Dual Temperature Monitoring & Control using IoT

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

by

RAGHUWAR JEE JHA (11701615033)

Under the supervision of

Mr. Budhaditya Biswas
Assistant Professor
Department of Electrical Engineering

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)


SAVE WATER

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 **Dual temperature monitoring & control using IoT** is the bonafide work carried out by **RAGHUWAR JEE JHA (11701615033),** a student 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 2018-19, 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

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

(External Examiner)

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.

RAGHUWAR JEE JHA (11701615033)

To

The Head of the Department
Department of Electrical Engineering
RCC Institute of Information Technology
Canal South Rd. Beliagahata, 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, I present the following thesis entitled "**Dual Temperature Monitoring & Control using IoT**". This work was performed under the valuable guidance of Mr. Budhadtiya Biswas, Assistant Professor in the Dept. of Electrical Engineering.

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

Yours Sincerely,

RAGHUWAR JEE JHA (11701615033)

Contents

List of Figures

List of Tables

ABBREVIATIONS AND ACRONYMS

IOT – Internet of Things
***FCC* -** Federal Communications Commission
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
GPIO – General Purpose Input Output
***API* –** Application Program Interface

ABSTRACT

In most of the industrial and domestic applications temperature monitoring is important and many problems can occur due to lack of proper temperature monitoring system. In this project we will monitor and regulate temperature of our room by the help DS18B20 temperature sensor along with node mcu. Monitoring temperature is important for many operations like in any industries where heaters or cooler are used, heat or cool up to a certain temperature is required. In this project we are using IoT based temperature monitoring system that can be access data anywhere and anytime through the internet.

The model will continuously monitor the temperature condition of the room and the data can be monitored and regulate at anytime and anywhere from the internet. Whenever the temperature of room crosses the predefined set limits, cooler or heater will on according to that. IoT web based temperature monitoring is one type of temperature reorder that monitors a temperature in a room and stores the data into a database and display the current temperature on LCD and also in Blynk Mobile app. This model can be used in various processes like in automotive industries, air conditioning, power plant and other industries that need the data to be saved and analyzed.

CHAPTER 1

(Introduction)

1.1 INTRODUCTION

The project proposes an efficient implementation for IoT (Internet of Things) used for monitoring and controlling the room temperature via World Wide Web. Home automation system uses the portable devices as a user interface. They can communicate with home automation network through an Internet gateway, by means of low power communication protocols like Zigbee, Wi-Fi etc. This project aims at controlling room temperature via android app using Wi-Fi as communication protocol and node MCU as server system.

NodeMCU is an open source IoT platform. The user here will move directly with the system through a android based interface over the web, whereas room temperature is remotely controlled through easy user interface. The server will be interfaced with relay hardware circuits that control the device for controlling the temperature.. The server communicates with the microcontroller and if the temperature sense by the DS18B20 precious temperature sensor exceed the limits then the controlling device will be switch ON through the relay..

1.2 IC Temperature sensor

An IC Temperature Sensor is a two terminal integrated circuit temperature transducer that produces an output current proportional to absolute temperature. The sensor package is small with a low thermal mass and a fast response time. The most common temperature range is -55 to 150°C (-58 to 302°F). The solid state sensor output can be analog or digital.

Voltage Output IC Sensors

- Typically 10mV per degree C with nominal output correlated to 0K, 25°C.
- Some sensors have an offset at 0°C so that they can be used and read below 0°C without having to use a negative power supply.
- Non-linearity typically less than 1°C across their temperature range.

Current Output IC Sensors

- Nominal Output: 298 µA at 25°C
- 1 µA output per °C

Digital Output IC Sensors

- Have built in A-D Converters
- The number of digits in the A-D converter provides the resolution
- 10 Bit plus sign provides temperature resolved in increments of 0.25°C
- 12 Bit plus sign provides temperature resolved in increments of 0.0625°C

Where are IC Sensors used?

- On circuit boards to monitor and control temperature.
- In computers to control CPU temperature.
- In telecommunications applications (cell phones & PDA™).
- In some industrial immersion applications.

Strengths and Weaknesses of IC Sensors

Strengths:

- Analog or Digital outputs available
- Low cost
- Direct voltage, current or digital output needing no additional circuitry
- Linear output, no curve fitting
- Direct reading of temperature (1.000 = 100C and 298ÂµA = 298K or 25°C) on some analog devices
- Various communication interfaces

Weaknesses:

- Narrow temperature range: -55 to 150°C Max
- Wider interchangeability than most RTDs and thermistors
- Wide variation in accuracy between different models
- Small package sizes can be a barrier to low cost applications in some immersion designs

1.3 Comparison of IC temperature sensors with other types of temperature seonsors

The analog IC solid state sensors provide an output as a voltage or current that is proportional with temperature without additional circuitry. The digital IC sensors provide an output that has been processed thru an integral A-D converter and is ready for input into digital control and monitoring systems. The IC sensors do not require linearization or other circuitry. The cost of IC sensors are also very competitive with, in some cases less costly than, RTD and thermistor sensors

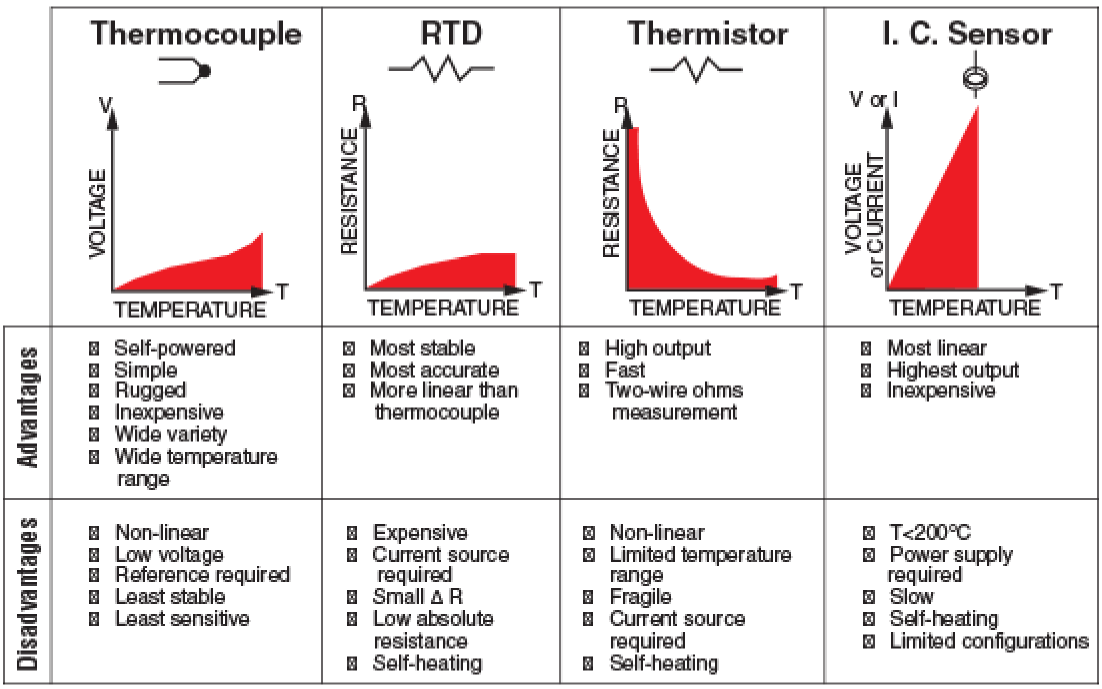


Figure 1: comparison of different temperature sensors

1.4 Overview and benefits of the project

Remote Control Technology's line of dependable, durable wireless remote switching systems can and will make money for you and your business. Wireless remote control benefits include:

No legal issues
Obtaining access to or traversing properties with hard lines is extremely difficult.

No copper wire to steal
As the price of copper increases, so does the possibility that your wire will be stolen. Using a wireless remote system means no wire for thieves to steal.

Extended range
Unlike much of the equipment on the market, Remote Control Technology's wireless remote equipment has long-range communication capabilities. It can be controlled anywhere from the globe.

Eliminate the need for wire and conduit
Wire and conduit are expensive and high maintenance. Typical wear-and-tear, digging, rodent damage, theft, etc., are all examples of problems that can damage wire. RCT's wireless remote systems put an end to these drawbacks of wired technology.

Higher profits
Wireless remote switching systems eliminate the costly, labor-intensive process of trenching and laying wire. As a result, the contractor can enjoy an increased profitability of 200 percent or more in this facet of the job.

No FCC licensing required
RCT equipment does not require FCC licensing, whereas much of the other equipment on the market does. This is a significant benefit, as the FCC licensing process alone may take up to 8 weeks.

Less maintenance and servicing
In many states a contractor is obligated by law to maintain pumping systems for up to a year after its installation. RCT switching systems eliminate a majority of these maintenance and servicing issues by automating the job. Fewer service calls mean higher profits.

Reliability and compatibility
All of the components that a contractor puts into a project must interface with one another and have the utmost reliability. RCT wireless remote equipment has proven to be highly compatible with standard equipment used in most industries, as well as offering unparalleled reliability in use with programmable logic controllers (PLCs), various switches and relays, etc.

1.5 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 operation of DS18B20 temperature sensor and how to interface with node mcu microcontroller are described there.

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 operation of the prototype circuit. A flow chart is presented on the actions which describes the principle of DS18B20 temperature sensor detection. Once the temperature is measured by the sensor the controller display it over a OLED screen and send it to a remote device through wifi.

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 proposed in [1] describes the system prototypes of continuous online monitoring of distribution transformer using IOT (internet of things). In this proposed real-time framework a voltage transformer, a current transformer and a lm35 temperature sensor for monitoring voltage, current, temperature respectively. These three analog values are taken in multiplexing mode connected to a programmable microcontroller arduino. Then the values are then sent directly through a wifi module under tcpip protocol to a dedicated ip that displays the data in real time chart form in any web connected pc / laptop/mobile for display. The real time data is also seen at the sending end upon a android app interfaced to the microcontroller .The supply of power is given through step down transformer 230/12v, which steps down the voltage to12v ac. This is converted to dc using a bridge rectifier and it is then regulated to +5v using a voltage regulator 7805 which is required for the operation of the arduino, 3.3 volt for the wifi unit and other component.

The system [2] proposes a Domestic Room Temperature Monitor and Regulate with the help of IoT which provides us with the information about Temperature and Humidity in a domestic environment. There are various types of sensors present in the prototype, using which the parameters can be measured. It can be used to monitor the temperature or humidity of a particular Room or a Place. The Proposed system continuously sends the data to the cloud to monitor the data from anywhere. For local monitoring, system uses a LCD to display the room temperature and humidity at that instant dynamically. The brain of the prototype is the ESP8266 based Wi-Fi module node mcu (12E).Temperature and Humidity sensor (DHT11) is connected to the node mcu. Whenever these values exceed a chosen threshold limit for each a notification is given to the user mobile through SMS. Based on the notification user can regulate the room temperature remotely using the Google assistant which is controlled through Artificial Intelligence. This system is effectively monitors and dynamically controls using voice and text commands.

The system [3] proposes the prototypes for monitoring the weather parameters with the help of IoT. The parameter that can be measured daylight, temperature and humidity. The data can bestored online, which can be used to forecast weather and eventually analyze climate patterns, as well as for other meteorological purposes. The system uses a good combination of analog and digital sensors in wired and wireless modes of operation. daylight can be monitor using a photodiode as a wired binary switch sensor and humidity can be measured Using a wired analog humidity sensor

The system [4] proposes temperature and humidity monitoring system for agriculture with the help of IoT. In this project a single channel relay on real time basis is controlled which can further be used to control water flow on the field. The main hardware of this system includes Raspberry pi with internet connectivity, Temperature and Humidity sensor. The data monitored is collected at the Web server with accurate date and time. The system is designed in such a way that system can work 24x7 and give precise data of temperature and humidity on real time basis. It can also be used in precision farming. The same system setup can also give facility to operate different kinds of devices such as water pumps, located remotely using a Mobile phone from anywhere using internet connectivity. Using these system farmers can switch on and off their pump from their home or where ever they want using their mobile phone.

The system [5] proposed the core idea is to interface temperature sensor with Raspberry Pi and collect the temperature readings and display those readings on the mobile phone. It would use hadoop cluster to save all the temperature changes. These temperature recordings maybe used for analysis in the future. it will placed the Raspberry Pi and the temperature sensor at Neova's server room and read current temperature using Cool Server Room app on the mobile phone basically.

The system [6] proposes a temperature and humidity monitoring Based on the current situation of the development. it put forward a new scheme of household temperature and humidity monitoring system. The hardware monitoring system is composed of raspberry pi, Wi-Fi dongle, DHT 11 sensor and an android mobile phone. This project deals with the automated temperature and humidity control.

CHAPTER 3

(Theory)

3.1 IoT (Internat of Things)

IoT (Internet of Things) is an advanced automation and analytics system which exploits networking, sensing, big data, and artificial intelligence technology to deliver complete systems for a product or service. These systems allow greater transparency, control, and performance when applied to any industry or system.

IoT systems have applications across industries through their unique flexibility and ability to be suitable in any environment. They enhance data collection, automation, operations, and much more through smart devices and powerful enabling technology.

3.1.1 IoT − Key Features

The most important features of IoT include artificial intelligence, connectivity, sensors, active engagement, and small device use. A brief review of these features is given below

- **AI** − IoT essentially makes virtually anything "smart", meaning it enhances every aspect of life with the power of data collection, artificial intelligence algorithms, and networks. This can mean something as simple as enhancing your refrigerator and cabinets to detect when milk and your favorite cereal run low, and to then place an order with your preferred grocer.
- **Connectivity** − New enabling technologies for networking, and specifically IoT networking, mean networks are no longer exclusively tied to major providers. Networks can exist on a much smaller and cheaper scale while still being practical. IoT creates these small networks between its system devices.
- **Sensors** − IoT loses its distinction without sensors. They act as defining instruments which transform IoT from a standard passive network of devices into an active system capable of real-world integration.
- **Active Engagement** − Much of today's interaction with connected technology happens through passive engagement. IoT introduces a new paradigm for active content, product, or service engagement.
- **Small Devices** − Devices, as predicted, have become smaller, cheaper, and more powerful over time. IoT exploits purpose-built small devices to deliver its precision, scalability, and versatility.

3.1.2 IoT − Advantages

The advantages of IoT span across every area of lifestyle and business. Here is a list of some of the advantages that IoT has to offer

- **Improved Customer Engagement** − Current analytics suffer from blind-spots and significant flaws in accuracy; and as noted, engagement remains passive. IoT completely transforms this to achieve richer and more effective engagement with audiences.
- **Technology Optimization** − The same technologies and data which improve the customer experience also improve device use, and aid in more potent improvements to technology. IoT unlocks a world of critical functional and field data.
- **Reduced Waste** − IoT makes areas of improvement clear. Current analytics give us superficial insight, but IoT provides real-world information leading to more effective management of resources.

- **Enhanced Data Collection** − Modern data collection suffers from its limitations and its design for passive use. IoT breaks it out of those spaces, and places it exactly where humans really want to go to analyze our world. It allows an accurate picture of everything.

3.1.3 IoT − Disadvantages

Though IoT delivers an impressive set of benefits, it also presents a significant set of challenges. Here is a list of some its major issues

- **Security** − IoT creates an ecosystem of constantly connected devices communicating over networks. The system offers little control despite any security measures. This leaves users exposed to various kinds of attackers.
- **Privacy** − The sophistication of IoT provides substantial personal data in extreme detail without the user's active participation.
- **Complexity** − Some find IoT systems complicated in terms of design, deployment, and maintenance given their use of multiple technologies and a large set of new enabling technologies.
- **Flexibility** − Many are concerned about the flexibility of an IoT system to integrate easily with another. They worry about finding themselves with several conflicting or locked systems.
- **Compliance** − IoT, like any other technology in the realm of business, must comply with regulations. Its complexity makes the issue of compliance seem incredibly challenging when many consider standard software compliance a battle.

3.1.4 Iot Software

IoT software addresses its key areas of networking and action through platforms, embedded systems, partner systems, and middleware. These individual and master applications are responsible for data collection, device integration, real-time analytics, and application and process extension within the IoT network. They exploit integration with critical business systems (e.g., ordering systems, robotics, scheduling, and more) in the execution of related tasks.

- **Data Collection**
  This software manages sensing, measurements, light data filtering, light data security, and aggregation of data. It uses certain protocols to aid sensors in connecting with real-time, machine-to-machine networks. Then it collects data from multiple devices and distributes it in accordance with settings. It also works in reverse by distributing data over devices. The system eventually transmits all collected data to a central server.

- **Device Integration**
  Software supporting integration binds (dependent relationships) all system devices to create the body of the IoT system. It ensures the necessary cooperation and stable networking between devices. These applications are the defining software technology of the IoT network because without them, it is not an IoT system. They manage the various applications, protocols, and limitations of each device to allow communication.

- **Real-Time Analytics**
  These applications take data or input from various devices and convert it into viable actions or clear patterns for human analysis. They analyze information based on various settings and designs in order to perform automation-related tasks or provide the data required by industry.

- **Application and Process Extension**
  These applications extend the reach of existing systems and software to allow a wider, more effective system. They integrate predefined devices for specific purposes such as allowing certain mobile devices or engineering instruments access. It supports improved productivity and more accurate data collection.

3.1.5 Internet of Things - Technology and Protocols

IoT primarily exploits standard protocols and networking technologies. However, the major enabling technologies and protocols of IoT are RFID, NFC, low-energy Bluetooth, low-energy wireless, low-energy radio protocols, LTE-A, and WiFi-Direct. These technologies support the specific networking functionality needed in an IoT system in contrast to a standard uniform network of common systems.

NFC and RFID

RFID (radio-frequency identification) and NFC (near-field communication) provide simple, lowenergy, and versatile options for identity and access tokens, connection bootstrapping, and payments.

- RFID technology employs 2-way radio transmitter-receivers to identify and track tags associated with objects.
- NFC consists of communication protocols for electronic devices, typically a mobile device and a standard device.

Low-Energy Bluetooth

This technology supports the low-power, long-use need of IoT function while exploiting a standard technology with native support across systems.

Low-Energy Wireless

This technology replaces the most power hungry aspect of an IoT system. Though sensors and other elements can power down over long periods, communication links (i.e., wireless) must remain in listening mode. Low-energy wireless not only reduces consumption, but also extends the life of the device through less use.

Radio Protocols

ZigBee, Z-Wave, and Thread are radio protocols for creating low-rate private area networks. These technologies are low-power, but offer high throughput unlike many similar options. This increases the power of small local device networks without the typical costs.

LTE-A

LTE-A, or LTE Advanced, delivers an important upgrade to LTE technology by increasing not only its coverage, but also reducing its latency and raising its throughput. It gives IoT a tremendous power through expanding its range, with its most significant applications being vehicle, UAV, and similar communication.

WiFi-Direct

WiFi-Direct eliminates the need for an access point. It allows P2P (peer-to-peer) connections with the speed of WiFi, but with lower latency. WiFi-Direct eliminates an element of a network that often bogs it down, and it does not compromise on speed or throughput.

3.1.6 Internet of Things - Common Uses

IoT has applications across all industries and markets. It spans user groups from those who want to reduce energy use in their home to large organizations who want to streamline their operations. It proves not just useful, but nearly critical in many industries as technology advances and we move towards the advanced automation imagined in the distant future.

Engineering, Industry, and Infrastructure

Applications of IoT in these areas include improving production, marketing, service delivery, and safety. IoT provides a strong means of monitoring various processes; and real transparency creates greater visibility for improvement opportunities.

The deep level of control afforded by IoT allows rapid and more action on those opportunities, which include events like obvious customer needs, nonconforming product, malfunctions in equipment, problems in the distribution network, and more.

Government and Safety

IoT applied to government and safety allows improved law enforcement, defense, city planning, and economic management. The technology fills in the current gaps, corrects many current flaws, and expands the reach of these efforts. For example, IoT can help city planners have a clearer view of the impact of their design, and governments have a better idea of the local economy.

Home and Office

In our daily lives, IoT provides a personalized experience from the home to the office to the organizations we frequently do business with. This improves our overall satisfaction, enhances productivity, and improves our health and safety. For example, IoT can help us customize our office space to optimize our work.

Health and Medicine

IoT pushes us towards our imagined future of medicine which exploits a highly integrated network of sophisticated medical devices. Today, IoT can dramatically enhance medical research, devices, care, and emergency care. The integration of all elements provides more accuracy, more attention to detail, faster reactions to events, and constant improvement while reducing the typical overhead of medical research and organizations.

3.2 NODE MCU

NodeMCU is an open source IoT platform. It includes firmware which runs on the ESP8266 from Espressif, and hardware which is based on the ESP12 module.The term "NodeMCU" by default refers to the firmware rather than the dev kits. The firmware uses the Lua scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson,and spiffs.

Figure 2: NODE MCU Development board

NodeMCU was created shortly after the ESP8266 came out. On December 30, 2013, Espressiv system began production of the ESP8266. The ESP8266 is a Wi-Fi SoC integrated with a Tensilica Xtensa LX106 core,widely used in IoT applications

The ESP8266 is a low-cost Wi-Fi microchip with full TCP/IP stack and microcontroller capability produced by Shanghai-based Chinese manufacturer, Espressif Systems.

