PROGRAMMABLE TIMER FOR SWITCHING RELAY USING BLUETOOTH

A Project report submitted in partial fulfillment
of the requirements for the degree of B. Tech in Electrical Engineering

by

AVIK SINHA (11701617065)
SYED ZEESHAN AHMAD (11701617021)
TAMAL MUKHERJEE (11701618001)
RAJIB TUDU (11701618005)

Under the supervision of

Mr. Budhaditya Biswas
Assistant Professor
Department of Electrical Engineering

श्रमम् बिना न किमपि साध्यम्

Department of Electrical Engineering
RCC INSTITUTE OF INFORMATION TECHNOLOGY
CANAL SOUTH ROAD, BELIAGHATA, KOLKATA – 700015, WEST BENGAL
Maulana Abul Kalam Azad University of Technology (MAKAUT)


COVID-19 Prevention
Wearing Face Mask
Washing hands
Eat hot foods
2 Meters
6 Feet
Social Distancing
Departure
Avoid risk places
Self quarantine


WEAR MASK
SAVE LIVES

Department of Electrical Engineering
RCC INSTITUTE OF INFORMATION TECHNOLOGY
CANAL SOUTH ROAD, BELIAGHATA, KOLKATA – 700015, WEST BENGAL
PHONE: 033-2323-2463-154, FAX: 033-2323-4668
Email: hodeercciit@gmail.com, Website: http://www.rcciit.org/academic/ee.aspx

CERTIFICATE

To whom it may concern

This is to certify that the project work entitled **Programmable Timer for Switching Relay Using Bluetooth** is the bonafide work carried out by **Avik Sinha (11701617065), Syed Zeeshan Ahmad (11701617021), Tamal Mukherjee (11701618001), Rajib Tudu (11701618005),** the students of B.Tech in the Department of Electrical Engineering, RCC Institute of Information Technology (RCCIIT), Canal South Road, Beliaghata, Kolkata-700015, affiliated to Maulana Abul Kalam Azad University of Technology (MAKAUT), West Bengal, India, during the academic year 2020-21, in partial fulfillment of the requirements for the degree of Bachelor of Technology in Electrical Engineering and that this project has not submitted previously for the award of any other degree, diploma and fellowship.

(Budhaditya Biswas)
Assistant Professor
Department of Electrical Engineering
RCC Institute of Information Technology

Countersigned by

HoD, Dept. of EE
RCC Institute of Information Technology
Kolkata-700015

(Dr. Debasish Mondal)
HOD, Electrical Engineering Dept
RCC Institute of Information Technology

ACKNOWLEDGEMENT

It is our great fortune that we have got opportunity to carry out this project work under the supervision of **Mr. Budhaditya Biswas** in the Department of Electrical Engineering, RCC Institute of Information Technology (RCCIIT), Canal South Road, Beliaghata, Kolkata-700015, affiliated to Maulana Abul Kalam Azad University of Technology (**MAKAUT**), West Bengal, India. We express our sincere thanks and deepest sense of gratitude to our guide for his constant support, unparalleled guidance and limitless encouragement.

We would also like to convey our gratitude to all the faculty members and staffs of the Department of Electrical Engineering, RCCIIT for their whole hearted cooperation to make this work turn into reality.

We are very thankful to our department and to the authority of RCCIIT for providing all kinds of infrastructural facility towards the research work.

Thanks to the fellow members of our group for working as a team.

Avik Sinha (11701617065)

Syed Zeeshan Ahmad (11701617021)

Tamal Mukherjee (11701618001)

Rajib Tudu (11701618005)

To

The Head of the Department
Department of Electrical Engineering
RCC Institute of Information Technology
Canal South Rd. Beliaghata, Kolkata-700015

Respected Sir,

In accordance with the requirements of the degree of Bachelor of Technology in the Department of Electrical Engineering, RCC Institute of Information Technology, We present the following thesis entitled "**PROGRAMMABLE TIMER FOR SWITCHING RELAY USING BLUETOOTH**". This work was performed under the valuable guidance of Mr. Budhaditya Biswas, Assistant Professor in the Dept. of Electrical Engineering.

We declare that the thesis submitted is our own, expected as acknowledge in the test and reference and has not been previously submitted for a degree in any other Institution.

Yours Sincerely,

Avik Sinha (11701617065)

Syed Zeeshan Ahmad (11701617021)

Tamal Mukherjee (11701618001)

Rajib Tudu (11701618005)

Contents

List of Figures

List of Tables

ABBREVIATIONS AND ACRONYMS

BT – Bluetooth
HVAC – Heating Ventilation and Air Conditioning
IC - Integrated Circuit
PCB – Printed Circuit Board
µC – Micro Controller
BJT - Bi-polar Junction Transistor
SPDT - Single Pole Double Throw
NO - Normally Open
NC - Normally Closed
COM – Common
LCD – Liquid Crystal Display
LED - Light Emitting Diode
POT – Potentiometer
AT – Attention Command
SMPS – Switch Mode Power Supply
RF – ***R***adio Frequency
ISM – Industrial, scientific and medical
USB – Universal serial bus
SPI – Serial Peripheral Interface
I^2C **–** Inter-Integrated Circuit

ABSTRACT

Programmable Relay Control (PRC) using 8051 microcontroller is a simple microcontroller based project that performs a sequential switching of load with programmable timings.

PLCs have their use in many areas, including basic relay control, motor or motion control, industrial process control and complex networking. PLCs are often very expensive because of their ability to handle very complex process control, high speed and precision.

This project deals with an interesting manner of achieving sequential switching of loads in repetitive nature of work with the help of 8051 microcontroller. A simple microcontroller based PLC like functionality is achieved where different loads are switched either according to the pre-programmed timings or timings set during run time.

The advantage of this project is that the setting can be done remotely using HC05 Bluetooth transceiver which is interfaced with the microcontroller. The settings for the relays are shown in a 16x2 LCD module which is also connected to the controller board. The setting is also shown remotely through the Bluetooth serial communication. This whole process can be fully controlled as per requirement from a distance using mobile or computer wirelessly. For that reason, the requirement of wires and other controlling devices will be less and the system will be economical. The power requirement will also be less here.

The applications of the project are in Industrial Automation and process control. But as it doesn't have the capability and redundancy to handle complex programs like PLCs, it can be successfully implemented for smaller industrial tasks.

CHAPTER 1

(Introduction)

INTRODUCTION

The aim of the proposed system is to develop a cost effective solution that will provide sequential control of mainly industrial loads remotely and enable Bluetooth controlled time setting of those loads from any electronics device with Bluetooth connectivity. So, this system will provide a distant, sequential, text based switching of loads for process industries mainly. Idea of this project is based on the process industrial control system with an affordable cost that should be mobile providing remote access to the sequential control of multiple loads. These loads should be controlled by turn on/off according to set time. Now this setting of time is done and can be changed by administrator at any time using his/her mobile device. It is a necessity to control that is to set the timing of the timers for the loads more effectively and efficiently at anytime from anywhere.
In this system, we are going to develop Bluetooth controlled timers to on and off the loads through relay. This system is designed for controlling relay devices connected with some specified loads. So, it includes an android device (end user has to connect his/her android device to the system) which is connected to the system via Bluetooth. When a signal is transmitted from a Bluetooth device, the paired **HC 05** module receive the signal. This signal is transmitted to the microcontroller through serial communication at 9600 baud rate. Microcontroller process the data. As per the set time of the timers, microcontroller sends trigger and actuates the relay driver. The relay driver drives the relay to switch on/off the sequential electrical loads. The controller also provides some feedback to ensure that the load switching has been done properly.The device switching is achieved by Relays. For instance, our system contains an alarm unit giving the user a remote on/off mechanism.

1.1 SEQUENTIAL PROCESS CONTROL

A **Process Control** in continuous industrial production processes is a discipline that uses industrial control systems to achieve a production level of consistency, economy and safety which could not be achieved purely by human manual control. It is implemented widely in industries such as automotive, mining, dredging, oil refining, pulp and paper manufacturing, chemical processing

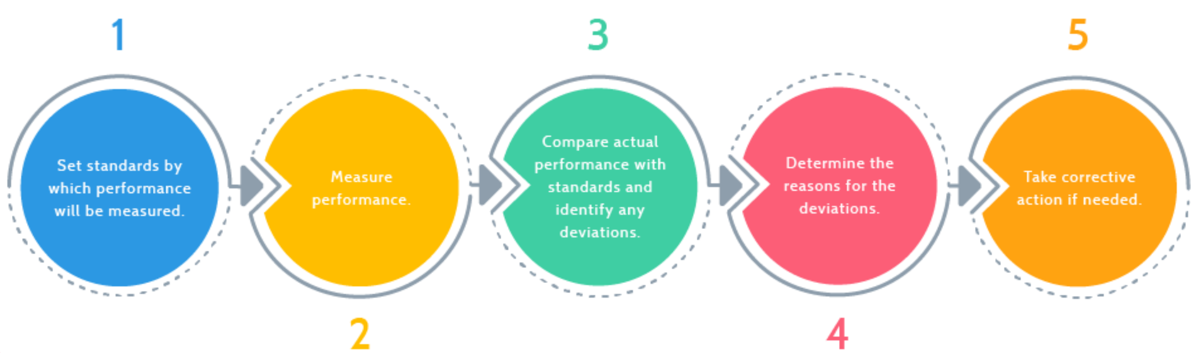


Figure 1: Concept of Industrial Process Control

and power generating plants. Now a control system in which the individual steps are processed in a predetermined order, progression from one sequence step to the next being dependent on defined conditions being satisfied, is a **Sequential Process Control**. Such a system is generally time-controlled, in which the step transition conditions are functions of time only or it may also be external-event dependent, where the conditions are functions of Input signals only or it also may be combinations of these (and perhaps more complex) conditions.

Early process control breakthroughs came most frequently in the form of water control devices. Later process controls inventions involved basic physics principles. With the dawn of the Industrial

Revolution in the 1760s, process controls inventions were aimed to replace human operators with mechanized processes.

Process control of large industrial plants has evolved through many stages. Initially, control would be from panels local to the process plant. However, this required a large manpower resource to attend to these dispersed panels, and there was no overall view of the process. The next logical development was the transmission of all plant measurements to a permanently-manned central control room. Effectively this was the centralization of all the localized panels, with the advantages of lower manning levels and easier overview of the process. With the coming of electronic processors, graphic displays and different communication protocols it became possible to replace these discrete controllers with computer-based algorithms, hosted on a network of input/output racks with their own control processors and also to replace wired control with wireless system.

Now generally process control is associated with industrial process industries where batch control of repeated processes is required. A great example of process control is temperature control during any industrial process. Different temperatures are maintained for different times spans according to settings. This also can be done manually but that will require large man power and time and will consist of heavy inaccuracy. Using this microcontroller based automated system, this can be done effective from anywhere at any time with great accuracy.

1.2 Overview and benefits

Maintained measurement and control in manufacturing processes helps facilitate a business' overall success. That's easier said than done, though. Overseeing the regulation of a large variety of processes can be extremely overwhelming. That's where the implementation of process control instrumentation comes in.

Effectively using the process control technologies at our fingertips will allow the business' manufacturing processes to achieve the measurable growth we've been waiting for. But it takes a lot of measurement skills and overall control to comprehensively understand the variables within the manufacturing area. The successful implementation of process control instrumentation, however, will facilitate the deliverance of quality products. And it'll allow the business to remain profitable.

Although process control technology has advanced rapidly since the mid-1980s, the latest systems still follow the traditional hierarchical or pyramid-like structure. The lowest level of the pyramid works to make sure a particular process doesn't vary by more than an allowed amount. It monitors the operation of each part of the process, identifies unwanted changes and initiates any necessary corrective actions. Lower level controls can't handle complex situations like equipment faults. These have to be dealt with either manually, by an operator, or by other controls at a higher level of the hierarchy. Further up the pyramid the system controls the overall production process and makes sure it continues to operate efficiently.

Business could benefit from a well-designed control system in many ways.

Measurements are one of the most important parts in a processing plant. Using the proper process control instrumentation to remodel and rework your internal operations allows your machines to reduce variability and run to the best of their abilities. Additionally, it'll keep the employees well-rested, level-headed and, most importantly, excited to come to work every day. By simple eliminating unnecessary machinery, or even physical labour, there will be more time and space for the business to grow.

One of the biggest benefits of the process control industry is automated efficiency. In fact, it's possible that after implementation of process control instrumentation and reworking of operations, there will be less of a need for existing machine or human labor. And there will be more income to go around, which means an increase in general performance.

Process control systems are central to maintaining product quality. Using proper instrumentation, control systems maintain the proper ratio of ingredients. Without this standard of control, products would vary and quality would be impaired. With improved quality comes higher levels of safety too. The process control systems automatically warn you of any abnormalities which minimizes the risk of accidents. By shifting focus to cost-effective and objective-reaching technologies, the ability to take on more work will increase significantly.

1.3 Organisation of thesis

The thesis is organised into five chapters including the chapter of introduction. Each chapter is different from the other and is described along with the necessary theory required to comprehend it.

Chapter 2 deals with the literature reviews. From this chapter we can see before our project who else works on this topic and how our project is different and advance from those projects.

Chapter 3 deals with the theory required to do the project. The basic of serial communication with microcontroller at 9600 baud rate and the communication with HC 05 are described here. The overview of the project and software simulation of the project is also listed in this chapter.

Chapter 4 deals with the hardware modelling of the projects. The main features, photographs, step by step operation of the prototype, component listing and the hardware interfacing of the required components are described here.

Chapter 5 describes the basic operation of the circuit. A flow chart is presented on the actions that would take in the controller beginning from the pairing of the android device with the Bluetooth module HC 05 to the switching on and off of loads. Advantages and disadvantages and cost estimation are listed in this chapter.

Chapter 6 concludes the work performed so far. The possible limitations in proceeding research towards this work are discussed. The future work that can be done in improving the current scenario is mentioned. The future potential along the lines of this work is also discussed.

Chapter 7 References are listed in this chapter

Appendix A, B & C Hardware description, software coding and datasheets are listed here.

CHAPTER 2

(Literature Review)

The system [1] proposes Design and Development of Microcontroller Based Programmable Timer for Supply Control. Here a microcontroller based programmable timer with digital display system is developed that can be configured to connect the AC main line to the instruments for a specific time interval.

The system proposed in [2] describes and prototypes an Industrial Appliances Control Using Android Mobile & Bluetooth Technology. Although sequential control is not implemented here, but control of the speed and direction of DC motor using wireless Bluetooth Technology along with temperature and light intensity monitor and control is implemented here using 18F4520 microcontroller.

The system [3] proposes Development of a Simple Programmable Control Timer. It is a simple counter based circuit. Using logic-control unit, timer is set and after timer count is over, using a zero detection circuit, trigger is given to the output relay. The device uniqueness is in its ability to switch ON an initially OFF appliance and at the same time switch OFF another initially ON appliance connected to it after a preset time.

The system [4] proposes a time controlled temperature regulation with sab 0529 programmable timer IC. The temperature regulating circuit of a special iron press (70 degree C) is described as an example to demonstrate a useful electronic technique for time-regulated temperature control. If these iron presses are used, the ironing process is finished after 30 minutes. The appliance automatically switches off. Wherever safety and energy saving are important, many other applications are possible - time-controlled temperature regulation for heating of plates, ovens, sauna equipment, swimming pools, hobby rooms, bathrooms and freezers (automatic deep freeze control), to name just a few.

The system proposed in [5] describes a Time Switching System Using Atmega328 Microcontroller towards Solving Problem of Electrical Power Wastage. The core technologies of the developed system involved the application of a microcontroller to control the switching times of any electrical appliances as desired by the users. There is a power supply to the microcontroller and instructions based on the user's times set, are executed by the controller through corresponding signals to the activation module where the electrical appliance was connected. The relay in the activation module is either activated or de-activated to switch ON or OFF of the appliance. As user pre-set or pre-defined the times for ON and OFF, the instructions are stored in the memory of the microcontroller for corresponding actions. The research is focused on designing and implementation of a time switching system using the Atmega328 microcontroller

The system proposed in [6] presents a multipurpose SMS activated programmable timer switch. The study aimed to design and develop a Multi-Purpose SMS Activated Programmable Timer Switch. Specifically, it utilized the mobile phone short messaging service (SMS) to activate the timer switch to the desired time in minutes. The device replies to the user the time set, which is equivalent to the time requested or texted by the user and the status of the switch if it is ON or OFF. Likewise, the user can interrupt the time or switch OFF the device by just texting STOP. Further, the user can inquire of the status if it is ON or OFF at the moment or the remaining time of the device by means of texting STATUS. Main components used here are Arduino Uno, GSM module and relay module.

In system [7] a Programmable Timer for Repeated Work is created. It mainly consists of a key pad, micro controller unit, drivers and relays and LCD display. The user can set any of the time given in the timer switch using key pad. The settings are stored in the micro controller. The details are displayed in the LCD display. The timer switch controls the appliance to run for the particular period. If the particular period is over, the timer switch automatically makes the appliance off. Four relays are used to drive output Load. It will provide fixed time for user (Not programmable).

The prototype which we developed as our project has some of the features described above. Apart from that our system has distant control using Bluetooth(As there are a large set of software and apps available or can be made as per our requirements for Bluetooth connectivity for mobile phones as well as laptops or any other electronic mobile devices. So visual control can be developed for future development. Except for that, Bluetooth connectivity is cheap, easy and quite better than some of the others). Our system is fully programmable (User can set timers of each relay separately).

Besides that, some measures have been taken to make our system more economic for same type of other industrial automation. We also have on spot LCD display for load status display. Also, our system can be controlled using any Bluetooth enabled mobile device. The prototype also has the features like ‘text mode control’, load status sharing to the administrator’s mobile devices etc.

CHAPTER 3

(Theory)

3.1 Basic of Serial Communication

The asynchronous serial protocol has a number of built-in rules - mechanisms that help ensure robust and error-free data transfers. These mechanisms, which we get for eschewing the external clock signal, are:

- Data bits,
- Synchronization bits,
- Parity bits,
- and Baud rate.

Through the variety of these signaling mechanisms, you'll find that there's no one way to send data serially. The protocol is highly configurable. The critical part is making sure that **both devices on a serial bus are configured to use the exact same protocols**.

3.1.1 Baud Rate

The baud rate specifies **how fast** data is sent over a serial line. It's usually expressed in units of bits-per-second (bps). If you invert the baud rate, you can find out just how long it takes to transmit a single bit. This value determines how long the transmitter holds a serial line high/low or at what period the receiving device samples its line.

Baud rates can be just about any value within reason. The only requirement is that both devices operate at the same rate. One of the more common baud rates, especially for simple stuff where speed isn't critical, is **9600 bps**. Other "standard" baud are 1200, 2400, 4800, 19200, 38400, 57600, and 115200.

The higher a baud rate goes, the faster data is sent/received, but there are limits to how fast data can be transferred. You usually won't see speeds exceeding 115200 - that's fast for most microcontrollers. Get too high, and you'll begin to see errors on the receiving end, as clocks and sampling periods just can't keep up.

3.1.2 Framing the data

Each block (usually a byte) of data transmitted is actually sent in a *packet* or *frame* of bits. Frames are created by appending synchronization and parity bits to our data.

Figure 2: A serial frame.

Some symbols in the frame have configurable bit sizes. Let's get into the details of each of these frame pieces.

3.1.3 Data chunk

The real meat of every serial packet is the data it carries. We ambiguously call this block of data a *chunk*, because its size isn't specifically stated. The amount of data in each packet can be set to anything from 5 to 9 bits. Certainly, the standard data size is your basic 8-bit byte, but other sizes have their uses. A 7-bit data chunk can be more efficient than 8, especially if you're just transferring 7-bit ASCII characters.

After agreeing on a character-length, both serial devices also have to agree on the **endianness** of their data. Is data sent most-significant bit (msb) to least, or vice-versa? If it's not otherwise stated, you can usually assume that data is transferred **least-significant bit (lsb) first**.

3.1.4 Synchronization bits

The synchronization bits are two or three special bits transferred with each chunk of data. They are the **start bit** and the **stop bit(s)**. True to their name, these bits mark the beginning and end of a packet. There's always only one start bit, but the number of stop bits is configurable to either one or two (though it's commonly left at one).

The start bit is always indicated by an idle data line going from 1 to 0, while the stop bit(s) will transition back to the idle state by holding the line at 1.

3.1.5 Parity bits

Parity is a form of very simple, low-level error checking. It comes in two flavors: odd or even. To produce the parity bit, all 5-9 bits of the data byte are added up, and the evenness of the sum decides whether the bit is set or not. For example, assuming parity is set to even and was being added to a data byte like `0b01011101`, which has an odd number of `1`'s (5), the parity bit would be set to `1`. Conversely, if the parity mode was set to odd, the parity bit would be `0`.
Parity is *optional*, and not very widely used. It can be helpful for transmitting across noisy mediums, but it'll also slow down your data transfer a bit and requires both sender and receiver to implement error-handling (usually, received data that fails must be re-sent).

3.1.6 9600 baud rate communication using AT89c51

9600 8N1 - 9600 baud, 8 data bits, no parity, and 1 stop bit - is one of the more commonly used serial protocols. So, what would a packet or two of 9600 8N1 data look like? Let's have an example

A device transmitting the ASCII characters 'O' and 'K' would have to create two packets of data. The ASCII value of *O* (that's uppercase) is 79, which breaks down into an 8-bit binary value of `01001111`, while *K*'s binary value is `01001011`. All that's left is appending sync bits.
It isn't specifically stated, but it's assumed that data is transferred least-significant bit first. Notice how each of the two bytes is sent as it reads from right-to-left.

Figure 3: 9600 baud rate bit pattern

Since we're transferring at 9600 bps, the time spent holding each of those bits high or low is 1/(9600 bps) or 104 µs per bit.

For every byte of data transmitted, there are actually 10 bits being sent: a start bit, 8 data bits, and a stop bit. So, at 9600 bps, we're actually sending 9600 bits per second or 960 (9600/10) bytes per second.

Now that you know how to construct serial packets, we can move on to the hardware section. There we'll see how those 1's and 0's and the baud rate are implemented at a signal level.

3.2 Serial Communication using HC05

Bluetooth is a **standardized protocol** for sending and receiving data via a 2.4GHz wireless link. It's a secure protocol, and it's perfect for short-range, low-power, low-cost, wireless transmissions between electronic devices.

These days it feels like *everything* is wireless, and Bluetooth is a big part of that wireless revolution. You'll find Bluetooth embedded into a great variety of consumer products, like headsets, video game controllers, or (of course) livestock trackers.

In our world of embedded electronics, Bluetooth serves as an excellent protocol for wirelessly transmitting relatively small amounts of data over a short range (<100m). It's perfectly suited as a wireless replacement for serial communication interfaces. Or you can use it to create a DIY HID Computer Keyboard. Or, with the right module, it can be used to build a homebrew, wireless MP3-playing speaker.

This tutorial aims to provide a quick overview of the Bluetooth protocol. We'll examine the specifications and profiles that form its foundation, and we'll go over how Bluetooth compares to other wireless protocols.

