www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 3 Issue 5 may, 2014 Page No. 6048-6052

Ambulance Controlled Traffic System Using RFID Technology with LabVIEW Simulation

***S. Chandrakanth Sagar*[1], *Dr. M. Narayana*[2]**

[1]B.TECH student, ECE Department
Jaya Prakash Narayan College of Engineering, Mahabubnagar-509001, Andhra Pradesh, INDIA
sagarjpnce@gmail.com

[2]H.O.D. and Professor, ECE Department
Jaya Prakash Narayan College of Engineering, Mahabubnagar-509001, Andhra Pradesh, INDIA
sai_15surya@yahoo.co.in

Abstract: ***Traffic management on the road has become a severe problem of today's society because of growth of the urbanization. This leads to traffic jam at the traffic junctions which in turn causes delay to ambulances. In order to overcome this problem, this paper presents a simple ambulance controlled traffic system. The main objective of this system is that to control the traffic, allowing an ambulance to arrive at a particular location without it having to stop anywhere until the destination is reached. This system includes RFID technology and LabVIEW software. An RFID reader reads the ID number from the corresponding ambulance RFID tag and then it is sent to microcontroller LPC 1768H, which is programmed, with the help of embedded C instructions. This microcontroller is capable of communicating with input and output modules. The RFID readers provide the information to the microcontroller so that it compares the received ID with default ID's stored in its memory. If the obtained ID gets matched with any of the ID's , then a green signal is given along the path of the ambulance or else no change in the signal takes place. The signal won't change from green color until the same tag is detected by the other reader in another route. If the tag is detected in other route, then a normal traffic signal operation is performed. This system includes the simulation observation. The operation which is performed on hardware circuit is similarly observed on front panel of the LabVIEW. Moreover, the designed system has simple architecture, fast response time, ease in understanding the working module, user friendliness and scope for further expansion.***

Keywords: Ambulance, Traffic System, RFID Technology, VISA tool, MAX 232, RS 232, LabVIEW, monitoring section.

1. Introduction

As we all know that traffic management on the road has become a severe problem of today's society because of growth of population and it causes delay to ambulances so that they won't reach the hospitals in time [11]. This in turn causes harm to the patient who is inside the ambulance. To solve this problem, traffic is to be controlled whenever ambulance arrives at the junction so that a green signal is to be given along its path [7]. This can be achieved by RFID technology and hence an efficient ambulance controlled traffic system using RFID technology with LabVIEW simulation is proposed. Radio frequency identification is a technique that uses the radio waves to identify the object uniquely. RFID technique is widely used in the various application areas like medical science, commerce, security, Electronic toll collection system, access control etc.

The main objective of this system is that allowing an ambulance to arrive at a particular location without it having to stop anywhere until the destination is reached. This project includes the LabVIEW simulation. The operation which is performed on the hardware circuit is similarly observed on front panel of the LabVIEW.

The hardware components used in this project are LPC 1768H, EM 18 RFID reader, RFID tag, MAX 232, RS 232, resistors, and LED's.

Microprocessors and microcontrollers are widely used in embedded systems products. Microcontroller is a programmable device. A microcontroller has a CPU in addition to a fixed amount of RAM, ROM, I/O ports and a timer embedded all on a single chip. The fixed amount of on-chip ROM, RAM and number of I/O ports in microcontrollers makes them ideal for many applications in which cost and space are critical.

LPC1768H is a Header Board designed for CORTEX M3 Based LPC1768 controller from NXP. The Board is a basic IO Pinout with options of onboard power and USB Device. The IO pins are taken out on 2.54 mm berg connector. The board has standard 20 PIN JTAG Connectivity for debugging. The board also has reset and ISP switches for in system programming. UART 0 can be used for ISP Programming. Internal System Programming is a way to write the flash memory through the UART0 ports in the microcontroller, when entering the boot loader after a reset, or launching the ISP mode from an already running user code. Pins on LPC1768 are divided into 5 ports starting from 0 to 4. Pin naming convention is Px.y where x is the port number and y is the pin number. For example P1.23 means Port 1, Pin 23. Each pin has four operating modes: GPIO (default), first alternate function, second alternate function, and third alternate function. Any pin of ports 0 and 2 can be used to generate an interrupt. The board can be programmed or debugged using standard JTAG. There are various Parallel / USB JTAG. Debuggers like Coo Cox can be used with this board [5]-[13].

