Plant Health Monitoring Using Wireless Technology

Tejas G. Mahale, Sanath S. Shetty, Salman B. Bandri, Gyanchandra Gupta, Prof. Gitanjali Korgaonkar

Department of Electronics and Telecommunication
Padmabhushan Vasantdada Patil Pratishthan's College of Engineering
Vasantdada Patil Educational Complex, Eastern Express Highway, Near
Everard Nagar, Sion, Chunabhatti, Mumbai – 400022

Abstract- An automated system for measuring plant leaf color, as an indicator of plant health status, has been developed for plantlets growing in a modified micro propagation system. Using a custom builthand device, sensors located on a pan and tilt system at the end of the device monitor plant growth and the ambient growing environment. Sensors include a compact color zoom camera, RGB (red, green and blue) color sensors. Leaf color sensors provide information, in a non-destructive manner, on the health status of tissue by comparing the sensor outputs to pre-determined optimum values. These low cost color sensors can be incorporated into a continuous automated system for monitoring leaf color of growing plants. Subtle color changes can be an early indication of stress from less than optimum nutrient concentrations. When combined with automated image sensing for growth analysis, and environmental sensing (RH, CO2 and temperature) in a controlled environment, optimized rapid growth with minimal human input can be achieved using a modified micro propagation system. In this project we detail the calibration technique for a RGB sensor and compare it with a high end spectrophotometer.

Index Terms- Wireless, Plant Health, Zigbee, Color sensor

I. Introduction

From the recent years, robotics has turned out to be an ingredient over which many people had shown their interest. Robotics has gained popularity due to the advancement of many technologies of computing and nano technologies. We have decided to design something that can make humans life easier and comfortable. Here our interest of review is to make a fully automated fire fighting robot which can help in dealing with many fire problems in households and small scale industries. Now we are not concentrating on making a firefighting robot that can deal with fire on large scale because we want to proceed step by step. The need of the hour is to make a device which can detect fire, even if it is small and take the necessary action to put it off.

2.1 Aim of the project

Aim of project is to design electronic device which is helpful for monitoring plant health status. This electronic device will work in remote area with help of wireless technology such as zigbee. With help of this device, we can prepare database of various stages of plant, so we can monitor plant growth.

2.2 Organization of Report

In chapter 2 we discuss the concepts involved in robotics and wireless technology. The chapter 3 deals with the implementation of speech recognition part. In chapter 4 we have discussed about software results obtained. In chapter 5 we have given conclusion and future scope of project.

II. Literature Review

Automated systems are becoming increasingly common in all economic sectors. In the past decade there has been a push towards more automation in the horticulture industry, and it has become more sophisticated and reliable, that we are beginning to see them used to undertake routine, often repetitive tasks, which are expensive to do using a highly paid labor force. With rapid strides in technological advancement, more and more applications have become possible**.** The increasing demands for crop production and quality have significantly increased the utilization of high quality and productivity green houses. Modern greenhouses are nowadays having great sizes and they are equipped with sophisticated monitoring and controlled actuation systems to assure maximum productivity and provide value-added quality. Increases in greenhouse sizes has increased the demand in increasing the monitoring points in order to provide real-time precise measurement of some important parameters and hence to avoid unnecessary exposure to unhealthy ambient conditions. The increase of monitoring points is certainly leads to increase the complexity of managing and maintaining them efficiently. Advances in electronics have empowered ***e***ngineers to build automatic systems that are capable of operating in unstructured environments .

In this project we are building an low cost automated system for farmers to increase the productivity of their farms and reduce the labor costs. By using color sensor for measuring plant leaf color, as an indicator of plant health status. Leaf color sensors provide information, in a non-destructive manner, on the health status of tissue by comparing the sensor outputs to pre-determined optimum values.

These low cost color sensors can be incorporated into a continuous automated system for monitoring leaf color of growing plants. Subtle color changes can be an early indication of stress from less than optimum nutrient concentrations. It can identify 16 million color shades giving RGB value for the detected color. The system consists of a number of local stations and a central station. The local stations are used to measure the environmental parameters and to control the operation of controlled actuators to maintain climate parameters at predefined set points. For each local station a PIC Microcontroller is used to store the instant values of the environmental parameters, send them to the central station and receive the control signals that are required for the operation of the actuators. The communication between the local stations and the central station is achieved via ZigBee wireless modules. Here the local station is the builthand device and central station is the laptop. ZigBee is used because these devices can transmit data over long distances. The color sensor gives RGB information as to microcontroller and it is displayed on the LCD display screen. This information is given t the ZigBee device through which the information is send to control room.

Previous work done

- There was also some work done in New Zealand by John Seelay and group on the same project.
- In John's work the setup, they used Robot with sensor arm instead of hand device with LCD.
- The reported system used an autonomous robotic arm containing RGB (red, green and blue) color, environmental and proximity sensors as well as a compact color camera

Wireless communication is the transfer of information between two or more points that are not connected by an electrical conductor.

The most common wireless technologies use radio. With radio waves distances can be short, such as a few meters for television or as far as thousands or even millions of kilometers for deep-space radio communications. It encompasses various types of fixed, mobile, and portable applications, including two-way radios, cellular telephones, personal digital assistants (PDAs), and wireless networking. Other examples of applications of radio *wireless technology* include GPS units, garage door openers, wireless computer mice, keyboards and headsets, headphones, radio receivers, satellite television, broadcast television and cordless telephones.

Somewhat less common methods of achieving wireless communications include the use of other electromagnetic wireless technologies, such as light, magnetic, or electric fields or the use of sound.

Wireless Technology

Modes of wireless communication

Wireless communications can be via:

Radio

Radio communication, microwave communication, for example long-range line-of-sight via highly directional antennas, or short-range communication,

Free-space optical

Free-space optical communication (FSO) is an optical communication technology that uses light propagating in free space to wirelessly transmit data for telecommunications or computer networking. "Free space" means the light beams travel through the open air or outer space. This contrasts with other communication technologies that use light beams traveling through transmission lines such as optical fiber or dielectric "light pipes".

The technology is useful where physical connections are impractical due to high costs or other considerations. For example free space optical links are used in cities between office buildings which are not wired for networking, where the cost of running cable through the building and under the street would be prohibitive.

Conclusion

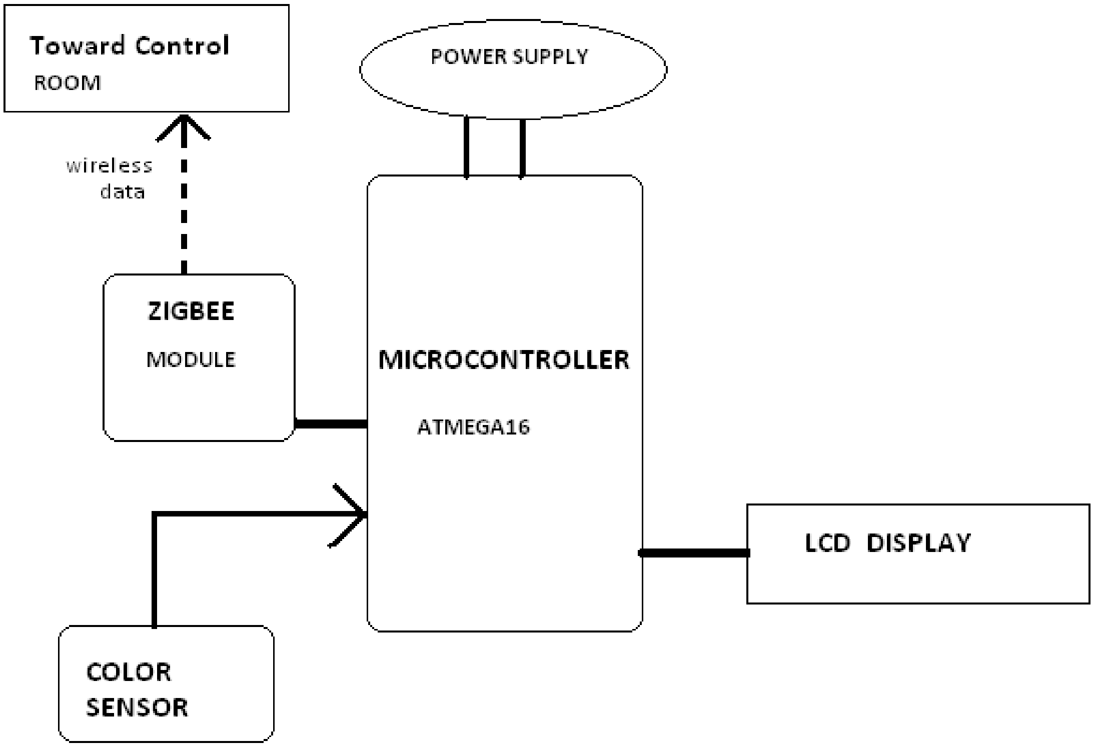
There are many wireless technologies available in today's world and also there is a large scope of development in the field of robotics. Combining these two fields can result in reducing human effort to a great extent.

Previous work done

- There was also some work done in New Zealand by John Seelay and group on the same project.
- In John's work the setup, they used Robot with sensor arm instead of hand device with LCD.
- The reported system used an autonomous robotic arm containing RGB (red, green and blue) color, environmental and proximity sensors as well as a compact color camera

III. SYSTEM ANALYSIS

4.1 BLOCK DIAGRAM



4.2 Block Diagram Description:-

The following are the brief explanations of the working principle of the various major blocks or sections used in the system.

A. Microcontroller ATmega32 :

The ATmega32 is a low-power 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega32 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed**.**

B. Color Sensor :

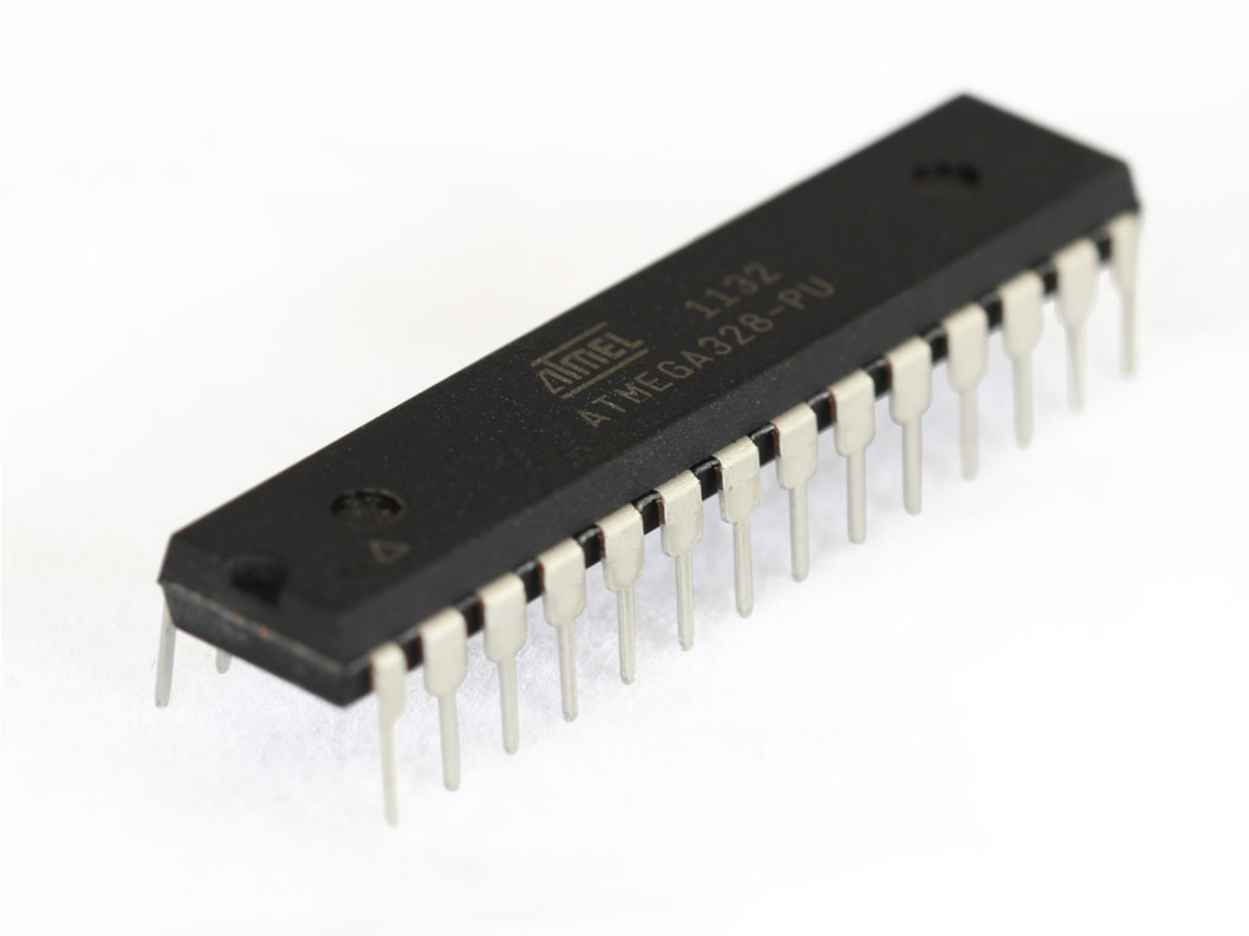
This color sensor identifies color and gives serial output of RBG value. The sensor switches each primary color RGB, one by one and checks what intensity of color is reflected by the surface of detection. This reflected intensity is converted to 8 bit value. For example a RED surface will strongly reflect RED. While a Yellow surface will reflect RED and GREEN both. According to the induction principle of the three primary colors which create various other colors in nature, once the value of three primary colors is confirmed, the color of the tested object is known.

