International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 4 Apr 2022, pp: 97-104 www.ijaem.net ISSN: 2395-5252

Application and Analyses of Microcontrollers in Real-Time Systems

Isizoh A.N.[1], Obianyo O.R.[2], Ejimofor I.A.[3], Ebih U.J.[4], Alagbu E.E.[5]

1,4,5: Dept. of Electronic and Computer Engineering, Nnamdi Azikiwe University, Awka, Nigeria,
2,3: Dept. of Computer Engineering, Madonna University of Nigeria, Akpugo Campus.

Submitted: 01-04-2022 Revised: 06-04-2022 Accepted: 11-04-2022

ABSTRACT
This paper presents the applications of microcontrollers in real-time systems. Critical analyses of real-time systems and microcontroller were done. A real-time system is a system that its operation is dependent on time condition; while a microcontroller is a single chip computer. The type of microcontroller used in this work for analyses is AT89C55 microcontroller, and can be used (applied) in several real-time systems. It has a large program memory capacity, unlike other microcontrollers which have little program memory capacity. It then means that it can accommodate a large control program in its memory. Any real-time system other than the traffic light used as a case study for this work can as well be handled by just programming the microcontroller for such application. Assembly language was used to program the microcontroller for different real-time applications. When the system was tested, it was observed that with a good and adequate program, a given microcontroller can be used for different real-time applications.

KEYWORDS: Software, microcontroller, real-time, controller, memory

I. INTRODUCTION

1.1 An Overview of Real-Time Systems

A system is said to be real-time if the total correctness of its operation depends not only upon its logical correctness, but also upon the time in which it is performed [1]. There are hard real-time or immediate real-time system and soft real-time system.

In a hard real-time system, the completion of an operation after its deadline is considered useless; and however, this may cause a critical failure of the complete system. Hard real-time system is used when it is imperative that an event must take place within a strict deadline. Such strong guarantee is required of a system for which if the event does not take place within a certain interval of time, would cause a great loss in some manner (missing the deadline constitutes failure of the entire system). For example, a car engine control system is a hard real-time system because a delayed signal may cause an engine failure or damage. Other examples of hard real-time embedded systems include medical child incubator system, industrial process controllers, etc.

A soft real-time system on the other hand will tolerate such lateness, and may respond with decreased service quality (for example omitting frames while displaying a video). Soft real-time systems are typically used where there is some issue of concurrent access and the need to keep a number of connected systems up to date with changing situations; for example, software that maintains and updates the flight plans for commercial airliners. Live audio-video systems are also usually soft real-time; violation of constraints results in degraded quality, but the system can continue to operate [2].

1.2 Fundamentals of Microcontrollers

A microcontroller is a single chip computer. It is a computer because it has all the component parts of a typical digital computer. Those component parts are: the Central Processing Unit (CPU), Memory (ROM and RAM) and Input/output units. The difference between a microprocessor and microcontroller is that a microprocessor is only a processor without an internal memory, but can access external memory; while a microcontroller contains a processor as well as an inbuilt (internal) memory. A typical

International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 4 Apr 2022, pp: 97-104 www.ijaem.net ISSN: 2395-5252

block diagram of a microcontroller is shown in Figure 1.

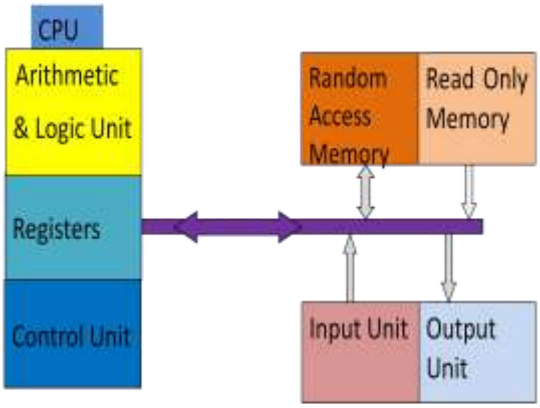

Figure 1: Block Diagram of a Typical Microcontroller

The block diagram of a microcontroller like the one used in this research (AT89C55) consists of the Central Processing Unit (CPU), the memory unit, and the Input/output unit.

The CPU - The Central Processing Unit (CPU) of AT89C55 microcontroller comprises of the Arithmetic and Logic unit (ALU), the Registers, and the Control unit.
The function of the ALU is to perform all the arithmetic and logic expressions from or within the CPU according to the instruction it receives. Such arithmetic instructions are addition, subtraction, etc; and logic expressions like AND, OR, NOT, etc.
However, the basic functions performed by the various registers are essentially the same in all processors. They are used to store data, addresses, instruction codes, and information on the status of various processor operations [3].
The Control unit controls the internal operations of the microcontroller and sends control signals to other parts of the microcontroller during program execution.

