WEATHER ESTIMATOR
EC-316 Microprocessor Lab Project

Sajal Gupta 151/EC/13
Urmi Arora 179/EC/13

Netaji Subhas Institute Of Technology,
New Delhi, India

sajgupta94@gmail.com
urmiarora3@gmail.com
Acknowledgement

*Feeling gratitude and not expressing it is like wrapping a present and not giving it”*

-William Arthur Ward

We would sincerely like to thank Prof. Dhananjay Gadre for providing us this opportunity as a part of EC-316 coursework and for his endless support and motivation during the entire project development. We would also like to thank Gaurav Kumar, Rohan Anand, Nikita Agarwal, Nivedita Shreejeth and all the CEDT members who helped us at every single step and made the whole process move at a smooth pace. We would also like to thank many others for their invaluable support in this project. Report was completed successfully because of the grace of the god and the blessings of our parents.
Contents

I. Synopsis 4
II. Introduction 5
III. Project Description and Block Diagram 6
IV. Various Blocks of the Project 7
_ Schematic
_ Power Supply and RAM ROM 8
_ Decoding Circuit and Interfacing Circuit 9
_ Output and Input ports 11
_ Board Layout 13
_ Some pictures of the Hardware 14
V. Flowchart of the code 15
VI. Testing of the board 16
VII. Gantt Chart Revisited 17
VIII. Materials and Tools Required 18
IX. Bill of Materials 19
X. Conclusion 20
XI. Bibliography 21
Synopsis

Weather forecasting is the attempt made by the meteorologists to predict the state of the atmosphere at some future time and the weather conditions that may be expected. Weather forecasting is the single most important practical reason for the existence of meteorology as a science. It is obvious that knowing the future of the weather can be important for individuals and organizations. Accurate weather forecasts can tell a farmer the best time to plant, an airport control tower what information to send to planes that are landing and taking off, and residents of a coastal region when a hurricane might strike.

The 8085 microprocessor is used to implement a weather estimator system. It operates on a real weather data of previous year stored in its memory (ROM). The system takes input using push buttons to calculate the average temperature, humidity, rainfall & dew point. The predicted information is then displayed on the LCD display.

The data will be accepted by the 8085 microprocessor through the push buttons. A future date and month is entered in the microprocessor. The microprocessor then evaluates it with the data stored in its ROM (look-up tables) and calculates the average temperature, humidity, rainfall & dew point on that day through a weighted average. The predicted information is then shown on the LCD screen.

In case, the user enters a date that doesn’t exist, LCD displays a message “Invalid Date” and takes the user back to the date set so that the date can be modified.

Keywords: Weather Estimator, Look-Up Tables, Invalid Date, 8085 Microprocessor
Introduction

The project is based upon 8085 Microprocessor, an 8-bit microprocessor with a 40 pin dual in line package. The 8085 Microprocessor has eight registers namely- A, F, B, C, D, E, H and L and can be used in pairs or individually as per the desired word length. The processor is connected to external RAM and EEPROM and needs a 5V power supply for operation. The address and data bus are multiplexed in this processor which helps in providing more control signals. 8085 microprocessor has 1 Non-mask able interrupt and 3 maskable interrupts. It provides serial interfacing with serial input data (SID) and serial output data (SOD). The 8085 uses a total of 246 bit patterns to form its instruction set. These 246 patterns represent only 74 instructions.

Justification of the Project

Initially, we had planned to make a game, but Prof. Dhananjay Gadre encouraged us to take up a project which had some practical significance. So, we thought of designing a weather estimator which would predict the temperature, humidity, rainfall and dew point of a particular date of the current year. The algorithm used for predicting the parameters is taking an average of the parameter value on that date and nearby days’ previous year records.
Project Description

The project requires connecting 6 push buttons for taking the input from the user and an LCD to display the output. Connections for the above input and output devices are facilitated by 8255(Programmable peripheral interface). A decoding circuit is also designed to generate the necessary control signals to access memory and input-output devices as per the requirement. The entire system is powered with a help of USB Connector supplied with 5 Volts. Thus, the above components with the 8085 Microprocessor, 32k EEPROM, 32k RAM and address latch are the basic building units of our project ‘Weather Estimator’.

Block Diagram
Explanation of various blocks in the schematic

1. **Power Supply**: We are providing power to the circuit through USB. Ceramic and electrolytic capacitors are provided to filter the high frequency and low frequency noise respectively.

