

Design and Implementation of a Bilingual LED Message Board

A. Zidouri, M. Deriche, M. Al-Otaibi, M. Al-Shahrani, M. Al-Mutairi,
T. Al-Maleki, S. Al-Ghunaim, A. Al-Nutaifi, H Al-Issa, F. Al-Anazi

Electrical Engineering Department KFUPM, Dhahran 31261, Saudi Arabia.

Abstract:

Messages are a very important aspect of communications. It is convenient to display messages regarding meeting times, exam dates, special events, class cancellations or any pertinent information. We present here a version of a message board realized as a senior project in our Electrical Engineering Department. The simplicity, reliability, high visibility and affordable price of the device make it marketable and competitive with other types of message boards. This can display messages either in English or Arabic in different modes, still, scrolling or still and blinking.

1. Introduction

The design contains four major blocks: the computer input, the board controller, the display and the communication system. We use MATLAB codes for computer input. For the switching we use row power-switching and column selection techniques. Figure 1 shows the design of one line of characters. A normal ASCII character in this board is displayed as 35 dots using 5 columns and 7 rows for Roman alphabet. This is used widely and gives nice representation of a character. We also used the same configuration for Arabic characters. We found that the cost increases sharply if 7x9 LED matrices are chosen. We traded off the esthetic for the cost of the display. This still gives a reasonably good looking Arabic message board. Some characters and symbols may not need five columns, like the letter I. Since our message is displayed in a horizontal fashion, it makes sense to store the representation of the character in an array whose elements represent the decimal value of the column. The MSB is taken to be the lower row. As depicted in Figure 1, the last element of the array is 401. This is used to account for characters and symbols that fall short than 5 columns. Although this is not necessary in MATLAB, it might ease programming in other languages or even other programming-styles. The row power-switch circuit behaves in an opposite way: closes on high logic and opens on low logic. We then invert the column data values. Since we only need 7 bits, the most significant bit, corresponding to P1.7, is not used as a data line for the columns. It is always at value one. So the actual representation of letter "P" is [80 F6 F6 F9].

The overall results proved to be very satisfactory. The message board was implemented in two working versions: one for English messages only and one for Arabic and English. The design has been exhibited during the last 6th Saudi Engineering Conference in the KFUPM stand. The applications of such message board

are more than to be enumerated. Many companies and institutions other than our University are interested in such work. Similar versions could be designed to suit the needs of different clients. This would strengthen the relations between industry and KFUPM. The paper is organized as follows: Section 2 discusses the block diagram of the proposed system. Section 3 discusses the electronic implementation of the system. Section 4 discusses the operation of the system. Section 5 discusses the software used to drive the system. Section 6 is the conclusion.

2. The Proposed Design

We designed a version of this message board that displays only one line of Roman characters. Figure 1 shows an example of characters representation. As an effort to build a system more useful to our community we decided to extend it to a two-line display of both Arabic and Roman characters. Actually, the screen consists of number of dot matrix LED displays connected in series. The display of message, letters and numbers, will be drawn on the LEDs by making some ON and other OFF depending on the mode of operation and the user input. This display system of such type is commonly seen in shop windows, airports, and other public buildings.

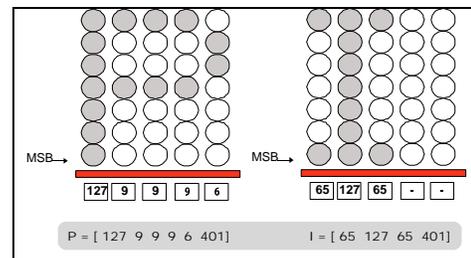


Figure 1: Examples of the P and the I Characters

2.1. Design Strategy

The proposed LED display consists of 120 columns x 7 rows of LEDs. It has in total 127 input lines to show a message. The rows inputs and the column inputs are used for different tasks from each other; either for input data or to enable a row or a column. As a result, there are two different strategies for controlling the display of message.

A. Row switching

In this strategy, the data is first loaded into the columns inputs serially with one data line. The enabling of the 7 rows of the display is carried row by row, i.e. only one row is enabled and others are OFF, then next row is enabled and others are disabled and so on it goes in unfinished loop as shown in figure 3. When this is carried fast enough, the message appears if it is ON all time as the human eyes cannot follow the enabling and disabling process.