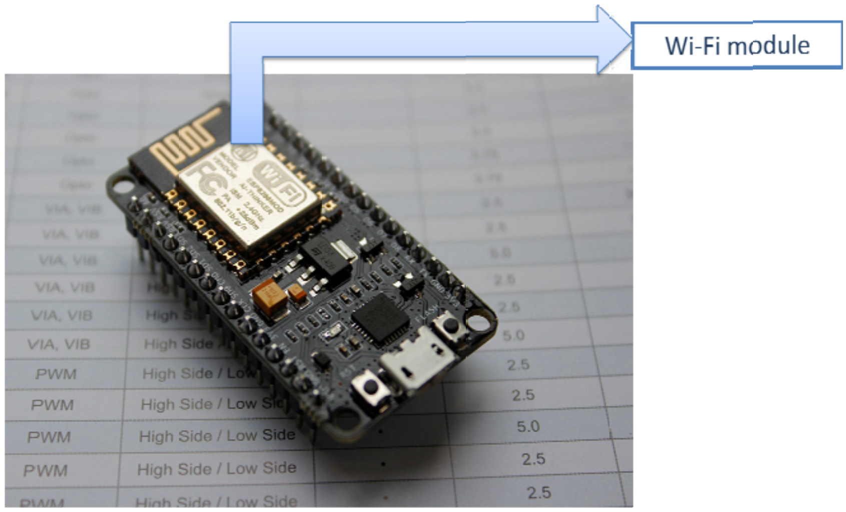


Figure 3: NODE MCU with inbuilt wifi module

3.2.1 Pin configuration of NODE MCU development board

This module provides access to the GPIO (General Purpose Input/Output) subsystem. All access is based on the I/O index number on the NodeMCU dev kits, not the internal GPIO pin. For example, the D0 pin on the dev kit is mapped to the internal GPIO pin 16.
Please refer to the below GPIO pin maps for the index↔gpio mapping.

Table 1: Node MCU index ↔ gpio mapping

IO index	ESP8266 pin	IO index	ESP8266 pin
0 [*]	GPIO16	7	GPIO13
1	GPIO5	8	GPIO15
2	GPIO4	9	GPIO3
3	GPIO0	10	GPIO1
4	GPIO2	11	GPIO9
5	GPIO14	12	GPIO10
6	GPIO12		

[*] D0 (GPIO16) can only be used as GPIO read/write. No support for open-drain/interrupt/pwm/i2c

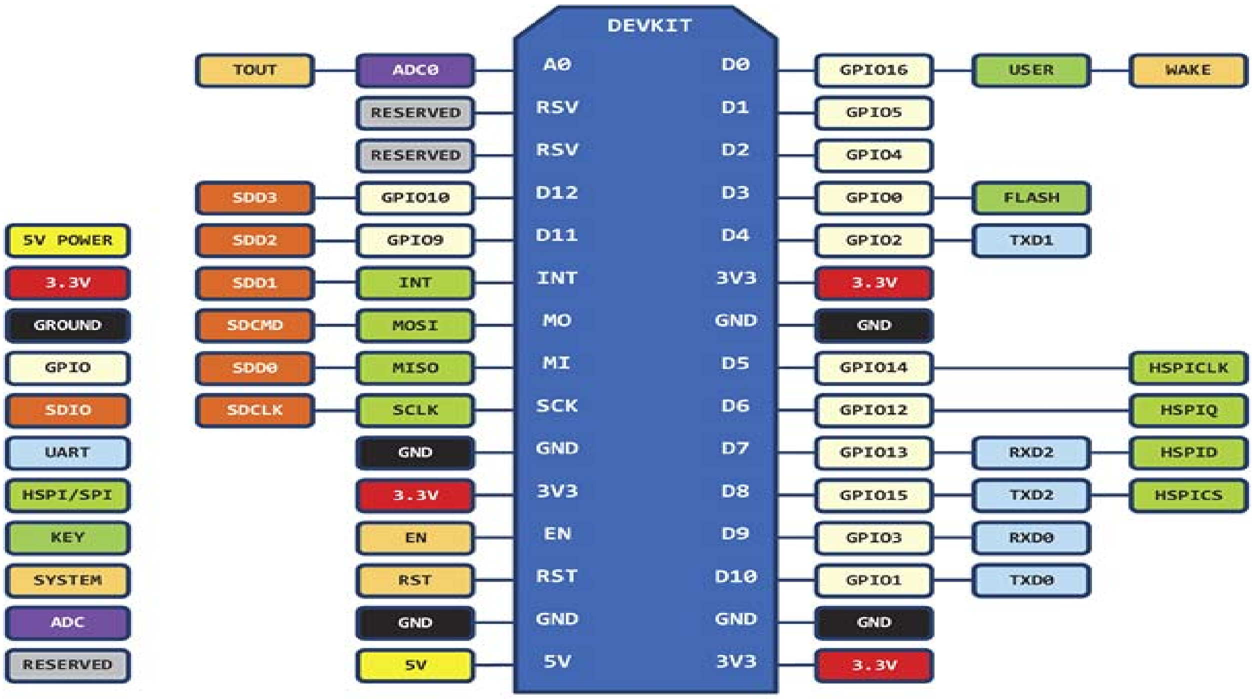


Figure 4: NODE MCU pin configuration

3.2.2 Installation of Node MCU & Coding

Mostly these days devices download and install drivers on their own, automatically. Windows doesn't know how to talk to the USB driver on the Node MCU so it can't figure out that the board *is* a Node MCU and proceed normally.

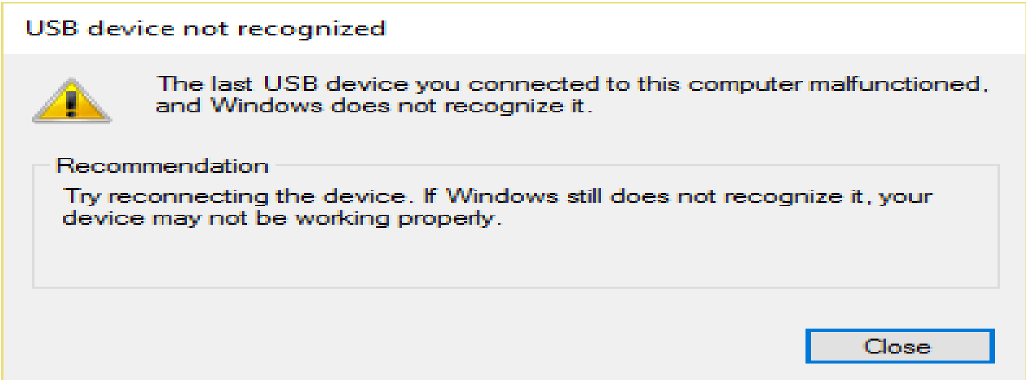


Figure 5: Snapshot of the installation process of NODE MCU

- Node MCU Amica is a ESP8266 wifi Module based development board. It has got Micro USB slot that can be directly connected to the computer or other USB host devices. It has got 15X2 Header pins and a Micro USB slot, the headers can be mounted on breadboard and the micro USB slot is for connection to USB host device that may be a computer. It has got CP2102 USB to serial converter.
- In order to install CP2102 (USB to Serial Converter), user will need to download the driver for the same.
- Once user downloaded drivers as per the proper operating system; the system has got connected with the node MCU.
- From the device manager of the computer note down the COM port allocated to the newly connected USB device i.e. the node MCU Amcia. This com port number will be required while using Node MCU Amica.

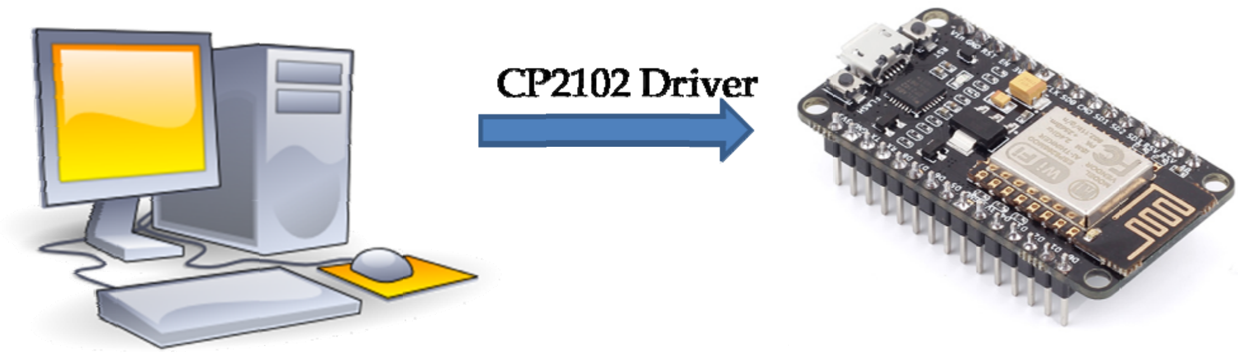


Figure 6: Driver Installation for NODE MCU

3.2.3 Interfacing of node mcu with arduino IDE

Firstly open the Arduino IDE. Go to files and click on the preference in the Arduino IDE

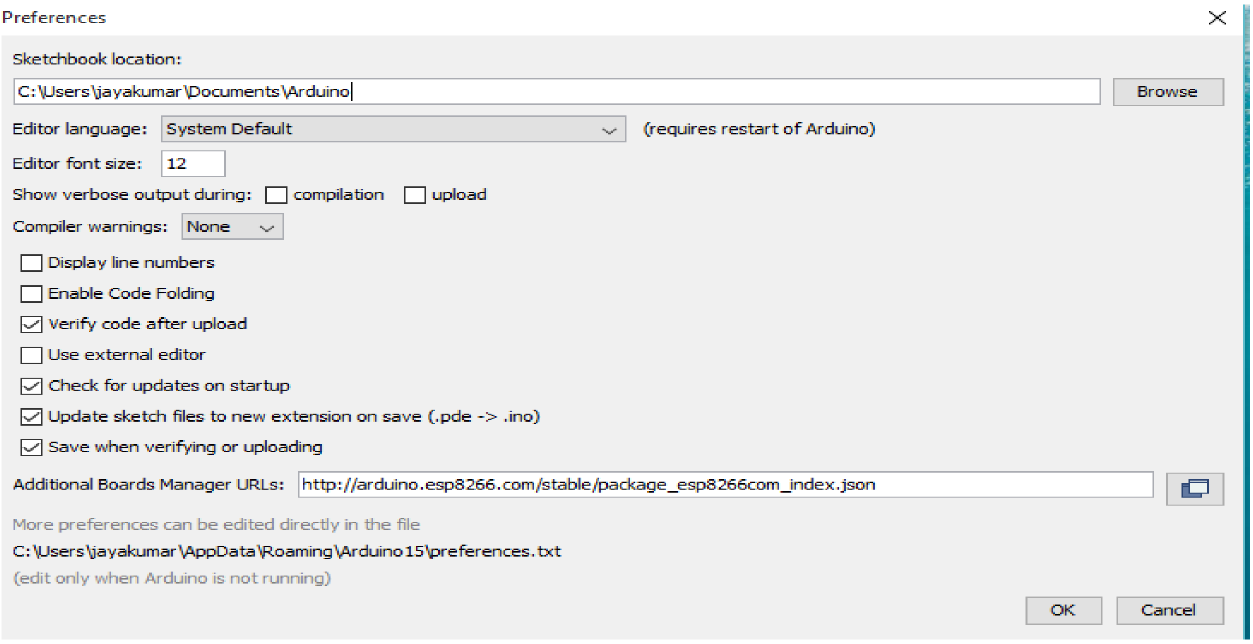


Figure 7: Arduino IDE preferences

copy the below code in the Additional boards Manager
http://arduino.esp8266.com/stable/package_esp8266com_index.json
click OK to close the preference Tab

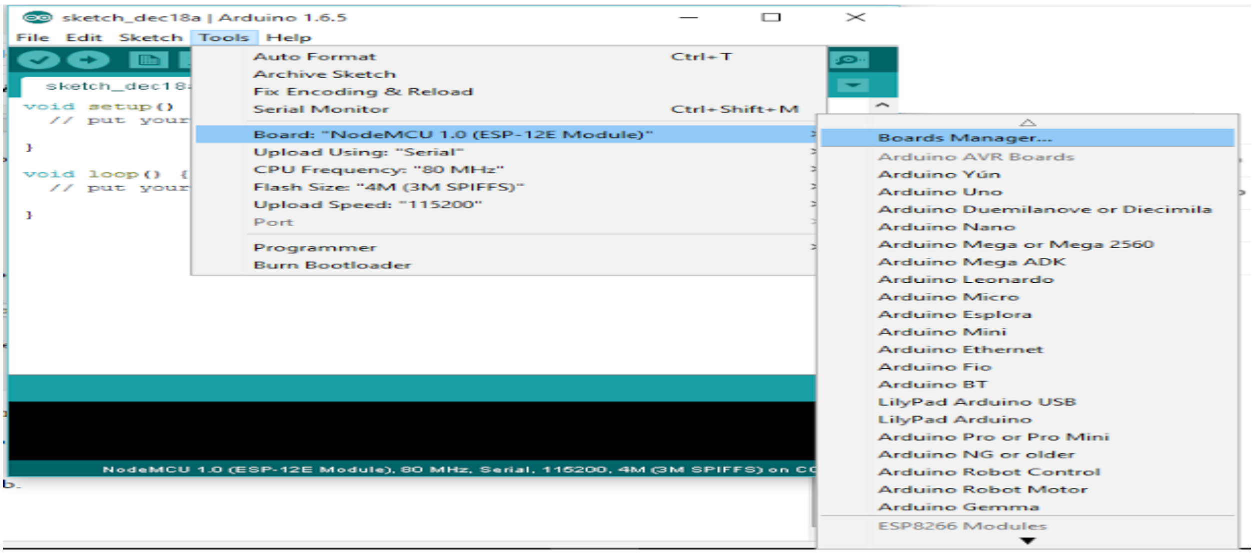


Figure 8: Arduino IDE board manager installation

After completing the above steps , go to Tools and board, and then select board Manager

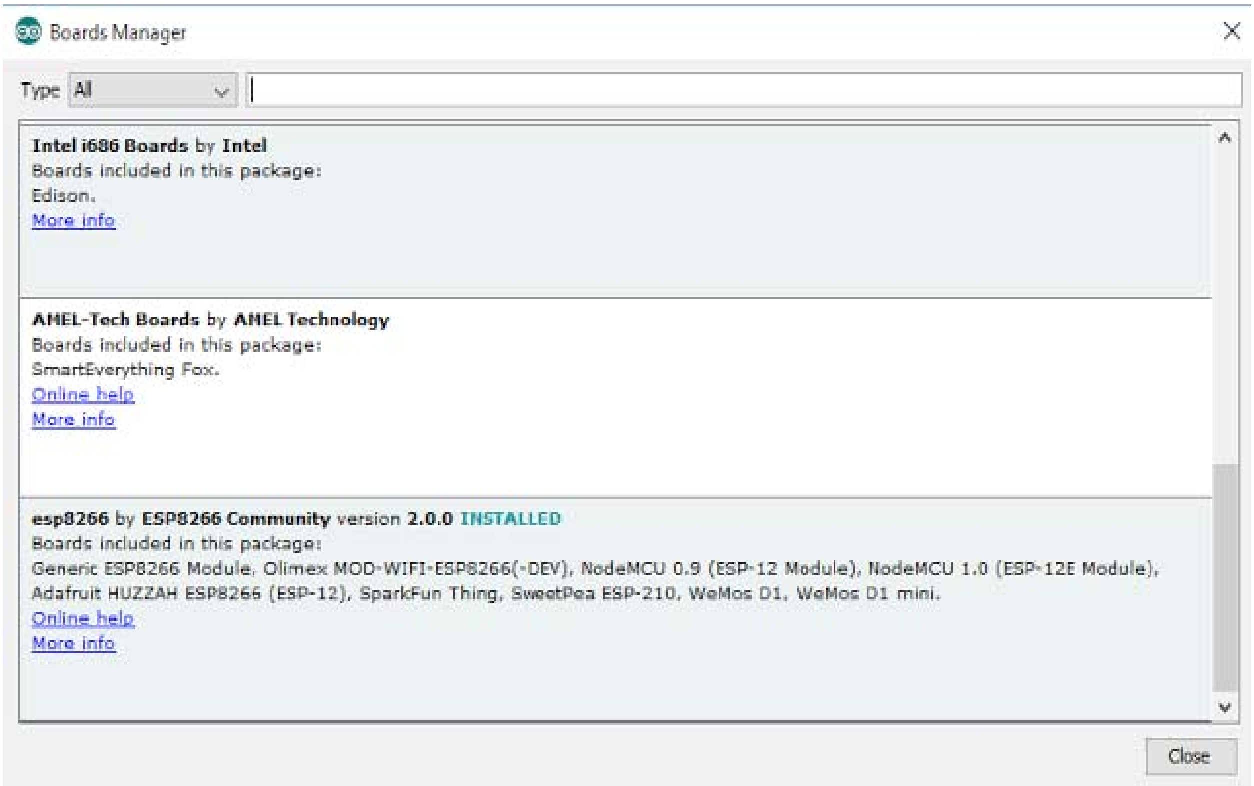


Figure 9: ESP 8266 board installation in Arduino

Navigate to esp8266 by esp8266 community and install the software for Arduino. Once all the above process had been completed we are ready to program our esp8266 with Arduino IDE.

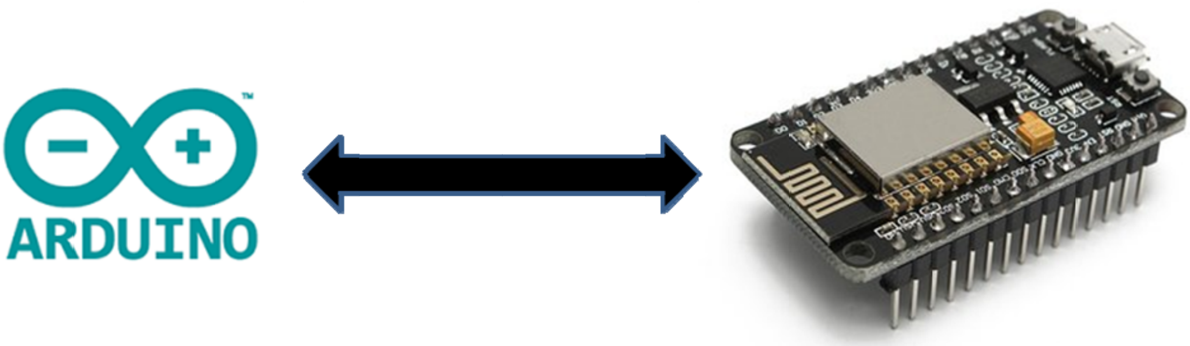


Figure 10: NODE MCU interfacing with Arduino

3.3 DS18B20 Programmable Resolution 1-Wire Digital Thermometer

The DS18B20 digital thermometer provides 9-bit to 12-bit Celsius temperature measurements and has an alarm function with nonvolatile user-programmable upper and lower trigger points. The DS18B20 communicates over a 1-Wire bus that by definition requires only one data line (and ground) for communication with a central microprocessor. It has an operating temperature range of -55°C to +125°C and is accurate to ±0.5°C over the range of -10°C to +85°C. In addition, the DS18B20 can derive power directly from the data line ("parasite power"), eliminating the need for an external power supply.

Each DS18B20 has a unique 64-bit serial code, which allows multiple DS18B20s to function on the same 1-Wire bus. Thus, it is simple to use one microprocessor to control many DS18B20s distributed over a large area. Applications that can benefit from this feature include HVAC environmental controls, temperature monitoring systems inside buildings, equipment, or machinery, and process monitoring and control systems.

3.3.1 FEATURES OF DS18B20 TEMPERATURE SENSOR

- Unique 1-Wire® Interface Requires Only One Port Pin for Communication
- Each Device has a Unique 64-Bit Serial Code Stored in an On-Board ROM
- Multidrop Capability Simplifies Distributed Temperature-Sensing Applications
- Requires No External Components
- Can Be Powered from Data Line; Power Supply Range is 3.0V to 5.5V
- Measures Temperatures from -55°C to +125°C (-67°F to +257°F)
- ±0.5°C Accuracy from -10°C to +85°C
- Thermometer Resolution is User Selectable from 9 to 12 Bits
- Converts Temperature to 12-Bit Digital Word in 750ms (Max)
- User-Definable Nonvolatile (NV) Alarm Settings
- Alarm Search Command Identifies and Addresses Devices Whose Temperature is Outside Programmed Limits (Temperature Alarm Condition)
- Available in 8-Pin SO (150 mils), 8-Pin µSOP, and 3-Pin TO-92 Packages
- Software Compatible with the DS1822
- Applications Include Thermostatic Controls, Industrial Systems, Consumer Products, Thermometers, or Any Thermally Sensitive System

3.3.2 PIN CONFIGURATIONS

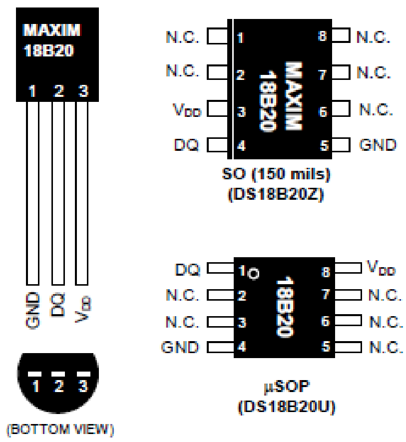


Figure 11: Pin description of DS18B20

3.3.3 OVERVIEW OF DS18B20

Figure 4 shows a block diagram of the DS18B20, and pin descriptions are given in the *Pin Description* table. The 64-bit ROM stores the device's unique serial code. The scratchpad memory contains the 2-byte temperature register that stores the digital output from the temperature sensor. In addition, the scratchpad provides access to the 1-byte upper and lower alarm trigger registers (TH and TL) and the 1-byte configuration register. The configuration register allows the user to set the resolution of the temperature-to-digital conversion to 9, 10, 11, or 12 bits. The TH, TL, and configuration registers are nonvolatile (EEPROM), so they will retain data when the device is powered down.