The Memory - Two on-chip memories exist in the chip. They are the Random Access Memory (RAM) and Read Only Memory (ROM).
The RAM - This is a volatile type of memory. The AT89C55 microcontroller uses the type of RAM called Static RAM (SRAM). Each SRAM cell contains at least four or five transistors to store a single bit of data.
The AT89C55 has a bank of 256 bytes of internal RAM.

The ROM – A ROM is a non-volatile memory. There is an on-chip Read Only Memory (ROM) embedded in AT89C55 microcontroller. There are different types of ROM: Programmable Read Only Memory (PROM), Erasable Programmable Read Only Memory (EPROM), Electrically Erasable Programmable Read Only Memory (EEPROM), and Flash. The type of ROM being used by AT89C55 is a 20K EEPROM. Its content can be erased by applying a suitable electrical voltage to the device.

I/O Ports - There are 32 input and output lines, which make up the four ports of the microcontroller. The ports enable the microcontroller to read and write to external memory and other components. The AT89C55 has four 8-bit I/O ports (Port 0 to 3). As the name suggests, the ports can act as inputs (to be read) or outputs (to be written to).

1.3 Internal Architecture of AT89C55 Microcontroller

The figure 2 shows the internal architecture of a typical AT89C55 microcontroller.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 4 Apr 2022, pp: 97-104 www.ijaem.net ISSN: 2395-5252

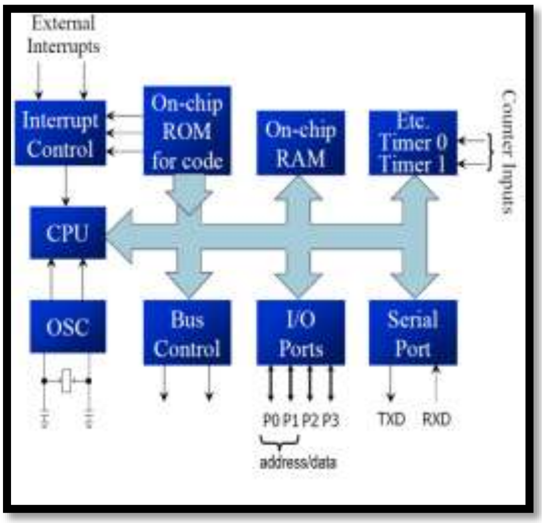

Figure 2: The Internal Architecture of AT89C55 Microcontroller

The CPU: The Central processing Unit (CPU) of the microcontroller comprises of 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 [4].

The Interrupt Control: The AT89C55 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 [5].

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 counts events. It is also employed in generating clock pulses (baud rates) when the microcontroller is used in serial communication.

The I/O Ports: The AT89C55 microcontroller has I/O ports used for interfacing the microcontroller to external devices like keypads, relays, sensors, ADCs, etc. Apart from the use as 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.

Table 1: 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	$\overline{\text{INT1}}$ (external interrupt 1)

P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{WR}$ (external data memory write strobe)
P3.7	$\overline{RD}$ (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. 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 [6].

Data Memory (On-Chip RAM): The AT89C55 microcontroller has an on-chip base memory of 256bytes; out of which 128bytes (memory addresses 00h-7Fh) are reserved for the user. The remaining 128bytes (memory addresses 80h-FFh) are reserved for the special function registers (SFR). Figure 3 shows the AT89C55 microcontroller RAM Structure. However, the RAM is usually defined with respect to the base memory available to the user. Hence, if a RAM is said to be 128bytes, it means that 128bytes is available to the user (without taking into account the special function registers).

