# INDEX

|  |  |
| --- | --- |
| **NAME OF CONTENTS ABSTRACT CHAPTER-1** | **PAGE NO** |
| 1. INTRODUCTION | 4 -11 |
| **CHAPTER-2**  2. LITERATURE REVIEW | 12-14 |
| **CHAPTER-3**  3 .PROPOSED SYSTEM | 15 |
| 3.1 BLOCK AND CIRCUIT DIAGRAM | 16 |
| **CHAPTER-4**  4. HARDWARE COMPONENTS | 17 |
| POWER SUPPLY | 17 |
| ARDUNIO | 19 |
| FEATURES | 20 |
| HARDWARE COMPONENTS | 22 |
| EXPLANATION ABOUT COMPONENTS SOFTWARE | 23  26 |
| **CHAPTER-5**  5.RESULT | 39 |

|  |  |
| --- | --- |
| **CHAPTER-6** |  |
| 6. CONCLUSION | 40 |
| **CHAPTER-7**  7. REFERENCES | 41-42 |

**ABSTRACT**:-

This project involves the development of an automated cradle system using an Arduino Uno, various environmental sensors, and an ESP-01 module for WiFi connectivity. The system monitors temperature, humidity, sound, and rain conditions to control a cradle mechanism powered by a servo motor. The DHT sensor measures temperature and humidity, the sound sensor detects noise levels, and the rain sensor monitors precipitation. Data from these sensors is processed by the Arduino, which then adjusts the cradle's movement based on predefined thresholds. The ESP-01 module provides remote monitoring and control capabilities through WiFi, enabling users to visualize sensor data and control the cradle via a web interface or mobile app. This project aims to enhance comfort and safety by automating cradle movement in response to environmental changes.

**Keywords**: Arduino Uno, automated cradle system, environmental monitoring, DHT sensor, sound sensor, rain sensor, ESP-01 module, WiFi connectivity, servo motor control, remote monitoring, Internet of Things (IoT), home automation

|  |  |  |
| --- | --- | --- |
| **LIST OF FIGURES**  **FIGURE NO** | **NAME OF THE FIGURE** | **PAGE NO** |
| 1.1 | Layered architecture of an embedded system | 6 |
| 1.2 | Block diagram of Embedded system architecture | 7 |
| 3.1  3.2 | Block diagram  Circuit diagram | 15  16 |
| 4.1 | Block diagram of power supply | 17 |
| 4.2 | Power supply circuit diagram | 18 |
| 4.3 | Arduino Nano | 23 |
| 4.4 | Bluetooth Module | 24 |
| 4.5 | Relay | 25 |
| 4.6 | LCD I2C module | 25-26 |
| 4.10 | Software logo | 27 |
| 4.11 | Arduino Nano | 27 |
| 4.12 | A-to-B USB cable | 28 |
| 4.13 | Device manager | 29 |
| 4.14 | Examples | 31 |
| 4.15 | Selection of board | 32 |
| 4.16 | Selection of serial port | 32 |
| 4.17  5.1 | Example code  Front view and Side view of kit | 33  39 |

# CHAPTER 1: INTRODUCTION

The advancement of smart technology has paved the way for innovative applications in home automation, significantly enhancing the convenience and safety of everyday life. One such application is the development of an automated cradle system that adapts to environmental conditions. This project aims to create a sophisticated cradle system using an Arduino Uno, equipped with various sensors and an ESP-01 module for WiFi connectivity, to provide a seamless and responsive user experience.

The core of the system revolves around the Arduino Uno, a versatile microcontroller known for its ease of use and compatibility with numerous sensors and modules. By leveraging this microcontroller, the system can efficiently process data from various sensors, including a DHT sensor for temperature and humidity, a sound sensor for noise detection, and a rain sensor. These sensors are critical in ensuring the cradle responds appropriately to changes in the environment, thereby maintaining optimal conditions for its user.

Temperature and humidity are crucial factors in ensuring comfort and safety in many environments. The DHT sensor, which accurately measures these parameters, provides essential data that the Arduino processes to determine the appropriate actions for the cradle system. For instance, if the temperature or humidity reaches uncomfortable levels, the system can adjust the cradle's movement to enhance ventilation and comfort.

Noise levels are another critical aspect monitored by the system. The inclusion of a sound sensor enables the cradle to respond to sudden loud noises or prolonged periods of noise, which could indicate distress or the need for intervention. This feature is particularly useful in settings such as nurseries or hospitals, where maintaining a quiet and stable environment is paramount.

The rain sensor adds an additional layer of functionality, allowing the system to detect precipitation. This is particularly useful for outdoor or semi-outdoor cradles, ensuring they are not exposed to harsh weather conditions. Upon detecting rain, the system can either alert the user or automatically adjust the cradle's position to a sheltered area, thereby protecting it from potential water damage.

To enhance user interaction and control, the system incorporates an ESP-01 module, enabling WiFi connectivity. This module allows users to remotely monitor environmental conditions and control the cradle's movements via a web interface or mobile application. Such remote capabilities ensure that users can maintain optimal conditions and respond to changes in real time, even when they are not physically present.

### Introduction To Embedded Systems

An embedded system can be defined as a computing device that does a specific focused job. Appliances such as the air-conditioner, VCD player, DVD player, printer, fax

machine, mobile phone etc. are examples of embedded systems. Each of these appliances will have a processor and special hardware to meet the specific requirement of the application along with the embedded software that is executed by the processor for meeting that specific requirement. The embedded software is also called “firm ware”. The desktop/laptop computer is a general purpose computer. You can use it for a variety of applications such as playing games, word processing, accounting, software development and so on. In contrast, the software in the embedded systems is always fixed listed below:

Embedded systems do a very specific task, they cannot be programmed to do different things. Embedded systems have very limited resources, particularly the memory. Generally, they do not have secondary storage devices such as the CDROM or the floppy disk. Embedded systems have to work against some deadlines. A specific job has to be completed within a specific time. In some embedded systems, called real-time systems, the deadlines are stringent. Missing a deadline may cause a catastrophe-loss of life or damage to property. Embedded systems are constrained for power. As many embedded systems operate through a battery, the power consumption has to be very low. Some embedded systems have to operate in extreme environmental conditions such as very high temperatures and humidity.

##### Overview Of Embedded System Architecture

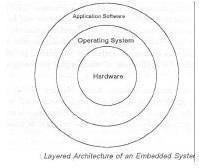
Every embedded system consists of custom-built hardware built around a Central Processing Unit (CPU). This hardware also contains memory chips onto which the software is loaded.

The software residing on the memory chip is also called the ‘firmware’. The embedded system architecture can be represented as a layered architecture as shown in Fig.

The operating system runs above the hardware, and the application software runs above the operating system. The same architecture is applicable to any computer including a desktop computer. However, there are significant differences. It is not compulsory to have an operating system in every embedded system. For small appliances such as remote control units, air conditioners, toys etc., there is no need for an operating system and you can write

onlythe software specific to that application.

For applications involving complex processing, it is advisable to have an operating system. In



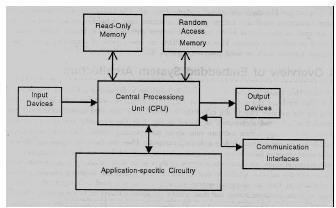
##### Fig 1.1 : Layered architecture of an embedded system

such a case, you need to integrate the application software with the operating system and then transfer the entire software on to the memory chip. Once the software is transferred to the memory chip, the software will continue to run for a long time you don’t need to reload new software.

Now, let us see the details of the various building blocks of the hardware of an embedded system. As shown in Fig. the building blocks are;

Central Processing Unit (CPU)

* Memory(Read-only Memory and Random Access Memory)
* Input Devices
* Output devices
* Communication interfaces

* 

##### * Fig 1.2 : Block diagram of Embedded system architecture

*

* Application-specific circuitry

##### Central Processing Unit (CPU)

The Central Processing Unit (processor, in short) can be any of the following: microcontroller, microprocessor or Digital Signal Processor (DSP). A micro-controller is a low-cost processor. Its main attraction is that on the chip itself, there will be many other components such as memory, serial communication interface, analog-to digital converter etc. So, for small applications, a micro-controller is the best choice as the number of external components required will be very less. On the other hand, microprocessors are more powerful, but you need to use many external components with them. D5P is used mainly for applications in which signal processing is involved such as audio and video processing.

##### Memory

The memory is categorized as Random Access 11emory (RAM) and Read Only Memory (ROM). The contents of the RAM will be erased if power is switched off to the chip, whereas ROM retains the contents even if the power is switched off. So, the firmware is stored in the ROM. When power is switched on, the processor reads the ROM; the program is program is executed.

##### Input devices

Unlike the desktops, the input devices to an embedded system have very limited capability. There will be no keyboard or a mouse, and hence interacting with the embedded system is no easy task. Many embedded systems will have a small keypad-you press one key to give a specific command. A keypad may be used to input only the digits. Many embedded systems used in process control do not have any input device for user interaction; they take inputs from sensors or transducers 1’fnd produce electrical signals that are in turn fed to other systems.

##### Output devices

The output devices of the embedded systems also have very limited capability. Some embedded systems will have a few Light Emitting Diodes (LEDs) to indicate the health status of the system modules, or for visual indication of alarms. A small Liquid Crystal Display (LCD) may also be used to display some important parameters.

##### Communication interfaces

The embedded systems may need to, interact with other embedded systems at they may have to transmit data to a desktop. To facilitate this, the embedded systems are provided with one or a few communication interfaces such as RS232, RS422, RS485, Universal Serial Bus (USB), IEEE 1394, Ethernet etc.

##### Application-specific circuitry

Sensors, transducers, special processing and control circuitry may be required fat an embedded system, depending on its application. This circuitry interacts with the processor to carry out the necessary work. The entire hardware has to be given power supply either through the 230 volts main supply or through a battery. The hardware has to design in such a waythat the power consumption is minimized.

##### Following Are The Advantages Of Embedded Systems

1. They are designed to do a specific task and have real time performance constraints which must be met.
2. Theyallow the system hardware to be simplified so costs are reduced.
3. They are usually in the form of small computerized parts in larger devices which serve a general purpose.

