



PROGRAMMABLE SWITCHING CONTROL FOR INDUSTRIAL AUTOMATION IN REPETITIVE WORKS

Mohd Sayeed¹, Syed Sayeed Ahmed², Farhan Malik Shaik³,

Owais Ahmed⁴, Mohammed Abdul Rahman Uzair⁵

¹Electrical & Electronics Engineering, Nawab Shah Alam Khan College of Engg. & Tech, Hyd, India

²Electrical & Electronics Engineering, Nawab Shah Alam Khan College of Engg. & Tech, Hyd, India

³Electrical & Electronics Engineering, Nawab Shah Alam Khan College of Engg. & Tech, Hyd, India

⁴Electrical & Electronics Engineering, Nawab Shah Alam Khan College of Engg. & Tech, Hyd, India

⁵Electrical & Electronics Engineering, Nawab Shah Alam Khan College of Engg. & Tech, Hyd, India

ABSTRACT

In this paper, we have illustrated how to switch industrial loads using a user programmable logic control device for sequential operation. This operation is generally used for repetitive nature of work. Programmable logic controllers used in industrial applications are very expensive for simple operations like sequential switching of loads. In this project we demonstrate the working of this simple operation using a microcontroller of 8052 family. The development of this application requires configuration of the program through input switches. In industries, there are many tasks are carried out which requires some repeated operation in various orders and time intervals. For example, certain loads need to be switched ON/OFF in specific time intervals. In order to achieve this, microcontroller is programmed in such a way that the loads a can be operated in three modes: Set mode, Auto mode and Manual mod. In set mode, through timers, the machinery works based on input time set by the user where as in auto mode it works on default time settings and finally in the manual mode it functions while respective switches are pressed depending on the user's need and flexibility. All the modes and status of loads are displayed on an LCD. Thus, tasks performed by costly PLCs can now be achieved using a microcontroller making the device cost effective. Further the project can be enhanced by interfacing it with a GSM modem where by sending an SMS to the control system we can select the mode and timing remotely.

Keywords: Embedded Systems, PCB, Proteus Software, Keil Software, Microcontrollers.

I. INTRODUCTION

Embedded Systems

An embedded system is a combination of software and hardware to perform a dedicated task. Some of the main devices used in embedded products are Microprocessors and Microcontrollers. Microprocessors are commonly referred to as general purpose processors as they simply accept the inputs, process it and give the output. In contrast, a microcontroller not only accepts the data as inputs but also manipulates it, interfaces the data with various devices, controls the data and thus finally gives the result.

Application specific systems

Embedded systems are not general-purpose computers. Embedded system designs are Optimized for a specific application. Many of the job characteristics are known before the hardware is designed. This allows the designer to focus on the specific design constraints of a well defined application. As such, there is limited user reprogram ability. Some embedded systems, however, require the flexibility of reprogrammability. Programmable DSPs are common for such applications.

Scope of Embedded Systems

Nearly 99 per cent of the processors manufactured end up in embedded systems. The embedded system market is one of the highest growth areas as these systems are used in very market segment- consumer electronics, office automation, industrial automation, biomedical engineering, wireless communication, data communication, telecommunications, transportation, military and so on.

Overview of Embedded System Architecture

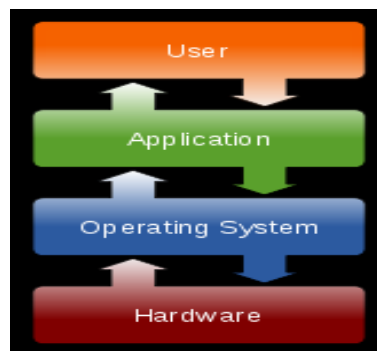


Fig 1.1.2 The Building Blocks Diagram

II. BLOCK DIAGRAM

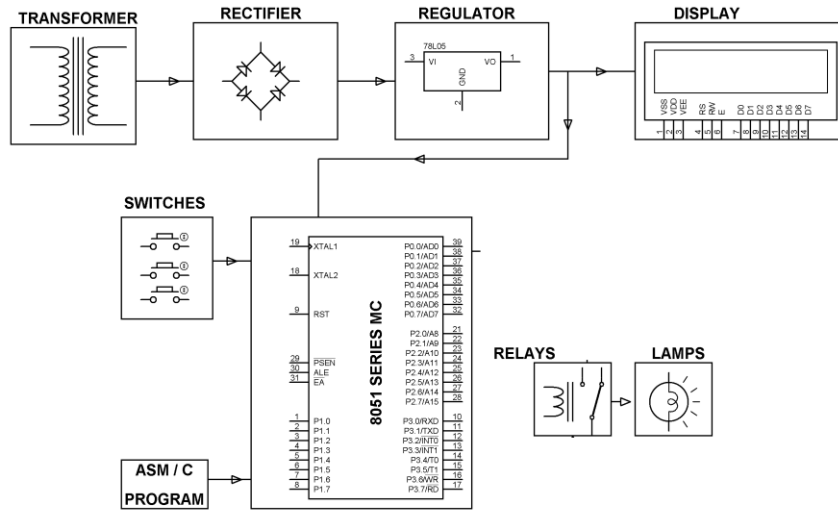


Fig 2.1 Block Diagram

CIRCUIT DIAGRAM

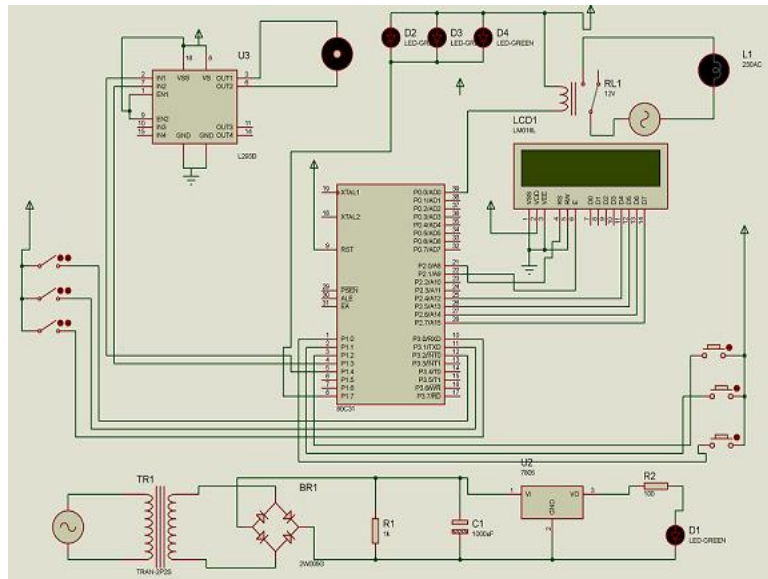


Fig 2.2 Circuit Diagram

III. HARDWARE COMPONENTS

- Power Supply
- LCD Display
- Microcontrollers
- Motors
- Relay



IV. SOFTWARE COMPONENTS

KEIL Software

Keil Compiler is software used where the machine language code is written and compiled. After compilation, the machine source code is converted into hex code which is to be dumped into the microcontroller for further processing. Keil compiler also supports C language code.

Steps to write an Assembly Language Program in KEIL and to Compile it:

1. Install the Keil Software in the PC in any of the drives.
2. After installation, an icon will be created with the name “Keil uVision3”. Just drag this icon onto the desktop so that it becomes easy whenever you try to write programs in keil.
3. Double click on this icon to start the keil compiler.
4. A page opens with different options in it showing the project workspace at the leftmost corner side, output window in the bottom and an ash colored space for the program to be written.
5. Now to start using the keil, click on the option “project”.
6. A small window opens showing the options like new project, import project, open project etc. Click on “New project”.
7. A small window with the title bar “Create new project” opens. The window asks the user to give the project name with which it should be created and the destination location. The project can be created in any of the drives available. You can create a new folder and then a new file or can create directly a new file.
8. After the file is saved in the given destination location, a window opens where a list of vendors will be displayed and you have to select the device for the target you have created.
9. The most widely used vendor is Atmel. So click on Atmel and now the family of microcontrollers manufactured by Atmel opens. You can select any one of the microcontrollers according to the requirement.
10. When you click on any one of the microcontrollers, the features of that particular microcontroller will be displayed on the right side of the page. The most appropriate microcontroller with which most of the projects can be implemented is the AT89S52. Click on this microcontroller and have a look at its features. Now click on “OK” to select this microcontroller.
11. A small window opens asking whether to copy the startup code into the file you have created just now. Just click on “No” to proceed further.
12. Now you can see the TARGET and SOURCE GROUP created in the project workspace.
13. Now click on “File” and in that “New”. A new page opens and you can start writing program in it.
14. After the program is completed, save it with any name but with the .asm extension. Save the program in the file you have created earlier.



15. You can notice that after you save the program, the predefined keywords will be highlighted in bold letters.
16. Now add this file to the target by giving a right click on the source group. A list of options open and in that select “Add files to the source group”. Check for this file where you have saved and add it.
17. Right click on the target and select the first option “Options for target”. A window opens with different options like device, target, output etc. First click on “target”.
18. Since the set frequency of the microcontroller is 11.0592 MHz to interface with the PC, just enter this frequency value in the Xtal (MHz) text area and put a tick the Use on-chip ROM. This is because the program what we write here in the keil will later be dumped into the microcontroller and will be stored in the inbuilt ROM in the microcontroller.
19. Now click the option “Output” and give any name to the hex file to be created in the “Name of executable” text area and put a tick to the “Create HEX file” option present in the same window. The hex file can be created in any of the drives. You can change the folder by clicking on “Select folder for Objects”.
20. Now to check whether the program you have written is errorless or not, click on the icon exactly below the “Open file” icon which is nothing but Build Target icon. You can even use the shortcut key F7 to compile the program written.
21. To check for the output, there are several windows like serial window, memory window, project window etc. Depending on the program you have written, select the appropriate window to see the output by entering into debug mode.
22. The icon with the letter “d” indicates the debug mode.
23. Click on this icon and now click on the option “View” and select the appropriate window to check for the output.
24. After this is done, click the icon “debug” again to come out of the debug mode.

The microcontroller program is the most important part of our project. It is written in assembly language. Following flow chart describes logic of program

PROTEUS Software

The Proteus Design Suite is a Windows application for schematic capture, simulation, and PCB layout design. It can be purchased in many configurations, depending on the size of designs being produced and the requirements for microcontroller simulation. All PCB Design products include an auto router and basic mixed mode SPICE simulation capabilities.

Schematic Capture

Schematic capture in the Proteus Design Suite is used for both the simulation of designs and as the design phase of a PCB layout project. It is therefore a core component and is included with all product configurations.

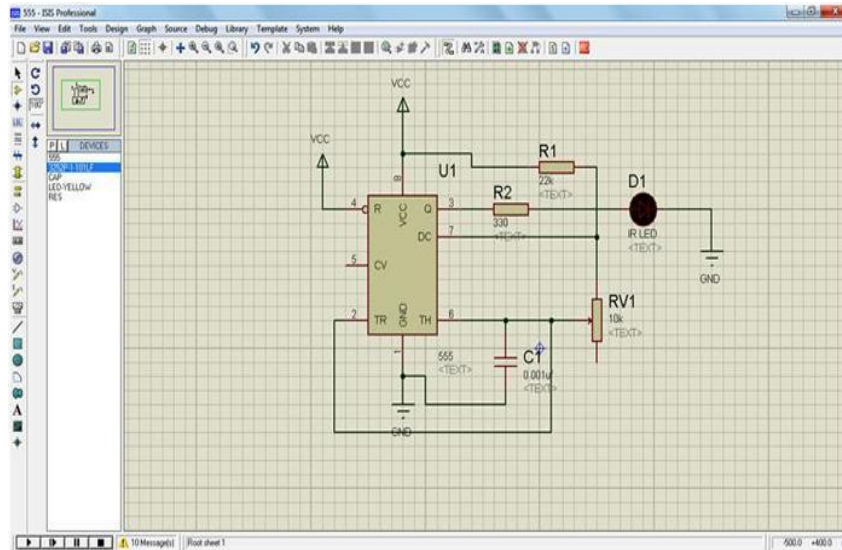


Fig 6.2 Schematic Capture

Microcontroller Simulation

The micro-controller simulation in Proteus works by applying either a hex file or a debug file to the microcontroller part on the schematic. It is then co-simulated along with any analog and digital electronics connected to it. This enables it's used in a broad spectrum of project prototyping in areas such as motor control, temperature control and user interface design. It also finds use in the general hobbyist community and, since no hardware is required, is convenient to use as training or teaching tool. Support is available for co-simulation of:

- Microchip Technologies PIC10, PIC12, PIC16, PIC18, PIC24, dsPIC33 Microcontrollers.
- Atmel AVR (and Arduino), 8051 and ARM Cortex-M3 Microcontrollers
- NXP 8051, ARM7, ARM Cortex-M0 and ARM Cortex-M3 Microcontrollers.
- Texas Instruments MSP430, PICCOLO DSP and ARM Cortex-M3 Microcontrollers.
- Parallax Basic Stamp, Freescale HC11, 8086 Microcontrollers.