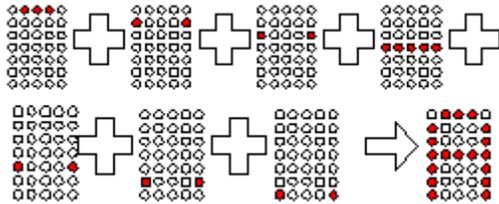


Figure 2: Row switching

B. Column Switching

In this approach, on the other hand, one column is selected to be enabled and at the same instant the data to be displayed is sent through the rows inputs and so on (for all columns) as in figure 4.

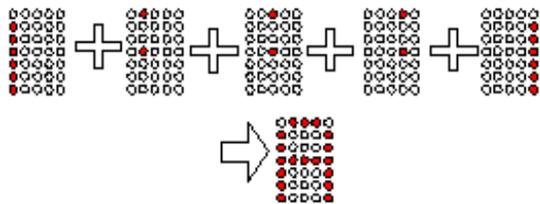


Figure 3: Column Switching.

Both of these two strategies have advantages and disadvantages. The column switching needs to be carried 120 times per cycle as there are 120 columns. On the other hand, the row switching should be repeated 7 times per cycle for the seven rows available. As a result, the column switching has to be repeated much faster than row switching for the message to appear ON all the time. This fast speed makes the display intensity decrease. However, the column switching needs much less current since the maximum number of LEDs that could be ON at same time is 7 while for row switching is 120.

Column and row switching only differ in software as the circuit is same for both. Since row switching needs about 2.4 Amp which is a high current for small ICs composing the circuit, and because the intensity of the output for column switching is reasonable, we chose to implement a column switching strategy.

2.4. Design of The Circuit

The circuit designed for the electronic message display can be divided into three main parts: the digital circuit, the logic circuit, and the output circuit, as shown in figure 4.

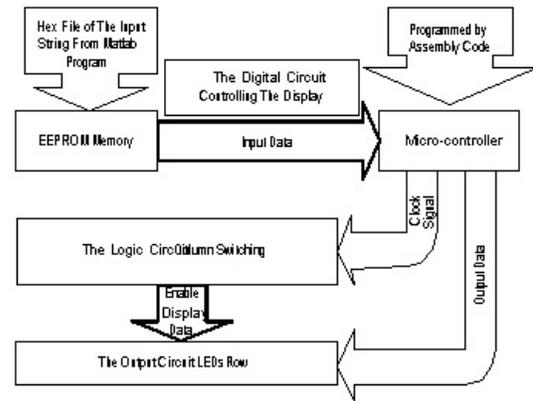


Figure 4: Proposed block diagram

The digital circuit consists of an EEPROM memory that reads the input string from a hex file represented by MATLAB code and a micro-controller that reads the input data sent from the memory and sends it to the output circuit and at same time it controls the logic circuit. The logic circuit consists of 34 ICs that carry the column switching to enable displaying data sent from the microcontroller. Finally, the output circuit consists of 24 LEDs that display the entered message. All these circuits are discussed in more details later.

3. The Electronic Circuitry

3.1 Output Display connection

The display consists of 48 5x7 LED dot matrix units. These units interlock to create a message board that has a display of two lines: one for Arabic message and the other for the English message. Each line consists of 24 units cascaded horizontally to produce a display of 120 columns and 7 rows. The design results in a 5.3 cm high by 91.2 cm long display.

To display the output we need to enable the columns and send the data through rows simultaneously, both one by one and 120 times per cycle. The micro-controller sends continuous clock signals to the logic circuit, which will be passed to enable the columns. The entered data string is saved in the external

memory in sequence as 120 data packets, with every packet containing 7 bits. At same time, the microcontroller sends clock signals, it reads a packet of data and sends it to the seven rows, a bit for every row. The 7 bits of every sent packet are displayed only on the 7 LEDs of the enabled column. In other words, only the LEDs of the enabled column have their cathode terminals on low voltage while the other columns LEDs are open circuit. If the sent bits at high level, the LED becomes closed circuit and current passes through and becomes ON. Notice that the micro-controller needs to continue being in repeated reading mode and sending clock and data signals.