The DS18B20 uses Maxim's exclusive 1-Wire bus protocol that implements bus communication using one control signal. The control line requires a weak pullup resistor since all devices are linked to the bus via a 3-state or open-drain port (the DQ pin in the case of the DS18B20). In this bus system, the microprocessor (the master device) identifies and addresses devices on the bus using each device's unique 64-bit code. Because each device has a unique code, the number of devices that can be addressed on one bus is virtually unlimited. The 1-Wire bus protocol, including detailed explanations of the commands and "time slots," is covered in the *1-Wire Bus System* section.

Another feature of the DS18B20 is the ability to operate without an external power supply. Power is instead supplied through the 1-Wire pullup resistor via the DQ pin when the bus is high. The high bus signal also charges an internal capacitor (CPP), which then supplies power to the device when the bus is low. This method of deriving power from the 1-Wire bus is referred to as "parasite power." As an alternative, the DS18B20 may also be powered by an external supply on VDD.

DS18B20 Block Diagram

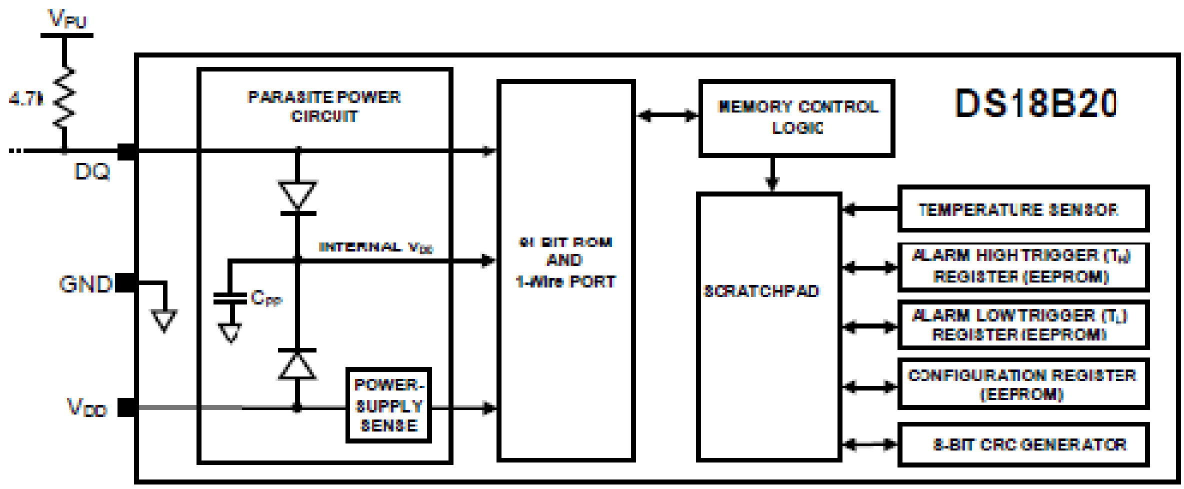


Figure 12: DS18B20 Block Diagram

3.3.4 OPERATION—MEASURING TEMPERATURE

The core functionality of the DS18B20 is its direct-to-digital temperature sensor. The resolution of the temperature sensor is user-configurable to 9, 10, 11, or 12 bits, corresponding to increments of 0.5°C, 0.25°C, 0.125°C, and 0.0625°C, respectively. The default resolution at power-up is 12-bit. The DS18B20 powers up in a low-power idle state. To initiate a temperature measurement and A-to-D conversion, the master must issue a Convert T [44h] command. Following the conversion, the resulting thermal data is stored in the 2-byte temperature register in the scratchpad memory and the DS18B20 returns to its idle state. If the DS18B20 is powered by an external supply, the master can issue "read time slots" after the Convert T command and the DS18B20 will respond by transmitting 0 while the temperature conversion is in progress and 1 when the conversion is done. If the

DS18B20 is powered with parasite power, this notification technique cannot be used since the bus must be pulled high by a strong pullup during the entire temperature conversion. The bus requirements for parasite power are explained in detail in the *Powering the DS18B20* section.

The DS18B20 output temperature data is calibrated in degrees Celsius; for Fahrenheit applications, a lookup table or conversion routine must be used. The temperature data is stored as a 16-bit sign-extended two's complement number in the temperature register (see Figure 13). The sign bits (S) indicate if the temperature is positive or negative: for positive numbers S = 0 and for negative numbers S = 1. If the DS18B20 is configured for 12-bit resolution, all bits in the temperature register will contain valid data. For 11-bit resolution, bit 0 is undefined. For 10-bit resolution, bits 1 and 0 are undefined, and for 9-bit resolution bits 2, 1, and 0 are undefined. Table 1 gives examples of digital output data and the corresponding temperature reading for 12-bit resolution conversions.

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LS BYTE	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}
	BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8
MS BYTE	S	S	S	S	S	2^6	2^5	2^4

S = SIGN

Figure 13: Temperature Register Format

Table 2: Temperature/Data Relationship

TEMPERATURE (°C)	DIGITAL OUTPUT (BINARY)	DIGITAL OUTPUT (HEX)
+125	0000 0111 1101 0000	07D0h
+85*	0000 0101 0101 0000	0550h
+25.0625	0000 0001 1001 0001	0191h
+10.125	0000 0000 1010 0010	00A2h
+0.5	0000 0000 0000 1000	0008h
0	0000 0000 0000 0000	0000h
-0.5	1111 1111 1111 1000	FFF8h
-10.125	1111 1111 0101 1110	FF5Eh
-25.0625	1111 1110 0110 1111	FE6Fh
-55	1111 1100 1001 0000	FC90h

*The power-on reset value of the temperature register is +85°C.

3.3.5 ROM COMMANDS

After the bus master has detected a presence pulse, it can issue a ROM command. These commands operate on the unique 64-bit ROM codes of each slave device and allow the master to single out a specific device if many are present on the 1-Wire bus. These commands also allow the master to determine how many and what types of devices are present on the bus or if any device has experienced an alarm condition. There are five ROM commands, and each command is 8 bits long.

The master device must issue an appropriate ROM command before issuing a DS18B20 function command.

SEARCH ROM [F0h]

When a system is initially powered up, the master must identify the ROM codes of all slave devices on the bus, which allows the master to determine the number of slaves and their device types. The master learns the ROM codes through a process of elimination that requires the master to perform a Search ROM cycle (i.e., Search ROM command followed by data exchange) as many times as necessary to identify all of the slave devices. If there is only one slave on the bus, the simpler Read ROM command (see below) can be used in place of the Search ROM process. For a detailed explanation of the Search ROM procedure, refer to the *iButton® Book of Standards* at www.maxim-ic.com/ibuttonbook. After every Search ROM cycle, the bus master must return to Step 1 (Initialization) in the transaction sequence.

READ ROM [33h]

This command can only be used when there is one slave on the bus. It allows the bus master to read the slave's 64-bit ROM code without using the Search ROM procedure. If this command is used when there is more than one slave present on the bus, a data collision will occur when all the slaves attempt to respond at the same time.

MATCH ROM [55h]

The match ROM command followed by a 64-bit ROM code sequence allows the bus master to address a specific slave device on a multidrop or single-drop bus. Only the slave that exactly matches the 64-bit ROM code sequence will respond to the function command issued by the master; all other slaves on the bus will wait for a reset pulse.

SKIP ROM [CCh]

The master can use this command to address all devices on the bus simultaneously without sending out any ROM code information. For example, the master can make all DS18B20s on the bus perform simultaneous temperature conversions by issuing a Skip ROM command followed by a Convert T [44h] command.

Note that the Read Scratchpad [BEh] command can follow the Skip ROM command only if there is a single slave device on the bus. In this case, time is saved by allowing the master to read from the slave without sending the device's 64-bit ROM code. A Skip ROM command followed by a Read Scratchpad command will cause a data collision on the bus if there is more than one slave since multiple devices will attempt to transmit data simultaneously.

ALARM SEARCH [ECh]

The operation of this command is identical to the operation of the Search ROM command except that only slaves with a set alarm flag will respond. This command allows the master device to

determine if any DS18B20s experienced an alarm condition during the most recent temperature conversion. After every Alarm Search cycle (i.e., Alarm Search command followed by data exchange), the bus master must return to Step 1 (Initialization) in the transaction sequence. See the *Operation—Alarm Signaling* section for an explanation of alarm flag operation.

3.3.6 DS18B20 FUNCTION COMMANDS

After the bus master has used a ROM command to address the DS18B20 with which it wishes to communicate, the master can issue one of the DS18B20 function commands. These commands allow the master to write to and read from the DS18B20's scratchpad memory, initiate temperature conversions and determine the power supply mode. The DS18B20 function commands, which are described below, are summarized in Table 3 and illustrated by the flowchart in Figure 12.

CONVERT T [44h]

This command initiates a single temperature conversion. Following the conversion, the resulting thermal data is stored in the 2-byte temperature register in the scratchpad memory and the DS18B20 returns to its low-power idle state. If the device is being used in parasite power mode, within 10µs (max) after this command is issued the master must enable a strong pullup on the 1-Wire bus for the duration of the conversion (tCONV) as described in the *Powering the DS18B20* section. If the DS18B20 is powered by an external supply, the master can issue read time slots after the Convert T command and the DS18B20 will respond by transmitting a 0 while the temperature conversion is in progress and a 1 when the conversion is done. In parasite power mode this notification technique cannot be used since the bus is pulled high by the strong pullup during the conversion.

WRITE SCRATCHPAD [4Eh]

This command allows the master to write 3 bytes of data to the DS18B20's scratchpad. The first data byte is written into the TH register (byte 2 of the scratchpad), the second byte is written into the TL register (byte 3), and the third byte is written into the configuration register (byte 4). Data must be transmitted least significant bit first. All three bytes MUST be written before the master issues a reset, or the data may be corrupted.

READ SCRATCHPAD [BEh]

This command allows the master to read the contents of the scratchpad. The data transfer starts with the least significant bit of byte 0 and continues through the scratchpad until the 9th byte (byte 8 – CRC) is read. The master may issue a reset to terminate reading at any time if only part of the scratchpad data is needed.

COPY SCRATCHPAD [48h]

This command copies the contents of the scratchpad TH, TL and configuration registers (bytes 2, 3 and 4) to EEPROM. If the device is being used in parasite power mode, within 10µs (max) after this command is issued the master must enable a strong pullup on the 1-Wire bus for at least 10ms as described in the *Powering the DS18B20* section.

RECALL E2 [B8h]

This command recalls the alarm trigger values (TH and TL) and configuration data from EEPROM and places the data in bytes 2, 3, and 4, respectively, in the scratchpad memory. The master device can issue read time slots following the Recall E2 command and the DS18B20 will indicate the status of the recall by transmitting 0 while the recall is in progress and 1 when the recall is done. The recall operation happens automatically at power-up, so valid data is available in the scratchpad as soon as power is applied to the device.

READ POWER SUPPLY [B4h]

The master device issues this command followed by a read time slot to determine if any DS18B20s on the bus are using parasite power. During the read time slot, parasite powered DS18B20s will pull the bus low, and externally powered DS18B20s will let the bus remain high. See the *Powering the DS18B20* section for usage information for this command.

Table 3: DS18B20 Function Command Set

COMMAND	DESCRIPTION	PROTOCOL	1-Wire BUS ACTIVITYAFTER COMMAND IS ISSUED	NOTES
TEMPERATURE CONVERSION COMMANDS				
Convert T	Initiates temperature conversion.	44h	DS18B20 transmits conversion status to master (not applicable for parasite-powered DS18B20s).	1
MEMORY COMMANDS				
Read Scratchpad	Reads the entire scratchpad including the CRC byte.	BEh	DS18B20 transmits up to 9 data bytes to master.	2
Write Scratchpad	Writes data into scratchpad bytes 2, 3, and 4 (T_H, T_L, and configuration registers).	4Eh	Master transmits 3 data bytes to DS18B20.	3
Copy Scratchpad	Copies T_H, T_L, and configuration register data from the scratchpad to EEPROM.	48h	None	1
Recall E^2	Recalls T_H, T_L, and configuration register data from EEPROM to the scratchpad.	B8h	DS18B20 transmits recall status to master.	
Read Power Supply	Signals DS18B20 power supply mode to the master.	B4h	DS18B20 transmits supply status to master.	

Note 1: For parasite-powered DS18B20s, the master must enable a strong pullup on the 1-Wire bus during temperature conversions and copies from the scratchpad to EEPROM. No other bus activity may take place during this time.
Note 2: The master can interrupt the transmission of data at any time by issuing a reset.
Note 3: All three bytes must be written before a reset is issued.

3.3.7 Interfacing the temperature sensor with the microcontroller

The one wire temperature sensor DS18B20 can be interfaced with most of the microcontroller. In this project it is connected with AT89c51 microcontroller. A 4.7 KΩ pullup resistor is required for the proper connection. The beauty of this temperature sensor is that it occupies only one digital IO pin of the microcontroller. It can be possible to connect infinite number of sensor in the same pin because each sensor has a unique 64 bit address. In figure 14 it is shown that a temperature sensor is connected to NODE MCU microcontroller.

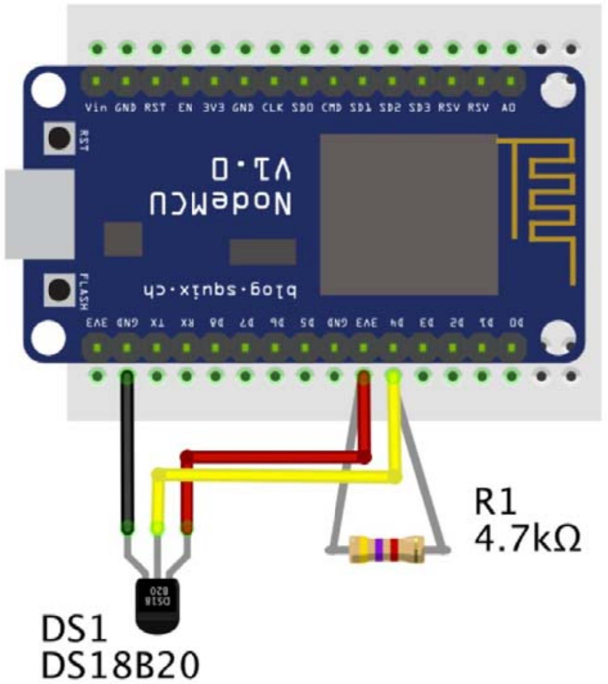


Figure 14: Interfacing DS18B20 with NODE MCU

3.4 Overview of the project

The following process describes how to create an account in Blynk and generate a unique ID against a particular device. The ID is the identifier for the particular device in the Blynk server.

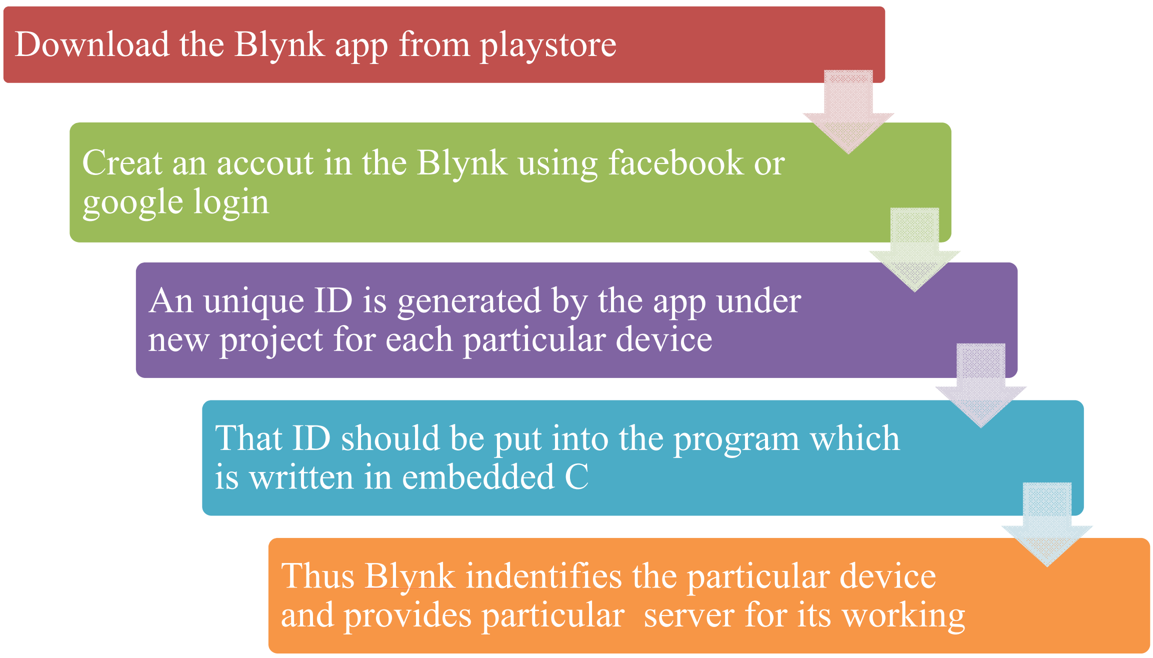


Figure 15: Account creation and generation of unique ID in Blynk server

Once the unique ID is generated the next step is to include that key in the coding which is written in embedded C for communication between the NODE MCU and Blynk server. The process is described below.

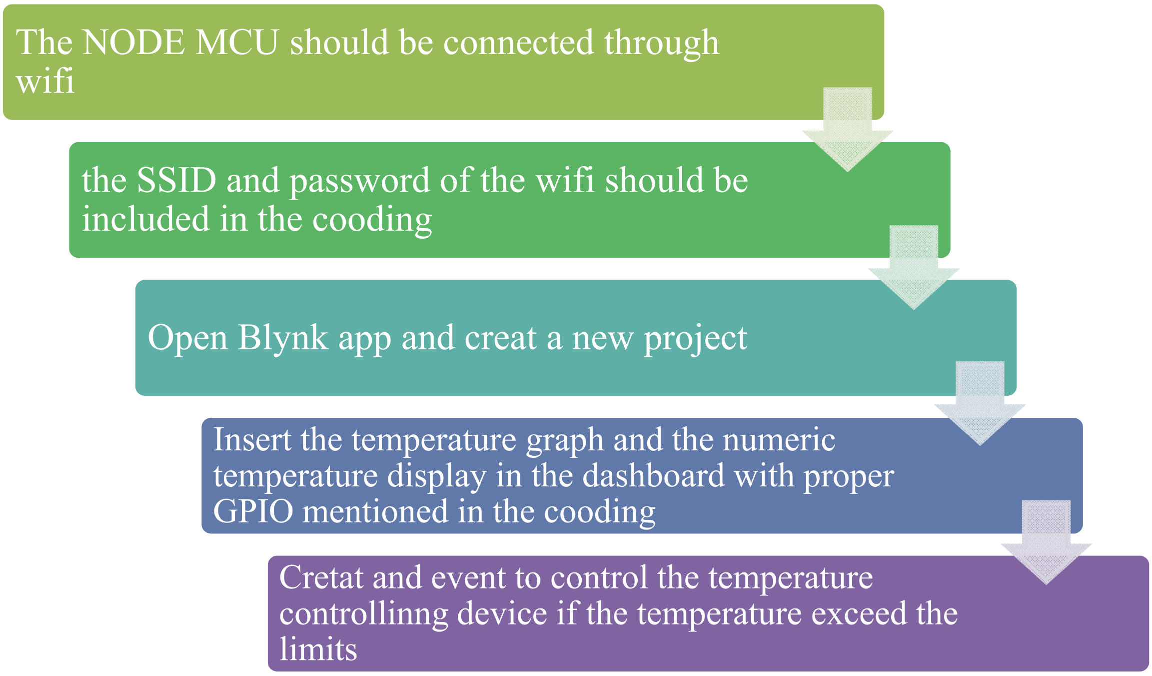


Figure 16: working process of the temperature control device

3.5 Circuit Diagram

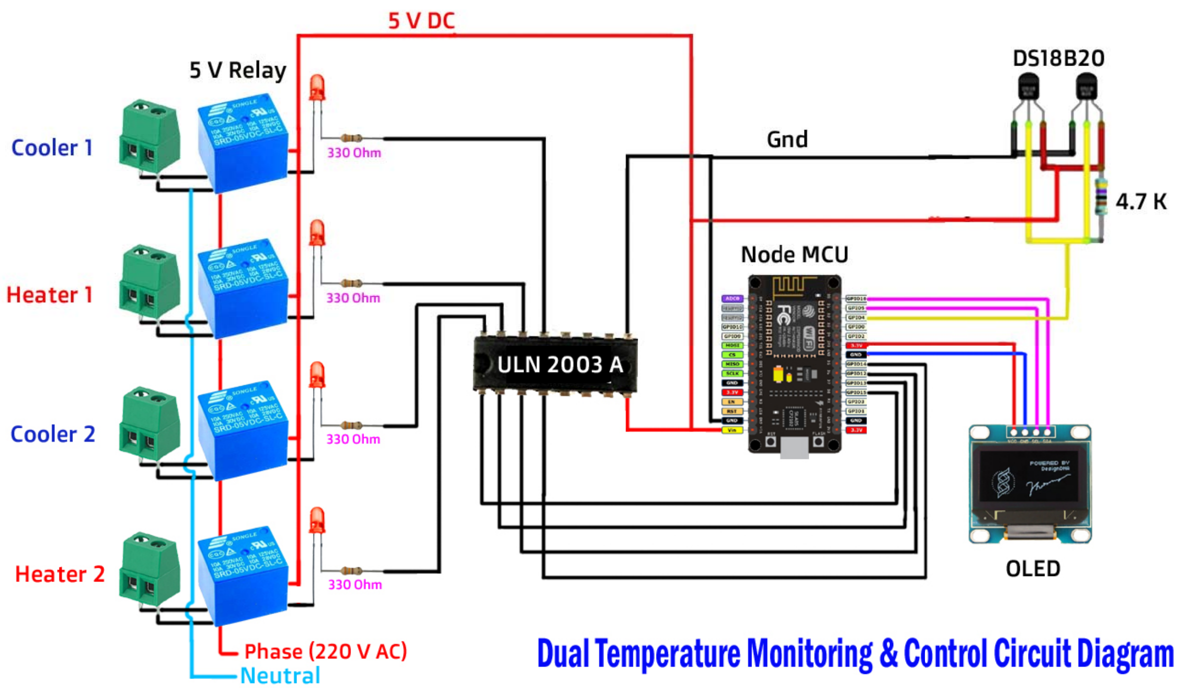


Figure 17: Connection diagram of the dual temperature sensor circuit

Figure 18: Blynk app user interface

CHAPTER 4

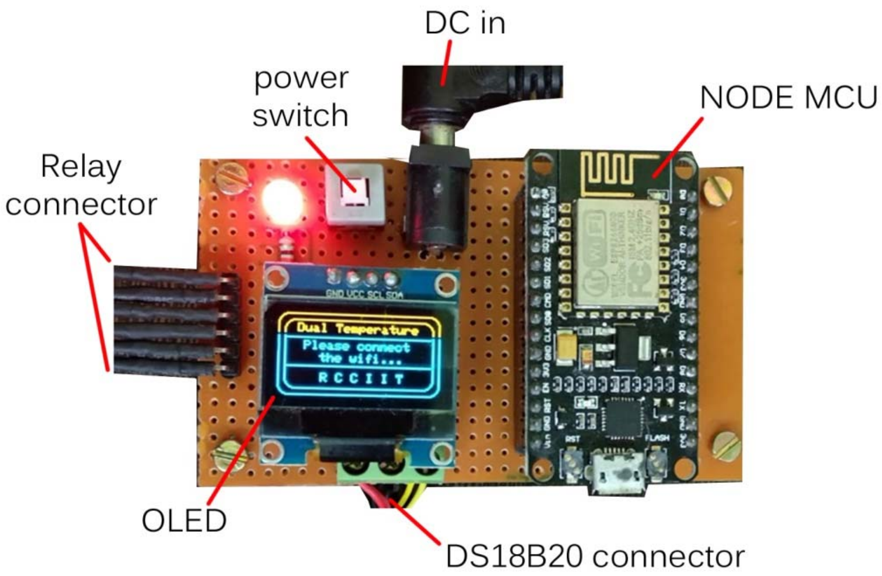
(Hardware Modeling)