C. Zigbee :

ZigBee is a low-cost, low-power, wireless mesh network standard targeted at wide development of long battery life devices in wireless control and monitoring applications. Zigbee devices have low latency, which further reduces average current. ZigBee chips are typically integrated with radios and with microcontrollers that have between 60-256 KB flash memory. ZigBee operates in the industrial, scientific and medical (ISM) radio bands: 2.4 GHz in most jurisdictions worldwide; 784 MHz in China, 868 MHz in Europe and 915 MHz in the USA and Australia. Data rates vary from 20 kbit/s (868 MHz band) to 250 kbit/s (2.4 GHz band).

4.3 DESCRIPTION OF HARDWARE USED

ATMega328



Features

• High Performance, Low Power AVR® 8-Bit Microcontroller

• Advanced RISC Architecture
– 131 Powerful Instructions – Most Single Clock Cycle Execution
– 32 x 8 General Purpose Working Registers
– Fully Static Operation
– Up to 20 MIPS Throughput at 20 MHz
– On-chip 2-cycle Multiplier

• High Endurance Non-volatile Memory Segments
– 4/8/16/32K Bytes of In-System Self-Programmable Flash progam memory
(ATmega48PA/88PA/168PA/328P)
– 256/512/512/1K Bytes EEPROM (ATmega48PA/88PA/168PA/328P)
– 512/1K/1K/2K Bytes Internal SRAM (ATmega48PA/88PA/168PA/328P)
– Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
– Data retention: 20 years at 85°C/100 years at 25°C(1)
– Optional Boot Code Section with Independent Lock Bits
In-System Programming by On-chip Boot Program
True Read-While-Write Operation
– Programming Lock for Software Security

• Peripheral Features
– Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
– One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture
Mode
– Real Time Counter with Separate Oscillator
– Six PWM Channels
– 8-channel 10-bit ADC in TQFP and QFN/MLF package
Temperature Measurement
– 6-channel 10-bit ADC in PDIP Package
Temperature Measurement
– Programmable Serial USART
– Master/Slave SPI Serial Interface
– Byte-oriented 2-wire Serial Interface (Philips I2C compatible)
– Programmable Watchdog Timer with Separate On-chip Oscillator
– On-chip Analog Comparator
– Interrupt and Wake-up on Pin Change

• Special Microcontroller Features
– Power-on Reset and Programmable Brown-out Detection
– Internal Calibrated Oscillator
– External and Internal Interrupt Sources
– Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby,
and Extended Standby

• I/O and Packages

– 23 Programmable I/O Lines
– 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF

• Operating Voltage:
– 1.8 - 5.5V for ATmega48PA/88PA/168PA/328P

• Temperature Range:
– -40°C to 85°C

• Speed Grade:
– 0 - 20 MHz @ 1.8 - 5.5V

• Low Power Consumption at 1 MHz, 1.8V, 25°C for ATmega48PA/88PA/168PA/328P:
– Active Mode: 0.2 mA
– Power-down Mode: 0.1 µA
– Power-save Mode: 0.75 µA (Including 32 kHz RTC)

Pin Descriptions

VCC: Digital supply voltage.
GND: Ground.

Port B (PB7:0) XTAL1/XTAL2/TOSC1/TOSC2

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running. Depending on the clock selection fuse settings, PB6 can be used as input to the inverting Oscillator amplifier and input to the internal clock operating circuit. Depending on the clock selection fuse settings, PB7 can be used as output from the inverting Oscillator amplifier.

If the Internal Calibrated RC Oscillator is used as chip clock source, PB7..6 is used as TOSC2..1 input for the Asynchronous Timer/Counter2 if the AS2 bit in ASSR is set.

Port C (PC5:0)

Port C is a 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The PC5..0 output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

PC6/RESET

If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C.

If the RSTDISBL Fuse is un-programmed, PC6 is used as a Reset input. A low level on this pin for longer than the minimum pulse length will generate a Reset, even if the clock is not running. The minimum pulse length is given in Table 28-3 on page 318. Shorter pulses are not guaranteed to generate a Reset.

Port D (PD7:0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

AVCC

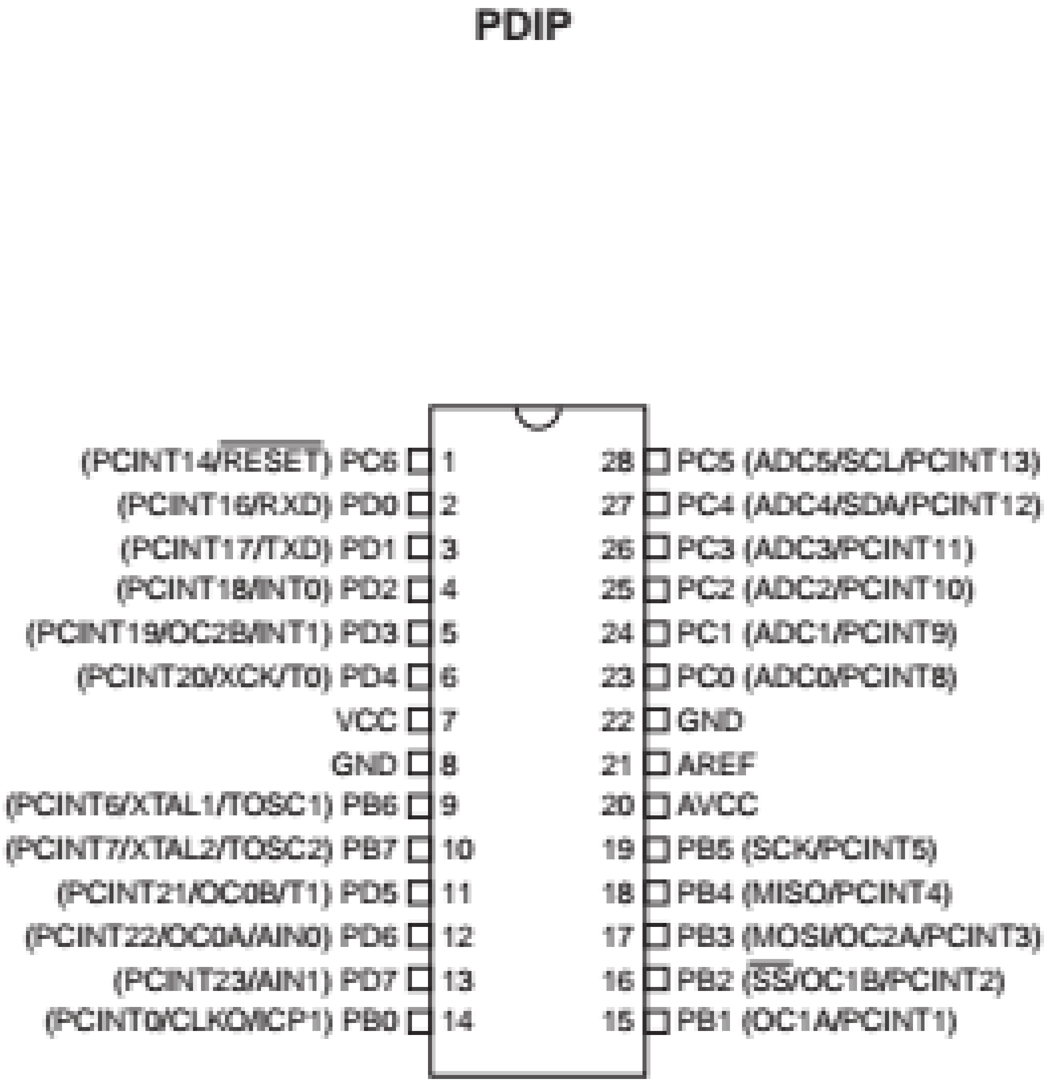
AVCC is the supply voltage pin for the A/D Converter, PC3:0, and ADC7:6. It should be externally connected to VCC, even if the ADC is not used. If the ADC is used, it should be connected to VCC through a low-pass filter. Note that PC6..4 use digital supply voltage, VCC.

AREF

AREF is the analog reference pin for the A/D Converter.

ADC7:6 (TQFP and QFN/MLF Package Only)

In the TQFP and QFN/MLF package, ADC7:6 serve as analog inputs to the A/D converter. These pins are powered from the analog supply and serve as 10-bit ADC channels.



IV. COLOR SENSOR

This color sensor identifies color and gives serial output of RBG value. It can identify 16.7 million color shades giving RGB value for the detected color. The detected color is identified as amount of three primary color values namely Red, Green & Blue with 8 bit accuracy for each primary color. Any color can be separated or combined into three primary colors Red, Green and Blue using the RBG values.

Features

· Individual RGB color detected
· Simple 5V operation
· Serial data output for complete
RGB values
· UART interface for direct connection to any MCU or USB-TTL convertor

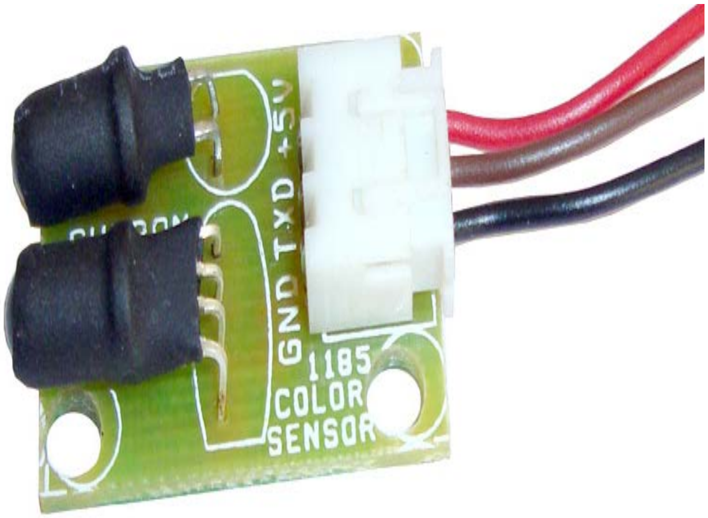
Applications

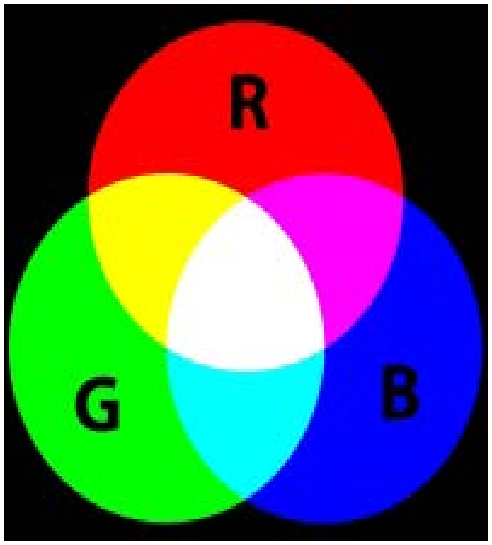
· Color Detection & Sorting operations
· Process control to printed materials
· Ambience light detection / Robotics color detection

Specification:
Parameter Value Unit Notes

Operating Voltage 5 V DC Provide regulated 5V supply
Current 20 mA
Color Detecting
Capacity
16.7 millions RGB R=8 bit (2^8=256 levels)
G=8 bit (2^8=256 levels)
B=8 bit (2^8=256 levels)
256x256x256=16.7 millions
shades detection

Color measuring range 350-750 Nm
Luminance range 100 Lux
Response time 500 ms
Output Data baud 9600 Bps 5V level output
UART Properties (8-N-1)
Start bit:1 bit
Data bit: 8 bits
Parity: None
Stop bit: 1 bit





The sensor switches each primary color RGB, one by one and checks what intensity of color is reflected by the surface of detection. This reflected intensity is converted to 8 bit value. For example a RED surface will strongly reflect RED. While a Yellow surface will reflect RED and GREEN both. According to the induction principle of the three primary colors which create various other colors in nature, once the value of three primary colors is confirmed, the color of the tested object is known. Knowing the value of RGB helps people gain the color of the light which is projected onto the sensor since each color correspond to only one value of RGB.