3.2.1 How Bluetooth Works

The Bluetooth protocol operates at 2.4GHz in the same unlicensed ISM frequency band where RF protocols like ZigBee and WiFi also exist. There is a standardized set of rules and specifications that differentiates it from other protocols. If you have a few hours to kill and want to learn every nook and cranny of Bluetooth, check out the published specifications, otherwise here's a quick overview of what makes Bluetooth special.

3.2.2 Masters, Slaves, and Piconets

Bluetooth networks (commonly referred to as **piconets**) use a master/slave model to control when and where devices can send data. In this model, a single master device can be connected to up to seven different slave devices. Any slave device in the piconet can only be connected to a single master.

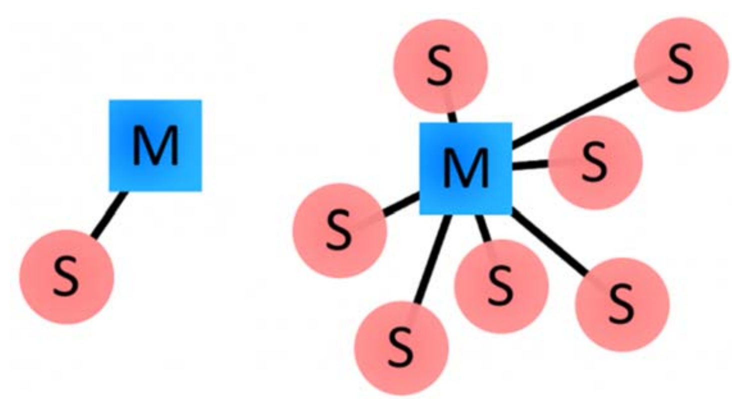


Figure 4: Examples of Bluetooth master/slave piconet topologies.

The master coordinates communication throughout the piconet. It can send data to any of its slaves and request data from them as well. Slaves are only allowed to transmit to and receive from their master. They can't talk to other slaves in the piconet.

3.2.3 Wireless Comparison

Bluetooth is far from the only wireless protocol out there. You might be reading this tutorial over a WiFi network. Or maybe you've even played with ZigBees or XBees. So what makes Bluetooth different from the rest of the wireless data transmission protocols out there?

Let's compare and contrast. We'll include BLE as a separate entity from *Classic* Bluetooth.

Table 1: wireless protocol comparison

Name	Bluetooth Classic	Bluetooth 4.0 Low Energy (BLE)	ZigBee	WiFi
IEEE Standard	802.15.1	802.15.1	802.15.4	802.11 (a, b, g, n)
Frequency (GHz)	2.4	2.4	0.868, 0.915, 2.4	2.4 and 5
Maximum raw bit rate (Mbps)	1-3	1	0.250	11 (b), 54 (g), 600 (n)
Typical data throughput (Mbps)	0.7-2.1	0.27	0.2	7 (b), 25 (g), 150 (n)
Maximum (Outdoor) Range (Meters)	10 (class 2), 100 (class 1)	50	10-100	100-250
Relative Power Consumption	Medium	Very low	Very low	High
Example Battery Life	Days	Months to years	Months to years	Hours
Network Size	7	Undefined	64,000+	255

Bluetooth isn't the best choice for every wireless job out there, but it does excel at short-range **cable-replacement**-type applications. It also boasts a typically more convenient connection process than its competitors (ZigBee specifically).

ZigBee is often a good choice for monitoring networks – like home automation projects. These networks might have dozens of wireless nodes, which are only sparsely active and never have to send a lot of data.

BLE combines the convenience of classic Bluetooth, and adds significantly lower power consumption. In this way it can compete with Zigbee for battery life. BLE can't compete with ZigBee in terms of network size, but for single device-to-device connectivity it's very comparable.

3.2.4 Asynchronous Serial communication using HC 05

Over the years, dozens of serial protocols have been crafted to meet particular needs of embedded systems. USB (universal *serial* bus), and Ethernet, are a couple of the more well-known computing serial interfaces. Other very common serial interfaces include SPI, I^2C, and the serial standard we're here to talk about today. Each of these serial interfaces can be sorted into one of two groups: synchronous or asynchronous.

A synchronous serial interface always pairs its data line(s) with a clock signal, so all devices on a synchronous serial bus share a common clock. This makes for a more straightforward, often faster serial transfer, but it also requires at least one extra wire between communicating devices. Examples of synchronous interfaces include SPI, and I^2C.

Asynchronous means that data is transferred **without support from an external clock signal**. This transmission method is perfect for minimizing the required wires and I/O pins, but it does mean we need to put some extra effort into reliably transferring and receiving data. The serial protocol we'll be discussing in this tutorial is the most common form of asynchronous transfers. It is so common, in fact, that when most folks say "serial" they're talking about this protocol (something you'll probably notice throughout this tutorial).

The clock-less serial protocol we'll be discussing in this tutorial is widely used in embedded electronics. If you're looking to add a GPS module, Bluetooth, XBee's, serial LCDs, or many other external devices to your project, you'll probably need to whip out some serial-fu.

3.3 Overview of the project

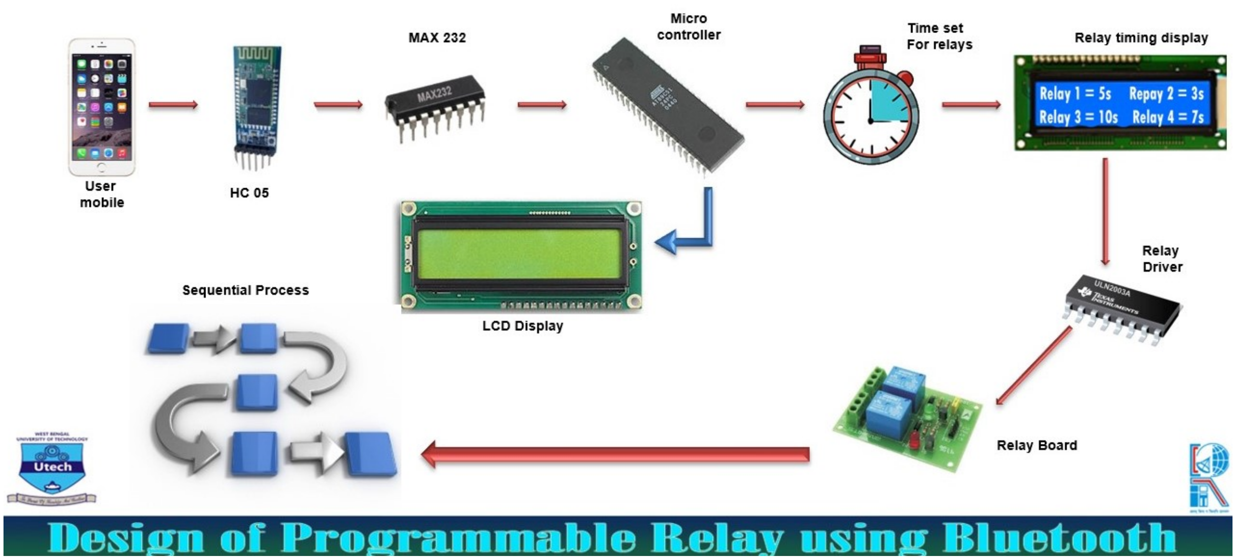


Figure 5: Overview of the complete project

In our project we interface a Bluetooth device (HC 05) to the microcontroller AT89c51. The Bluetooth device is working in slave mode. The interfacing is done through the serial communication using 9600 baud rate. The Bluetooth device is able to receive data from any Bluetooth enable device and send the data to the microcontroller. It also receives data from the microcontroller and sends the data to the Bluetooth enable device. Here other Bluetooth devices send some predefine data to the microcontroller through HC05 and based on the data, microcontroller sets the timer for each relay control and according to triggers from the timers, switch on & off of the relays connected to the system take place in a sequential manner. The complete layout of the project is shown in the figure 5. Here we implemented LCD display feedback section. The relay board has a relay driver (ULN 2003A). The relays are controlled by the microcontroller and they are able to handle 230 V ac with 7 A load current. A 16 × 2 LCD is displaying the status of the circuit and the load status all the time.

3.4 Circuit Diagram

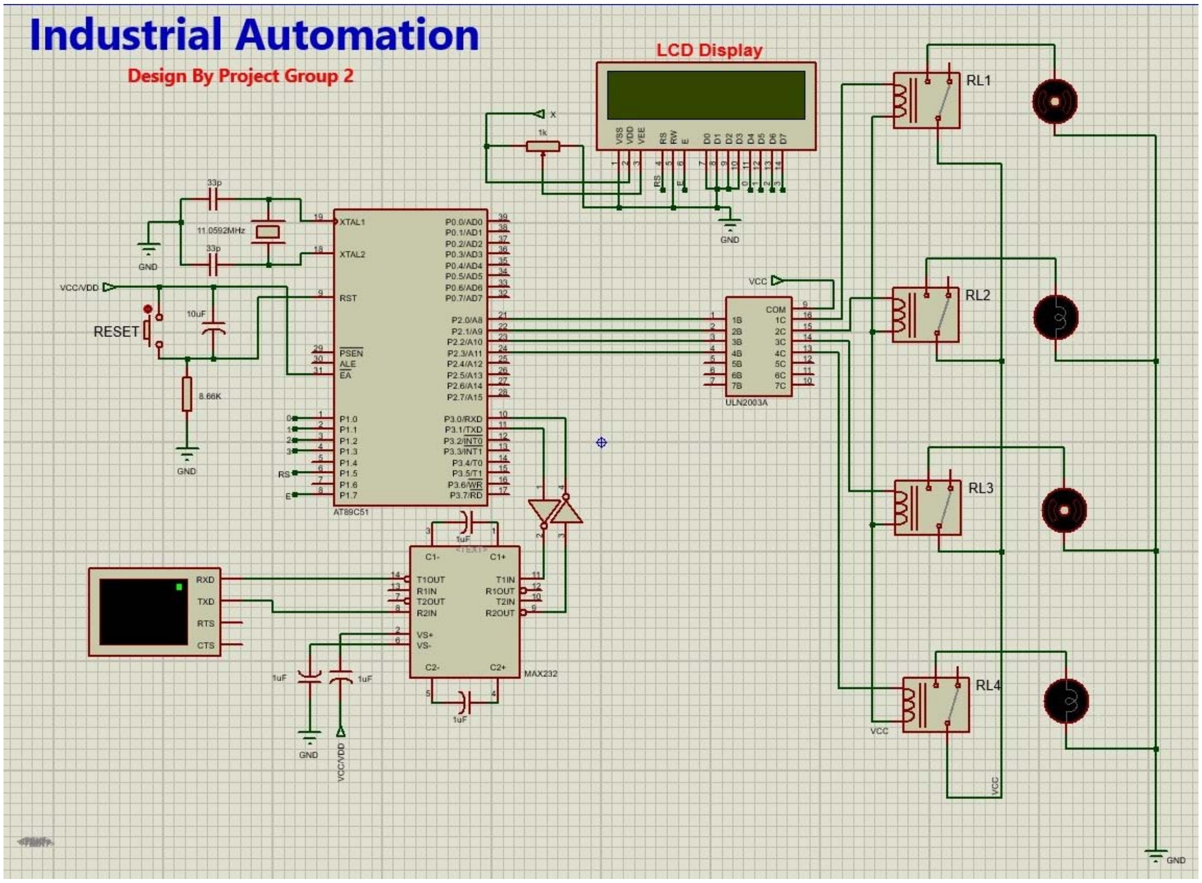


Figure 6: Complete circuit Diagram of the Project

The complete circuit is first simulated in the PROTEUS software, the screenshot of the PROTEUS window is shown in figure 6. The circuit is running well in this software. The software simulation includes microcontroller (AT89c51), serial terminal (replica of HC 05), level converter (TTL to RS232 level shifter and vice versa), the LCD panel (for displaying the condition and status of the load), relay driver section (low power to high power converter) and the relay section. Here we incorporate 4 loads which can be controlled sequentially as per user's set time within a time range.

The LCD is connected in 4 bit mode to save number of pins required in the microcontroller. The microcontroller uses 11.0592 MHz crystal for generating exactly 9600 baud rate which is the main criterion for HC05 serial communication.

CHAPTER 4

(Hardware Modeling)

4.1 Main features of the prototype

The features of the developed prototype are:

- LCD display (showing the timer for four relays)
- 4 independent load control (250 volt, 7 amp max, ON/OFF control)
- Inbuilt relay driver
- Only one mode of control (text based)
- Can be connected with any Bluetooth device
- In texted mode the load status will be displayed in the remote device
- 5 Volt operation (both control board and relay board)
- Excellent range (more than 30 m)

4.2 Photographs of the prototype

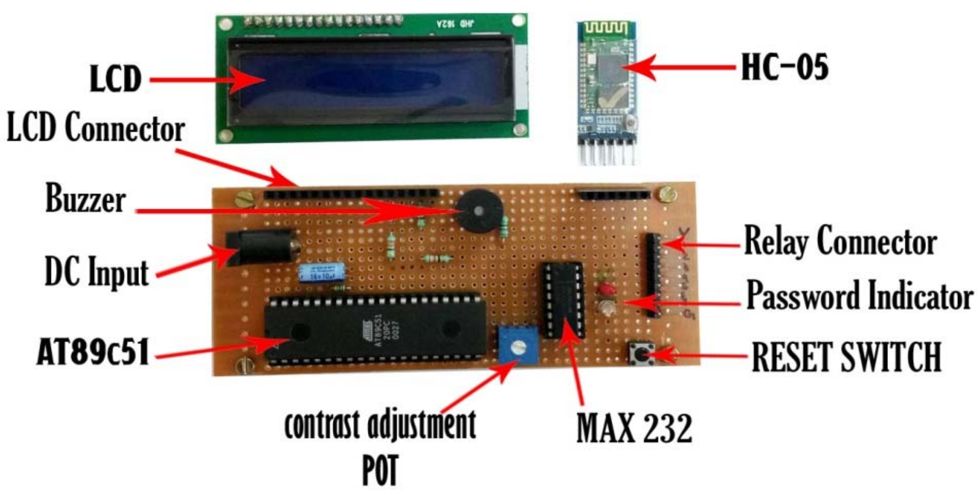


Figure 7: Main Controller board

4.3 Step by step operation of the prototype

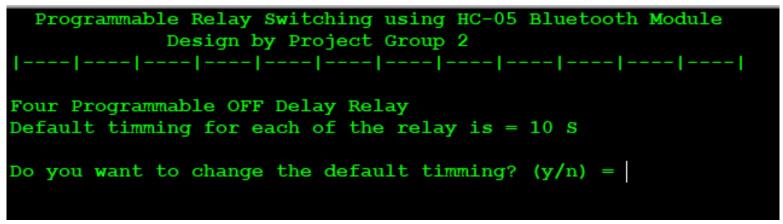
1. Connect the DC adapter (5V, 1A) to the DC jack.

2. Power On the circuit

3. Open the application (for mobile 'Bluetooth MCU" or "tera term", and for windows "tera term")

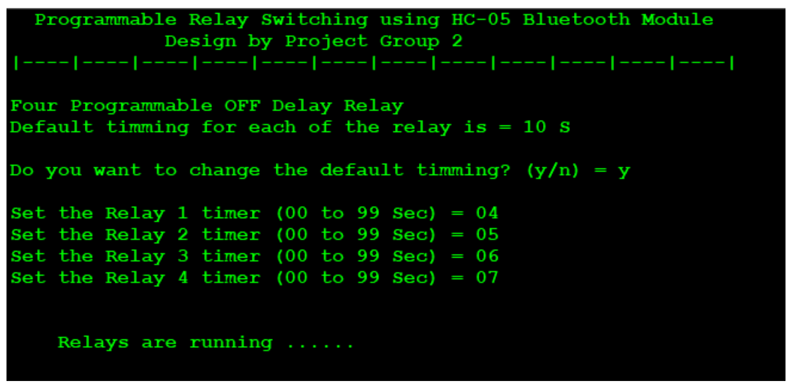
4. Connect the HC 05 from the mobile or laptop

5. It will ask the time to set by the user or use the default 10 s time for each relay



6. User need to set the time in XX format and in second, e.g. 05 sec or 12 sec etc

7. Set the time for each four relays

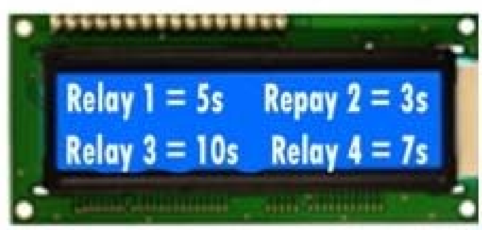


8. Time set timer for each relay will be shown in the LCD display

9. Now run the program. The decreasing time will also be shown in the LCD display.

10. It will run continuously until press reset button.

Relay timing display



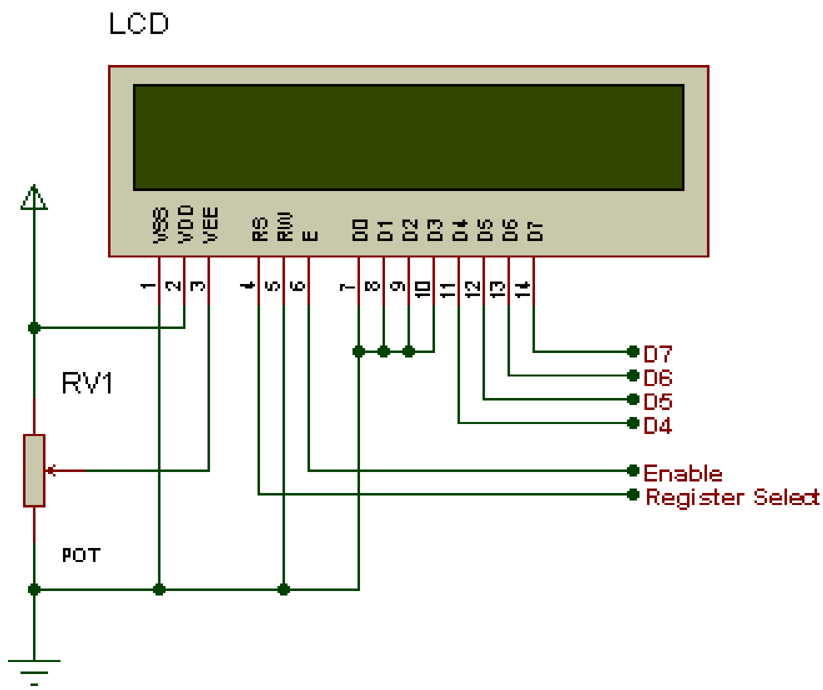
4.4 Components required

Sl. No.	Components	Quantity
1	HC-05	1
2	MAX 232	1
3	AT89c51 Microcontroller	1
4	33 pf Capacitor	2
5	0.1 μF Capacitor	2
6	10μF Capacitor	2
7	1 μF, 16V Capacitor	4
8	330 Ω resistance	9
9	10 K POT	1
10	16x2 LCD Display	1
11	ULN 2003A IC (Relay Driver)	1
12	5 volt static Relay	4
13	Male pin header	2
14	Female pin header	1
15	3 mm LED (Red/Green)	9
16	General blank PCB	1
17	11.0592 MHz Crystal	1
18	Jumper wire	9
19	Piezo Buzzer	1
20	16 pin IC base	1
21	40 pin IC base	1
22	Single stand wire	3m
23	Wire nipper	1
24	Wire striper	1
25	Soldering Iron	1
26	Soldering material	1
27	De-soldering pump	1

Table 2: Component listing

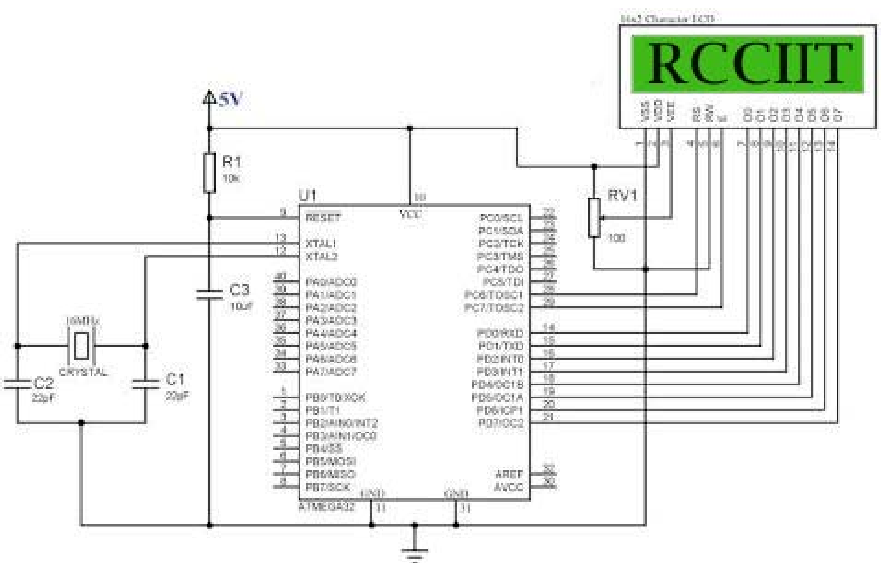
4.5 Hardware interfacing

4.5.1 16 × 2 LCD Module interfacing



4 bit mode

Figure 8: 4 bit mode LCD Interfacing



8 bit mode

Figure 9: 8 bit mode LCD Interfacing

4.5.2 LCD AT Commands

LCD Command Codes

Code (Hex)	Command to LCD Instruction Register
1	Clear display screen
2	Return home
4	Decrement cursor (shift cursor to left)
6	Increment cursor (shift cursor to right)
5	Shift display right
7	Shift display left
8	Display off, cursor off
A	Display off, cursor on
C	Display on, cursor off
E	Display on, cursor blinking
F	Display on, cursor blinking
10	Shift cursor position to left
14	Shift cursor position to right
18	Shift the entire display to the left
1C	Shift the entire display to the right
80	Force cursor to beginning to 1st line
C0	Force cursor to beginning to 2nd line
38	2 lines and 5x7 matrix

Table 3: LCD Attention (AT) Command

4.5.3 LCD Reset subroutine in AT89c51

```
lcd_reset:                  ;LCD reset sequence
        mov lcd_port, #0FFH
        mov delay,#20        ;20mS delay
        acall delayms
        mov lcd_port, #83H    ;Data = 30H, EN = 1, First Init
        mov lcd_port, #03H    ;Data = 30H, EN = 0
        mov delay,#10        ;Delay 10mS
        acall delayms
        mov lcd_port, #83H    ;Second Init, Data = 30H, EN = 1
        mov lcd_port, #03H    ;Data = 30H, EN = 0
        mov delay,#1         ;Delay 5mS
        acall delayms
        mov lcd_port, #83H    ;Third Init
        mov lcd_port, #03H
        mov delay,#1         ;Delay 5mS
```

```
        acall delayms
        mov lcd_port, #82H      ;Select Data width (20H for 4bit)
        mov lcd_port, #02H      ;Data = 20H, EN = 0
        mov delay,#1            ;Delay 5mS
        acall delayms
ret
```