2. **RAM and ROM**
   
a. **RAM** or Random Access Memory is used basically to store data while running the program. We can both read and write data into it. This memory is volatile. When the power is turned off, all its contents are destroyed.

b. **ROM** or Read Only Memory is the main memory or the brain of 8085. 8085 fetches instructions saved sequentially in the ROM. Saved on the ROM is the entire program that the 8085 has to execute. We can only read from the ROM once it is connected to the board. We have used EEPROM which is Electrically Erasable Programmable Read Only Memory, since the information stored in this memory can be altered using electrical signals. Unlike RAM its contents do not get destroyed when power is turned off and to write data into it we need an EEPROM programmer.

c. **RAM** and **ROM** come in various sizes. We have chosen 32K ROM and 32K RAM. Our work could have been easily accomplished by 8K ROM and 8K RAM but to be on the safe side we took a larger memory. A larger memory leads to slight variations only in the interfacing circuitry rest of the connections remain the same.

Figure (a): RAM  
Figure (b): ROM
3. Decoding Techniques

For memory: We used 74HCT138 3x8 decoder to generate the memory read, memory write, I/O read and I/O write control signals.
4. Interfacing Circuit for Input & Output:

The 8255A is a programmable peripheral interface (PPI) device designed for use in Intel microcomputer systems. Its function is that of a general purpose I/O device connected to Interface peripheral equipment to the microcomputer system bus. The functional configuration of the 8255A is programmed by the systems software so that normally no external logic is necessary to interface peripheral devices or structures. We are using 8255 in Mode 0.

PORT CONFIGURATION:-

PORT A - Input using 6 push buttons (2 lines not connected)
PORT B - Output
PORT C UPPER - Not Connected
PORT C LOWER - Control Signals to LCD
5. **Output ports:**

We have used 8255 for interfacing LCD as the output port. LCD is connected to Port B of 8255 in Mode 0 and the control signals are being provided by Port C.

LCD: LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred over seven segments and other multi segment LEDs. The reasons being: LCDs are economical; easily programmable; have no limitation of displaying special & even custom characters (unlike in seven segments), animations and so on.

A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data.
The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD. We are using it in 8 bit mode.

6. Input ports:

The push buttons are connected in active high configuration in the circuit. So when a switch is pressed it gives a zero on its corresponding pin. We have connected these switches to the the port A of 8255 and configured port A as input. Switch usage:-

1. Date-Up: To increment date
2. Date-Down: To decrement date
3. Month-Up: To increment month
4. Month-Down: To decrement month
5. Set: To fix the date entered by the user
6. Next: To view all the parameters
7. Reset Button
Board Layout

SAJAL GUPTA (151/EC/13) URMI ARORA (178/EC/13)
Pictures of the Hardware
Testing the Board

1. Each of the individual ICs was checked by connecting a multimeter to their power pins, and it was ensured that the power signal reaches them for their proper functioning.
2. All the working LEDs and push buttons were first segregated using multimeter and were then soldered.
3. All control signals were checked on the oscilloscope.
4. Tested the SID-SOD program to check the basic working and connections between the 8085, RAM, ROM and the Address Latch.
5. Checked the functioning of 8255 on the oscilloscope.
6. Testing of the LCD was done by writing a simple code that initialized the LCD and displayed a simple message on the LCD.

Some of the major problems we faced during the course of the project are as follows:

1. While de-soldering a resistor, due to negligence, its pads got damaged and the board had to be changed.
2. Exactly 4 ROMs were changed as 2 were not working at all, while 2 were locked and had to be unlocked. But even after unlocking the ROMs didn't work. We spent a lot of time figuring out why the code didn't burn on the ROM and it turned out that the ROM was a faulty one.
3. We were storing data at a location (7000H) in ROM, however, due to some error in eeprom programmer the data was not being written into the ROM at that location and every time the message “Invalid Date” was displayed on LCD. So, we wrote the data in some other position.
4. There was a lot of havoc while converting data into hexadecimal. Since the data was huge, so we decided to use an online converter so as to reduce error. Also the code had to be altered as the data in some cases exceeded 8 bits during summation, which resulted in overflow. So, for instance, in case of humidity data for only 3 days was used for calculating the average.
Gantt Chart Revisited

Gantt Chart - Proposed

Gantt Chart

04-Jan 24-Jan 18-Feb 04-Mar 24-Mar 18-Apr 05-May

- Research on various proposals
- Seek Sr's Approval
- Prepare basic block diagram
- Research on formulas for weather...
- Prepare schematic
- Get the schematic checked
- Incorporation of changes in the schematic
- Get the schematic approved
- Testing of the schematic on bread board
- Preparing the board layout
- Get the board layout verified from seniors
- Preparation of revised board
- Send the board for fabrication
- Writing the pseudo code
- Writing the final code
- Testing the LCD code on a demo board
- Buying the components and IC
- Retrieval of the data from the Internet
- Preparation of the Project Report
- Soldering of the components
- Testing
- Debugging