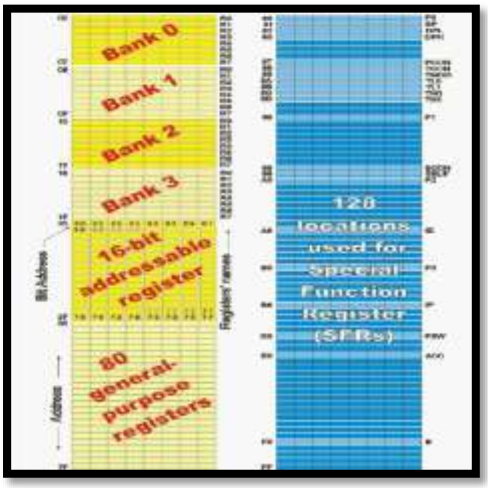

Figure 3: The AT89C55 Microcontroller RAM Structure

The on-chip data memory is divided as follows: the first 32 bytes made of four memory bank (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 AT89C55 microcontroller has a universal asynchronous receiver and transmitter (UART) otherwise known as serial port which it uses for transmission of data in the form of pulses over longer distances where parallel connection is impracticable. The AT89C55 takes care of the protocols needed for serial communication and all the programmer needs to do is to 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 [7].

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

1.4 The Ports/Pins Configuration in AT89C55 Microcontroller

The Ports/Pins Configuration in AT89C55 Microcontroller is shown in Figure 4.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 4 Apr 2022, pp: 97-104 www.ijaem.net ISSN: 2395-5252

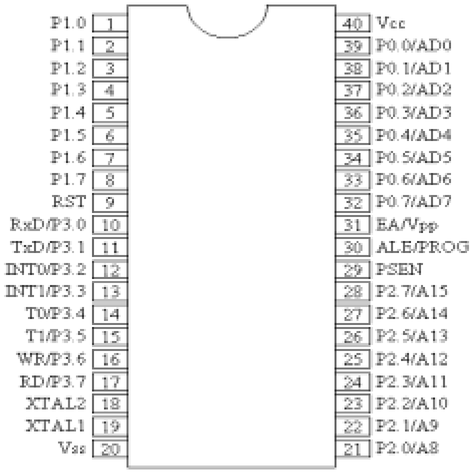

Figure 4: The Ports/Pins Configuration in AT89C55 Microcontroller

The AT89C55 Microcontroller has four ports (P0, P1, P2 and P3). Each of these ports can be addressed to in a program for the purpose fetching or sending instruction. There are 40 pins in this microcontroller. Of these 40 pins, about eight pins are not included in the ports and they are called special pins. They are special in the sense that their connections in a circuit are specific. These eight pins are Pin 9, Pin 18, Pin 19, Pin 20, Pin 29, Pin 30, Pin 31 and Pin 40. The rest of the 32 pins are used for input and output interfaces [8].

II. ANALYSES OF THE APPLICATIONS OF AT89C55 MICROCONTROLLER IN REAL-TIME SYSTEMS

Microcontrollers are independently programmable and can have a great deal of additional functionality combined on the same integrated circuit. One good thing about AT89C55 microcontroller is that it has a large memory which can allow the microcontroller to perform different real-time applications, since it can conveniently hold large programs. Just as it is applicable to other microcontrollers, when applying this type of microcontroller for real-time operations, what is needed is to develop an appropriate control program for the task and "burn" it inside the microcontroller. Of course, without a program inside a microcontroller, no control operation can take place. With adequate control programs where time constraints are included, the use of AT89C55 microcontroller will help in monitoring and controlling several real-time operations [9].

AT89C55 Microcontroller can be used in achieving different ranges of systems that operate in real time because of its relatively large memory. This includes temperature controller, real time clock, traffic lights, timed bomb, space applications, copiers, cable television terminal equipment, lawn sprinkling controllers, credit card phone equipment, cellular phones, fax machines, automotive applications such as engine control modules, anti-lock braking systems, automobile suspension control, keyless entry systems, display systems, and a host of other industrial and consumer applications [10]. It should be good to analyze some of these real-time systems examples where the AT89C55 microcontroller could be applied.

In a computer storage system having multiple disk drives and multiple power supplies, an environmental monitoring unit (EMU) is controlled by a microcontroller. The EMU performs a variety of tasks including monitoring power supply voltage and currents, fan speeds, temperature of the storage system enclosure, and the status of the various disk drives in the storage system.
In household appliances, microcontrollers are part of microwave ovens, televisions, calculators, remote controls, clocks, etc.

In a car, they are used in the engine control modules, the antilock braking systems, the sound systems, and automobile suspension control modules. In antilock braking systems, the microcontroller monitors the rotational speed of the tires through sensors attached to the tires.

As an example, let us briefly discuss how a microcontroller is used for display. The display in this example is to scroll the word "FULL" on a seven segment display.

Seven segment has a common anode type and a common cathode type. In a common anode type, all the anodes of the LEDs are tied together and grounded, while in a common cathode type, the cathodes of all the LEDs are tied together and grounded.