Further details on RGB model is here.

Serial Data Output format

The serial data at 9600 baud rate consist of 25 bytes for each 500ms interval.

When RED shade of color is detected you would get following type of data in terminal
R=130 G=030 B=030 L=010
Here value of RED is 130 while Green and Blue are 30 both

L=10 means the amount of Light reflected by surface, White surface will reflect most and black the least, This L value you can use to detect the darkness of surface. We recently added this L parameter since it was difficult to detect white and black surface from only RGB values. The sample code of microcontroller and VB software does not implement L value processing but it works with only RGB values. L value can be used to detect white/black surface.

ZIGBEE

Features:

- Build on high performance SiRF Star III chipset
- Average Cold Start time and under 45 seconds
- Low Power Consumption
- 20 channels “All-in-View” tracking
- 200,000 effective correlators for fast TTFF and high sensitivity acquisitions
- Integrated ARM7TDMI CPU with software engineering services and available for embedded customer defined applications.
- On Chip 1Mb SRAM
- Dual TTL level serial ports with one for GPS receiver command message interface, another one for RTCM-104 DGPS input.
- Compact Board Size 1" x 1" x 0.27"(25.4 x 25.4 x 7 mm) for easy integration into hand-held device
- Reacquisition Time 0.1 seconds
- Support Standard NMEA-0183 and SiRF Binary protocol
- Support Accurate 1PPS Output Signal Aligned with GPS Timing
- Multi-Path Mitigation Hardware
- On-Board RTCM SC-104 DGPS and WAAS / EGNOS Demodulator
- Built-in a lithium battery make GPS fast positioning

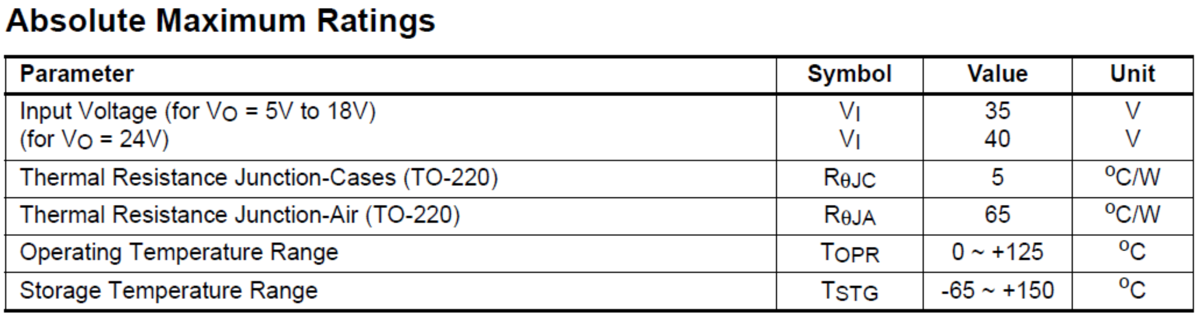
LM7805

Description

The MC78XX/LM78XX/MC78XXA series of three terminal positive regulators are available in the TO-220/D-PAK package and with several fixed output voltages, making them useful in a wide range of applications. Each type employs internal current limiting,thermal shut down and safe operating area protection, making it essentially indestructible. If adequate heat sinkingis provided, they can deliver over 1A output current.Although designed primarily as fixed voltage regulators,these devices can be used with external components to obtain adjustable voltages and currents.

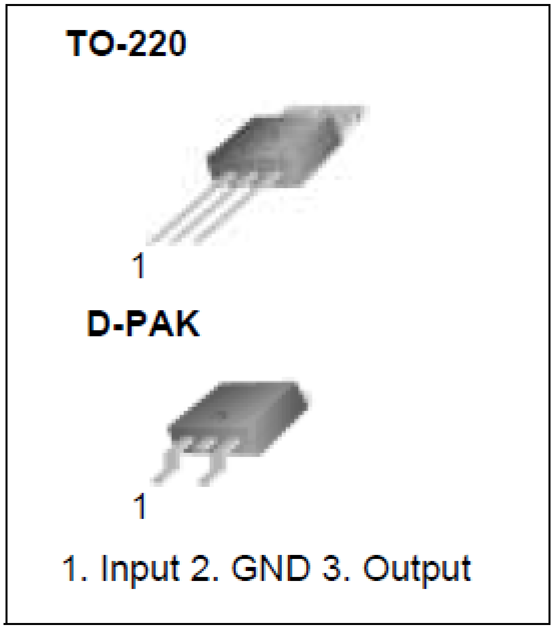
Features

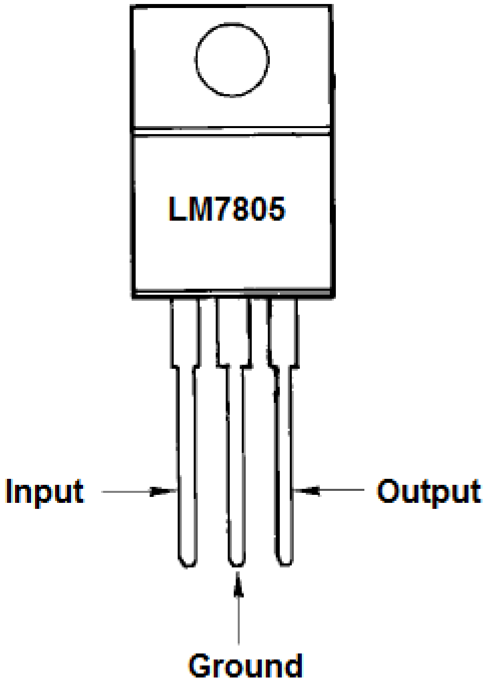
- Output Current up to 1A
- Output Voltages of 5, 6, 8, 9, 10, 12, 15, 18, 24V
- Thermal Overload Protection
- Short Circuit Protection
- Output Transistor Safe Operating Area Protection



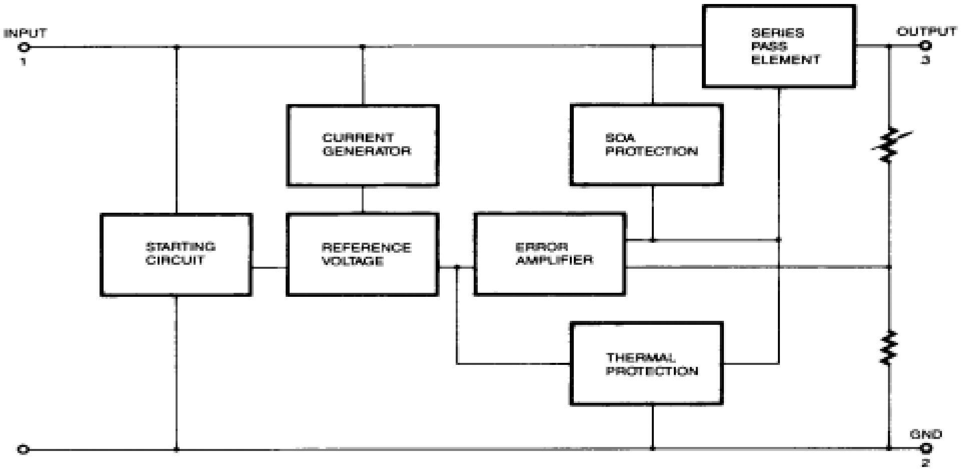
Absolute Maximum Ratings

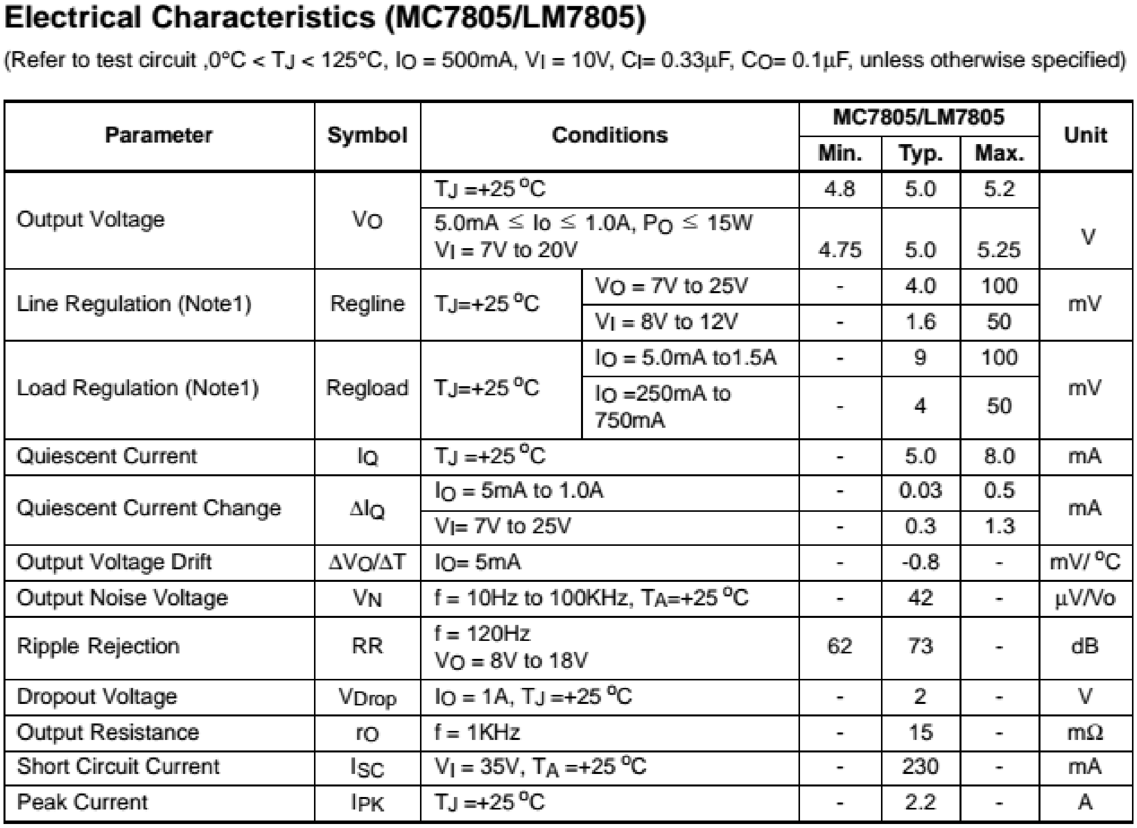
Parameter	Symbol	Value	Unit
Input Voltage (for V_O = 5V to 18V) (for V_O = 24V)	V_I V_I	35 40	V V
Thermal Resistance Junction-Cases (TO-220)	$R_{\theta JC}$	5	°C/W
Thermal Resistance Junction-Air (TO-220)	$R_{\theta JA}$	65	°C/W
Operating Temperature Range	T_{OPR}	0 ~ +125	°C
Storage Temperature Range	T_{STG}	-65 ~ +150	°C





Internal Block Digram





Electrical Characteristics (MC7805/LM7805)

(Refer to test circuit ,0°C < T_J < 125°C, I_O = 500mA, V_I = 10V, C_I= 0.33μF, C_O= 0.1μF, unless otherwise specified)

Parameter	Symbol	Conditions		MC7805/LM7805			Unit
				Min.	Typ.	Max.	
Output Voltage	V_O	T_J =+25 °C		4.8	5.0	5.2	V
		5.0mA ≤ Io ≤ 1.0A, P_O ≤ 15W V_I = 7V to 20V		4.75	5.0	5.25	
Line Regulation (Note1)	Regline	T_J=+25 °C	V_O = 7V to 25V	-	4.0	100	mV
			V_I = 8V to 12V	-	1.6	50	
Load Regulation (Note1)	Regload	T_J=+25 °C	I_O = 5.0mA to1.5A	-	9	100	mV
			I_O =250mA to 750mA	-	4	50	
Quiescent Current	I_Q	T_J =+25 °C		-	5.0	8.0	mA
Quiescent Current Change	ΔI_Q	I_O = 5mA to 1.0A		-	0.03	0.5	mA
		V_I= 7V to 25V		-	0.3	1.3	
Output Voltage Drift	$\Delta V_O/\Delta T$	I_O= 5mA		-	-0.8	-	mV/ °C
Output Noise Voltage	V_N	f = 10Hz to 100KHz, T_A=+25 °C		-	42	-	µV/Vo
Ripple Rejection	RR	f = 120Hz V_O = 8V to 18V		62	73	-	dB
Dropout Voltage	V_{Drop}	I_O = 1A, T_J =+25 °C		-	2	-	V
Output Resistance	r_O	f = 1KHz		-	15	-	mΩ
Short Circuit Current	I_{SC}	V_I = 35V, T_A =+25 °C		-	230	-	mA
Peak Current	I_{PK}	T_J =+25 °C		-	2.2	-	A

PL-2303 USB to Serial Adapter

The PL-2303 USB to Serial adapter is your smart and convenient accessory for connecting RS-232 serial devices to your USB-equipped Windows host computer. It provides a bridge connection with a standard DB 9-pin male serial port connector in one end and a standard Type-A USB plug connector on the other end. You simply attach the serial device onto the serial port of the cable and plug the USB connector into your PC USB port. It allows a simple and easy way of adding serial connections to your PC without having to go thru inserting a serial card and traditional port configuration. This USB to Serial adapter is ideal for connecting modems, cellular phones, PDAs, digital cameras, card readers and other serial devices to your computer. It provides serial connections up to 1Mbps of data transfer rate. And since USB does not require any IRQ resource, more devices can be attached to the system without the previous hassles of device and resource conflicts.

Finally, the PL-2303 USB to Serial adapter is a fully USB Specification compliant device and therefore supports advanced power management such as suspend and resume operations as well as remote wakeup. The PL-2303 USB Serial cable adapter is designed to work on all Windows operating systems.

Features & Specifications

Smart USB to RS-232 (DB 9-pin male serial port) interface
Supports various serial devices like modems, PDAs, cellular phones, digital cameras, card readers, and more.
Easy Plug and Play Installation

Full Compliance with the Universal Serial Bus Specification v1.1
Supports the standard RS-232 Serial Interface
Supports automatic handshake mode
Over 1Mbps data transfer rate

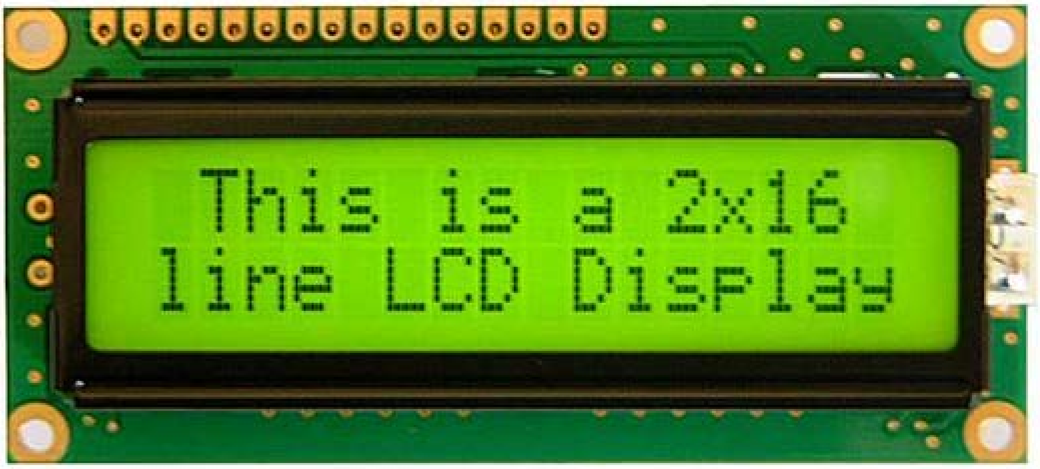
Supports Remote Wake-up and Intelligent Power Management
Provides Dual Buffers for upstream and downstream data transfer
No IRQ resource required
Bus Powered – no separate power supply or battery required

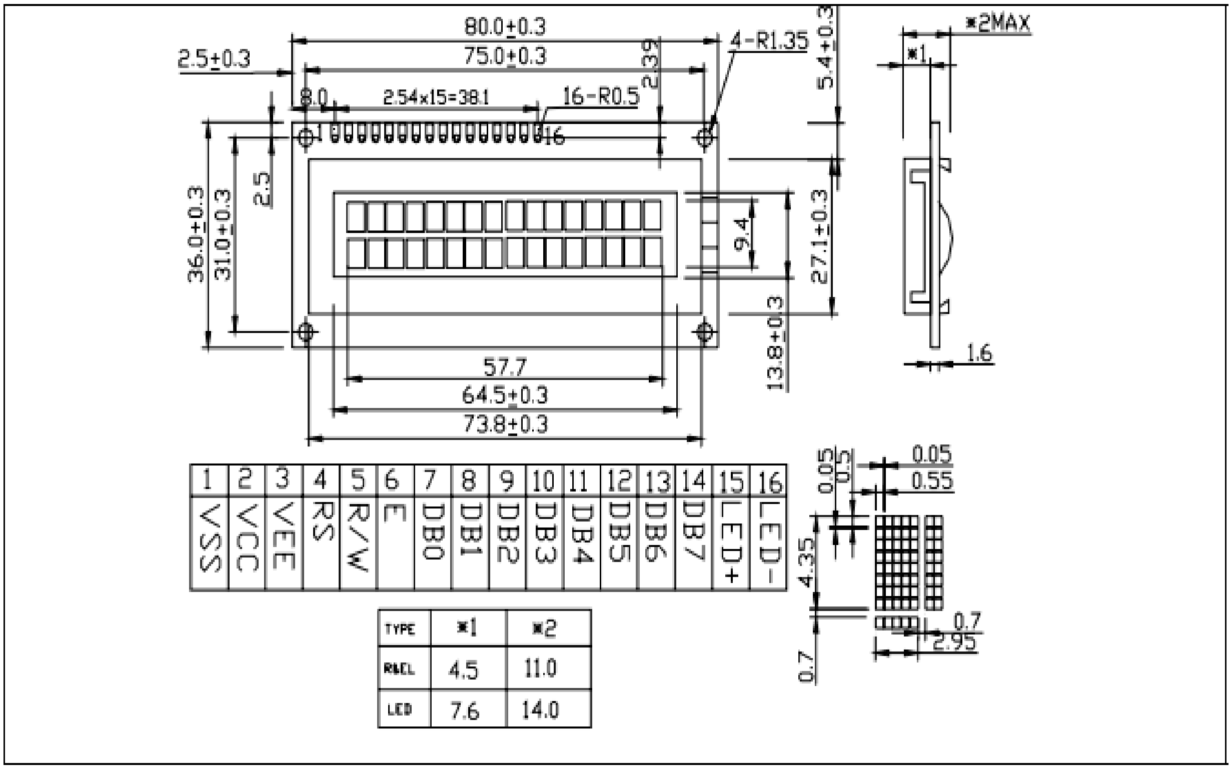
Dimensions L x W x H: _________ (mm)
Weight: ____grams (approximate)

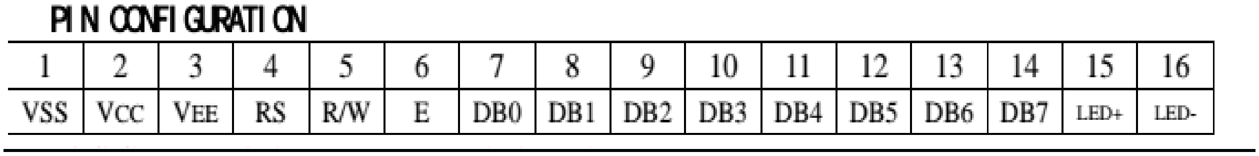
System Requirements

Windows 98, Windows ME, Windows 2000, or Windows XP operating system
AMD or Intel Pentium 133MHz or better based computer USB 1.1 or higher port
USB to RS-232 (Serial Port) Cable Adapter
USB to RS-232 Driver Utility Disk

LCD DISPLAY



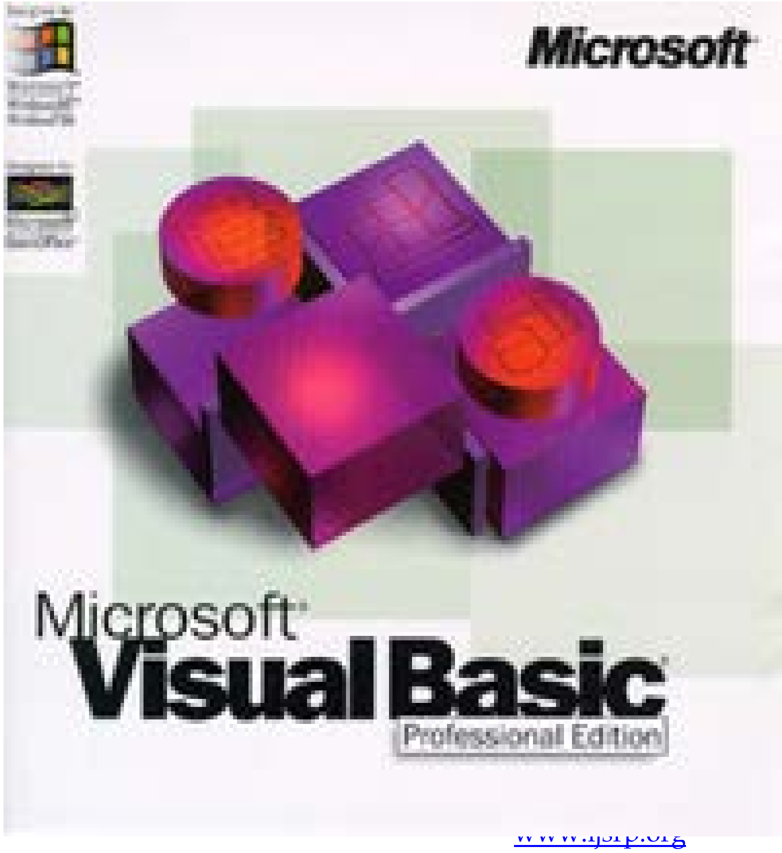




PIN CONFIGURATION

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
VSS	VCC	VEE	RS	R/W	E	DB0	DB1	DB2	DB3	DB4	DB5	DB6	DB7	LED+	LED-

CHARACTERISTICS:
DISPLAY CONTENT 16 CHAR x 2ROW
CHAR. DOTS 5 x 8
DRIVING MODE 1/16D
AVAILABLE TYPES
TN STN(YELLOW GREEN GREY B/W)
REFLECTIVE WITH EL OR LED BACKLIGHT
EL/100VAC 400HZ
LED/4.2VDC



4.4 DESCRIPTION OF SOFTWARE USED

Visual Basic (**VB**) is an event driven programming language and associated development environment from Microsoft for its COM programming model. VB has been replaced by Visual Basic .NET. The older version of VB was derived heavily from BASIC and enables the rapid application development (RAD) of graphical user interface (GUI) applications, access to databases using DAO, RDO, or ADO, and creation of ActiveX controls and objects.