Gantt Chart - Original

Gantt Chart

04-Jan 24-Jan 3-Feb 04-Mar 18-Mar 18-Apr 05-May 3-Jun

- Research on various proposals
- Seek Sr’s Approval
- Prepare basic block diagram
- Research on formulas for weather...
- Prepare schematic
- Get the schematic checked
- Incorporation of changes in the schematic
- Get the schematic approved
- Testing of the schematic on bread board
- Preparing the board layout
- Get the board layout verified from seniors
- Preparation of revised board
- Send the board for fabrication
- Writing the pseudo code
- Writing the final code
- Testing the LCD code on a demo board
- Buying the components and IC
- Retrieval of the data from the Internet
- Preparation of the Project Report
- Soldering of the components
- Testing
- Debugging
The progress of project was mostly at par with the original Gantt Chart submitted as a part of interim project report. The project was almost ready by 9th of May, 2016. However, testing and debugging along with report writing and filming the video, were delayed due to end semester examinations.

The Project was finally prepared by 1st June, 2016. Overall, we are happy and satisfied with the progress made by us during the entire process. The Original Gantt Chart proved to be a great motivation for us to keep on working with a definite target insight.

**Tools Used**

The various Tools (Software and Hardware) required during the course of project development are as below:

**SOFTWARE TOOLS:**

(i) EAGLE 7.3.0 for schematic and board layout

(ii) 8085 Simulator IDE by OshonSoft

(iii) EEPROM Programmer.jar by Anshuman Mishra, NSIT

**HARDWARE TOOLS:**

(i) EEPROM Programmer Board (equipped with Shift Registers and Arduino Nano)

(ii) Soldering Iron

(iii) Solder

(iv) Multimeter

(v) +5V DC Power Supply

(vi) Cutter, Tweezer, Hand Files, etc.
## Bill of Materials

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Device</th>
<th>Package</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>10-XX</td>
<td>B3F-10XX</td>
<td>OMRON SWITCH</td>
</tr>
<tr>
<td>1</td>
<td>58C256P</td>
<td>DIL28-6</td>
<td>MEMORY</td>
</tr>
<tr>
<td>1</td>
<td>62256P</td>
<td>DIL28-6</td>
<td>MEMORY</td>
</tr>
<tr>
<td>1</td>
<td>8085</td>
<td>DIL40</td>
<td>MICROCOMPUTER/PERIPHERAL DEVICE</td>
</tr>
<tr>
<td>1</td>
<td>8255</td>
<td>DIL40</td>
<td>MICROCOMPUTER/PERIPHERAL DEVICE</td>
</tr>
<tr>
<td>1</td>
<td>74HCT573N</td>
<td>DIL20</td>
<td>8-bit D latch BUS DRIVER</td>
</tr>
<tr>
<td>9</td>
<td>C-EU025-024X044</td>
<td>C025-024X044</td>
<td>CAPACITOR, European symbol</td>
</tr>
<tr>
<td>2</td>
<td>CPOL-EUE5-8.5</td>
<td>E5-8,5</td>
<td>POLARIZED CAPACITOR, European symbol</td>
</tr>
<tr>
<td>1</td>
<td>74HCT138N</td>
<td>DIL16</td>
<td>3-line to 8-line DECODER/DEMULTIPLEXER</td>
</tr>
<tr>
<td>1</td>
<td>16X2LCD</td>
<td>16X2_LCD</td>
<td>ALPHANUMERIC-LCD</td>
</tr>
<tr>
<td>1</td>
<td>74HCT04N</td>
<td>DIL14</td>
<td>Hex INVERTER</td>
</tr>
<tr>
<td>2</td>
<td>LED5MM</td>
<td>LED5MM</td>
<td>LED5MM</td>
</tr>
<tr>
<td>1</td>
<td>CRYSTALHC49US</td>
<td>HC49US</td>
<td>Crystals</td>
</tr>
<tr>
<td>18</td>
<td>R-US_0207/7</td>
<td>0207/7</td>
<td>RESISTOR, American symbol</td>
</tr>
<tr>
<td>4</td>
<td>M10</td>
<td>10P</td>
<td>AMP QUICK CONNECTOR</td>
</tr>
<tr>
<td>1</td>
<td>USB(POWER)</td>
<td>USB(POWER)</td>
<td>USB(POWER)</td>
</tr>
</tbody>
</table>
Conclusion

It was a great learning experience for us. Implementing a standalone project on 8085 helped us understand this microprocessor more. It introduced us to various aspects of PCB development. We also learned how hardware and software can join hands and work together to make something worthy. We enjoyed and learned a lot during the project development and recognized the importance of Time Management and Team-work for completing important tasks. Overall we are satisfied with the end result of our project and appreciate all the learning we received in the process.

Link for the project video: https://youtu.be/gdkCRgvEYYA
Bibliography

2. //www.wunderground.com/