4.5.4 LCD Initialization subroutine in AT89c51 (4 bit mode)

```
  lcd_init:
        mov a,#28H              ;4-bit, 2 line, 5x7 dots
        acall CMND
        mov a,#0CH              ;LCD_DATAlay ON cursor OFF
        acall CMND
        mov a,#06H              ;Set entry mode (Auto increment)
        acall CMND
        mov a,#80H              ;Bring cursor to line 1
        acall CMND
  ret
```

4.5.5 LCD Command subroutine in AT89c51 (4 bit mode)

```
  CMND:             ;LCD command Routine
        mov temp,a          ;Save a copy of command to temp
        swap a              ;Swap to use higher nibble
        anl a,#0FH          ;Mask the first four bits
        add a,#80H          ;Enable = 1, RS = 0, RW = 0
        mov lcd_port,a      ;Move it to lcd port
        mov delay,#3        ;5mS delay
        acall delayms
        CLR EN

        mov a,temp          ;Reload the command from temp
        anl a,#0FH          ;Mask first four bits
        add a,#80H          ;Enable = 1
        mov lcd_port,a      ;Move to port
        mov delay,#3        ;5mS delay
        acall delayms
        CLR EN
  ret
```

4.5.6 LCD Display subroutine in AT89c51 (4 bit mode)

```
  DISP:                     ;LCD data Routine
        mov temp,a          ;Keep copy of data in temp
        swap a              ;We need higher nibble
```

```
        anl a,#0FH          ;Mask first four bits
        add a,#0A0H         ;Enable = 1, RS = 1, RW = 0
        mov lcd_port,a      ;Move to lcd port
        mov delay,#3        ;5mS delay
        acall delayms
        clr en              ;Enable = 0

        mov a,temp          ;Reload the data from temp
        anl a,#0FH          ;we need lower nibble now
        add a,#0A0H         ;Enable = 1, RS = 1, RW = 0
        mov lcd_port,a      ;Move to lcd port
        mov delay,#3        ;5mS delay
        acall delayms
        clr en              ;Enable = 0
ret
```

4.5.7 Relay Driver interfacing with microcontroller

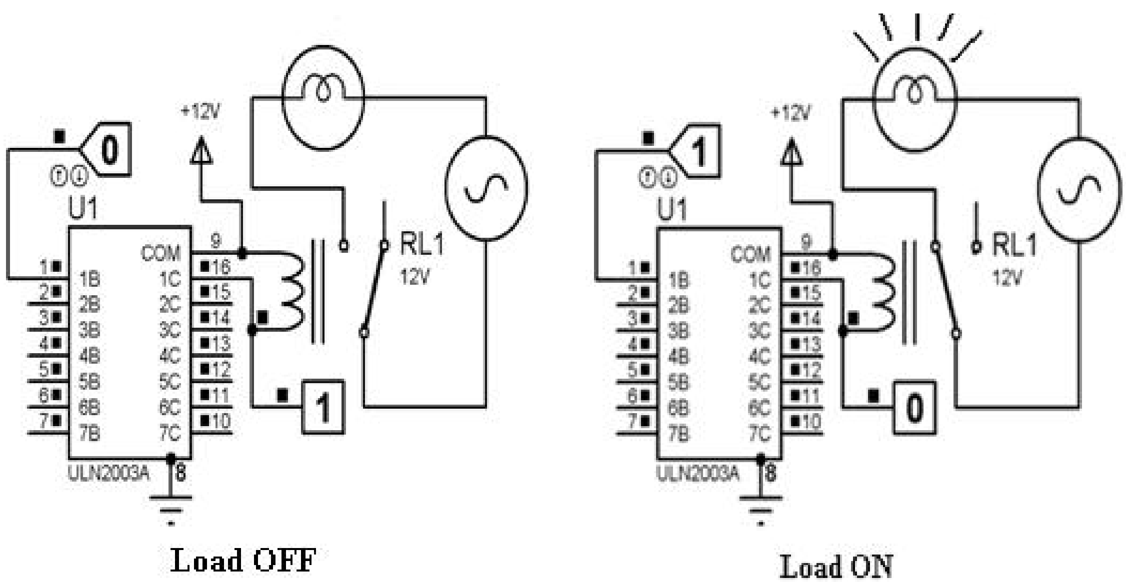


Figure 10: ULN2003A interfacing with microcontroller

The ULN2003A is a active high relay driver. 7 relays are controlled by this relay driver. Pin 1-7 are for controlling the relay which are connected to pin 10-16. For a ‘0’ from microcontroller the corresponding relay is turned off and a ‘1’ from microcontroller is turned on the relay.

CHAPTER 5

(Logic & Operation)

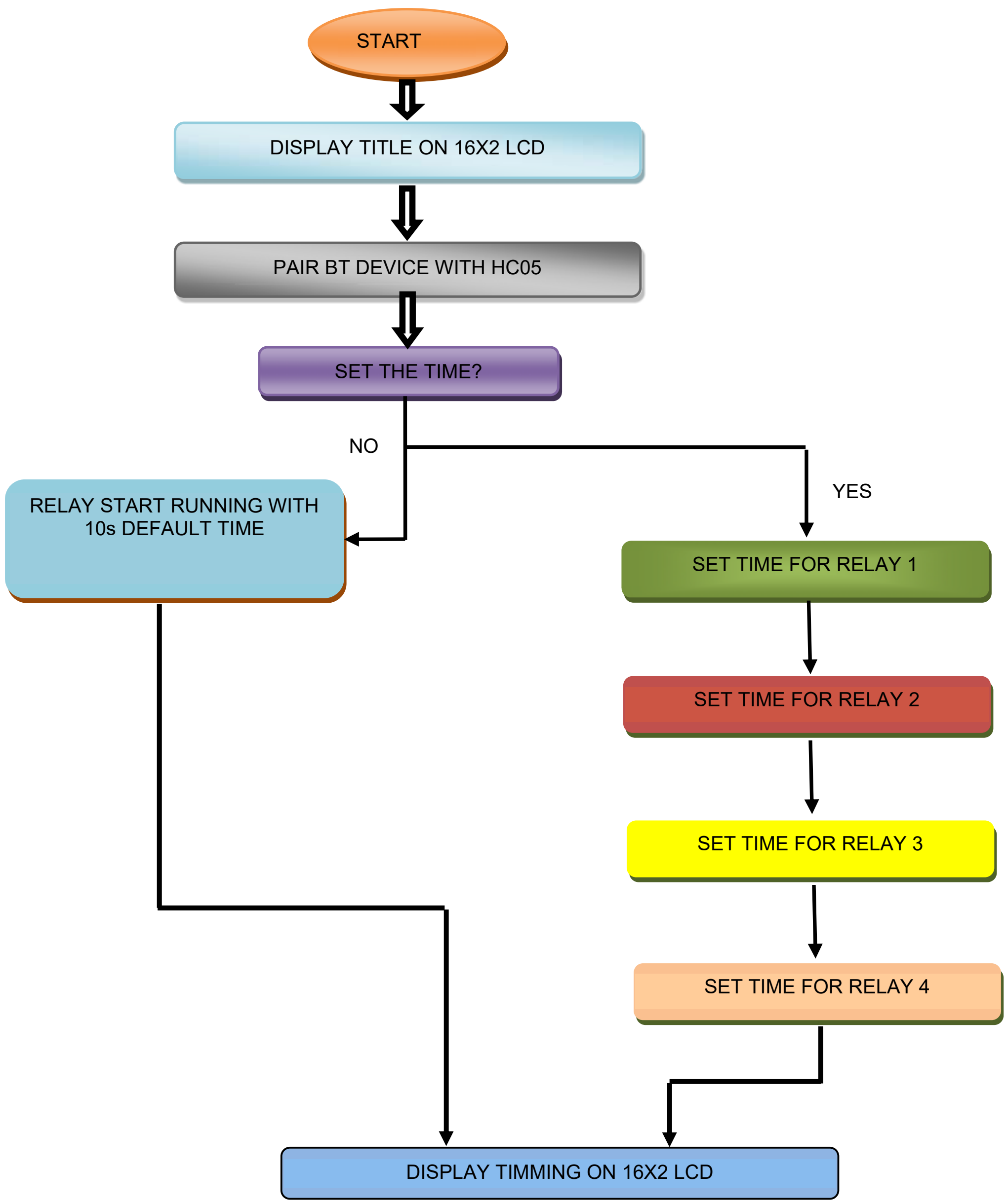
5.1 INTRODUCTION

After assembling the system, what remains is to observe its operation and efficiency of the system. The total system is divided in several sub systems, like

- BT section
- Microcontroller section
- LCD section
- Timer set function
- Relay section

The operation of the whole circuit is depending on every section's performance.

5.2 Flow Chart



5.3 Principle & Operations

The Bluetooth standard, like WiFi, uses the **FHSS** technique (***Frequency-Hopping Spread Spectrum***), which involves splitting the frequency band of 2.402-2.480 GHz into 79 channels (called *hops*) each 1MHz wide, then transmitting the signal using a sequence of channels known to both the sending and receiving stations. Thus, by switching channels as often as 1600 times a second, the Bluetooth standard can avoid interference with other radio signals.

First, we pair the android device with the Bluetooth module. Initially the HC 05 module blinks at the rate of 2 blinks per second. But as soon as it is paired with a device, the blinking rate decreases. When a signal is transmitted from a Bluetooth device, the paired **HC 05** module receive the signal. This signal is transmitted to the microcontroller through serial communication at 9600 baud rate. The microcontroller reads 5V but the HC 05 module sends signal in 3.3 volt. Here the MAX 232 level converter comes into play. It converts the 3.3 V to 5V, so that the signal is readable by the microcontroller. Microcontroller process the data. User need to set the timer for the four relays used here or he can use the default time 10 sec. The microcontroller output is 15mA. Relay needs 50mA to operate. So the relay driver converts this 15mA to 50mA. This relay driver, as the name implies, actually drives the relay. The controller also provides some feedback to ensure that the load switching has been done properly. The loads are connected to the relays. Thus, the loads can be controlled.

After setting the timer of the relays run the program. The relays will be switch on based on the timer set by the user. It's a sequential and continuous process. The relays are switched ON one by one and based on the time set by the user it will remain ON. The process is continuous until the reset button is pressed.

5.4 Advantages of the BT sequential relay switching

A. Maintenance: It is an economical system that requires very less maintenance as compared to conventional system as it has no complicated circuits and delicate mechanisms. This saves the additional maintenance cost.

B. Cost **:** The main advantage of this project is it has very low cost than the conventional one available in markets. For example, some commercial controllers use microcontrollers which alone costs around Rs.900. Some controllers even have a price range of Rs.2000-Rs. 4000. But for our system, the components used are less in number and easily available. Hence losses will be less leading to a better efficiency.

C. Construction: The construction of a BT based load switching system is very simple as it requires only a few components. The circuit involved is also relatively simpler. The space and power requirement to operate this system is very less.

D. Skill Required: Since the system we implement is simpler than the ones conventionally available, it can be easily made at home. The controller can also be easily operated by anyone.

5.5 Disadvantages

- The range of load switching is limited.
- No backup action will take for any false switching by controller itself.

5.6 Cost estimation of the project

Table 4: Costing of the projects

Sl. No.	Components	Quantity	Cost(Rs)
1	HC-05	1	350
2	MAX 232	1	20
3	AT89c51 Microcontroller	1	45
4	33 pf Capacitor	2	2
5	0.1 µF Capacitor	2	2
6	10µF Capacitor	2	4
7	1 µF, 16V Capacitor	4	8
8	330 Ω resistance	9	9
9	10 K POT	1	10
10	16x2 LCD Display	1	130
11	ULN 2003A IC (Relay Driver)	1	20
12	5 volt static Relay	7	140
13	Male pin header	2	20
14	Female pin header	1	20
15	3 mm LED (Red/Green)	9	10
16	General blank PCB	1	30
17	Wire nipper	1	-
18	11.0592 MHz Crystal	1	10
19	Jumper wire	9	30
20	Piezo Buzzer	1	10
21	16 pin IC base	1	5
22	40 pin IC base	1	5
23	Single strand wire	3m	30
24	Wire striper	1	-
25	Soldering Iron	1	-
26	Soldering material	1	-
27	De-soldering pump	1	-
	Total		890

5.7 Photographs of the protype

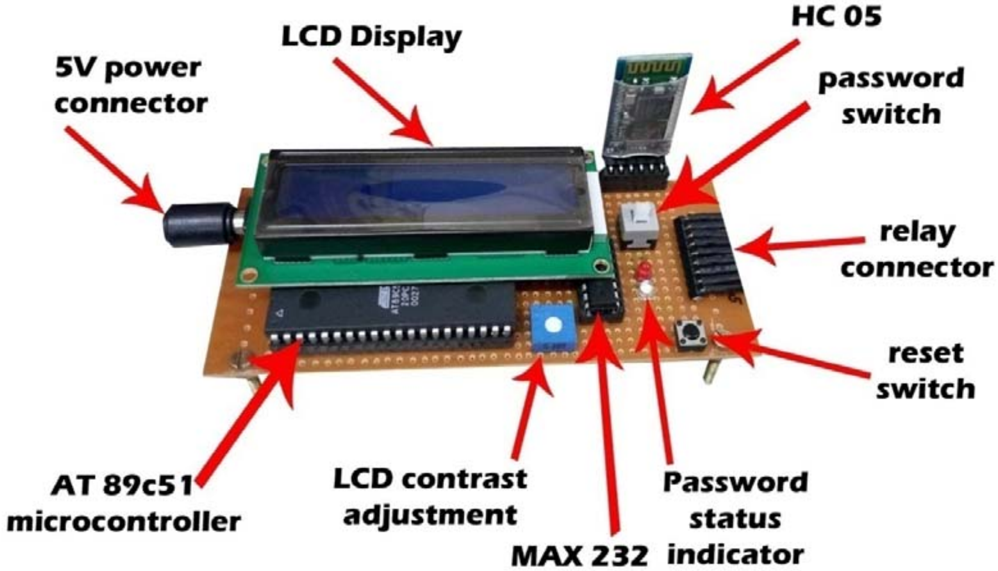


Figure 11: Main Controller Board

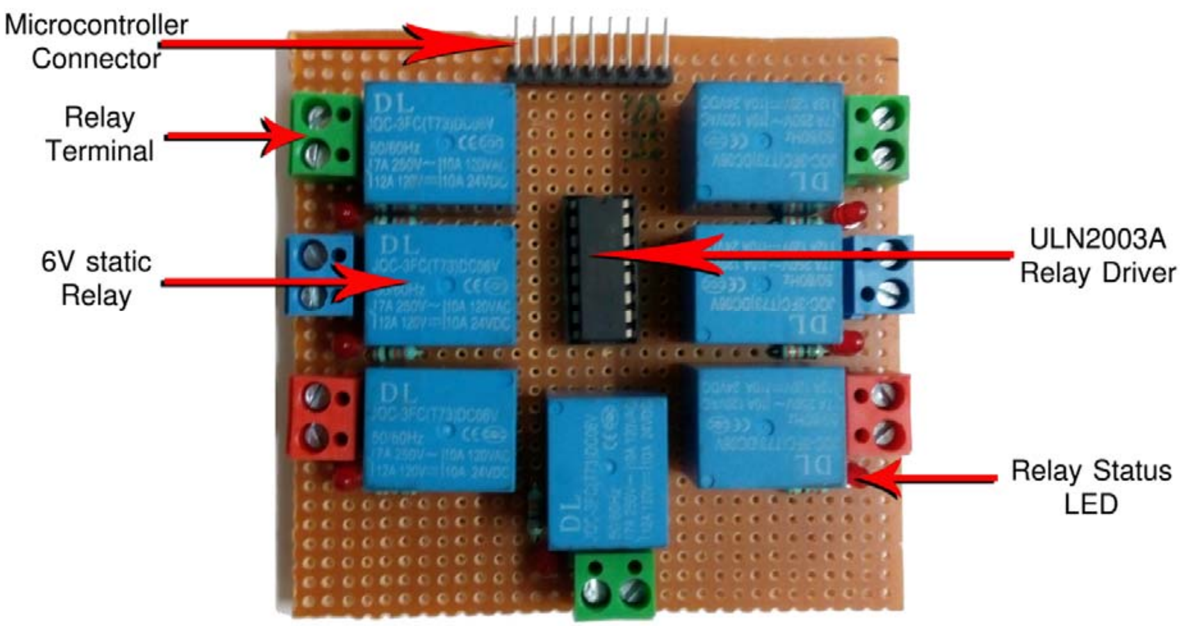


Figure 12: The Relay Board

Chapter 6

(Conclusion & Future Scope)

6.1 CONCLUSION

Here we have developed a 'Programmable Timer for Switching Relay Using Bluetooth' circuit which could be used for sequential load switching in process industries, it also limits the exceeding electricity bill. The circuit mainly consists of three parts such as BT section, visual feedback using LED display and also on distant mobile device connected via Bluetooth, sequential switching on and off the loads. When a signal is transmitted from a Bluetooth device the paired HC 05 module receive the signal which is transmitted to the microcontroller for data processing. According to given data from user, microcontroller sets timer for each relay. Then the relay driver drives the relay to switch on/off the sequential industrial electrical loads.

6.2 RESULTS

The experimental model was made according to the circuit diagram and the results were as expected. The loads are switched on and off according to the set time by the user. The user can set time using the distant device connected via Bluetooth. And according to set time of each timer, each relay is triggering the load sequentially. LCD display also showing the set time status and also countdown.

6.3 FUTURE WORK

The range of the Bluetooth transmission is limited to small distance (25 meter), user cannot control the devices from a large distance. So, we planned to use IOT devices (ESP 8266) to control the appliances over the internet in the future.

Figure 13: ESP 8266 (01)

Chapter 7

(References)

1. Md. Shahzamal, Mohammad Abu Sayid Haque, Md. Nasrul Haque Mia, Md. Anzan-Uz-Zaman, Sardar Masud Rana, Mahbubul Hoq, Fahmida Akter, Farhana Hafiz, Mahmudul Hasan, "Design and Development of Microcontroller Based Programmable Timer for Supply Control", NUCLEAR ELECTRONICS DIVISION, INSTITUTE OF ELECTRONICS ATOMIC ENERGY RESEARCH ESTABLISHMENT SAVAR, GPO BOX 3787, DHAKA-1349, BANGLADES, IE-46/NED-14, NOVEMBER 2013.

2. Deepashri K M , Sachidanand P B , Latha.H.S, "Industrial Appliances Control Using Android Mobile International Journal of Engineering and Manufacturing Science & Bluetooth Technology", International Journal of Electronics and Communication Engineering. ISSN 2249-3115 Volume 8, Number 1 (2018) pp. 33-42

3. Jonathan Gana KOLO, Umar Suleiman DAUDA, "Development of a Simple Programmable Control Timer", Leonardo Journal of Sciences ISSN 1583-0233, Issue 12, January-June 2008 p. 175-186.

4. Salako Emmanuel Adekunle, Department of Computer Science, FCT College of Education, Zuba-Abuja, Nigeria, "A Time Switching System Using Atmega328 Microcontroller towards Solving Problem of Electrical Power Wastage ", I.J. Image, Graphics and Signal Processing, 2019, 6, 18-26 Published Online June 2019 in MECS (http://www.mecs-press.org/) DOI: 10.5815/ijigsp.2019.06.03.

5. Julius Moises, S. Albalos, University of Northern Philippines, "MULTIPURPOSE SMS ACTIVATED PROGRAMMABLE TIMER SWITCH", International Journal of Scientific & Engineering Research Volume 10, Issue 7, July-2019, ISSN 2229-5518.

6. Samir Satpute, Manav Gupta, Mayur Bodke, Saurabh Katariya, Prof. S. Y. Kanawade, "Programmable Timer For Repeated Work", International Research Journal of Engineering and Technology (IRJET), Volume: 03 Issue: 04, Apr-2016.

7. Bernhard Schwager, " TIME CONTROLLED TEMPERATURE REGULATION WITH SAB 0529 PROGRAMMABLE TIMER IC", October 1984 .

Appendix A

(Hardware description)

Transformer less AC to DC power supply circuit using dropping capacitor

Production of low voltage DC power supply from AC power is the most important problem faced by many electronics developers and hobbyists. The straight forward technique is the use of a step down transformer to reduce the 230 V or 110V AC to a preferred level of low voltage AC. But *SMPS* power supply comes with the most appropriate method to create a low cost power supply by avoiding the use of bulky transformer. This circuit is so simple and it uses a voltage dropping capacitor in series with the phase line. Transformer less power supply is also called as capacitor power supply. It can generate 5V, 6V, 12V 150mA from 230V or 110V AC by using appropriate zener diodes.

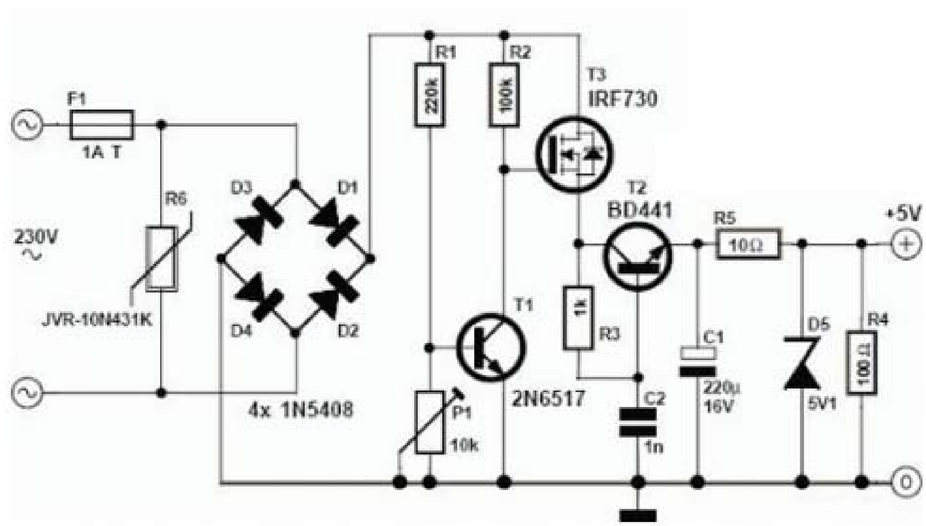


Figure 14: Transformer less SMPS 5 volt power supply

Working of Transformer less capacitor power supply

- This transformer less power supply circuit is also named as capacitor power supply since it uses a special type of AC capacitor in series with the main power line.
- A common capacitor will not do the work because the mains spikes will generate holes in the dielectric and the capacitor will be cracked by passing of current from the mains through the capacitor.
- X rated capacitor suitable for the use in AC mains is vital for reducing AC voltage.
- A X rated dropping capacitor is intended for 250V, 400V, 600V AC. Higher voltage versions are also obtainable. The dropping capacitor is non polarized so that it can be connected any way in the circuit.
- The 470kΩ resistor is a bleeder resistor that removes the stored current from the capacitor when the circuit is unplugged. It avoids the possibility of electric shock.
- Reduced AC voltage is rectified by bridge rectifier circuit. We have already discussed about bridge rectifiers. Then the ripples are removed by the 1000µF capacitor.