A programmer can put together an application using the components provided with Visual Basic itself. Programs written in Visual Basic can also use the Windows API, but doing so requires external function declarations.

In business programming, Visual Basic has one of the largest user bases. With 62% of developers using some form of Visual Basic, it currently competes with C++ and JavaScript as the third most popular programming language behind C# and Java.

Visual Basic was designed to be easy to learn and use. The language not only allows programmers to easily create simple GUI applications, but also has the flexibility to develop fairly complex applications as well. Programming in VB is a combination of visually arranging components or controls on a form, specifying attributes and actions of those components, and writing additional lines of code for more functionality. Since default attributes and actions are defined for the components, a simple program can be created without the programmer having to write many lines of code. Performance problems were experienced by earlier versions, but with faster computers and native code compilation this has become less of an issue.

Although programs can be compiled into native code executables from version 5 onwards, they still require the presence of runtime libraries of approximately 2 MB in size. This runtime is included by default in Windows 2000 and later, but for earlier versions of Windows it must be distributed together with the executable.

Introduction to Visual Basic:

Welcome to Microsoft Visual Basic, the fastest and easiest way to create applications for Microsoft Windows®. Whether you are an experienced professional or brand new to Windows programming, Visual Basic provides you with a complete set of tools to simplify rapid application development.

So what is Visual Basic? The "Visual" part refers to the method used to create the graphical user interface (GUI). Rather than writing numerous lines of code to describe the appearance and location of interface elements, you simply add prebuilt objects into place on screen. If you've ever used a drawing program such as Paint, you already have most of the skills necessary to create an effective user interface.

The "Basic" part refers to the BASIC (Beginners All-Purpose Symbolic Instruction Code) language, a language used by more programmers than any other language in the history of computing. Visual Basic has evolved from the original BASIC language and now contains several hundred statements, functions, and keywords, many of which relate directly to the Windows GUI. Beginners can create useful applications by learning just a few of the keywords, yet the power of the language allows professionals to accomplish anything that can be accomplished using any other Windows programming language.

The Visual Basic programming language is not unique to Visual Basic. The Visual Basic programming system, Applications Edition included in Microsoft Excel, Microsoft Access, and many other Windows applications uses the same language.

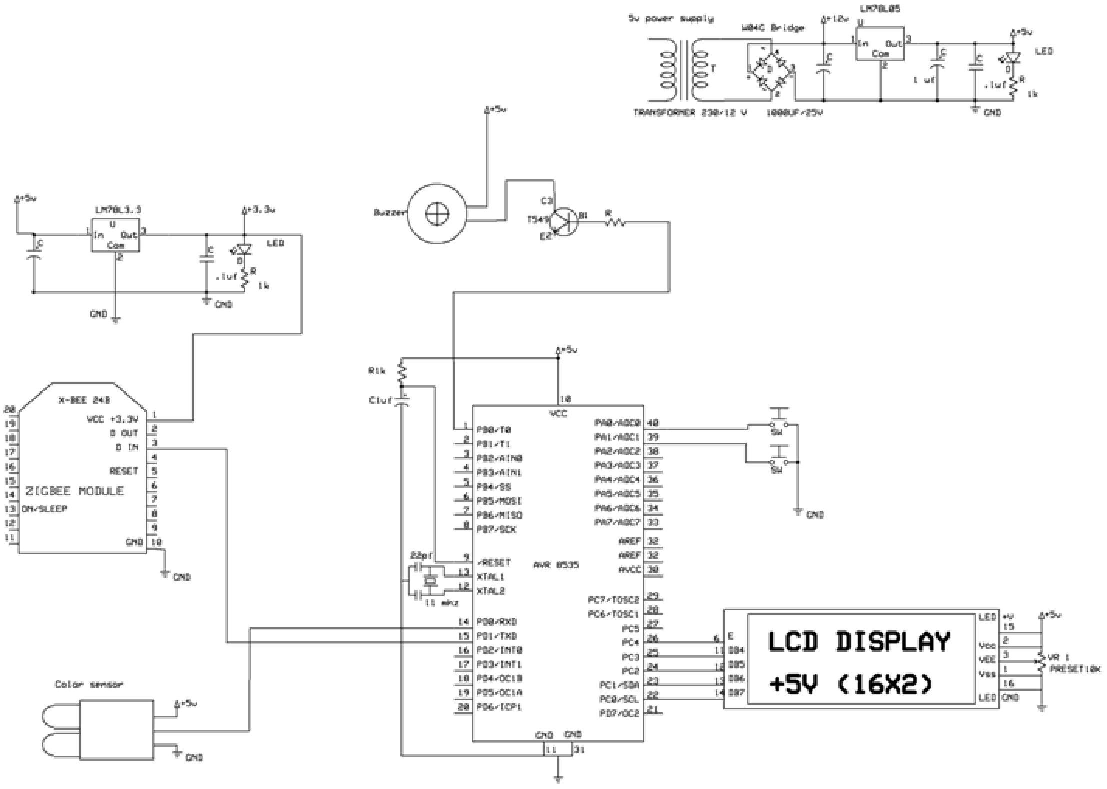
The Visual Basic Scripting Edition (VBScript) is a widely used scripting language and a subset of the Visual Basic language. The investment you make in learning Visual Basic will carry over to these other areas.

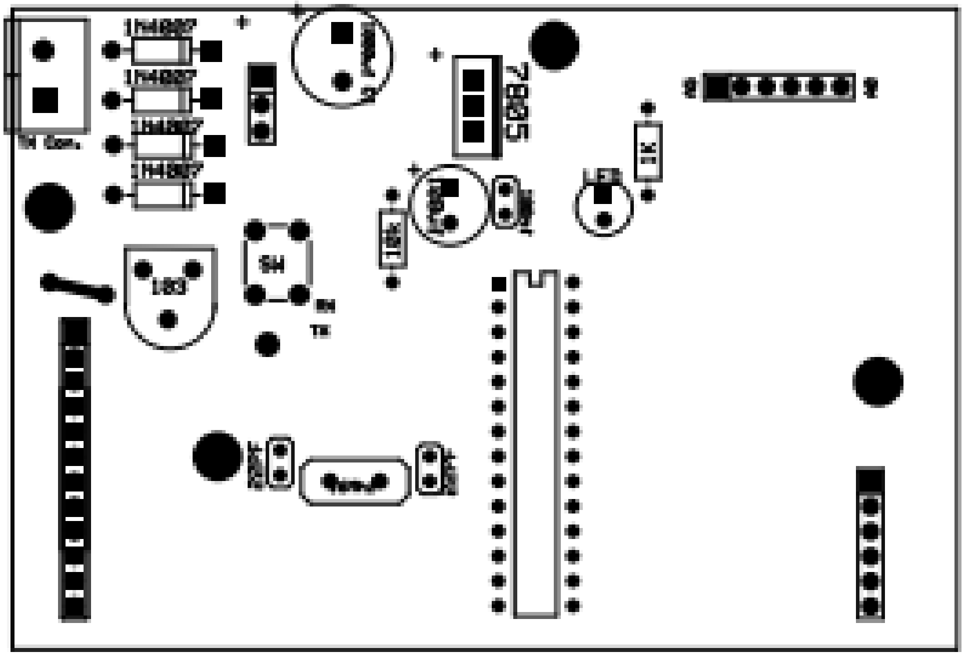
Whether your goal is to create a small utility for yourself or your work group, a large enterprise-wide system, or even distributed applications spanning the globe via the Internet, Visual Basic has the tools you need.

- Data access features allow you to create databases, front-end applications, and scalable server-side components for most popular database formats, including Microsoft SQL Server and other enterprise-level databases.
- ActiveX™ technologies allow you to use the functionality provided by other applications, such as Microsoft Word processor, Microsoft Excel spreadsheet, and other Windows applications. You can even automate applications and objects created using the Professional or Enterprise editions of Visual Basic.
- Internet capabilities make it easy to provide access to documents and applications across the Internet or intranet from within your application, or to create Internet server applications.
- Your finished application is a true .exe file that uses a Visual Basic Virtual Machine that you can freely distribute.

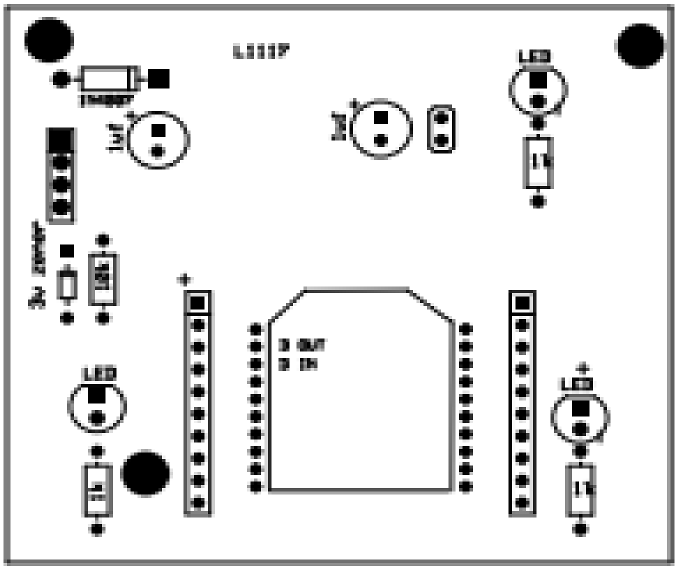
V. SYSTEM IMPLEMENTATION

5.1 CIRCUIT DIAGRAM

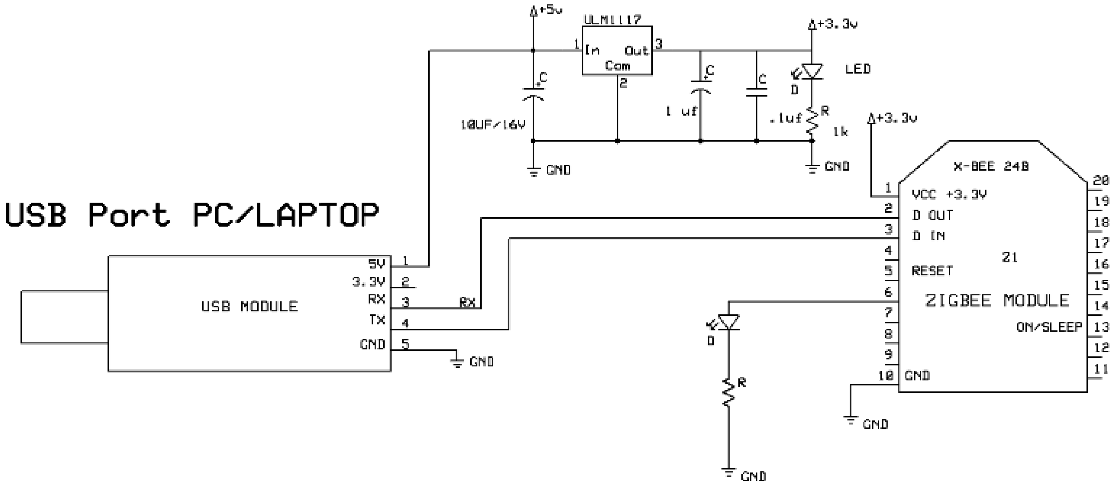




In the above circuit diagram, we are using ATmega16 8-bit microcontroller which has in built adc . Lcd display is connected to port C of the microcontroller ATmega16 which requires 5v power supply.input of color sensor Color sensor and ZigBee device is connected to port D of microcontroller. Buzzer is connected to port B with the help of transistor.