RFID is an acronym for radio frequency identification. Briefly the RF stand for “radiofrequency” and ID means “identifier” that allows an item, for instance a library book, to be identified, accessed, stored, reprogrammed and communicated by using radio waves. Radio Frequency Identification (RFID) is a generic term for non-contacting technologies that use radio waves to automatically identify people or objects. There are several methods of identification, but the most common is to store a unique serial number that identifies a person or object on a microchip that is attached to an antenna. The combined antenna and microchip are called an "RFID transponder" or "RFID tag" and work in combination with an "RFID reader". An RFID system consists of a reader and one or more tags. The reader's antenna is used to transmit radio frequency (RF) energy. The tag will then modulate the electromagnetic waves generated by the reader in order to transmit its data back to the reader. The reader receives the modulated waves and converts them into digital data [10]. There are two major types of tag technologies. "Passive tags" are tags that do not contain their own power source or transmitter. When radio waves from the reader reach the chip’s antenna, the energy is converted by the antenna into electricity that can power up the microchip in the tag. The tag is then able to send back any information stored on the tag by reflecting the electromagnetic waves as described above. "Active tags" have their own power source and transmitter. The power source, usually a battery, is used to run the microchip's circuitry and to broadcast a signal to a reader. Due to the fact that passive tags do not have their own transmitter and must reflect their signal to the reader, the reading distance is much shorter than with active tags. However, active tags are typically larger, more expensive, and require occasional service.RFID system uses various frequencies but most common and popularly used frequency is low, high and ultra high frequency. Low frequency is around 125 KHz, high is around 13.56 MHz and ultra high varies between 860-960 MHz [2].

The MAX232 from Maxim was the first IC which in one package contains the necessary drivers (two) and receivers (also two), to adapt the RS-232 signal voltage levels to TTL logic. It became popular, because it just needs one voltage (+5V) and generates the necessary RS-232 voltage levels (approx. -10V and +10V) internally.

In telecommunications, RS-232 is the traditional name for a series of standards for serial binary single ended data and control signals connecting between DTE (data terminal equipment) and DCE (data circuit-terminating equipment, originally defined as data communication equipment). It is commonly used in computer serial ports [6].

A resistor is a two-terminal electronic component that produces a voltage across its terminals that is proportional to the electric current passing through it.

A light-emitting diode (LED) is a semiconductor light source. LED’s are used as indicator lamps in many devices, and are increasingly used for lighting.

The hardware circuit is interfaced with the software called LabVIEW. LabVIEW (short for **Lab**oratory **V**irtual **I**nstrumentation **E**ngineering **W**orkbench) is a platform and development environment for a visual programming language from National Instruments. The graphical language is named "G". Originally released for the Apple Macintosh in 1986, LabVIEW is commonly used for data acquisition, instrument control, and industrial automation on a variety of platforms including Microsoft Windows, various flavors of UNIX, Linux, and Mac OS. The latest version of LabVIEW is version LabVIEW 2011. The code files have the extension “.vi”, which is an abbreviation for “Virtual Instrument”[1].

The graphical approach also allows non-programmers to build programs simply by dragging and dropping virtual representations of lab equipment with which they are already familiar. The LabVIEW programming environment, with the included examples and the documentation, makes it simple to create small applications. This is a benefit on one side, but there is also a certain danger of underestimating the expertise needed for good quality "G" programming. For complex algorithms or large-scale code, it is important that the programmer possess an extensive knowledge of the special LabVIEW syntax and the topology of its memory management. The most advanced LabVIEW development systems offer the possibility of building stand-alone applications. Furthermore, it is possible to create distributed applications, which communicate by a client/server scheme, and are therefore easier to implement due to the inherently parallel nature of G-code [3].

LabVIEW programs are called **V**irtual **I**nstruments, or VIs, because their appearance and operation imitate physical instruments, such as oscilloscopes and millimeters. LabVIEW contains a comprehensive set of tools for acquiring analyzing, displaying, and storing data, as well as tools to help you troubleshoot your code. Each VI has three components: a block diagram, a front panel, and a connector panel. The last is used to represent the VI in the block diagrams of other, calling VIs. Controls and indicators on the front panel allow an operator to input data into or extract data from a running virtual instrument. However, the front panel can also serve as a programmatic interface. Thus a virtual instrument can either be run as a program, with the front panel serving as a user interface, or, when dropped as a node onto the block diagram, the front panel defines the inputs and outputs for the given node through the connector pane. This implies each VI can be easily tested before being embedded as a subroutine into a larger program [4].

