BN:36 *SEC:2*

2025

EMBEDDED SYSTEM

PREPARED BY

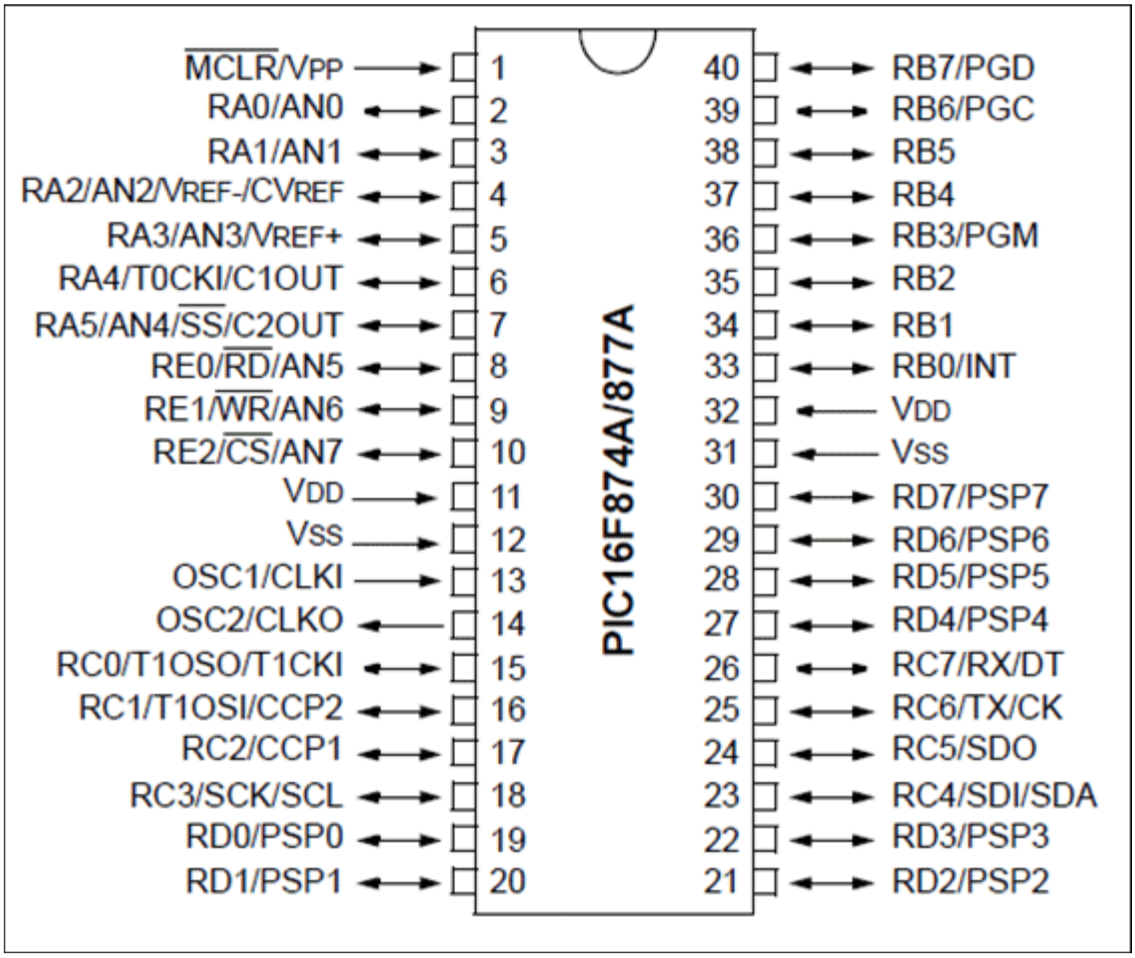
MOSTAFA WALID SAAD

1.INTRODUCTION TO EMBEDD SYSTEM

An **embedded system** is a specialized computer that combines **hardware** and (**firmware/software)** to perform specific functions within a larger system, unlike a general-purpose computer used for multiple tasks.