Each type contains seven LEDs connected in a segmented form lettered a, b, c, d, e, f, and g. In a common anode type, if you set zero (0) to any letter, the LED of that letter will be ON, but if you set it one (1), it will be OFF. However, in a common cathode, the reverse is the case. If you set zero (0) to any letter, the LED of that letter will be OFF, but if you set it one (1), it will be ON.

The real-time simulation of AT89C55 microcontroller for scrolling a word “FULL” in a common anode 7-segment is shown in Figure 5.

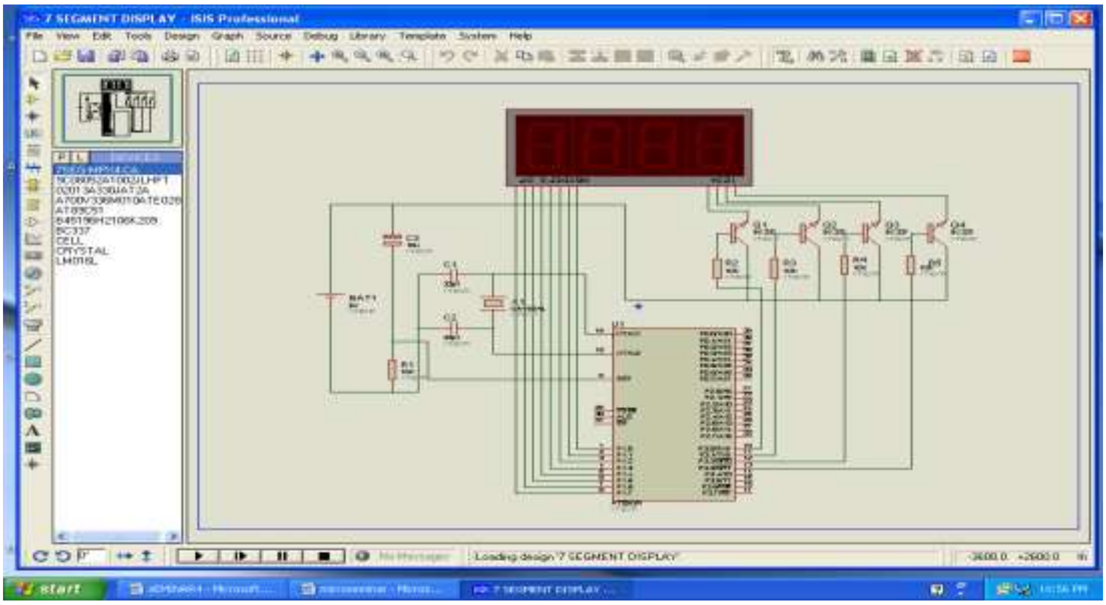

Figure 5: Seven Segment Display Interfaced to AT89C55 Microcontroller in Proteus

In designing a program for this, you must note that each of the display segments would be equated to a particular memory location. Also note that there are as many as 0 – 127 memory locations you can use for this equation, and each memory location is 8-bit which is equal to one byte.

As usual there must be five subroutine programs. The five subroutines are: Shift, Screen, Screen_delay, Letters, and Delay subroutines. The 7-segment codes for the display are represented thus:

F = 10001110b

U = 11000001b

L = 11000111b

L = 11000111b

In another example, the application of AT89C55 microcontroller in real-time system for a Five-Way Traffic Light Control System can be analyzed. Its real-time simulation could be performed using Proteus ISIS Professional software. All the electronic components used for the circuit were sourced from the internal components arrangement within the Proteus software. The essence of performing this real-time simulation is to show how this AT89C55 microcontroller could be applied in the design of a traffic light control system.

When the circuit in Figure 6 was connected and run in Proteus software, the traffic light controller (which is basically the AT89C55 microcontroller plus few electronic components) is seen operating as programmed.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 4 Apr 2022, pp: 97-104 www.ijaem.net ISSN: 2395-5252