Interfacing of Zigbee module with USB Port of PC



5.2 PCB FABRICATION

Printed circuit boards (PCBs),which are specifically made for each circuit, come in handy. These are very compact and wire-free. Five steps to PCB making

1. Draw the schematic of the circuit on a computer using the required software
2. Design the PCB on the computer using the required software
3. Print the PCB design through a laser printer
4. Take the impression of the circuit on a copper-clad board
5. Remove the excess copper by Etching

There are many software available on the Internet to help you draw the schematic of the circuit. Eagle software isware. So here we will use ExpressSCH for drawing the schematic. ExpressPCB (which includes ExpressSCH) is available on the Internet for free. Circuit description. Let us assemble a simple buffer circuit using the 74HC244 buffer IC. We will put a tap switch at one input and see that we get the same output as the input by putting an LED light at the output pin. So when we push the switch to 'on' position the LED will glow, and when we push it to 'off' position the LED will turn off.

Fig. 1 shows the screenshot of the buffer circuit drawn in ExpressSCH software.

Screenshot description. Icons marked 1 through 24 allow you to per-form various tasks as follows:

1. Rotate the component as well as the text by 90°. It can also be used to rotate only the text
2. Rotate only the component by 90°
3. It is the component and symbol manager. All the components can be found here
4. Options. You can change the color and dimensions of the screen as well as the grid spacing and other options

5. Previous zoom. It resumes the previous screen shape
6. Zoom to fit
7. Zoom out
8. Zoom in
9. Undo
10. Save
11. Open
12. General-purpose pointer, select and moving tool
13. Zoom into selected area
14. Place a component. It will not open the library window
15. Place a signal or symbol level, like the symbols of power supply and ground
16. Place a wire. All the connections are made using this icon
17. Place a corner in the wire, i.e., break a continuous line into two
18. Disconnect a wire from a component and connect it to the other
19. Make a rectangle for a new customised component. You can make
any component which is not present in the library
20. Add a line into a customised component
21. Add a circle or arc in a new component
22. Place a new pin in a component
23. Place text
24. Snap to grid. By keeping this option 'on,' the wires will movemake angles at specific gap. This makes it easy to draw straight lines and designs. By keeping this option 'off,' you will be able to make high-precision angles in the circuit

Designing the PCB

While designing a PCB, try to make it as compact as possible.

Fig. 2 shows the PCB of the circuit (shown in Fig. 1) designed in ExpressPCB. Yellow lines show the silkscreen layer (component layout), red lines show the top copper layer and green portion is the bottom copper layer. Screenshot description. Icons 1 to 31 in Fig. 2 allow you to perform following tasks:
1. Rotate the component as well as the text by 90°
2. Rotate only the component by 90°
3. Move a selected item to the bottom copper layer
4. Move a selected item to the top copper layer
5. Component manager (opens the library window)
6. Options (can change the color codes as well as all the dimensions and
measurements)
7. Zoom to previous state
8. Zoom to fit the screen
9. Zoom out
10. Zoom in
11. Undo
12. Save file
13. Open file
14. General-purpose pointer, select and moving tool
15. Zoom into selected area
16. Place a pad (either through-hole or surface-mount pad). It is helpful in creating multilayer PCBs
17. Place a component. It will not open the library window
18. Place a trace or wire
19. Put a corner in a trace
20. Disconnect a trace
21. Place text in the layout
22. Place a rectangle
23. Place a filled power or ground plane
24. Place a circle or an arc
25. Insert an edge segment in the board perimeter
26. Pad information
27. Highlight net connections. For using this tool, you need to link the schematic of the circuit to this PCB. So go to File→Link Schematic to PCB and choose the respective schematic file. Now while this tool is enabled, if you click a pin of the component it will show the pin to which it is connected. This is very helpful for complex circuits
28. Toggle display of the silkscreen layer
29. Toggle display of the top copper layer
30. Toggle display of the bottom copper layer
31. Toggle snapping to grid

PCB description. The circuit's PCB is described below, with the actual names of the components as in the library given within brackets:
1. At the centre is a 20-pin base (20 dip)
2. The 3-pin IC at the top is 7805 (semiconductor TO-220 without mounting hole, i.e., 7805 vreg)
3. On the left of 7805 is a 1000µF capacitor base (cap-lead spacing 4.5 mm (0.177 inch))
4. On the right of 7805 is a 0.1µF capacitor base (cap-lead spacing 2.5 mm (0.1 inch))
5. The three holes on the left are for the DC female input jack; in it 12V power supply will be connected
6. On the right of the 20-pin base is the base for a 1-kilo-ohm resistor (0.25-watt resistor with lead spacing of 0.35 inch)
7. Just below the 1-kilo-ohm resis-tor base is the LED base (LED-T1)

Rules of PCB making.
1. There should be no 90° connections of wires in the PCB. All connections should preferably be made at 45° (refer Fig. 3).
2. The mirror image of the PCB layout will always be imprinted on the copper side (the solder side of the copper-clad board) of the actual PCB.
So for any text to be written on the copper side, the text should be mirrored during layout design so that the textappears normal on the copper side of the PCB.
3. Always print from a laser printer. Printout from inkjet or any other printer will not work as it may not be sharp enough.
4. You can print on both top and bottom parts of the board, but here we will print text only on the top copper layer.
5. There are standards for using traces of different thicknesses for different purposes:
(a) 0.6mm (0.025-inch) trace for single tracks
(b) 1.3mm (0.050-inch) trace for power and ground tracks
(c) 0.2mm to 0.4mm (0.010- to 0.015-inch) traces for going between IC and component pads
6. Tracking from large to small and then back to large is known as 'necking' or 'necking down' (see Fig. 4).
7. You can also let rest of the spaceremain covered by copper, leaving clearance beside each line, so our PCB will look as

shown in Fig. 5. We will not use it as this type of filled circuit is best for industrial-level machine etching.
8. The standards for clearances for electrical conductors are listed here in
the table (on previous page).
9. For thin tracks (<0.6mm traces), it is good to add ‘chamfer’ to ‘T’ junction, thus eliminating 90° angles (see Fig. 6).

Printing the PCB layout

Go to File→Print. A window as shown in Fig. 7 will open. Select ‘Layers to
Print.’ Keep ‘Enlarge to Fit Page’ option unchecked as it will not give the exact layout of the components. Print the PCB layout from a laser printer. If you don’t have a laser printer at home, convert the file into a PDF file (using PDF995 software) and then print it elsewhere.

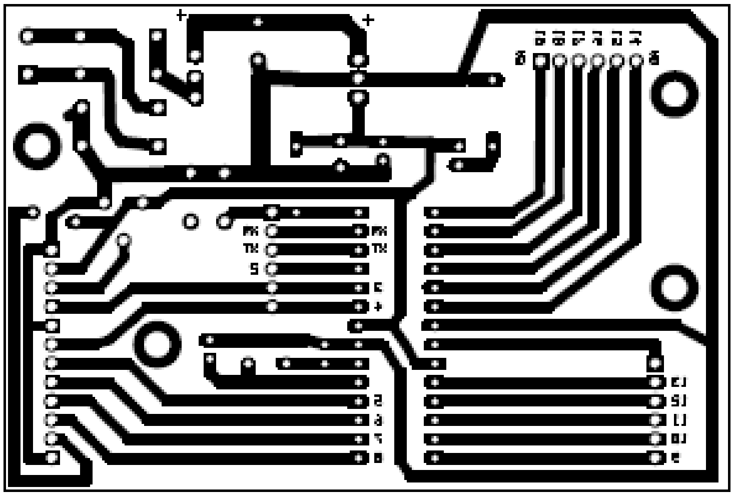
Etching

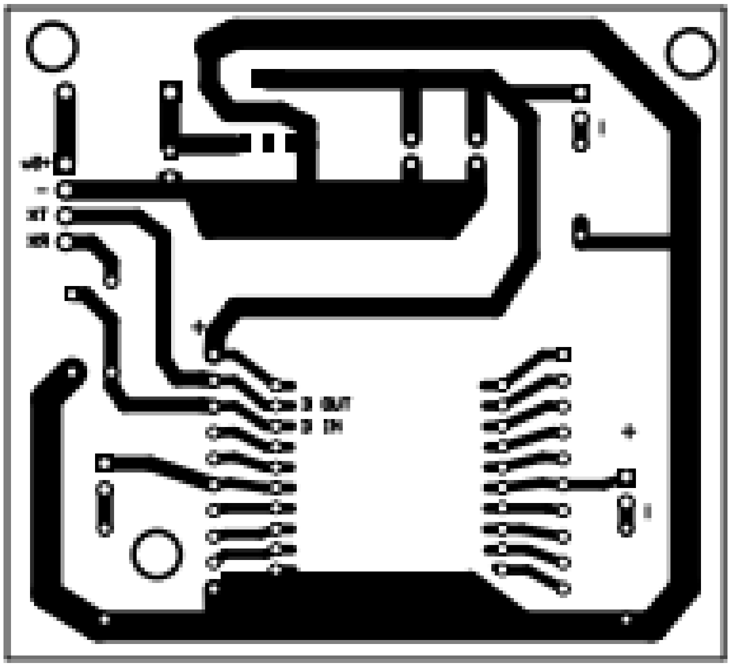
Cut the copper-clad board to a size matching the size of the PCB design printout. You can also use a glass epoxy board but it’s costlier than a copper-clad board. Fig. 9 shows a full-size copper-clad board. Put the paper printout on the board with the printed side facing the copper side. Affix the paper to the board using cellotape so that the paper does not move while ironing (Fig. 10). Now take your household electric iron and set its temperature to the maximum (Fig. 11). Press the hot iron on the paper and carefully move it across the paper for about four minutes.

While doing so, check for impressions. Continue ironing until the complete impression of the circuit comes on the copper side (Fig. 12).

Complete the blanks, if any, with a good permanent marker. If any line is not dark enough, redraw it on the board using the permanent marker. Wash the board in normal tap water. Drill IC holes using a 1mm hand PCB drill as shown in Fig. 14. Redraw the lines using the permanent marker if they have been defaced by the drill. (ferric chloride) powder in hot water. The reaction Now mix some FeCl3 is vigorous, so take safety precaution. Put the copper-clad board in the solution and constantly tilt the container from side to side without spilling its contents (Fig. 15). This is done to speed up the reaction. It takes five to six minutes to wash away all the excess copper. In between, keep checking the board. Ensure that the marker or carbon of the impression does not wash away. Take the board out (Fig. 16) and wash it under tap water to remove the permanent marker ink. Remove carbon by using nail polish remover. Use a scrubber to gently scrub the copper surface (Fig. 17) till it shines.

