EE318 Electronic Design Lab Project Report, EE Dept, IIT Bombay, April 2007

Learning Aid using LED Matrix

Group D-7

Pratyush Kumar (04D07036) <pratyush@ee.iitb.ac.in>
Siva Theja M (04D07037) <sivatheja@ee.iitb.ac.in>
Albert Minj (04D07005) <albertminj@ee.iitb.ac.in>

Supervisor:
Prof. D. K. Sharma

Course Instructor : Prof. H. Narayanan

Abstract

Light Emitting Diodes (LEDs) offer the advantages of long life, low cost, effeciency, brightness and a full range of colors. Due to these properties, they are today widely used for all kinds of displays in electronic devices. A simple but non-intutive property of LEDs allows them to be used as photo sensors. In this project, we have used standard LED matrices as touch sensitive devices, in addition to being used as display devices. The main advantage of this technique is that it requires absolutely no additional hardware to add the touch sensing functionality. The application which has been implemented is that of a Learning Aid, which helps someone learn characters of various languages stored in the microprocessor. A light pen is used as stylus to write on the matrix.

1 Introduction

Light Emitting Diodes (LEDs) have become ubiquitous as simple displays in various kinds of electronic devices. They bring in the advantages of being inexpensive, bright, highly effecient, long lasting and provide the whole spectrum of output colors. Apart from emitting light, the LEDs possess another intersting property that can be used to sense the amount of light incident on the LED. The reverse lekage current in an LED is small, just like in any other common diode. Also in the presence of light, a photocurrent can be created which helpsdischarge the diode faster when reverse biased. It has been found that the time of discharge is a strong function of the amount of light incident on the LED and thus can be used to detect the brightness of light. Thus an LED can now be used to detect light in a certain mode and display information in its standard mode of operation, acting thus as an effective touch screen application.

The most important advantage of using LEDs to implement a touch screen is that there is no additional hardware that goes into adding the touch sensing functionlity to an existing LED which is used for only displaying. This has a crucial bearing on minimizing the cost of the implementation. Most of the changes need to be made in software that drives the LEDs.

The application which we have implemented to demonstrate this idea is a Learning Aid. This tool uses two 5x7 LED matrices put together (70 pixels). A set of languages are stored in the microcontroller interfacing to these matrices. Using a set of interfacing 'button' pixels a user can load up different languages and characters from the memory. The device can then animate these characters in the way they are written. The user can then trace these characters using a stylus, which is a light pen.

2 Using LEDs as light sensors

The usage of LEDs as light sensors has been known for a long time now. Infact, devices built on this idea have been demonstrated to build communication devices [1] and standard buttons [2]. Let us briefly investigate this property.

In its standard mode of operation (Fig 1(a)), an LED is forward biased, that is a current is made to flow from its anode to its cathode. This causes an increased rate of recombination in the depeletion region of the diode. The transition from the conduction band to the valence band of electrons in the depletion region, causes the emission of light of a particular wavelenght, which is seen as the light emitted.

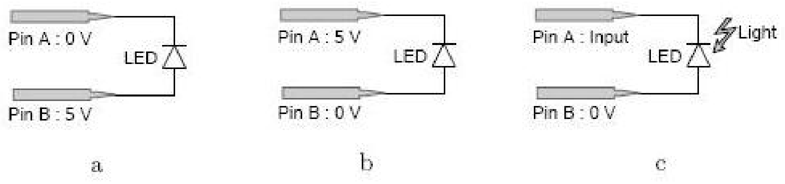


Figure 1 : The different modes of operation of the LED setup.

Now let us consider the sensing mode of operation. First we reverse bias the diode (Fig 1(b)) that is raise the cathode to high while anode is kept at a low voltage, thereby storing a charge on the diode. The reverse leakage currents are very small, thus the charge would ideally remain for a long time. Now consider the setup in Fig 1(c), where the two ends of the diode are connected to two I/O pins of a microcontroller configured in input mode with the pull up resistors on the port pins disabled. We measure the voltage across the diode. This would ideally remain the voltage we have charged the diode to. But due to non-zero leakage current, the voltage would fall slowly, exhibiting an exponential variation with time. Now assume that while in the setup of Fig 1(c), light is incident on the diode. The incidence of light would cause generation of electron hole pairs in the depletion region, just like in a photo-diode. The effeciency of generation and thus the magnitude of photocurrent caused might not be as large as in specifically engineered photo-diodes, but still would be large enough to cause a faster dishcarge of the reverse biased diode. Thus the voltage sensed by the microcontroller pins would reduce faster, but still with an exponential variation. Thus by measuring the time taken for the microcontroller pins to sense a digital zero voltage across the diode from the time it was reverse biased, one could make an estimate on the amount of light incident on the diode.