##### Application Areas

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

##### Consumer appliances

At home we use a number of embedded systems which include digital camera, digital diary, DVD player, electronic toys, microwave oven, remote controls for TV and air- conditioner, VCO player, video game consoles, video recorders etc. Today’s high-tech car has about 20 embedded systems for transmission control, engine spark control, air- conditioning, navigation etc. Even wristwatches are now becoming embedded systems. The palmtops are powerful embedded systems using which we can carry out many general- purpose tasks such as playing games and word processing.

##### Office automation

The office automation products using em embedded systems are copying machine, fax machine, keytelephone, modem, printer, scanner etc.

##### Industrial automation

Today a lot of industries use embedded systems for process control. These include pharmaceutical, cement, sugar, oil exploration, nuclear energy, electricity generation and transmission. The embedded systems for industrial use are designed to carry out specific tasks such as monitoring the temperature, pressure, humidity, voltage, current etc., and then take appropriate action based on the monitored levels to control other devices or to send information to a centralized monitoring station. In hazardous industrial environment, where human presence has to be avoided, robots are used, which are programmed to do specific jobs. The robots are now becoming very powerful and carry out many interesting and complicated tasks such as hardware assembly.

##### Medical electronics

Almost every medical equipment in the hospital is an embedded system. These equipments include diagnostic aids such as ECG, EEG, blood pressure measuring devices, X-ray scanners; equipment used in blood analysis, radiation, colonscopy, endoscopy etc. Developments in medical electronics have paved way for more accurate diagnosis of diseases.

##### Computer networking

Computer networking products such as bridges, routers, Integrated Services Digital Networks (ISDN), Asynchronous Transfer Mode (ATM), X.25 and frame relay switches are embedded systems which implement the necessary data communication protocols. For

example, a router interconnects two networks. The two networks may be running different protocol stacks. The router’s function is to obtain the data packets from incoming pores, analyze the packets and send them towards the destination after doing necessary protocol conversion. Most networking equipments, other than the end systems (desktop computers) we use to access the networks, are embedded systems

##### Telecommunications

In the field of telecommunications, the embedded systems can be categorized as subscriber terminals and network equipment. The subscriber terminals such as key telephones, ISDN phones, terminal adapters, web cameras are embedded systems. The network equipment includes multiplexers, multiple access systems, Packet Assemblers Dissemblers (PADs), satel1ite modems etc. IP phone, IP gateway, IP gatekeeper etc. are the latest embedded systems that provide very low-cost voice communication over the Internet.

##### Wireless technologies

Advances in mobile communications are paving way for many interesting applications using embedded systems. The mobile phone is one of the marvels of the last decade of the 20th century. It is a very powerful embedded system that provides voice communication while we are on the move. The Personal Digital Assistants and the palmtops can now be used to access multimedia services over the Internet. Mobile communication infrastructure such as base station controllers, mobile switching centers are also powerful embedded systems.

##### Insemination

Testing and measurement are the fundamental requirements in all scientific and engineering activities. The measuring equipment we use in laboratories to measure parameters such as weight, temperature, pressure, humidity, voltage, current etc. are all embedded systems. Test equipment such as oscilloscope, spectrum analyzer, logic analyzer, protocol analyzer, radio communication test set etc. are embedded systems built around powerful processors. Thank to miniaturization, the test and measuring equipment are now becoming portable facilitating easytesting and measurement in the field by field-personnel.

##### Security

Security of persons and information has always been a major issue. We need to protect our homes and offices; and also the information we transmit and store. Developing embedded systems for security applications is one of the most lucrative businesses nowadays. Security

devices at homes, offices, airports etc. for authentication and verification are embedded systems. Encryption devices are nearly 99 per cent of the processors that are manufactured end up in~ embedded systems. Embedded systems find applications in.every industrial segment-consumer electronics, transportation, avionics, biomedical engineering, manufacturing, process control and industrial automation, data communication, telecommunication, defense, security etc. Used to encrypt the data/voice being transmitted on communication links such as telephone lines. Biometric systems using fingerprint and face recognition are now being extensively used for user authentication in banking applications as well as for access control in high security buildings.

##### Finance

Financial dealing through cash and cheques are now slowly paving way for transactions using smart cards and ATM (Automatic Teller Machine, also expanded as Any Time Money) machines. Smart card, of the size of a credit card, has a small micro-controller and memory; and it interacts with the smart card reader! ATM machine and acts as an electronic wallet. Smart card technology has the capability of ushering in a cashless society. Well, the list goes on. It is no exaggeration to say that eyes wherever we go, we can see, or at least feel, the work of an embedded system.

# CHAPTER 2: LITTERATURE SURVEY

**W. A. Jabbar, H. K. Shang, S. N. I. S. Hamid, A. A. Almohammedi, R. M. Ramli and M. A. H. Ali, "IoT-BBMS: Internet of Things-Based Baby Monitoring System for Smart Cradle,"**

The current number of working mothers has greatly increased. Subsequently, baby care has become a daily challenge for many families. Thus, most parents send their babies to their grandparents' house or to baby care houses. However, the parents cannot continuously monitor their babies' conditions either in normal or abnormal situations. Therefore, an Internet of Things-based Baby Monitoring System (IoT-BBMS) is proposed as an efficient and low-cost IoT-based system for monitoring in real time. We also proposed a new algorithm for our system that plays a key role in providing better baby care while parents are away. In the designed system, Node Micro-Controller Unit (NodeMCU) Controller Board is exploited to gather the data read by the sensors and uploaded via Wi-Fi to the AdaFruit MQTT server. The proposed system exploits sensors to monitor the baby's vital parameters, such as ambient temperature, moisture, and crying. A prototype of the proposed baby cradle has been designed using Nx Siemens software, and a red meranti wood is used as the material for the cradle. The system architecture consists of a baby cradle that will automatically swing using a motor when the baby cries. Parents can also monitor their babies' condition through an external web camera and switch on the lullaby toy located on the baby cradle remotely via the MQTT server to entertain the baby. The proposed system prototype is fabricated and tested to prove its effectiveness in terms of cost and simplicity and to ensure safe operation to enable the baby-parenting anywhere and anytime through the network. Finally, the baby monitoring system is proven to work effectively in monitoring the baby's situation and surrounding conditions according to the prototype.

**J. K. Reena and R. Parameswari, "A Smart Health Care Monitor System in IoT Based Human Activities of Daily Living: A Review,"**

IoT is extensively used in all applications. In the healthcare system IoT plays a major role of connecting doctors to the patients by using health monitoring devices. The elderly and the disabled people find this to be very economic and beneficial. There are various methods that monitor the wellbeing of the elderly and in this paper comparison various data mining methods that is being used from the data that is got from things like smart meter, appliance usage, video surveillance and their prediction accuracy is reviewed.

**S. Ananth, P. Sathya and P. Madhan Mohan, "Smart Health Monitoring System through IOT,"**

IOT is one of the booming field in forthcoming years and plays a major role in the field of health care. IOT helps in connecting the people by empowering their health and wealth in a smart way through wearable gadgets. Recent improvement in wireless sensor networks have created a new trend in internet of things. Smart health is important application in Internet of things. Patients with abnormal health conditions can be quickly monitored through smart health care system and provide a rapid solution for the patients. This type of solutions can be encountered through wearable gadgets that continuously monitor the activity and condition of the patient in a predictable method. The main aim of this work is to provide an extensive research in capturing the sensor data's, analyzing the data and providing a feedback to patients based on different health parameters.

**H. Singh et al., "Neo-Bedside Monitoring Device for Integrated Neonatal Intensive Care Unit (iNICU),"**

We report the design and implementation of an affordable bedside device, Neo is capable of acquiring vital data in real time by integrating with diverse devices connected to newborns in neonatal intensive care units (NICUs). NICUs are equipped with multiple vital sign monitoring devices that are connected to the premature newborn and acquire a few gigabytes of data every day. The continuous vital data from these devices are manually documented every hour. This introduces errors and loses a large amount of high-resolution data. The intermittent documentation of physiological data also makes it difficult for clinicians to visualize and detect the trends of diagnostic utility. Neo is built on an affordable Internet of Things platform that aggregates and sends the real-time data to a cloud-based big data platform called integrated NICU. Apart from minimizing documentation errors, the device enables data acquisition at sufficiently real-time rates as to indicate the current status of all the patients at the NICU. Neo automates immediate vital sign status and past trends as graphs or charts that the doctor and nurses can view from anywhere through the Internet. Physiological signal and clinical parameters from Neo is used to score different diseases like sepsis, respiratory distress syndrome, necrotizing enterocolitis, and retinopathy of prematurity. This score predicts the physiological health of the newborn and aid clinicians in decision-making ensuring timely intervention.

**M. Koli, P. Ladge, B. Prasad, R. Boria and N. J. Balur, "Intelligent Baby Incubator,"**

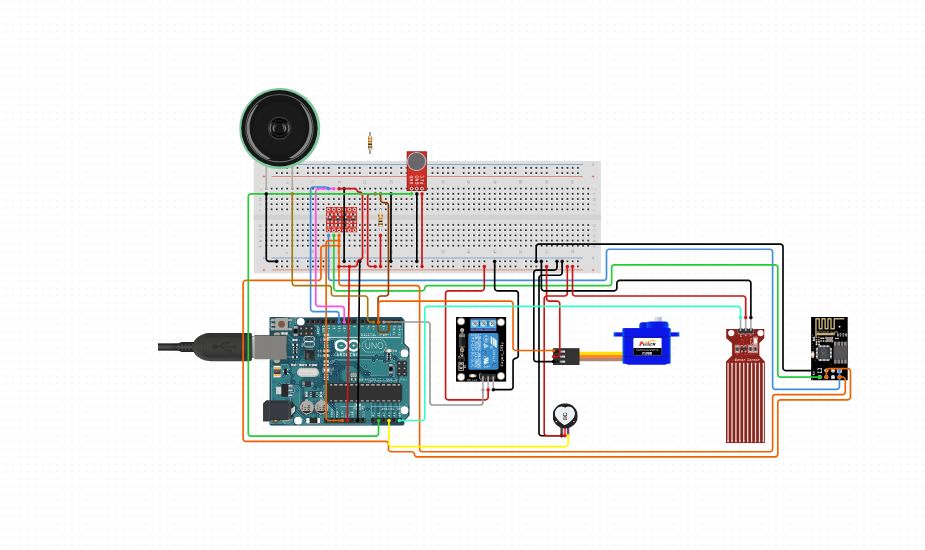
In today's world where temperature is an important part of our environment. Any variations in the temperature can affect all the living organism and also some semiconductor materials. Our project is to provide a control on the changes in temperature for certain applications such as baby incubator. Worldwide incubator is used for the infants to improve the survival of babies by providing them warm environment and reducing the heat loss from the baby's body. Basically, incubator is the standard method used worldwide. In this we are using Arduino, temperature sensor for monitoring or controlling the temperature of the baby's body. The constant temperature of 36.5-37.2° C is required by the baby as it is required by the baby as in mother's womb. The programming code of Arduino is used to control the temperature. Buzzer is used for the indication of change in temperature in a neonatal nursing home where a number of incubators are kept. IOT web designing is used for monitoring the changes in temperature anywhere at all time.