4.1 Main features of the prototype

The features of the developed prototype are:

- Globally controlled (throughout the world)
- Real time temperature display in the OLED and mobile screen
- 2 independent loads control (250 volt, 7 amp max, ON/OFF control)
- Extension for relay board connection
- Secure control
- 5 Volt operation (both control board and relay board)

4.2 Photographs of the prototype



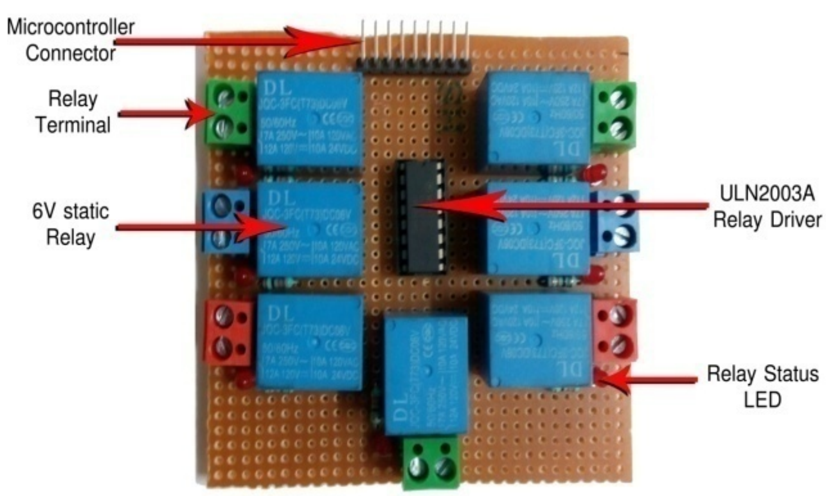
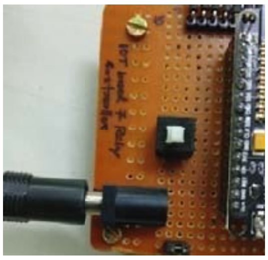


Figure 19: Main Controller and Relay board

4.3 Step by step operation of the prototype

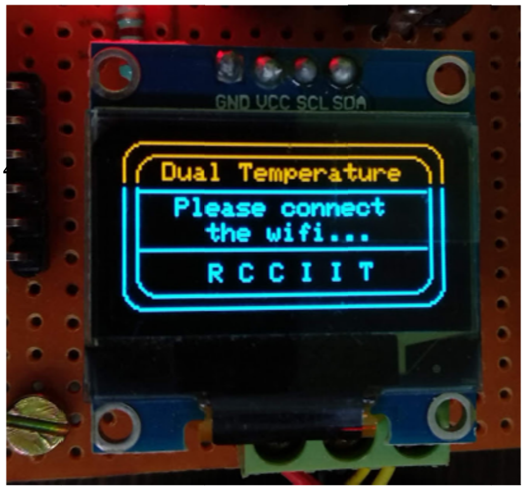


1. Connect the DC adapter (5V, 1A) to the DC jack.

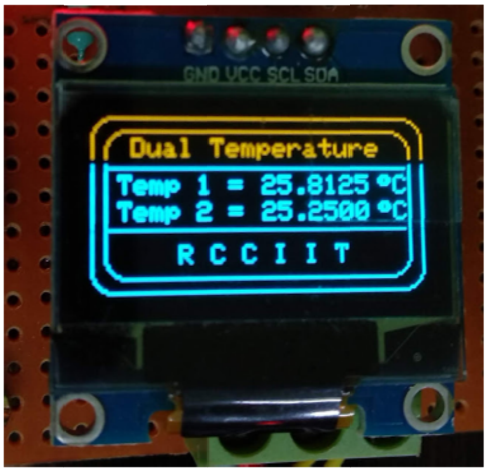
2. Enable the pre-specified wifi (known SSID and password mentioned in the code)

3. After enabling the wifi check the circuit OLED display.

Before connection



After connection



4. Open the blynk app in the mobile and check the temperature.

5. In the ‘Eventer’ wizard of the Blynk app set the temperature limits and the corresponding IO to operate the controlling device (heater / cooler).

6. Monitor the temperature.

4.4 Components required

Table 4: Component listing

Sl. No.	Component	Qtn
1.	DS 18B20	2
2.	NODE MCU	1
3.	4.7 k Resistance	1
4.	ULN 2003 A IC	1
5.	Static Relay (5 volt)	4
6.	0.96” OLED	1
7.	General blank PCB (KS 100)	1
8.	3 mm LED	1
9.	Relay terminals	4
10.	Single strand wire	2m
11.	IC base (18 pin)	1
12.	Female PCB Header Connector	1
13.	SPDT switch	1
14.	Male PCB Header Connector	1

4.5 Hardware connection

4.5.1 Relay Driver interfacing with microcontroller

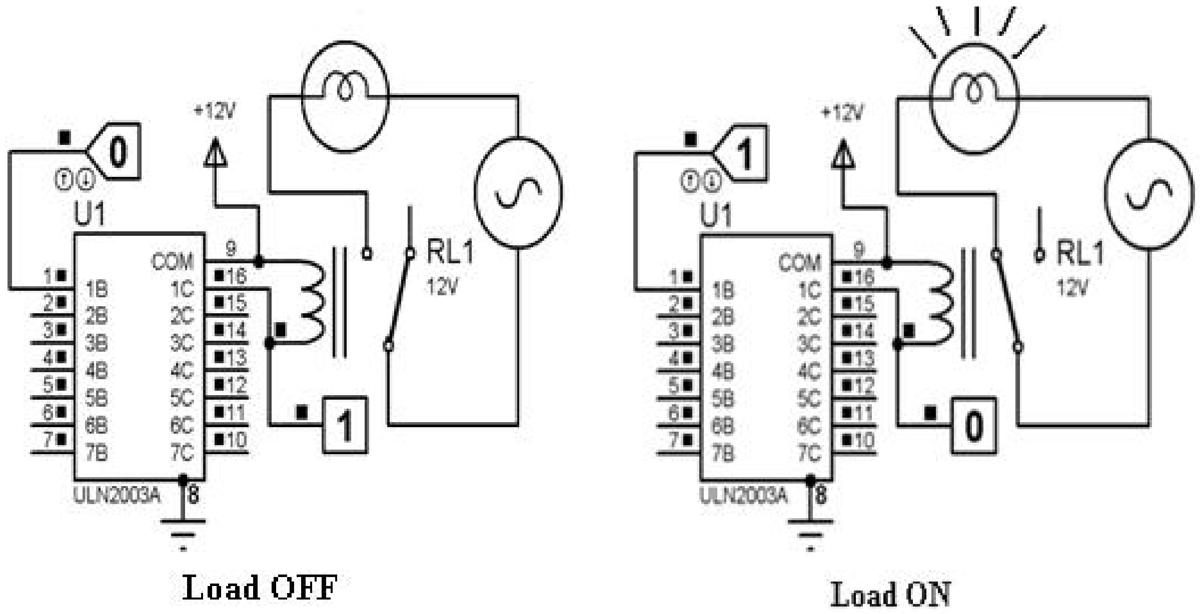


Figure 20: ULN2003A interfacing with microcontroller

The ULN2003A is a active high relay driver. 7 relays are controlled by this relay driver. Pin 1-7 are for controlling the relays 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.

4.5.2 DS18B20 interfacing with NODE MCU

The DS18B20 is a digital temperature sensor what makes it good to use even over long distances! These 1-wire digital temperature sensors are fairly precise (±0.5°C over much of the range) and can give up to 12 bits of precision from the onboard digital-to-analog converter. They work great with the Node MCU using a single digital pin, and you can even connect multiple ones to the same pin, each one has a unique 64-bit ID burned in at the factory to differentiate them.

The sensor works from 3.0 to 5.0V, what means that it can be powered directly from one of the 3.3V Node MCU pins.

The sensor has 3 wires:

- Black: GND
- Red: VCC
- Yellow: 1-Wire Data

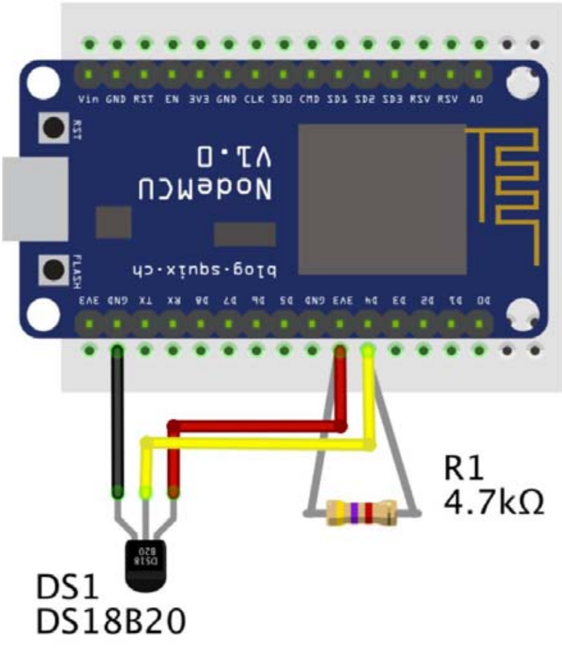


Figure 21: DS18B20 interfacing with Node MCU

4.5.3 OLED Graphic Display Interfacing with NodeMCU

OLED Display Module

The OLED module shown in the figure 22 is a very popular module available in the market. There are many variants of this module available in market, having different resolutions, communication protocol or pixel colors. They do not require backlight since the display creates its own light. Hence, they consume less power. Both I2C and SPI based OLED modules are available in market.

A Node MCU can communicate with this module using I2C communication protocol.

Pin Description

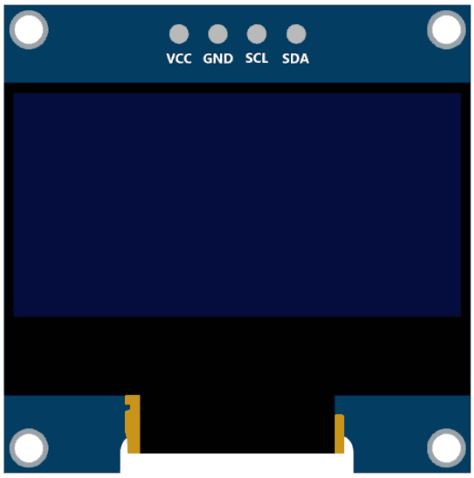


Figure 22: 128x64 I2C based OLED module.

VCC: This is the power pin for the module. A supply of 3.3V or 5V can be provided to this pin to power the display.

GND: This is the ground pin for the module.

SCL and SDA: These are the serial clock and serial data pins for I2C Interface.

Interfacing Diagram

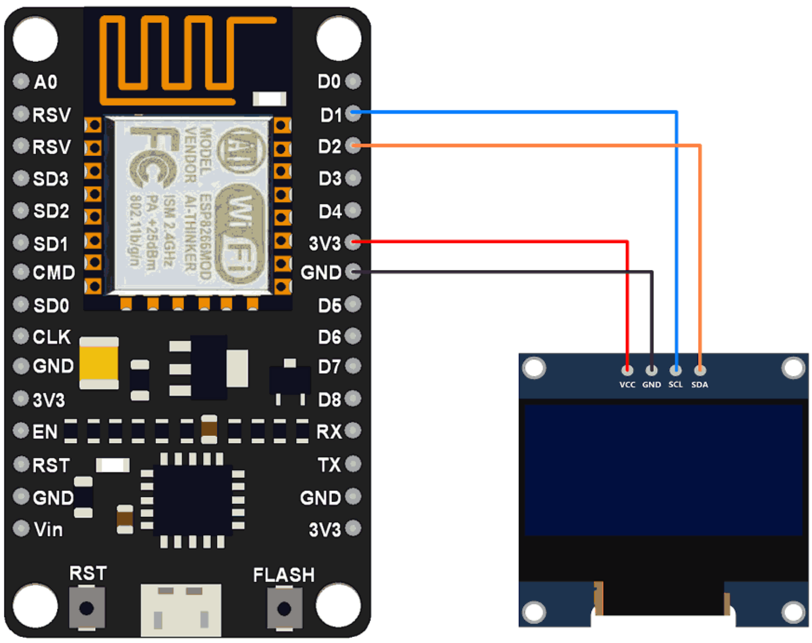


Figure 23: Interfacing OLED Display with NodeMCU

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

- Node MCU Section
- DS18B20 Section
- OLED Section
- Relay Section

The operation of the whole circuit is depending on every sections performance.

5.2 Flow Chart

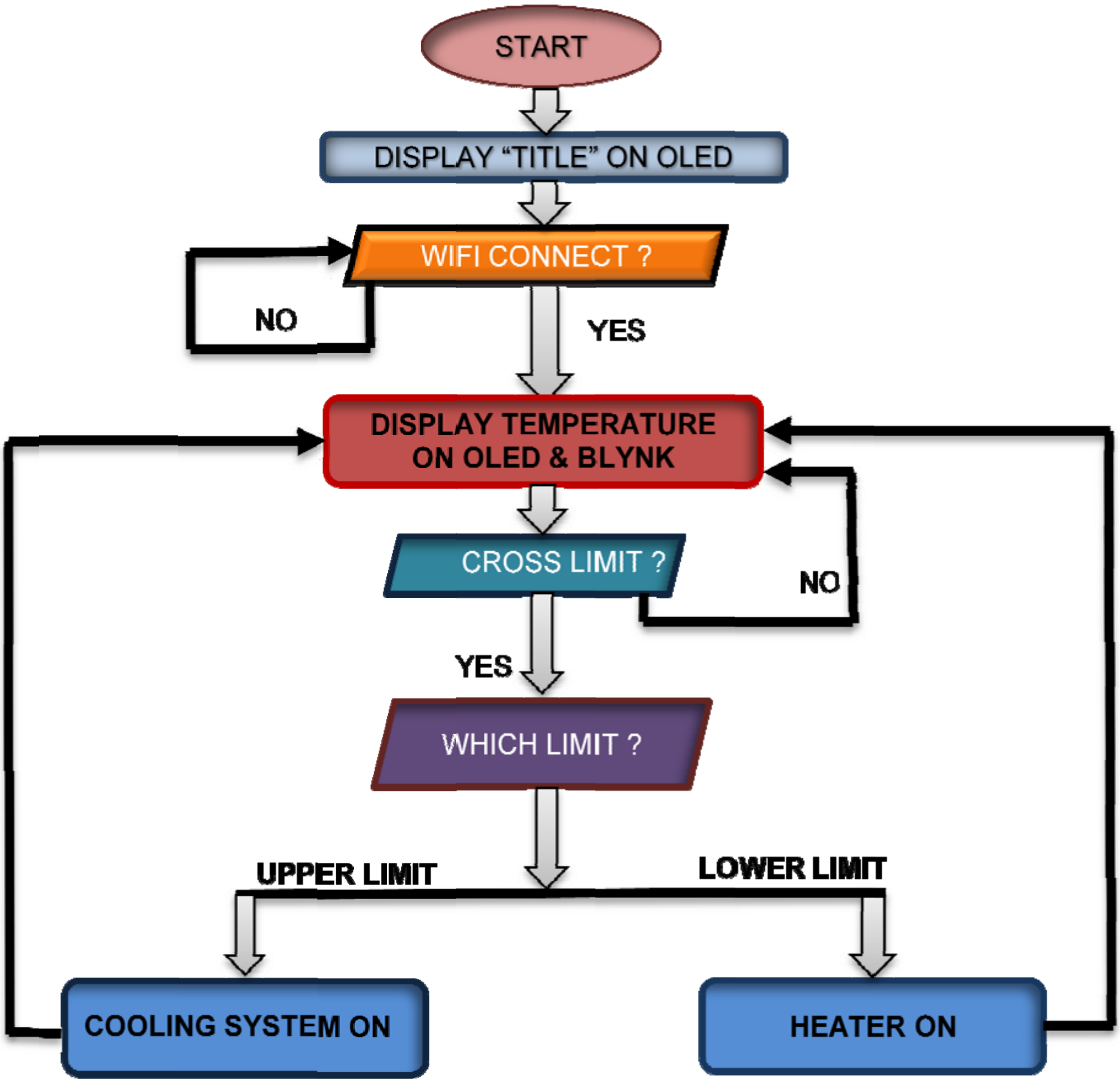


Figure 24: Flow chart of the program

5.3 Principle & Operations

NodeMCU is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The term "NodeMCU" by default refers to the firmware rather than the development kits. The firmware uses the Lua scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson, and spiffs.

5.3.1 Advantages of the NODE MCU

- ***Low cost :*** The Node MCU is less costlier than any other IOT based Devices. Because the wifi module which is used in it is of lowest cost.

- ***Hardware Part:*** It has Arduino Like hardware I/O. It is becoming very popular in these days that Arduino IDE has extended their software to work in the field of ESP 8266 Field module version.

- ***Network API:*** Node MCU has easily configurable network API.

- ***Integrated Wifi Module:*** ESP 8266 is incorporated in NODE MCU.It is an easily accessible wifi module.

5.3.2 Disadvantages

- The operation of the circuit depends on the working internet connection. If the working internet connection is not available then it will not run.
- It also depends on the free server provided by the third party, if the free server is not working then it will not run.
- NODE MCU has less resources of official documentation

5.4 Blynk app

Blynk was designed for the Internet of Things. It can control hardware remotely, it can display sensor data, it can store data, visualize it and do many other things.
There are three major components in the platform:

- **Blynk App** - allows to you create amazing interfaces for your projects using various widgets we provide.
- **Blynk Server** - responsible for all the communications between the smartphone and hardware. You can use our Blynk Cloud or run your private Blynk server locally. It's open-source, could easily handle thousands of devices and can even be launched on a Raspberry Pi.
- **Blynk Libraries** - for all the popular hardware platforms - enable communication with the server and process all the incoming and outgoing commands.

Now imagine: every time you press a Button in the Blynk app, the message travels to space the Blynk Cloud, where it magically finds its way to your hardware. It works the same in the opposite direction and everything happens in a blynk of an eye.

