III Semester B. Tech - Computer and Communication Engineering

19CCE201 Microcontroller and Interfacing

Term Work

Title: Automatic temperature control in air conditioner

Prepared by

Ashwanthram A C (CB.EN.U4CCE21009)

Divyasri S J (CB.EN.U4CCE21017)

Rishikumar K (CB.EN.U4CCE21053)

Sowmya M S (CB.EN.U4CCE21069)

Department of Electronics and Communication Engineering

Amrita School of Engineering, Coimbatore - 641112

2022 – 2023 (odd)

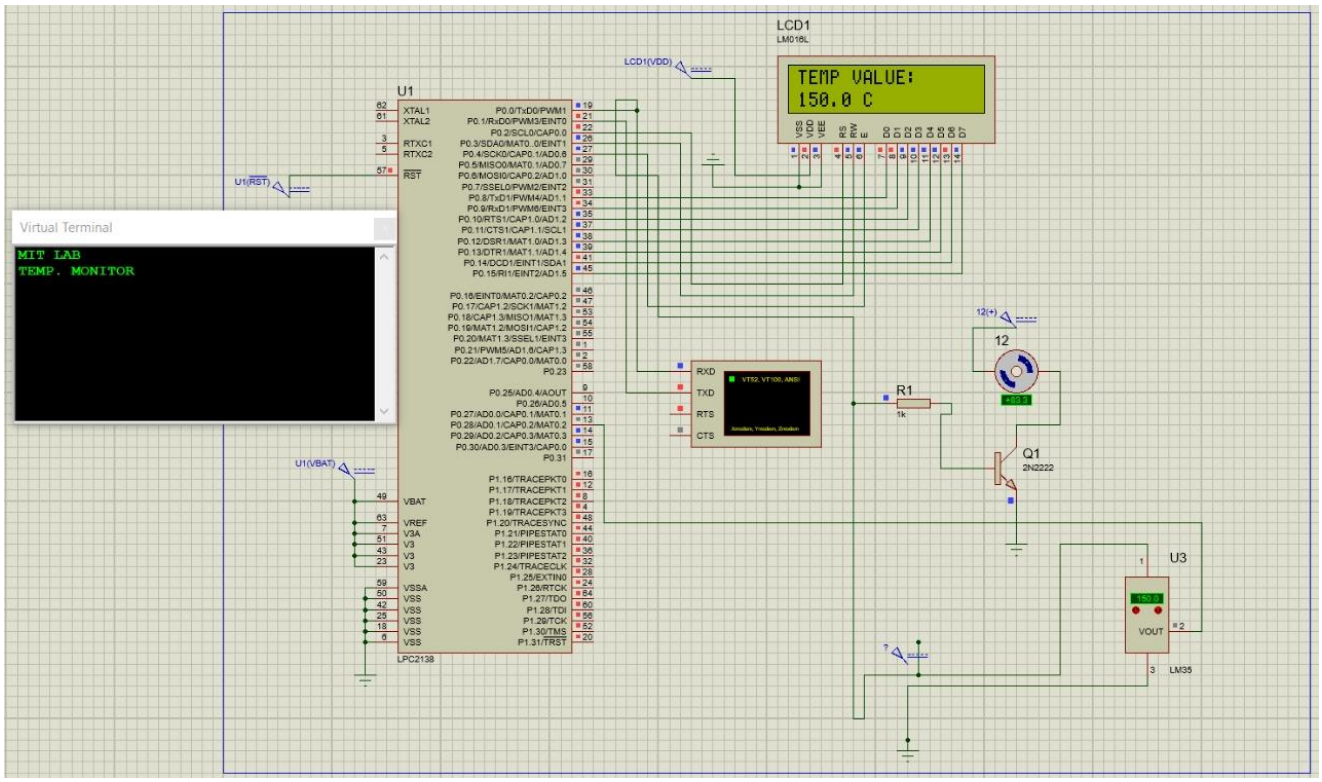
MOTIVATION:

The motivation for this project is to create a system that automatically controls the temperature of an air conditioner. This can be useful in various settings, such as homes and offices, where maintaining a comfortable temperature is important. The apparatus used to control temperature in such environments are manual in nature. A person should constantly monitor the system and manually turn on or turnoff the device. Also, there may be considerable wastage of power in manual operation of these devices. Therefore , an automatic air conditioning system is required to perform the function of both the cooling devices and heating devices.

THEORY:

We can construct automatic air conditioning system in a variety of way. We may use microprocessor/microcontroller based system or other custom designed circuits for the task. The LPC2138 microcontroller is used in this project to control the air conditioner. The microcontroller receives input from a temperature sensor, which is used to measure the current room temperature. The microcontroller then compares this temperature to a desired temperature set by the user. If the current temperature is higher than the desired temperature, the microcontroller will activate the air conditioner to cool the room. If the current temperature is lower than the desired temperature, the microcontroller will turn off the air conditioner, or increase the temperature. Basically, it consists of microcontroller as a central processor of the entire control operation. The temperature sensor gives the analog output voltage based on the temperature of the room. This analog voltage is fed to the A/D converter. The A/D converter then converts the analog input voltage from the temperature sensor into equivalent binary bits. The converted binary data from the A/D converter is applied to microcontroller. The microcontroller reads binary data from A/D converter, converts it to suitable form and performs different operations based on the value of temperature read from A/D converter.