# CHAPTER 3: PROPOSED SYSTEM

The proposed automated cradle system is designed to enhance comfort and safety by integrating environmental monitoring, automation, and remote control capabilities. Utilizing an Arduino Uno microcontroller, the system incorporates a range of sensors and modules to monitor and respond to changes in temperature, humidity, sound, and rain conditions. The DHT sensor measures temperature and humidity levels, providing crucial data for maintaining optimal environmental conditions around the cradle. The sound sensor detects noise levels, allowing the system to respond to sudden loud noises or prolonged noise, which could indicate distress or the need for intervention. Additionally, the rain sensor monitors for the presence of rain, which is particularly important for outdoor or semi-outdoor settings, ensuring the cradle is protected from adverse weather conditions.

A servo motor controls the cradle's movement, adjusting its position based on the sensor inputs to ensure comfort and safety. A relay module acts as a switch to control high-power devices if needed, expanding the system's capability to manage additional components. The ESP-01 module provides WiFi connectivity, enabling remote monitoring and control. Users can access the system via a web interface or mobile application, allowing for real-time adjustments and monitoring.

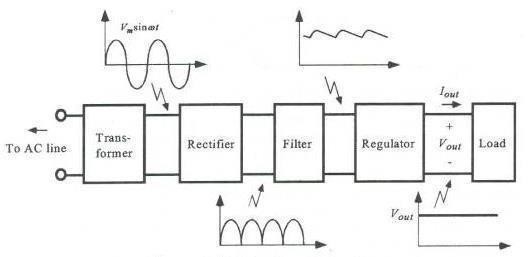
The system operates by continuously monitoring environmental conditions through the sensors, with the Arduino Uno collecting and processing this data to make informed decisions about the cradle's operation. Based on the sensor data, the Arduino Uno controls the servo motor to adjust the cradle's movement. For instance, if the temperature or humidity levels exceed comfortable thresholds, the system can modify the cradle's position to improve ventilation. Similarly, if the sound sensor detects a high noise level, the system can respond by adjusting the cradle or sending an alert. If rain is detected, the system can either alert the user or automatically move the cradle to a sheltered position to protect it from water damage.

* 1. **CIRCUIT DIAGRAM**

# CHAPTER 4: HARDWARE COMPONENTS

##### 4.1 Power Supply

The input to the circuit is applied from the regulated power supply. The a.c. input i.e., 230V from the mains supply is step down by the transformer to 12V and is fed to a rectifier. The output obtained from the rectifier is a pulsating d.c voltage. So in order to get a pure d.c voltage, the output voltage from the rectifier is fed to a filter to remove any a.c components present even after rectification. Now, this voltage is given to a voltage regulator to obtain a pure constant dc voltage.



##### Transformer:

Usually, DC voltages are required to operate various electronic equipment and these voltages are 5V, 9V or 12V. But these voltages cannot be obtained directly. Thus the a.c input available at the mains supply i.e., 230V is to be brought down to the required voltage level. This is done by a transformer. Thus, a step down transformer is employed to decrease the voltage to a required level.

##### Rectifier:

The output from the transformer is fed to the rectifier. It converts A.C. into pulsating

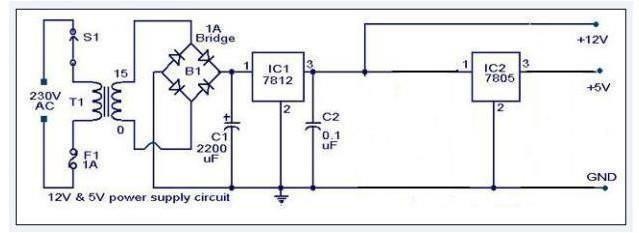
D.C. The rectifier may be a half wave or a full wave rectifier. In this project, a bridge rectifier is used because of its merits like good stability and full wave rectification.

##### Filter:

Capacitive filter is used in this project. It removes the ripples from the output of rectifier and smoothens the D.C. Output received from this filter is constant until the mains voltage and load is maintained constant. However, if either of the two is varied, D.C. voltage received at this point changes. Therefore a regulator is applied at the output stage.

##### Voltage regulator:

As the name itself implies, it regulates the input applied to it. A voltage regulator is an electrical regulator designed to automatically maintain a constant voltage level. In this project, power supply of 5V and 12V are required. In order to obtain these voltage levels, 7805 and 7812 voltage regulators are to be used. The first number 78 represents positive supply and the numbers 05, 12 represent the required output voltage levels.



##### Fig 4.2 : Power supply circuit diagram

**ARDUINO**

**Arduino** is an open-source computer hardware and software company, project and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical and digital world. The project's products are distributed as open-source hardware and software, which are licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL),permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially in preassembled form or as do-it-yourself (DIY) kits.

Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards or breadboards (shields) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers are typically programmed using a dialect of features from the programming languages C and C++. In addition to using traditional compiler toolchains, the Arduino project provides an integrated development environment (IDE) based on the Processing language project.

The Arduino project started in 2003 as a program for students at the Interaction Design Institute Ivrea in Ivrea, Italy.aiming to provide a low-cost and easy way for novices and professionals to create devices that interact with their environment using sensors and actuators. Common examples of such devices intended for beginner hobbyists include simple robots, thermostats and motion detectors.

The name Arduino comes from a bar in Ivrea, Italy, where some of the founders of the project used to meet. The bar was named after Arduin of Ivrea, who was the margrave of the March of Ivrea and King of Italy from 1002 to 1014.

##### Features

1. High Performance, Low Power Atmel AVR 8-Bit Microcontroller Family
   * Advanced RISC Architecture
     + 131 Powerful Instructions
     + Most Single Clock Cycle Execution
     + 32 x 8 General Purpose Working Registers
     + Fully Static Operation
     + Up to 20 MIPS Throughput at 20MHz
     + On-chip 2-cycle Multiplier
   * High Endurance Non-volatile Memory Segments
     + 32KBytes of In-System Self-Programmable Flash program
2. Memory

* 1KBytes EEPROM
* 2KBytes Internal SRAM
* Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
* Data Retention: 20 years at 85°C/100 years at 25°C(1)
* Optional Boot Code Section with Independent Lock Bits
* In-System Programming by On-chip Boot Program
* True Read-While-Write Operation
  + Programming Lock for Software Security
* Atmel® QTouch® Library Support
  + Capacitive Touch Buttons, Sliders and Wheels
  + QTouch and QMatrix® Acquisition
  + Up to 64 sense channels

1. Atmel-42735B-ATmega328/P\_Datasheet\_Complete-11/2016
   * Peripheral Features
     + Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
     + One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
     + Real Time Counter with Separate Oscillator
     + Six PWM Channels
     + 8-channel 10-bit ADC in TQFP and QFN/MLF package
   * Temperature Measurement
     + 6-channel 10-bit ADC in PDIP Package
   * Temperature Measurement
     + Two Master/Slave SPI Serial Interface
     + One Programmable Serial USART
     + One Byte-oriented 2-wire Serial Interface (Philips I2C compatible)
     + Programmable Watchdog Timer with Separate On-chip Oscillator
     + One On-chip Analog Comparator
     + Interrupt and Wake-up on Pin Change
   * Special Microcontroller Features
     + Power-on Reset and Programmable Brown-out Detection
     + Internal Calibrated Oscillator
     + External and Internal Interrupt Sources
     + Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power- down, Standby, and
2. Extended Standby
   * I/O and Packages
     + 23 Programmable I/O Lines
     + 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF
   * Operating Voltage:
     + 1.8 - 5.5V
   * Temperature Range:
     + -40°C to 105°C
   * Speed Grade:
     + 0 - 4MHz @ 1.8 - 5.5V
     + 0 - 10MHz @ 2.7 - 5.5V
     + 0 - 20MHz @ 4.5 - 5.5V
   * Power Consumption at 1MHz, 1.8V, 25°C
     + Active Mode: 0.2mA
     + Power-down Mode: 0.1μA
     + Power-save Mode: 0.75μA (Including 32kHz RTC)