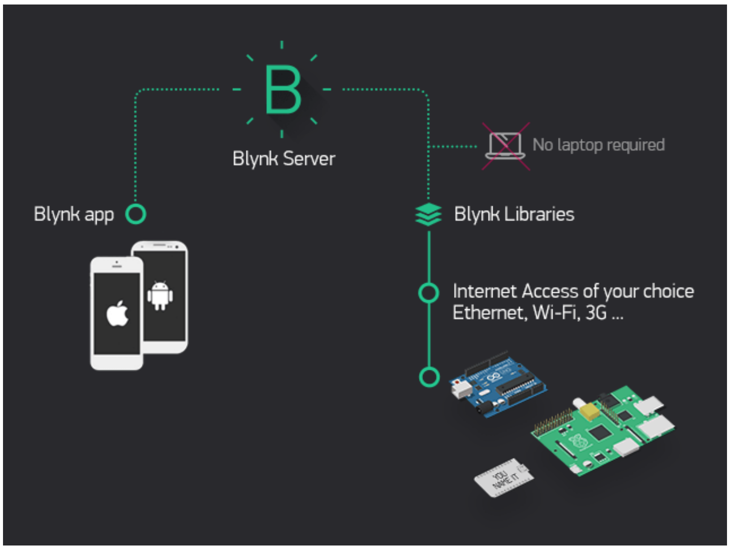


Figure 25: Blynk working principle

5.5 DS18B20 temperature sensors features

- Unique 1-Wire® Interface Requires Only One Port Pin for Communication
- Each Device has a Unique 64-Bit Serial Code Stored in an On-Board ROM
- Requires No External Components
- Can Be Powered from Data Line; Power Supply Range is 3.0V to 5.5V
- Measures Temperatures from -55°C to +125°C (-67°F to +257°F)
- 0.5°C Accuracy from -10°C to +85°C
- Thermometer Resolution is User Selectable from 9 to 12 Bits
- Converts Temperature to 12-Bit Digital Word in 750ms (Max)
- User-Definable Nonvolatile (NV) Alarm Settings

5.6 Cost estimation of the project

In this project we have used the cheapest IOT module NODE MCU. So the total cost of the project is reduced compare to the other IOT project. The total estimated cost of the complete project is listed in table 3.

Table 5: Costing of the projects

Sl. No.	Component	Qtn	Price (Rs)
1.	DS 18B20	2	300
2.	NODE MCU	1	350
3.	4.7 k Resistance	1	1
4.	ULN 2003 A IC	1	30

5.	Static Relay (5 volt)	4	100
6.	0.96” OLED	1	400
7.	General blank PCB (KS 100)	1	40
8.	3 mm LED	5	10
9.	Relay terminals	4	20
10.	Single strand wire	2m	20
11.	IC base (18 pin)	1	4
12.	Female PCB Header Connector	1	10
13.	SPDT switch	1	5
14.	Male PCB Header Connector	1	10
Total			**1300**

5.7 Photographs of the prototype

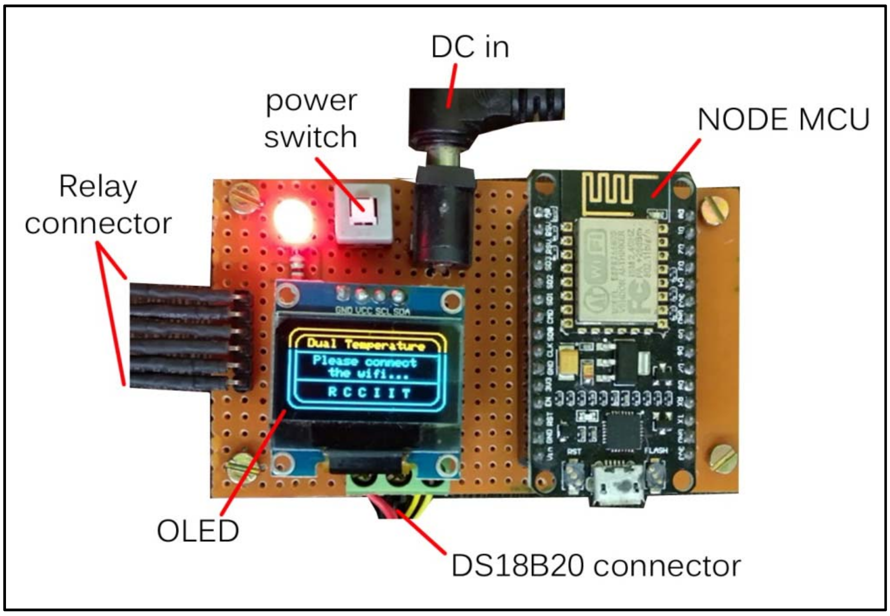


Figure 26: Main Controller Board

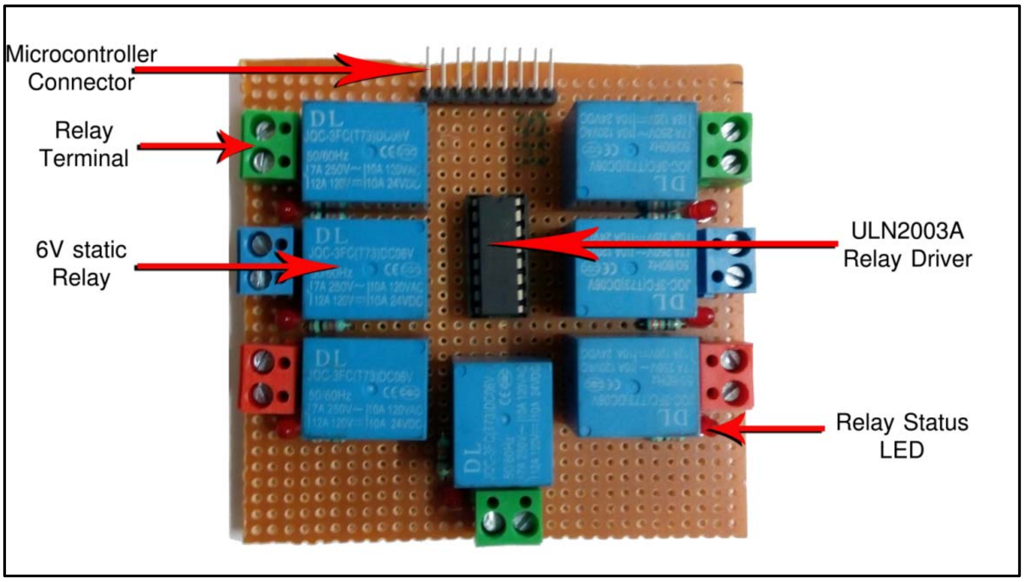


Figure 27: The Relay Board

Figure 28: The Blynk app user interface

Chapter 6

(Conclusion & Future Scope)

6.1 Conclusion

Here we developed a circuit which will regulate and monitor the temperature of a room using IOT. It also limits the problem due to lack of proper temperature monitoring system in industries and home. It consists mainly following parts wifi device, NodeMcu, OLED, temperature sensor (DS18B20). First it needs to be check whether our module is connected to wifi or not. If it is connected, it will directly show the temperature on OLED as well as on mobile. It continuously monitors the temperature of a room. Whenever the temperature crosses the predefined set limits heater or cooler will be on according to the condition.

6.2 Result

The experimental model was made according to the circuit diagram and the results were as expected. The blink app and the OLED show the temperature of the room as soon as it is connected to the wifi. After proper monitoring it switch on cooler or heater according to the situation.

6.3 Future work

Right now we are using Blink App platform to operate our module, in future will try to develop own app platform for our module, because in case this blink app stop to work in future we may face problem.

Chapter 7

(References)

1. Deepraj Duttachowdhury, Vivek Patil, Arya Parab, Raj Patel,Karuna Nikum **“Transformer Monitoring and Control Using Iot”,** IOSR Journal of Engineering (IOSRJEN), Volume 10, Pg. 40-43

2. K.Kishore Raju, G.P Saradhi Varma, A Akhil Varma, **“Domestic Room Temperature Monitoring and,Regulation Using IoT and Artificial,Intelligence (AI)”,** International Journal of Innovative Research in Computer and Communication Engineering, Volume 6

3. PRACHI H. KULKARNI, PRATIK D. KUTE, **“INTERNET OF THINGS BASED SYSTEM FOR REMOTE MONITORING OF WEATHER PARAMETERS AND APPLICATIONS”,** International Journal of Advances in Electronics and Computer Science, Volume-3, Issue-2, Feb.-2016

4. Akash,Amit Birwal, **“IoT-based Temperature and Humidity Monitoring System for Agriculture”,** International Journal of Innovative Research in Science, Engineering and Technology, Vol. 6, Issue 7, July 2017

5. M.Lavanya, P. Muthukannan, Y.S.S. Bhargav, V. Suresh, “**Iot Based Automated Temperature and Humidity Monitoring and Control”,**Journal of Chemical and Pharmaceutical Sciences

6. Archana Ghumare,Tejaswini Kochar, Pooja Jain, Sneha Ahirrao,Prof.R.R.Bhandari, **“Smart Server IOT Based Temperature Monitoring System”,** International Journal of Modern Trends in Engineering and Research, e-ISSN No.:2349-9745

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.

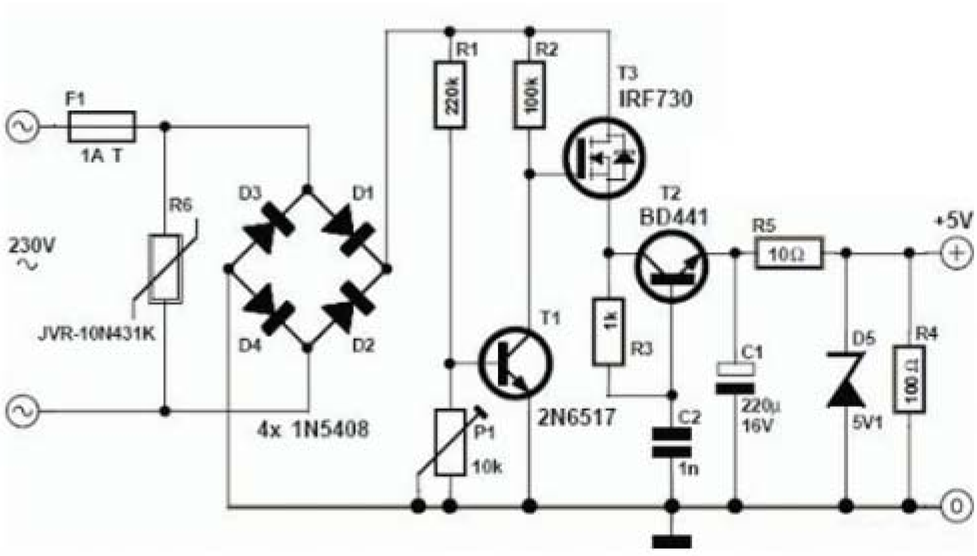


Figure 29: Transformer less SMPS 5 volt power supply

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.

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.

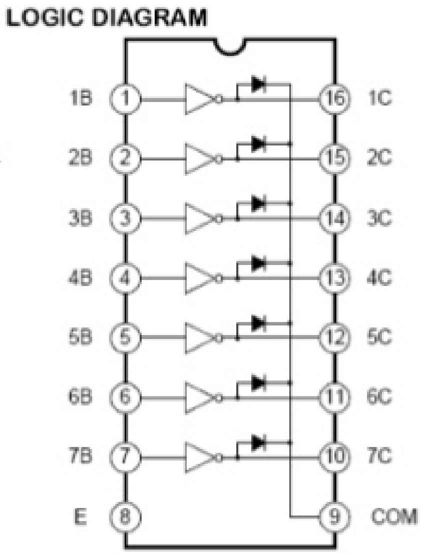


Figure 30: ULN2003A Internal Block Diagram

Resistor

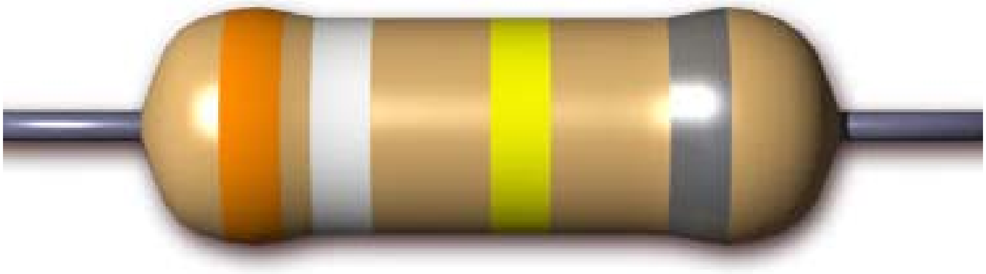


Figure 31: Resistor

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

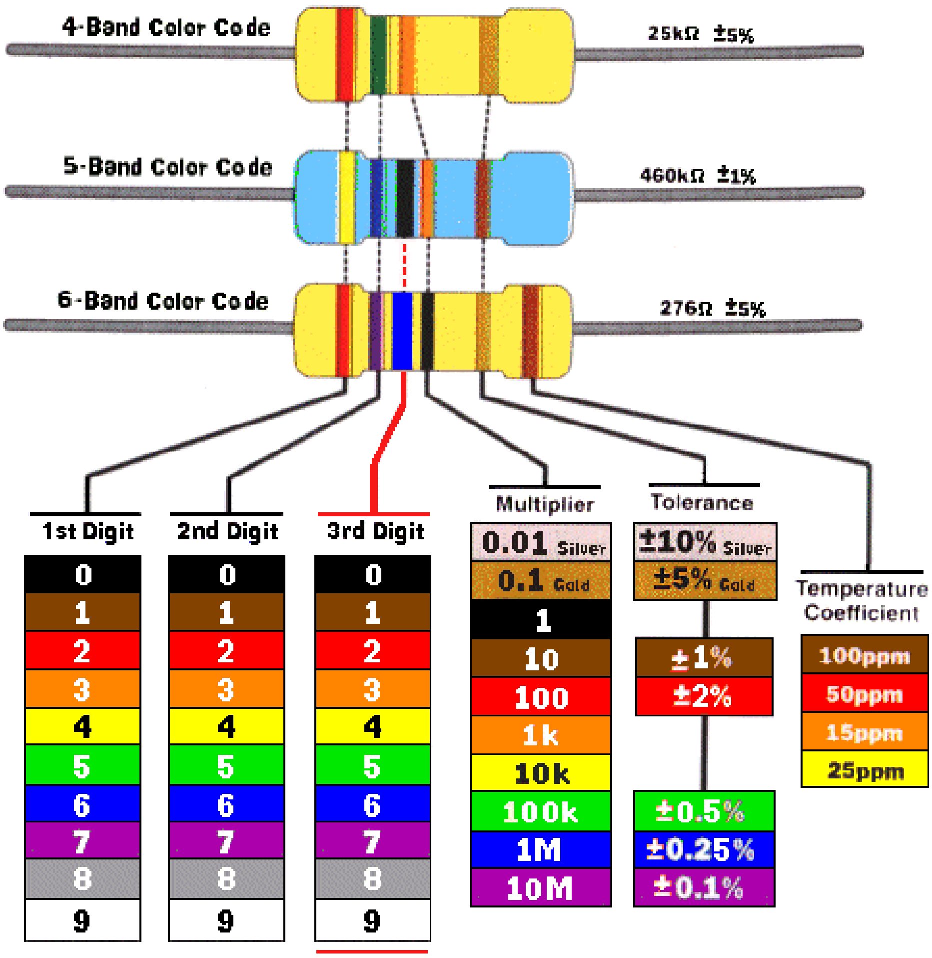


Figure 32: Color Code for resistance

RELAY

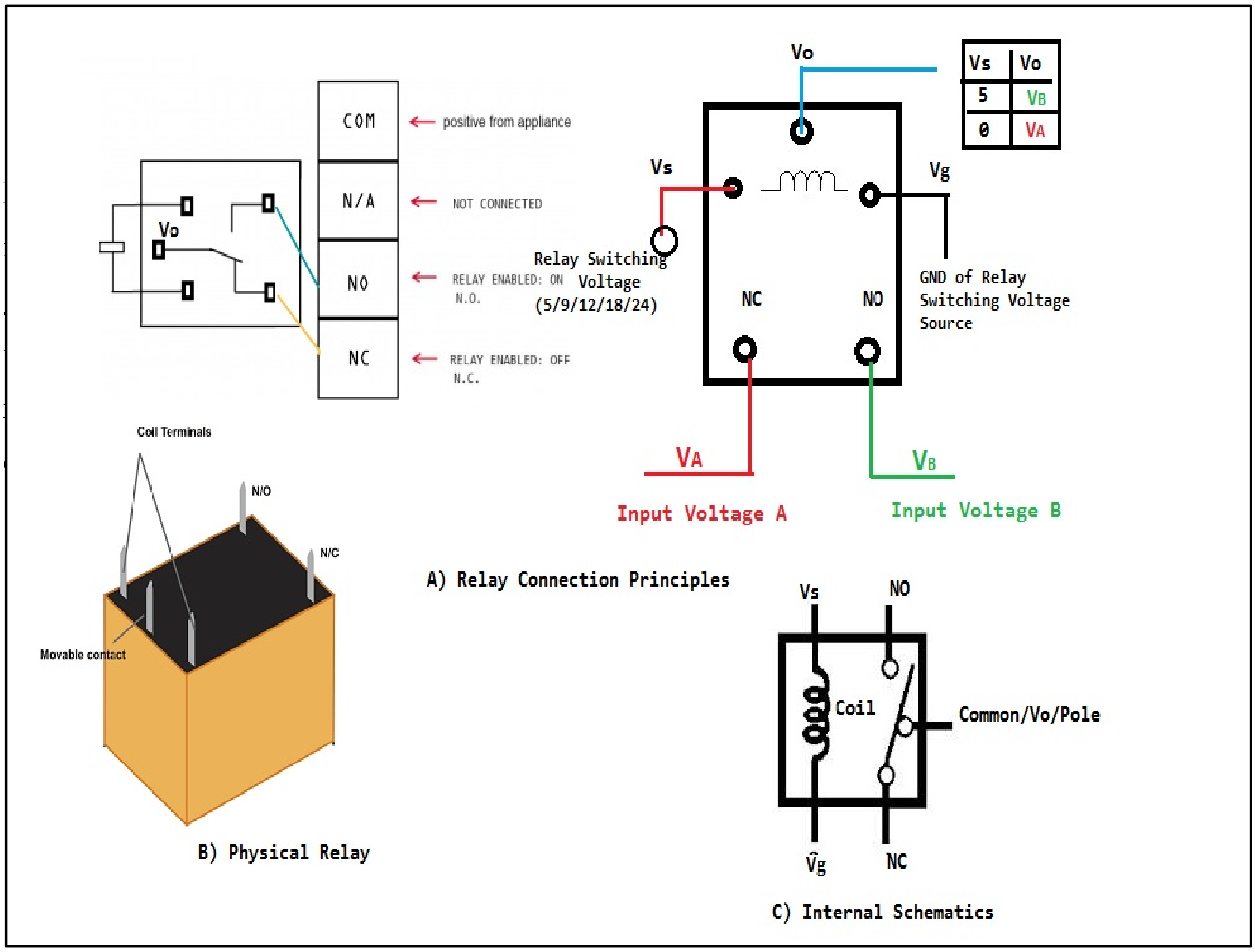


Figure 33: 6 volt Cube Relay

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.

OLED

An organic light-emitting diode (OLED) is a light-emitting diode (LED) in which the emissive electroluminescent layer is a film of organic compound that emits light in response to an electric current. This organic layer is situated between two electrodes; typically, at least one of these electrodes is transparent. OLEDs are used to create digital displays in devices such as television screens, computer monitors, portable systems such as smart phones, handheld game consoles and PDAs. A major area of research is the development of white OLED devices for use in solid-state lighting applications.

Figure 34: 128 X 64 OLED Module

NodeMCU

NodeMCU is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The term "NodeMCU" by default refers to the firmware rather than the development kits. The firmware uses the Lua scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson and SPIFFS.

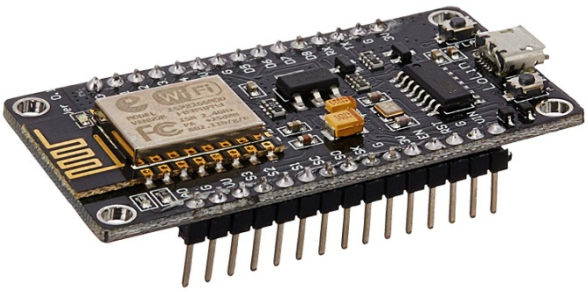


Figure 35: Node MCUModule

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 36: 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.

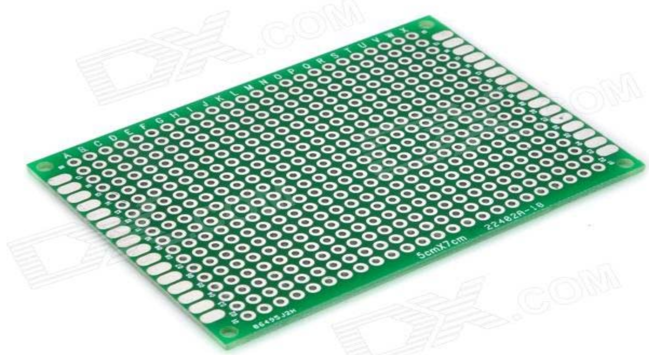


Figure 37: Blank glass epoxy PCB Board

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.