Psudeo code for an example cycle of the LEDs :
Here PORTA pins are assumed to be interfaced to the negative terminals and PORTB pins to the positive terminals of the LEDs. Also the code presented here is a simplistic case and does not appear in this form in the acutal code.

```
void main() {
    while(ON) {
        load(DATA);
        set_mode_pin_a(OUTPUT);
        set_mode_pin_b(OUTPUT);
        output_pin_a(0);
        output_pin_b(DATA);              // DATA - 1 if that LED needs to be switched on.
        output_pin_a(1);
        output_pin_b(0);                 // Reverse bias the LED.
        set_mode_pin_a(INPUT);
        set_pull_up_resistor_pin_a(DISABLE);
        int16 result_tm = 0;
        while (input_pin_a() && result_tm<ON_LIMIT)
             result_tm++;                // Wait till you sense a low on pin a.
        if (result_tm < ON_LIMIT)
             take_action();              // Action might change DATA
    }
    return;
}
```

3 The application

With the understanding of implementing a touch sensor device using a standard LED matrix, we shall now discuss the application we have designed for the hardware. The aim of the application is to be able to teach a user of how to write certain characters of certain languages which have been stored in the microcontroller interfacing the matrix. So there needs to be a bank of characters in the microcontroller. The user can select the language of her choice and withing the language the character of choice. The application should provide for an animated display of how the character should be written - pixel by pixel. Then the user should be provided with a practise mode where she can trace out the characte while it is still being displayed on the matrix. Such input needs to be validated and appropriate feedback should be given to the user. Errors made could be shown as flashing LEDs to reinforce learning. Further a test mode should be provided where the user can write out the character in a blank screen and thus check her proficiency.

4 Design Approach

From the understanding of the previous two sections, we realize that LEDs can be effectively used as bidirectional devices and effectively used in the application proposed. They can be used as input devices by sensing the light incident on them. Thus for a user to interact with the device, she needs to able to change the light incident on the LEDs. The approach used in [2] was to light certain LEDs of an LED matrix and use others in the sensing mode. So when the user takes her finger close to the device, the light emitted by the on LEDs reflect and fall on those LEDs which are in sensing mode, thereby changing the discharge times. This is effectively used as a button. But we felt that by using other LEDs to provide the light reduces the applicability of the device. Because while the matrix is being used to sense user input, the display on it is predefined and not carrying any information, that is simulatneous usage as input and output is not possible. To support the claim of a true touch screen the presence of simultaneous input and output is imperative. Thus it was decided to use external light sources, the absence or presence of which will convey user interaction. The LED matrix would toggle between 'sensing' and 'displaying' modes with all the pins being dedicated to the function it is performing at a given time, thereby decoupling the information that is displayed and the user input.

The initial choice of external light source was ambient light. So when the user would cover a certain LED with her finger, the absence of ambient light conveyed by longer discharge times would be conveyed as a user input. But this choice led to many complications. Firstly the wide variation of the ambient light conditions under which the device could be used required effective calibrations to be done by the system. Futher the detection of the exact LED from out of the entire matrix, which was covered is difficult to determine as the directional properties of the finger placing and position of light source cause different LEDs to exhibit higher discharge times. This particular point brings to light a complication associated with sensing LEDs in a LED matrix as opposed to sensing in a single LED. The approach to sensing must be competent enough in identifying the exact LED which was intended by the user. This is another major deviation from the application proposed in [1] and [2]. Another problem with covering up the LEDs to cut off ambient light, was that the discharge times of the LEDs were very large. Infact we did measure times in the range of 1s, when the LEDs were covered up. From the point of view of the application this is unacceptable, as while the LEDs are in sensing mode, the display is turned off and thus very large discharge times would translate to flickering of the display, which would now repeat after long periods of time. Thus the execution of the code would not be self-timed but would rather be dictated by the ambient light conditions. Thus the usage of ambient light was not considered.