##### HARDWARE COMPONENTS

ARDUINO NANO BLUETOOTH MODULE LCD I2C

RELAY 4 – CHANNEL JUMPER WIRES

DC FAN BULBS MOTOR

POWER SOCKET

**EXPLANATION ABOUT COMPONENTS**

## ARDUINO UNO

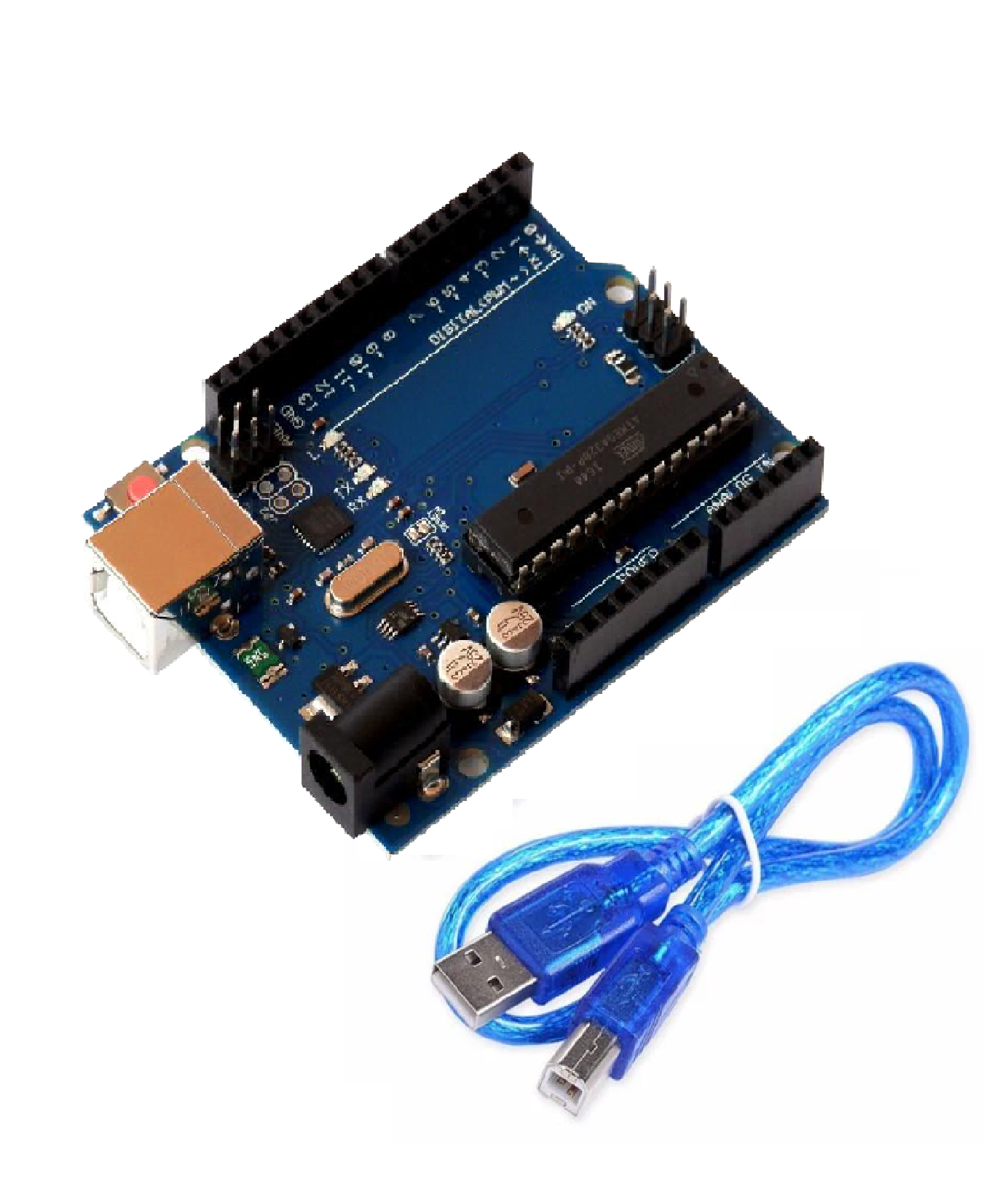


Fig 4 ARDUINO UNO

Power (USB / Barrel Jack)

Every Arduino board needs a way to be connected to a power source. The Arduino UNO can be powered from a USB cable coming from your computer or a wall power supply (like this) that is terminated in a barrel jack. In the picture above the USB connection is labeled **(1)** and the barrel jack is labeled **(2)**.

The USB connection is also how you will load code onto your Arduino board. More on how to program with Arduino can be found in our Installing and Programming Arduino tutorial.

**NOTE:** Do NOT use a power supply greater than 20 Volts as you will overpower (and thereby destroy) yourArduino. The recommended voltage for most Arduino models is between 6 and 12 Volts.

Pins (5V, 3.3V, GND, Analog, Digital, PWM, AREF)

The pins on your Arduino are the places where you connect wires to construct a circuit (probably in conjuction with a breadboard and some wire. They usually have black plastic ‘headers’ that allow you to just plug a wire right into the board. The Arduino has several different kinds of pins, each of which is labeled on the board and used for different functions.

* **GND (3)**:Short for ‘Ground’. There are several GND pins on the Arduino, any of which can be used to ground your circuit.
* **5V (4) & 3.3V (5)**: As you might guess, the 5V pin supplies 5 volts of power, and the 3.3V pin supplies 3.3 volts of power. Most of the simple components used with the Arduino run happily off of 5 or 3.3 volts.
* **Analog (6)**: The area of pins under the ‘Analog In’ label (A0 through A5 on the UNO) areAnalogIn pins. These pins can read the signal from an analog sensor (like a temperature sensor) and convert it into a digital value that we can read.
* **Digital (7)**: Across from the analog pins are the digital pins (0 through 13 on the UNO). These pins can be used for both digital input (like telling if a button is pushed) and digital output (like powering an LED).
* **PWM (8)**: You may have noticed the tilde (~) next to some of the digital pins (3, 5, 6, 9, 10, and 11 on the UNO). These pins act as normal digital pins, but can also be used for something called Pulse-Width Modulation (PWM). We have a tutorial on PWM, but for now, think of these pins as being able to simulate analog output (like fading an LED in and out).
* **AREF (9)**: Stands for Analog Reference. Most of the time you can leave this pin alone. It is sometimes used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

**Reset Button**

Just like the original Nintendo, the Arduino has a reset button **(10)**. Pushing it will temporarily connect the reset pin to ground and restart any code that is loaded on the Arduino. This can be very useful if your code doesn’t repeat, but you want to test it multiple times. Unlike the original Nintendo however, blowing on the Arduino doesn’t usually fix any problems.

**Power LED Indicator**

Just beneath and to the right of the word “UNO” on your circuit board, there’s a tiny LED next to the word ‘ON’ **(11)**. This LED should light up whenever you plug your Arduino into a power source. If this light doesn’t turn on, there’s a good chance something is wrong. Time to re-check your circuit!

**TX RX LEDs**

TX is short for transmit, RX is short for receive. These markings appear quite a bit in electronics to indicate the pins responsible for serial communication. In our case, there are two places on the Arduino UNO where TX and RX appear – once by digital pins 0 and 1, and a second time next to the TX and RX indicator LEDs . These LEDs will give us some nice visual indications whenever our Arduino is receiving or transmitting data (like when we’re loading a new program onto the board).

**Main IC**

The black thing with all the metal legs is an IC, or Integrated Circuit . Think of it as the brains of our Arduino. The main IC on the Arduino is slightly different from board type to board type, but is usually from the ATmega line of IC’s from the ATMEL company. This can be important, as you may need to know the IC type (along with your board type) before loading up a new program from the Arduino software. This information can usually be found in writing on the top side of the IC. If you want to know more about the difference between various IC’s, reading the datasheets is often a good idea.

**Voltage Regulator**

The voltage regulator is not actually something you can (or should) interact with on the Arduino. But it is potentially useful to know that it is there and what it’s for. The voltage regulator does exactly what it says – it controls the amount of voltage that is let into the Arduino board. Think of it as a kind of gatekeeper; it will turn away an extra voltage that might harm the circuit. Of course, it has its limits, so don’t hook up your Arduino to anything greater than 20 volts.

## ESP8266 Module

ESP-01 WiFi module is developed by Ai-thinker Team. core processor ESP8266 in smaller sizes of the module encapsulates Tensilica L106 integrates inddustry-leading ultra low power 32-bit MCU micro, with the 16-bit short mode, Clock speed support 80 MHz, 160 MHz, supports the RTOS, integrated Wi-Fi MAC/BB/RF/PA/LLNA, on-board antenna.The module supports standard IEEE802.11 b/g/n agreement, complete TCP/IP protocol **s**tack. Users can use the add modules to an existing device networrking, or building a separate network controller.ESP8266 is high integration wireless SOCs, designed for space and power constrained mobile platform designers. It provides unsurpassed ability to embed Wi-Fi capabilities within other systems, or to function as a standalone application, with the lowest cost, and minnimal space requirement.

ESP8266EX offers a complete and self-contained Wi-Fi networking solution; it can be used to host the application or to offload Wi-Fi networking functions from another application processor When ESP8266EX hosts the application, it boots up directly from an external flash. In has integrated cache to improve the performance of the system in such applications. Alternately, serving as a Wi-Fi adapter, wireless internet access can be added to any micro controller based design with simple connectivity (SPI/SDIO or I2C/UART interface).

ESP8266EX also integrates an enhanced version of Tensilica’s L106 Diamond series 32-bit processor, with on-chip SRAM, besides the Wi-Fi functionalities. ESP8266EX is often integrated with external sensors annd other application specific devices through its GPIOs; codes for such applications are provided in examples in the SDK.

Espressif Systems’ Smart Connectivitty Platform (ESCP) demonstrates sophisticated system**-**level features include fast sleep/wake context switching for energy-efficient VoIP, adaptive radio biasing. for low-power operation, advance signal processing, and spur cancellation and radio co-existence features for common cellular, Bluetooth, DDR, LVDS, LCD interference mitigation.

### Features

* 802.11 b/g/n
* Integrated low power 32-bit MCU
* Integrated 10-bit ADC
* Integrated TCP/IP protocol stack
* Integrated TR switch, balun, LNA, power amplifier and matching network
* Integrated PLL, regulators, and power management units
* Supports antenna diversity
* Wi-Fi 2.4 GHz, support WPA/WPA2
* Support STA/AP/STA+AP operation modes
* Support Smart Link Function for both Android and iOS devices
* Support Smart Link Function for both Android and iOS devices
* SDIO 2.0, (H) SPI, UART, I2C, I2S, IRDA, PWM, GPIO
* STBC, 1x1 MIMO, 2x1 MIMO
* A-MPDU & A-MSDU aggregation and 0.4s guard interval
* Deep sleep power <10uA, Power down leakage current < 5Ua
* Wake up and transmit packets in < 2ms
* Standby power consumption of < 1.0mW (DTIM3)
* +20dBm output power in 802.11b mode
* Operating temperature range -40C ~ 125C

## LCD2004 Parallel LCD Display with IIC/I2C Interface



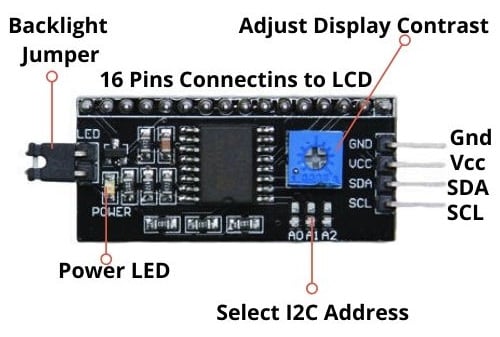
If you want to add some visual output to your Arduino projects, you’ll need a [display](https://robu.in/product-category/led-lcd-and-display-boards/). If you need only a little to display, the LCD2004 Parallel [LCD Display](https://robu.in/product-category/led-lcd-and-display-boards/lcd-display/) with **IIC/I2C interface**is a quite good solution.