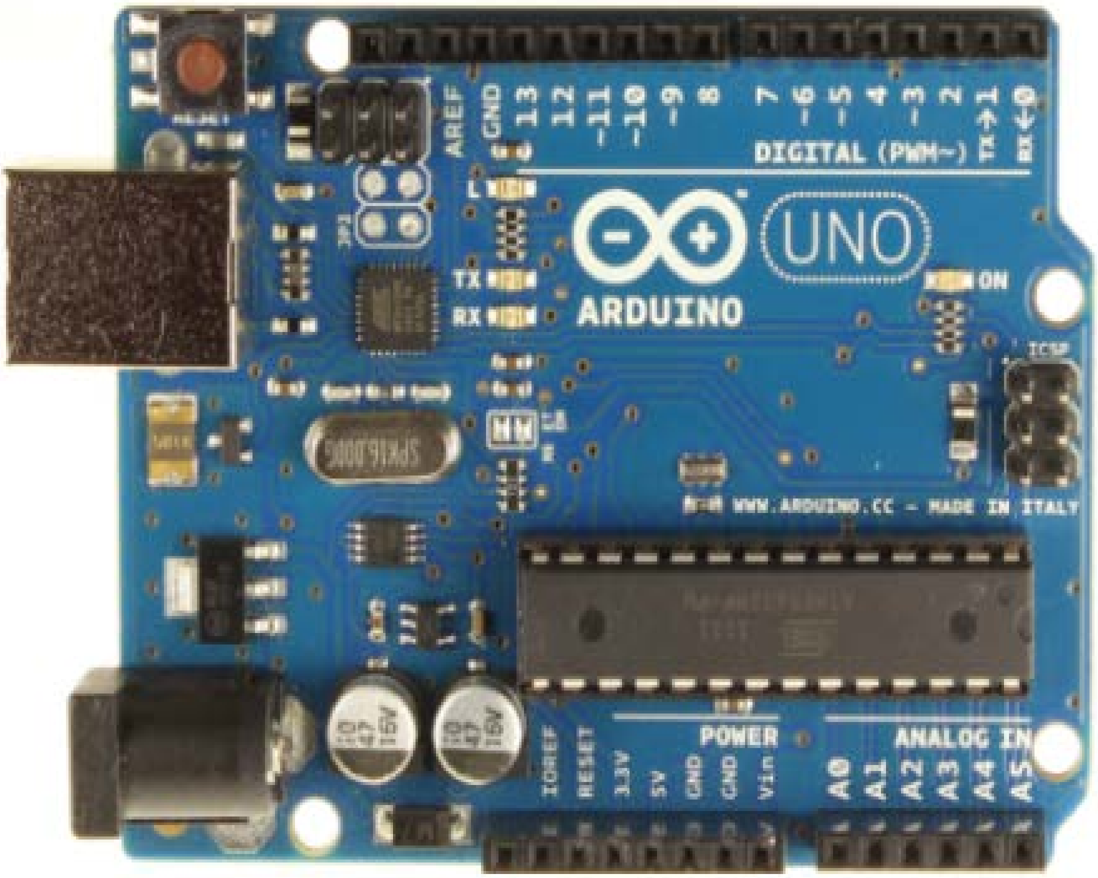
5.3 PCB LAYOUT





5.4 SOFTWARE IMPLEMENTATION

Arduino Uno R3 (Atmega328)



The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

Summary
Microcontroller ATmega328
Operating Voltage 5V
Input Voltage (recommended) 7-12V
Input Voltage (limits) 6-20V
Digital I/O Pins 14 (of which 6 provide PWM output)
Analog Input Pins 6
DC Current per I/O Pin 40 mA
DC Current for 3.3V Pin 50 mA
Flash Memory 32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM 2 KB (ATmega328)
EEPROM 1 KB (ATmega328)
Clock Speed 16 MHz

Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a.inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A SoftwareSerial library allows for serial communication on any of the Uno's digital pins.The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details. For SPI communication, use the SPI library.

Programming

The Arduino Uno can be programmed with the Arduino software (download). Select "Arduino Uno from the Tools > Board menu (according to the microcontroller on your board). For details, see theThe ATmega328 on the Arduino Uno comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available . The ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by:

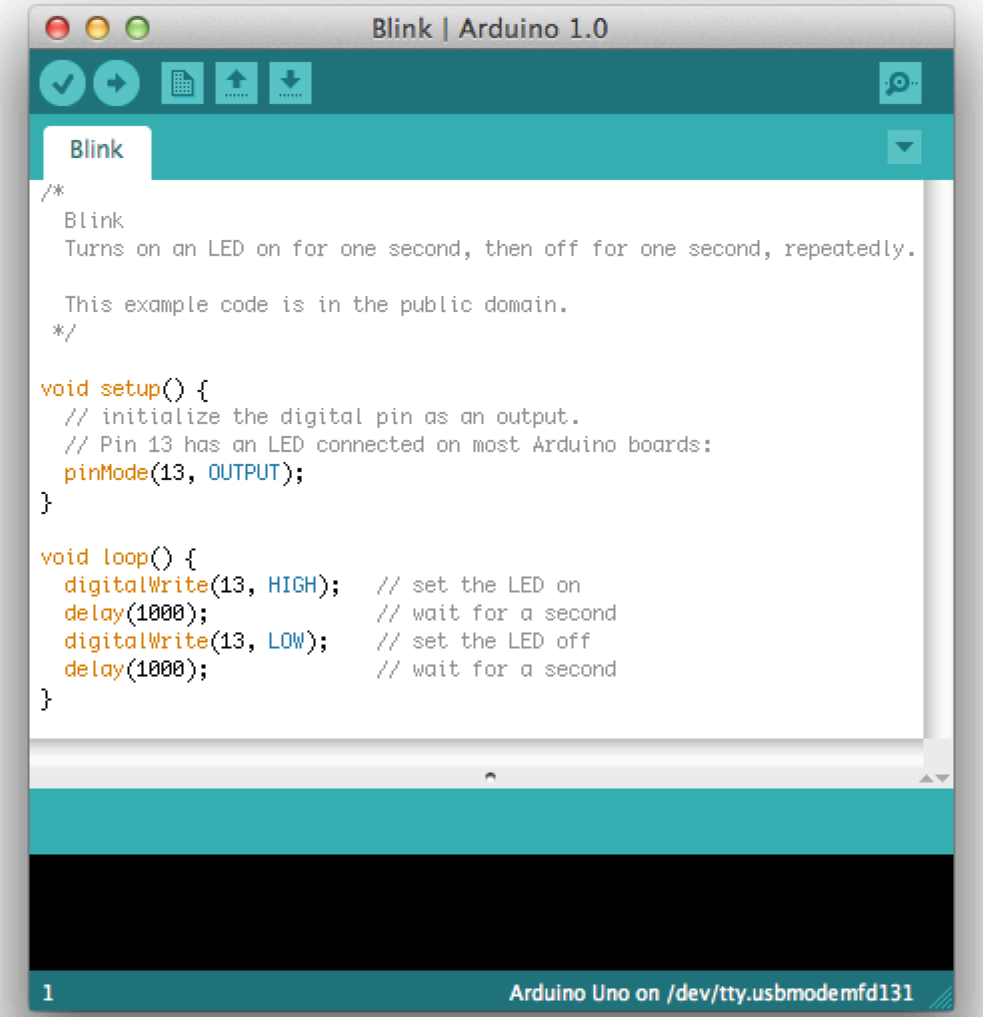
□ On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.

□ On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See this user-contributed tutorial for more information.

Automatic (Software) Reset

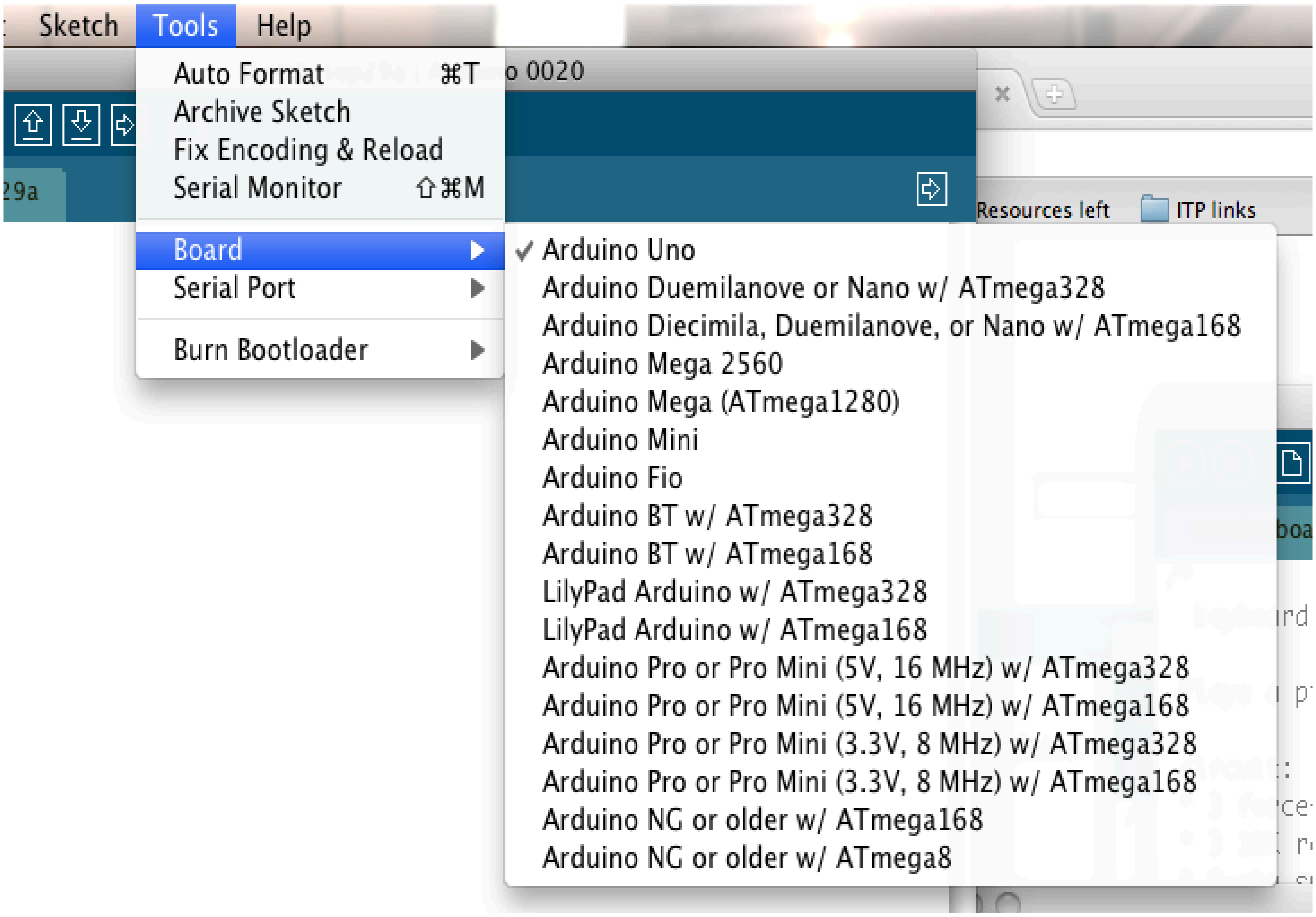
Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of theATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread for details.



```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);   // set the LED on
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // set the LED off
  delay(1000);              // wait for a second
}
```

Select your board

You'll need to select the entry in the Tools > Board menu that corresponds to your Arduino.


Sketch
Tools
Help
Auto Format ⌘T
Archive Sketch
Fix Encoding & Reload
Serial Monitor ⇧⌘M
Board
Serial Port
Burn Bootloader
o 0020
Resources left
ITP links
✓ Arduino Uno
Arduino Duemilanove or Nano w/ ATmega328
Arduino Diecimila, Duemilanove, or Nano w/ ATmega168
Arduino Mega 2560
Arduino Mega (ATmega1280)
Arduino Mini
Arduino Fio
Arduino BT w/ ATmega328
Arduino BT w/ ATmega168
LilyPad Arduino w/ ATmega328
LilyPad Arduino w/ ATmega168
Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega328
Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega168
Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega328
Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega168
Arduino NG or older w/ ATmega168
Arduino NG or older w/ ATmega8

Selecting an Arduino Uno

For Duemilanove Arduino boards with an ATmega328 (check the text on the chip on the board), select Arduino Duemilanove or Nano w/ ATmega328. Previously, Arduino boards came with an ATmega168; for those, select Arduino Diecimila, Duemilanove, or Nano w/ ATmega168. (Details of the board menu entries are available on the environment page.)

Select your serial port

Select the serial device of the Arduino board from the Tools | Serial Port menu. This is likely to beCOM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu; the entry that disappears should be the Arduino board. Reconnect the board and select that serial port.

Upload the program

Now, simply click the "Upload" button in the environment. Wait a few seconds - you should see the RX and TX leds on the board flashing. If the upload is successful, the message "Done uploading." will appear in the status bar. (*Note:* If you have an Arduino Mini, NG, or other board, you'll need to physically present the reset button on the board immediately before pressing the upload button.)