Appendix B

(Software coding)

PROGRAM CODE:

```
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
char auth[] = "d971cdeea74248279efc51d9bd98aa26";

/* WiFi credentials */
char ssid[] = "Budhaditya";
char pass[] = "buddy1234";

/* TIMER */
#include <SimpleTimer.h>
SimpleTimer timer;

#define OLED_RESET D5  //14
Adafruit_SSD1306 display(OLED_RESET);

#if (SSD1306_LCDHEIGHT != 64)
#error("Height incorrect, please fix Adafruit_SSD1306.h!");
#endif
/* DS18B20 Temperature Sensor */
#include <OneWire.h>
#include<DallasTemperature.h>
#define ONE_WIRE_BUS 0 // DS18B20 on arduino pin2 corresponds to D3 on physical board
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature DS18B20(&oneWire);
float temp_0;
float temp_1;

void setup()
{
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  display.clearDisplay();
  display.setCursor(0,0);
  display.drawRoundRect(0, 0, 128, 64, 8, WHITE);
  display.drawRoundRect(5, 5, 118, 54, 8, WHITE);
// Sets the color to black with a white background
  display.setTextColor(WHITE);
  display.setCursor(15,8);
  display.println("Dual Temperature");
  display.drawLine(6,17,120,17, WHITE);
```

```
  display.setCursor(20,20);
  display.println("Please connect");
  display.setCursor(32,30);
  display.println("the wifi...");
  display.drawLine(6,40,120,40, WHITE);
  display.setCursor(33,46);
  display.println("R C C I I T");
  display.display();

  Serial.begin(115200);
  Serial.println("Connecting to wifi.......");
  Blynk.begin(auth, ssid, pass);
  DS18B20.begin();
  timer.setInterval(1000L, getSendData);
  Serial.println(" ");
  Serial.println("Testing Dual Sensor data");
}

void loop()
{
  timer.run(); // Initiates SimpleTimer
  Blynk.run();

  display.clearDisplay();
  display.setCursor(0,0);
  display.drawRoundRect(0, 0, 128, 64, 8, WHITE);
  display.drawRoundRect(5, 5, 118, 54, 8, WHITE);

// Sets the color to black with a white background
  display.setTextColor(WHITE);
  display.setCursor(15,8);
  display.println("Dual Temperature");
  display.drawLine(6,17,120,17, WHITE);
  display.setCursor(10,20);
  display.println("Temp 1 = ");
  display.setCursor(10,30);
  display.println("Temp 2 = ");
  display.setCursor(65,20);
  display.println(temp_0, 4);
  display.setCursor(110,20);
  display.print((char)247); // degree symbol
  display.println("C");
  display.setCursor(65,30);
  display.println(temp_1, 4);
  display.setCursor(110,30);
  display.print((char)247); // degree symbol
  display.println("C");
```

```
  display.drawLine(6,40,120,40, WHITE);
  display.setCursor(33,46);
  display.println("R C C I I T");
  display.display();
}
/***************************************************
 * Send Sensor data to Blynk
 **************************************************/
void getSendData()
{
  DS18B20.requestTemperatures();
  temp_0 = DS18B20.getTempCByIndex(0); // Sensor 0 will capture Temp in Celcius
  temp_1 = DS18B20.getTempCByIndex(1); // Sensor 0 will capture Temp in Celcius

  Serial.print("Temp_0: ");
  Serial.print(temp_0, 4);
  Serial.print(" oC . Temp_1: ");
  Serial.print(temp_1, 4);
  Serial.println(" oC");

  Blynk.virtualWrite(10, temp_0); //virtual pin V10
  Blynk.virtualWrite(11, temp_1); //virtual pin V11
}
```

Appendix C

(Data sheets)

DS18B20

Programmable Resolution 1-Wire Digital Thermometer

General Description

The DS18B20 digital thermometer provides 9-bit to 12-bit Celsius temperature measurements and has an alarm function with nonvolatile user-programmable upper and lower trigger points. The DS18B20 communicates over a 1-Wire bus that by definition requires only one data line (and ground) for communication with a central microprocessor. In addition, the DS18B20 can derive power directly from the data line ("parasite power"), eliminating the need for an external power supply.

Each DS18B20 has a unique 64-bit serial code, which allows multiple DS18B20s to function on the same 1-Wire bus. Thus, it is simple to use one microprocessor to control many DS18B20s distributed over a large area. Applications that can benefit from this feature include HVAC environmental controls, temperature monitoring systems inside buildings, equipment, or machinery, and process monitoring and control systems.

Applications

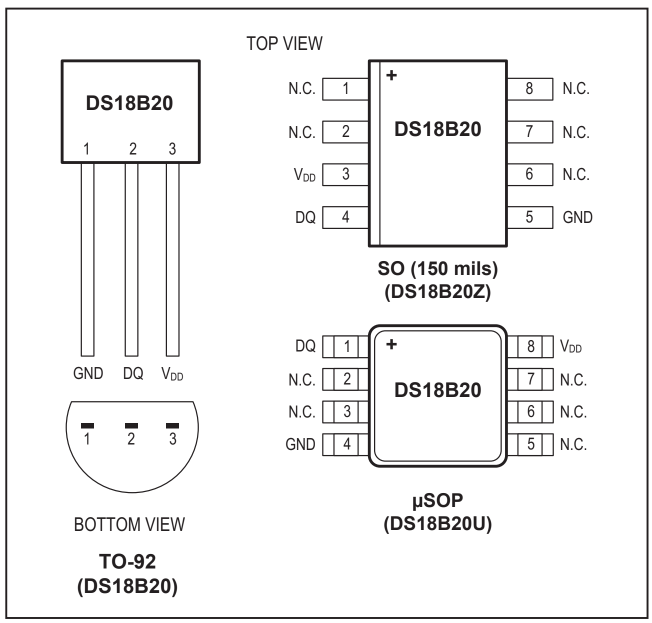
- Thermostatic Controls
- Industrial Systems
- Consumer Products
- Thermometers
- Thermally Sensitive Systems

***Ordering Information** appears at end of data sheet.*

Benefits and Features

- Unique 1-Wire® Interface Requires Only One Port Pin for Communication
- Reduce Component Count with Integrated Temperature Sensor and EEPROM
  - Measures Temperatures from -55°C to +125°C (-67°F to +257°F)
  - ±0.5°C Accuracy from -10°C to +85°C
  - Programmable Resolution from 9 Bits to 12 Bits
  - No External Components Required
- Parasitic Power Mode Requires Only 2 Pins for Operation (DQ and GND)
- Simplifies Distributed Temperature-Sensing Applications with Multidrop Capability
  - Each Device Has a Unique 64-Bit Serial Code Stored in On-Board ROM
- Flexible User-Definable Nonvolatile (NV) Alarm Settings with Alarm Search Command Identifies Devices with Temperatures Outside Programmed Limits
- Available in 8-Pin SO (150 mils), 8-Pin µSOP, and 3-Pin TO-92 Packages

Pin Configurations



Absolute Maximum Ratings

Voltage Range on Any Pin Relative to Ground....-0.5V to +6.0V
Operating Temperature Range......................... -55°C to +125°C
Storage Temperature Range........................... -55°C to +125°C
Solder Temperature..............................Refer to the IPC/JEDEC J-STD-020 Specification.

These are stress ratings only and functional operation of the device at these or any other conditions above those indicated in the operation sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods of time may affect reliability.

DC Electrical Characteristics

(-55°C to +125°C; V_{DD} = 3.0V to 5.5V)

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS
Supply Voltage	V_{DD}	Local power (Note 1)		+3.0		+5.5	V
Pullup Supply Voltage	V_{PU}	Parasite power	(Notes 1, 2)	+3.0		+5.5	V
		Local power		+3.0		V_{DD}	
Thermometer Error	t_{ERR}	-10°C to +85°C	(Note 3)			±0.5	°C
		-30°C to +100°C				±1	
		-55°C to +125°C				±2	
Input Logic-Low	V_{IL}	(Notes 1, 4, 5)		-0.3		+0.8	V
Input Logic-High	V_{IH}	Local power	(Notes 1,6)	+2.2		The lower of 5.5 or V_{DD} + 0.3	V
		Parasite power		+3.0			
Sink Current	I_L	$V_{I/O}$ = 0.4V		4.0			mA
Standby Current	I_{DDS}	(Notes 7, 8)			750	1000	nA
Active Current	I_{DD}	V_{DD} = 5V (Note 9)			1	1.5	mA
DQ Input Current	I_{DQ}	(Note 10)			5		µA
Drift		(Note 11)			±0.2		°C

Note 1: All voltages are referenced to ground.
Note 2: The Pullup Supply Voltage specification assumes that the pullup device is ideal, and therefore the high level of the pullup is equal to V_{PU}. In order to meet the V_{IH} spec of the DS18B20, the actual supply rail for the strong pullup transistor must include margin for the voltage drop across the transistor when it is turned on; thus: $V_{PU\_ACTUAL} = V_{PU\_IDEAL} + V_{TRANSISTOR}$.
Note 3: See typical performance curve in Figure 1. Thermometer Error limits are 3-sigma values.
Note 4: Logic-low voltages are specified at a sink current of 4mA.
Note 5: To guarantee a presence pulse under low voltage parasite power conditions, V_{ILMAX} may have to be reduced to as low as 0.5V.
Note 6: Logic-high voltages are specified at a source current of 1mA.
Note 7: Standby current specified up to +70°C. Standby current typically is 3µA at +125°C.
Note 8: To minimize I_{DDS}, DQ should be within the following ranges: GND ≤ DQ ≤ GND + 0.3V or V_{DD} – 0.3V ≤ DQ ≤ V_{DD}.
Note 9: Active current refers to supply current during active temperature conversions or EEPROM writes.
Note 10: DQ line is high (“high-Z” state).
Note 11: Drift data is based on a 1000-hour stress test at +125°C with V_{DD} = 5.5V.

AC Electrical Characteristics–NV Memory

(-55°C to +125°C; V_{DD} = 3.0V to 5.5V)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
NV Write Cycle Time	t_{WR}			2	10	ms
EEPROM Writes	N_{EEWR}	-55°C to +55°C	50k			writes
EEPROM Data Retention	t_{EEDR}	-55°C to +55°C	10			years

AC Electrical Characteristics

(-55°C to +125°C; V_{DD} = 3.0V to 5.5V)

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS
Temperature Conversion Time	t_{CONV}	9-bit resolution	(Note 12)			93.75	ms
		10-bit resolution				187.5	
		11-bit resolution				375	
		12-bit resolution				750	
Time to Strong Pullup On	t_{SPON}	Start convert T command issued				10	µs
Time Slot	t_{SLOT}	(Note 12)		60		120	µs
Recovery Time	t_{REC}	(Note 12)		1			µs
Write 0 Low Time	t_{LOW0}	(Note 12)		60		120	µs
Write 1 Low Time	t_{LOW1}	(Note 12)		1		15	µs
Read Data Valid	t_{RDV}	(Note 12)				15	µs
Reset Time High	t_{RSTH}	(Note 12)		480			µs
Reset Time Low	t_{RSTL}	(Notes 12, 13)		480			µs
Presence-Detect High	t_{PDHIGH}	(Note 12)		15		60	µs
Presence-Detect Low	t_{PDLOW}	(Note 12)		60		240	µs
Capacitance	$C_{IN/OUT}$					25	pF

Note 12: See the timing diagrams in Figure 2.
Note 13: Under parasite power, if t_{RSTL} > 960µs, a power-on reset can occur.

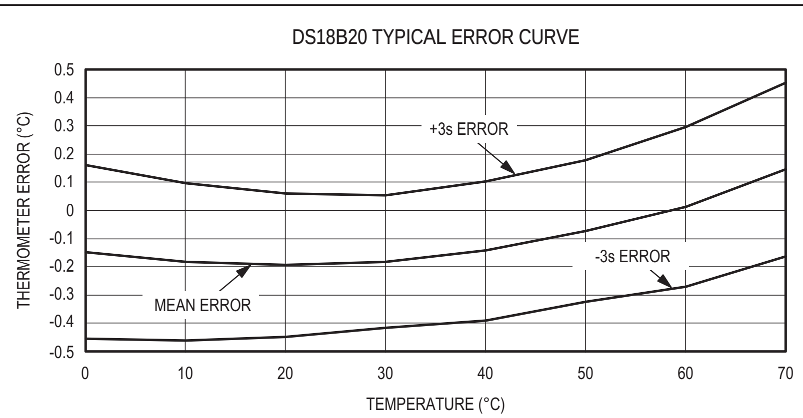


Figure 1. Typical Performance Curve

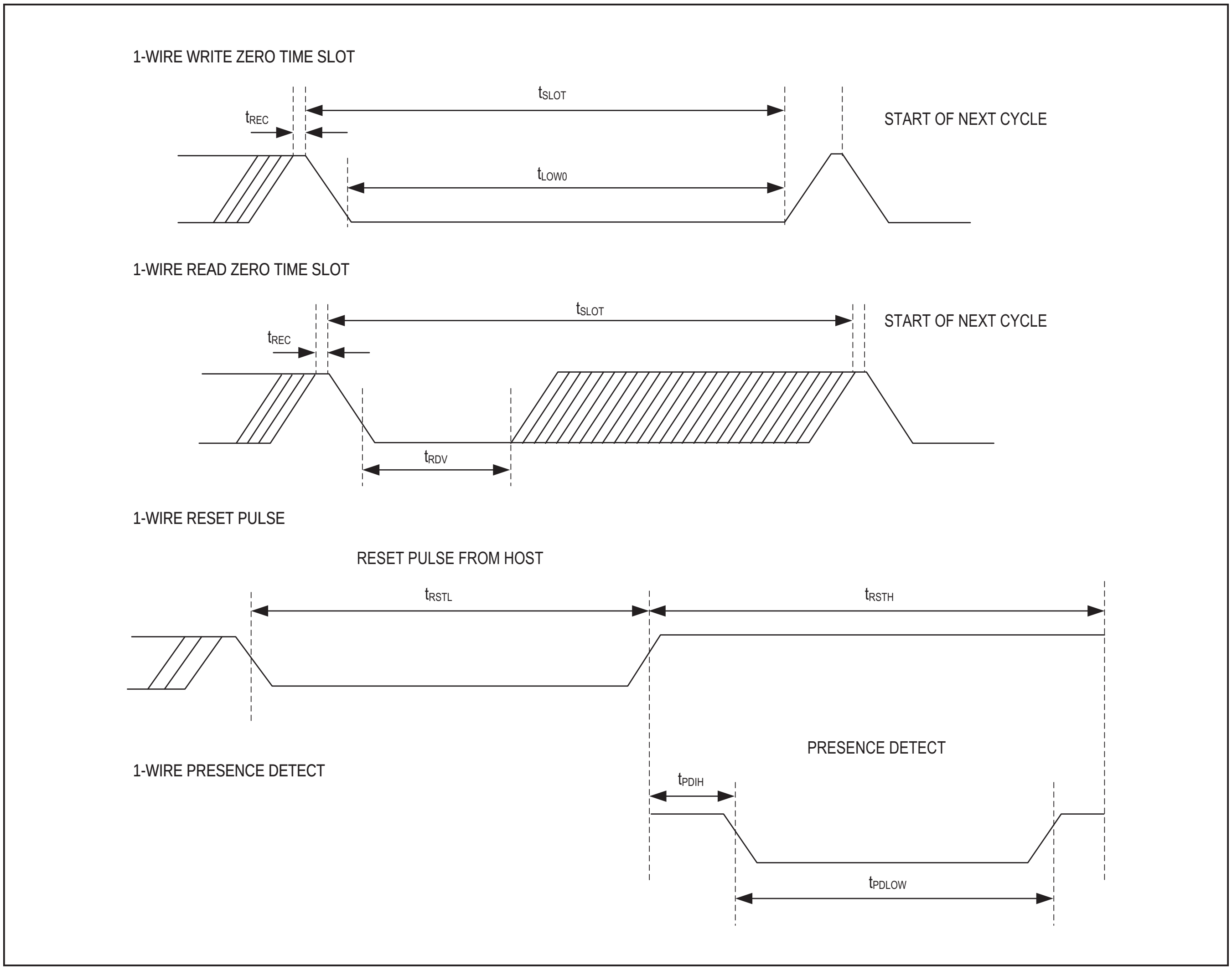


Figure 2. Timing Diagrams

Pin Description

PIN			NAME	FUNCTION
SO	µSOP	TO-92		
1, 2, 6, 7, 8	2, 3, 5, 6, 7	—	N.C.	No Connection
3	8	3	V_{DD}	Optional V_{DD}. V_{DD} must be grounded for operation in parasite power mode.
4	1	2	DQ	Data Input/Output. Open-drain 1-Wire interface pin. Also provides power to the device when used in parasite power mode (see the *Powering the DS18B20* section.)
5	4	1	GND	Ground

Overview

Figure 3 shows a block diagram of the DS18B20, and pin descriptions are given in the *Pin Description* table. The 64-bit ROM stores the device's unique serial code. The scratchpad memory contains the 2-byte temperature register that stores the digital output from the temperature sensor. In addition, the scratchpad provides access to the 1-byte upper and lower alarm trigger registers (T_H and T_L) and the 1-byte configuration register. The configuration register allows the user to set the resolution of the temperature-to-digital conversion to 9, 10, 11, or 12 bits. The T_H, T_L, and configuration registers are nonvolatile (EEPROM), so they will retain data when the device is powered down.

The DS18B20 uses Maxim's exclusive 1-Wire bus protocol that implements bus communication using one control signal. The control line requires a weak pullup resistor since all devices are linked to the bus via a 3-state or open-drain port (the DQ pin in the case of the DS18B20). In this bus system, the microprocessor (the master device) identifies and addresses devices on the bus using each device's unique 64-bit code. Because each device has a unique code, the number of devices that can be addressed on one bus is virtually unlimited. The 1-Wire bus protocol, including detailed explanations of the commands and "time slots," is covered in the *1-Wire Bus System* section.

Another feature of the DS18B20 is the ability to operate without an external power supply. Power is instead supplied through the 1-Wire pullup resistor through the DQ pin when the bus is high. The high bus signal also charges an internal capacitor (C_{PP}), which then supplies power to the device when the bus is low. This method of deriving power from the 1-Wire bus is referred to as "parasite power." As an alternative, the DS18B20 may also be powered by an external supply on V_{DD}.

Operation—Measuring Temperature

The core functionality of the DS18B20 is its direct-to-digital temperature sensor. The resolution of the temperature sensor is user-configurable to 9, 10, 11, or 12 bits, corresponding to increments of 0.5°C, 0.25°C, 0.125°C, and 0.0625°C, respectively. The default resolution at power-up is 12-bit. The DS18B20 powers up in a low-power idle state. To initiate a temperature measurement and A-to-D conversion, the master must issue a Convert T [44h] command. Following the conversion, the resulting thermal data is stored in the 2-byte temperature register in the scratchpad memory and the DS18B20 returns to its idle state. If the DS18B20 is powered by an external supply, the master can issue "read time slots" (see the *1-Wire Bus System* section) after the Convert T command and the DS18B20 will respond by transmitting 0 while the temperature conversion is in progress and 1 when the conversion is done. If the DS18B20 is powered with parasite power, this notification technique cannot be used since the bus must be pulled high by a strong pullup during the entire temperature conversion. The bus requirements for parasite power are explained in detail in the *Powering the DS18B20* section.

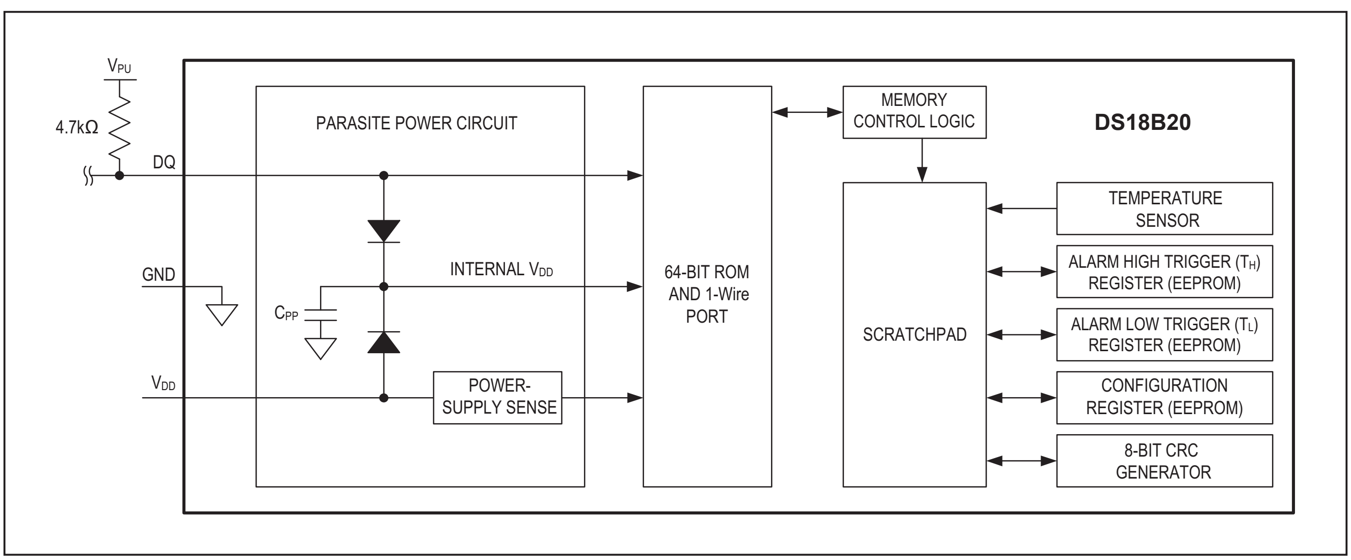


Figure 3. DS18B20 Block Diagram

The DS18B20 output temperature data is calibrated in degrees Celsius; for Fahrenheit applications, a lookup table or conversion routine must be used. The temperature data is stored as a 16-bit sign-extended two's complement number in the temperature register (see Figure 4). The sign bits (S) indicate if the temperature is positive or negative: for positive numbers S = 0 and for negative numbers S = 1. If the DS18B20 is configured for 12-bit resolution, all bits in the temperature register will contain valid data. For 11-bit resolution, bit 0 is undefined. For 10-bit resolution, bits 1 and 0 are undefined, and for 9-bit resolution bits 2, 1, and 0 are undefined. Table 1 gives examples of digital output data and the corresponding temperature reading for 12-bit resolution conversions.

Operation—Alarm Signaling

After the DS18B20 performs a temperature conversion, the temperature value is compared to the user-defined two's complement alarm trigger values stored in the 1-byte T_H and T_L registers (see Figure 5). The sign bit (S) indicates if the value is positive or negative: for positive numbers S = 0 and for negative numbers S = 1. The T_H and T_L registers are nonvolatile (EEPROM) so they will retain data when the device is powered down. T_H and T_L can be accessed through bytes 2 and 3 of the scratchpad as explained in the *Memory* section.