One benefit of LabVIEW over other development environments is the extensive support for accessing instrumentation hardware. Drivers and abstraction layers for many different types of instruments and buses are included or are available for inclusion. These present themselves as graphical nodes. The abstraction layers offer standard software interfaces to communicate with hardware devices. The provided driver interfaces save program development time. The sales pitch of National Instruments is, therefore, that even people with limited coding experience can write programs and deploy test solutions in a reduced time frame when compared to more conventional or competing systems. A new hardware driver topology (DAQmxBase), which consists mainly of G-coded components with only a few register calls through NI Measurement Hardware DDK (Driver Development Kit) functions, provides platform independent hardware access to numerous data acquisition and instrumentation devices. The DAQmxBase driver is available for LabVIEW on Windows, Mac OS and Linux platforms [8].

2. Proposed System

2.1 Block diagram of the hardware circuit

Firstly, whenever the ambulance arrives at particular junction, the ambulance driver has to show his corresponding RFID tag to RFID reader. Then, the reader detects the radio frequency signal from the tag and generates corresponding identification code. This code is in turn sent to micro controller LPC 1768H. Here, microcontroller compares the received ID with the default ID numbers stored in its memory during programming. If any of the ID gets matched with the received ID, then microcontroller changes the state of signal by giving green signal along the path of the ambulance. This signal won't change from green until the ambulance driver shows the same tag to the another RFID reader which is located on the other route. For instance, assume that a green light glows along route 3 and an ambulance is coming across route 1. Then, as soon as ambulance arrives near to RFID reader1, the driver shows tag to reader and then reader detects tag and generates corresponding ID which is in turn compared with the default ID's by micro controller. If ID gets matched, then a green signal is given along route 1. The signal remains same until the tag is detected by any one of the readers of three routes that are route 2, route 3, and route 4.

Secondly, the microcontroller generates the value ranging from 0 to 8 in terms of voltage and then MAX 232 adapts the TTL logic to RS-232 signal voltage levels.

Finally, an 8 bit serial data is transferred from MAX 232 to processor with the help of RS 232 probe. Based on this received data, the signal on front panel of LabVIEW gets varied. Thus, it is practically found that the performances of both hardware and simulation are similar.

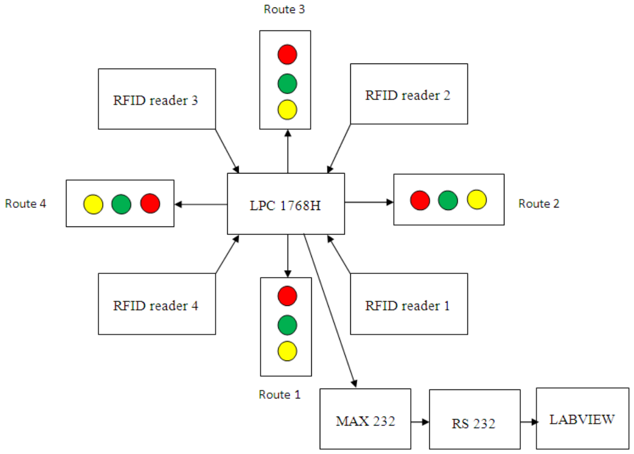


Figure 1: Block diagram of the hardware circuit

2.2 Block diagram of the LabVIEW

Block diagram plays a crucial role in the overall LabVIEW application. From the block diagram, it is clear that initially a VISA tool reads the string from COM1 port and then the string is passed to property node which extracts the properties of a received string. In turn string is sent to VISA read tool which receives the string and forward it to case structure. Here, in the case structure, based on the received string corresponding operation will be performed. Usually, a string generated is in between 0 and 8. This also includes the timer tool which specifies number of milli seconds to wait to read the next serial data from the hardware. Reading Times and Reading Stop tools act as a counter and stop buttons respectively. As soon as execution starts, the counter value keeps on increase until we stop the execution.

In the case structure, we can observe that respected leds are wired for two conditional tools true and false. For a received byte 0 the case structure is as below which resembles that when the condition meets all the yellow signals that is Y1, Y2, Y3, and Y4 should gets on. Similarly, there will be different case structure for all the different received bytes.