PROGRAM

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(19,4,5,6,7,8);

void setup()  {

 lcd.begin(16, 2);
 lcd.print("  Plant Health  ");
 lcd.setCursor(0,1);
 lcd.print("    Monitor   ");
 delay(1000);
 lcd.clear();

 Serial.begin(9600);

}

void loop() {

  while (Serial.available() > 0) {

   int red = Serial.parseInt();       // look for the next valid integer
in the incoming serial stream:
   int green = Serial.parseInt();
   int blue = Serial.parseInt();
   int white = Serial.parseInt();

   if (Serial.read() == '\n') {       // look for the next valid integer
in the incoming serial stream:
   red = 255 - constrain(red, 0, 255); // constrain the values to 0
- 255 and invert
   green = 255 - constrain(green, 0, 255);
   blue = 255 - constrain(blue, 0, 255);

   lcd.setCursor(0,0);
   lcd.print("R:");
   lcd.print(red,DEC);
   lcd.print("   ");

   lcd.setCursor(9,0);
   lcd.print("G:");
   lcd.print(green,DEC);
   lcd.print("   ");

   lcd.setCursor(0,1);
   lcd.print("B:");
   lcd.print(blue,DEC);
   lcd.print("   ");

   lcd.setCursor(9,1);
   lcd.print("W:");
   lcd.print(white,DEC);
   lcd.print("   ");

   Serial.print("R:" );
   Serial.print(red, DEC);

   Serial.print(" G:" );
   Serial.print(green, DEC);

   Serial.print(" B:" );
   Serial.print(blue, DEC);

   Serial.print(" W:" );
   Serial.println(white, DEC);

   }
  }
}
```

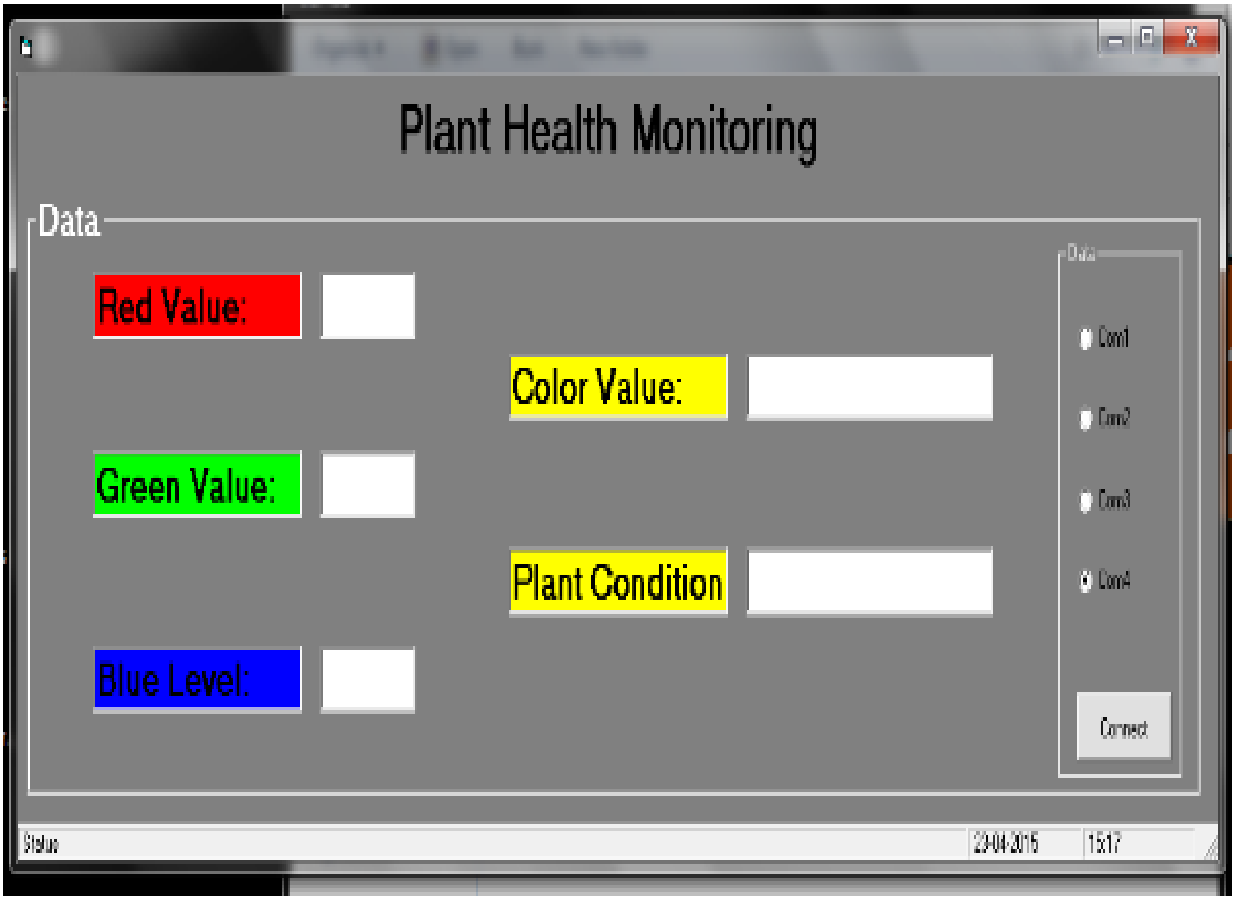
VB PROGRAM

```
Option Explicit

Dim id As Integer

Dim RCount As Integer

Dim CHNo As Variant

Dim Data As Variant

Dim c As Byte

Dim RChr1 As String

Dim RChr As String

Dim data1 As Variant

Dim strBuffer As Variant

Dim con As New ADODB.Connection

Dim rst As New ADODB.Recordset

Dim RowValue As Integer

Dim rs As New ADODB.Recordset

Private Sub Command1_Click()

  If Port.PortOpen = True Then

    Port.PortOpen = False

  End If



  If optCom1.Value Then

    Port.CommPort = 1

  End If

  If optCom2.Value Then

    Port.CommPort = 2

  End If

  If optCom3.Value Then

    Port.CommPort = 3

  End If

  If optCom4.Value Then

    Port.CommPort = 4

  End If



  Port.Settings = "9600,N,8,1"

  Port.InputLen = 1

  Port.InBufferSize = 2000

  Port.OutBufferSize = 2000

  Port.RThreshold = 1
```

```
    Port.PortOpen = True

    Timer1.Enabled = True


End Sub

Private Sub Form_Load()

Text1.Text = "o"

Text2.Text = "o"

Text3.Text = "o"

Text4.Text = "o"

Text5.Text = "o"

End Sub

Private Sub Form_Unload(Cancel As Integer)

    End

End Sub

Private Sub Port_OnComm()

    If Not Port.CommEvent = comEvReceive Then Exit Sub


    RChr = Port.Input


    RCount = RCount + 1


If RChr = "M" Then

RCount = 0


If CHNo = 1 Then

Text1.Text = Data

End If

If CHNo = 2 Then

Text2.Text = Data

End If

If CHNo = 3 Then

Text3.Text = Data

End If

If CHNo = 4 Then

If Val(Data) = 1 Then

Text4.Text = "LightGreen"

Text4.BackColor = &HC0FFC0

Text5.Text = " Young Leaf"

End If

If Val(Data) = 2 Then

Text4.Text = "Dark Green"

Text4.BackColor = &H8000&

Text5.Text = "Plant Healthy"

End If

If Val(Data) = 3 Then

Text4.Text = "Pale Green"

Text4.BackColor = &H8080&

Text5.Text = "Mal Nutrition"

End If

If Val(Data) = 4 Then

Text4.Text = "Yellow"

Text4.BackColor = &H80FF&

Text5.Text = "Getting Old"

End If

If Val(Data) = 5 Then

Text4.Text = "Red Spots"

Text4.BackColor = &HFF&

Text5.Text = "Pest Problem"

End If
```

```
If Val(Data) = 6 Then
Text4.Text = "Brown"
Text4.BackColor = &H80&
Text5.Text = " Plant Dying"
End If
End If
Data = ""
Exit Sub
End If

  If RCount = 1 Then
    CHNo = RChr
    Exit Sub
  End If

  If RCount >= 2 Then
    Data = Data & RChr
  End If

End Sub
Private Sub Timer1_Timer()
Call log
End Sub
Private Sub log()
con.Open "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" &
App.Path & "\db1.mdb;Persist
Security Info=False"
rs.ActiveConnection = con
rs.CursorLocation = adUseClient
rs.CursorType = adOpenDynamic
rs.LockType = adLockOptimistic
rs.Open "select * from log"
If rs.RecordCount <= 10000 Then
rs.AddNew
With rs
!ch1 = Val(Text1.Text)
!ch2 = Val(Text2.Text)
!ch3 = Val(Text3.Text)
!Date = Date
!Time = Time
.Update
.Close
End With
Else
rs.MoveFirst
rs.Delete
rs.Close
End If
con.Close
End Sub
```

VB Output Screen



VI. CONCLUSION

In this project, we discussed the wireless solution of greenhouse monitoring and control system based on ZigBee technology, and designed the wireless nodes, network establishment and software system. With the capabilities of self-organizing, self-configuring, self-diagnosing and self-healing, the ZigBee based monitoring and control system provides nearly unlimited installation flexibility for sensors, increases network robustness, and considerably reduces costs.monitoring systems can reliably and economically augment manual, route-based condition monitoring programs. Critical information from remote and difficult-to-access equipment can be continuously acquired, transmitted, and integrated into systems that help optimize the activities of maintenance , ultimately improving health of plants by adding required fertilizers and taking necessary precautions and eliminating unplanned failures.

VII. FUTURE SCOPE

The future scope of our project involves completing the entire robotic system with fully integrated sensors. This will allow investigations to be carried out to optimize plant growth based on the information obtained from the color sensors, by relating color information to plant quality. By using integrated sensors we can use it for sensing the environmental conditions such as light ,humidity ,soil moisture which cannot be measured by the human eye. We can also implement this project on PDA'S and mobile devices. By implementing more advance program we can make this project which will be able to monitor plants of different colors and species by using web technology we can make this system portable which can be accessed from any part of the world. Through wireless technology data acquisition from remote places and database storage is the supporting structure of the system which can be used for further research studies like soil content analysis using different sensors and also get information of varying conditions .

REFERENCES

[1] Tanaka A.,RitsumeikanUniversity,Japan, “Wireless self-powered plant health monitoring sensor system.

[2] AtmelCorporation,Datasheet Rev.2466T-07/10 onwards [3]"ZigBee Specification FAQ". Zigbee Alliance.Retrieved 14 June 2013.

[3] ZigBee Document 053474r06, Version 1.0, ZigBee.

[4] "ZigBee Wireless Networking", Drew Gislason (via EETimes)

[5] “A computer-vision based power plant monitoring system’’ by Chan, From City Polytech. of Hong Kong,IEEE publish date 08 Nov 1991

AUTHORS

First Author – Tejas G. Mahale, Department of Electronics and Telecommunication, Padmabhushan Vasantdada Patil Pratishthan’s College of Engineering, Vasantdada Patil Educational Complex, Eastern Express Highway, Near Everard Nagar, Sion, Chunabhatti, Mumbai – 400022

Second Author – Sanath S. Shetty, Department of Electronics and Telecommunication, Padmabhushan Vasantdada Patil Pratishthan’s College of Engineering, Vasantdada Patil Educational Complex, Eastern Express Highway, Near Everard Nagar, Sion, Chunabhatti, Mumbai – 400022

Third Author – Salman B. Bandri, Department of Electronics and Telecommunication, Padmabhushan Vasantdada Patil Pratishthan’s College of Engineering, Vasantdada Patil Educational Complex, Eastern Express Highway, Near Everard Nagar, Sion, Chunabhatti, Mumbai – 400022

Fourth Author – Gyanchandra Gupta, , Department of Electronics and Telecommunication, Padmabhushan Vasantdada Patil Pratishthan’s College of Engineering, Vasantdada Patil Educational Complex, Eastern Express Highway, Near Everard Nagar, Sion, Chunabhatti, Mumbai – 400022

Fifth Author – Prof. Gitanjali Korgaonkar, , Department of Electronics and Telecommunication, Padmabhushan Vasantdada Patil Pratishthan’s College of Engineering, Vasantdada Patil Educational Complex, Eastern Express Highway, Near Everard Nagar, Sion, Chunabhatti, Mumbai – 400022