A close-up of a logo

Description automatically generated with medium confidenceLogo, company name

Description automatically generated

**AIR CONDION SYSTEM**

8051’s Microcontroller

Team:

Mahmoud Morsy Abdallah Abdelrahman

Abdelrahman Mohamed AboAlfotoh

Mahmoud Atif Ahmed

Mahmoud Sherif Abdalazem

Mahmoud Hamed Alsaid

Supervisor: **DR. Mahmoud Adel**

**Project Definition:**

The Air Conditioning System with Fan Speed Control is a project aimed at providing efficient temperature regulation in a controlled environment. This project utilizes assembly language programming and the popular 8051 microcontroller along with essential components such as a temperature sensor (LM35), an analog-to-digital converter (ADC0804), and a fan. The system automatically adjusts the fan speed based on the measured temperature, providing optimal cooling performance while minimizing power consumption.

The primary objective of this project is to create a reliable and cost-effective air conditioning system using the 8051 microcontroller's capabilities. By leveraging assembly language programming, we can achieve precise control over the hardware components, ensuring efficient temperature management and energy utilization.

System Operation:

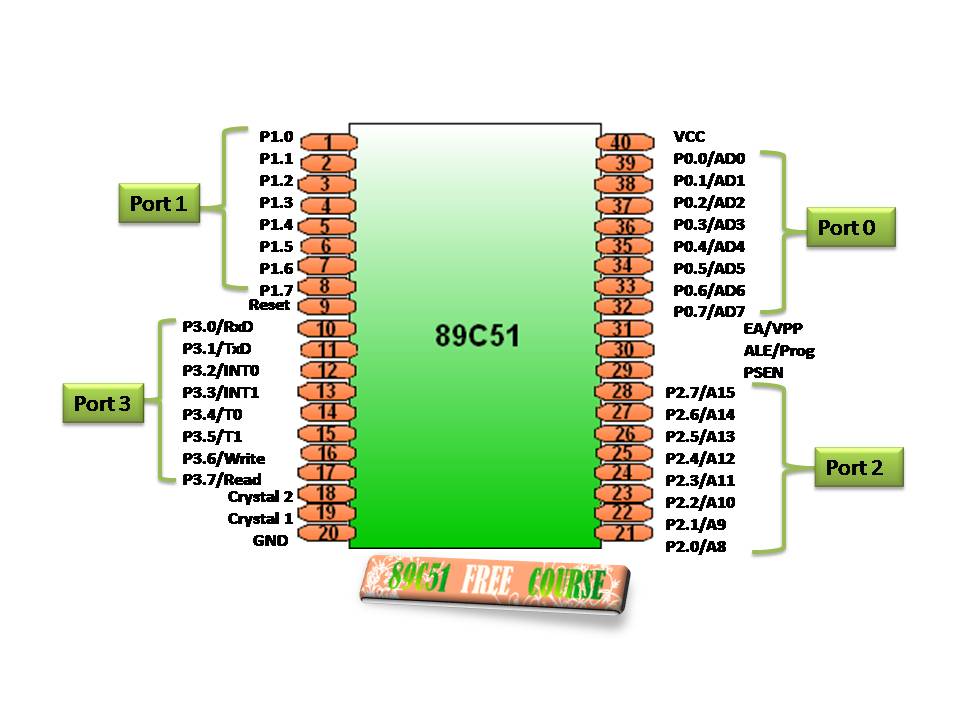
The system operates by continuously monitoring the ambient temperature using the LM35 temperature sensor. The analog voltage output from the LM35 is then fed into the ADC0804 ADC converter, which converts it into a digital value. The 8051 microcontroller reads this digital value and applies a temperature-to-fan-speed mapping algorithm to determine the appropriate fan speed level.

Based on the temperature reading, the microcontroller adjusts the fan speed accordingly. When the temperature is high, the fan operates at maximum speed to provide rapid cooling. For moderate temperatures, the fan operates at a lower speed, ensuring a comfortable environment without excessive power consumption. In cold conditions, the fan operates at its lowest speed to prevent excessive cooling.

The use of assembly language programming and the 8051 microcontroller offers precise control over the air conditioning system's functionality, enabling efficient temperature regulation while considering power conservation.