The outer square structure is known while loop. A While loop repeats the sub diagram inside it until the conditional terminal, an input terminal, receives a particular Boolean value. The Boolean value depends on the continuation behavior of the While Loop. Right-click the conditional terminal and select **Stop if True** or **Continue if True** from the shortcut menu. The While Loop always executes at least once.

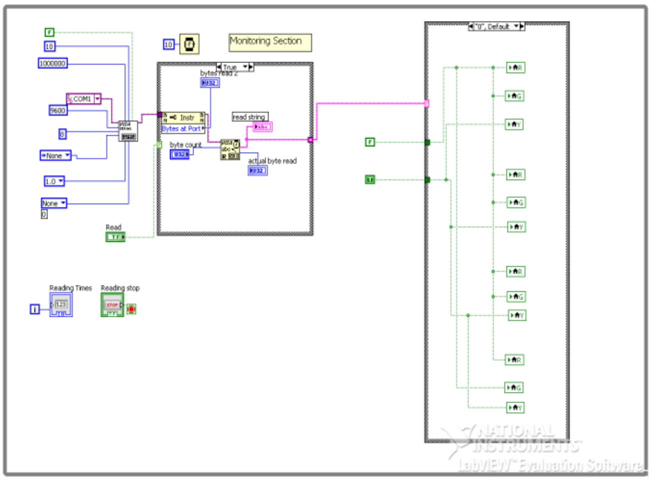
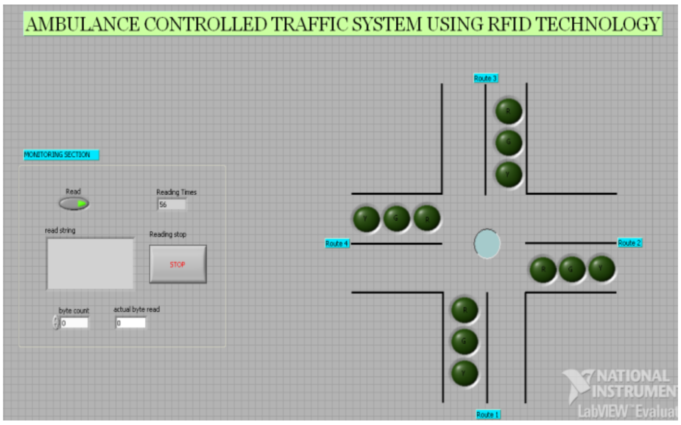


Figure 2: Block diagram of the LabVIEW

2.3 Front panel of the LabVIEW

The figure 3 represents the front panel of the LabVIEW. The front panel consists of two sections, one is monitoring section and the other is operating section.

In the monitoring section, read string specifies the string that read from RS 232, reading time indicates that number of times that RS 232 reads the string until execution is stopped. Byte count is the control that is an input at where the number of bytes to read is to be mentioned. Actual byte read is an indicator that displays the actual number of bytes reading by RS232.


AMBULANCE CONTROLLED TRAFFIC SYSTEM USING RFID TECHNOLOGY

MONITORING SECTION

Read

Reading Times 56

read string

Reading stop STOP

byte count 0

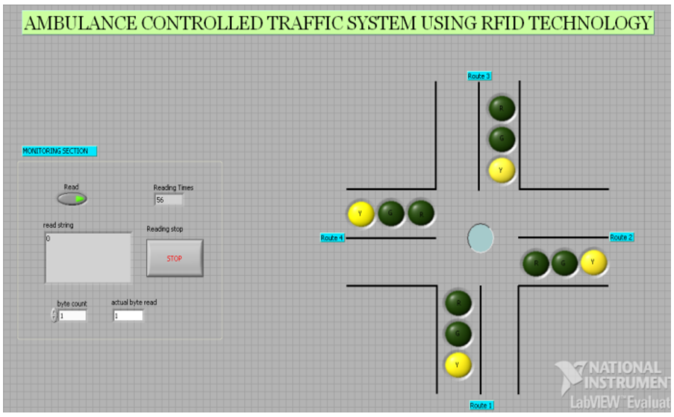
actual byte read 0

Figure 3: Front panel of the LabVIEW

In the operating section, four routes are observed. Each route has corresponding three signals of three different colors that are red, green, and yellow. The signals in route 1 are named as R1, G1, and Y1, route 2 are named as R2, G2, and Y2, route 3 are named as R3, G3, and Y3 and route 4 are named as R4, G4, and Y4[12]. Based on the generated byte value, the corresponding signal gets on.

