International Journal of Basic Science and Technology
June, Volume 11, Number 2, Pages 159 - 169 http://www.ijbst.fuotuoke.edu.ng /
https://10.5281/zenodo.15601071 ISSN 2488-8648

IoT-Enabled Electronic Display System Using LED Matrix Technology

Ifeagwu E.N.
Department of Electrical and Electronic Engineering,
Federal Universitv Otuoke. Bavelsa State.

Article Information
Article # 100276
Received date:8th Jan. 2025
Revision:28th March, 2025
Acceptance:4th June, 2025
Published: 29th June, 2025

Key Words
AT89C51 microcontroller
ESP 32
LED
Wi-fi

Abstract

In the era of smart systems and connected devices, real-time information dissemination is essential across sectors such as public transportation, education, retail, and healthcare. This paper presents the design and implementation of an IoT-enabled electronic display system utilizing LED dot matrix array technology. The system allows remote control and dynamic content updates via an intuitive web or mobile interface. Leveraging Wi-Fi-enabled microcontroller like the ESP32,which interconnects with the AT89C51, the display retrieves data from a cloud server or user inputs and presents messages on a scrolling LED matrix. Key features of the system include low power consumption, scalability, and ease of deployment. This solution provides a more efficient, automated, and cost-effective alternative to traditional manually updated signage, suitable for both indoor and outdoor applications

Corresponding Author: Ifeagwu E.N: ifeagwuen@fuotuoke.edu.ng

Introduction

In the digital age, effective communication plays a pivotal role in the smooth operation of educational institutions. Tertiary institutions such as universities, colleges, and polytechnics are complex environments where the timely dissemination of information is critical. Information such as lecture schedules, examination timetables, announcements, event notifications, emergency alerts, and administrative instructions must reach students and staff promptly and efficiently. Traditional modes of communication such as bulletin boards, printed notices, and word of mouth often fall short in terms of timeliness, accessibility, and visual appeal. It is often inefficient, outdated, and environmentally unfriendly (Jones, 2020). These communication gaps can lead to missed information, confusion, low event participation, and delayed responses in emergencies. An innovative and increasingly popular solution to this challenge is the use of electronic LED matrix display systems. These systems utilize arrays of light-emitting diodes (LEDs) to display dynamic textual or graphical content(Yanchuang & Jinying, 2018). The messages can be easily updated in real-time via a centralized control system, reducing the need for printed materials and allowing for instant information dissemination across campus locations. LED matrix displays can be placed in strategic areas like lecture halls, administrative buildings, cafeterias, and entrances to keep students and faculty well-informed.

The benefits of LED matrix displays extend beyond communication efficiency. They contribute to the institution's modern image, reduce operational costs associated with paper-based communications, and support eco-friendly initiatives by minimizing paper waste (Kumar and Sharma, 2020). The flexibility of these systems allows for customized messages, multi-lingual support, and even integration with emergency response systems. Technological advancements have also made LED matrix displays more energy-efficient and cost-effective, further enhancing their appeal for institutional use. Despite their potential, the adoption of LED matrix display systems in many tertiary institutions particularly in developing countries remains limited (Edward, 2015). Factors such as cost, lack of technical expertise, and inadequate infrastructure contribute to the slow implementation rate. However, with the growing reliance on digital technologies in education and the increasing need for real-time communication, the importance of such systems is more pronounced than ever(Foram, 2020).

This paper seeks the implementation of an affordable, scalable, and user-friendly electronic LED matrix display system tailored for tertiary institutions. The system will allow administrators to broadcast vital information swiftly and effectively across various locations within the campus. Using microcontroller-based architecture (such as Arduino or Raspberry Pi) and an intuitive software interface, the system will demonstrate how low-cost digital technology can enhance communication in educational settings. By implementing a prototype of the display system, this project aims to provide a blueprint that institutions can adopt and scale according to their needs and budgets. The system will not only improve the dissemination of information but also serve as a platform for future enhancements, such as integration with mobile apps, web portals, or biometric systems for personalized

International Journal of Basic Science and Technology
June, Volume 11, Number 2, Pages 159 - 169 http://www.ijbst.fuotuoke.edu.ng /
https://10.5281/zenodo.15601071 ISSN 2488-8648

notifications. Specific objectives include: developing a cost-effective LED matrix display prototype using microcontroller technology, creating a centralized control interface that allows administrators to update messages in real-time deploying the system in a simulated campus environment to test its effectiveness in broadcasting academic and administrative information and evaluating the performance of the system in terms of visibility, reliability, and user satisfaction.

ESP32 (Wi-Fi Module)

The ESP32 Wi-Fi Module links the microcontroller to the Wi-Fi network. The ESP32 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor. The ESP32 module is an extremely cost-effective board. This module has a powerful enough onboard processing and storage capability that allows it to be integrated with the sensors and other application-specific application devices through its GPIOs with minimal development up-front and minimal loading during runtime. Its high degree of on-chip integration allows for minimal external circuitry, including the front-end module, which is designed to occupy minimal PCB area (Floyd, 2018).

The AT89C51 Microcontrollers

The 8051 microcontroller is a small microcomputer and became very popular after Intel licensed other manufacturers like Atmel, Phillips, AMD, Infineon (formerly Siemens), Mantraz, Maxim, Nuvoton etc. to produce varying but code-compatible versions of the microcontroller (Holliday, 2010). Features of the AT89C51 Microcontroller (Wakerly, 2011), 4K Bytes of In-System Reprogrammable Flash Memory, Endurance: 1,000 Write/Erase Cycles
Fully Static Operation: 0 Hz to 24 MHz, Three-level Program Memory Lock, 128 x 8-bit Internal RAM, 32 Programmable I/O Lines, Two 16-bit Timer/Counters, Eight Interrupt Sources, Programmable Serial Channel Low-power Idle and Power-down Modes

Figure 1 shows the internal architecture of a typical AT89C51 microcontroller.