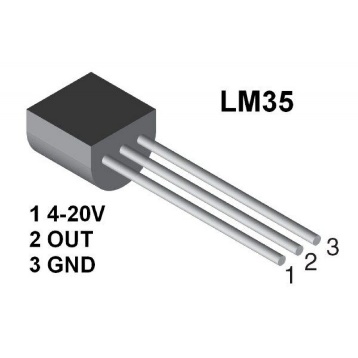
We use the Timer and Interrupt, and using them because there is no PWM we get from the microcontroller, so we get by controlling the timer to control the pulses.

**Key Components:**

1. **AT89C51 Microcontroller:** This microcontroller, like the 8051 Microcontroller, serves as the main control unit for the project. It provides the necessary processing power and I/O capabilities to interface with the various components.

2. **Crystals (12MHz):** Two crystals are used to provide the necessary clock signal for the microcontroller, ensuring accurate timing and synchronization of operations.

3. **Push Buttons:** Seven push buttons are incorporated into the interface to allow users to input the desired angle of rotation for the servo motor. These buttons enable the user to increment or decrement the angle and provide control over the rotation.

2. LM35 Temperature Sensor: A highly accurate analog temperature sensor that provides the microcontroller with real-time temperature readings.

3. ADC0804 ADC Converter: Converts the analog voltage output from the LM35 sensor into digital values that the microcontroller can process.

4. Fan: An essential component used for air circulation and cooling, controlled by the microcontroller to regulate its speed.

A diagram of a program

Description automatically generated with low confidenceFlow chart:

A picture containing text, diagram, plan, parallel

Description automatically generatedSimulations:

ORG 0000H

LJMP MAIN

; p0 for ADC

; p2 for LCD

ORG 000BH

LJMP INTERRUPT

ORG 0030H

MAIN:

MOV P0,#0FFH ;ADC Converter The Analog Signal That Comes From Tempreture Sensor to the 8051

SETB P1.5 ; The Motor

MOV A,#38H ;tells the lcd to set the cursor to the first line and the first column

ACALL COMMAND

ACALL DELAY

MOV A,#0EH ;tells the lcd to clear the screen

ACALL COMMAND

ACALL DELAY

MOV A,#01H ;tells the lcd to set the display mode

ACALL COMMAND

ACALL DELAY

MOV A,#080H ;tells the lcd to display the character at the address 080h

ACALL COMMAND

ACALL DELAY

LJMP AGAIN1

COMMAND:

MOV P2,A ;LCD

CLR P3.1 ;RS

CLR P3.0 ;RW

SETB P3.2 ;E

ACALL DELAY

CLR P3.2 ;E

RET

DELAY:

MOV R3,#0FFH

AGAIN : DJNZ R3,AGAIN

RET

AGAIN1:

MOV A,#' '

ACALL LCDWRITE

ACALL DELAY

MOV A,#'M'

ACALL LCDWRITE

ACALL DELAY

MOV A,#'I'

ACALL LCDWRITE

ACALL DELAY

MOV A,#'C'

ACALL LCDWRITE

ACALL DELAY

MOV A,#'R'

ACALL LCDWRITE

ACALL DELAY

MOV A,#'O'

ACALL LCDWRITE

ACALL DELAY

MOV A,#'C'

ACALL LCDWRITE

ACALL DELAY

MOV A,#'O'

ACALL LCDWRITE

ACALL DELAY

MOV A,#'N'

ACALL LCDWRITE

ACALL DELAY

MOV A,#'T'

ACALL LCDWRITE

ACALL DELAY

MOV A,#'R'

ACALL LCDWRITE

ACALL DELAY

MOV A,#'O'

ACALL LCDWRITE

ACALL DELAY

MOV A,#'L'

ACALL LCDWRITE

ACALL DELAY

MOV A,#'L'

ACALL LCDWRITE

ACALL DELAY

MOV A,#'E'

ACALL LCDWRITE

ACALL DELAY

MOV A,#'R'

ACALL LCDWRITE

ACALL DELAY

MOV A,#0C1H ;turn you to the next row

ACALL COMMAND

ACALL DELAY

MOV A,#'P'

ACALL LCDWRITE

ACALL DELAY

MOV A,#'R'