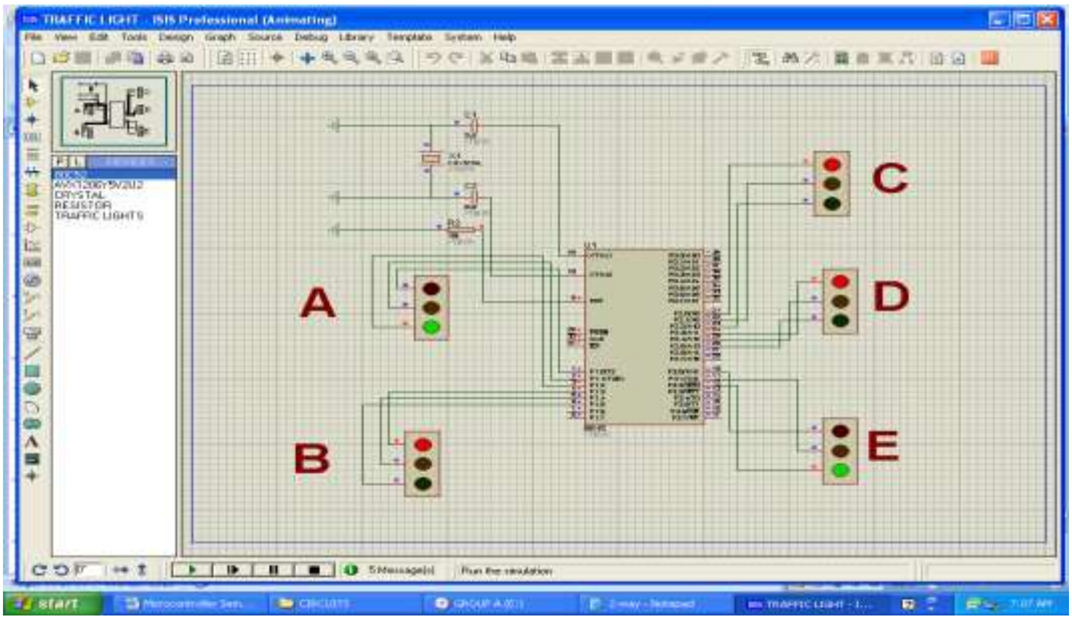

Figure 6: A 5-Way Traffic Light Control System Using AT89C55 Microcontroller

III. CONCLUSION

It has been shown that with adequate control programs, the use of AT89C55 microcontroller will help in monitoring and controlling several real-time operations using that single microcontroller, unlike using lower ROM-capacity microcontrollers such as AT80C51, AT89C51 (with 4K of ROM). Lower ROM-capacity microcontrollers cannot contain many programs, and therefore their control operations are limited and not up to that of AT89C55.

When AT89C55 microcontroller is used for real-time applications, its function is to interpret input (user and environmental), communicate with other devices, and output data to a variety of different devices. In this way, it reduces the bill of material cost of the system significantly by reducing the number of components required to carry out complex functions, add a great deal of "user friendliness" to a product for a very little additional cost. These days, AT89C55 microcontroller is effectively used for real-time control applications and this seminar has demonstrated one of such applications in the control of traffic lights.

IV. RECOMMENDATION

The use of AT89C55 microcontrollers is highly recommended for real-time system controls. This is because of its relatively low cost, availability in markets, low power consumption, larger program memory capacity, etc. Researchers should be making use of this microcontroller instead of using lower-memory capacity microcontrollers for real-time systems; as AT89C55 will offer one the opportunity of using one microcontroller for several operations.

REFERENCES

[1]. Isizoh A.N., NwobodoNzeribe H.N., Aniedu A.N., "Assembly Language Programming for Embedded Systems and Real-Time Applications", Scoa Heritage Publishers Ltd, Awka, Nigeria, 2018.

[2]. Schuster A., "Microcontroller Principles and Applications", Maxon Press Ltd, Rochester, 2008.

[3]. Cookie J. L., "Theory of Illumination Technology", Sunde Publications, Canada, 2013.

[4]. Predko Myke, "Handbook of Microcontrollers", McGraw Hill, New York, USA, 2007, Pp 1-17.

[5]. Douglas V.H., "Microcontrollers and Interfacing: "Programming Hardware", McGraw Hill Inc, New York, 2008, Pp 270-344.

[6]. Holliday D. and Resmick Robert, "Fundamentals of Microcontrollers", Don Peters Press Ltd, Fulmar, 2008, P 88.

[7]. Kopetz H., "Real-Time Systems: Design Principles for Distributed Embedded Applications", Kluwer Academic Publishers, Dordrecht, 2009, Pp 6-11.

[8]. Katz P., "Digital Controls Using Microcontrollers", Prentice Hall International Inc, New Delhi, 2009, Pp 15-43.

[9]. Wakerly John F., "Digital Designs: Principles and Practices", 4th Edition, PHI Learning Private Limited, New Delhi, 2008, Pp 5-6.
[10]. Floyd T.F., "Digital Fundamental", 6th Edition, Prentice-Hall International Inc, New Delhi, 2008, Pp. 4-7.