This Display provides a simple and cost-effective solution for adding a 20×4 White on RGB Liquid Crystal Display into your project. **The display is 20 character by 4 line display has a very clear and high contrast white text upon a blue background/backlight.**

This is a great blue backlight [LCD display](https://robu.in/product-category/led-lcd-and-display-boards/lcd-display/). It is fantastic for Arduino-based projects. This Display with Blue Backlight is very easy to interface with [Arduino](https://robu.in/product-category/arduino-2/arduino/) or [Other Microcontrollers.](https://robu.in/product-category/development-board/)

**This display overcomes the drawback of**[**LCD2004 Parallel LCD Display**](https://robu.in/product/standard-lcd-20x4-white-on-blue/)**in which you’ll waste about 8 Pins on your**[**Arduino**](https://robu.in/product-category/arduino-2/arduino/)**for the display to get working. Luckily in this product, an**[**I2C adapter**](https://robu.in/product/iici2c-serial-interface-adapter-module/)**is directly soldered right onto the pins of the**[**display**](https://robu.in/product-category/led-lcd-and-display-boards/)**. So all you need to connect are the I2C pins, which show a good library and little of coding.**

The [I2C](https://robu.in/product/iici2c-serial-interface-adapter-module/) is a type of serial bus developed by Philips, which uses two bidirectional lines, called SDA (Serial Data Line) and SCL (Serial Clock Line). Both must be connected via pulled-up resistors. The usage voltages are standard as 5V and 3.3V.

[](https://robu.in/wp-content/uploads/2017/02/LCD1602-Parallel-LCD-Display-with-IICI2C-interface-3.jpg)

If you already have the [**I2C**](https://robu.in/product/iici2c-serial-interface-adapter-module/)adapter soldered onto the board, like in this product, the wiring is quite easy. You should usually have only four pins to hook up. **VCC**and **GND**of course. This display works with 5 volts. So we go for the 5V pin.

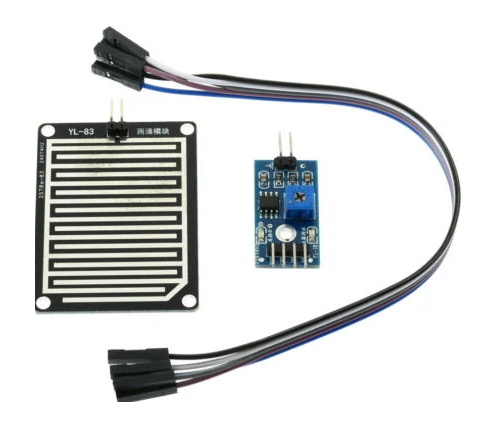
The values shown on the display can be either simple text or numerical values read by the [sensors](https://robu.in/product-category/sensors/), such as temperature or pressure, or even the number of cycles that the [Arduino](https://robu.in/product-category/arduino-2/arduino/) is performing.

## SERVO MOTOR

## 

* TowerPro Servo Motors are optimum-quality and affordable cost servos.! They are suitable for a wide range of applications, including RC aircraft, automobiles, and robotics, Or just to have some fun with whatever crazy project you’re working on.
* When you purchase TowerPro motors in India, they are almost usually NOT ORIGINAL, and these are no exception… These are NOT ORIGINAL Tower Pro Servos, either. However, they are dead cheap and Serve the purpose. We put them to the test and found them to be of good quality for the price.
* You can buy [E Max Servos](https://robu.in/product-category/motors/servo-motors/emax-servo-motor/) OR [Orange servos](https://robu.in/product-category/motors/servo-motors/orange-servo-motors/) if you’re seeking high-quality servos.
* Wire Description:
* RED – Positive
* Brown – Negative
* Orange – Signal
* We have imported this copy of original TowerPro SG90 9g Mini Servo from our trusted supplier. This good quality servo motor is in very close competition to the original TowerPro SG90 1.2kgCm 180-degree servo motor.

**RAINDROPS DETECTION SENSOR**

****

* The Raindrops Detection sensor module is used for rain detection. It is also for measuring rainfall intensity. Rain sensor can be used for all kinds of weather monitoring and translated into output signals and AO.
* Raindrops Detection Sensor Module Rain Weather Module for [Arduino](https://robu.in/product-category/arduino-2/arduino/), etc. Rain sensor can be used to monitor a variety of weather conditions and turned into several fixed output signal and Analog output.
* It includes a printed circuit board (control board) that “collects” the raindrops. As raindrops are collected on the circuit board, they create paths of parallel resistance that are measured via the op-amp. The lower the resistance (or the more water), the lower the voltage output. Conversely, the less water, the greater the output voltage on the analog pin. A completely dry board, for example, will cause the module to output 5V.
* The module includes a rain board and a control board that is separate for more convenience. It has a power indicator LED and an adjustable sensitivity through a potentiometer. The module is based on the LM393 op-amp.

**Heart Beat Pulse Sensor**



Heart Rate data can be used in many Electronic design and microcontroller projects. But the heart rate data is difficult to read, however, the Pulse Sensor Amped help us to read heart rate. The Heart Beat Pulse Sensor Amped is a plug-and-play heart-rate sensor for [Arduino](https://robu.in/product-category/arduino-2/arduino/).

It can be used by students, artists, athletes, makers, and game & mobile developers who want to easily incorporate live heart-rate data into their projects.

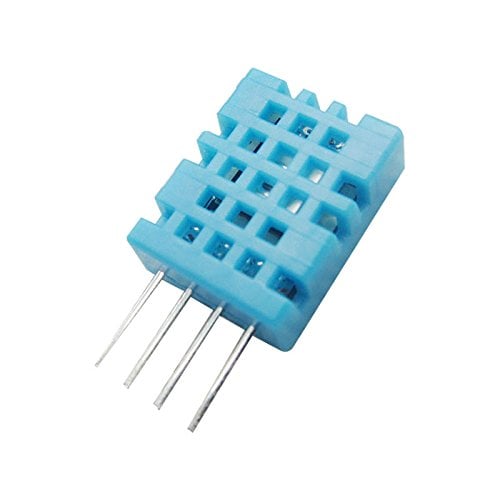
It essentially combines a simple optical [heart rate sensor](https://robu.in/product-tag/heart-rate-monitor/) with amplification and noise cancellation circuitry making it fast and easy to get reliable pulse readings.

Pulse Sensor Amped adds amplification and noise cancellation circuitry to the hardware. It’s noticeably faster and easier to get reliable pulse readings. Pulse Sensor Amped works with either a 3V or 5V Arduino.

Pulse Sensor Arduino Hookup

[](https://robu.in/wp-content/uploads/2016/06/arduino.jpg)

**DHT 11 Sensor**



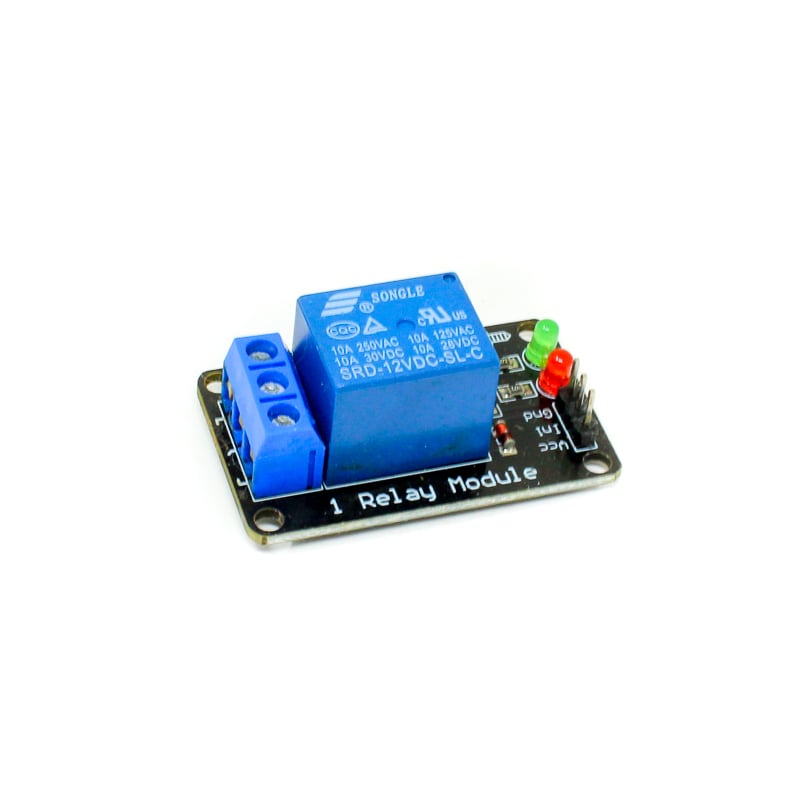
The DHT-11 Digital Temperature And Humidity Sensor is a basic, ultra low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air and spits out a digital signal on the data pin (no analog input pins needed).

Its fairly simple to use, but requires careful timing to grab data. The only real downside of this sensor is you can only get new data from it once every 2 seconds, so in your code please use sensor reading interval at 2 seconds or more. Compared to the [DHT22](https://robu.in/product/dht22-digital-temperature-humidity-sensor-temperature-humidity-module-am2302/), this sensor is less precise, less accurate and works in a smaller range of temperature/humidity.

But despite its disadvantages over [DHT22](https://robu.in/product/dht22-digital-temperature-humidity-sensor-temperature-humidity-module-am2302/), it is smaller and less expensive sensor for temperature and humidity measurement.

Note: This package does not include Arduino board, you have to buy it separately.