The alternative to using ambient light, was to use a light pen, that is a stylus with a light source. So when the user places the light source on top of a certain LED, the discharge time of that LED would decrease and thus can be used to identify user input. This method overcomes all the shortcomings of ambient light. Firstly calibration need be done only once, and variation of ambient light would not affect the operation, not unless the light source on the stylus is able to decrease the discharge times sufficiently lower than the brightest ambient light conditions. Also the light source could be designed to be focussed enough to affect only a single LED and thus not cause problems in identifying the exact LED. Also since user input translates to a shorter dishcarge time, the time spent in the sensing mode can be given an upper bound making the operation faster and self-timed.

5 Choice of hardware

The number of hardware components involved in the project is few. This indeed is an important advantage of this design. The display was chosen as two standard 5x7 LED matrices put together. A major issue in selection of the matrices was whether the light would leak within the matrix thereby giving low discharge times for a set of neighbouring LEDs as opposed to a single LED. Such a problem would have required us to design a matrix of discrete LEDs with sufficient barriers between LEDs to prevent such erroneous operation. However, preliminary tests with the matrix that we obtained showed sufficient isolation between LEDs and thus were chosen for the design.

The other choice that needed to be made was with regards to the microcontroller. The requirements of a microcontroller needed for this design are: sufficiently large program memory (to store the character sets), configurable pull-up resistor on the port pins (to sense the 'analog' voltage across the LEDs in the sense mode), self-programmability (for the further scope of being able to download character sets from a computer/other unit) and sufficient on current on the port pins (to be able to drive the LED matrix brightly). These requirements were met by the microcontroller ATMega 16L from Atmel. Since there exists an internal clock source in the microcontroller which runs at upto 8 MHz, the need for an external oscillator is also removed. For the stylus which needs to be used to provide input, we chose a bright LED driven pen.

Table 1: Cost of components

ATMega 16L Microcontroller	Rs 120.00
5x7 LED Matrix - 2 units	Rs 100.00
Light Pen - Super Bright LED	Rs 5.00
Total Cost	Rs 225.00

There exist other technologies for touch sensing. But we believe that this is a more economical design implementation. Thus cost is an important factor in the selection of hardware. The of the components chosen are as given in Table 1. Thus, keeping the cost under Rs 250, a touch screen with 70 pixels has been designed.

6 Circuit Diagram

The two 5x7 LED matrices are joined together. The joined display is used as a 8x7 main display with the last row reserved for menu items. The rows (8 pins) are connected to PORTB, while the last row is connected to pin 0 of PORT C. The columns (7 pins) are connected to PORTA.

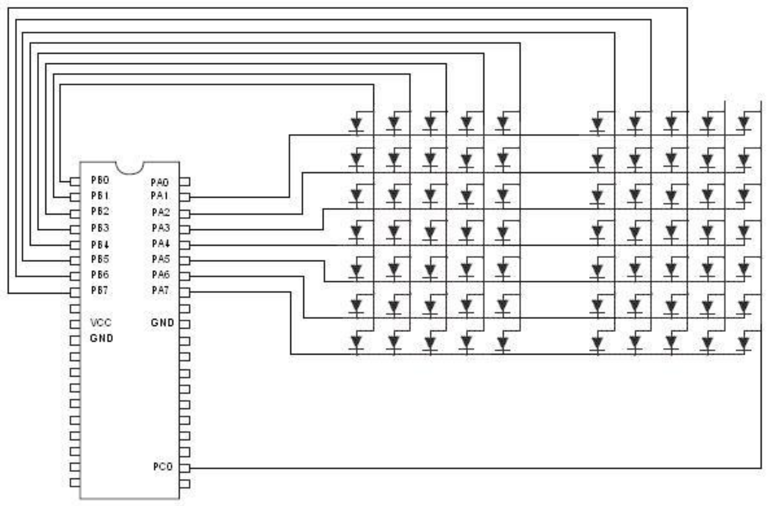


Figure 2 : The circuit diagram.

7 Software Design

Carrying forward from the previous section, the software was essentially designed to work broadly as a menu driven touch sensing display. The last row was a set of options, which triggered actions. The upper 8 rows, was the touch screen, hereby refered to as the screen section of the display, which displayed and sensed user input.