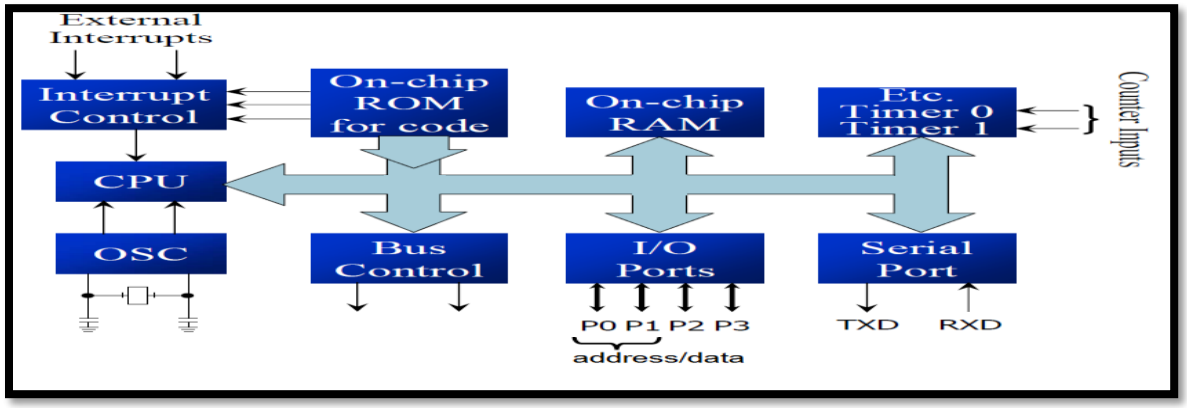

Fig. 1: The internal architecture of the AT89C51 microcontroller (Nikhil, 2021)

The CPU: The Central Processing Unit (CPU) of the microcontroller comprises the arithmetic and logic unit (ALU) which performs arithmetic and logic operations on the data fetched from the data memory. It has an instruction decoder that decodes machine code instruction words. It also has registers that store operand addresses/instructions and the control unit (Katz, 2009).

The Interrupt Control: Table 1 (a) shows the summary of the AT89C51 Microcontroller Pin Functions. The AT89C51 microcontroller has two external interrupt control pins INT0 and INT1. When the external interrupt mode is selected, hardware components connected to these pins can be serviced when they send an interrupt request IRQ by making either INT0 or INT1 active (low). When this occurs, the normal execution of programs stops while the microcontroller services the hardware (runs a special subroutine known as the interrupt service routine ISR for the interrupting device) and when the hardware has been serviced, the program execution continues from where it stopped before the interrupt. This sequence of actions is handled by the interrupt controller (Muhammad, 2018).

Bus Control: This controls the flow of instructions as well as data within the microcontroller. It prevents bus contention.

The Timer Counters: Timer 0 and Timer 1 provide time measurement and count events. It is also employed in generating clock pulses (baud rates) when the microcontroller is used in serial communication.The AT89C51 has two 16-bit

International Journal of Basic Science and Technology
June, Volume 11, Number 2, Pages 159 - 169 http://www.ijbst.fuotuoke.edu.ng /
https://10.5281/zenodo.15601071 ISSN 2488-8648

timers/counters, which make it easier to generate periodic signals or count signal transitions.

The I/O Ports: The AT89C51 microcontroller has I/O ports used for interfacing the microcontroller to external devices like keypads, relays, sensors, ADC's etc [Nikhil .M.]. Apart from the use of I/O ports, three out of the four ports have alternate functions. Port 0 and port2 are used when interfacing with external memory and all the pins in port 3 have alternate functions as illustrated in Table 1(b).

Table 1(a):Summary of the AT89C51 Microcontroller Pin Functions (Predko, 2017)

Pin	Symbol	Input/Output	AT89C51 Function
1	(T2)P1.0	I/O	Port 1, bit 0, timer 2 external input
2	T2(EX)P1.1	I/O	Port 1, bit 1 timer 2 external reload/capture
3	P1.2	I/O	Port 1, bit 2
4	P1.3	I/O	Port 1, bit 3
5	P1.4	I/O	Port 1, bit 4
6	P1.5	I/O	Port 1, bi 5
7	P1.6	I/O	Port 1, bit 6
8	P1.7	I/O	Port 1, bit 7
9	Reset	Input	Reset system
10	(RXD) P3.0	I/O	Port 3, bit 0 serial receive
11	(TXD)P3.1	I/O	Port 3, bit 1; serial transmit
12	(INTO)P3.2	I/O	Port 3, bit 2;external interrupt 0
13	(INTI) P3.3	I/O	Port 3,bit 3; external interrupt 1
14	(TO) P3.4	I/O	Port 3, bit 4; Timer 0 external input
15	(T1) P3.5	I/O	Port 3, bit 5, timer 1 external input
16	(WR) P3.6	I/O	Port 3, bit 6; writes strobe for external data memory
17	(RD)P3.7	I/O	Port 3, bit 7; reads strobe for external data memory
18	XTAL 2	I/O	Inverting oscillator amplifier (crystal)
19	XTAL 1	I/O	Inverting oscillator amplifier (crystal)
20	GND	I/O	Circuit ground
21	(A8)P2.0	I/O	Port 2, bit 0; address bit 8
22	(A9)P2.1	I/O	Port 2, bit 1; address bit 9
23	(A10)P2.2	I/O	Port 2, bit 2; address bit 10
24	(A11)P3.3	I/O	Port 2, bit 3; address bit 11
25	(A12)P3.4	I/O	Port 2, bit 4; address bit 12
26	(A13)P2.5	I/O	Port 2, bit 5; address bit 13
27	(A14)P2.6	I/O	Port 2, bit 6; address bit 14
28	(A15)P2.7	I/O	Port 2, bit 7; address bit 15
29	PSEN	Output	Program store enable, reads strobe for external program memory
30	ALE	Output	Address latch enable
31	EA	Input	External access enabled for the program
32	(AD7) P0.7	I/O	Port 0, bit 7; Address/data bit 7
33	(AD6) P0.6	I/O	Port 0, bit 6; Address/data bit 6
34	(AD5) P0.5	I/O	Port 0, bit 5; Address/data bit 5
35	(AD4) P0.4	I/O	Port 0, bit 4; Address/data bit 4
36	(AD3) P0.3	I/O	Port 0, bit 3; Address/data bit 3
37	(AD2) P0.2	I/O	Port 0, bit 2; Address/data bit 2
38	(AD1) P0.1	I/O	Port 0, bit 1; Address/data bit 1
39	(AD0) P0.0	Input	Port 0, bit 0; Address/data bit 0
40	Vcc	Input	Supply Voltage

International Journal of Basic Science and Technology
June, Volume 11, Number 2, Pages 159 - 169 http://www.ijbst.fuotuoke.edu.ng /
https://10.5281/zenodo.15601071 ISSN 2488-8648

Table 1 (b): Alternate Functions of Port 3

Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{\text{INT0}}$ (external interrupt 0)
P3.3	$\overline{\text{INT0}}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{\text{WR}}$ (external data memory write strobe)
P3.7	$\overline{\text{RD}}$ (external data memory read strobe)

Those functions that have a bar above them are made active when low.

Program Memory (On-Chip ROM): This is where the control programs executed by the microcontroller are stored (Rogers, 2013). The sizes and types of ROM vary in order to meet consumer's needs; but generally, they are in the other of a few kilobytes and flash ROMs are common these days.

Data Memory (On-Chip RAM): The AT89C51 microcontroller has an on-chip base memory of 128 bytes out of which 64 bytes (memory addresses 00h-7Fh) are reserved for the user. The remaining 64 bytes (memory addresses 80h-FFh) are reserved for the special function registers (SFR). However, the RAM is usually defined concerning the base memory available to the user (Isizoh, 2018). Hence, if RAM is said to be 128 bytes it is available to the user (without taking into account the special function registers).
The on-chip memory is divided as follows: the first 32 bytes made of four memory banks (bank 0-bank 3) are 8-bit registers with each bank possessing eight registers (R0-R7) of one byte each. The next 16 bytes are made up of 128-bit addressable memory locations from 00h through 7Fh, occupying 20h to 2Fh of internal RAM. The remaining 80bytes occupying RAM location 30h to 7Fh is the general purpose memory/registers.

Serial port: The AT89C51 microcontroller has a universal asynchronous receiver and transmitter (UART) otherwise known as a serial port which it uses for transmission of data in the form of pulses over longer distances where a parallel connection is impracticable. The AT89C51 takes care of the protocols needed for serial communication and all the programmer needs to do is select the serial communication mode and the baud rate (number of sent/received bits per second) and then the data to be sent can be placed on the SBUF register. The microcontroller's serial ports automatically take care of many of the details of serial communications. On the transmit side, the serial port translates bytes to be sent into serial data, including adding start and stop bits and writing the data in a timed sequence to SER OUT. On the receive side, the serial port accepts serial data at SER In and sets a flag to indicate that a byte has been received.

Oscillator: The oscillator produces equalized pulses needed to ensure the harmonic and synchronous operation of all the parts of the microcontroller.

External Interrupts - INTO and INTI are external interrupt inputs, which detect logic levels or transitions that interrupt the CPU and cause it to branch to a pre-defined program location.
Additional Control Inputs - Two additional control inputs need to be mentioned: the RESET pin and the EA pin inputs.
A logic high on the **RESET** resets the chip and causes it to begin executing the program that begins at 0000h in code memory.
EA is the External Memory Access. It determines whether the chip will access internal or external code memory in the area of 000h to 07FFh. When this pin

International Journal of Basic Science and Technology
June, Volume 11, Number 2, Pages 159 - 169 http://www.ijbst.fuotuoke.edu.ng /
https://10.5281/zenodo.15601071 ISSN 2488-8648

is tied low, the controller executes codes from external code memory.

Power Supply Connections - The chip has two pins for connecting to a +5 Volts DC power supply (Vcc) and Ground (Vss).

Reviewed Related Works

In the work (Isizoh, 2018) titled "Assembly Language Programming for Embedded Systems and Real-Time Applications" the author highlighted that several input devices or output devices can be interfaced with a microprocessor or microcontroller to achieve any desired display. Such I/Os can be light light-emitting diodes, seven segments, liquid crystal displays (LCD), sensors/transducers, electric motors, bulbs, etc.

The work of (Shams, 2011) described several ways of interfacing microcontrollers with other I/O devices in his work, titled, "Microcontroller Engineering". The work particularly discussed LCD interface with microcontroller and analyzed different kinds of microcontroller interfaces.

In (Jones, 2012), the work "Real-time Computing with Microcontrollers", proposed several ways of interfacing microcontrollers with I/O devices for real-time systems. Analyses on display interfaces were also done. The author also stated that the first microcontroller which was Intel 8051 is a Harvard architecture; a single-chip computer, developed by Intel in 1980 for use in embedded systems.

Comparison Between Existing Works

It was reviewed that other segment displays from other authors' work could not display certain characters. The work from other authors showed that their interfaces cannot display some alphabets like X, Z, V, W, etc; which necessitated the use of LED in this paper.

Materials and Methods

Materials

The materials used for the work include the hardware components: Microcontroller power supply unit, controller unit, row driver unit, column driver unit, and the LED dot Matrix display unit, jumper wires and PCB for assembly. The software used includes Arduino IDE for programming the microcontroller, IoT platform (blynk/firebase), Wi-Fi for connection to the internet, mobile app

Block Diagram of the Proposed System

The block diagram of the system is shown in Figure 2.

