A Fully Automated Ambulance System

G. Sivaselvan[1], L. Karunakaran[2], R. Bakirathan[3], P. Santhosh[4], M. Vijaya Prathap[5]
[1, 2, 3, 4, 5] Department of CSE
[1, 2, 3, 4, 5] K.S.R college of Engineering, Tiruchengode.

Abstract- *The main aim of this paper is to develop a network that allowed for the transmitting and receiving of videos from camera nodes at the ambulance to a base station with a help of Wireless sensor network. This paper is based on monitoring of remote patients at ambulance itself for the emergency. In this project, the patient's body temperature, pulse rate are monitored and sends the data to corresponding doctor by using Zigbee protocol.*

Keywords- ZigBee, Wireless Technology, Heart Rate machine, Pulse Rate Machine.

I. INTRODUCTION

Nowadays the road accidents in modern urban areas are increased to uncertain level. The loss of human life due to accident is to be avoided. Traffic congestion and tidal flow are major facts that cause delay to ambulance. The main goal of our design is to develop a network that allowed for the transmitting and receiving of videos from camera nodes at the ambulance to a base station with a help of Wireless sensor network. Statistics reveal that every minute a human is losing his/her life across the globe because of the accidents. Everyday many lives are affected by heart attacks, pulse rate variation and more importantly because the patients did not get timely and proper first aid. This paper is based on monitoring of remote patients at ambulance itself for the emergency. In this project , the patient's body temperature , pulse rate are monitored and sends the data to corresponding doctor by using Zigbee protocol. Also the wireless camera is placed to monitor the visual process of patients. Thus this fully automatic ambulance system will be reasons to safe the affected people at emergency and by video capturing, some of the organs theft should be avoided.

II. EXISTING SYSTEM

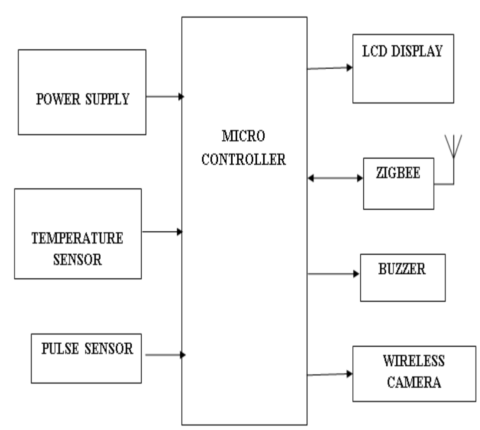
The ambulance is simply to transport patients to, from or between places of treatment which provide care to patients with an acute illness or injury. The ambulance are equipped with lifesaving machines like ventilator, cardiac monitors, etc.Here, only human body temperature of the patient could be measured i.e., it was used only for single purpose and hence it made high cost.

Here they did not use any communication purposes to connect with hospitals.

III. PROPOSED SYSTEM

In the proposed method, wearable wirelesses sensors are used for monitoring human body temperature as well as pulse rate are also monitored and sends the data immediately to the doctor through the Zigbee module, This system has multi-purpose application. This system is easy to design and is flexible. Cost is less. By using 16 X 2 LCD display, continuously monitoring the health conditions at the ambulance. Such as blood pressure level, temperature variation, heartbeat rate are monitored using the application of embedded system and the captured signals are transferred using zigbee based wireless communication technology. In order to reduce the time delay for patients to measure their health condition, this project is going to be made.

IV. BLOCK DIAGRAM

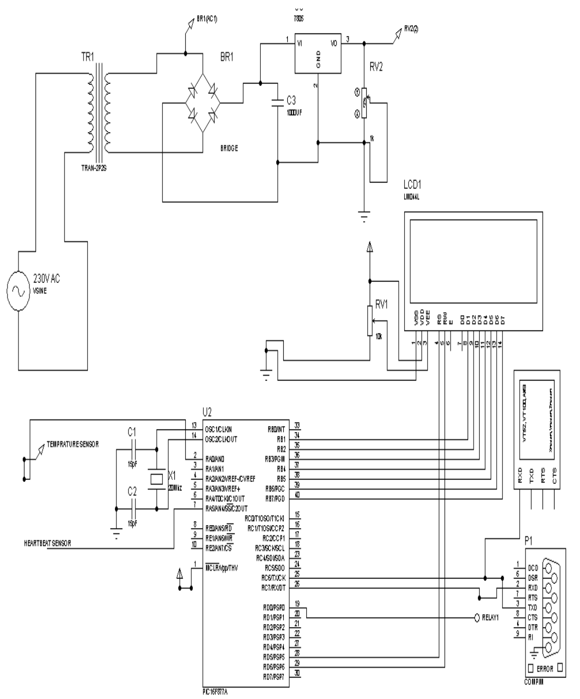


V. HARDWARE DESCRIPTION

5.1 MICROCONTROLLER

PIC microcontroller is one of the most popular microcontrollers used for the industrial purpose. PIC stands for Peripheral Interface Controller. It is being produced by Microchip Technology Inc, USA, which is one of the largest chip manufacturers in the world. PIC is highly cost effective and is field programmable.

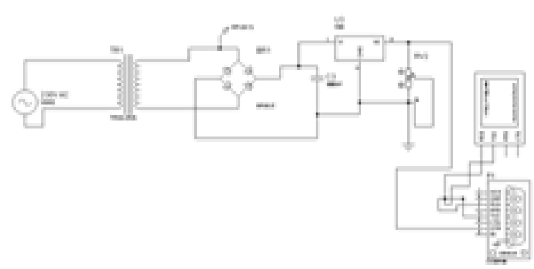
5.1.1 ARCHITRCTURE OF 16F877A



5.1.2 MEMORY ORGANIZATION

There are two memory blocks in Memory. Program memory and Data memory. Each block has its own bus, so that access to each block can occur during the same oscillator cycle. The data memory can further be broken down into General Purpose RAM and the Special Function Registers (SFRs). The operations of the SFRs that control the “core” are described here. The SFRs used to control the peripheral modules are described in the section discussing each individual peripheral module.

5.1.3 CIRCUIT DIAGRAM



5.2 POWER SUPPLY

The ac voltage, typically 120 V rms, is connected to a transformer, which steps that ac voltage down to the level for the desired dc output. A diode rectifier then provides a full-wave rectified voltage that is initially filtered by a simple capacitor filter to produce a dc voltage. This resulting dc voltage usually has some ripple or ac voltage variation. A regulator circuit can use this dc input to provide a dc voltage that not only has much less ripple voltage.

5.3 HEART BEAT SENSOR

The IR based heart beat pulse sensor gives the output of low and high Pulse. Heart beat sensor is designed to give digital output of heat beat when a finger is inserted between the IR Transmitter and Receiver. When the heart beat detector is working, the beat LED flashes with each heartbeat. This digital output can be connected to microcontroller directly to measure the Beats per Minute (BPM) rate. It works on the principle of light modulation by blood flow through finger at each pulse.

5.4 TEMPERATURE SENSOR

The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of ±1/4°C at room temperature and ±3/4°Cover a full -55 to +150°C temperature range. Low cost is assured by trimming and calibration at the wafer level. The LM35’s low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only 60 µA from its supply, it has very low self-heating, less than 0.1°C in still air.

5.5 ZIGBEE

ZigBee is a specification for a suite of high level communication protocols using small, low-power digital radios based on an IEEE 802 standard for personal area networks. ZigBee devices are often used in mesh network form to transmit data over longer distances, passing data through intermediate devices to reach more distant ones. This allows ZigBee networks to be formed ad-hoc, with no centralized control or high-power transmitter/receiver able to reach all of the devices. Any ZigBee device can be tasked with running the network. ZigBee is targeted at applications that require a low data rate, long battery life, and secure networking. ZigBee has a defined rate of 250 kbit/s, best suited for periodic or intermittent data or a single signal transmission from a sensor or input device. Applications include wireless light switches, electrical meters with in-home-displays, traffic management systems, and other consumer and industrial equipment that requires short-range wireless transfer of data at relatively low rates. The technology defined by the ZigBee specification is intended to be simpler and less expensive than other WPANs, such as Bluetooth or Wi-Fi. ZigBee operates in the industrial, scientific and medical (ISM) radio bands; 868 MHz in Europe, 915 MHz in the USA and Australia and 2.4 GHz in most jurisdictions worldwide.

VI. SOFTWARE DESCRIPTION

6.1 MPLAB IDE

MPLAB IDE is a software program that runs on a PC to develop applications for Microchip microcontrollers. It is called an Integrated Development Environment, or IDE, because it provides a single integrated "environment" to develop code for embedded microcontrollers. The rest of this chapter briefly explains embedded systems development and how MPLAB IDE is used.

A development system for embedded controllers is a system of programs running on a desktop PC to help write, edit, debug and program code - the intelligence of embedded systems applications - into a microcontroller. MPLAB IDE runs on a PC and contains all the components needed to design and deploy embedded systems applications.

TASKS FOR DEVELOPING AN EMBEDDED CONTROLLER APPLICATION ARE:

Create the high level design. From the features and performance desired, decide which PICmicro MCU or dSPIC DSC device is best suited to the application, then design the associated hardware circuitry. After determining which peripherals and pins control the hardware, write the firmware - the software that will control the hardware aspects of the embedded application. A language tool such as an assembler, which is directly translatable into machine code, or a compiler that allows a more natural language for creating programs, should be used to write and edit code. Assemblers and compilers help make the code understandable, allowing function labels to identify code routines with variables that have names associated with their use, and with constructs that help organize the code in a maintainable structure.

1. Compile, assemble and link the software using the assembler and/or compiler and linker to convert your code into "ones and zeroes" - machine code for the PICmicro MCUs. This machine code will eventually become the firmware (the code programmed into the microcontroller).
2. Test your code. Usually a complex program does not work exactly the way imagined, and "bugs" need to be removed from the design to get proper results. The debugger allows you to see the "ones and zeroes" execute, related to the source code you wrote, with the symbols and function names from your program. Debugging allows you to experiment with your code to see the value of variables at various points in the program, and to do "what if" checks, changing variable values and stepping through routines.

6.1.1 TUTORIAL FOR MPLAB IDE

MPLAB Integrated Development Environment (IDE) is a comprehensive editor, project manager and design desktop for application development of embedded designs using Microchip PICmicro MCUs and dsPIC DSCs.

The initial use of MPLAB IDE is covered here. How to make projects, edit code and test an application will be the subject of a short tutorial. By going through the tutorial, the basic concepts of the Project Manager, Editor and Debugger can be quickly learned. The complete feature set of MPLAB IDE is covered in later chapters.

This section details the installation and uninstall of MPLAB IDE. It is followed by a simple step-by-step tutorial that creates a project and explains the elementary debug capabilities of MPLAB IDE. Someone unfamiliar with MPLAB IDE will get a basic understanding of using the system to develop an application. No previous knowledge is assumed, and comprehensive technical details of MPLAB IDE and its components are omitted in order to present the basic framework for using MPLAB IDE.

6.1.2 COMPONENTS OF MPLAB IDE

The MPLAB IDE has both built-in components and plug-in modules to configure the system for a variety of software and hardware tools.

- MPLAB IDE Built-In Components
- Additional Optional Components for MPLAB IDE

1) MPLAB IDE BUILT-IN COMPONENTS

The built-in components consist of:

PROJECT MANAGER:

The project manager provides integration and communication between the IDE and the language tools.

EDITOR:

The editor is a full-featured programmer's text editor that also serves as a window into the debugger.

ASSEMBLER/LINKER AND LANGUAGE TOOLS:

The assembler can be used stand-alone to assemble a single file, or can be used with the linker to build a project from separate source files, libraries and recompiled objects. The linker is responsible for positioning the compiled code into memory areas of the target microcontroller.

DEBUGGER:

The Microchip debugger allows breakpoints, single stepping, watch windows and all the features of a modern debugger for the MPLAB IDE. It works in conjunction with the editor to reference information from the target being debugged back to the source code.

EXECUTION ENGINES:

There are software simulators in MPLAB IDE for all PICmicro MCU and dsPIC DSC devices. These simulators use the PC to simulate the instructions and some peripheral functions of the PICmicro MCU and dsPIC DSC devices. Optional in-circuit emulators and in-circuit debuggers are also available to test code as it runs in the applications of hardware.

COMPILER LANGUAGE TOOLS:

MPLAB C18 and MPLAB C30 C compilers from Microchip provide fully integrated, optimized code. Along with compilers from HI-TECH, IAR, micro Engineering Labs, CCS and Byte Craft, they are invoked by the MPLAB IDE project manager to compile code that is automatically loaded into the target debugger for instant testing and verification.

6.1.3 PROGRAMMERS

PICSTART Plus, PICkit 1 and 2, PRO MATE II, MPLAB PM3 as well as MPLAB ICD 2 can program code into target devices. MPLAB IDE offers full control over programming both code and data, as well as the Configuration bits to set the various operating modes of the target microcontrollers or digital signal controllers.

6.1.4 MPLAB IDE FEATURES AND INSTALLATION

MPLAB IDE is a Windows Operating System (OS) based Integrated Development Environment for the PIC micro MCU families and the ds PIC Digital Signal Controllers. The MPLAB IDE provides the ability to:

- Create and edit source code using the built-in editor.
- Assemble, compile and link source code.
- Debug the executable logic by watching program flow with the built-in simulator or in real time with in-circuit emulators or in-circuit debuggers.
- Make timing measurements with the simulator or emulator.

The PCB, PCM and PCH are separate compilers. PCB is for 12 bit opcodes, PCM is for 14 bit opcodes and PCH is for the 16 and 18 bit PICmicro MCU. Since much is in common between the compilers both are covered in this reference manual. Features and limitations that apply to only specific controllers are indicated within. These compilers are specially designed to meet the special needs of the PICmicro MCU controllers. These tools allow developers to quickly design application software for these controllers in a highly readable high-level language. developers to quickly design application software for these controllers in a highly readable high-level language.

The compilers have some limitations when compared to a more traditional C compiler. The hardware limitations make many traditional C compilers ineffective. As an example of the limitations, the compilers will not permit pointers to constant arrays. This is due to the separate code/data segments in the PICmicro MCU hardware and the inability to treat ROM areas as data. On the other hand, the compiler has knowledge about the hardware limitations and does the work of deciding how to best implement your algorithms.

VII. SIMULATIONRESULT

This research work investigates the potential of `Fully Automated Ambulance', which is the aim of the extends the distance in near future. The analysis and implementation of the automated ambulance system technology using Zigbee protocol to communicate between the ambulance system and the control system such as measuring the Heart Beat rate and the Temperature using heart beat sensor and the temperature sensor and that can be communicated using Zigbee protocol is presented in this paper. The proposed research work is focused on functionality of the Zigbee protocol, which allows the doctors to communicate with the ambulance system away from a particular distance. This system provides early detection of heart attacks by alerting hospital doctors in case a patient’s heart rate, body temperature is detected as abnormal also it is having provision to set the ranges depending on the age group of the patient and it will alarm if critical ranges have been achieved. Eliminates delay in receiving medical treatment as patient can be monitored remotely. Improves balance services to at risk population. Saves lives and improves quality of living.

VIII. CONCLUSION

Thus the project is to monitor and detect the patient's real-time body temperature, heart rate information's, and transmit them to the concerned doctor. Now a day with the increase of biomedical sensor we are going into this process of detecting the patient's real-time body temperature, heart rate and other physiological information's. The temperature sensor and pulse sensor module are used for acquiring medical information from the outside, and then converts them to digital signals. From the medical monitoring unit patient's temperature is taken and given to the ADC.

REFERENCES

[1] W. Y. Toh, Y. K. Tan, W. S. Koh and L. Siek, "Autonomous Wearable Sensor Nodes With Flexible Energy Harvesting," in IEEE Sensors Journal, vol. 14, no. 7, pp. 2299-2306, July 2014.

[2] Qadeer A Khan, Sarvesh J Bang."Energy Harvesting for Self Powered Wearable Health Monitoring System",2013.

[3] F. Zhang , Y. Zhang , J. Silver , Y. Shakhsheer , M. Nagaraju , A. Klinefelter , J. Pandey , J. Boley , E. Carlson , A. Shrivastava , B. Otis and B. Calhoun, "Abatteryless 19 $\mu\hbox{W}$ MICS/ISM-bandenergy harvesting body area sensor node SoC", IEEEInt. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers, pp. 298-300, 2012.

[4] Xiong Liu, Peng Wang, Poh Chiang Loh, F. Blaabjerg, " A Compact Three-Phase Single-Input/Dual-Output Matrix Converter ," IEEE Trans. on Industrial Electronics, vol. 59, no. 1, pp. 6-16, Jan 2012.

[5] V. Leonov, "Thermoelectric energy harvester on the heated human machine", J.Micromech. Microeng., vol. 21, no. 12, pp. 125013-125013, 2011.

[6] S. Sudevalayam and P. Kulkarni, "Energy Harvesting Sensor Nodes: Survey and Implications," IEEE Commun. Surveys Tuts., 2011.

[7] A. Pantelopoulos and N. G. Bourbakis, "A survey on wearable sensor-based systems for health monitoring and prognosis ", IEEE Trans. Syst., Man, Cybern. C, Appl. Rev., vol. 40, no. 1, pp. 1-12, 2010.

[8] M.Donovan , B.Bourne and J.Roche, "Efficiency VS. irradiance characterization of PV modules requires angle-of-incidence and spectral corrections",2010.

[9] G. Anastasi, M. Conti, M. D. Francesco and A. Passarella, "Energy conservation in wireless sensor networks: A survey", Ad Hoc Netw., vol. 7, no. 3, pp. 537-568, 2009

[10] T. Torfs, S. Sanders, C. Winters, S. Brebels, C. Van Hoof, "Wireless network of autonomous environmental sensors", Proceedings of IEEE Sensors 2004, Vienna, 24-27 October 2004.

[11] S. Whitehead, "Adopting wireless machine-to-machine technology", Comput. Control Eng., vol. 15, no. 5, pp. 40-46, 2004.

[12] N.Shenck,J.Paradiso,"Energy scavenging for mobile and wireless electronics," IEEE Micro, vol.21, pp.30-42, 2001.