**RELAY MODULE**

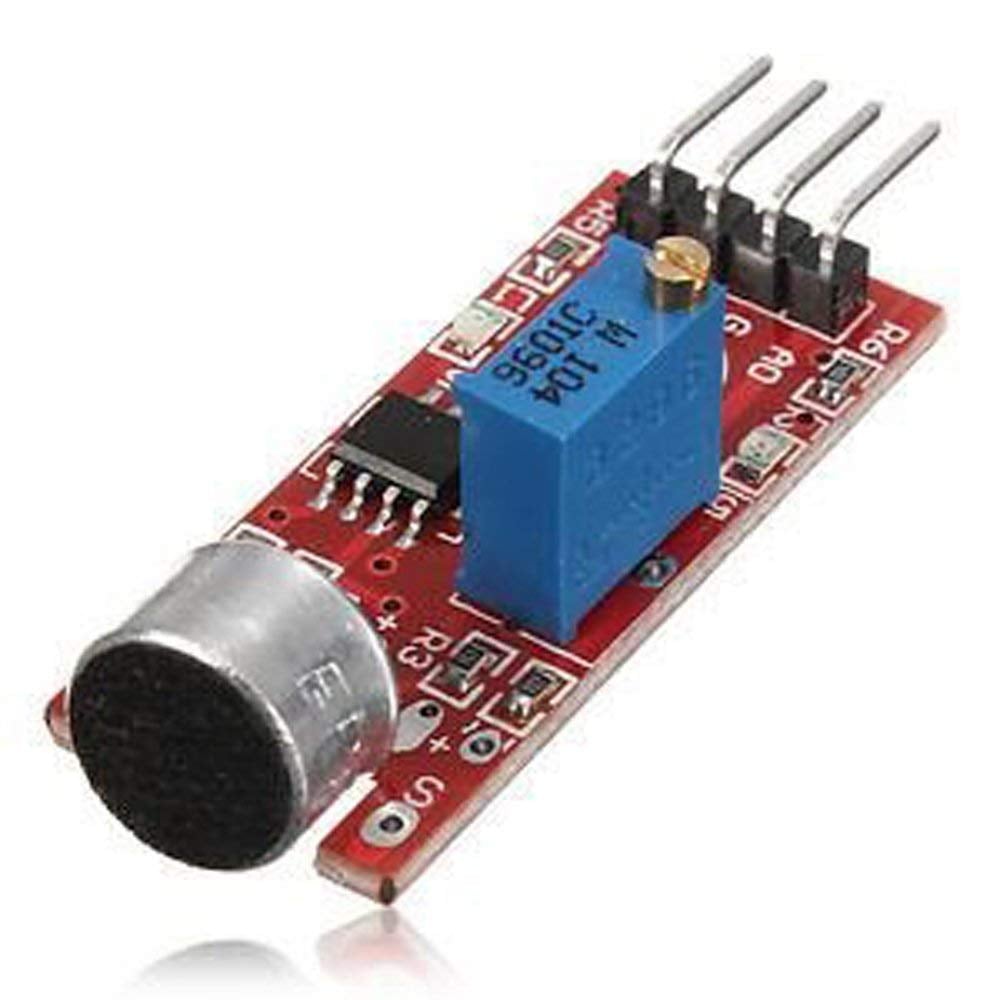


V 1 Channel Relay Module is a small and easy to use . Use it to control one 240V power appliance directly from Arduino, Raspberry Pi, and other microcontrollers or low voltage circuits. Perfect for switching 240V appliances – lights, fans, etc, and even high power motors at lower voltages.

The board uses a high-quality relay, which can handle a maximum of 10A/250 V AC or 15A/125V AC. Each relay has all three connections – Common, Normally Open, Normally Closed brought out to 3 pin screw terminals which make it easy to make and remove connections. The board has a power indication and a relay status LED to ease debugging. The board can accept control inputs within a wide range of voltages from 4V to 12V.

Power input and relay control signals are brought to header pins on the board. Hence, the board can be easily interfaced with our development boards using our female to female jumper wires.

**Sound Detection Sensor**



Sound Detection Module Sensor for Intelligent Vehicle Compatible With Arduino is a Single channel signal output Sensor. The output is effective to the low-level sound signal with good fidelity, When there is sound, outputs low level and signal light.

It can be used for Acoustic control light, give sound and light alarm working with the Photosensitive sensor, and sound control, sound detect. (The Maximum induction distance is 0.5M).

**SOFTWARES:**

This tutorial will walk you through downloading, installing, and testing the [Arduino software](http://arduino.cc/en/Main/Software) (also known as the Arduino IDE - short for Integrated Development Environment). Before you jump to the page for your operating system, make sure you've got all the right equipment.



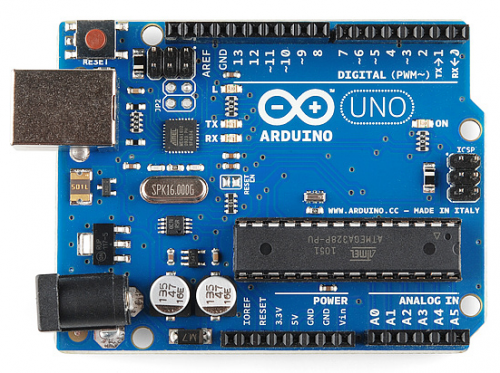
##### Fig 4.10 Arduino Logo

Required Materials

 A computer (Windows, Mac, or Linux)

 An Arduino-compatible microcontroller (anything from [this guide](https://www.sparkfun.com/standard_arduino_comparison_guide) should work)

 A USB A-to-B cable, or another appropriate way to connect your Arduino-compatible microcontroller to your computer (check out this [USB buying guide](https://www.sparkfun.com/pages/USB_Guide) if you're not sure which cable to get).

****

**Fig 10An Arduino Uno**

****

**Fig 11 An A-to-B USB Cable**

[How to Install CH340 Drivers](https://learn.sparkfun.com/tutorials/how-to-install-ch340-drivers) [AUGUST 6, 2019](https://learn.sparkfun.com/tutorials/how-to-install-ch340-drivers)

How to install CH340 drivers (if you need them) on Windows, Mac OS X, and Linux.

If you're ready to get started, click on the link in the column on the left that matches up with your operating system, or you can jump to your operating system here.

 [Windows](https://learn.sparkfun.com/tutorials/installing-arduino/windows)  [Mac](https://learn.sparkfun.com/tutorials/installing-arduino/mac)

 [Linux](https://learn.sparkfun.com/tutorials/installing-arduino/linux)

 Windows

This page will show you how to install and test the Arduino software with a Windows operating system (Windows 8, Windows 7, Vista, and XP).

 Go to the Arduino [download page](http://arduino.cc/en/Main/Software) and download the latest version of the Arduino software for Windows.

 When the download is finished, un-zip it and open up the Arduino folder to confirm that yes, there are indeed some files and sub-folders inside. The file structure is important so don't be moving any files around unless you really know what you're doing.

 Power up your Arduino by connecting your Arduino board to your computer with a USB cable (or FTDI connector if you're using an Arduino pro). You should see the an LED labed 'ON' light up. ([this diagram](https://learn.sparkfun.com/tutorials/what-is-an-arduino/whats-on-the-board) shows the placement of the power LED on the NANO).

Drivers for Arduino Nano on Windows

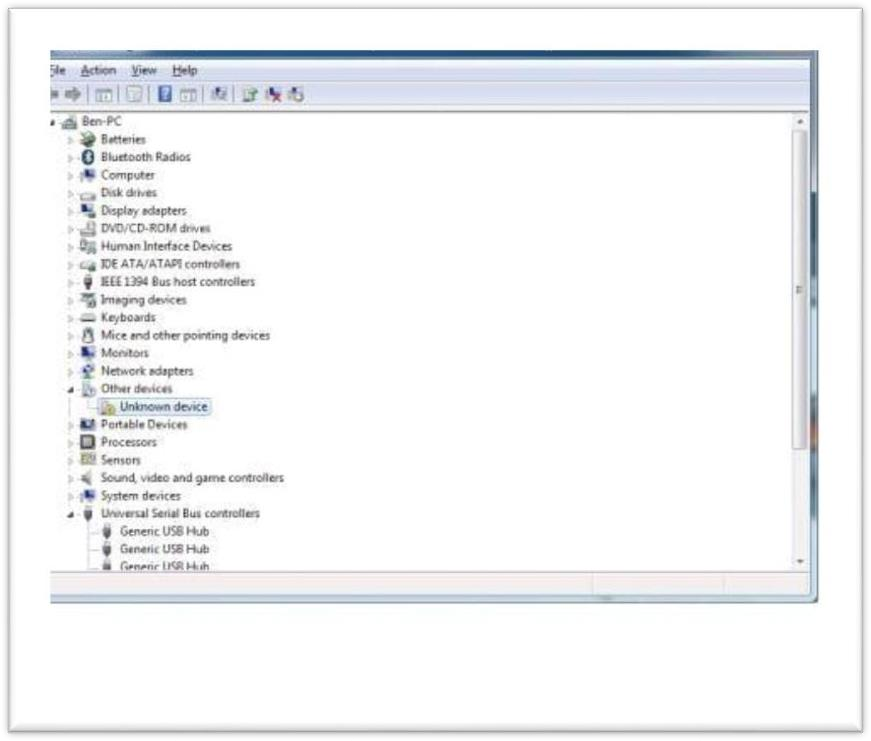
Installing the Drivers for the Arduino Nano (from Arduino.cc)

 Plug in your board and wait for Windows to begin it's driver installation process  After a few moments, the process will fail, despite its best efforts

 Click on the Start Menu, and open up the Control Panel

 While in the Control Panel, navigate to System and Security. Next, click on System  Once the System window is up, open the Device Manager

 Look under Ports (COM & LPT). You should see an open port named "Arduino NANO (COMxx)". If there is no COM & LPT section, look under 'Other Devices' for 'Unknown



Device'

##### Fig 4.13: device manager

 Right click on the "Arduino NANO (COMxx)" or "Unknown Device" port and choose the "Update Driver Software" option

 Next, choose the "Browse my computer for Driver software" option



##### Fig 4.12update driver software

 Finally, navigate to and select the Nano's driver file, named "ArduinoNANO.inf", located in the "Drivers" folder of the Arduino Software download (not the "FTDI USB Drivers" sub- directory). If you cannot see the .inf file, it is probably just hidden. You can select the 'drivers' folder with the 'search sub-folders' option selected instead.

 Windows will finish up the driver installation from there

For earlier versions of the Arduino boards (e.g. ArduinoDuemilanove, Nano, or Diecimila) check out [this page](http://arduino.cc/en/Guide/Windows) for specific directions.