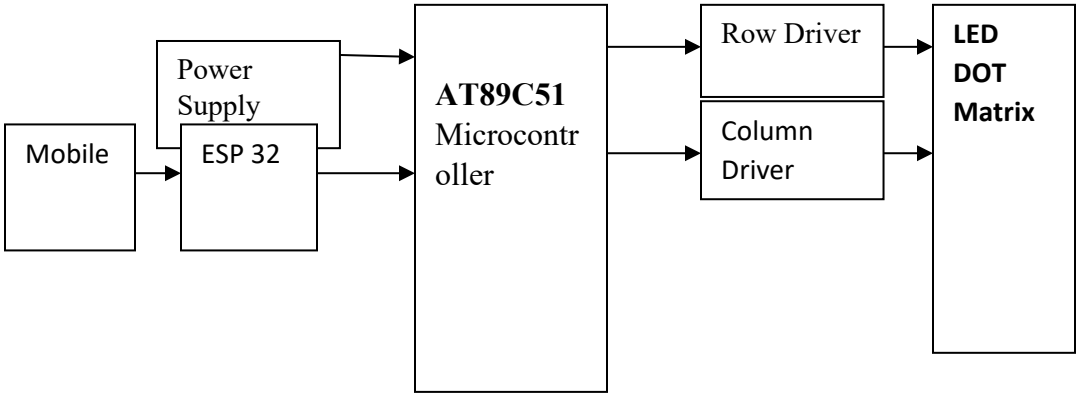

Figure 2: The Block Diagram of the System

Description of Block Diagram

Power Supply: This is responsible for the generation of 9V required for the running of the entire system. It is made up of a switch mode power pack of 9V by 2A current.

Controller: This is the brain of the system. It controls all the functionalities of the entire system. The controller used in this paper is AT89C51 of the 8051 family.

LED DOT Matrix: This is the display unit where all the needed information is displayed. It is made up of LEDs arranged in matrix form.

Row Driver: This is responsible for driving data (information) received from the controller to the LED dot matrix for display

Column Driver: This is responsible for the switching of the various columns of the dot matrix display. It is made up of transistors and resistors.

Android Mobile Phone and App. This Android phone has an application installed on it and it serves as the transmitter (Prachee *et al.,* 2017)

ESP32(Wi-fi Module). This serves as the receiver. The signal is given to the Wi-fi module ESP32 which

International Journal of Basic Science and Technology
June, Volume 11, Number 2, Pages 159 - 169 http://www.ijbst.fuotuoke.edu.ng /
https://10.5281/zenodo.15601071 ISSN 2488-8648

is connected to the matrix display through a USB connector. It connects to a server or phone app to enable IoT functionality.

The hardware setup is shown in Figure 3.

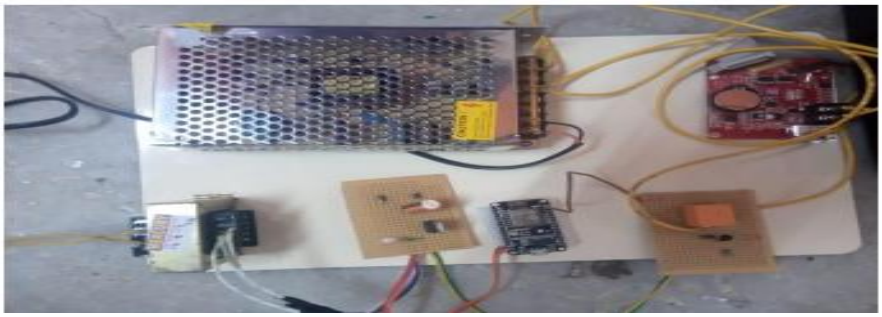

Figure 3: Hardware Setup

Communication Method

The method deployed here is called MQTT (Message Queuing Telemetry Transport). It is a lightweight protocol designed for low bandwidth. Low energy or unreliable networks are perfect for IoT applications. The ESP 32 acts as MQTT client while the LED display is a subscriber. All messages are routed through a MQTT broker. The ESP32 connects to the Wi-fi router and publishes text messages or sensor data to a predefined MQTT topic. The LED Display's controller connected to the MQTT broker subscribes to this topic. Whenever a new message is published the subscriber's call back parses the message and updates its displayed text in real time.

Working Principle of IOT-based LED Matrix

The aim of this paper is that whatever we speak on an Android phone should be displayed on the LED MATRIX. For the data to be displayed on the matrix, the speech signal which is given as input to the android must be converted into text. Thus, in the input section, we have an Android phone with an application installed on it. This application or GUI makes use of Google Voice for speech-to-text conversion. This converted sound signal is given to the Wi-fi module ESP 32. The Wi-fi module is further connected to the matrix display through a USB connector. The Android app acts like a transmitter and converter of sound signal to text signal or the Wi-Fi module acts like the receiver for speech-to-text converted signal and sends it further to the matrix display via microcontroller 8051. The medium of communication between transmitter and receiver is Wi-Fi.

The Microcontroller is used for controlling the driver circuitry of the matrix Display. The microcontroller is used to control the LED driver which consists of latches and line decoders and also to control the data lines which supply the data to the controller for lighting up the LEDS in a particular fashion. The look-up table decides which LEDS in respective columns and rows will glow for which particular character. The led driver consists of latches and line decoders. They control more number of LEDs. The word or the data that is given as input to the display is stored in the RAM IC which is interfaced externally with 8051. The DATA transmitted is sent to the display line by line i.e column by column

Power Supply Unit

This consists of a 220/12V step-down transformer in which the primary turns are connected to an AC supply, while the secondary turns are connected to a bridge rectifier whose function is to convert an alternating current (AC) input to a direct current (DC) output. A 1000μF capacitor is connected to the bridge rectifier to filter the D.C. signal that is produced. The positive terminal of the bridge rectifier with that of the capacitor is connected to the V_{in} of both the 7805 regulator which produces 5V and the 7812 regulators which produces 12V. The negative terminal of the bridge rectifier is connected to the ground of the capacitor and the regulators. The 5V signal is supplied to the registers, microcontroller and LEDs; while the 12V signal is supplied to the relay.