DESIGN:



CODE:

```
#include <LPC213X.H>

/****************LCD DIRECTIVES***************/

#define LCD_CLEAR                0x01

#define CURSOR_OFF       0x0C

#define FIRST_ROW        0x80

#define SECOND_ROW       0xC0

#define Enable_Pulse()
        IOSET0|=1<<EN;Delay_ms(1);IOCLR0=1<<EN;Delay_ms(1);

/*Pin Configuration for LCD*/

#define         RS                      2

#define         RW                      3

#define         EN                      4

/***************Function Prototypes************/

void UART0_Init(void);
```

```
void UART0_Write(unsigned char value);
void UART0_Write_Text(unsigned char msg[]);
unsigned char UART0_Read(void);
void Lcd_Init(void);
void Lcd_Cmd(unsigned char value);
void Lcd_Write(unsigned char value);
void Lcd_Write_Text(unsigned char msg[]);
void Lcd_Data_Shift(unsigned char value);
void ADC0_Init(void);
unsigned int ADC0_Read(void);
void Delay_ms(unsigned long times);
unsigned long adc_data;
int main()
{
        unsigned char msg[] = "MIT LAB";
        unsigned char LM35_Temperature[] = "TEMP. MONITOR";
        unsigned char data_received[] = "TEMP VALUE:";
        unsigned char ones,tens,hundreds,thousands;
        unsigned long temp;
        Lcd_Init();
        UART0_Init();
        Delay_ms(10);
        UART0_Write_Text(msg);
        UART0_Write(10);
        UART0_Write(13);
        Lcd_Write_Text(msg);
        Lcd_Cmd(SECOND_ROW);
        Lcd_Write_Text(LM35_Temperature);
        UART0_Write_Text(LM35_Temperature);
        UART0_Write(10);
```

```
UART0_Write(13);
Delay_ms(500);
Lcd_Cmd(LCD_CLEAR);
Lcd_Write_Text(data_received);
Lcd_Cmd(SECOND_ROW);
ADC0_Init();
while(1)
{
        adc_data = ADC0_Read();
        adc_data = adc_data*3300*7.1770334928229*0.9973404255319;
        adc_data = adc_data/1023;                    //Value of Voltage in Milli Volts

        /*Display Text on LCD*/
        temp = adc_data;
        ones = temp % 10;
        temp = temp / 10;
        tens = temp % 10;
        temp = temp / 10;
        hundreds = temp % 10;
        temp = temp / 10;
        thousands = temp % 10;
        ones |= 0x30;
        tens |= 0x30;
        hundreds |= 0x30;
        thousands |= 0x30;
        Lcd_Cmd(SECOND_ROW);
        Lcd_Write(thousands);
        Lcd_Write(hundreds);
        Lcd_Write(tens);
        Lcd_Write('.');
        Lcd_Write(ones);
```

```
                Lcd_Write(' ');
                Lcd_Write('C');
                Delay_ms(10);
        }
}
/*****************Function Definition**********/

/*****************Delay Function***************/
void Delay_ms(unsigned long times)
{
        unsigned long i,j;
        for(j=0;j<times;j++)
                for(i=0;i<7500;i++);
}
/*****************LCD Functions***************/
void Lcd_Init(void)
{
        PINSEL0 = 0x00;
        IODIR0 |= (1<<RS);                      //RS Pin as Output Pin
        IODIR0 |= (1<<RW);                      //RW Pin as Output Pin
        IODIR0 |= (1<<EN);                      //EN Pin as Output Pin

        IODIR0 |= 0x0000FF00;                   //P0.8 to P0.15 as Data Line of
LCD

        Lcd_Cmd(0x38);                                  //Send 8-bit
initialization command to lcd
        Delay_ms(10);
        Lcd_Cmd(CURSOR_OFF);                    //Cursor OFF
        Delay_ms(10);
        Lcd_Cmd(LCD_CLEAR);
```

```
	Delay_ms(1);
	Lcd_Cmd(FIRST_ROW);
}
void Lcd_Data_Shift(unsigned char value)
{
	/*
	This Function will shift the eight bit data stored in variable value,
	to the Port Pin P0.8 to P0.15 Successfully.
	*/
	unsigned char i;
	for(i=0;i<8;i++)
	{
		if(value & 0x01)
		{
			IOSET0 |= (1<<(i+8));
		}
		else
		{
			IOCLR0 |= (1<<(i+8));
		}
		value = value >> 1;
	}
}
void Lcd_Cmd(unsigned char value)
{
	/*Configure LCD for receiving Command Data*/
	IOCLR0 |= (1<<RS);
	IOCLR0 |= (1<<RW);
	IOSET0 |= (1<<EN);
	Lcd_Data_Shift(value);
```

```
	Enable_Pulse();
}
void Lcd_Write(unsigned char value)
{
	/*Configure LCD for receiving Display Data*/
	IOSET0 |= (1<<RS);
	IOCLR0 |= (1<<RW);
	IOSET0 |= (1<<EN);
	Lcd_Data_Shift(value);
	Enable_Pulse();
}
void Lcd_Write_Text(unsigned char msg[])
{
	while(*msg)
	{
		Lcd_Write(*msg);
		msg++;
	}
}
/***************UART-0 Functions**************/
void UART0_Init(void)
{
	PINSEL0 = 0x00000005;				//P0.0 as TX0 and P0.1 as RX0
	U0LCR = 0x83;						//Enable access to Divisor Latches
	//and Set 8 bit Character Length with 1 Stop bit and Parity Disabled
	//Access to Divisor Latches is Enabled, in order to write Baud Rate Generator Registers

	//Values to be written in Baud Rate Registers U0DLM and U0LL
	/*
```