- This circuit provides 24 volts at 160 mA current at the output. This 24 volt DC can be regulated to necessary output voltage using an appropriate 1 watt or above zener diode.

- Here we are using 6.2V zener. You can use any type of zener diode in order to get the required output voltage.

AT 89c51 Microcontroller

AT89C51 is an 8-bit microcontroller and belongs to Atmel's 8051 family. **ATMEL 89C51** has 4KB of Flash programmable and erasable read only memory (PEROM) and 128 bytes of RAM. It can be erased and program to a maximum of 1000 times.

In 40 pin AT89C51, there are four ports designated as P_1, P_2, P_3 and P_0. All these ports are 8-bit bi-directional ports, *i.e.*, they can be used as both input and output ports. Except P_0 which needs external pull-ups, rest of the ports have internal pull-ups. When 1s are written to these port pins, they are pulled high by the internal pull-ups and can be used as inputs. These ports are also bit addressable and so their bits can also be accessed individually.

Port P_0 and P_2 are also used to provide low byte and high byte addresses, respectively, when connected to an external memory. Port 3 has multiplexed pins for special functions like serial communication, hardware interrupts, timer inputs and read/write operation from external memory. AT89C51 has an inbuilt UART for serial communication. It can be programmed to operate at different baud rates. Including two timers & hardware interrupts, it has a total of six interrupts.

PIN Diagram:

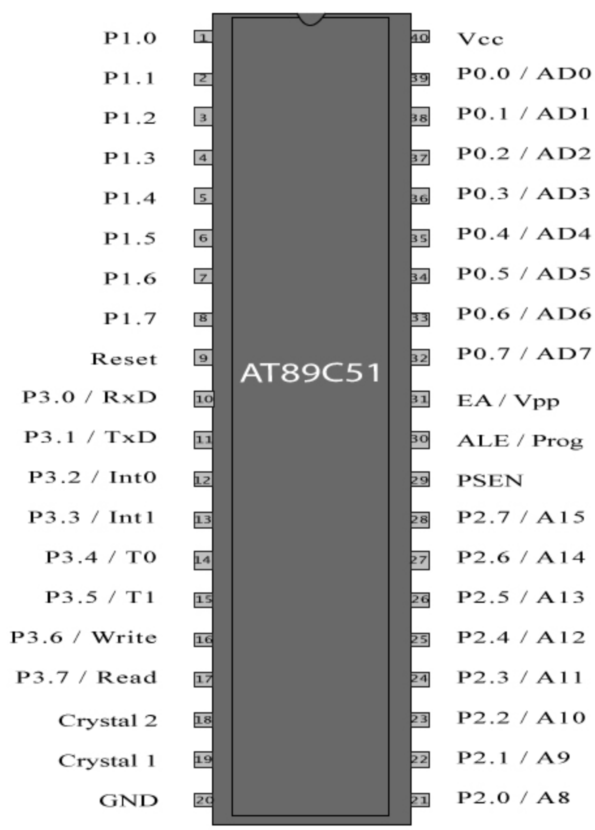


Figure **15**: 89c51 Microcontroller Pin Diagram

PIN Description:

Pin No	Function			Name
1	8 bit input/output port (P_1) pins			$P_1.0$
2				$P_1.1$
3				$P_1.2$
4				$P_1.3$
5				$P_1.4$
6				$P_1.5$
7				$P_1.6$
8				$P_1.7$
9	Reset pin; Active high			Reset
10	Input (receiver) for serial communication	RxD	8 bit input/output port (P_3) pins	$P_3.0$
11	Output (transmitter) for serial communication	TxD		$P_3.1$
12	External interrupt 1	Int0		$P_3.2$
13	External interrupt 2	Int1		$P_3.3$
14	Timer1 external input	T_0		$P_3.4$
15	Timer2 external input	T_1		$P_3.5$
16	Write to external data memory	Write		$P_3.6$
17	Read from external data memory	Read		$P_3.7$
18	Quartz crystal oscillator (up to 24 MHz)			Crystal 2
19				Crystal 1
20	Ground (0V)			Ground
21	8 bit input/output port (P_2) pins / High-order address bits when interfacing with external memory			$P_2.0/ A_8$
22				$P_2.1/ A_9$
23				$P_2.2/ A_{10}$
24				$P_2.3/ A_{11}$
25				$P_2.4/ A_{12}$
26				$P_2.5/ A_{13}$
27				$P_2.6/ A_{14}$
28				$P_2.7/ A_{15}$
29	Program store enable; Read from external program memory			PSEN
30	Address Latch Enable			ALE
	Program pulse input during Flash programming			Prog
31	External Access Enable; Vcc for internal program executions			EA
	Programming enable voltage; 12V (during Flash programming)			Vpp
32	8 bit input/output port (P_0) pins Low-order address bits when interfacing with external memory			$P_0.7/ AD_7$
33				$P_0.6/ AD_6$
34				$P_0.5/ AD_5$
35				$P_0.4/ AD_4$
36				$P_0.3/ AD_3$
37				$P_0.2/ AD_2$
38				$P_0.1/ AD_1$
39				$P_0.0/ AD_0$
40	Supply voltage; 5V (up to 6.6V)			Vcc

Table 5: Pin Description of 89c51 microcontroller

16x2 LCD Module:

- 16 character 2 lines display
- 4 bit and 8 bit data transfer mode
- display alpha numeric display
- backlight compatible
- contrast adjustment
- backlight intensity adjustment
- 5 volt operation
- compatible to almost every microcontroller

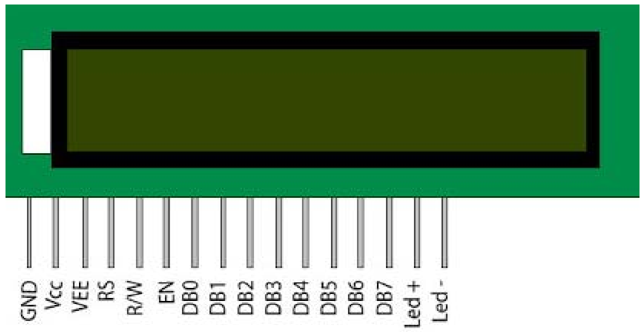


Figure 16: 16X2 LCD Module

LCD Pin outs

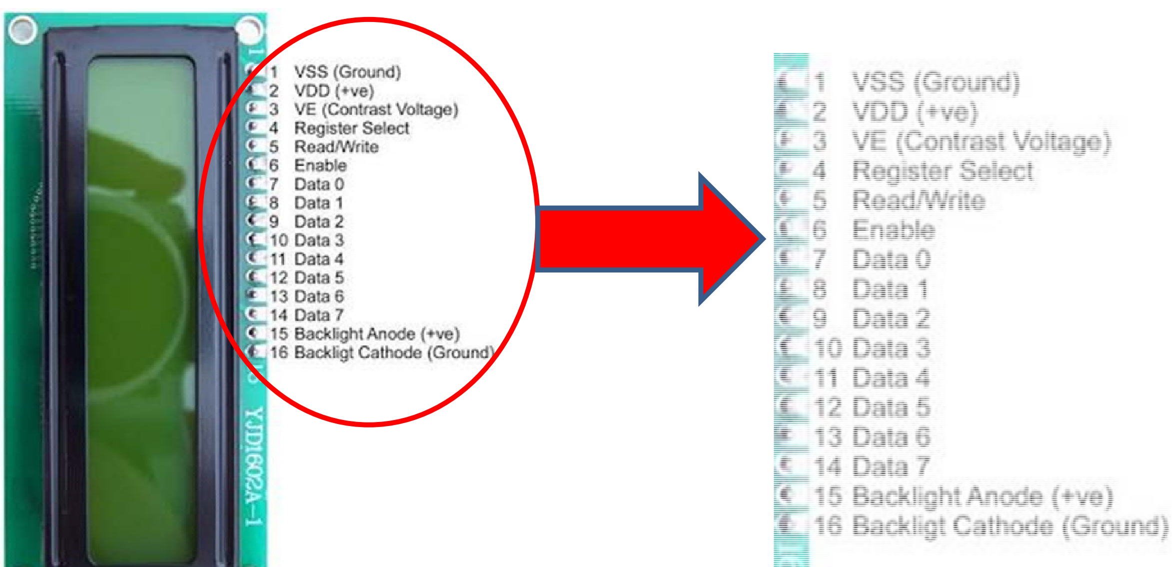


Figure 17: LCD Pin Diagram

Relay Driver

- The ULN2003 is a monolithic high voltage and high current Darlington transistor arrays.
- It consists of seven NPN Darlington pairs that features high-voltage outputs with common-cathode clamp diode for switching inductive loads.
- The collector-current rating of a single Darlington pair is 500mA.
- The ULN functions as an inverter.
- If the logic at input 1B is high then the output at its corresponding pin 1C will be low.

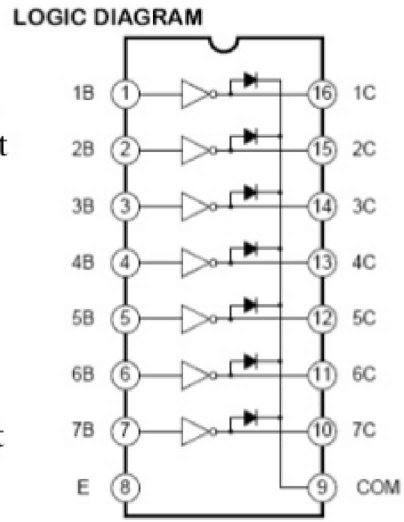


Figure 22: ULN2003A Internal Block Diagram

Resistor

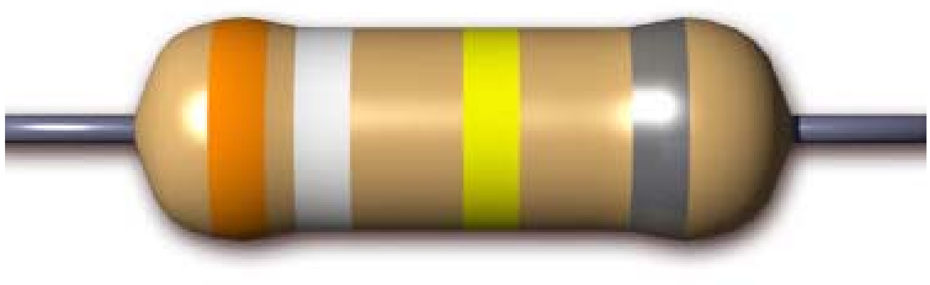


Figure 18: Resistor

Resistance is the opposition of a material to the current. It is measured in Ohms Ω. All conductors represent a certain amount of resistance, since no conductor is 100% efficient. To control the electron flow (current) in a predictable manner, we use resistors. Electronic circuits use calibrated lumped resistance to control the flow of current. Broadly speaking, resistor can be divided into two groups viz. fixed & adjustable (variable) resistors. In fixed resistors, the value is fixed & cannot be varied. In variable resistors, the resistance value can be varied by an adjuster knob. It can be divided into (a) Carbon composition (b) Wire wound (c) Special type. The most common type of resistors used in our projects is carbon type. The resistance value is normally indicated by color bands. Each resistance has four colors, one of the band on either side will be gold or silver, this is called fourth band and indicates the tolerance, others three band will give the value of resistance (see table). For example if a resistor has the following marking on it say red, violet, gold. Comparing these colored rings with the color code, its value is 27000 ohms or 27 kilo ohms and its tolerance is ±5%. Resistor comes in various sizes (Power rating).The bigger the size, the more power rating of 1/4 watts. The four color rings on its body tells us the value of resistor value.

Color Code of the resistor

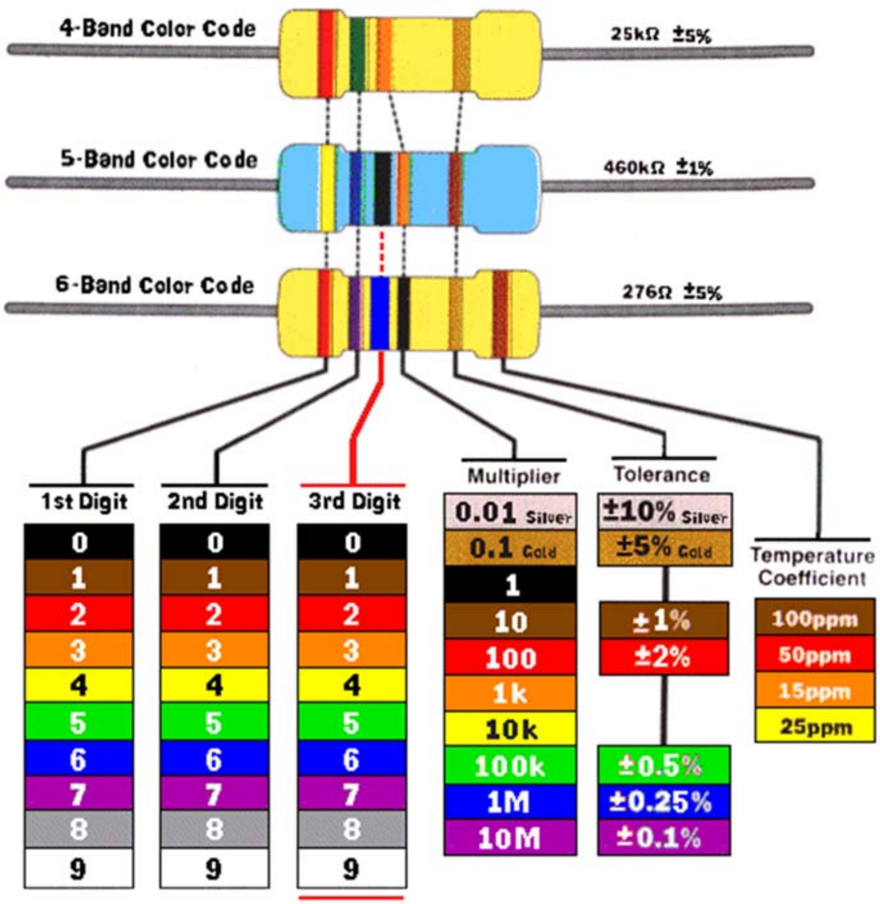


Figure 19: Color Code for resistance

RELAY

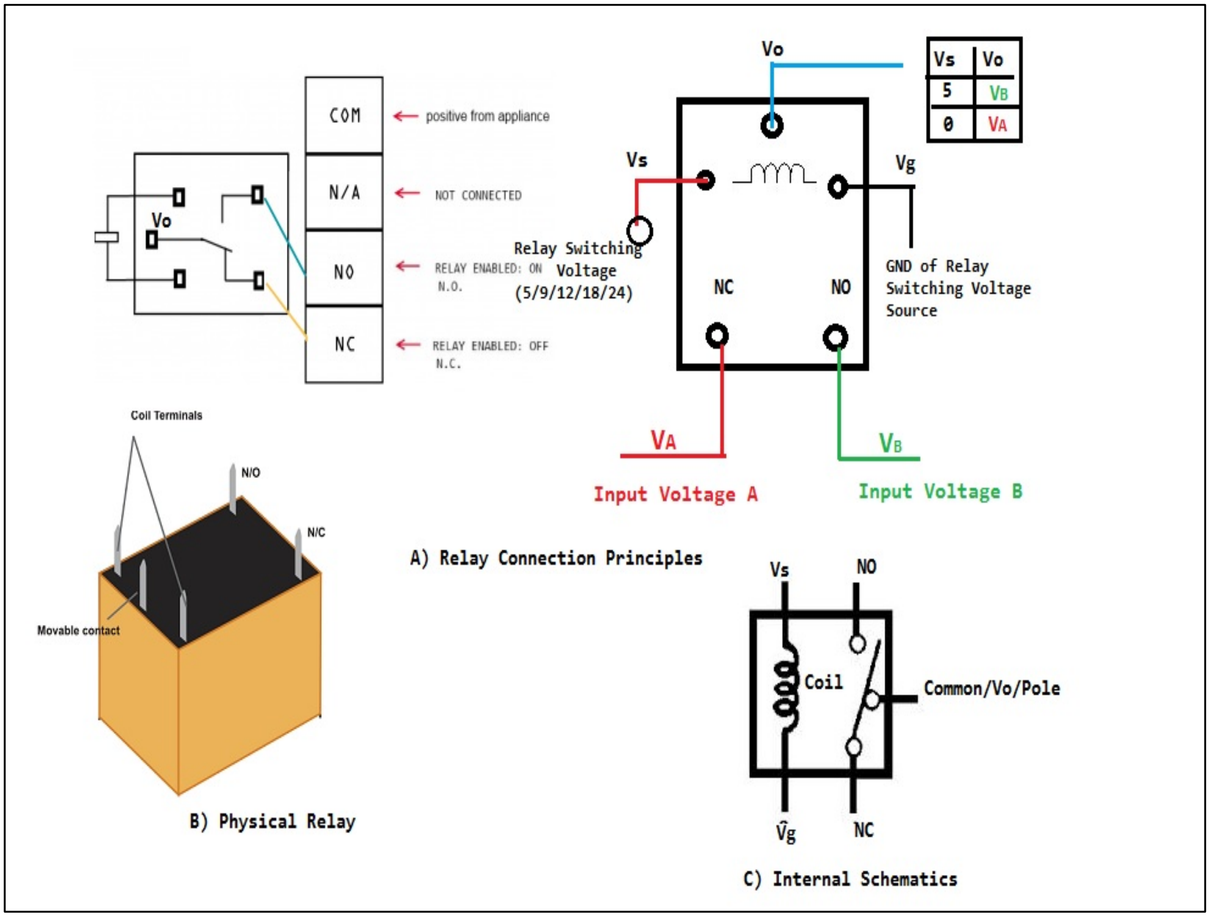


Figure 20: 6 volt Cube Relay

A relay is an electrically operated switch. Current flowing through the coil of the relay creates a magnetic field which attracts a lever and changes the switch contacts. The coil current can be on or off so relays have two switch positions and they are double throw (changeover) switches.

The relay's switch connections are usually labeled COM (POLE), NC and NO:

COM/POLE= Common, NC and NO always connect to this, it is the moving part of the switch.

NC = Normally Closed, COM/POLE is connected to this when the relay coil is not magnetized.

NO = Normally Open, COM/POLE is connected to this when the relay coil is MAGNETIZED and vice versa.

Capacitors

It is an electronic component whose function is to accumulate charges and then release it.

To understand the concept of capacitance, consider a pair of metal plates which all are placed near to each other without touching. If a battery is connected to these plates the positive pole to one and the negative pole to the other, electrons from the

Figure 21: Types of capacitors

battery will be attracted from the plate connected to the positive terminal of the battery. If the battery is then disconnected, one plate will be left with an excess of electrons, the other with a shortage, and a potential or voltage difference will exists between them. These plates will be acting as capacitors. Capacitors are of two types: - (1) **fixed type** like ceramic, polyester, electrolytic capacitors - these names refer to the material they are made of aluminum foil. (2) **Variable type** like gang condenser in radio or trimmer. In fixed type capacitors, it has two leads and its value is written over its body and variable type has three leads. Unit of measurement of a capacitor is farad denoted by the symbol F. It is a very big unit of capacitance. Small unit capacitor are pico-farad denoted by pf (1pf=1/1000,000,000,000 f) Above all, in case of electrolytic capacitors, it's two terminal are marked as (-) and (+).

Crystal Oscillator

A crystal oscillator is an electronic oscillator circuit that uses the mechanical resonance of a vibrating crystal of piezoelectric material to create an electrical signal with a precise frequency. This frequency is commonly used to keep track of time, as in quartz wristwatches, to provide a stable clock signal for digital integrated circuits, and to stabilize frequencies for radio transmitters and receivers. The most common type of piezoelectric resonator used is the quartz crystal, so oscillator circuits incorporating them became known as crystal oscillators, but other piezoelectric materials including polycrystalline ceramics are used in similar circuits.

Figure 22: Crystal Oscillator

Quartz crystals are manufactured for frequencies from a few tens of kilohertz to hundreds of megahertz. More than two billion crystals are manufactured annually. Most are used for consumer devices such as wristwatches, clocks, radios, computers, and cell phones. Quartz crystals are also found inside test and measurement equipment, such as counters, signal generators, and oscilloscopes.

Piezo buzzer

A buzzer or beeper is an audio signaling device, which may be mechanical, electromechanical, or piezoelectric. Typical uses of buzzers and beepers include alarm devices, timers and confirmation of user input such as a mouse click or keystroke. A piezoelectric element may be driven by an oscillating electronic circuit or other audio signal source, driven with a piezoelectric audio amplifier. Sounds commonly used to indicate that a button has been pressed are a click, a ring or a beep.

Figure 23: Piezo Buzzer

Blank PCB

A **printed circuit board** (**PCB**) mechanically supports and electrically connects electronic components using conductive tracks, pads and other features etched from copper sheets laminated onto a non-conductive substrate. PCBs can be *single sided* (one copper layer), *double sided* (two copper layers) or *multi-layer* (outer and inner layers). Multi-layer PCBs allow for much higher component density. Conductors on different layers are connected with plated-through holes called vias. Advanced PCBs may contain components - capacitors, resistors or active devices - embedded in the substrate.

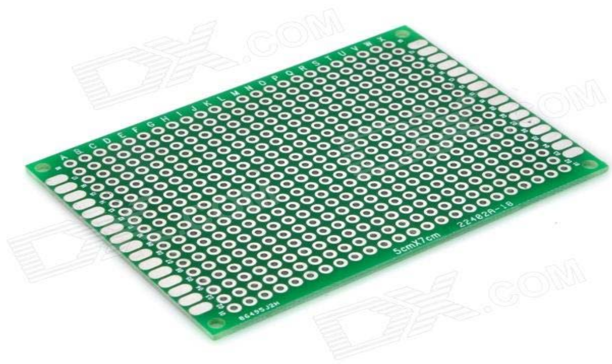


Figure 24: Blank glass epoxy PCB Board

FR-4 glass epoxy is the primary insulating substrate upon which the vast majority of rigid PCBs are produced. A thin layer of copper foil is laminated to one or both sides of an FR-4 panel. Circuitry interconnections are etched into copper layers to produce printed circuit boards. Complex circuits are produced in multiple layers.

Printed circuit boards are used in all but the simplest electronic products. Alternatives to PCBs include wire wrap and point-to-point construction. PCBs require the additional design effort to lay out the circuit, but manufacturing and assembly can be automated. Manufacturing circuits with PCBs is cheaper and faster than with other wiring methods as components are mounted and wired with one single part. Furthermore, operator wiring errors are eliminated.