The transformer: The transformer used is a simple step-down transformer with secondary windings of 220V/50Hz, a primary winding of 12V and a current of 1000mA (which is greater than the current requirement of the circuit).

The primary and the secondary windings are related by the equation:

$$\frac{Vp}{Vs} = \frac{Is}{Ip} = \frac{Np}{Ns} \quad (1)$$

Where Vp, $Is$and Np = primary voltage, secondary current and number of turns of the primary coil; Vs, Ip and Ns = secondary voltage, primary current and number of turns in the secondary coil.

International Journal of Basic Science and Technology
June, Volume 11, Number 2, Pages 159 - 169 http://www.ijbst.fuotuoke.edu.ng /
https://10.5281/zenodo.15601071 ISSN 2488-8648

Since the maximum required voltage is 12V and the mains voltage supply is 220V, we have:

$$\frac{220}{12} = \frac{Np}{Ns} = \frac{55}{3} \quad (2)$$

Hence the ratio of the primary to the secondary coil of the transformer used is 55:3.

The bridge rectifier: A full wave bridge rectifier is used with a step-down transformer to convert the AC voltage coming into the circuit into a DC voltage. There are four diodes used. D1 and D2 are forward–biased and conduct current when the input cycle is positive. The positive half of the input cycle is made from the voltage across the load resistance, D3 and D4 are reversed biased. Throughout the negative input cycle, D3 and D4 are forward-biased and conduct current. A voltage is again made across the load resistance in the same direction as during the positive half-cycle. The full-wave rectified waveform of the output voltage across the load resistance is shown in Figure 3

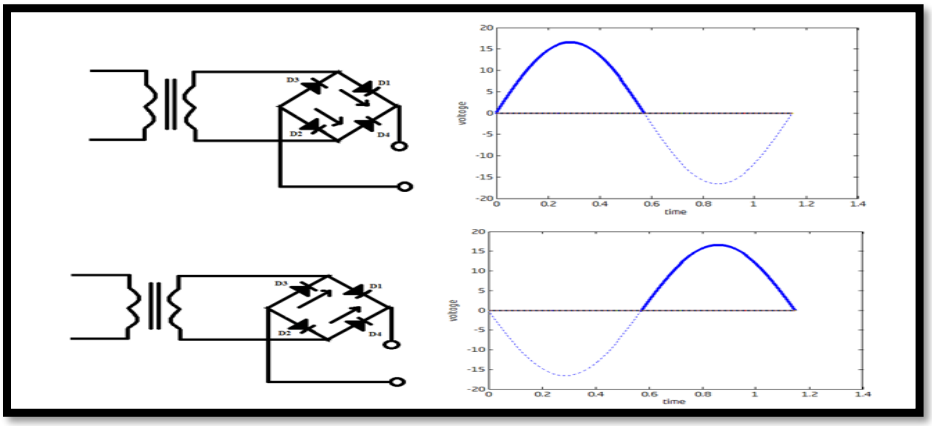

Figure 3: Positive and negative cycles of a full-wave bridge rectifier

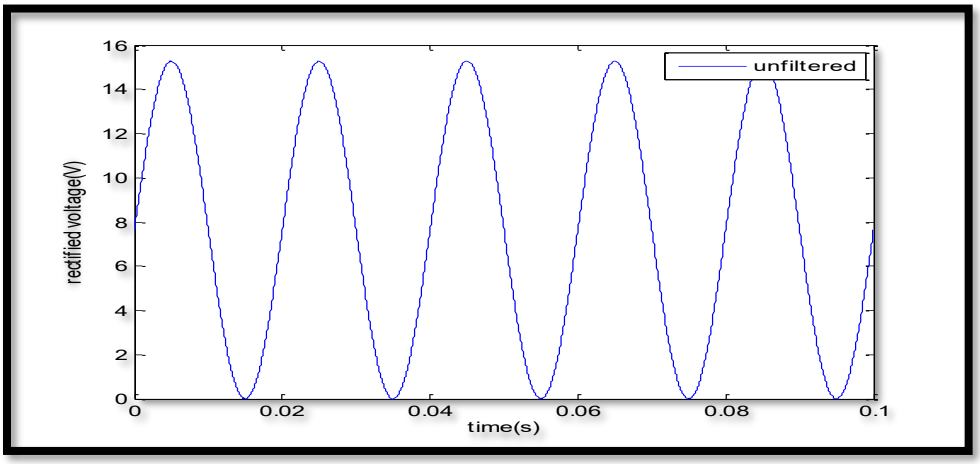

Figure .4: Full wave rectified voltage output

The filter: A capacitor is positioned in parallel with the output of the bridge rectifier to minimize the ripples in the rectified voltage, which will create a clean DC voltage. The filter capacitor filters the output voltage as shown in Figure 5.

International Journal of Basic Science and Technology
June, Volume 11, Number 2, Pages 159 - 169 http://www.ijbst.fuotuoke.edu.ng /
https://10.5281/zenodo.15601071 ISSN 2488-8648

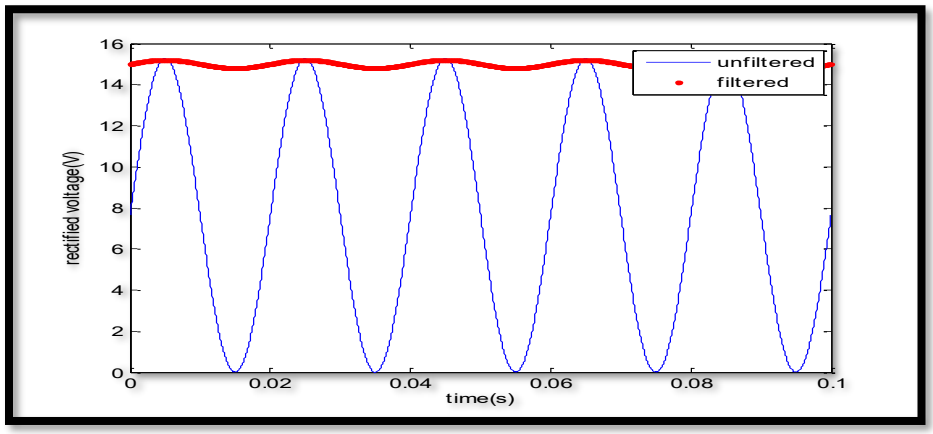

Figure 5 Filtered output waveform

The formula below is used to calculate the capacitance value for the filtering capacitor:

$$C = \frac{I}{2Vppf} \quad (3)$$

Where:

V_{pp} is the peak-to-peak ripple voltage

*I*s the current in the circuit

f is the frequency of the AC power

C is the capacitance.