Based on the role played by the screen, the software can essentially be thought to operate in one of the following modes:

1. **Idle**
   The **Idle** mode is the default mode in which the device starts. In this mode, the screen just displays a particular character. It does not sense for any user input and does not give any feedback too.

2. **Animation**
   The **Animation** mode animates the currently active character, in the order that one is generally expected to use to write the character. It must be noted that there do not exist any standards for this, and intutive conventions have been assumed to design the charcter animations. Further the animation is only a guide, evaluation of user feedback does not take into consideration the order while writing the characters. The animation mode too does not sense user input on the screen.

3. **Practise**
   The **Practise** mode displays the actual character faintly on the screen. It senses for user input and marks the pixels so far traced by the user more brightly. Thus the practise mode can be used to trace out the character which is already lit and thus get acquainted with the character set. Evaluation mode follows this mode.

4. **Test**
   The **Test** mode is very similar to the Practise mode, but it does not display the actual character as Practise does. It starts with a blank screen, and fills up the screen with pixels traced by the user until then. Test mode is intended to be exactly just that - a test. The Evaluation mode follows this mode too.

5. **Evaluation**
   The **Evaluation** mode follows either the Practise mode or the Test mode. In either of these modes the user inputs the character. In the evaluation mode the correctness of the input is evaluated and appropriate feedback is displayed on the screen. In the case of accurate input from the user, the screen flashes a 'tick' sign. Whereas in the situation of a wrong input from the user the screen flashes the right character followed by the user input repeatedly. No user input sensing is done in this mode.

To be able to interact with the system. a menu is provided in the last row of the display. This provides many options to the user to effectively change between the various states that the screen can be in. The following are the menu buttons and the functions they perform:

1. **Language Change**
   The characters stored in the microcontroller memory can be classified into many different sets called Languages, for ease of access. The first menu button can be used to scroll through these languages. Activating this button, will load the first character of the next language in the list. When in the last language, on activating this button, the device to load back the first language. This button can only be accessed from the Idle state and returns control to the Idle state.

2. **Next Character**
   The Next Character button when activated would load the next character in the chosen language. When activated while at the last character it would scroll up and load the first character of that language. Again this button can only be accessed in the Idle state and would return control to the Idle state.

3. **Previous Character**
   The Previous Character button when activated would load the previous character in the chosen language. When activated while at the first character it would scroll up and load the last character of that language. Again this button can only be accessed in the Idle state and would return control to the Idle state.

4. **Enter**
   The Enter button has multiple functions and can be interpreted as being similar to the Return key in most systems. This button is accessible in the Test, Write and Evaluate states of the screen. If activated while in the Write or Test states, it conveys the end of the user input towarding writing of the character. This would then fire up the Evaluation state. If activated while already in the Evaluation state, it stops displaying the Evaluation of the previous character and again enters the Idle state with the current character being displayed.

5. **Replay**
   The Replay button can be used to animate the character that is currently selected. It is accessible only in the Idle state. The Replay can also be used to stop an ongoing animation. It returns control to the Idle state.

6. **Practise**
   The Practise button works parallel to the Practise state of the screen. It is only accessible in the Idle state and on being activated it starts the Practise state of the screeen for the current character.

7. **Test**
   The Test button too works parallel to the Test state of the screen. It is only accessible in the Idle state and on being activated it starts the Test state of the screen for the current character.

8 Program Flow Diagram

The above program structure has been summarised in the following diagram.

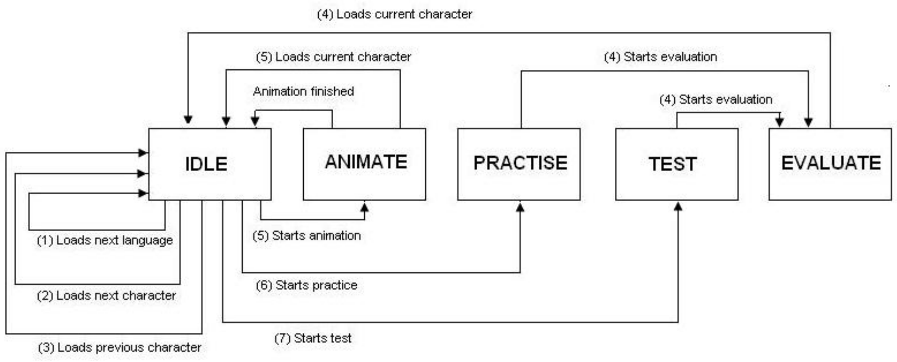


Figure 3 : The program flow diagram. The bracketed numbers indicate the activation of the menu buttons of that number.