V. PCB DESIGN

The PCB Layout module is automatically given connectivity information in the form of a **netlist** from the schematic capture module. It applies this information, together with the user specified **design rules** and various design automation tools, to assist with error free board design. Design Rule Checking does not include high speed design constraints. PCB's of up to 16 copper layers can be produced with design size limited by product configuration.

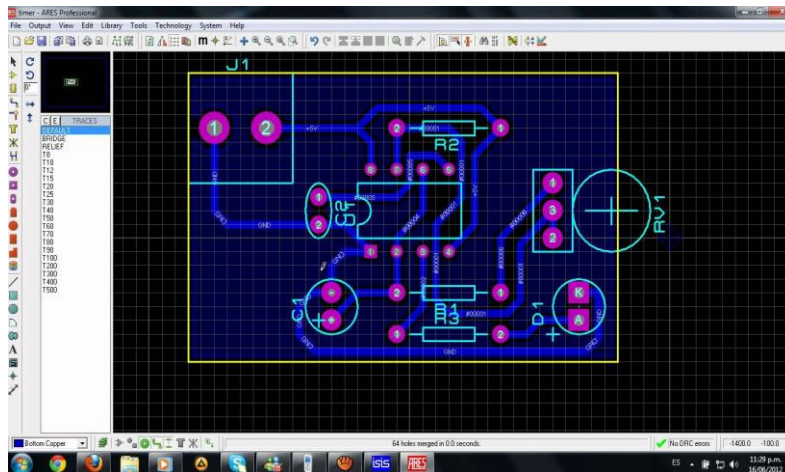


Fig 6.2.1 PCB Design

VI. 3D Verification

The 3D Viewer module allows the board under development to be viewed in 3D together with a semi-transparent height plane that represents the board's enclosure. STEP output can then be used to transfer to mechanical CAD software such as Solid works or Autodesk for accurate mounting and positioning of the board.

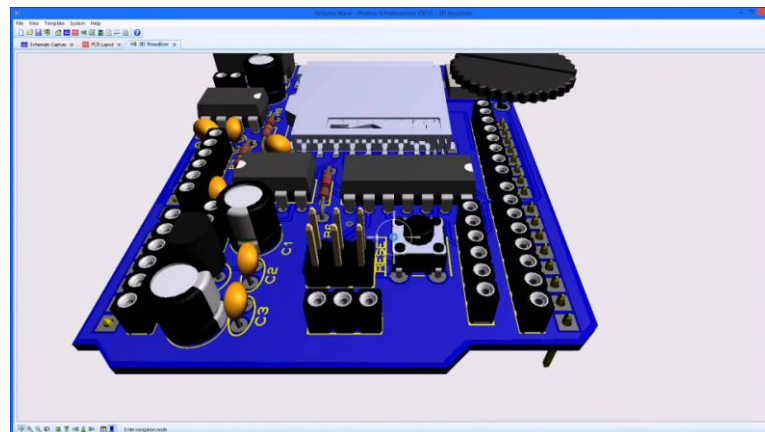


Fig 6.2.2 3D Verification

VII. ADVANTAGES

- Increased productivity (repetitive nature).
- High accuracy (due to program).
- Less human error.
- It is reliable.



VIII. APPLICATIONS

- a) It can be used in processing industries.
- b) It can be used in manufacturing industries.
- c) Automation Factories.

IX. CONCLUSION

The project “**programmable switching control for industrial automation in repetitive nature of work**” has been successfully designed and tested. It has been developed by integrating features of all the hardware components used. Presence of every module has been reasoned out and placed carefully thus contributing to the best working of the unit.

Secondly, using highly advanced ic’s and with the help of growing technology the project has been successfully implemented.

ACKNOWLEDGMENTS

- Mohd Sayeed is thankful to all the co-authors for their support.
- Syed Sayeed Ahmed is grateful to all the people who helped in coming out with this work.
- Farhan Malik Shaik expresses thanks to all the co-authors for their cooperation.
- Owais Ahmed is grateful to all the co-authors for their support.
- Mohammed Abdul Rahman Uzair is thankful to all the contributors for this work.

REFERENCES

- [1] www.arm.com
- [2] www.wikipedia.org
- [3] www.zigbee.org
- [4] www.nxp.com
- [5] www.keil.com
- [6] www.cadence.com
- [7] www.ieee.org