From the transformer used,

$V_{pp} = \sqrt{2} \times 11.5 = 16.26V$;

I = 0.5A; *f* = 50Hz.

Hence, $C = \frac{0.5}{2 \times 16.26 \times 50} = 307.44\mu F$

The minimum capacitance that the calculations give when using the formula is not used, in practice a larger value is used so that the capacitor can charge more (1000µF is used).

Regulators: A voltage regulator IC called 7805 and 7812 are used to produce the 5V and 12V DC voltages needed for the microcontroller and the relay circuit. The output from the filter circuit is 15V when tested, and after regulation, we have 5V and 12V. The final circuit for the power module is shown in Figure 6.

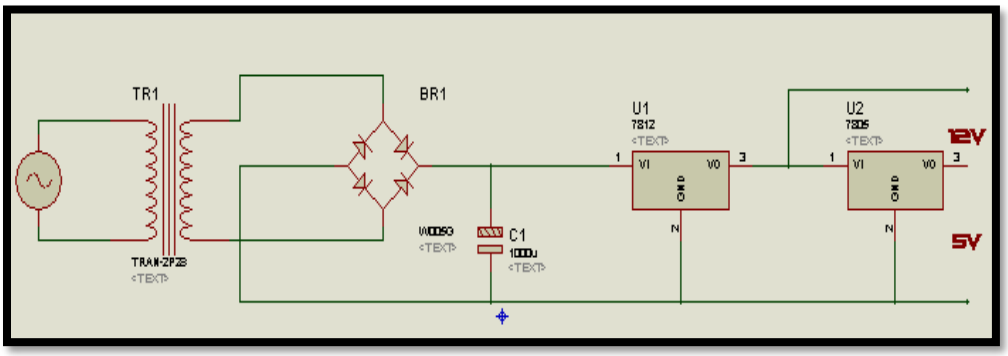

Figure 6: The sub-circuit diagram of the power unit

MicroController unit

The Controller unit, shown in Figure 7 is the main brain of the system because all the process flows are controlled by this hardware according to how it is programmed (Douglas 2018). It consists of one device called the microcontroller. The work of the microcontroller is to monitor and control the activities of the devices/components in the system. The type of microcontroller used is Atmel 89C51 which has 8 special pins and 32 programmable input and output pins. The AT89C51 microcontroller was chosen as the controller for the project since it offers various functions that are applicable to the system to be developed; also it is the most available microcontroller in the market. AT89C51 has a power circuit, reset circuit, and clock circuit.

Power circuit: The power circuit provides power for the microcontroller. The AT89C51 uses a voltage of 5V DC and this is supplied by the power module. The power circuit of the microcontroller simply involves connecting pin-40 of the controller to the 5V supply and pin-20 to the ground.

The reset circuit: This circuit resets the device when a high is on the reset pin (pin-9) for two machine cycles.

International Journal of Basic Science and Technology
June, Volume 11, Number 2, Pages 159 - 169 http://www.ijbst.fuotuoke.edu.ng /
https://10.5281/zenodo.15601071 ISSN 2488-8648

Clock circuit: This provides timing for the microcontroller. The AT89C51 can generate its internal clock signal. In order to generate a clock or the microcontroller, the output of the clock circuit must be connected to XTAL1 (pin-18) and XTAL2 (pin-19). An oscillator and 2 capacitors are required for the connection. If a crystal oscillator is used, then the capacitors required will be 30pF±10 and if a resonator is used, the capacitors will be 40pF±10.