9 Conclusions and Further Scope

A fully functional device should be capable of faithfuly working in all the states of the screen mentioned earlier, and thereby provide for a steep learning curve for the user. All the features mentioned in this report have been succesfully implemented in our design. It has thus been demonstrated that use of LEDs as touch sensors is an effective technique for low-end products, and is a cost-effective alternative to many existing techniques.

The following are some directions in which this work can be furthered:

1. Providing for using gestures for navigation, akin to mouse gestures in modern browsers.
2. Providing for a scratch pad where anything could be written, erased, copied, pasted, purged and so on, modelled close to a scratch pad or clipboard.
3. Providing for downloading of character sets from another sister device or a computer.
4. Desinging the port pin - LED interface when using a current buffer to drive the LEDs. This has been ommited in the design, and would be an important step to scale the design.

5. Investigating the use of matched resistors on the Port pins to aid faster discharges to speed up sensing in large displays.

6. Evaluating the effectiveness of this design for larger matrices, with a motivation of trying to use organic LED matrices.

10 References

[1] Paul Dietz, William Yerazunis and Darren Leigh, **Very Low-Cost Sensing and Communication Using Bidirectional LEDs**, Mitsubishi Electric Research Laboratories (MERL), TR2003-35 July 2003.
[2] Scott E. Hudson, **Using Light Emitting Diode Arrays as Input and Output Devices**, Human-Computer Interaction Institute, Carnegie Mellon University, Pittsburgh.

11 Acknowledgements

We are extremely thankful to Prof. D. K. Sharma for his invaluable technical support at each step of this project. We would also like to express our gratitude to Prof. H. Narayanan, for inspiring us and showing us the right direction at all stages. We also want to thank the TAs at the lab, who helped us with whatever doubts we had. Last but not the least, we would like to highlight the contribution of the internet as an effective resource for identifying this topic.

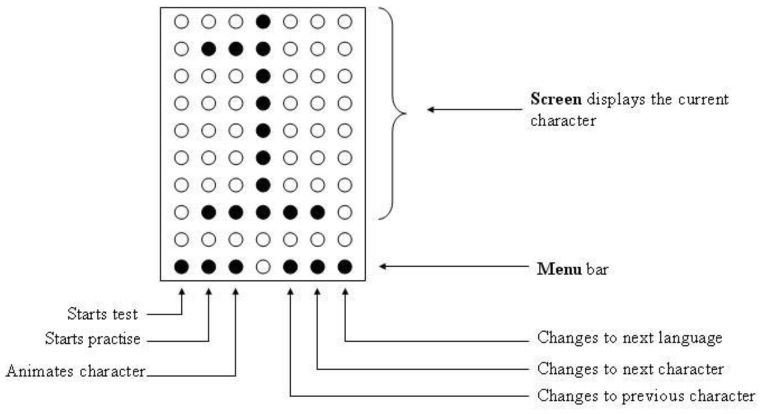
User Manual

This document contains the User Manual for Learning Aid using LED Matrix. Most details in this document are conveyed with the help of images for ease of usage.

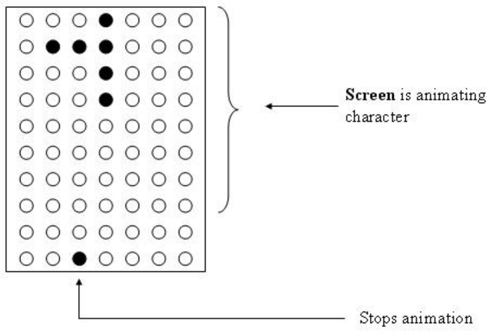
Learning Aid using LED Matrix uses a novel way of both displaying and writing data on a matrix of only LEDs. This product is meant for learning and perfecting how to write characters. The characters are stored in the device and sorted as character sets called languages. To give input to the system, one has to hold the stylus exactly over the LED desired. It is imperative that the stylus be held right on top of the LED for accurate operation. This design is fairly immune to existing ambient light situation. Though theoretically it would fail if the ambient light was greater than a threshold value. But is our belief that practically such situations will not be encountered.

