Logo, company name

Description automatically generated

[Document subtitle]

Ahmed Zakaria

**50v DC Voltmeter**

**عمل الطلاب: -**

**احمد زكريا عبد العال يوسف**

**احمد محمد محمود**

**اسلام محمود توفيق**

**احمد فتحي عواد**

**رمزي احمد رمزي**

**اشرف محمد إبراهيم**

**عبد الله حسن**

**مادة القياسات الكهربية**

**د: طارق الكمار**

**م:معتصم قطب**

**Equipment: -**

**1-LCD 16\*2**

**2-ressistor 2 k ohm**

**3-RESISTOR 18K OHM**

**4-wires**

**5-22PF CAPACITORS**

**6-8MHZ Crystal Oscillator**

**7-DC SOURCE**

**8-voltage regulators 7805**

**9-POT**

**10-PIC 16F877A Microcontroller**

**Circuit connection: -**

A computer screen shot of a computer

Description automatically generated

A diagram of a circuit board

Description automatically generated

**PIC** **16F877A specifications**

**Electrical Characteristics**

1. Operating Voltage Range: 4.0V to 5.5V
2. Input Low Voltage (VIL): -0.3V to 0.8V
3. Input High Voltage (VIH): 2.0V to VDD + 0.3V
4. Output Low Voltage (VOL): 0.6V maximum at 4.0mA sink current
5. Output High Voltage (VOH): VDD - 0.6V minimum at 3.2mA source current
6. Input Leakage Current (IIL): ±1 μA maximum at VDD = 5.5V
7. Input Capacitance (CIN): 10 pF maximum
8. Operating Current (I\_DD):
   * 13 mA typical at 4 MHz, 5V
   * 35 mA typical at 20 MHz, 5V
9. Idle Current (I\_IDLE):
   * 9 μA typical at 5V, typical conditions
10. Sleep Current (I\_SLEEP):
    * 1 μA typical at 5V, typical conditions
11. Brown-out Reset (BOR) Thresholds:
    * VBOR = 4.0V typical
    * VBOR = 4.3V typical
    * VBOR = 4.6V typical
12. Power-up Timer (PWRT): 72 ms typical at 5V
13. Power-on Reset (POR):
    * VDD rising: 1 ms typical at 5V
    * VDD falling: 2 ms typical at 5V

**Pinout connections**

A diagram of a circuit board

Description automatically generated

## How PIC 16F877A Works

The PIC16F877A, a 40-pin 8-bit CMOS microcontroller from Microchip, is a cornerstone of embedded systems design. Its robust hardware and flexible software enable efficient computation and interaction with the real world, making it a popular choice for diverse applications. To understand its operation, we must delve into its intricate architecture and operational cycle.

Hardware Architecture:

RISC-based CPU: Executes instructions in a single cycle of 200 nanoseconds, enabling rapid processing.

Memory Hierarchy:

8KB Flash Program Memory: Stores the program code defining the microcontroller's behavior.

368-byte Data Memory (RAM): Holds temporary data during program execution.

256-byte EEPROM Data Memory: Provides non-volatile storage for critical data.

Comprehensive I/O Ports: Offers communication channels with external devices:

Five Ports (A-E) with 40/44 pins: Each pin has specific functionalities, including digital input/output (I/O), analog-to-digital conversion (ADC) inputs, and serial communication interfaces.

Integrated Peripherals: Enhance functionality and enable diverse applications:

8-channel, 10-bit ADC: Converts analog signals from sensors or other devices into digital values for processing.

Universal Synchronous Asynchronous Receiver Transmitter (USART): Facilitates serial communication with other devices.

Capture/Compare/PWM Modules: Enable precise timing control, pulse generation, and control of actuators like motors or LEDs.

Timers and Counters: Provide timekeeping and event triggering functionalities.

Software and Programming:

Assembly Language or C Code: Defines the microcontroller's behavior and interaction with peripherals.

Integrated Development Environments (IDEs): Simplify coding, debugging, and program management.

Operational Cycle:

Power-On Reset: Initializes internal registers and fetches the first instruction from Flash memory.

Instruction Fetch-Decode-Execute Cycle:

Fetch: CPU retrieves the next instruction from Flash memory.

Decode: Instruction is decoded into control signals for various functional units.

Execute: Control signals trigger actions like arithmetic operations, data manipulation, or peripheral activation.

Data Processing and Manipulation: Data is accessed and manipulated in RAM and other memory locations.

I/O Communication: Data is transferred between the microcontroller and external devices through I/O ports and peripherals.

Peripheral Utilization: Peripherals like ADC, timers, and communication interfaces operate concurrently under program control, enhancing functionality.

Impact and Applications:

The PIC16F877A's versatility and power make it ideal for a wide range of applications:

Industrial Automation and Control: Monitoring and regulating industrial processes for efficiency and precision.

Motor Control Systems: Precisely controlling the speed and direction of motors in robotics, drones, and other devices.

Sensor Data Acquisition and Processing: Gathering and analyzing data from sensors for various applications like environmental monitoring, medical diagnostics, and robotics.

Embedded Systems Design: Integrating various components into a single, compact unit for diverse tasks.

Consumer Electronics and Appliances: Implementing intelligent features and control mechanisms in devices like TVs, washing machines, and smart home systems.

Conclusion:

The PIC16F877A is a powerful and flexible microcontroller that combines efficient hardware with versatile software. Its comprehensive architecture, diverse functionalities, and ease of programming make it a valuable tool for engineers and developers across various fields. Understanding its technical details unlocks its full potential for innovative and impactful applications.

This revised version provides a more technical and professional tone, focusing on specific details of the architecture, operational cycle, and impact of the PIC16F877A.

**Crystal Oscillator**



***What is a crystal oscillator?***

A crystal oscillator is an electronic circuit that uses a piezoelectric crystal to generate a precise, stable electrical signal with a specific frequency.

The crystal acts like a tiny tuning fork, vibrating at a very specific frequency when an electric field is applied.

This vibration is converted back into an electrical signal by the same piezoelectric effect, creating a highly accurate and stable clock signal.

**Why is a crystal oscillator important for the PIC16F877A?**

The PIC16F877A relies on a clock signal to determine its timing and execution of instructions.

A precise and stable clock signal ensures accurate timing for all internal operations, such as fetching instructions, performing calculations, and communicating with peripherals.

Crystal oscillators provide a much more stable and accurate clock signal compared to other methods like RC oscillators, making them crucial for reliable operation of the microcontroller.

**How is the crystal oscillator used in the PIC16F877A?**

The PIC16F877A typically uses a quartz crystal as the main component of its crystal oscillator circuit.

The crystal is connected to specific pins on the microcontroller that are dedicated to clock input.

Internal circuitry in the PIC16F877A amplifies the crystal's signal and uses it to drive its internal clock signal.

The specific frequency of the crystal determines the operating frequency of the microcontroller.

***Additional points to consider:***

Different PIC16F877A variants may use different crystal frequencies depending on their intended applications.

The accuracy and stability of the crystal oscillator directly impact the performance and reliability of the microcontroller.

Choosing the right crystal and ensuring proper circuit design are crucial for optimal performance.

**PCB LAYOUT: -**

**A circuit board with many points

Description automatically generated**

**Code: -**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated