Home Automation System Using GSM Technology

K. Thamaraiselvi[1], G. Mariselvam[2], A. Dinesh Kumar[3], R. Lavanya[4], J. Sumaiya Parveen[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 control home appliances using GSM Technology which serves for global. This paper mainly focuses on the controlling of home appliances remotely and providing security. When the user is away from the place of the system, it can be SMS based and uses wireless technology to revolutionize the standards of living.*

Keywords*-* GSM, Wireless Technology, Short Messaging Service, Mobile Phone.

I. INTRODUCTION

The aim of the paper is to investigate a cost effective solution that will provide controlling ofHome appliances remotely and will also enable home security against intrusion in the absence of home owner. The motivation is to facilitate the users to automate their homes having ubiquitous access. The system provides availability due to development of a low cost system. The home appliances control system with an affordable cost was thought to be built that should be mobile providing remote access to the appliances and allowing home security. Home security has been a major issue where crime is increasing and everybody wants to take proper measures to prevent intrusion. In addition there was a need to automate home so that user can take advantage of the technological advancement in such a way that a person getting off the office does not get melted with the hot climate. Therefore this paper proposes a system that allows user to be control home appliances ubiquitously and also provide security on detection of intrusion via SMS using GSM technology.

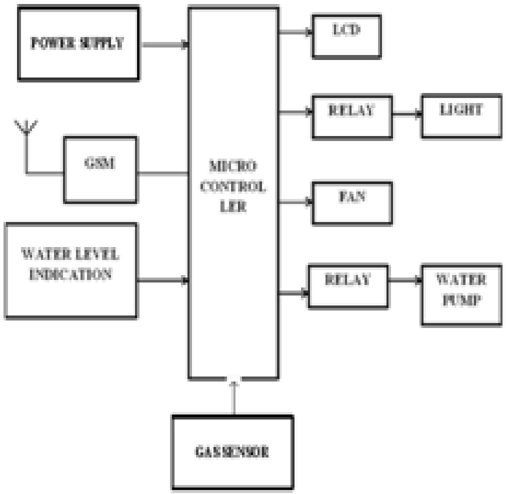
II. EXISTING SYSTEM

The design remains the existing electrical switches and provides more safety control on the switches with low voltage activating method. The switches status is synchronized in all the control system whereby every user interface indicates the real time existing switches status. Today's homes require sophistication control in its different gadgets which are basically electronic appliances. This has revolutionized the area of home automation with respect to an increased level of affordability and simplicity through the integration of home appliances with smart phone and tablet connectivity

III. PROPOSED SYSTEM

The terms "Smart Home", "Intelligent Home" followed and has been used to introduce the concept of networking appliances and devices in the house. Due to the advancement of wireless technology, there are several different of connections are introduced such as GSM, WIFI, ZIGBEE, and Bluetooth. Each of the connection has their own unique specifications and applications. Among the four popular wireless connections that often implemented in HAS project, Bluetooth is being chosen with its suitable capability. Home Automation System (HAS) has been designed for mobile phones having Android platform to automate an 8 bit Bluetooth interfaced microcontroller which controls a number of home appliances like lights, fans, bulbs and many more using on/off relay.

IV. BLOCK DIAGRAM

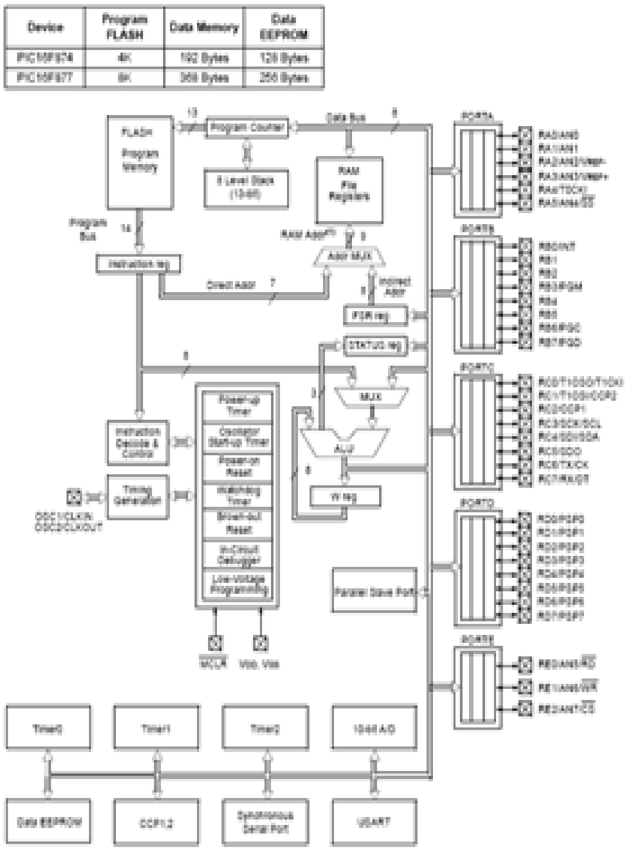


V. HARDWARE DESCRIPTION

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

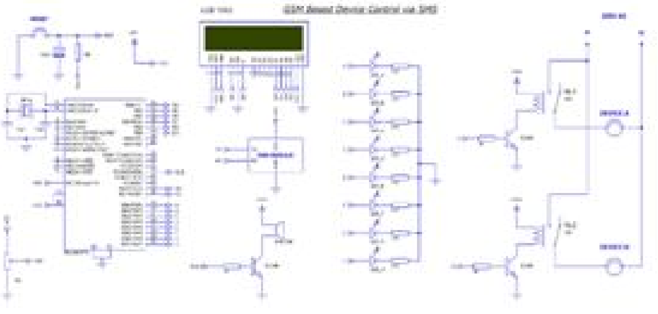


Device	Program FLASH	Data Memory	Data EEPROM
PIC16F874	4K	192 Bytes	128 Bytes
PIC16F877	8K	368 Bytes	256 Bytes

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 220V, is connected to a transformer, which steps that ac voltage down to the level of 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.

5.3 GSM MODEM

A GSM modem is a wireless modem that works with a GSM wireless network. A wireless modem behaves like a dial-up modem. The main difference between them is that a dial-up modem sends and receives data through a fixed telephone line while a wireless modem sends and receives data through radio waves. A GSM modem can be an external device or a PC Card / PCMCIA Card. Typically, an external GSM modem is connected to a computer through a serial cable or a USB cable. A GSM modem in the form of a like a GSM mobile phone, a GSM modem requires a SIM card from a wireless carrier in order to operate. As mentioned in earlier sections of this SMS tutorial, computers use AT commands to control modems. Both GSM modems and dial-up modems support a common set of standard AT commands. You can use a GSM modem just like a dial-up modem.In addition to the standard AT commands, GSM modems support an extended set of AT commands. These extended AT commands are defined in the GSM standards. With the extended AT commands, you can do things like

- Reading, writing and deleting SMS messages.
- Sending SMS messages.
- Monitoring the signal strength.
- Monitoring the charging status and charge level of the battery.
- Reading, writing and searching phone book entries.

The number of SMS messages that can be processed by a GSM modem per minute is very low -- only about six to ten SMS messages per minute.

5.4 WATER LEVEL SENSOR

This Sensor is used to indicate the level of a water in a tank.

Description:

Water sensor brick is designed for water detection, which can be widely used in sensing the rainfall, water level, even the liquate leakage. The brick is mainly comprised of three parts an Electronic brick connector, a 1 MΩ resistor, and several lines of bare conducting wires. This sensor works by having a series of exposed traces connected to ground and interlaced between the grounded traces are the sense traces. The sensor traces have a weak pull-up resistor of 1 MΩ. The resistor will pull the sensor trace value high until a drop of water shorts the sensor trace to the grounded trace. This item can judge the water level through with a series of exposed parallel wires stitch to measure the water droplet/water size. This item can easily change the water size to analog signal, and output analog value can directly be used in the program function, then to achieve the function of water level alarm. This item have low power consumption, and high sensitivity, which are the biggest characteristics of this module.