Drivers for RedBoard on Windows

If you are using a RedBoard programmed for Arduino, please go to [How to Install FTDI Drivers,](https://learn.sparkfun.com/tutorials/usb-serial-driver-quick-install-) for specific instructions on how to install the drivers.

[USB Serial Driver Quick Install](https://learn.sparkfun.com/tutorials/usb-serial-driver-quick-install-)

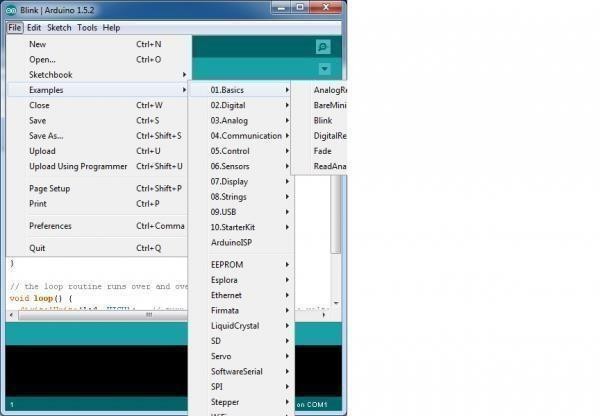
How to install USB serial drivers on Windows, MacOS , and Linux. Launch and Blink!

After following the appropriate steps for your software install, we are now ready to test your first program with your Arduino board!

 Launch the Arduino application

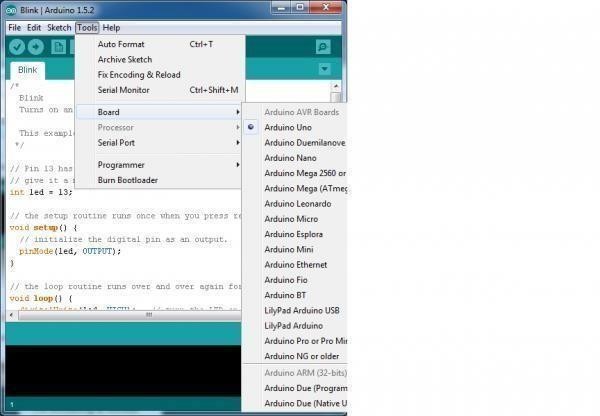
 If you disconnected your board, plug it back in

 Open the Blink example sketch by going to: File > Examples > 1.Basics > Blink

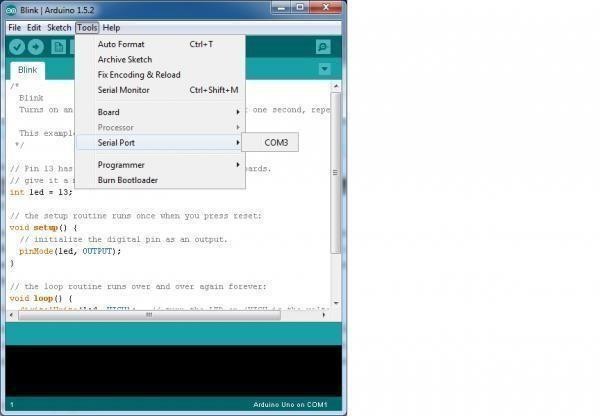


##### Fig 4.14 : examples

 Select the type of Arduino board you're using: Tools > Board > your board type



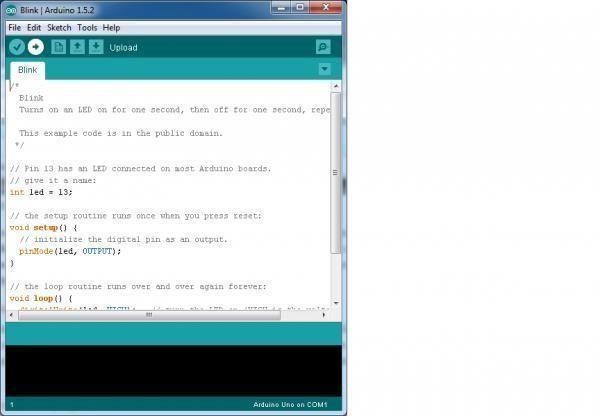
##### Fig 4.15 selection of board

 Select the serial/COM port that your Arduino is attached to: Tools > Port >COMxx

##### Fig 4.16 selection of serial port

 If you're not sure which serial device is your Arduino, take a look at the available ports, then unplug your Arduino and look again. The one that disappeared is your Arduino.

 With your Arduino board connected, and the Blink sketch open, press the 'Upload' button



##### Fig 4.17 example code

 After a second, you should see some LEDs flashing on your Arduino, followed by the message 'Done Uploading' in the status bar of the Blink sketch.

 If everything worked, the onboard LED on your Arduino should now be blinking! You just programmed your first Arduino!

Troubleshooting

[This guide](http://arduino.cc/en/Guide/Windows) from Arduino has some more details and troubleshooting tips if you get stuck.

 Mac

This page will show you how to install and test the Arduino software on a Mac computer running OSX.

 Go to the Arduino [download page](http://arduino.cc/en/Main/Software) and download the latest version of the Arduino software for Mac.

 When the download is finished, un-zip it and open up the Arduino folder to confirm that yes, there are indeed some files and sub-folders inside. The file structure is important so don't be moving any files around unless you really know what you're doing.

 Power up your Arduino by connecting your Arduino board to your computer with a USB cable (or FTDI connector if you're using an Arduino pro). You should see the an LED labed 'ON' light up. ([this diagram](https://learn.sparkfun.com/tutorials/what-is-an-arduino/whats-on-the-board) shows the placement of the power LED on the NANO).

* Move the Arduino application into your Applications folder.

FTDI Drivers

If you have an NANO, Mega2560, or Redboard, you shouldn't need this step, so skip it!

 For other boards, you will need to install drivers for the FTDI chip on your Arduino.  Go to the [FTDI website](http://www.ftdichip.com/Drivers/VCP.htm) and download the latest version of the drivers.

 Once you're done downloading, double click the package and follow the instructions from the installer.

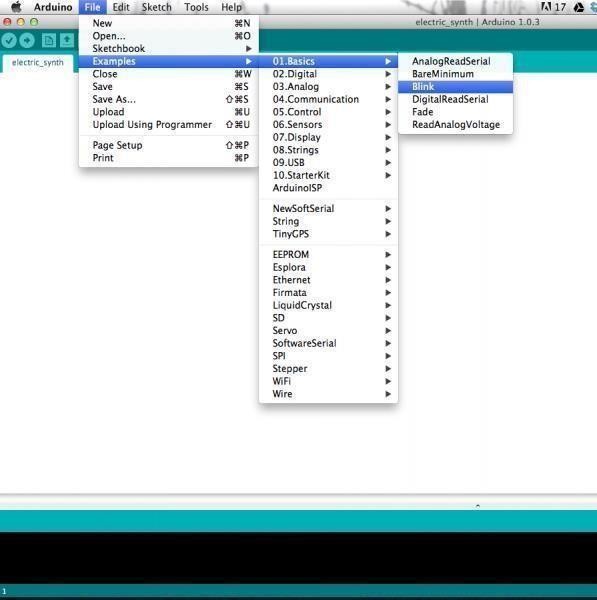
 Restart your computer after installing the drivers.

Launch and Blink!

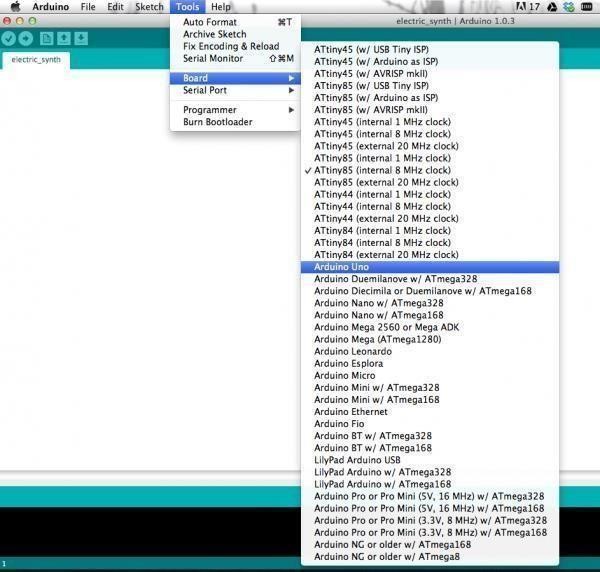
After following the appropriate steps for your software install, we are now ready to test your first program with your Arduino board!

 Launch the Arduino application

 If you disconnected your board, plug it back in



 Open the Blink example sketch by going to: File > Examples > 1.Basics > Blink



 Select the type of Arduino board you're using: Tools > Board > your board type

Fig 13 board selection in MAC

 Select the serial port that your Arduino is attached to: Tools > Port >xxxxxx (it'll probably look something like "/dev/tty.usbmodemfd131" or "/dev/tty.usbserial-131" but probably with a different number)

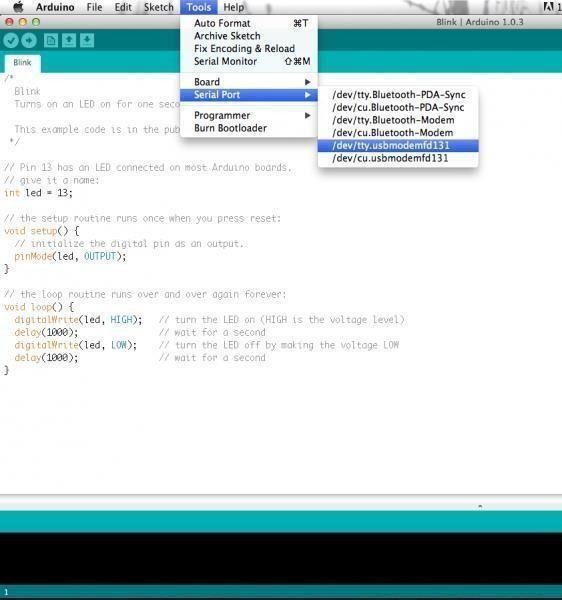


Fig 14 board selection in MAC

 If you're not sure which serial device is your Arduino, take a look at the available ports, then unplug your Arduino and look again. The one that disappeared is your Arduino.