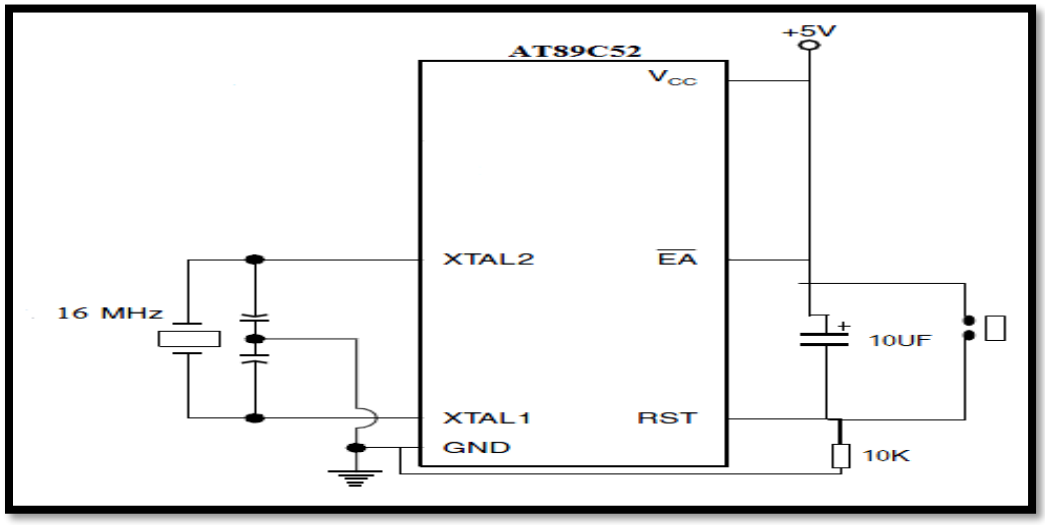

Figure 7: The Microcontroller Unit

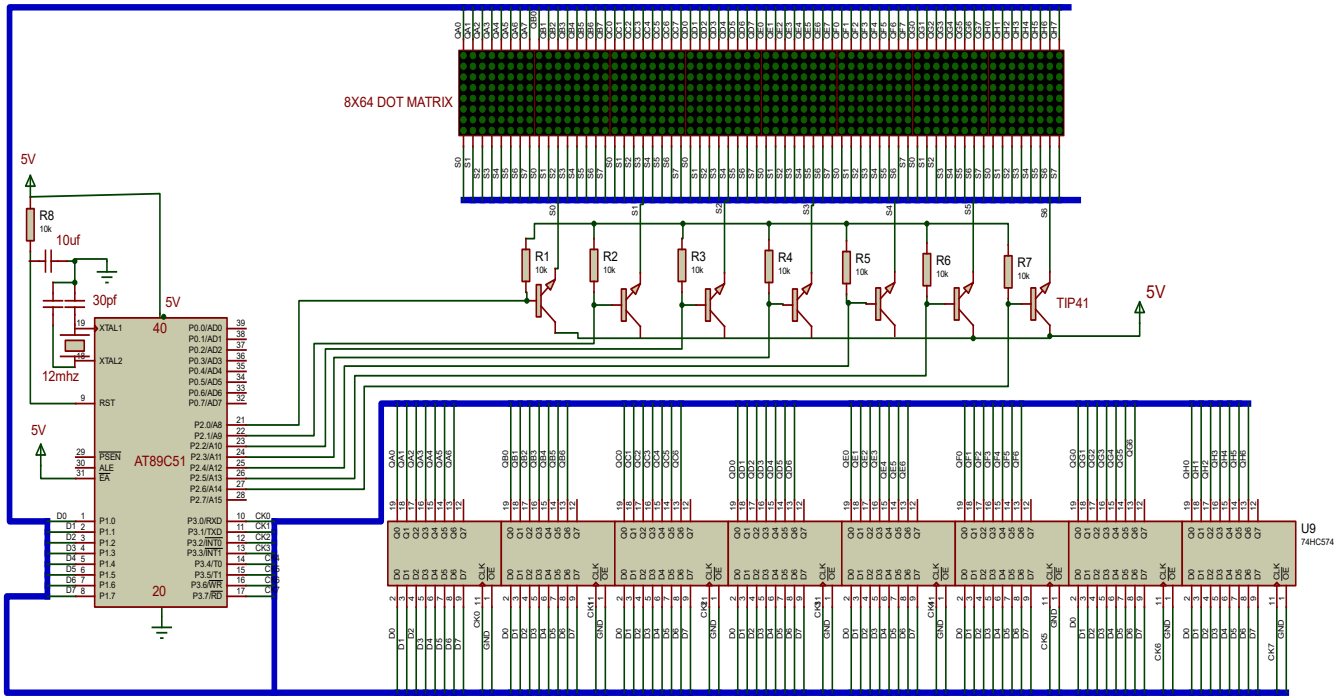

Figure 8: Circuit Diagram of the Dot Matrix Display

The flowchart in Figure 8 makes coding simpler since it elaborates on the interaction between the various functional elements of the system as well as the objects.

International Journal of Basic Science and Technology
June, Volume 11, Number 2, Pages 159 - 169 http://www.ijbst.fuotuoke.edu.ng /
https://10.5281/zenodo.15601071 ISSN 2488-8648

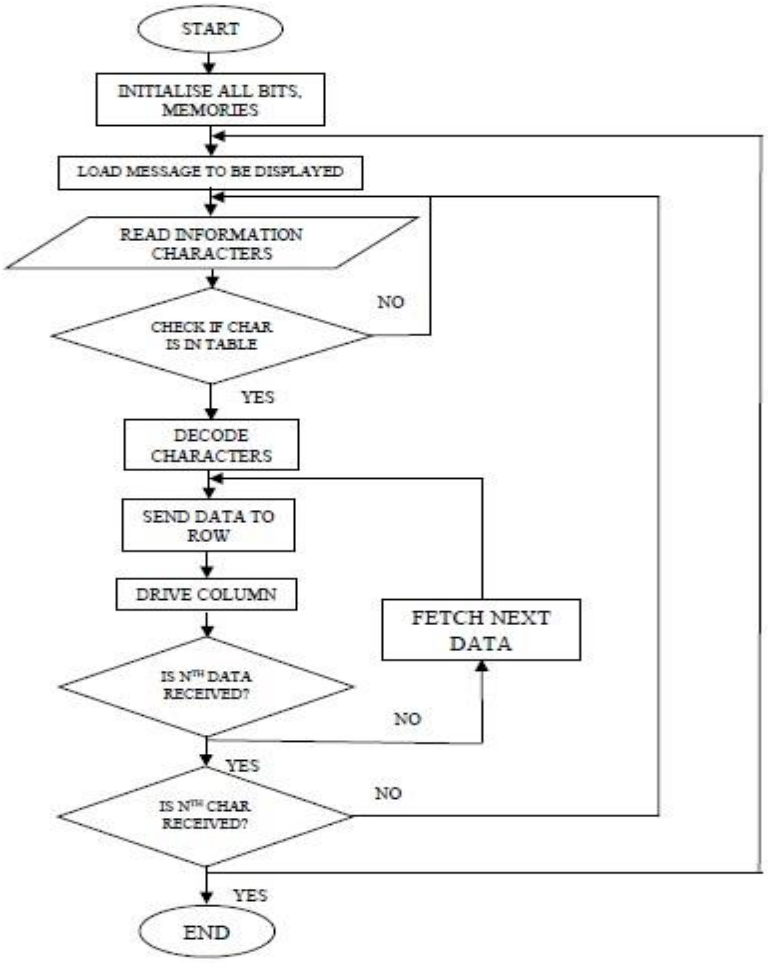

Figure 9: The flowchart for the Dot Matrix Display

The flowchart in Figure 9 makes coding simpler since it elaborates on the interaction between the various functional elements of the system as well as the objects.