HC 05

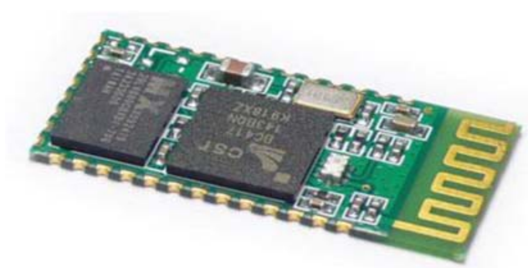


Figure 25: HC 05 Bluetooth Module

HC-05 module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup.
Serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses CSR Bluecore 04-External single chip Bluetooth system with CMOS technology and with AFH(Adaptive Frequency Hopping Feature). It has the footprint as small as 12.7mmx27mm. Hope it will simplify your overall design/development cycle.

Specifications
Hardware features

- Typical -80dBm sensitivity
- Up to +4dBm RF transmit power
- Low Power 1.8V Operation ,1.8 to 3.6V I/O
- PIO control
- UART interface with programmable baud rate
- With integrated antenna
- With edge connector

Software features

- Default Baud rate: 38400, Data bits:8, Stop bit:1,Parity:No parity, Data control: has. Supported baud rate: 9600,19200,38400,57600,115200,230400,460800.
- Given a rising pulse in PIO0, device will be disconnected.
- Status instruction port PIO1: low-disconnected, high-connected;
- PIO10 and PIO11 can be connected to red and blue led separately. When master and slave are paired, red and blue led blinks 1time/2s in interval, while disconnected only blue led blinks 2times/s.
- Auto-connect to the last device on power as default.
- Permit pairing device to connect as default.
- Auto-pairing PINCODE:"0000" as default
- Auto-reconnect in 30 min when disconnected as a result of beyond the range of connection.

Hardware

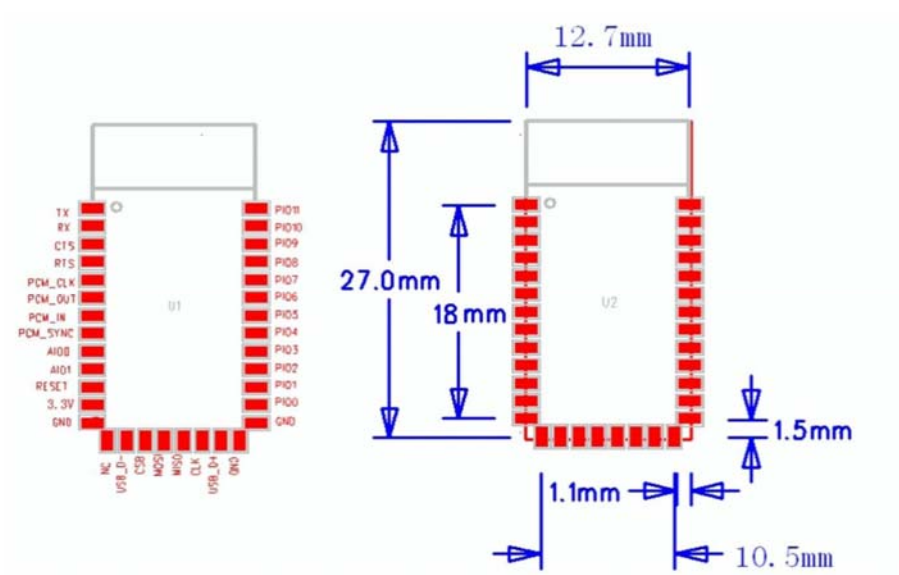


Figure 26: internal hardware connection of HC 05

Pin Description

The HC-05 Bluetooth Module has 6 pins. They are as follows:

ENABLE: When enable is pulled LOW, the module is disabled which means the module will not turn on and it fails to communicate.When enable is left open or connected to 3.3V, the module is enabled i.e the module remains on and communication also takes place.

Vcc: Supply Voltage 3.3V to 5V

GND: Ground pin

TXD & RXD: These two pins acts as an UART interface for communication.

STATE: It acts as a status indicator. When the module is not connected to / paired with any other Bluetooth device, signal goes Low. At this low state, the led flashes continuously which denotes that the module is not paired with other device. When this module is connected to/paired with any other Bluetooth device, the signal goes high. At this high state, the led blinks with a constant delay say for example 2s delay which indicates that the module is paired.

BUTTON SWITCH: This is used to switch the module into AT command mode. To enable AT command mode, press the button switch for a second. With the help of AT commands, the user can change the parameters of this module but only when the module is not paired with any other BT device. If the module is connected to any other Bluetooth device, it starts to communicate with that device and fails to work in AT command mode.

Appendix B

(Software coding)

PROGRAM CODE:

```
;                    Programmable Relay Switch
;--------------------------------------------------------
----------------------
         lcd_port equ P1          ;LCD connected to
Port1
         en equ P1.7              ;Enable connected to
P1.7
         delay equ 30H
         var1 equ 31H
         temp equ 32H             ; Default counting
time for all the relay
         temp1 equu 33H
         temp2 equ 34H
         temp3 equ 35H
         temp4 equ 36H
         temp5 equ 37H
         temp6 equ 38H
ORG 00H
CLR P3.2
CLR P3.3
CLR P3.4
CLR P3.5
START:
         mov TMOD,#20h                ;Timer 2 (8
bit auto reload mode)
      mov TH1,#0FDH                   ;9600 Baud
rate in 11.0592 MHz Crystal
      mov SCON,#50h
      setb TR1
;--------------------------------------------------------
---------------------------
;                              Initial Message Section
;--------------------------------------------------------
---------------------------
; LCD module is connected at Port 1 (data),
Controls are connected at port P1.7 (E)
; P1.5 (RS) - 4 bit mode
;--------------------------------------------------------
---------------------------
         ACALL LCD_RESET                    ;4
bit LCD RESET (software reset)
         acall lcd_init           ;4 bit mode LCD
initialization
MOV A, #82H
ACALL CMND
MOV DPTR, #MYDATA
         ;'Programmable' display
H_1:     CLR A
         MOVC A, @A+DPTR
         JZ b_1
         ACALL DISP
         INC DPTR
         SJMP H_1
b_1:
MOV A,#0C0H                          ;Jump    to
second line, position 2
ACALL CMND
         MOV DPTR, #MYDATA1    ;'Relay
Switching' display
H_2:     CLR A
         MOVC A, @A+DPTR
         JZ b_2
         ACALL DISP
         INC DPTR
         SJMP H_2
b_2:     MOV DELAY, #250
         ACALL DELAYmS
         MOV DELAY, #250
         ACALL DELAYmS
         MOV DELAY, #250
         ACALL DELAYmS
MOV A,#01H                           ;Clear
screen
ACALL CMND
MOV A,#82H                           ;Cursor line
one , position 4
ACALL CMND
         MOV DPTR, #MYDATA2    ;'Developed
by' display
H_3:     CLR A
         MOVC A, @A+DPTR
         JZ b_3
         ACALL DISP
         INC DPTR
         SJMP H_3
b_3:
MOV A,#0C1H                          ;Jump    to
second line, position 1
b_4:     MOV DELAY, #250
         ACALL DELAYmS
         MOV DELAY, #250
         ACALL DELAYmS
         MOV DELAY, #250
         ACALL DELAYmS
;--------------------------------------------------------
----------------------
;                    SERIAL          MONITOR
DISPLAY
;--------------------------------------------------------
----------------------
```

```
MOV DPTR, #MYDATA4 ; Wireless Temperature Sensor using HC-05 Bluetooth Module
H_5:    CLR A
        MOVC A, @A+DPTR
        JZ B_5
        ACALL TRANS
        INC DPTR
        SJMP H_5
B_5:
MOV A, #0DH ; for 'ENTER'
ACALL TRANS
MOV DPTR, #MYDATA5 ; Design by Budhaditya Biswas Display
H_6:    CLR A
        MOVC A, @A+DPTR
        JZ B_6
        ACALL TRANS
        INC DPTR
        SJMP H_6
B_6:
;-----------------------------------------------------------------------------
MOV A, #0DH ; for 'ENTER'
ACALL TRANS
MOV DPTR, #MYDATA6 ; Default timming for each Relay = 10 Sec
H_7:    CLR A
        MOVC A, @A+DPTR
        JZ B_7
        ACALL TRANS
        INC DPTR
        SJMP H_7
B_7:
MOV A, #0DH ; for 'ENTER'
ACALL TRANS
MOV DPTR, #MYDATA7 ; Do you want to change the default timming? (y/n) - display
H_8:    CLR A
        MOVC A, @A+DPTR
        JZ B_8
        ACALL TRANS
        INC DPTR
        SJMP H_8
B_8:
        ACALL RECEIVE
        CJNE A, #6EH, YES0 ;if 'n(6E)' go to YES2
        ACALL TRANS
        MOV A, #10H ; default timming =10s (it should be in the decimal coded HEX format)
        MOV TEMP3, #10H
        MOV TEMP4, #10H
        MOV TEMP5, #10H
        MOV TEMP6, #10H
        ACALL TIMERVALUE ;timer value is loading the relays time
        LJMP YES3

YES0:   CJNE A, #79H, B_8   ;'79H = y'
        ACALL TRANS
        MOV A, #0DH
        ACALL TRANS
ACALL MSGPRINT
MOV DPTR, #MYDATA10 ; Set the Relay 1 timer (00 to 99 Sec) =  - display
H_9:    CLR A
        MOVC A, @A+DPTR
        JZ B_9
        ACALL TRANS
        INC DPTR
        SJMP H_9
B_9:    MOV A, #83H
        ACALL CMND
        ACALL STORE ; STORE is the subroutine for storing the timer value
        MOV A, R5
        MOV TEMP3, A

MOV DPTR, #MYDATA11 ; Set the Relay 2 timer (00 to 99 Sec) =  - display
H_10:   CLR A
        MOVC A, @A+DPTR
        JZ B_10
        ACALL TRANS
        INC DPTR
        SJMP H_10
B_10:   MOV A, #8CH
        ACALL CMND
        ACALL STORE ; STORE is the subroutine for storing the timer value
        MOV A, R5
        MOV TEMP4, A

MOV DPTR, #MYDATA12 ; Set the Relay 3 timer (00 to 99 Sec) =  - display
H_11:   CLR A
        MOVC A, @A+DPTR
        JZ B_11
        ACALL TRANS
        INC DPTR
        SJMP H_11
```

```
B_11:	MOV A, #0C3H
	ACALL CMND
	ACALL STORE		; STORE is the subroutine for storing the timer value
	MOV A, R5
	MOV TEMP5, A

MOV DPTR, #MYDATA13		; Set the Relay 4 timer (00 to 99 Sec) =  - display
H_12:	CLR A
	MOVC A, @A+DPTR
	JZ B_12
	ACALL TRANS
	INC DPTR
	SJMP H_12
B_12:	MOV A, #0CCH
	ACALL CMND
	ACALL STORE		; STORE is the subroutine for storing the timer value
	MOV A, R5
	MOV TEMP6, A
ACALL RUNNING
ACALL MSGPRINT
AGAIN:
ACALL DEFAULTPRINT
MOV A, TEMP3
ACALL TIMERVALUE
MOV temp1, R6
MOV R5, TEMP2
Z7:
SETB P3.2
MOV A,#83H		;Jump to second line, position 11
ACALL CMND
MOV A, R5
ACALL Print
DJNZ temp1, Z7
CLR P3.2
ACALL DEFAULTPRINT
MOV A, TEMP4
ACALL TIMERVALUE
MOV TEMP1, R6
MOV R5, TEMP2
Z8:
SETB P3.3
MOV A,#8CH		;Jump to second line, position 11
ACALL CMND
MOV A, R5
ACALL Print
DJNZ temp1, Z8
CLR P3.3
ACALL DEFAULTPRINT
MOV A, TEMP5
ACALL TIMERVALUE
MOV temp1, R6
MOV R5, TEMP2
Z9:
SETB P3.4
MOV A,#0C3H		;Jump to second line, position 11
ACALL CMND
MOV A, R5
ACALL Print
DJNZ temp1, Z9
CLR P3.4
ACALL DEFAULTPRINT
MOV A, TEMP6
ACALL TIMERVALUE
MOV TEMP1, R6
MOV R5, TEMP2
Z10:
SETB P3.5
MOV A,#0CCH		;Jump to second line, position 11
ACALL CMND
MOV A, R5
ACALL Print
DJNZ temp1, Z10
CLR P3.5
ACALL DEFAULTPRINT
LJMP AGAIN
;--------------------------------------------------------------------------------
YES3:
ACALL RUNNING
ACALL MSGPRINT
AGAIN1:
ACALL DEFAULTPRINT
MOV temp1, R6
MOV R5, TEMP2
Z1:
SETB P3.2
MOV A,#83H		;Jump to second line, position 11
ACALL CMND
MOV A, R5
ACALL Print
DJNZ temp1, Z1
CLR P3.2
ACALL DEFAULTPRINT
MOV TEMP1, R6
MOV R5, TEMP2
Z2:
SETB P3.3
MOV A,#8CH		;Jump to second line, position 11
ACALL CMND
```

```
MOV A, R5
ACALL Print
DJNZ temp1, Z2
CLR P3.3
ACALL DEFAULTPRINT
MOV temp1, R6
MOV R5, TEMP2
Z3:
SETB P3.4
MOV A,#0C3H                ;Jump to second line, position 11
ACALL CMND
MOV A, R5
ACALL Print
DJNZ temp1, Z3
CLR P3.4
ACALL DEFAULTPRINT
MOV TEMP1, R6
MOV R5, TEMP2
Z4:
SETB P3.5
MOV A,#0CCH                ;Jump to second line, position 11
ACALL CMND
MOV A, R5
ACALL Print
DJNZ temp1, Z4
CLR P3.5
ACALL DEFAULTPRINT
LJMP AGAIN1
;---------------------------------------------------------------------------------
;                    To display the timer in said position
;---------------------------------------------------------------------------------
PRINT:
CJNE A, #00H, CHECK1
MOV A, #30H
ACALL DISP
MOV A, #30H
ACALL DISP
LJMP ttttt
CHECK1:     CJNE A, #0AH, CHECK2
MOV A, #31H
ACALL DISP
MOV A, #30H
ACALL DISP
DEC R5
ACALL DELAY1S
SJMP ttttt
CHECK2:
ANL A, #0FH
CJNE A, #0FH, P_1
        MOV A, R5
        ANL A, #0F0H
        ADD A,#09H
        MOV R5, A
        ANL A, #0F0H
        ADD A, #03H
        SWAP A
        ACALL DISP
        MOV A, R5
        ANL A, #0FH
        ADD A, #30H
        ACALL DISP
        ACALL DELAY1S
        DEC R5
        MOV A, R5
        SJMP ttttt
P_1:
        MOV A, R5
        ANL A, #0F0H
        ADD A, #03H
        SWAP A
        ACALL DISP
        MOV A, R5
        ANL A, #0FH
        ADD A, #30H
        ACALL DISP
        ACALL DELAY1S
        DEC R5
        MOV A, R5
        SJMP ttttt
ttttt:  RET
;---------------------------------------------------------------------------------
;                    To display the DEFAULT TIMER
;---------------------------------------------------------------------------------
DEFAULTPRINT:
MOV A, #83H
ACALL CMND
MOV A, TEMP3
ANL A, #0F0H
SWAP A
ADD A, #30H
ACALL DISP
MOV A, TEMP3
ANL A, #0FH
ADD A, #30H
ACALL DISP
MOV A, #8CH
ACALL CMND
MOV A, TEMP4
ANL A, #0F0H
SWAP A
```

```
ADD A, #30H
ACALL DISP
MOV A, TEMP4
ANL A, #0FH
ADD A, #30H
ACALL DISP
MOV A, #0C3H
ACALL CMND
MOV A, TEMP5
ANL A, #0F0H
SWAP A
ADD A, #30H
ACALL DISP
MOV A, TEMP5
ANL A, #0FH
ADD A, #30H
ACALL DISP
MOV A, #0CCH
ACALL CMND
MOV A, TEMP6
ANL A, #0F0H
SWAP A
ADD A, #30H
ACALL DISP
MOV A, TEMP6
ANL A, #0FH
ADD A, #30H
ACALL DISP
RET
;-----------------------------------------------------------------
--------------------------
RUNNING:
        MOV DPTR, #MYDATA14         ;
Programming Running
H_17:   CLR A
        MOVC A, @A+DPTR
        JZ B_17
        ACALL TRANS
        INC DPTR
        SJMP H_17
B_17:   RET
;-----------------------------------------------------------------
--------------------------
STORE:
Z5:     ACALL RECEIVE
        MOV R5, A
        ANL A, #0F0H
        CJNE A, #30H, Z5    ;accept only the
numbers in ASCII
        MOV A, R5
        ANL A, #0FH
        SUBB A, #0AH
        JNC Z5
        CLR C
        MOV A, R5
        ACALL TRANS
        ACALL DISP
        MOV A, R5
        SUBB A, #30H
        MOV R5, A
Z6:     ACALL RECEIVE
        MOV R6, A
        ANL A, #0F0H
        CJNE A, #30H, Z6    ;accept only the
numbers in ASCII
        MOV A, R6
        ANL A, #0FH
        SUBB A, #0AH
        JNC Z6
        CLR C
        MOV A, R6
        ACALL TRANS
        ACALL DISP
        MOV A, R6
        SUBB A, #30H
        SWAP A
        ADD A, R5
        SWAP A
        MOV R5, A
RET
;-----------------------------------------------------------------
--------------------------
TIMERVALUE:
        MOV TEMP1, A            ; Default
counting time stored in TEMP1 (like 90s=(90H))
        MOV TEMP2, A
        MOV R5, A           ; also in R5
        ADD A, #01H
        MOV R6, A           ; ALSO IN
R6=A+1 (FOR 00 Printing)
RET
;-----------------------------------------------------------------
--------------------------
MSGPRINT:
MOV A,#01H                  ;Clear
screen
ACALL CMND
MOV A,#80H                  ;Cursor line
one , position 4
ACALL CMND
        MOV DPTR, #MYDATA8  ;'R1 = R2
= ' display
H_15:   CLR A
        MOVC A, @A+DPTR
        JZ b_15
        ACALL DISP
        INC DPTR
        SJMP H_15
```

```
b_15:
MOV A,#0C0H                     ;Jump to second line, position 1
ACALL CMND
        MOV DPTR, #MYDATA9   ;'R3 =   R4 = ' display
H_16:   CLR A
        MOVC A, @A+DPTR
        JZ b_16
        ACALL DISP
        INC DPTR
        SJMP H_16
b_16:   RET
;---------------------------------------------------------------------------------
TRANS:
        MOV SBUF, A
        JNB TI, $
        CLR TI
RET

RECEIVE:
        JNB RI,$
        clr RI
        mov A,SBUF
RET
;------------------------------------------------------------------------------------------
lcd_reset:                  ;LCD reset sequence
        mov lcd_port, #0FFH
        mov delay,#20          ;20mS delay
        acall delayms
        mov lcd_port, #83H     ;Data = 30H, EN = 1, First Init
        mov lcd_port, #03H     ;Data = 30H, EN = 0
        mov delay,#10          ;Delay 10mS
        acall delayms
        mov lcd_port, #83H     ;Second Init, Data = 30H, EN = 1
        mov lcd_port, #03H     ;Data = 30H, EN = 0
        mov delay,#1           ;Delay 5mS
        acall delayms
        mov lcd_port, #83H     ;Third Init
        mov lcd_port, #03H
        mov delay,#1           ;Delay 5mS
        acall delayms
        mov lcd_port, #82H     ;Select Data width (20H for 4bit)
        mov lcd_port, #02H     ;Data = 20H, EN = 0
        mov delay,#1           ;Delay 5mS
        acall delayms
ret
;-------------------------------------------------------------------------------
lcd_init:
        acall lcd_reset            ;Call LCD Reset sequence
        mov a,#28H             ;4-bit, 2 line, 5x7 dots
        acall CMND
        mov a,#0CH             ;LCD_DATAlay ON cursor OFF
        acall CMND
        mov a,#06H             ;Set entry mode (Auto increment)
        acall CMND
        mov a,#80H             ;Bring cursor to line 1
        acall CMND
ret
;-------------------------------------------------------------------------------
CMND:                  ;LCD command Routine
        mov temp,a                 ;Save a copy of command to temp
        swap a             ;Swap to use higher nibble
        anl a,#0FH         ;Mask the first four bits
        add a,#80H             ;Enable = 1, RS = 0, RW = 0
        mov lcd_port,a       ;Move it to lcd port
        mov delay,#3         ;5mS delay
        acall delayms
        CLR EN

        mov a,temp           ;Reload the command from temp
        anl a,#0FH           ;Mask first four bits
        add a,#80H           ;Enable = 1
        mov lcd_port,a       ;Move to port
        mov delay,#3         ;5mS delay
        acall delayms
        CLR EN
ret
;-------------------------------------------------------------------------------
DISP:                        ;LCD data Routine
        mov temp,a           ;Keep copy of data in temp
        swap a               ;We need higher nibble
        anl a,#0FH           ;Mask first four bits
        add a,#0A0H            ;Enable = 1, RS = 1, RW = 0
        mov lcd_port,a       ;Move to lcd port
        mov delay,#3         ;5mS delay
        acall delayms
        clr en               ;Enable = 0
```

```
        mov a,temp              ;Reload the data from temp
        anl a,#0FH              ;we need lower nibble now
        add a,#0A0H             ;Enable = 1, RS = 1, RW = 0
        mov lcd_port,a          ;Move to lcd port
        mov delay,#3            ;5mS delay
        acall delayms
        clr en                  ;Enable = 0
ret
;---------------------------------------------------------------------------------------------
delayms:
        mov var1,#200                   ;changed from 230
d:
        nop
        nop
        djnz var1,d
        djnz delay,delayms
        ret
;---------------------------------------------------------------------------------------------
DELY:
        MOV R1, #01H
LOOP:
MOV   TMOD, #01H ; Counter 1, Mode 0
        MOV TL0, #00H       ; EE00 provides 5 ms Delay (11.0592 MHz)
        MOV TH0, #0EEH
        SETB TR0
        JNB TF0, $
        CLR TR0
        CLR TF0
        DJNZ R1, LOOP
RET
;-----------------------------------------------------
DELAY1S:                        ; for display the content for 1 sec
        MOV R1, #0DH
LOOP2:
MOV   TMOD, #01H ; Counter 1, Mode 0
        MOV TL0, #0C5H    ;   06C5   provides 900 ms Delay (11.0592 MHz)
        MOV TH0, #06H
        SETB TR0
        JNB TF0, $
        CLR TR0
        CLR TF0
        DEC R1
        MOV A, R1
        JNZ LOOP2
RET
;-----------------------------------------------------
MYDATA:             DB 'Programmable', 0
MYDATA1:   DB 'Relay Switching', 0
MYDATA2:   DB 'Developed By', 0
MYDATA3:   DB 'Project Group 2', 0
MYDATA4:   DB '     Programmable Relay Switching using HC-05 Bluetooth Module', 0
MYDATA5:   DB '              Design by Project Group 2',0DH, '|----|----|----|----|----|----|----|----|----|----|----|----|',0DH, 0
MYDATA6:   DB 'Four Programmable OFF Delay Relay',0DH, 'Default timming for each of the relay is = 10 S', 0
MYDATA7:   DB 0DH,'Do you want to change the default timming? (y/n) = ', 0
MYDATA8:   DB 'R1=   S  R2=   S', 0
MYDATA9:   DB 'R3=   S  R4=   S', 0
MYDATA10: DB 0DH,'Set the Relay 1 timer (00 to 99 Sec) = ', 0
MYDATA11: DB 0DH,'Set the Relay 2 timer (00 to 99 Sec) = ', 0
MYDATA12: DB 0DH,'Set the Relay 3 timer (00 to 99 Sec) = ', 0
MYDATA13: DB 0DH,'Set the Relay 4 timer (00 to 99 Sec) = ', 0
MYDATA14: DB 0DH,0DH,0DH,'     Relays are running ......', 0
;---------------------------------------------------------------------------------------------
END
```

Appendix C

(Data sheets)

ULN200x, ULQ200x High-Voltage, High-Current Darlington Transistor Arrays

1 Features

- 500-mA-Rated Collector Current (Single Output)
- High-Voltage Outputs: 50 V
- Output Clamp Diodes
- Inputs Compatible With Various Types of Logic
- Relay-Driver Applications

2 Applications

- Relay Drivers
- Stepper and DC Brushed Motor Drivers
- Lamp Drivers
- Display Drivers (LED and Gas Discharge)
- Line Drivers
- Logic Buffers

3 Description

The ULx200xA devices are high-voltage, high-current Darlington transistor arrays. Each consists of seven NPN Darlington pairs that feature high-voltage outputs with common-cathode clamp diodes for switching inductive loads.