Formula is

Baud_Rate = PCLK*MulVal / [(16*(256*U0DLM+U0DLL)*(MulVal + DivAddVal))]

Example:-

MulVal = 1;

DivAddVal = 0;

Baud_Rate = 9600;

PCLK = 15MHz

U0DLM = 0;

Hence,

U0DLL = 15000000/(9600*16) = 97.65625 = 98

U0DLL = 98 = 0x62

*/

U0DLM = 0x00;

U0DLL = 0x62;
//Baud Rate of 9600

U0LCR = 0x03;
//Disable Access to Divisor Latches

}

void UART0_Write(unsigned char value)

{

/*

THRE bit can be extracted by this U0LSR & 0x20

THRE = 0 means data is present.

THRE = 1 means register is empty.

In order to transmit data, we have to wait will the THRE = 1,

then only we can transmit data.

*/

```
	while(!(U0LSR&0x20));                                   //THRE = 0 stay here
	U0THR = value;
}
void UART0_Write_Text(unsigned char msg[])
{
	while(*msg)
	{
		UART0_Write(*msg);
		msg++;
	}
}
unsigned char UART0_Read(void)
{
	/*
	Receiver Data Ready = U0LSR.0 bit
	RDR bit can be extracted by this U0LSR & 0x01
	RDR = 0 means no Data is Received in U0RBR
	RDR = 1 means that Data is present in U0RBR
	*/
	while(!(U0LSR & 0x01));                                 //RDR = 0 stay here
	return (U0RBR);
}
/*****************ADC Functions***************/
void ADC0_Init(void)
{
	/*************Initialize ADC AD0.0*************/
	AD0CR = 1<<21;                                          //A/D is Operational
```

```
	AD0CR = 0<<21;                                          //A/D is in
Power Down Mode

	PCONP = (PCONP &0x001817BE) | (1UL<<12);

	PINSEL0 = 0x00;

	PINSEL1 = 0x00400000;                                   //P0.27 is Configured as
Analog to Digital Converter Pin AD0.0


	AD0CR = 0x00200401;
	//CLKDIV=4,Channel-0.0 Selected,A/D is Operational

	/*

	A/D Clock = PCLK /(CLKDIV+1);

	*/

}

unsigned int ADC0_Read(void)

{

	unsigned long adc_data;

	AD0CR |= 1UL<<24;                                       //Start Conversion

	do

	{

		adc_data = AD0GDR;

	}while(!(adc_data & 0x80000000));

	//Wait untill the DONE bits Sets

	AD0CR &= ~0x01000000;                                   //Stops the A/D
Conversion


	adc_data = adc_data >> 6;

	adc_data = adc_data & 0x3FF;    //Clearing all other Bits


return (adc_data);

}

/*********************************************/
```

RESULTS:

The system was tested by setting the desired temperature to different values and observing the behavior of the air conditioner. The system was able to accurately control the temperature of the room and maintain it at the desired level.

Following are the general results of this project:

i. Temperature of the room is measured.

ii. The temperature is displayed in LCD.

iii. Temperature of the room is monitored and maintained according to the necessity.

INFERENCES:

This project has presented a means of controlling temperature of a room and maintain it within a limit. This system can be useful in various settings for maintaining a comfortable temperature and saving energy by only using the air conditioner when necessary. It is also very marketable because of its simplicity, low cost, low power consumption and small size. It can be used in various other industrial applications such as to control the temperature in boilers, refrigerators, computers and Laboratories. The automatic temperature control system using the LPC2138 microcontroller was able to successfully control the air conditioner and maintain the temperature of the room at the desired level in this project.