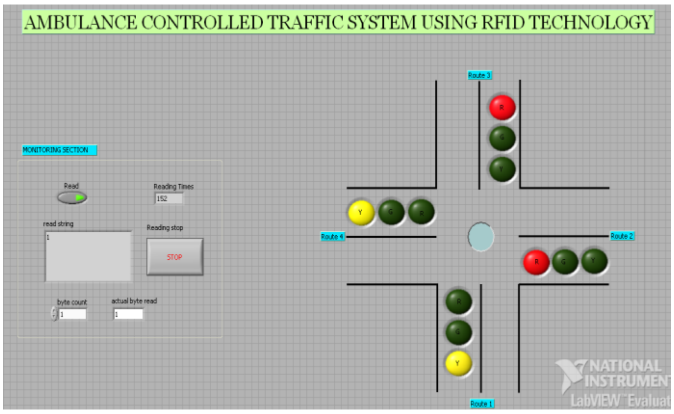
3. Results and Discussions

This system performs as a normal traffic system in the absence of ambulance. The operation gets altered whenever the ambulance arrives at a particular junction. The performance will be observed on the front panel of the LabVIEW. The signal state gets changed based on the byte transferred from MAX 232. The front panels of LabVIEW for 0, 1, and 2 are as follows respectively.



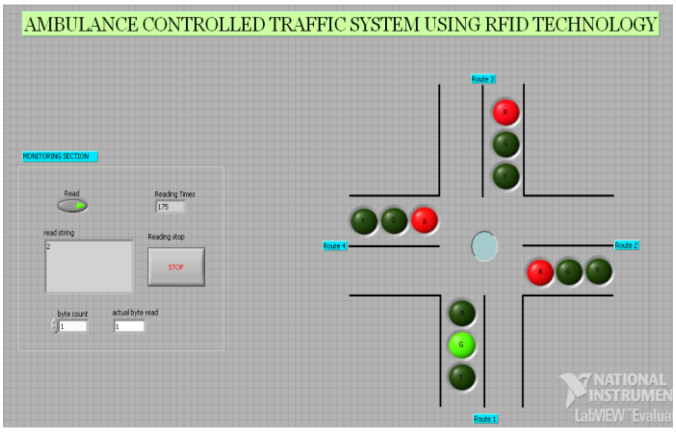
Reading Times 56
read string 0
byte count 1
actual byte read 1

Figure 4: Front panel for string 0



Reading Times 152
read string 1
byte count 1
actual byte read 1

Figure 5: Front panel for string 1



Reading Times 175
read string 2
byte count 1
actual byte read 1

Figure 6: Front panel for string 2

Similarly, the signal changes respectively for the respected sting value. The table 1 represents signal output for corresponding read string value [14].

Table 1: Output for corresponding read string

Read byte	Route 1	Route 2	Route 3	Route 4
0	Y1	Y2	Y3	Y4
1	Y1	R2	R3	Y4
2	G1	R2	R3	R4
3	Y1	Y2	R3	R4
4	R1	G2	R3	R4
5	R1	Y2	Y3	R4
6	R1	R2	G3	R4
7	R1	R2	Y3	Y4
8	R1	R2	R3	G4

The designed system has simple architecture, fast response time, ease in understanding the working module, user friendliness and scope for further expansion. Thus, this technology can be used in many industrial and military applications.