The Pseudocode is as follows:

Step 1: Begin
Step 2: Initialize all the bits, memories
Step 3: Load messages to be displayed
Step 4: Read the information characters
Step 5: If the characters are not on the table, then move to step 4
Step 6: Else move to step 7
Step 7: Decode the characters
Step 8: Send data to the row
Step 9: Drive the column
Step 10: If the n^{th} data is not received, fetch the next data and move to step 8
Step 11: Else if the nth data is received, move to step 13
Step 12: Else move to step 4
Step 13: Move to step 3
Step 14: Stop

After successful coding, the program was saved with a file name ending with the '.asm' extension. Next, it was run by clicking the 'build' command in the MIDE-51 (the Assembler used in developing the program code) toolbar.

Since the run program was without errors, the '. *Hex*'s file generated was used in the real-time simulation of the circuit with ISIS schematic capture tool in the Proteus 7.5 software. It is also this hex file that was burnt into the microcontroller chip using a Wellon Universal Programmer.

Results and Discussion

Operational Result

The work is 6 x 48 inches, and it is a microcontroller-based system for Dot Matrix display. AT89C51 microcontroller was interfaced with LEDs, registers and other discrete electronics components. The program was written in Assembly language so that the system could display:

WELCOME TO THE DEPARTMENT OF COMPUTER ENGINEERING, MADONNA UNIVERSITY NIGERIA, AKPUGO CAMPUS. THE LECTURERS IN THE DEPT ARE: DR. E.C. IFEAGWU (HOD), PROF. C. UZOAMAKA, PROF. MRS. KALU, DR. A. ISIZOH, DR. MRS. EZE, DR. MRS. EJIMOFOR, ENGR. UJU OBIANYO, ENGR A. ANI, AND ENGR. M. OBI.

The above display is usually seen whenever the system is switched on, and the system works perfectly.

Testing

The circuit of this system was tested module by module as it was being integrated. These tests include:
Each component was checked to see if they were ok before they were used in the circuitry.
The potentiometer was adjusted to the proper range.
The power supply was properly checked to see if it falls within the tolerance calculated theoretically so as to avoid damage to the components.
The components which were not functioning properly were changed as soon as possible to avoid other components being affected by the damage.
Testing equipment was in proper range of the output measured at any point of the circuit, or component so as to avoid wrong readings.

Conclusion

The work showed that information could be programmed and allowed to be displayed on a microcontroller-based LED Matrix Display System. The AT89C51 microcontroller coordinates the activities of other devices/components interfaced with it; such as registers, LEDs and crystal, oscillators. The system was programmed in Assembly Language and was later simulated using Proteus 7.5 software, to ensure the workability of the project in a real-life situation. The subsystems (modules) were later interconnected to form a Dot Matrix system of 6 x 48 inches, which was tested and it worked satisfactorily.

References

Douglas, V. (2018). *Microcontrollers and Interfacing: Programming Hardware.*McGraw Hill Inc., New York. Pp. 270-344.

Edward, H.(2015). *Electrical and Electronics Technology*. Pearson Education Ltd, India PP. 634.

Floyd, T.F.(2018). *Digital Fundamental*. 6th Edition. Prentice-Hall International Inc, New Delhi, Pp. 4-7.

Foram, K., Anubbhav, M. and Pritish, M. (2020). Display Message on Notice Board using GSM. IJEIT, 3(7) pp. 827-832.

Holliday, D. (2010). *Fundamentals of Microcontrollers*. Don Peters Press Ltd, Fulmar, P. 88.

Isizoh A.N., (2018). Assembly Language Programming for Embedded Systems and Real-Time Applications. Scoa Heritage Publishers Nig. Ltd, Awka, Pp.15-28.

Jones, C.N., (2012). The Concise Book of real time systems. Timesys Corporation, Pittsburgh, 2012.

Katz, P. (2009). Digital Controls Using Microcontrollers. Prentice Hall International Inc, New Delhi, Pp. 15-43.

Kumar, R. and Sharma A. (2020). Design and Implementation of an IOT based smart notice board using ESP8266. *International Journal of Electronics and Communication Engineering*, 12(4), 25-30.

Millman, H. (2003). *Integrated Electronics*, McGraw – Hills Kogakusha, Pp. 112-114.

Muhammad, A.M.(2018). The 8051 Microcontroller and Embedded Systems.Prentice Hall, New Jersey USA.

Nikhil, M. (2021). Microprocessor and Interfacing, S.K. Kataria & Sons, New Delhi, .

Predko, M.(2017). Handbook of Microcontrollers. McGraw Hill, New York, USA, Pp 1-17.

Prachee, U., Ketkar, K.P.,Tayade, A.P., Kulkarni, R.M. and Tugnayat, M. (2017). GSM Mobile Phone Based LED *Scrolling Message Display System.* International Journal of Scientific Engineering and Technology, 2 (3), 149-155.

Rogers, S. P.(2013). *Software Engineering, a practitioner's approach*. McGraw-Hill, New York.

Shams H. K.(2011). Real-Time Systems: Design Principles for Distributed Embedded Applications, Kluwer Academic Publishers, Dordrecht, Pp. 6-11.

Wakerly, J.F. (2011). Digital Designs: Principles and Practices. 4th Edition, PHI Learning Private Limited, *New Delhi*, Pp. 5-6.

Yanchuang, D. and Jinying G. (2018). LED Display Screen Design and Proteus Simulation Based on Single-chip Microcontroller, *International Conference on Information Eng and Computer science (ICIECS)* (pp. 25-28).