The collector-current rating of a single Darlington pair is 500 mA. The Darlington pairs can be paralleled for higher current capability. Applications include relay drivers, hammer drivers, lamp drivers, display drivers (LED and gas discharge), line drivers, and logic buffers. For 100-V (otherwise interchangeable) versions of the ULx2003A devices, see the SLRS023 data sheet for the SN75468 and SN75469 devices.

The ULN2002A device is designed specifically for use with 14-V to 25-V PMOS devices. Each input of this device has a Zener diode and resistor in series to control the input current to a safe limit. The ULx2003A devices have a 2.7-kΩ series base resistor for each Darlington pair for operation directly with TTL or 5-V CMOS devices.

The ULx2004A devices have a 10.5-kΩ series base resistor to allow operation directly from CMOS devices that use supply voltages of 6 V to 15 V. The required input current of the ULx2004A device is below that of the ULx2003A devices, and the required voltage is less than that required by the ULN2002A device.

Device Information(1)

PART NUMBER	PACKAGE	BODY SIZE (NOM)
ULx200xD	SOIC (16)	9.90 mm × 3.91 mm
ULx200xN	PDIP (16)	19.30 mm × 6.35 mm
ULN200xNS	SOP (16)	10.30 mm × 5.30 mm
ULN200xPW	TSSOP (16)	5.00 mm × 4.40 mm

(1) For all available packages, see the orderable addendum at the end of the data sheet.

Simplified Block Diagram

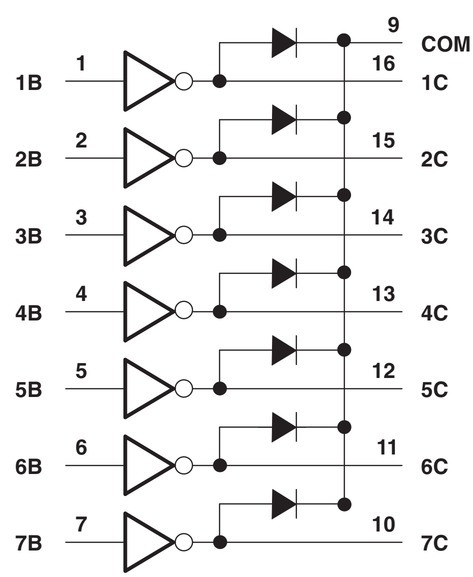


Table of Contents

4 Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from Revision N (June 2015) to Revision O — **Page**

- Changed Pin Functions table to correct typographical error. ... 3

Changes from Revision M (February 2013) to Revision N — **Page**

- Added *Pin Configuration and Functions* section, *ESD Ratings* table, *Feature Description* section, *Device Functional Modes*, *Application and Implementation* section, *Power Supply Recommendations* section, *Layout* section, *Device and Documentation Support* section, and *Mechanical, Packaging, and Orderable Information* section ... 1
- Deleted *Ordering Information* table. No specification changes. ... 1
- Moved *Typical Characteristics* into *Specifications* section. ... 8

Changes from Revision L (April 2012) to Revision M — **Page**

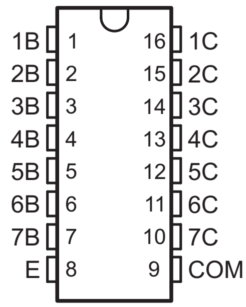
- Updated temperature rating for ULN2003AI in the ORDERING INFORMATION table ... 1

Changes from Revision K (August 2011) to Revision L — **Page**

- Removed reference to obsolete ULN2001 device ... 1

5 Pin Configuration and Functions

D, N, NS, and PW Package
16-Pin SOIC, PDIP, SO, and TSSOP
Top View



Pin Functions

PIN		I/O[(1)]	DESCRIPTION
NAME	NO.		
1B	1		
2B	2		
3B	3		
4B	4	I	Channel 1 through 7 Darlington base input
5B	5		
6B	6		
7B	7		
1C	16		
2C	15		
3C	14		
4C	13	O	Channel 1 through 7 Darlington collector output
5C	12		
6C	11		
7C	10		
COM	9	—	Common cathode node for flyback diodes (required for inductive loads)
E	8	—	Common emitter shared by all channels (typically tied to ground)

(1) I = Input, O = Output

6 Specifications

6.1 Absolute Maximum Ratings

at 25°C free-air temperature (unless otherwise noted)(1)

			MIN	MAX	UNIT
V_{CC}	Collector-emitter voltage			50	V
	Clamp diode reverse voltage(2)			50	V
V_I	Input voltage(2)			30	V
	Peak collector current, See Figure 4 and Figure 5			500	mA
I_{OK}	Output clamp current			500	mA
	Total emitter-terminal current			–2.5	A
T_A	Operating free-air temperature range	ULN200xA	–20	70	°C
		ULN200xAI	–40	105	
		ULQ200xA	–40	85	
		ULQ200xAT	–40	105	
T_J	Operating virtual junction temperature			150	°C
	Lead temperature for 1.6 mm (1/16 inch) from case for 10 seconds			260	°C
T_{stg}	Storage temperature		–65	150	°C

(1) Stresses beyond those listed under *Absolute Maximum Ratings* may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under *Recommended Operating Conditions* is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.
(2) All voltage values are with respect to the emitter/substrate terminal E, unless otherwise noted.

6.2 ESD Ratings

			VALUE	UNIT
$V_{(ESD)}$	Electrostatic discharge	Human body model (HBM), per ANSI/ESDA/JEDEC JS-001(1)	±2000	V
		Charged device model (CDM), per JEDEC specification JESD22-C101(2)	±500	

(1) JEDEC document JEP155 states that 500-V HBM allows safe manufacturing with a standard ESD control process.
(2) JEDEC document JEP157 states that 250-V CDM allows safe manufacturing with a standard ESD control process.

6.3 Recommended Operating Conditions

over operating free-air temperature range (unless otherwise noted)

		MIN	MAX	UNIT
V_{CC}	Collector-emitter voltage (non-V devices)	0	50	V
T_J	Junction temperature	–40	125	°C

6.4 Thermal Information

THERMAL METRIC(1)		ULx200x				UNIT
		D (SOIC)	N (PDIP)	NS (SO)	PW (TSSOP)	
		16 PINS	16 PINS	16 PINS	16 PINS	
$R_{\theta JA}$	Junction-to-ambient thermal resistance	73	67	64	108	°C/W
$R_{\theta JC(top)}$	Junction-to-case (top) thermal resistance	36	54	n/a	33.6	°C/W
$R_{\theta JB}$	Junction-to-board thermal resistance	n/a	n/a	n/a	51.9	°C/W
ψ_{JT}	Junction-to-top characterization parameter	n/a	n/a	n/a	2.1	°C/W
ψ_{JB}	Junction-to-board characterization parameter	n/a	n/a	n/a	51.4	°C/W

(1) For more information about traditional and new thermal metrics, see the *Semiconductor and IC Package Thermal Metrics* application report, SPRA953.

6.5 Electrical Characteristics: ULN2002A

T_A = 25°C

PARAMETER		TEST FIGURE	TEST CONDITIONS	ULN2002A MIN	TYP	MAX	UNIT
$V_{I(on)}$	ON-state input voltage	Figure 14	V_{CE} = 2 V, I_C = 300 mA			13	V
V_{OH}	High-level output voltage after switching	Figure 18	V_S = 50 V, I_O = 300 mA	V_S – 20			mV
$V_{CE(sat)}$	Collector-emitter saturation voltage	Figure 12	I_I = 250 µA, I_C = 100 mA		0.9	1.1	V
			I_I = 350 µA, I_C = 200 mA		1	1.3	
			I_I = 500 µA, I_C = 350 mA		1.2	1.6	
V_F	Clamp forward voltage	Figure 15	I_F = 350 mA		1.7	2	V
I_{CEX}	Collector cutoff current	Figure 9	V_{CE} = 50 V, I_I = 0			50	µA
		Figure 10	V_{CE} = 50 V, T_A = 70°C, I_I = 0			100	
			V_{CE} = 50 V, T_A = 70°C, V_I = 6 V			500	
$I_{I(off)}$	OFF-state input current	Figure 10	V_{CE} = 50 V, I_C = 500 µA	50	65		µA
I_I	Input current	Figure 11	V_I = 17 V		0.82	1.25	mA
I_R	Clamp reverse current	Figure 14	V_R = 50 V, T_A = 70°C			100	µA
			V_R = 50 V			50	
C_i	Input capacitance		V_I = 0, f = 1 MHz			25	pF

6.6 Electrical Characteristics: ULN2003A and ULN2004A

T_A = 25°C

PARAMETER		TEST FIGURE	TEST CONDITIONS	ULN2003A MIN	TYP	MAX	ULN2004A MIN	TYP	MAX	UNIT
$V_{I(on)}$	ON-state input voltage	Figure 14	V_{CE} = 2 V, I_C = 125 mA						5	V
			V_{CE} = 2 V, I_C = 200 mA			2.4			6	
			V_{CE} = 2 V, I_C = 250 mA			2.7				
			V_{CE} = 2 V, I_C = 275 mA						7	
			V_{CE} = 2 V, I_C = 300 mA			3				
			V_{CE} = 2 V, I_C = 350 mA						8	
V_{OH}	High-level output voltage after switching	Figure 18	V_S = 50 V, I_O = 300 mA	V_S – 20			V_S – 20			mV
$V_{CE(sat)}$	Collector-emitter saturation voltage	Figure 13	I_I = 250 µA, I_C = 100 mA		0.9	1.1		0.9	1.1	V
			I_I = 350 µA, I_C = 200 mA		1	1.3		1	1.3	
			I_I = 500 µA, I_C = 350 mA		1.2	1.6		1.2	1.6	
I_{CEX}	Collector cutoff current	Figure 9	V_{CE} = 50 V, I_I = 0			50			50	µA
		Figure 10	V_{CE} = 50 V, T_A = 70°C, I_I = 0			100			100	
			V_{CE} = 50 V, T_A = 70°C, V_I = 6 V						500	
V_F	Clamp forward voltage	Figure 16	I_F = 350 mA		1.7	2		1.7	2	V
$I_{I(off)}$	Off-state input current	Figure 11	V_{CE} = 50 V, T_A = 70°C, I_C = 500 µA	50	65		50	65		µA
I_I	Input current	Figure 12	V_I = 3.85 V		0.93	1.35				mA
			V_I = 5 V					0.35	0.5	
			V_I = 12 V					1	1.45	
I_R	Clamp reverse current	Figure 15	V_R = 50 V			50			50	µA
			V_R = 50 V, T_A = 70°C			100			100	
C_i	Input capacitance		V_I = 0, f = 1 MHz		15	25		15	25	pF

6.7 Electrical Characteristics: ULN2003AI

T_A = 25°C

PARAMETER		TEST FIGURE	TEST CONDITIONS		ULN2003AI MIN	TYP	MAX	UNIT
$V_{I(on)}$	ON-state input voltage	Figure 14	V_{CE} = 2 V	I_C = 200 mA			2.4	V
				I_C = 250 mA			2.7	
				I_C = 300 mA			3	
V_{OH}	High-level output voltage after switching	Figure 18	V_S = 50 V, I_O = 300 mA		V_S – 50			mV
$V_{CE(sat)}$	Collector-emitter saturation voltage	Figure 13	I_I = 250 µA,	I_C = 100 mA		0.9	1.1	V
			I_I = 350 µA,	I_C = 200 mA		1	1.3	
			I_I = 500 µA,	I_C = 350 mA		1.2	1.6	
I_{CEX}	Collector cutoff current	Figure 9	V_{CE} = 50 V,	I_I = 0			50	µA
V_F	Clamp forward voltage	Figure 16	I_F = 350 mA			1.7	2	V
$I_{I(off)}$	OFF-state input current	Figure 11	V_{CE} = 50 V,	I_C = 500 µA	50	65		µA
I_I	Input current	Figure 12	V_I = 3.85 V			0.93	1.35	mA
I_R	Clamp reverse current	Figure 15	V_R = 50 V				50	µA
C_i	Input capacitance		V_I = 0,	f = 1 MHz		15	25	pF

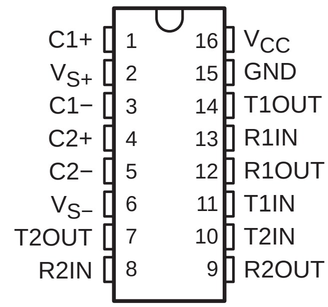
6.8 Electrical Characteristics: ULN2003AI

T_A = –40°C to 105°C

PARAMETER		TEST FIGURE	TEST CONDITIONS		ULN2003AI MIN	TYP	MAX	UNIT
$V_{I(on)}$	ON-state input voltage	Figure 14	V_{CE} = 2 V	I_C = 200 mA			2.7	V
				I_C = 250 mA			2.9	
				I_C = 300 mA			3	
V_{OH}	High-level output voltage after switching	Figure 18	V_S = 50 V, I_O = 300 mA		V_S – 50			mV
$V_{CE(sat)}$	Collector-emitter saturation voltage	Figure 13	I_I = 250 µA,	I_C = 100 mA		0.9	1.2	V
			I_I = 350 µA,	I_C = 200 mA		1	1.4	
			I_I = 500 µA,	I_C = 350 mA		1.2	1.7	
I_{CEX}	Collector cutoff current	Figure 9	V_{CE} = 50 V,	I_I = 0			100	µA
V_F	Clamp forward voltage	Figure 16	I_F = 350 mA			1.7	2.2	V
$I_{I(off)}$	OFF-state input current	Figure 11	V_{CE} = 50 V,	I_C = 500 µA	30	65		µA
I_I	Input current	Figure 12	V_I = 3.85 V			0.93	1.35	mA
I_R	Clamp reverse current	Figure 15	V_R = 50 V				100	µA
C_i	Input capacitance		V_I = 0,	f = 1 MHz		15	25	pF

- **Meets or Exceeds TIA/EIA-232-F and ITU Recommendation V.28**
- **Operates From a Single 5-V Power Supply With 1.0-μF Charge-Pump Capacitors**
- **Operates Up To 120 kbit/s**
- **Two Drivers and Two Receivers**
- **±30-V Input Levels**
- **Low Supply Current . . . 8 mA Typical**
- **ESD Protection Exceeds JESD 22**
  - **2000-V Human-Body Model (A114-A)**
- **Upgrade With Improved ESD (15-kV HBM) and 0.1-μF Charge-Pump Capacitors is Available With the MAX202**
- **Applications**
  - **TIA/EIA-232-F, Battery-Powered Systems, Terminals, Modems, and Computers**

MAX232 . . . D, DW, N, OR NS PACKAGE
MAX232I . . . D, DW, OR N PACKAGE
(TOP VIEW)



description/ordering information

The MAX232 is a dual driver/receiver that includes a capacitive voltage generator to supply TIA/EIA-232-F voltage levels from a single 5-V supply. Each receiver converts TIA/EIA-232-F inputs to 5-V TTL/CMOS levels. These receivers have a typical threshold of 1.3 V, a typical hysteresis of 0.5 V, and can accept ±30-V inputs. Each driver converts TTL/CMOS input levels into TIA/EIA-232-F levels. The driver, receiver, and voltage-generator functions are available as cells in the Texas Instruments LinASIC™ library.

ORDERING INFORMATION

T_A	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
0°C to 70°C	PDIP (N)	Tube of 25	MAX232N	MAX232N
	SOIC (D)	Tube of 40	MAX232D	MAX232
		Reel of 2500	MAX232DR	
	SOIC (DW)	Tube of 40	MAX232DW	MAX232
		Reel of 2000	MAX232DWR	
	SOP (NS)	Reel of 2000	MAX232NSR	MAX232
−40°C to 85°C	PDIP (N)	Tube of 25	MAX232IN	MAX232IN
	SOIC (D)	Tube of 40	MAX232ID	MAX232I
		Reel of 2500	MAX232IDR	
	SOIC (DW)	Tube of 40	MAX232IDW	MAX232I
		Reel of 2000	MAX232IDWR	

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.

Function Tables

EACH DRIVER

INPUT TIN	OUTPUT TOUT
L	H
H	L

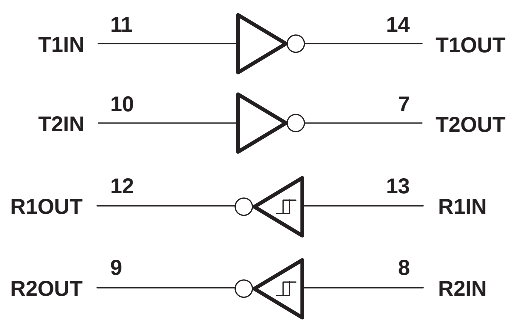
H = high level, L = low level

EACH RECEIVER

INPUT RIN	OUTPUT ROUT
L	H
H	L

H = high level, L = low level

logic diagram (positive logic)



absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

Input supply voltage range, V_{CC} (see Note 1)	−0.3 V to 6 V
Positive output supply voltage range, V_{S+}	V_{CC} − 0.3 V to 15 V
Negative output supply voltage range, V_{S-}	−0.3 V to −15 V
Input voltage range, V_I: Driver	−0.3 V to V_{CC} + 0.3 V
Receiver	±30 V
Output voltage range, V_O: T1OUT, T2OUT	V_{S-} − 0.3 V to V_{S+} + 0.3 V
R1OUT, R2OUT	−0.3 V to V_{CC} + 0.3 V
Short-circuit duration: T1OUT, T2OUT	Unlimited
Package thermal impedance, θ_{JA} (see Notes 2 and 3): D package	73°C/W
DW package	57°C/W
N package	67°C/W
NS package	64°C/W
Operating virtual junction temperature, T_J	150°C
Storage temperature range, T_{stg}	−65°C to 150°C

† Stresses beyond those listed under “absolute maximum ratings” may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under “recommended operating conditions” is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTES: 1. All voltages are with respect to network GND.
2. Maximum power dissipation is a function of T_J(max), θ_{JA}, and T_A. The maximum allowable power dissipation at any allowable ambient temperature is $P_D = (T_J(max) - T_A)/\theta_{JA}$. Operating at the absolute maximum T_J of 150°C can affect reliability.
3. The package thermal impedance is calculated in accordance with JESD 51-7.

recommended operating conditions

			MIN	NOM	MAX	UNIT
V_{CC}	Supply voltage		4.5	5	5.5	V
V_{IH}	High-level input voltage (T1IN,T2IN)		2			V
V_{IL}	Low-level input voltage (T1IN, T2IN)				0.8	V
R1IN, R2IN	Receiver input voltage				±30	V
T_A	Operating free-air temperature	MAX232	0		70	°C
		MAX232I	−40		85	

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted) (see Note 4 and Figure 4)

PARAMETER		TEST CONDITIONS	MIN	TYP‡	MAX	UNIT
I_{CC}	Supply current	V_{CC} = 5.5 V, All outputs open, T_A = 25°C		8	10	mA

‡ All typical values are at V_{CC} = 5 V and T_A = 25°C.

NOTE 4: Test conditions are C1−C4 = 1 μF at V_{CC} = 5 V ± 0.5 V.

DRIVER SECTION

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature range (see Note 4)

PARAMETER			TEST CONDITIONS	MIN	TYP†	MAX	UNIT
V_{OH}	High-level output voltage	T1OUT, T2OUT	R_L = 3 kΩ to GND	5	7		V
V_{OL}	Low-level output voltage‡	T1OUT, T2OUT	R_L = 3 kΩ to GND		−7	−5	V
r_o	Output resistance	T1OUT, T2OUT	$V_{S+} = V_{S-}$ = 0, V_O = ±2 V	300			Ω
I_{OS}§	Short-circuit output current	T1OUT, T2OUT	V_{CC} = 5.5 V, V_O = 0		±10		mA
I_{IS}	Short-circuit input current	T1IN, T2IN	V_I = 0			200	µA

† All typical values are at V_{CC} = 5 V, T_A = 25°C.
‡ The algebraic convention, in which the least-positive (most negative) value is designated minimum, is used in this data sheet for logic voltage levels only.
§ Not more than one output should be shorted at a time.
NOTE 4: Test conditions are C1−C4 = 1 µF at V_{CC} = 5 V ± 0.5 V.

switching characteristics, V_{CC} = 5 V, T_A = 25°C (see Note 4)

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
SR	Driver slew rate	R_L = 3 kΩ to 7 kΩ, See Figure 2			30	V/µs
SR(t)	Driver transition region slew rate	See Figure 3		3		V/µs
	Data rate	One TOUT switching		120		kbit/s

NOTE 4: Test conditions are C1−C4 = 1 µF at V_{CC} = 5 V ± 0.5 V.