 With your Arduino board connected and the Blink sketch open, press the 'Upload' button

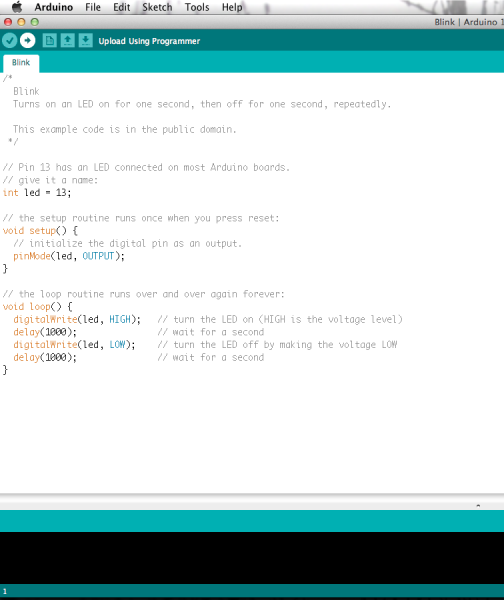


Fig 15 example code

 After a second, you should see some LEDs flashing on your Arduino, followed by the message 'Done Uploading' in the status bar of the Blink sketch.

 If everything worked, the onboard LED on your Arduino should now be blinking! You just programmed your first Arduino!

Troubleshooting

If you're having problems, check out [this troubleshooting guide](http://arduino.cc/en/Guide/Troubleshooting) from Arduino.

 Linux

If you are a Linux user, you probably know that there are many different distribution 'flavors' of Linux out there. Unsurprisingly, installing Arduino is slightly different for many of these distributions.

Luckily, the Arduino community has done an excellent job of providing instructions for most of the popular versions. Click on the link below that covers your flavor of Linux:

 [ArchLinux](http://playground.arduino.cc/Linux/ArchLinux)  [Debian](http://playground.arduino.cc/Linux/Debian)

 [Fedora](http://playground.arduino.cc/Linux/Fedora)



[Gentoo](http://playground.arduino.cc/Linux/Gentoo)

 [MEPIS](http://playground.arduino.cc/Linux/MEPIS)

 [Mint](http://playground.arduino.cc/Linux/Mint)

 [openSUSE](http://playground.arduino.cc/Linux/OpenSUSE)  [Puppy](http://playground.arduino.cc/Linux/Puppy)

 [Pussy](http://playground.arduino.cc/Linux/Pussy)

 [Slackware](http://playground.arduino.cc/Linux/Slackware)  [Ubuntu](http://playground.arduino.cc/Linux/Ubuntu)

 [Xandros (Debian derivative) on Asus Eee PC](http://playground.arduino.cc/Linux/Xandros)

If the above directions did not work for you, or you don't see your distribution, try this [catch-all guide.](http://playground.arduino.cc/Linux/All)

You can go to the [download page](http://arduino.cc/en/Main/Software) and download the latest version of Arduino for Linux (there are 32- bit and 64-bit versions available) when your system is properly set up.

Launch and Blink!

After following the appropriate steps for your software install, we are now ready to test your first program with your Arduino board!

 Launch the Arduino application

 If you disconnected your board, plug it back in

 Open the Blink example sketch by going to: File > Examples > 1.Basics > Blink  Select the type of Arduino board you're using: Tools > Board > your board type

 Select the serial port that your Arduino is attached to: Tools > Port >xxxxxx (it'll probably look something like "/dev/tty.usbmodemfd131" or "/dev/tty.usbserial-131" but probably with a different number)

 If you're not sure which serial device is your Arduino, take a look at the available ports, then unplug your Arduino and look again. The one that disappeared is your Arduino.

 With your Arduino board connected and the Blink sketch open, press the 'Upload' button

 After a second, you should see some LEDs flashing on your Arduino, followed by the message 'Done Uploading' in the status bar of the Blink sketch.

 If everything worked, the onboard LED on your Arduino should now be blinking! You just programmed your first Arduino!

Troubleshooting

The [Arduino Playground Linux section](http://playground.arduino.cc/Learning/Linux) is a great resource for figuring out any problems with your Arduino installation.

 Board Add-Ons with Arduino Board Manager

With Arduino v1.6.4+, a new boards manager feature makes it easy to add third-party boards (like the [SparkFunRedboard, Digital Sandbox, and RedBot](https://github.com/sparkfun/Arduino_Boards#sparkfun-arduino-boards)) to the Arduino IDE.

To start, highlight and copy (CTRL + C / CMD + C) the text below for the boards manager URL. You'll need this to configure Arduino.

COPY

CODEhttps://raw.githubusercontent.com/sparkfun/Arduino\_Boards/master/IDE\_Board\_Manager/pac kage\_sparkfun\_index.json

Open up Arduino:

 Configure the Boards Manager

* For Windows and Linux, head to File>Preferences>Additional Boards Manager URLs and paste (CTRL + V / CMD + V) the link
* For Macs, head to Arduino>Preferences>Additional Boards Manager URLs and paste (CTRL + V / CMD + V) the link

 Click on Tools>Board>Boards Manager...

 Select the Type as "Contributed" from the drop down menu.  Click on the SparkFun AVR Boards and then click Install

That's it! Boards are all installed. This also gives you access to all of our library files as well through the built-in Library Manager tool in Arduino. Looking for more information about adding other customboards? Check out the the following [tutorial to install other Arduino cores.](https://learn.sparkfun.com/tutorials/installing-board-definitions-in-the-arduino-ide)

# CHAPTER 5: RESULTS

**Fig 5.1) Front View Of The Home Automation**

**Fig 5.12) Side View Of Home Automation**

# CHAPTER 6: CONCLUISON

The development of an automated cradle system using an Arduino Uno, various environmental sensors, and an ESP-01 module for WiFi connectivity demonstrates the potential of integrating smart technology into everyday applications. This project successfully showcases how such a system can enhance comfort and safety by dynamically responding to temperature, humidity, noise, and rain conditions. The DHT sensor, sound sensor, and rain sensor provide critical data that the Arduino processes to control the cradle's movements, ensuring a responsive and adaptive environment.

The inclusion of the ESP-01 module for WiFi connectivity further extends the system's functionality, allowing users to remotely monitor and control the cradle via a web interface or mobile app. This feature not only adds convenience but also ensures that optimal conditions can be maintained even when the user is not physically present, highlighting the advantages of IoT in home automation.

By leveraging the versatility and ease of use of the Arduino Uno, this project demonstrates how various sensors and modules can be effectively integrated to create a sophisticated and user-friendly automated system. The project's success underscores the potential for future developments in smart home technology, where similar principles can be applied to a wide range of applications to improve daily living.

**CHAPTER 7** : **REFERENCES**

## REFERENCES

[1] W. A. Jabbar, H. K. Shang, S. N. I. S. Hamid, A. A. Almohammedi, R. M. Ramli, and M. A. H. Ali, “IoT-BBMS: Internet of Things-Based Baby Monitoring System for Smart Cradle,” IEEE Access, vol. 7, pp. 93791–93805, 2019. [2] S. Maloji, S. Malakonda Sai Lokesh, K. Nikhil Sai, M. Vasavi Prasanna, M. K. Ashwaq, and S. Arunmetha, “An innovative approach for infant monitoring system using movel s.Odi based iot system,” Int. J. Adv. Sci. Technol., vol. 29, no. 6, pp. 3623–3630, 2020. [3] K. Jose Reena and R. Parameswari, “A Smart Health Care Monitor System in IoT Based Human Activities of Daily Living: A Review,” Proc. Int. Conf. Mach. Learn. Big Data, Cloud Parallel Comput. Trends, Prespectives Prospect. Com. 2019, pp. 446–448, 2019, doi: 10.1109/COMITCon.2019.8862439. [4] S. Ananth, P. Sathya, and P. Madhan Mohan, “Smart health monitoring system through IoT,” Proc. 2019 IEEE Int. Conf. Commun. Signal Process. ICCSP 2019, pp. 968–970, 2019, doi: 10.1109/ICCSP.2019.8697921. [5] M. V. Narayana, K. Dusarlapudi, K. Uday Kiran, and B. Sakthi Kumar, “IoT based real time neonate monitoring system using arduino,” J. Adv. Res. Dyn. Control Syst., vol. 9, no. Special issue 14, pp. 1764– 1772, 2017. [6] H. Singh et al., “Neo-Bedside Monitoring Device for Integrated Neonatal Intensive Care Unit (iNICU),” IEEE Access, vol. 7, no. c, pp. 7803–7813, 2019, doi: 10.1109/ACCESS.2018.2886879. [7] M. Koli, P. Ladge, B. Prasad, R. Boria, and N. J. Balur, “Intelligent Baby Incubator,” Proc. 2nd Int. Conf. Electron. Commun. Aerosp. Technol. ICECA 2018, no. Iceca, pp. 1036–1042, 2018, doi: 10.1109/ICECA.2018.8474763. [8] M. Kumar, “TEMPERATURE CONTROL AND MONITORING,” vol. 22, no. 1, 2016. [9] H. Patke, M. Borkar, N. Kenkre, and A. Gupta, “An innovative approach for infant monitoring system using pulse rate and oxygen level,” in 2017 International Conference on Intelligent Computing and Control Systems (ICICCS), 2017, pp. 1054–1058. [10] Z. U. Ahmed, M. G. Mortuza, M. J. Uddin, M. H. Kabir, M. Mahiuddin, and M. J. Hoque, “Internet of Things Based Patient Health Monitoring System Using Wearable Biomedical Device,” 2018 Int. Conf. Innov. Eng. Technol. ICIET 2018, no. December, pp. 1–5, 2019, doi: 10.1109/CIET.2018.8660846. [11] K. V. Sowmya and J. K. R. Sastry, “Performance evaluation of IOT systems - basic issues,” Int. J. Eng. Technol., vol. 7, no. 2, pp. 131– 137, 2018, doi: 10.14419/ijet.v7i2.7.10279. [12] P. Gopi Krishna, K. Sreenivasa Ravi, K. Hari Kishore, K. Krishna Veni, K. N. Siva Rao, and R. D. Prasad, “Design and development of bi-directional IoT gateway using ZigBee and Wi-Fi technologies with MQTT protocol,” Int. J. Eng. Technol., vol. 7, no. 2, pp. 125–129, 2018, doi: 10.14419/ijet.v7i2.8.10344.