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

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 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. SIMULATION RESULT

This research work investigates the potential of `Full Home Control', which is the aim of the Home Automation Systems in near future. The analysis and implementation of the home automation technology using Global System for Mobile Communication (GSM) modem to control home appliances such as light, conditional system, and security system via Short Message Service (SMS) text messages is presented in this paper. The proposed research work is focused on functionality of the GSM protocol, which allows the user to control the target system away from residential using the frequency bandwidths. The concept of serial communication and AT-commands has been applied towards development of the smart GSM-based home automation system. Home owners will be able to receive feedback status of any home appliances under control whether switched on or off remotely from their mobile phones. PIC16F887 microcontroller with the integration of GSM provides the smart automated house system with the desired baud rate of 9600 bps. The proposed prototype of GSM based home automation system was implemented and tested with maximum of four loads and shows the accuracy of ≥98%.

VIII. CONCLUSION

In the paper low cost, secure, ubiquitously accessible, auto-configurable, remotely controlled solution for automation of homes has been introduced. The approach discussed in the paper is novel and has achieved the target to control home appliances remotely using the SMS-based system satisfying user needs and requirements. GSM technology capable solution has proved to be controlled remotely, provide home security and is cost-effective as compared to the previously existing systems. Hence we can conclude that the required goals and objectives of HACS have been achieved. software for these controllers in a highly readable high-level language.

REFERENCES

[1] M. Chan, D. Esteve, C. Escriba, E. Campo, “A review of smart homes—Present state and future challenges”, Computer methods and programs in biomedicine, Elsevier, 9I, pp. 55-81, 2008.

[2] J. Haartsen, "BLUETOOTH—The universal radio interface for ad hoc, wireless connectivity", Ericsson Review No. 3, pp. 110-117, 1998. [3] Why Android ?, http://www.android.com/about/, last seen on October 2013.

[3] R. Llamas, R Reith, M. Shiere, “Apple Cedes Market Share in Smartphone Operating System Market as Android Surges and Windows Phone Gains, According to IDC” 7th August 2013, IDC Press.

[4] N. Sriskanthan, F. Tan, A. Karande, "Bluetooth based home automation system", Microprocessors and Microsystems, Elsevier, 26, pp. 281- 289, 2002.

[5] S. Panth, M. Jivani, “Designing Home Automation system (HAS) using Java ME for Mobile Phone”, International Journal of Electronics and Computer Science Engineering, Vol. 2 No. 02, pp. 798-807, July 2013

[6] Jawarkar, N. P., Ahmed, V., Ladhake, S. A. & Thakare, R. D., Micro-controller based Remote Monitoring using Mobile through Spoken Commands. Journal of Networks, 3(2), 58-63, 2008.

[7] Malik, S. H. K., Aihab, K. and Erum, S., SMS Based Wireless Home Appliance Control System (HACS) for Automating Appliances and Security. Issues in Informing Science and Information Technology, 6, 887-894, 2009.

[8] Ahmad, B. I., Yakubu, F., Bagiwa, M. A and Abdullahi, U. I., Remote Home Management: An alternative for working at home while away. World of Computer Science and Information Technology Journal (WCSIT), 1, 4, 144-147, 2011.

[9] Mohamed Salman and Jayavrinda Vrindavanam, “Efficient Interactive Control System based on GSM”, International Journal of Latest Trends in Engineering and Technology (IJLTET), Vol. 3 No. 2, pp. 50=56, November 2013

[10] Mahesh Jivani and Sharon Panth, “Home Automation through Bluetooth”, Electronics For You, Vol. 2. Issue 7, pp. 99-104, ISSN 0013-5-16X, November 2013.

[11] Sharon Panth and Mahesh Jivani, “Designing Home Automation system (HAS) using Java ME for Mobile Phone”, International Journal of Electronics and Computer Science Engineering (IJECSE), Vol. 2. No. 2, pp. 798-807, 2013.