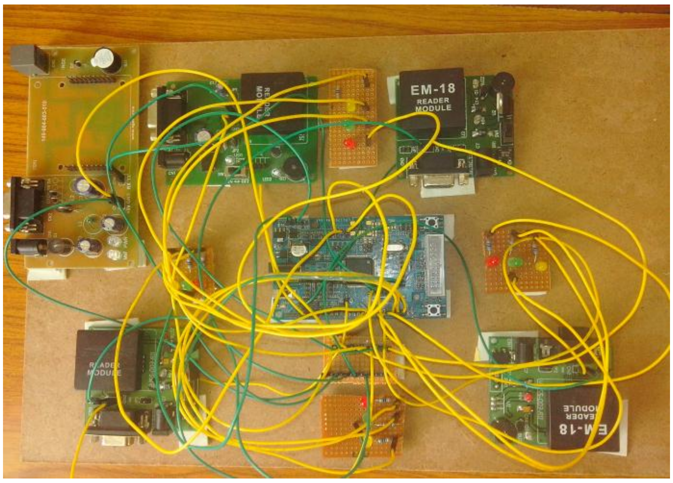


Figure 7: Proposed Hardware Circuit

4. Conclusion and Future scope

This system will definitely help the ambulance by giving the way to it when there is a heavy traffic on the road. This system uses the LabVIEW software to monitor the traffic whenever ambulance arrives at the junction. The operation which is performed in the hardware circuit is similarly observed on the front panel of LabVIEW. The designed system has simple architecture, fast response time, ease in understanding the working module, user friendliness and scope for further expansion [9].

This control system can be extended by introducing GSM module, which sends alert intimation to the respected hospital authorities stating that ambulance is near to the hospital. In future we can use this system in several applications by adding additional components to it.

References

[1] Peter A.Blume, “The LabVIEW Style Book”, Prentice Hall Publication, New Delhi, February 27, 2007, 2nd edition.

[2] Chong hua Li, “Automatic Vehicle Identification System based on RFID”, Anti Counterfeiting Security and Identification in Communication (ASID), 2010, pp 281-284.

[3] Gary W. Johnson, Richard Jennings, Paperback, “Lab VIEW graphical programming”, McGraw- Hill Publication, New Delhi, July 19,2007, 3rd edition.

[4] LabVIEW for Everyone: Graphical Programming Made Easy and Fun, Jeffrey Travis, Jim Kring, Third Edition. Prentice Hall Professional, 2007 ISBN-10: 0131856723.

[5] UM10360, LPC176X/5X user manual, Rev 3, December 2013.

[6] "RS232 Tutorial on Data Interface and cables". ARC Electronics. 2010. Retrieved 28 July 2011.

[7] Automatic Ambulance Rescue System, International Journal of Advanced Technology and Engineering Research, Volume 2, Issue 2, May 2012, ISSN NO: 2250-3536

[8] LabVIEW tutorial by Department of Electrical Engineering, Information Technology and Cybernetics Telemark University College.

[9] Traffic Control System Using LabVIEW published in Global Journal of Advanced Engineering Technologies, Volume 2, Issue 2, 2013, ISSN 2277-6370.

[10] RFID Essentials, by Bill Grover and Himanshu Bhatt, O’Reilly.

[11] Wang Wei, Fang Hanbo, “*Traffic accident auto-matic detection and remote alarm device*”, Proceed-ings of International Conference on Electric Information and Control Engineering, pages: 910-913, 2011.

[12] Zhaosheng Yang. “*Study on the schemes of traffic signal timing for Priority Vehicles Based on Navigation System*”, Proceedings of International Vehicle Electronics Conference, pages: 249-254, 2001.

[13] CoiNel Technology Solutions LLP presented LPC 1768H Header Board Overview material.

[14] Xu Li, Wei Shu, Minglu Li, Hong-Yu Huang, Pei-En Luo, Min-You Wu, “Performance Evaluation of Vehicle-Based Mobile Sensor Networks for Traffic Monitoring” IEEE transactions on vehicular technology, May 2009, vol. 58, no. 4, pp. 1647-1653.

Authors Profile

S. Chandrakanth Sagar is currently pursuing B.TECH. in Electronics and Communication Engineering department in Jaya Prakash Narayan College of Engineering, Mahabubnagar affiliated to JNTU-Hyderabad, INDIA. He had completed his intermediate (12th standard) from Scholars Junior College, Wanaparthy, INDIA with 96.80 % and he stood one among the top 15 state rankers. During his B.TECH academic, he had an industrial training in National Thermal Power Corporation, Ramagundam, which made him very enthusiastic towards research works.

Dr.M. Narayana is currently Professor and Head of ECE department, in Jaya Prakash Narayan College of Engineering, Mahabubnagar, INDIA. He received his Ph. D from JNTU Ananthapur. He Completed M. Tech from JNTU (Hyd) and B.Tech (ECE) from G. Pulla Reddy College of Engg. He had 13 years of teaching experience. His research interests are in the areas of Signal and Image Processing, Digital systems, signals and systems.