Only bits 11 through 4 of the temperature register are used in the T_H and T_L comparison since T_H and T_L are 8-bit registers. If the measured temperature is lower than

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LS BYTE	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}
	BIT 15	**BIT 14**	**BIT 13**	**BIT 12**	**BIT 11**	**BIT 10**	**BIT 9**	**BIT 8**
MS BYTE	S	S	S	S	S	2^6	2^5	2^4

S = SIGN

Figure 4. Temperature Register Format

Table 1. Temperature/Data Relationship

TEMPERATURE (°C)	DIGITAL OUTPUT (BINARY)	DIGITAL OUTPUT (HEX)
+125	0000 0111 1101 0000	07D0h
+85*	0000 0101 0101 0000	0550h
+25.0625	0000 0001 1001 0001	0191h
+10.125	0000 0000 1010 0010	00A2h
+0.5	0000 0000 0000 1000	0008h
0	0000 0000 0000 0000	0000h
-0.5	1111 1111 1111 1000	FFF8h
-10.125	1111 1111 0101 1110	FF5Eh
-25.0625	1111 1110 0110 1111	FE6Fh
-55	1111 1100 1001 0000	FC90h

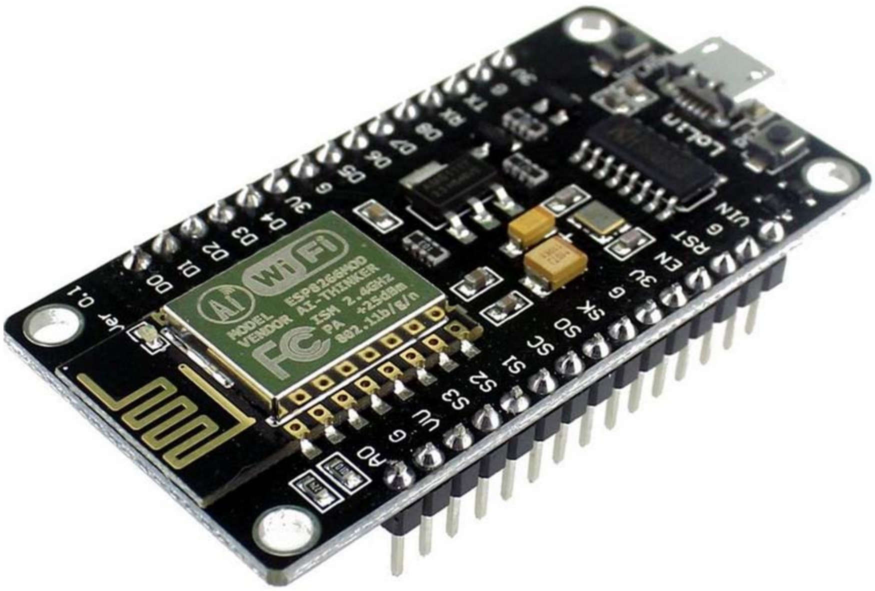
**The power-on reset value of the temperature register is +85°C.*

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
S	2^6	2^5	2^4	2^3	2^2	2^1	2^0

Figure 5. T_H and T_L Register Format

User Manual V1.2

ESP8266 NodeMCU WiFi Devkit



The ESP8266 is the name of a micro controller designed by Espressif Systems. The ESP8266 itself is a self-contained WiFi networking solution offering as a bridge from existing micro controller to WiFi and is also capable of running self-contained applications.

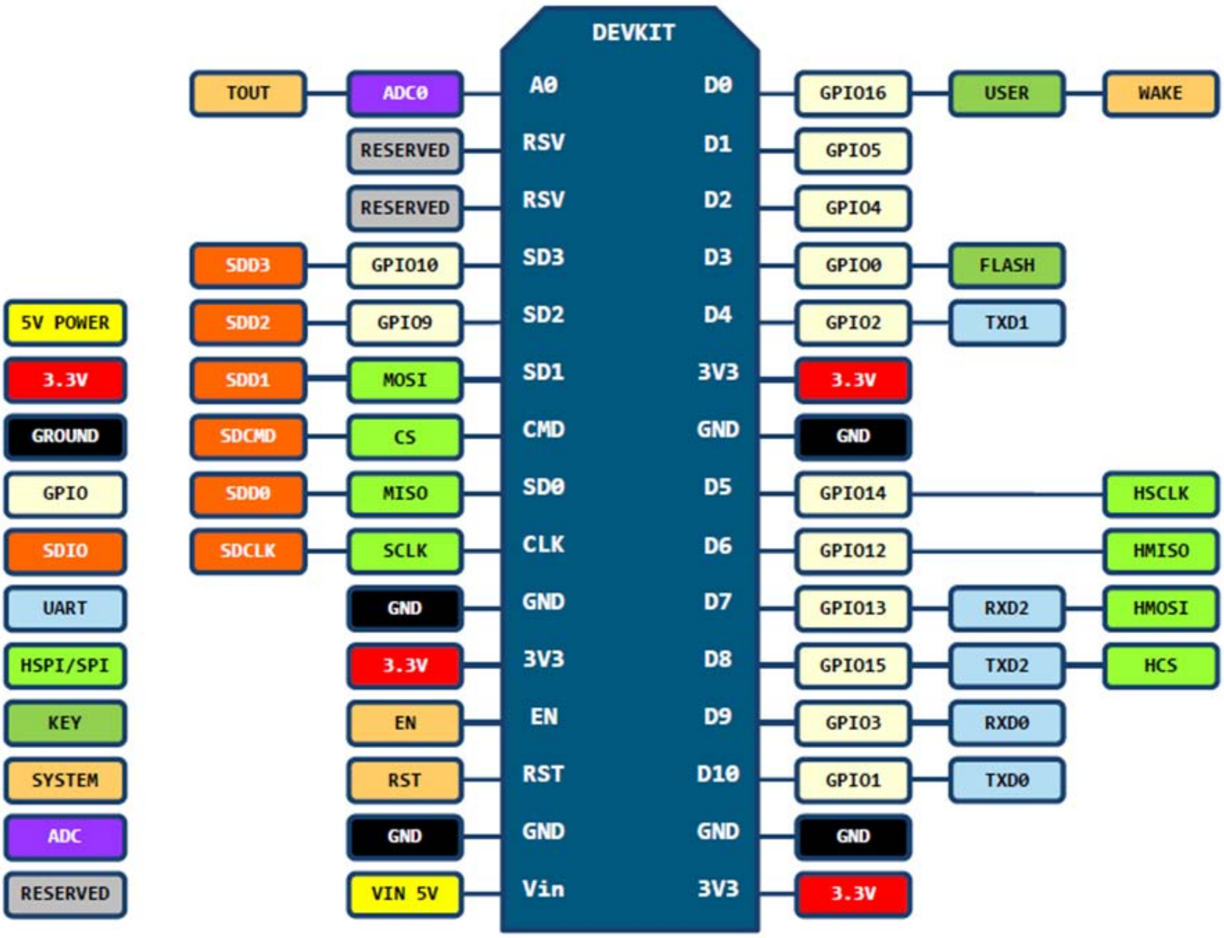
This module comes with a built in USB connector and a rich assortment of pin-outs. With a micro USB cable, you can connect NodeMCU devkit to your laptop and flash it without any trouble, just like Arduino. It is also immediately breadboard friendly.

Table of Contents

1. Specification:

- Voltage:3.3V.
- Wi-Fi Direct (P2P), soft-AP.
- Current consumption: 10uA~170mA.
- Flash memory attachable: 16MB max (512K normal).
- Integrated TCP/IP protocol stack.
- Processor: Tensilica L106 32-bit.
- Processor speed: 80~160MHz.
- RAM: 32K + 80K.
- GPIOs: 17 (multiplexed with other functions).
- Analog to Digital: 1 input with 1024 step resolution.
- +19.5dBm output power in 802.11b mode
- 802.11 support: b/g/n.
- Maximum concurrent TCP connections: 5.

2. Pin Definition:



D0(GPIO16) can only be used as gpio read/write, no interrupt supported, no pwm/i2c/ow supported.

3. Using Arduino IDE

The most basic way to use the ESP8266 module is to use serial commands, as the chip is basically a WiFi/Serial transceiver. However, this is not convenient. What we recommend is using the very cool Arduino ESP8266 project, which is a modified version of the Arduino IDE that you need to install on your computer. This makes it very convenient to use the ESP8266 chip as we will be using the well-known Arduino IDE. Following the below step to install ESP8266 library to work in Arduino IDE environment.

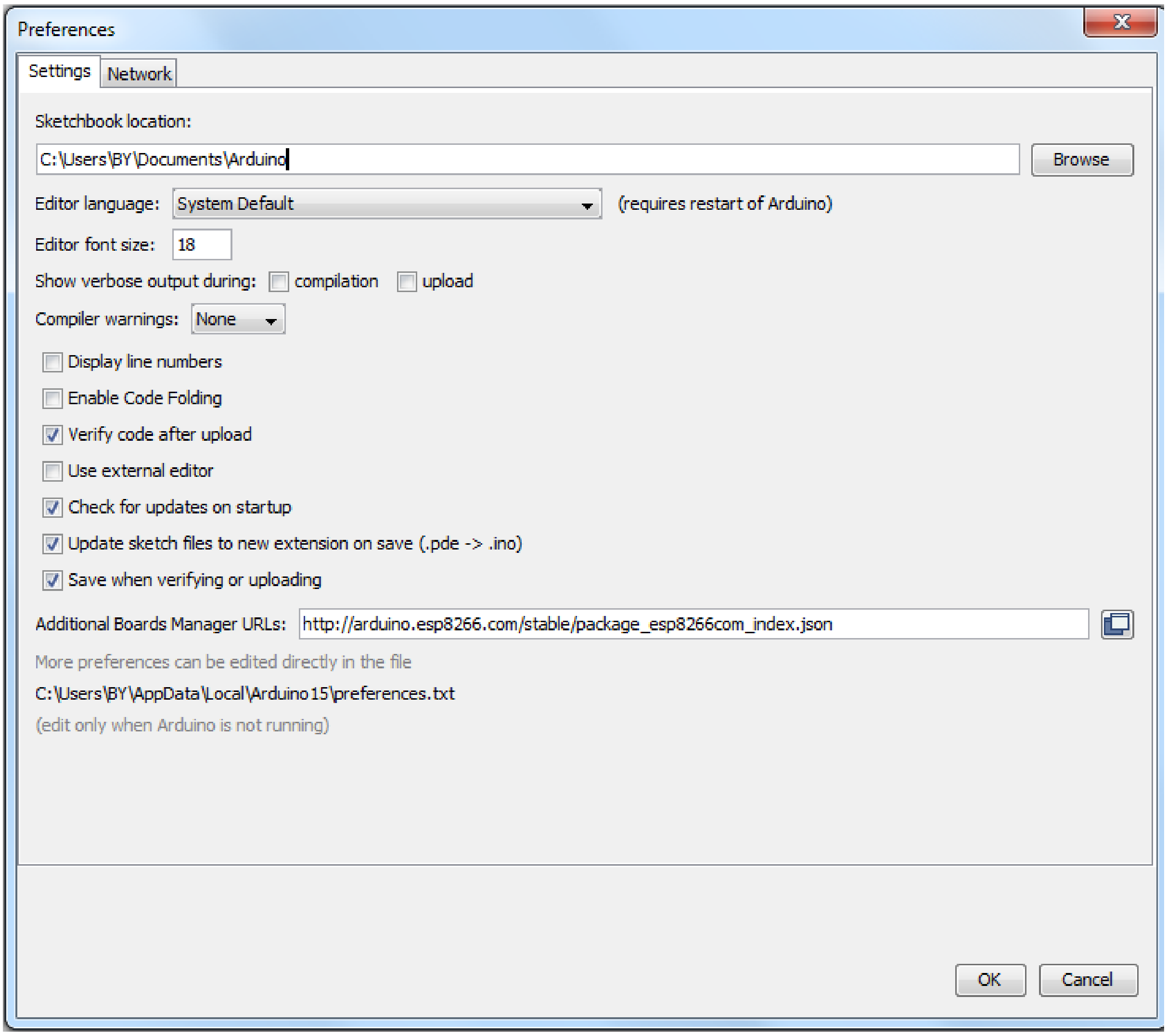
3.1 Install the Arduino IDE 1.6.4 or greater

Download Arduino IDE from Arduino.cc (1.6.4 or greater) - don't use 1.6.2 or lower version! You can use your existing IDE if you have already installed it.

You can also try downloading the ready-to-go package from the ESP8266-Arduino project, if the proxy is giving you problems.

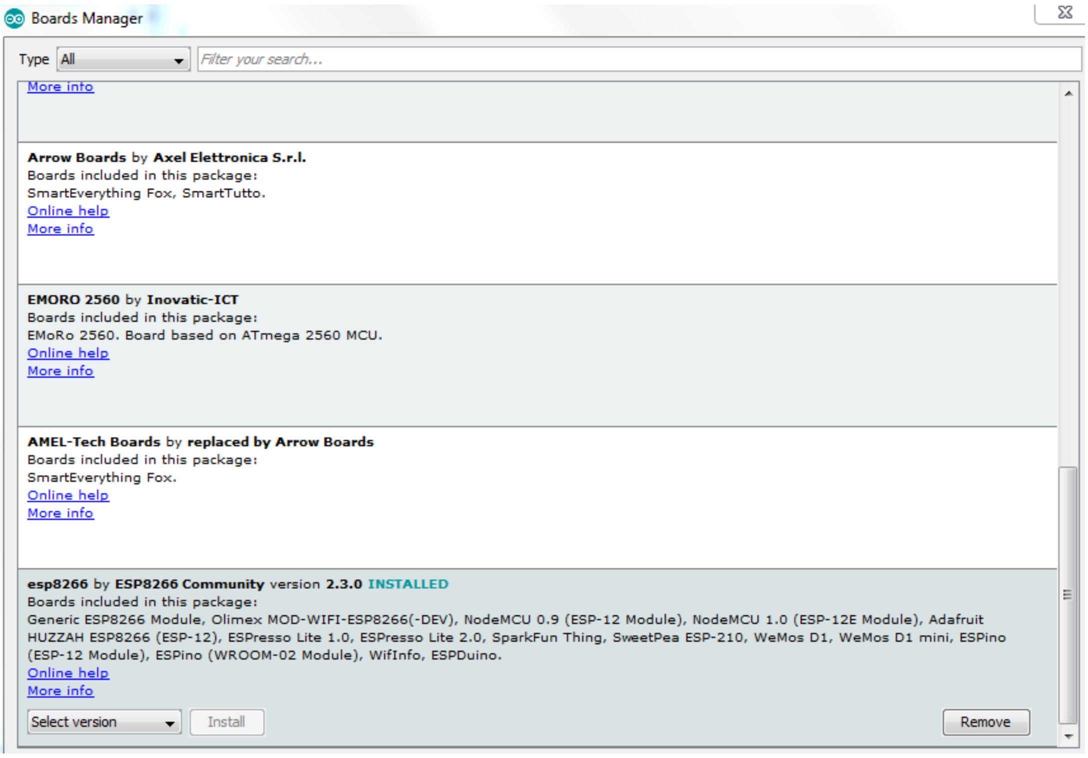
3.2 Install the ESP8266 Board Package

Enter ***http://arduino.esp8266.com/stable/package_esp8266com_index.json*** into *Additional Board Manager URLs* field in the Arduino v1.6.4+ preferences.



Click 'File' -> 'Preferences' to access this panel.

Next, use the Board manager to install the ESP8266 package.

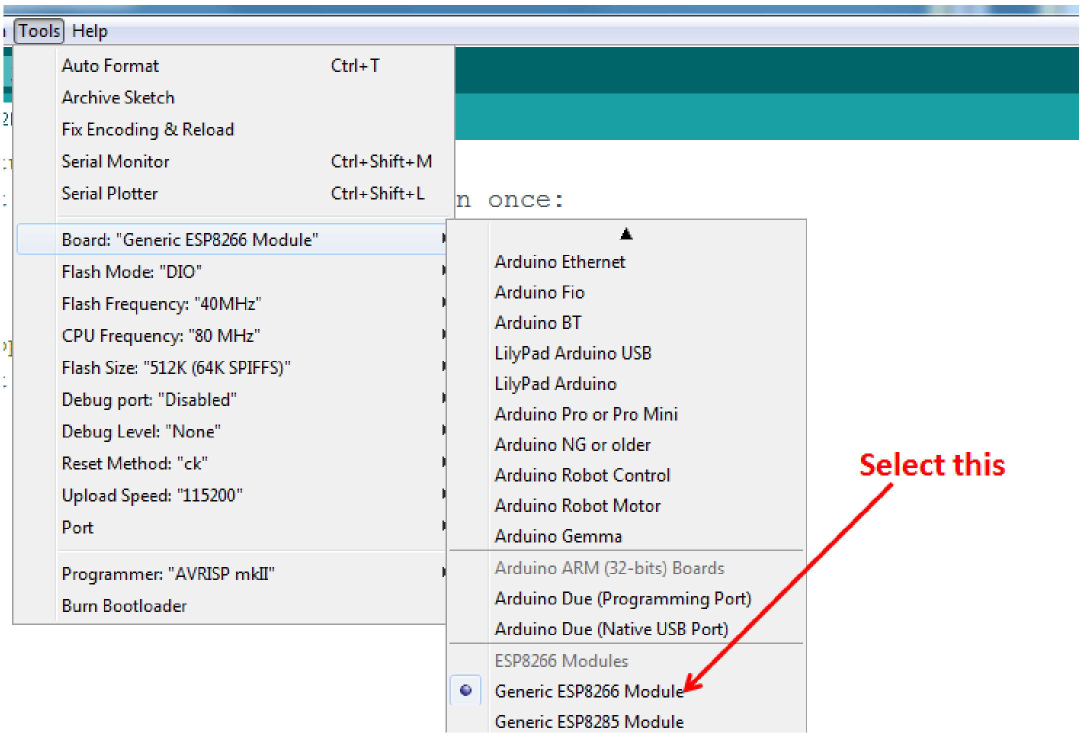


Click 'Tools' -> 'Board:' -> 'Board Manager…' to access this panel.

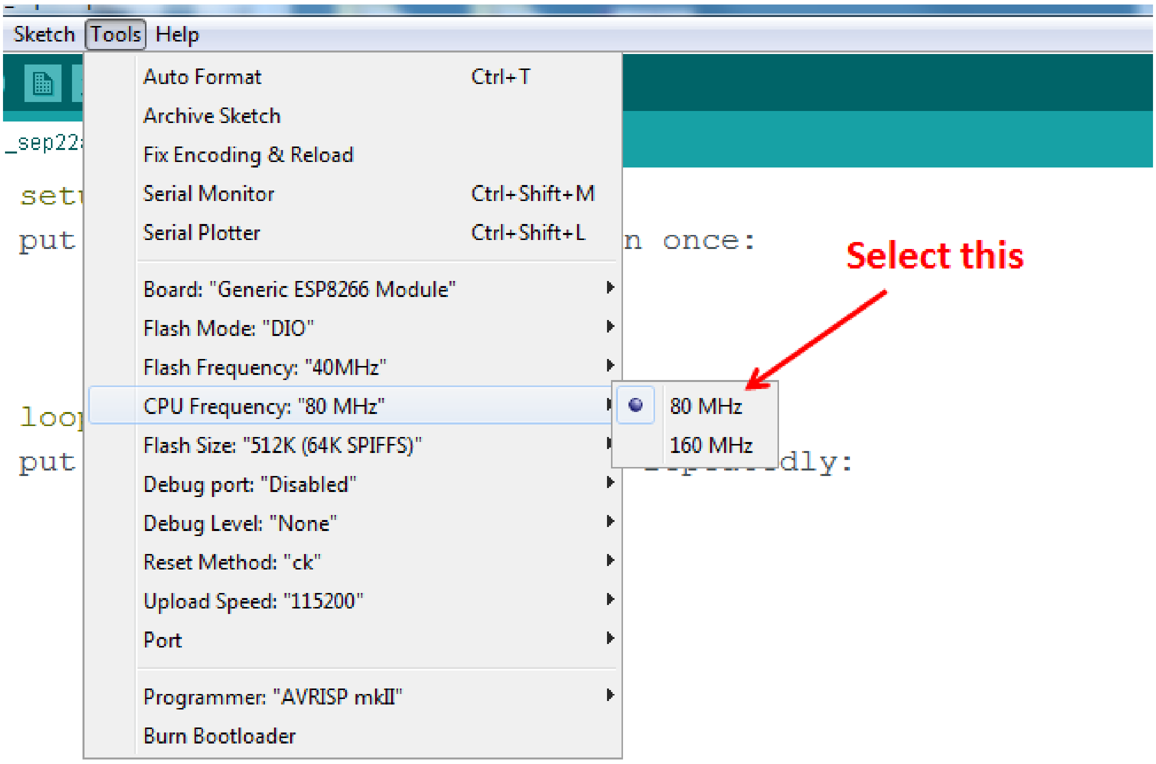
Scroll down to ' esp8266 by ESP8266 Community ' and click "Install" button to install the ESP8266 library package. Once installation completed, close and re-open Arduino IDE for ESP8266 library to take effect.