3.2 Connection of Columns

There are 120 columns, hence 120 inputs are needed. These columns inputs are connected to the output of the logic circuit driven by the micro-controller. According to the adopted strategy which is column switching, all 120 should be enabled part of time in every cycle. Enabling means making the selected column at any time at low voltage (0V), which makes the cathode of all 7 LEDs at low voltage and then turning the LED ON or OFF depends on the voltage level at anode terminal

3.3 Connection of Rows

The rows of the 24 LEDs were tied together on series so that only 7 inputs are needed to control all of the rows, and every row contains 120 LEDs. These rows inputs are connected to the data output of the micro-controller. Also, 120 packets of output data are sent from micro-controller to the seven rows one by one in every cycle. To drive the LED ON, the sent data should be at high level (5V), which makes the anode at high voltage.

3.4 The Digital Circuit

This circuit is instructed by the micro-controller to carry the enabling of 120 LEDs columns. Consequently, 120 bits of signal paths are needed for every stage in this circuit. This is achieved by using a number of ICs: 15 pieces of 4015 shift registers, 18 pieces of ULN2003 Darlington sink drivers and one piece of 4009 hex buffer.

3.5 The Logic Circuit

This logic circuit is an intermediate section between the micro-controller and the output display LEDs. It is mainly composed of a shift register, NPN array and a buffer.

4. The System Operation

The EEPROM chip is programmed so that it contains the data of the string that desired to be displayed. The data of the string is saved in 120 bytes. Every byte contains 8 bits, 7 of them will be sent, to the seven rows of LEDs. All data of every byte and its address are latched internally in the EEPROM. The data with their addresses are sent to the eight latches chip. So that, the

addresses for data are reserved before they are sent to Port 0. When Port 0 of the microcontroller receives the data, it will be transferred to Port 1 as output. From Port 1, 7 bits are sent to the row as data that should be displayed at LED and the 8th bit is sent as a clock signal to logic circuit.

The LED will be ON if its anode connected to the row receives a high signal. This happens when the bit sent from the microcontroller port is at low voltage because that will make the intermediate transistor let the current flows.

P3.1 will send a high voltage signal to DATA pin of shift register in the logic circuit making the first column enabled at the same time Port 1 sending data. After that, the Port 0 will read another byte and will send data to LEDs and clock signal to shift register through Port 1. This process will be repeated 120 times until all columns data stored in EEPROM are read. Then, the loop will start again by setting P3.1 and same process done again. It can be observed that the microcontroller have to send one signal through P3.1, then send signals 120 times through Port 1. This synchronization can be implemented through programming the microcontroller and EEPROM.

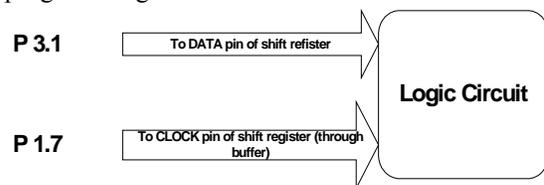


Figure 5: Ports of microcontroller controlling logic

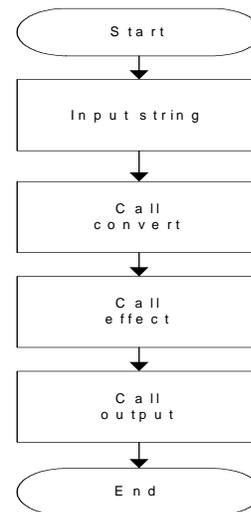


Figure 6: Flow chart for Main Program

5. Software Implementation

To implement the instructions sent by the microcontroller to the other parts of the circuit it needs programming. Programming means loading a hex file that results from simulating assembly code

written to the chip by a device called programmer. Also, EEPROM chip should be loaded with the hex file result from the MATLAB code that receives the entered string.

5.1. The MATLAB Code

To convert text message into a format that can be stored in the EEPROM, a MATLAB program is used. There are three main programs: “convert.m”, “effect.m” and “output.m” as shown in Figure 6.

The function “convert” can be used to display an Arabic sentence but the problem is with the way of mapping. The difference between converting Arabic string and English string into mapped dots to be displayed on the LED dots board; is that the Arabic letters need care in connecting Consecutive letters in every word, by knowing the letters coming before and after every letter. So, multiple cases for every letter should be defined and mapped separately.