RECEIVER SECTION

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature range (see Note 4)

PARAMETER			TEST CONDITIONS	MIN	TYP†	MAX	UNIT
V_{OH}	High-level output voltage	R1OUT, R2OUT	I_{OH} = −1 mA	3.5			V
V_{OL}	Low-level output voltage‡	R1OUT, R2OUT	I_{OL} = 3.2 mA			0.4	V
V_{IT+}	Receiver positive-going input threshold voltage	R1IN, R2IN	V_{CC} = 5 V, T_A = 25°C		1.7	2.4	V
V_{IT-}	Receiver negative-going input threshold voltage	R1IN, R2IN	V_{CC} = 5 V, T_A = 25°C	0.8	1.2		V
V_{hys}	Input hysteresis voltage	R1IN, R2IN	V_{CC} = 5 V	0.2	0.5	1	V
r_i	Receiver input resistance	R1IN, R2IN	V_{CC} = 5, T_A = 25°C	3	5	7	kΩ

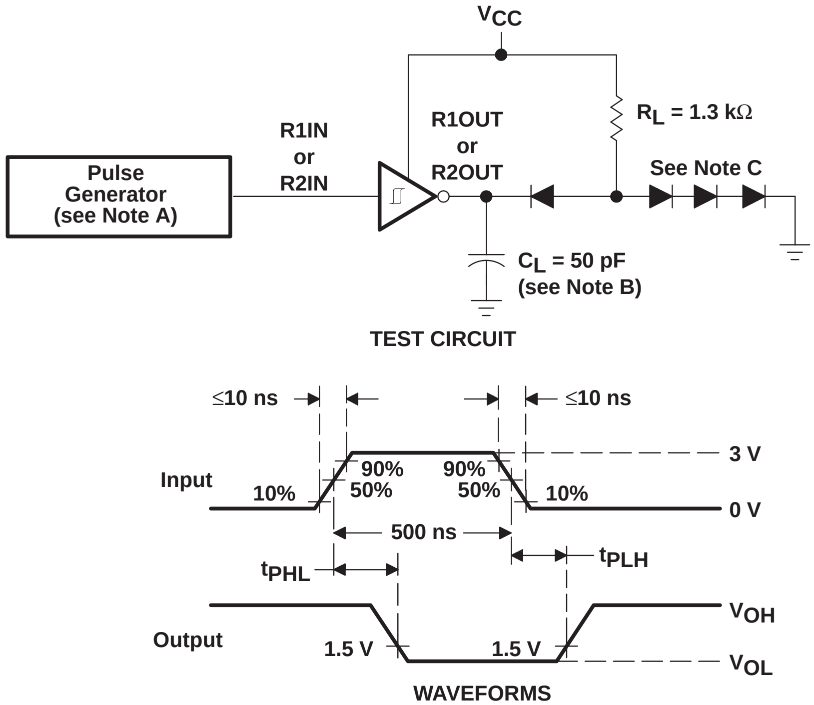
† All typical values are at V_{CC} = 5 V, T_A = 25°C.
‡ The algebraic convention, in which the least-positive (most negative) value is designated minimum, is used in this data sheet for logic voltage levels only.
NOTE 4: Test conditions are C1−C4 = 1 µF at V_{CC} = 5 V ± 0.5 V.

switching characteristics, V_{CC} = 5 V, T_A = 25°C (see Note 4 and Figure 1)

PARAMETER		TYP	UNIT
$t_{PLH(R)}$	Receiver propagation delay time, low- to high-level output	500	ns
$t_{PHL(R)}$	Receiver propagation delay time, high- to low-level output	500	ns

NOTE 4: Test conditions are C1−C4 = 1 µF at V_{CC} = 5 V ± 0.5 V.

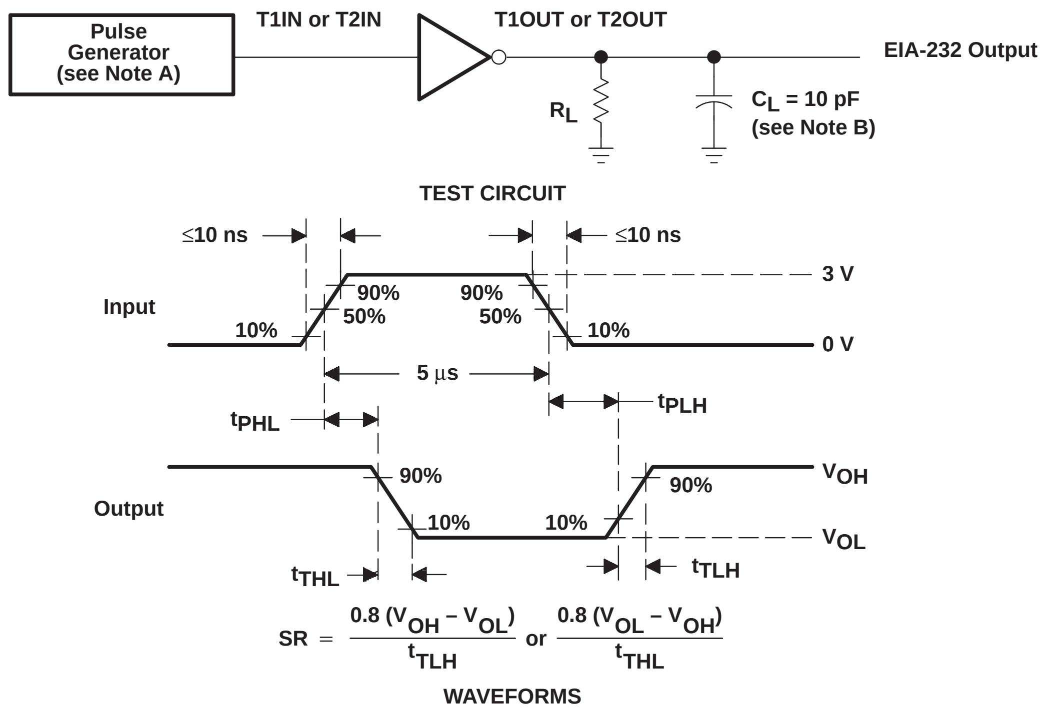
PARAMETER MEASUREMENT INFORMATION



NOTES: A. The pulse generator has the following characteristics: Z_O = 50 Ω, duty cycle ≤ 50%.
B. C_L includes probe and jig capacitance.
C. All diodes are 1N3064 or equivalent.

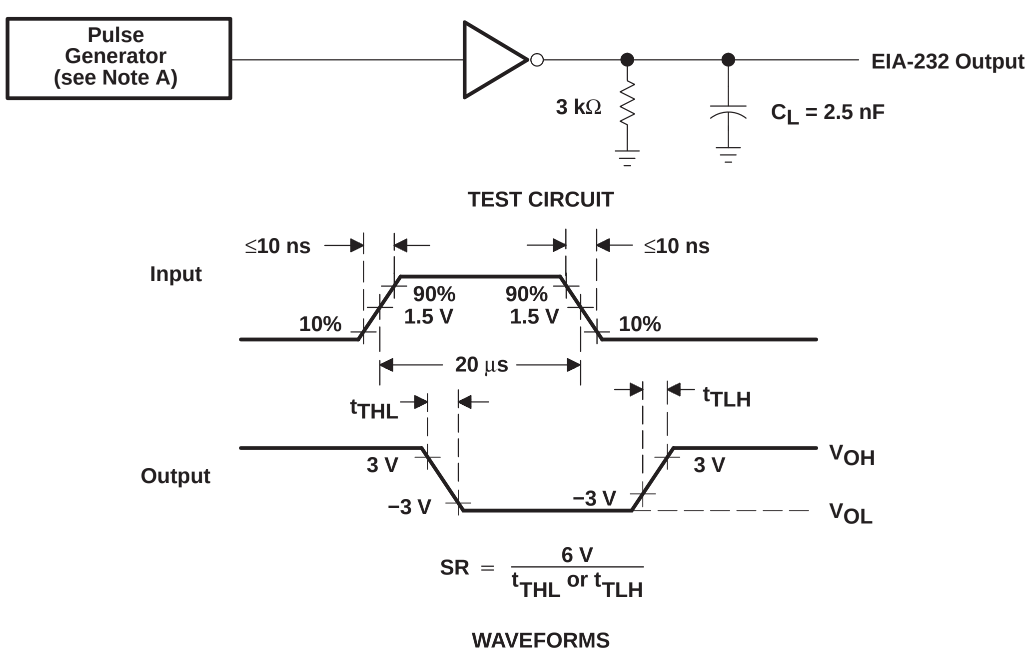
Figure 1. Receiver Test Circuit and Waveforms for t_{PHL} and t_{PLH} Measurements

PARAMETER MEASUREMENT INFORMATION



NOTES: A. The pulse generator has the following characteristics: Z_O = 50 Ω, duty cycle ≤ 50%.
B. C_L includes probe and jig capacitance.

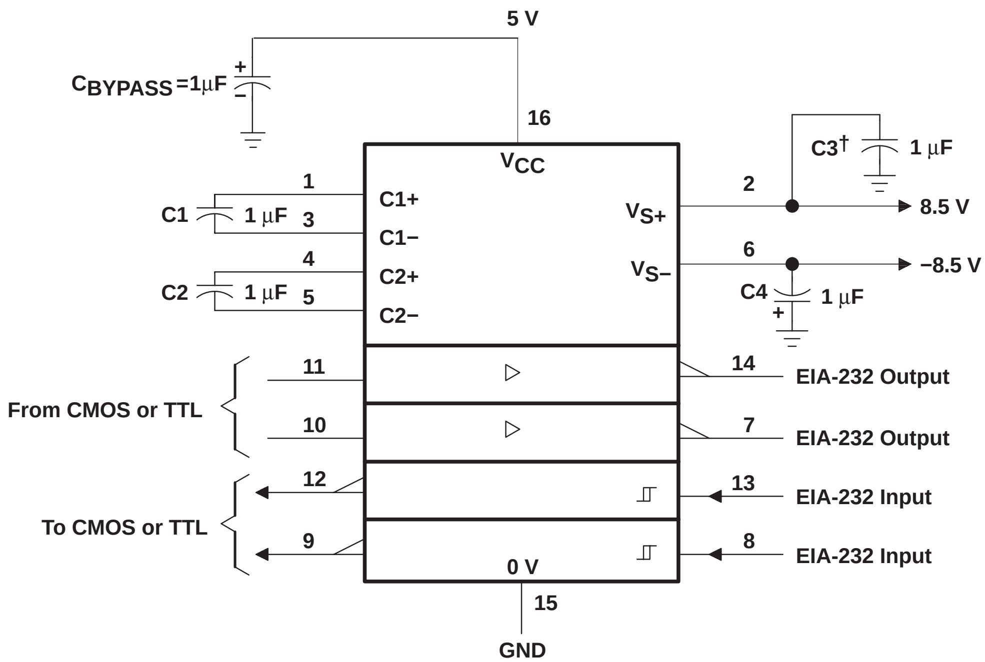
Figure 2. Driver Test Circuit and Waveforms for t_{PHL} and t_{PLH} Measurements (5-µs Input)



NOTE A: The pulse generator has the following characteristics: Z_O = 50 Ω, duty cycle ≤ 50%.

Figure 3. Test Circuit and Waveforms for t_{THL} and t_{TLH} Measurements (20-µs Input)

APPLICATION INFORMATION



† C3 can be connected to V_{CC} or GND.

NOTES: A. Resistor values shown are nominal.

B. Nonpolarized ceramic capacitors are acceptable. If polarized tantalum or electrolytic capacitors are used, they should be connected as shown. In addition to the 1-µF capacitors shown, the MAX202 can operate with 0.1-µF capacitors.

Figure 4. Typical Operating Circuit

PACKAGING INFORMATION

Orderable Device	Status (1)	Package Type	Package Drawing	Pins	Package Qty	Eco Plan (2)	Lead/Ball Finish	MSL Peak Temp (3)
MAX232D	ACTIVE	SOIC	D	16	40	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232DE4	ACTIVE	SOIC	D	16	40	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232DR	ACTIVE	SOIC	D	16	2500	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232DRE4	ACTIVE	SOIC	D	16	2500	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232DW	ACTIVE	SOIC	DW	16	40	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232DWE4	ACTIVE	SOIC	DW	16	40	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232DWR	ACTIVE	SOIC	DW	16	2000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232DWRE4	ACTIVE	SOIC	DW	16	2000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232ID	ACTIVE	SOIC	D	16	40	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232IDE4	ACTIVE	SOIC	D	16	40	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232IDR	ACTIVE	SOIC	D	16	2500	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232IDRE4	ACTIVE	SOIC	D	16	2500	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232IDW	ACTIVE	SOIC	DW	16	40	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232IDWE4	ACTIVE	SOIC	DW	16	40	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232IDWG4	ACTIVE	SOIC	DW	16	40	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232IDWR	ACTIVE	SOIC	DW	16	2000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232IDWRE4	ACTIVE	SOIC	DW	16	2000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232IDWRG4	ACTIVE	SOIC	DW	16	2000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232IN	ACTIVE	PDIP	N	16	25	Pb-Free (RoHS)	CU NIPDAU	N / A for Pkg Type
MAX232INE4	ACTIVE	PDIP	N	16	25	Pb-Free (RoHS)	CU NIPDAU	N / A for Pkg Type
MAX232N	ACTIVE	PDIP	N	16	25	Pb-Free (RoHS)	CU NIPDAU	N / A for Pkg Type
MAX232NE4	ACTIVE	PDIP	N	16	25	Pb-Free (RoHS)	CU NIPDAU	N / A for Pkg Type
MAX232NSR	ACTIVE	SO	NS	16	2000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232NSRE4	ACTIVE	SO	NS	16	2000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM

(1) The marketing status values are defined as follows:

ACTIVE: Product device recommended for new designs.
LIFEBUY: TI has announced that the device will be discontinued, and a lifetime-buy period is in effect.
NRND: Not recommended for new designs. Device is in production to support existing customers, but TI does not recommend using this part in a new design.
PREVIEW: Device has been announced but is not in production. Samples may or may not be available.
OBSOLETE: TI has discontinued the production of the device.

(2) Eco Plan - The planned eco-friendly classification: Pb-Free (RoHS), Pb-Free (RoHS Exempt), or Green (RoHS & no Sb/Br) - please check http://www.ti.com/productcontent for the latest availability information and additional product content details.
TBD: The Pb-Free/Green conversion plan has not been defined.
Pb-Free (RoHS): TI's terms "Lead-Free" or "Pb-Free" mean semiconductor products that are compatible with the current RoHS requirements for all 6 substances, including the requirement that lead not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, TI Pb-Free products are suitable for use in specified lead-free processes.
Pb-Free (RoHS Exempt): This component has a RoHS exemption for either 1) lead-based flip-chip solder bumps used between the die and package, or 2) lead-based die adhesive used between the die and leadframe. The component is otherwise considered Pb-Free (RoHS compatible) as defined above.
Green (RoHS & no Sb/Br): TI defines "Green" to mean Pb-Free (RoHS compatible), and free of Bromine (Br) and Antimony (Sb) based flame retardants (Br or Sb do not exceed 0.1% by weight in homogeneous material)

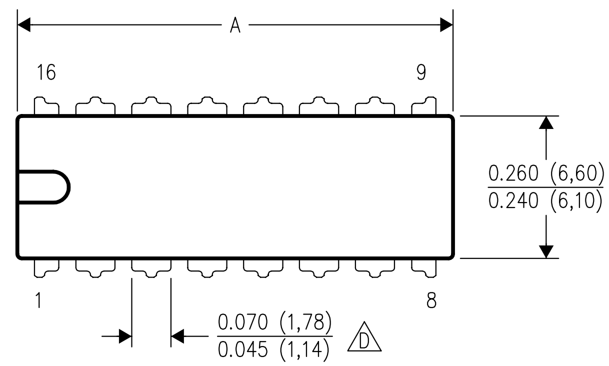
(3) MSL, Peak Temp. -- The Moisture Sensitivity Level rating according to the JEDEC industry standard classifications, and peak solder temperature.

Important Information and Disclaimer: The information provided on this page represents TI's knowledge and belief as of the date that it is provided. TI bases its knowledge and belief on information provided by third parties, and makes no representation or warranty as to the accuracy of such information. Efforts are underway to better integrate information from third parties. TI has taken and continues to take reasonable steps to provide representative and accurate information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. TI and TI suppliers consider certain information to be proprietary, and thus CAS numbers and other limited information may not be available for release.

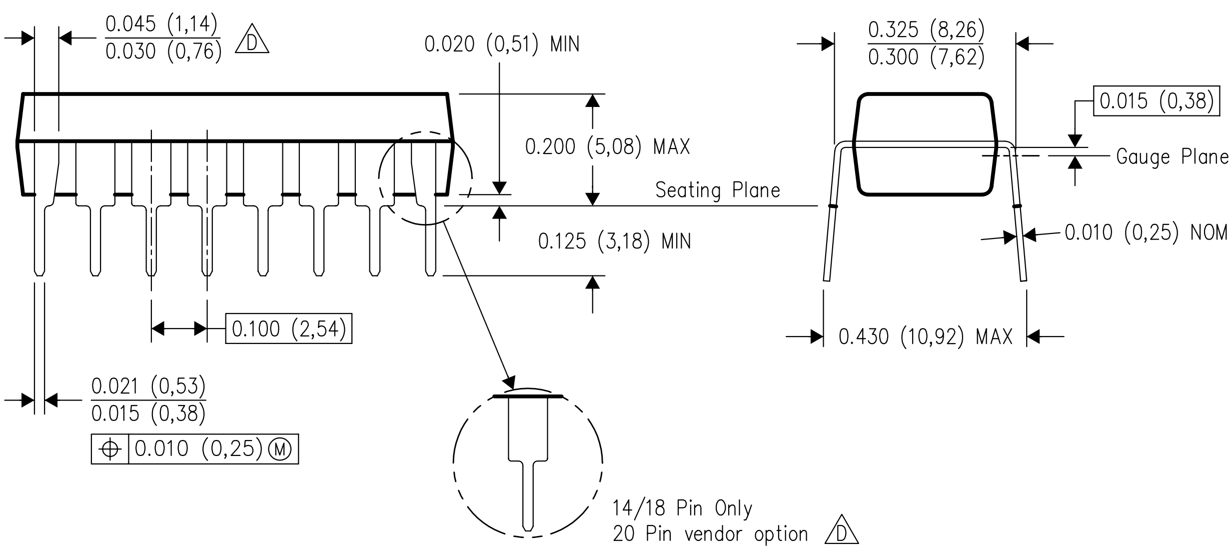
In no event shall TI's liability arising out of such information exceed the total purchase price of the TI part(s) at issue in this document sold by TI to Customer on an annual basis.

N (R–PDIP–T**) PLASTIC DUAL–IN–LINE PACKAGE

16 PINS SHOWN



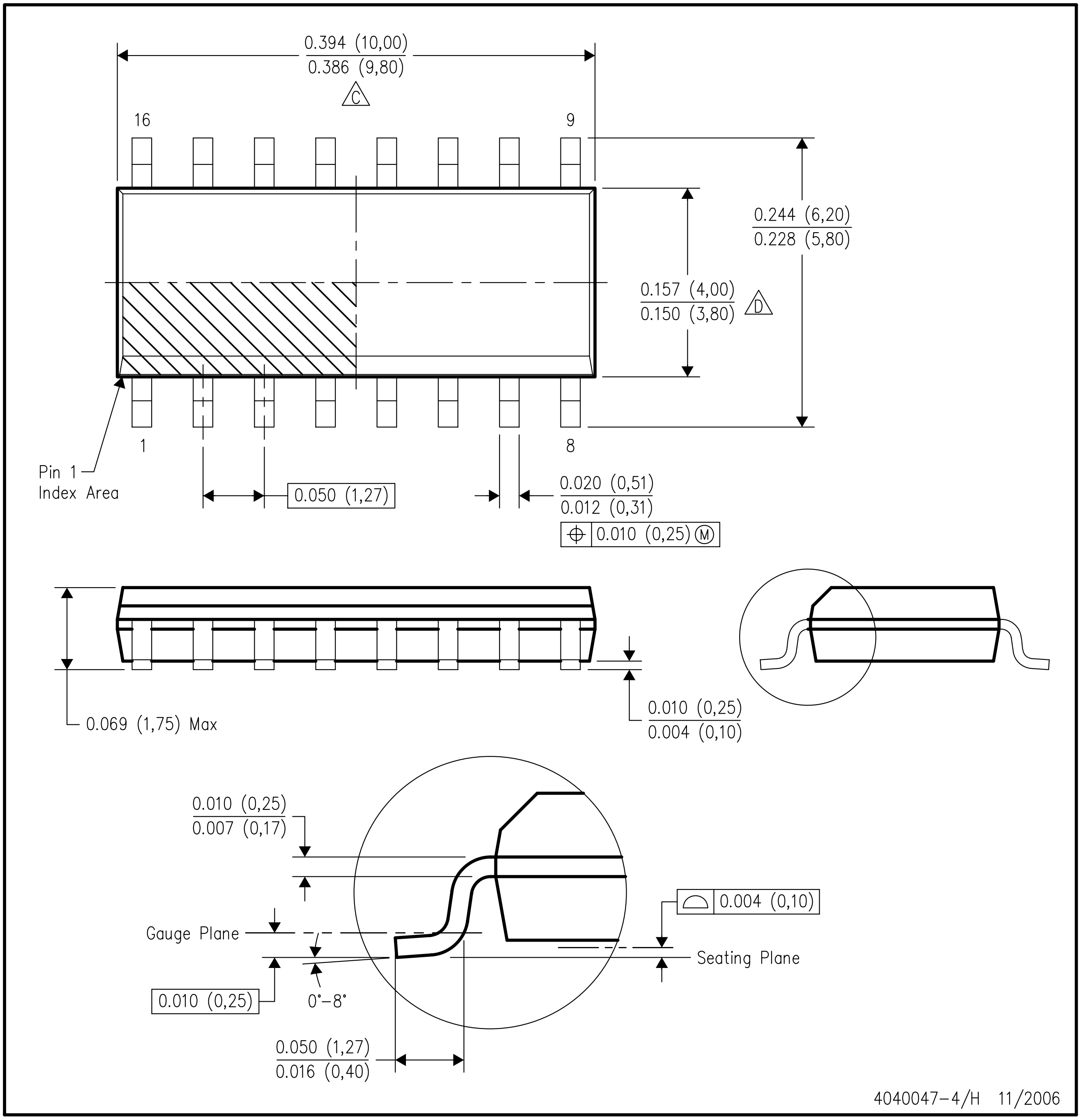
DIM \ PINS **	14	16	18	20
A MAX	0.775 (19,69)	0.775 (19,69)	0.920 (23,37)	1.060 (26,92)
A MIN	0.745 (18,92)	0.745 (18,92)	0.850 (21,59)	0.940 (23,88)
C MS–001 VARIATION	AA	BB	AC	AD



4040049/E 12/2002

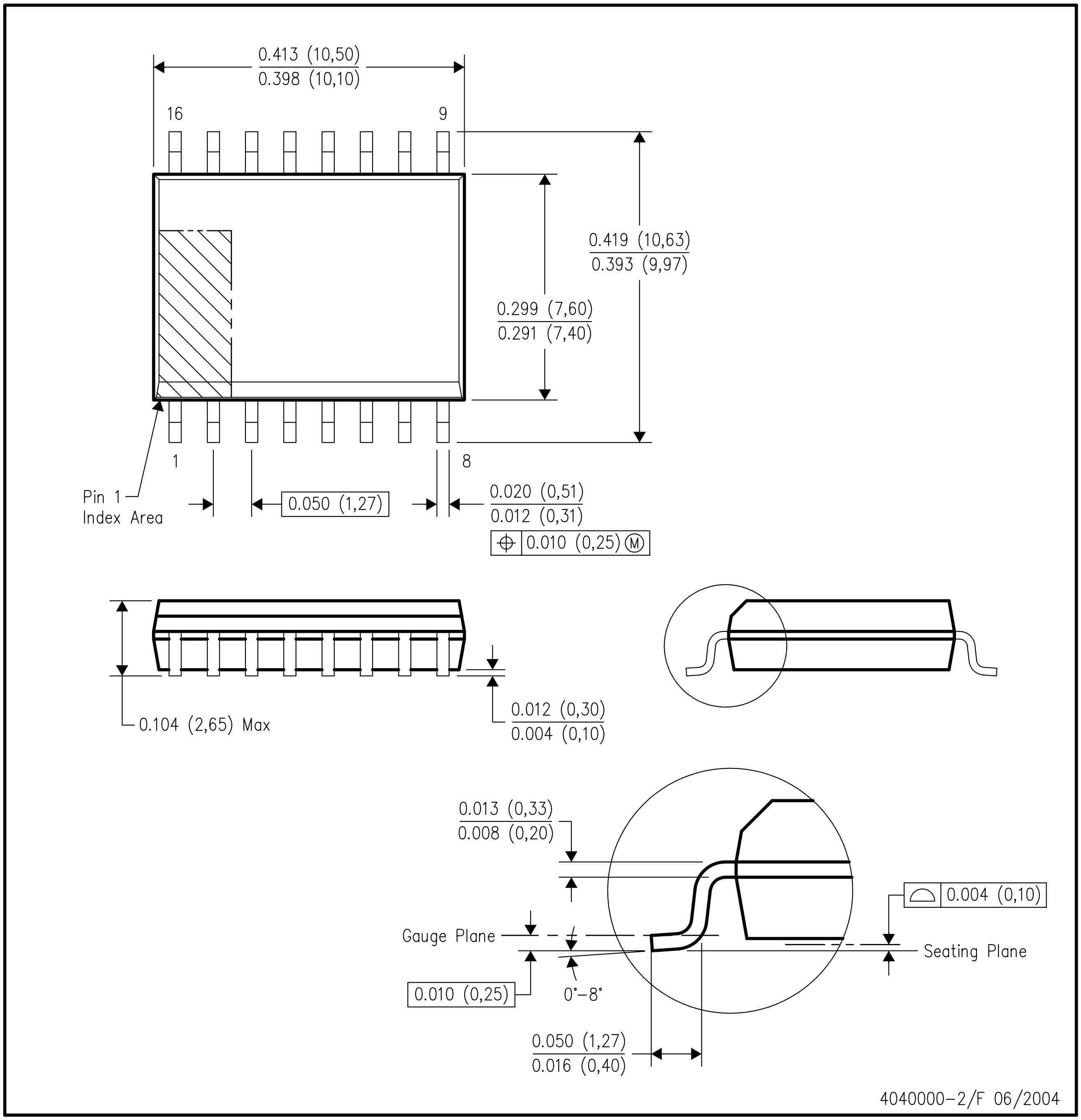
NOTES: A. All linear dimensions are in inches (millimeters).
B. This drawing is subject to change without notice.
C. Falls within JEDEC MS–001, except 18 and 20 pin minimum body length (Dim A).
D. The 20 pin end lead shoulder width is a vendor option, either half or full width.