3.3 Setup ESP8266 Support

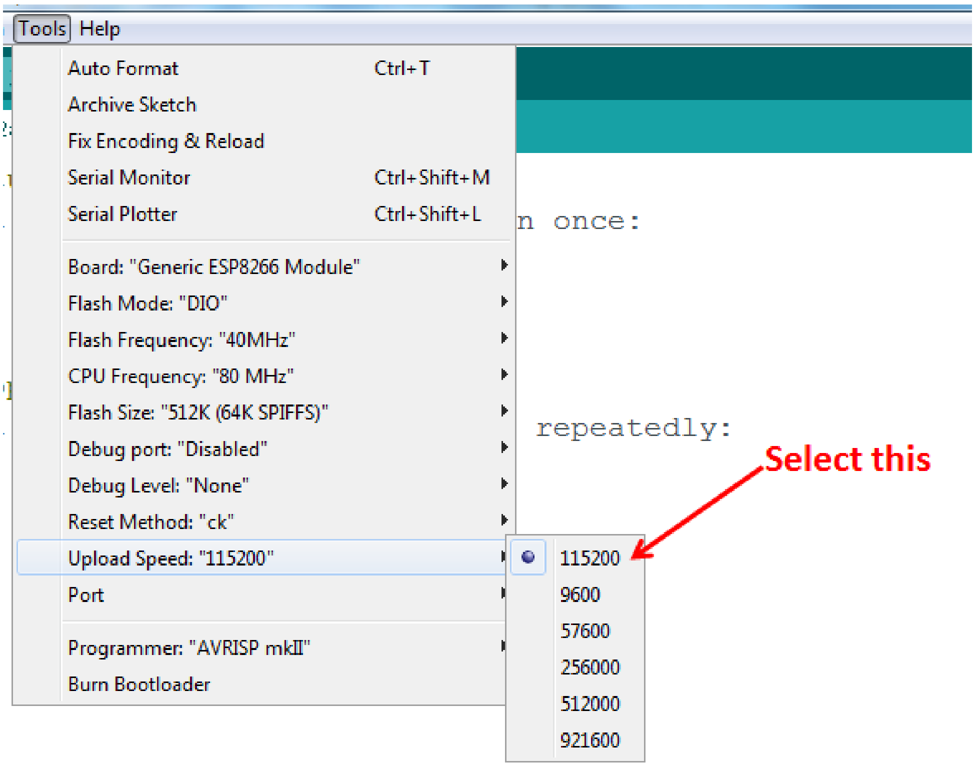
When you've restarted Arduino IDE, select 'Generic ESP8266 Module' from the 'Tools' -> 'Board:' dropdown menu.



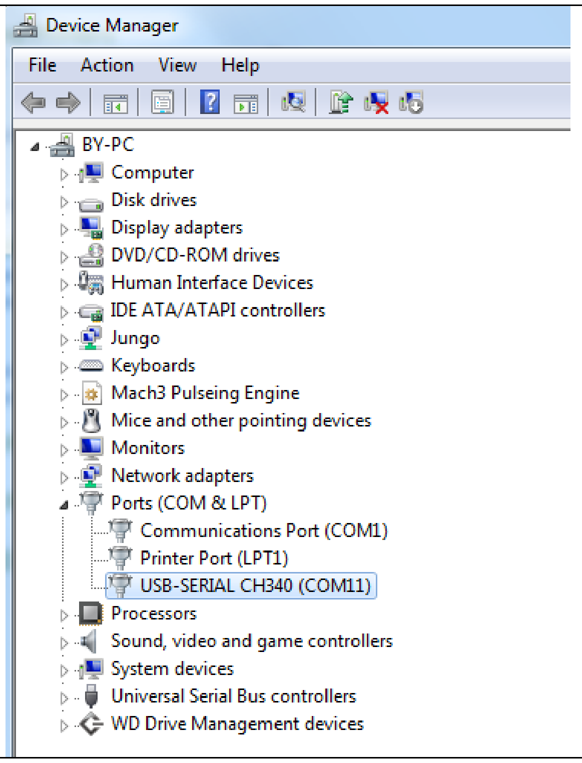
Select 80 MHz as the CPU frequency (you can try 160 MHz overclock later)

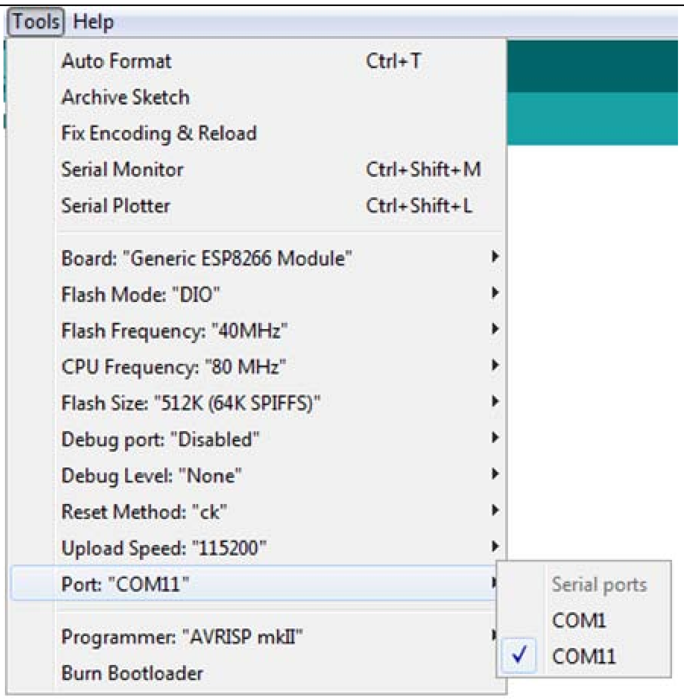


Select '115200' baud upload speed is a good place to start - later on you can try higher speeds but 115200 is a good safe place to start.



Go to your Windows 'Device Manager' to find out which Com Port 'USB-Serial CH340' is assigned to. Select the matching COM/serial port for your CH340 USB-Serial interface.





Find out which Com Port is assign for CH340	Select the correct Com Port as indicated on 'Device Manager"

<u>Note: if this is your first time using CH340 " USB-to-Serial " interface, please install the driver first before proceed the above Com Port setting. The CH340 driver can be download from the below site:</u>

<u>https://github.com/nodemcu/nodemcu-devkit/tree/master/Drivers</u>

3.4 Blink Test

We'll begin with the simple blink test.

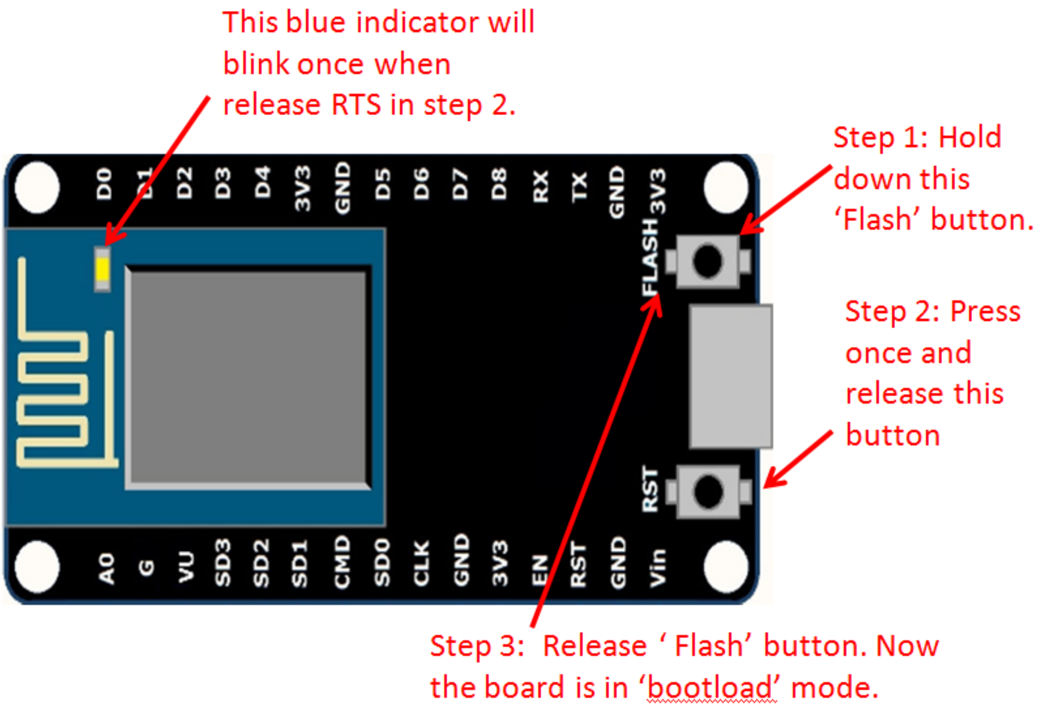
Enter this into the sketch window (and save since you'll have to). Connect a LED as shown in Figure3-1.

```
void setup() {
  pinMode(5, OUTPUT);   // GPIO05, Digital Pin D1
}

void loop() {
  digitalWrite(5, HIGH);
  delay(900);
  digitalWrite(5, LOW);
  delay(500);
}
```

Now you'll need to put the board into bootload mode. You'll have to do this before each upload. There is no timeout for bootload mode, so you don't have to rush!

- Hold down the 'Flash' button.
- While holding down ' Flash', press the 'RST' button.
- Release 'RST', then release 'Flash'

- When you release the 'RST' button, the blue indication will blink once, this means its ready to bootload.



Once the ESP board is in bootload mode, upload the sketch via the IDE, Figure 3-2.

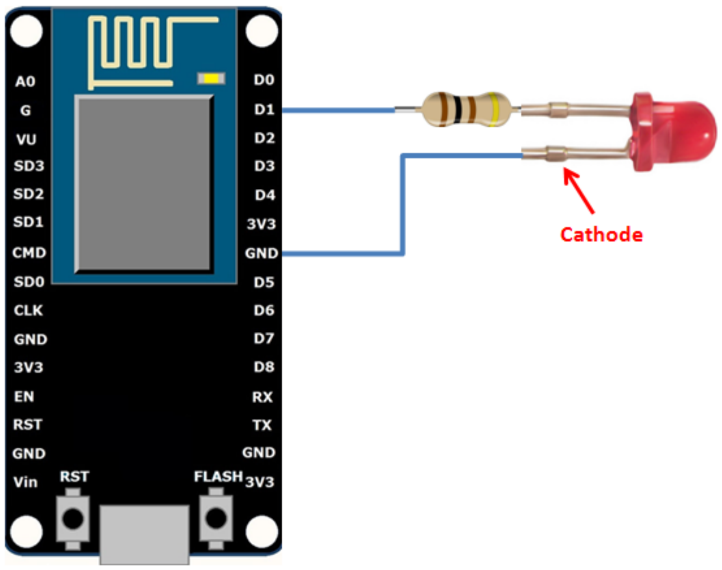


Figure3-1: Connection diagram for the blinking test

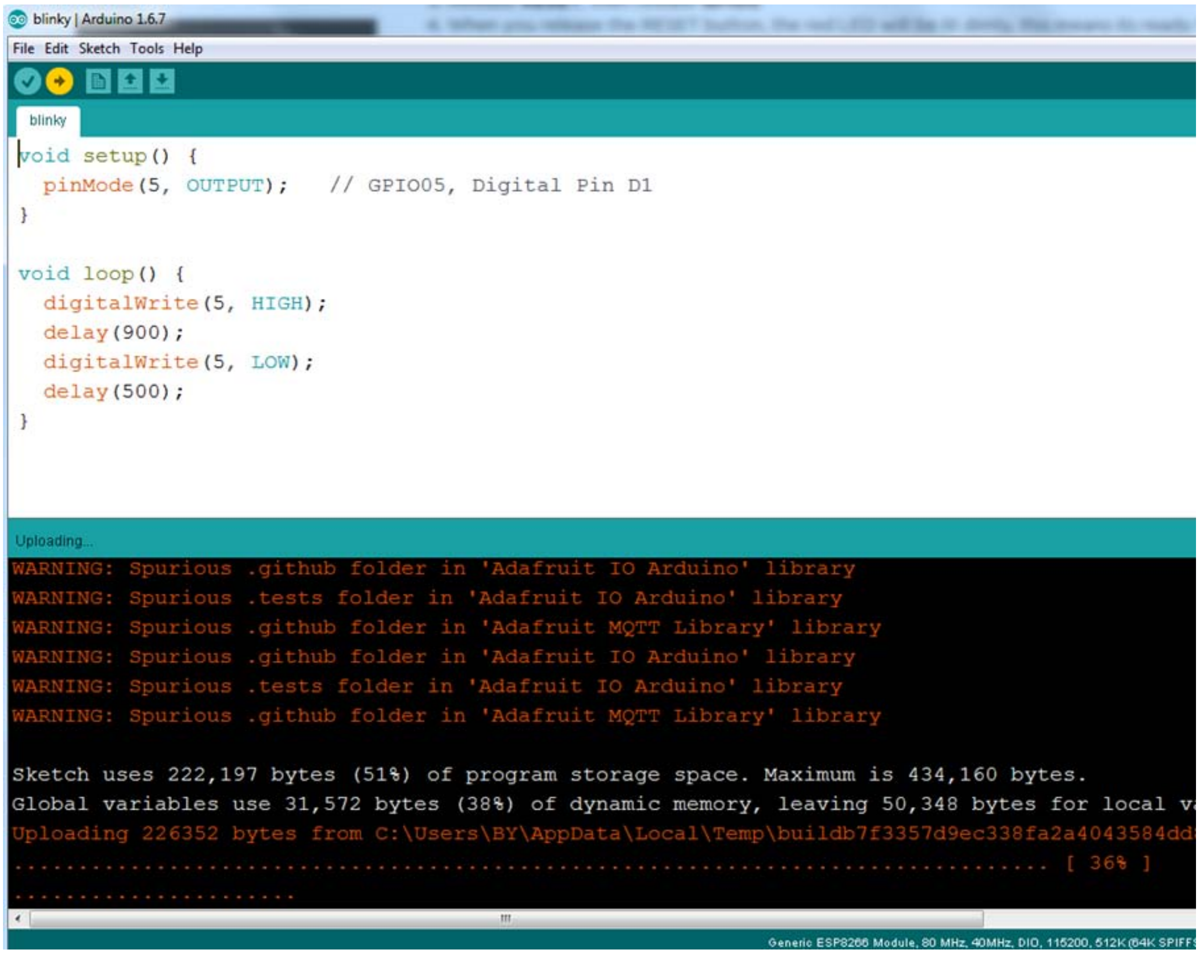


Figure 3.2: Uploading the sketch to ESP8266 NodeMCU module.

The sketch will start immediately - you'll see the LED blinking. Hooray!

3.5 Connecting via WiFi

OK once you've got the LED blinking, let’s go straight to the fun part, connecting to a webserver. Create a new sketch with this code:

Don’t forget to update:

const char* ssid   = "yourssid";

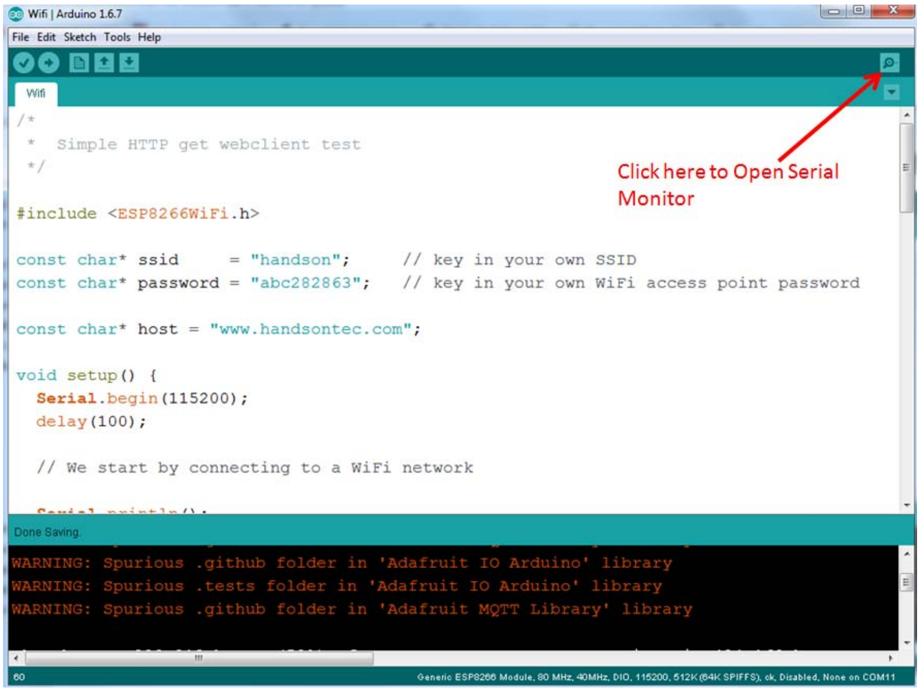
const char* password = "yourpassword";

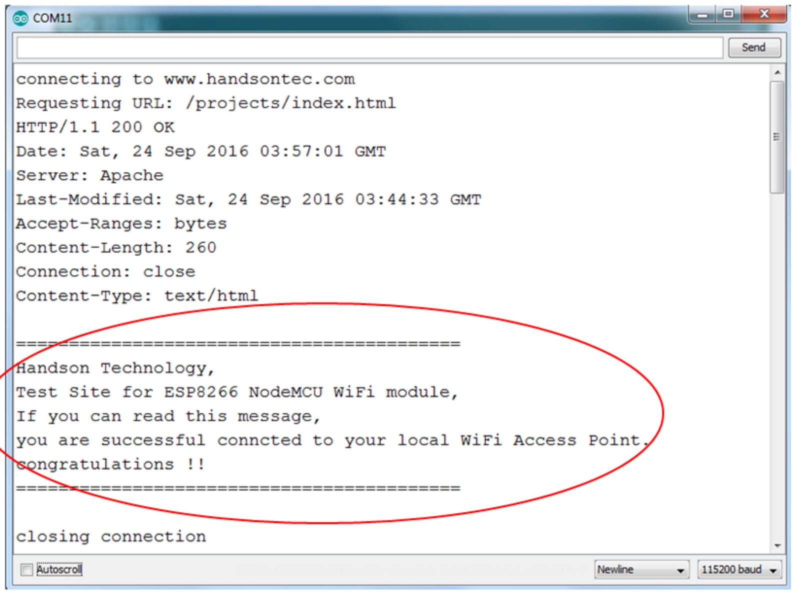
to your WiFi access point and password, then upload the same way: get into bootload mode, then upload code via IDE.

```
/*
 *  Simple HTTP get webclient test
 */

#include <ESP8266WiFi.h>

const char* ssid     = "handson";      // key in your own SSID
const char* password = "abc1234";      // key in your own WiFi access point
password
```

```
const char* host = "www.handsontec.com";

void setup() {
  Serial.begin(115200);
  delay(100);

  // We start by connecting to a WiFi network

  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

int value = 0;

void loop() {
  delay(5000);
  ++value;

  Serial.print("connecting to ");
  Serial.println(host);

  // Use WiFiClient class to create TCP connections
  WiFiClient client;
  const int httpPort = 80;
  if (!client.connect(host, httpPort)) {
    Serial.println("connection failed");
    return;
  }

  // We now create a URI for the request
  String url = "/projects/index.html";
  Serial.print("Requesting URL: ");
  Serial.println(url);

  // This will send the request to the server
  client.print(String("GET ") + url + " HTTP/1.1\r\n" +
               "Host: " + host + "\r\n" +
               "Connection: close\r\n\r\n");
  delay(500);

  // Read all the lines of the reply from server and print them to Serial
  while(client.available()){
    String line = client.readStringUntil('\r');
    Serial.print(line);
  }

  Serial.println();
  Serial.println("closing connection");
}
```

Open up the IDE serial console at 115200 baud to see the connection and webpage printout!





That's it, pretty easy right ! This section is just to get you started and test out your module.

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

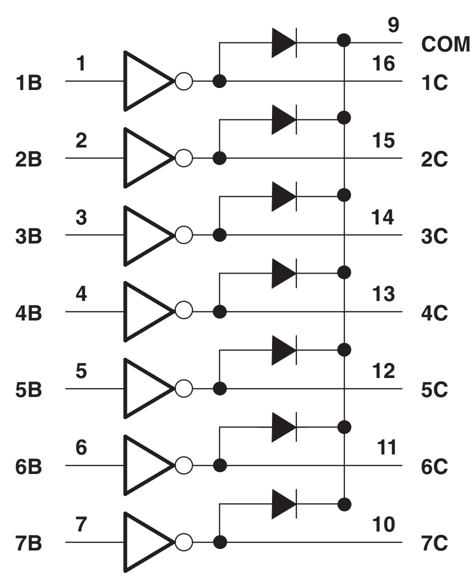


Table of Contents

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**

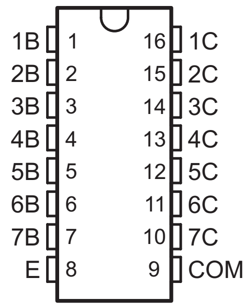
- Updated temperature rating for ULN2003AI in the ORDERING INFORMATION table 1

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	I	Channel 1 through 7 Darlington base input
2B	2		
3B	3		
4B	4		
5B	5		
6B	6		
7B	7		
1C	16	O	Channel 1 through 7 Darlington collector output
2C	15		
3C	14		
4C	13		
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			UNIT
					MIN	TYP	MAX	
$V_{I(on)}$	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_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			ULN2004A			UNIT
					MIN	TYP	MAX	MIN	TYP	MAX	
$V_{I(on)}$	ON-state input voltage	Figure 14	V_{CE} = 2 V	I_C = 125 mA						5	V
				I_C = 200 mA			2.4			6	
				I_C = 250 mA			2.7				
				I_C = 275 mA						7	
				I_C = 300 mA			3				
				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_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	ULN2003AI TYP	ULN2003AI 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	ULN2003AI TYP	ULN2003AI 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