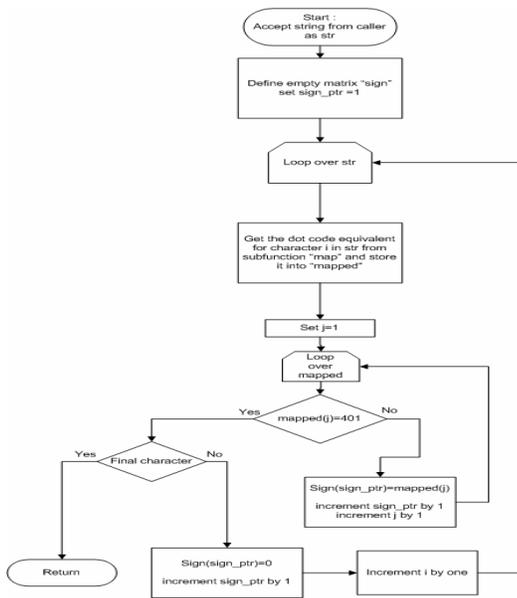


Figure 7: Flow chart of Convert Function

The programming technique should take care of the letter after the letter dealt with, unless some letters, because the beginning and middle cases have same shapes. The program will check if there is a space after the letter taking care with: if yes, it will map it with case1, and if no, it will map it with the case2.

However, there are some special cases that violate this rule:

- Letters “ح،خ،ج” when no letter comes after them, according to our mapping on 7 LEDs height, the program have to check the letter before to choose the one proper shape from two.
- Letters “غ،ع” both when no letter comes after and when it comes, the program should take care always of the letter comes before to choose the one proper shape from two in both two cases.

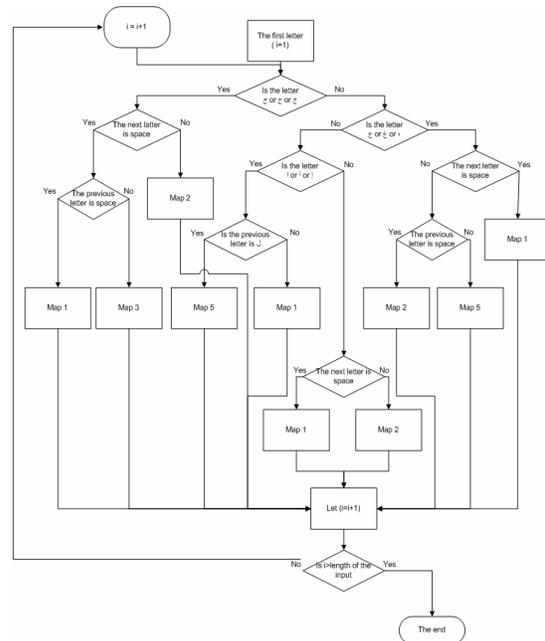


Figure 8: Flowchart of Arabic Mapping

- Letter “ح” when no letter comes after it, it has different shape and when a letter comes after it, it has two different shapes depending on the letter before.
- Letter “ل” needs checking the letter after it, if it is “ا” or not.

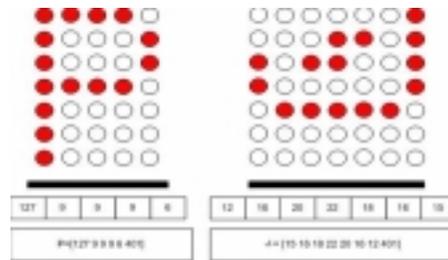


Figure 9: Examples of the P and ل Characters

6. Conclusion

In this paper, we discussed the details of designing and implementing a bi-lingual electronic board. The design is simple and can work on a stand alone basis. Future extensions will include the design of a PC interface to control the messages on the board.

References

1. M. Deriche, A. Zidouri “Electronic Message Display Board” Senior Project Report”, by S. Al-Ghunaim, A. Al-Nutaifi, H Al-Issa, F. Al-Anazi EE Dept. KFUPM. May 2002.
2. “Dual 4-Stage Static Shift Register with Serial/ Parallel output” SGS-Thomson microelectronics Data Sheet.
3. “3-State Octal D-type Transparent Latches and edge Triggered Flip Flops” Fairchild Semiconductor Data Sheet.