D (R–PDSO–G16) PLASTIC SMALL–OUTLINE PACKAGE



NOTES: A. All linear dimensions are in inches (millimeters).
B. This drawing is subject to change without notice.
C. Body length does not include mold flash, protrusions, or gate burrs. Mold flash, protrusions, or gate burrs shall not exceed .006 (0,15) per end.
D. Body width does not include interlead flash. Interlead flash shall not exceed .017 (0,43) per side.
E. Reference JEDEC MS–012 variation AC.

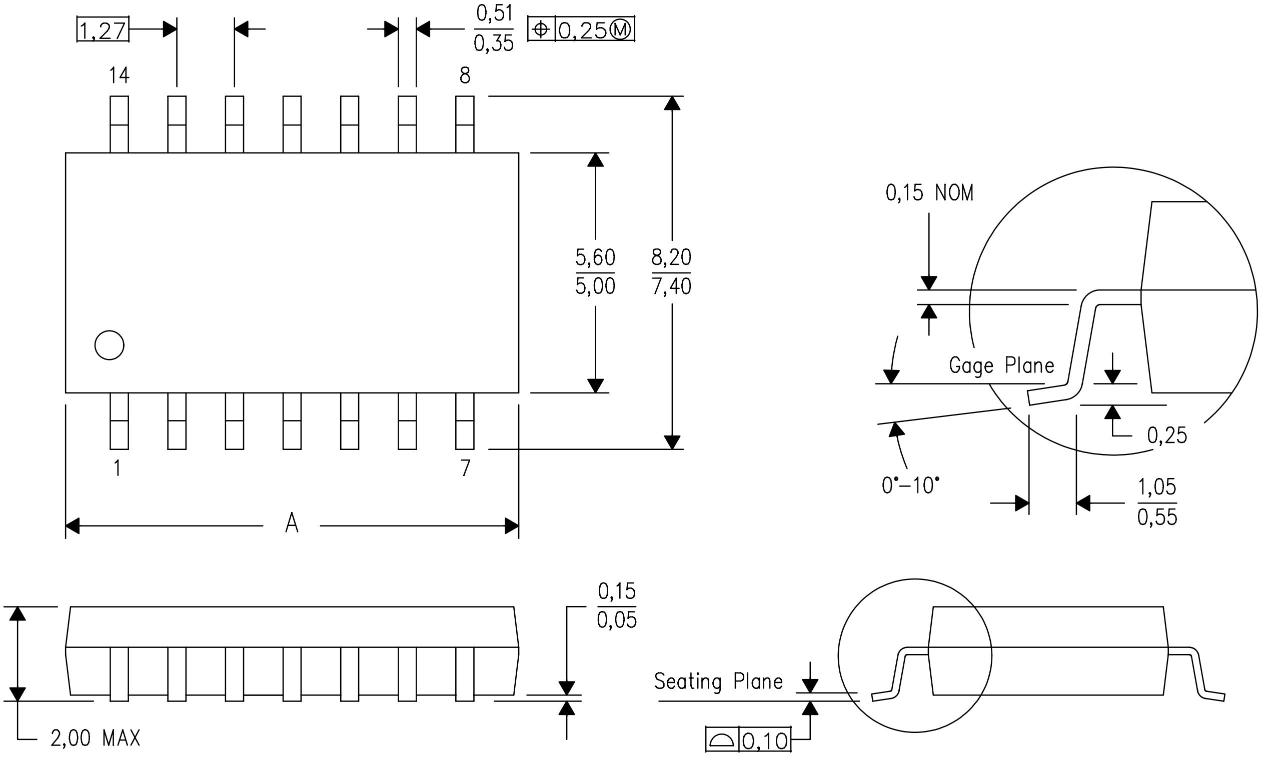
DW (R–PDSO–G16) PLASTIC SMALL–OUTLINE PACKAGE



NOTES:
- A. All linear dimensions are in inches (millimeters).
- B. This drawing is subject to change without notice.
- C. Body dimensions do not include mold flash or protrusion not to exceed 0.006 (0,15).
- D. Falls within JEDEC MS–013 variation AA.

NS (R-PDSO-G)** **PLASTIC SMALL-OUTLINE PACKAGE**

14-PINS SHOWN



PINS ** / DIM	14	16	20	24
A MAX	10,50	10,50	12,90	15,30
A MIN	9,90	9,90	12,30	14,70

4040062/C 03/03

NOTES:
- A. All linear dimensions are in millimeters.
- B. This drawing is subject to change without notice.
- C. Body dimensions do not include mold flash or protrusion, not to exceed 0,15.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
Low Power Wireless	www.ti.com/lpw	Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265

Features

- **Compatible with MCS-51™ Products**
- **2K Bytes of Reprogrammable Flash Memory**
  - **Endurance: 1,000 Write/Erase Cycles**
- **2.7V to 6V Operating Range**
- **Fully Static Operation: 0 Hz to 24 MHz**
- **Two-level Program Memory Lock**
- **128 x 8-bit Internal RAM**
- **15 Programmable I/O Lines**
- **Two 16-bit Timer/Counters**
- **Six Interrupt Sources**
- **Programmable Serial UART Channel**
- **Direct LED Drive Outputs**
- **On-chip Analog Comparator**
- **Low-power Idle and Power-down Modes**

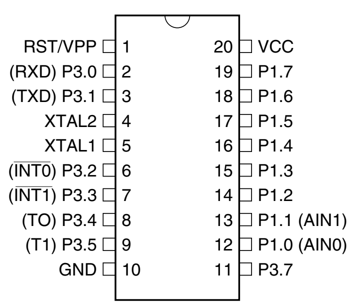
Description

The AT89C2051 is a low-voltage, high-performance CMOS 8-bit microcomputer with 2K bytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard MCS-51 instruction set. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C2051 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications.

The AT89C2051 provides the following standard features: 2K bytes of Flash, 128 bytes of RAM, 15 I/O lines, two 16-bit timer/counters, a five vector two-level interrupt architecture, a full duplex serial port, a precision analog comparator, on-chip oscillator and clock circuitry. In addition, the AT89C2051 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port and interrupt system to continue functioning. The power-down mode saves the RAM contents but freezes the oscillator disabling all other chip functions until the next hardware reset.

Pin Configuration

PDIP/SOIC

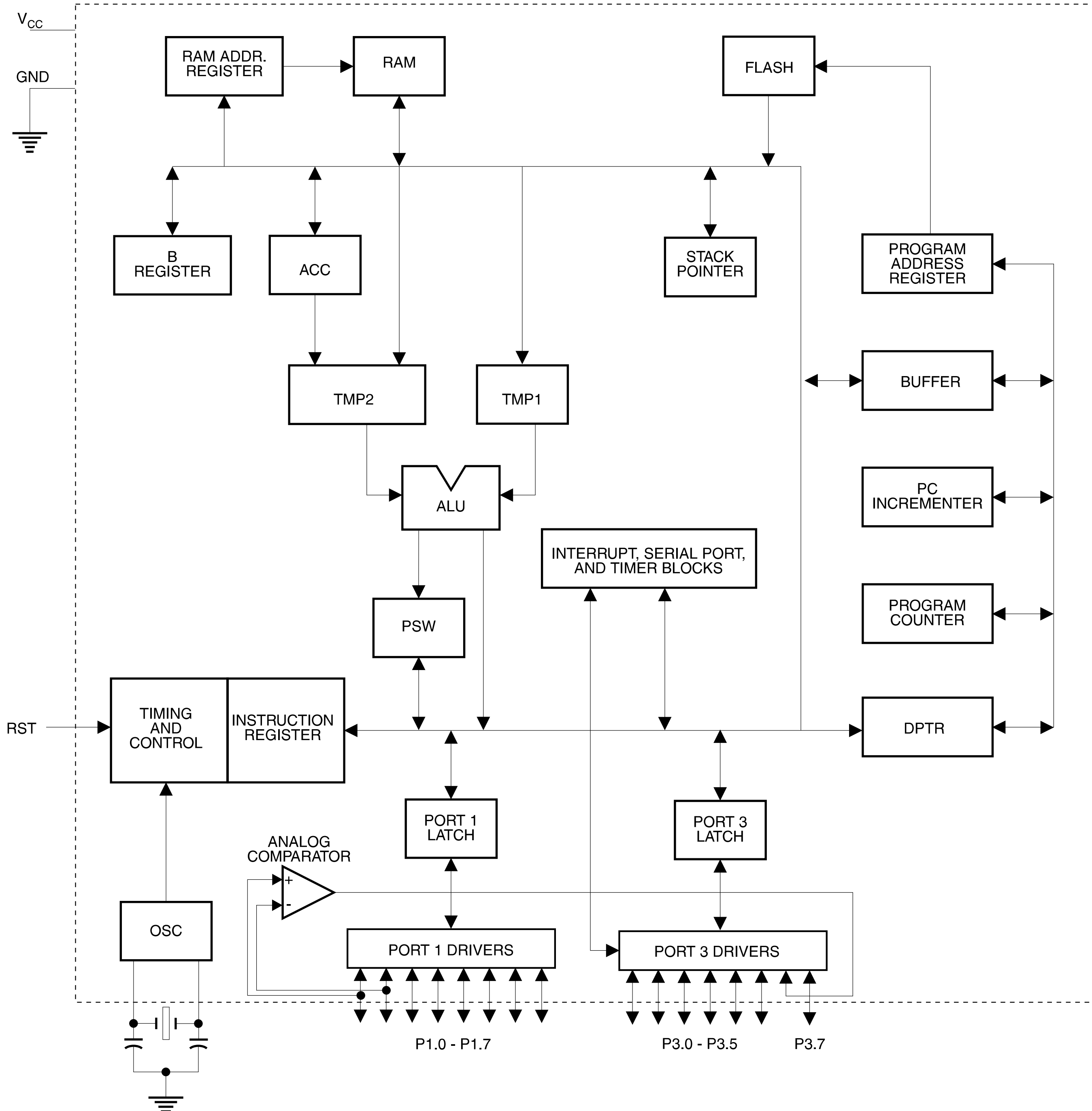


8-bit Microcontroller with 2K Bytes Flash

AT89C2051

Rev. 0368E–02/00

Block Diagram



Pin Description

VCC

Supply voltage.

GND

Ground.

Port 1

Port 1 is an 8-bit bi-irectional I/O port. Port pins P1.2 to P1.7 provide internal pullups. P1.0 and P1.1 require external pullups. P1.0 and P1.1 also serve as the positive input (AIN0) and the negative input (AIN1), respectively, of the on-chip precision analog comparator. The Port 1 output buffers can sink 20 mA and can drive LED displays directly. When 1s are written to Port 1 pins, they can be used as inputs. When pins P1.2 to P1.7 are used as inputs and are externally pulled low, they will source current (I_{IL}) because of the internal pullups.

Port 1 also receives code data during Flash programming and verification.

Port 3

Port 3 pins P3.0 to P3.5, P3.7 are seven bi-irectional I/O pins with internal pullups. P3.6 is hard-wired as an input to the output of the on-chip comparator and is not accessible as a general purpose I/O pin. The Port 3 output buffers can sink 20 mA. When 1s are written to Port 3 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pullups.

Port 3 also serves the functions of various special features of the AT89C2051 as listed below:

Port 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)

Port 3 also receives some control signals for Flash programming and verification.

RST

Reset input. All I/O pins are reset to 1s as soon as RST goes high. Holding the RST pin high for two machine cycles while the oscillator is running resets the device.

Each machine cycle takes 12 oscillator or clock cycles.

XTAL1

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

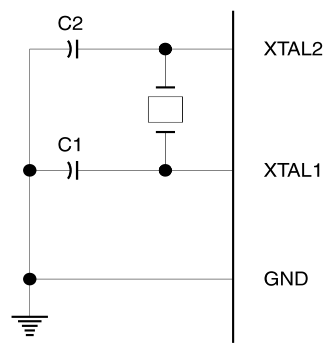
XTAL2

Output from the inverting oscillator amplifier.

Oscillator Characteristics

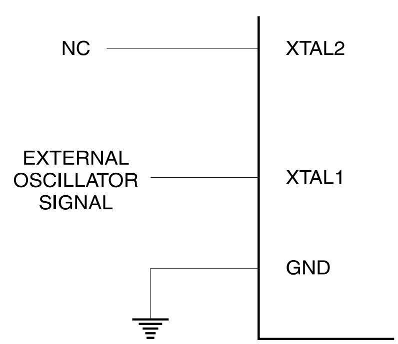
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 2. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Figure 1. Oscillator Connections



Note: C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

Figure 2. External Clock Drive Configuration



Special Function Registers

A map of the on-chip memory area called the Special Function Register (SFR) space is shown in the table below.

Note that not all of the addresses are occupied, and unoccupied addresses may not be implemented on the chip. Read accesses to these addresses will in general return random data, and write accesses will have an indeterminate effect.

User software should not write 1s to these unlisted locations, since they may be used in future products to invoke new features. In that case, the reset or inactive values of the new bits will always be 0.

Table 1. AT89C2051 SFR Map and Reset Values

0F8H									0FFH
0F0H	B 00000000								0F7H
0E8H									0EFH
0E0H	ACC 00000000								0E7H
0D8H									0DFH
0D0H	PSW 00000000								0D7H
0C8H									0CFH
0C0H									0C7H
0B8H	IP XXX00000								0BFH
0B0H	P3 11111111								0B7H
0A8H	IE 0XX00000								0AFH
0A0H									0A7H
98H	SCON 00000000	SBUF XXXXXXXX							9FH
90H	P1 11111111								97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000			8FH
80H		SP 00000111	DPL 00000000	DPH 00000000				PCON 0XXX0000	87H

Restrictions on Certain Instructions

The AT89C2051 and is an economical and cost-effective member of Atmel's growing family of microcontrollers. It contains 2K bytes of flash program memory. It is fully compatible with the MCS-51 architecture, and can be programmed using the MCS-51 instruction set. However, there are a few considerations one must keep in mind when utilizing certain instructions to program this device.

All the instructions related to jumping or branching should be restricted such that the destination address falls within the physical program memory space of the device, which is 2K for the AT89C2051. This should be the responsibility of the software programmer. For example, LJMP 7E0H would be a valid instruction for the AT89C2051 (with 2K of memory), whereas LJMP 900H would not.

1. Branching instructions:

LCALL, LJMP, ACALL, AJMP, SJMP, JMP @A+DPTR

These unconditional branching instructions will execute correctly as long as the programmer keeps in mind that the destination branching address must fall within the physical boundaries of the program memory size (locations 00H to 7FFH for the 89C2051). Violating the physical space limits may cause unknown program behavior.

CJNE [...], DJNZ [...], JB, JNB, JC, JNC, JBC, JZ, JNZ With these conditional branching instructions the same rule above applies. Again, violating the memory boundaries may cause erratic execution.

For applications involving interrupts the normal interrupt service routine address locations of the 80C51 family architecture have been preserved.

2. MOVX-related instructions, Data Memory:

The AT89C2051 contains 128 bytes of internal data memory. Thus, in the AT89C2051 the stack depth is limited to 128 bytes, the amount of available RAM. External DATA memory access is not supported in this device, nor is external PROGRAM memory execution. Therefore, no MOVX [...] instructions should be included in the program.

A typical 80C51 assembler will still assemble instructions, even if they are written in violation of the restrictions mentioned above. It is the responsibility of the controller user to know the physical features and limitations of the device being used and adjust the instructions used correspondingly.

Program Memory Lock Bits

On the chip are two lock bits which can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the table below:

Lock Bit Protection Modes[1]

Program Lock Bits			
	LB1	LB2	Protection Type
1	U	U	No program lock features.
2	P	U	Further programming of the Flash is disabled.
3	P	P	Same as mode 2, also verify is disabled.

Note: 1. The Lock Bits can only be erased with the Chip Erase operation.

Idle Mode

In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

P1.0 and P1.1 should be set to “0” if no external pullups are used, or set to “1” if external pullups are used.

It should be noted that when idle is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

Power-down Mode

In the power down mode the oscillator is stopped, and the instruction that invokes power down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power down mode is terminated. The only exit from power down is a hardware reset. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

P1.0 and P1.1 should be set to “0” if no external pullups are used, or set to “1” if external pullups are used.

Programming The Flash

The AT89C2051 is shipped with the 2K bytes of on-chip PEROM code memory array in the erased state (i.e., contents = FFH) and ready to be programmed. The code memory array is programmed one byte at a time. *Once the array is programmed, to re-program any non-blank byte, the entire memory array needs to be erased electrically.*

Internal Address Counter: The AT89C2051 contains an internal PEROM address counter which is always reset to 000H on the rising edge of RST and is advanced by applying a positive going pulse to pin XTAL1.

Programming Algorithm: To program the AT89C2051, the following sequence is recommended.

1. Power-up sequence:
   Apply power between V_{CC} and GND pins
   Set RST and XTAL1 to GND
2. Set pin RST to “H”
   Set pin P3.2 to “H”
3. Apply the appropriate combination of “H” or “L” logic levels to pins P3.3, P3.4, P3.5, P3.7 to select one of the programming operations shown in the PEROM Programming Modes table.

To Program and Verify the Array:

4. Apply data for Code byte at location 000H to P1.0 to P1.7.
5. Raise RST to 12V to enable programming.
6. Pulse P3.2 once to program a byte in the PEROM array or the lock bits. The byte-write cycle is self-timed and typically takes 1.2 ms.
7. To verify the programmed data, lower RST from 12V to logic “H” level and set pins P3.3 to P3.7 to the appropiate levels. Output data can be read at the port P1 pins.
8. To program a byte at the next address location, pulse XTAL1 pin once to advance the internal address counter. Apply new data to the port P1 pins.
9. Repeat steps 5 through 8, changing data and advancing the address counter for the entire 2K bytes array or until the end of the object file is reached.
10. Power-off sequence:
    set XTAL1 to “L”
    set RST to “L”
    Turn V_{CC} power off

$\overline{\text{Data}}$ Polling: The AT89C2051 features $\overline{\text{Data}}$ Polling to indicate the end of a write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written data on P1.7. Once the write cycle has been completed, true data is valid on all outputs, and the next cycle may begin. $\overline{\text{Data}}$ Polling may begin any time after a write cycle has been initiated.

Ready/$\overline{\text{Busy}}$: The Progress of byte programming can also be monitored by the RDY/$\overline{\text{BSY}}$ output signal. Pin P3.1 is pulled low after P3.2 goes High during programming to indicate BUSY. P3.1 is pulled High again when programming is done to indicate READY.

Program Verify: If lock bits LB1 and LB2 have not been programmed code data can be read back via the data lines for verification:

1. Reset the internal address counter to 000H by bringing RST from “L” to “H”.
2. Apply the appropriate control signals for Read Code data and read the output data at the port P1 pins.
3. Pulse pin XTAL1 once to advance the internal address counter.
4. Read the next code data byte at the port P1 pins.
5. Repeat steps 3 and 4 until the entire array is read.

The lock bits cannot be verified directly. Verification of the lock bits is achieved by observing that their features are enabled.

Chip Erase: The entire PEROM array (2K bytes) and the two Lock Bits are erased electrically by using the proper combination of control signals and by holding P3.2 low for 10 ms. The code array is written with all “1”s in the Chip Erase operation and must be executed before any non-blank memory byte can be re-programmed.

Reading the Signature Bytes: The signature bytes are read by the same procedure as a normal verification of locations 000H, 001H, and 002H, except that P3.5 and P3.7 must be pulled to a logic low. The values returned are as follows.

(000H) = 1EH indicates manufactured by Atmel
(001H) = 21H indicates 89C2051

Programming Interface

Every code byte in the Flash array can be written and the entire array can be erased by using the appropriate combination of control signals. The write operation cycle is self-timed and once initiated, will automatically time itself to completion.

All major programming vendors offer worldwide support for the Atmel microcontroller series. Please contact your local programming vendor for the appropriate software revision.