1.A.PIN DESCRIPTION OF PIC (16F877A)

Pin Numbers	Direction	Alternate Functions	Main Functions	Port
VDD (11, 32), VSS (12, 31), MCLR (1), OSC1 (13), OSC2 (14)	-	-	Power, Ground, Reset, Clock	**Power & Oscillator**
Pins 2–7	Bidirectional (6-bit)	AN0–AN4 (ADC)	Digital I/O	**PORTA (RA0–RA5)**
Pins 33–40	Bidirectional (8-bit)	RB0/INT (External interrupt), RB4–RB7 (Interrupt-on-change)	Digital I/O	**PORTB (RB0–RB7)**
Pins 15–18, 23–26	Bidirectional (8-bit)	RC6/TX & RC7/RX (USART), RC3/SCL & RC4/SDA (I^2C), RC3/SCK, RC4/SDI, RC5/SDO (SPI), CCP1/CCP2 (PWM/Compare)	Digital I/O	**PORTC (RC0–RC7)**
Pins 19–22, 27–30	Bidirectional (8-bit)	General-purpose digital I/O, Parallel Slave Port (PSP)	Digital I/O	**PORTD (RD0–RD7)**
Pins 8–10	Bidirectional (3-bit)	AN5–AN7 (ADC), PSP control lines	Digital I/O	**PORTE (RE0–RE2)**



1.B. CORE ARCHITECTURAL BLOCKS OF THE (PIC16F877A)

Block	Role / Function
ALU (Arithmetic Logic Unit)	• Executes arithmetic operations (ADD, SUB) • Executes logic operations (AND, OR, XOR) • Processes data in registers
Status & Control	• An 8-bit register to hold flags set by the ALU, such as Carry (C), Digit Carry (DC), and Zero (Z). These flags are very important for decision-making in code (e.g., conditional branching).
Program Counter (PC)	• Tracks current instruction address • Determines next instruction or jump (GOTO, CALL)
Flash Program Memory	• Stores the program permanently (Non-volatile) • Provides fast access for instruction execution
Instruction Register	• Holds the current instruction fetched from memory • Sends instruction to Instruction Decoder
Instruction Decoder	• Decodes the instruction • Directs MCU units (ALU, Registers, Peripherals) to execute it

1.C.TROUBLESHOOTING:LED ON RA4 NOT FLASHING

RA4 is not a standard output.

it can only sink current (LOW)→ (Open-Drain) and floats when HIGH, not providing +5V.

SOLUTION:

1.(Phisically)Connect LED anode to VCC (+5V) resistor and cathode to RA4 → (pull up Resistance)

2.(Coding)When On R4 (0)→ON || When On R4 (1)→OFF

- LED lights when RA4 is LOW (0)
- turns off when HIGH (1) ;

Characteristic	PIC16F877A	ATmega328P (AVR)
Flash Program Memory	8K words (~14 KB), limited for small to medium projects.	32 KB, much larger, suitable for complex code (e.g., Arduino IDE).
SRAM	368 Bytes only, restricts variables and buffers.	2 KB, allows larger programs and more temporary data storage.
EEPROM	256 Bytes for non-volatile storage.	1 KB, about 4× bigger for permanent storage (e.g., system settings).
I/O Pins	33 pins (5 full ports: PORTA–E).	23 pins (in DIP package), but widely extended on Arduino boards.
Operating Voltage	2.0V – 5.5V, mostly operates at 5V.	1.8V – 5.5V, more flexible for low-voltage battery systems.
Power Consumption	Moderate consumption, limited sleep modes.	Very low consumption with multiple sleep modes (Idle, Standby, Power-down).
Clock Speed	Up to 20 MHz with external crystal.	Up to 20 MHz, with an internal 8 MHz oscillator option.
Peripherals	Supports: ADC (10-bit, 8-ch), UART, SPI, I2C, Timers.	Supports: ADC (10-bit, 6-ch), UART, SPI, I2C, Timers, wider PWM support.

EXAMPLES WHERE ATMEGA328P IS A BETTER CHOICE THAN PIC16F877A:

1. **Arduino Uno projects** → faster prototyping, easy programming (IoT, robotics, sensor interfacing).
2. **Low-power battery devices** – such as wireless sensor nodes, due to **lower energy consumption**.