling real-time and continuous monitoring of analog signals.
- All data transfer and conversion operations are synchronized with the configured ADC conversion clock.
- After each read operation, the ADC waits for the next sampling command to restart the conversion process.
- ADC is an asynchronous protocol because the conversion process is internally timed and does not require clock synchronization between two separate devices.

INITIALIZATION STEPS

1. Select ADC Input Channel
Configure the required analog input pin (e.g., AN0, AN1) using the ADCON0 register.
2. Configure ADC Port Pins
Set corresponding TRIS registers as input (e.g., TRISA0 = 1 for AN0).
3.Configure ADCON1 / ANSEL / ANSELH registers to define which pins function as analog inputs.
4. Select ADC Conversion Clock
Choose clock source for ADC in ADCON0 / ADCON2
5. Select Voltage Reference
Choose reference voltage (Vref+ and Vref−) using ADCON1/ADCON2 if needed

6. Enable ADC Module
Set the ADON bit = 1 in ADCON0 to turn ON the ADC module.

7. Start Conversion
Set GO/DONE bit = 1 to begin ADC conversion.
8. After conversion Read Result
Read the result from ADRESH (high byte) and ADRESL (low byte) based on selected justification.

EXAMPLE

```
void ADC_Init()
{
    TRISA = 0xFF;       // Set PORTA pins as input (for analog channels)

    ADCON1 = 0x80;      // Configure result right justified, Vref = Vdd & Vss
                        // (Also sets all AN pins to analog input)

    ADCON0 = 0x41;      // ADC ON, Select channel AN0 (01000001)
                        // Fosc/8 ADC clock, ADON = 1
}
```

ADC PINS IN PIC16F877A

Analog Channel - AN0 - AN7

PIN NUM - RA0 - RA5 RE0 - RE2

FUNCTION PIN

- Vref+ (Reference Voltage +) RA3 / Pin 5
- Vref− (Reference Voltage −) RA2 / Pin 4
- ADRESH / ADRESL ADC result registers
- GO/DONE bit Starts & indicates conversion
- ADON bit Enables ADC

- All ANx pins must be configured as input (TRISx = 1).
- The analog/digital mode is selected using ADCON1 / ANSEL registers.

HARDWARE REQUIREMENTS

- Voltage Reference (internal or external Vref pin)

  For accurate ADC conversion range.

- Crystal Oscillator (e.g., 20MHz)

  Provides system clock for timing operations.

- Resistors / Capacitors

  For filtering, stabilization, and biasing.

PRACTICAL APPLICATION

1. A temperature sensor (LM35) is connected to the AN0 ADC pin of the PIC microcontroller to measure temperature as an analog voltage.

2. The PIC reads the analog value using ADC and converts it into a digital value.

3. The microcontroller compares the temperature reading with a predefined threshold value.

4. When the temperature exceeds the limit, the PIC turns ON a cooling fan or buzzer through a digital output pin.

5. The system provides automatic temperature control, widely used in industrial machines, incubators, and smart cooling systems.

THANK YOU