ACALL LCDWRITE

ACALL DELAY

MOV A,#'O'

ACALL LCDWRITE

ACALL DELAY

MOV A,#'J'

ACALL LCDWRITE

ACALL DELAY

MOV A,#'E'

ACALL LCDWRITE

ACALL DELAY

MOV A,#'C'

ACALL LCDWRITE

ACALL DELAY

MOV A,#'T'

ACALL LCDWRITE

ACALL DELAY

ACALL DELAY1

MOV A,#01H ;to clear lcd

ACALL COMMAND

ACALL DELAY

MOV A,#' '

ACALL LCDWRITE

ACALL DELAY

MOV A,#'T'

ACALL LCDWRITE

ACALL DELAY

MOV A,#'E'

ACALL LCDWRITE

ACALL DELAY

MOV A,#'M'

ACALL LCDWRITE

ACALL DELAY

MOV A,#'P'

ACALL LCDWRITE

ACALL DELAY

MOV A,#' '

ACALL LCDWRITE

ACALL DELAY

MOV A,#'C'

ACALL LCDWRITE

ACALL DELAY

MOV A,#'O'

ACALL LCDWRITE

ACALL DELAY

MOV A,#'N'

ACALL LCDWRITE

ACALL DELAY

MOV A,#'T'

ACALL LCDWRITE

ACALL DELAY

MOV A,#'R'

ACALL LCDWRITE

ACALL DELAY

MOV A,#'O'

ACALL LCDWRITE

ACALL DELAY

MOV A,#'L'

ACALL LCDWRITE

ACALL DELAY

MOV A,#' '

ACALL LCDWRITE

ACALL DELAY

MOV A,#'F'

ACALL LCDWRITE

ACALL DELAY

MOV A,#'A'

ACALL LCDWRITE

ACALL DELAY

MOV A,#'N'

ACALL LCDWRITE

ACALL DELAY

LJMP AGAIN2

LCDWRITE:

MOV P2,A

SETB P3.1

CLR P3.0

SETB P3.2

ACALL DELAY

CLR P3.2

RET

DELAY1:

MOV R3,#0FFH

HERE1: MOV R5,#0FFH

HERE2: MOV 75H,#02FH

HERE3: DJNZ 75H,HERE3

HERE4: DJNZ R5,HERE2

DJNZ R3,HERE1

RET

AGAIN2:

SETB P3.5

SETB P3.3

CLR P3.4

SETB P3.4

HERE5: JB P3.5,HERE5

CLR P3.3 ; intialization for ADC

MOV A,#0C1H ; next to the next row

ACALL COMMAND

ACALL DELAY

MOV TMOD,#02H ;timer

MOV IE,#82H ;interrupt

MOV R1,P0

MOV A,R1 ;can not turn from r1 to r4

MOV R4,A

ACALL COMPARE

MOV A,R4

LCALL CONVERSION

LCALL LCDWRITETMP

ACALL DELAY1

LJMP MAIN

COMPARE:

CLR C

CJNE R1,#35,GAIN

GAIN: JNC GAIN1

CLR C

CJNE R1,#25,GAIN2

GAIN2: JNC GAIN3

CLR TR0

LJMP GAIN4

GAIN1: ACALL GREATER

LJMP GAIN4

GAIN3: ACALL LOWER

GAIN4: CLR C

RET

GREATER:

CLR TR0

MOV R2,#0AAH

MOV TH0,#0FFH

SETB TR0

RET

LOWER:

CLR TR0

MOV R2,#0AAH

MOV TH0,#1FH

SETB TR0

RET

CONVERSION:

MOV B,#10

DIV AB

MOV R7,B

MOV B,#10

DIV AB

MOV R6,B

MOV A,R6

ADD A,#30H

MOV R6,A

MOV A,R7

ADD A,#30H

MOV R7,A

RET

LCDWRITETMP:

MOV A,R6

ACALL LCDWRITE

ACALL DELAY

MOV A,R7

ACALL LCDWRITE

ACALL DELAY

MOV A,#'C'

ACALL LCDWRITE

ACALL DELAY

RET

INTERRUPT:

CPL P1.5

CLR TR0

MOV 76H,R2

HERER: DJNZ 76H,HERER

SETB TR0

CPL P1.5

RETI

END