In this product the first 8 rows are called the screen. It is this region where the characters would be displayed and one would write the desired characters. The last row of the display is called the menu bar. This row contains seven buttons, which can be used to communicate various actions with the system. The device works in different modes based on the action performed by the screen. Explanations of these states follows.

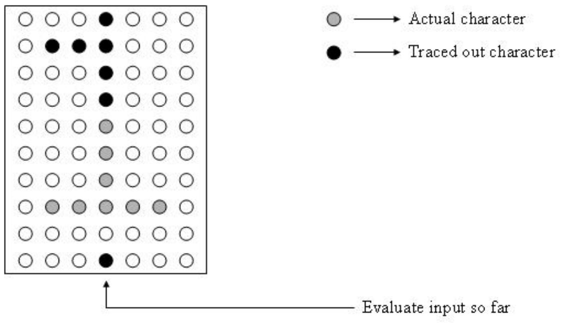
1. Idle
   The Idle state is the default state the system starts in. The figure accompanying every state contains information as to what is being displayed or sensed in the screen using an example character. Also it highlights the activated buttons and the actions which would be performed on using them.


Screen displays the current character
Menu bar
Starts test
Starts practise
Animates character
Changes to next language
Changes to next character
Changes to previous character

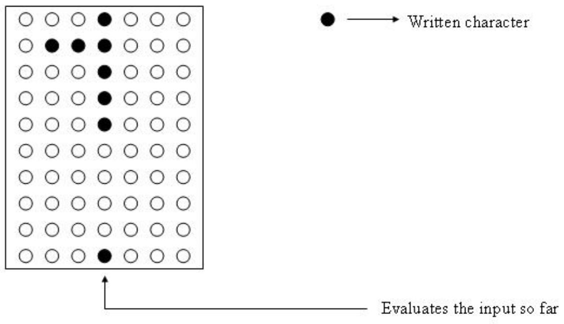
2. Animation
   The animation mode can be used to view the way in which the character may be written. There exist no exact guidelines in this regard and thus what is shown is only a guide. It is important to note that while evaluating one's input the device does not discrimate based on the order of writing the character.



3. Practise
   This state can be used to familiarize oneself with the writing of a character. The screen displays the current character faintly over which one can trace the character. Traced out pixels appear darker.

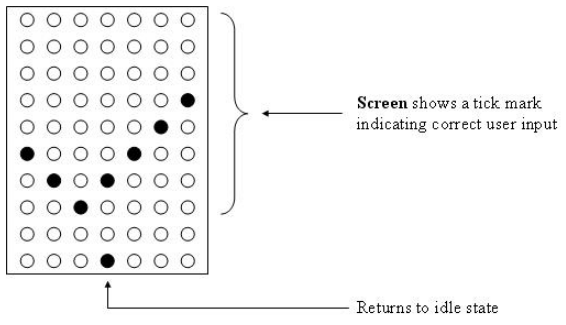


4. Test
   This mode can be used to test one's ability. One is expected to trace out the character on a blank screen and then evaluate correctness.

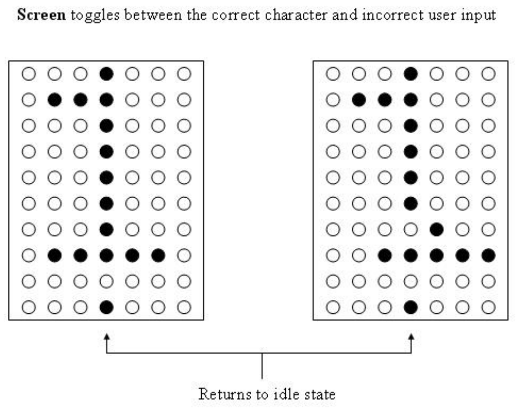


5. Evaluate - Correct Response

After the Practise/Test mode one's response would be automatically evaluated. If the input was correct a standard 'tick' mark would be displayed.



6. Evaluation - Incorrect Response
   In the case of an incorrect response, the screen would toggle between the correct character and the incorrect input from the user.


Screen toggles between the correct character and incorrect user input
Returns to idle state

We hope that your learning experience is a pleasant one.

Let there be light.