HEALTH MONITORING SYSTEM FOR CATTLE USING AI

A PROJECT REPORT

Submitted by

JAGADESH.S	**211419105049**
ARUNPRAKASH.K	**211419105015**
KAVIMARAN.P	**211419105066**
DHANASEKERAN.M	**211419105039**

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

ELECTRICAL AND ELECTRONICS ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

APRIL 2023

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report " **HEALTH MONITORING SYSTEM FOR CATTLE USING AI** " is the bonafide work of "**S.JAGADESH (211419105049) , K.ARUNPRAKASH (211419105015) , P.KAVIMARAN (211419105066) , M.DHANASEKERAN (211419105039)**" who carried out the project work under my supervision.

Dr. S. SELVI M.E Ph.D,	**Mr. R. SAKTHIVEL M.E.,**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
PROFESSOR	**ASSISTANT PROFESSOR**
Department of Electrical and	Department of Electrical and
Electronics Engineering,	Electronics Engineering,
Panimalar Engineering College,	Panimalar Engineering College,
Chennai – 123.	Chennai – 123.

Submitted for End Semester Project Viva Voce held on ______________ at Panimalar Engineering College, Chennai.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

Our sincere thanks to our Honourable Founder and Chairman, **Dr.JEPPIAAR,M.A.,B.L.,Ph.D.,** for his sincere endeavour in educating us in his premier institution.

We would like to express our deep gratitude to our beloved **Secretary and Correspondent, Dr.P.CHINNADURAI, M.A.,M.Phil.,Ph.D** for his enthusiastic motivation which inspired us a lot in completing this project and our sincere thanks to our Directors **Mrs.C.VIJAYA RAJESWARI, Dr.C.SAKTHI KUMAR, M.E.,Ph.D** and **Dr.SARANYASREE SAKTHIKUMAR,B.E,M.B.A,Ph.D** for providing us with the necessary facilities for the completion of this project.

We would like to express thanks to our Principal, **Dr. K. Mani M.E., Ph.D.,** for having extended his guidance and cooperation.

We would also like to thank our **Head of the Department**, **Dr.S. Selvi, M.E., Ph.D., Professor and Head, Department of Electrical and Electronics Enginering** for her encouragement.

Personally, we thank our Guide **Mr. R. SAKTHIVEL M.E., in Department of Electrical and Electronics engineering** for the persistent motivation and support for this project, who at all times was the mentor of germination of the project from a small idea.

We express our sincere thanks to the project coordinators **Dr.S.Deepa & Dr.N.MANOJ KUMAR,M.E.,Ph.D., in Department of Electrical and Electronics Engineering** for the Valuable suggestions from time to time at every stage of our project.

Finally, we would like to take this opportunity to thank our family members, faculty and non-teaching staff members of our department, friends, well-wishers who have helped us for the successful completion of our project.

ABSTRACT

This project is designed to help farmers to monitor the health of their cattle by leveraging Artificial Intelligence technology. Because to farm automation and cutting-edge technical procedures, agricultural output has significantly increased in recent years. The use of AI in cow health monitoring systems is the current research focus in farm automation. Several businesses can benefit from a higher level of surveillance thanks to AI. The infrastructure, hardware, software, and representative physiological instruments are all part of the suggested monitoring system. Cattle health is crucial for boosting farm productivity. Large herds of cattle can be found in many dairies. As a result, it is too tough to care for them and regularly check on their well-being. So, the owner of the dairy and regional authorities find this job to be quite adamantine. The image processing algorithm will analyze the features such as color, texture, and shape to distinguish between healthy and diseased cattle. The system will also use temperature sensors to monitor the temperature in the environment and maintain a comfortable temperature for the cattle during rainy days, which can otherwise lead to hypothermia. The system will provide farmers with real-time alerts in case of any potential health issues and suggest appropriate actions to prevent the spread of diseases. The project aims to help farmers to take proactive measures to maintain the health of their cattle and reduce the economic losses due to animal illness. Overall, the proposed project could have significant benefits for the cattle industry by improving the health and well-being of cattle, reducing economic losses due to animal illness, and minimizing the workload and stress of farmers.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATION

AI	Artificial Intelligence
GPIO	General Purpose Input Output
ADC	Analog to Digital Converter
USB	Universal Serial Bus
UART	Universal Asynchronous Receiver-Transmitter
PWM	Pulse Width Modulation
LCD	Liquid Crystal Display
LED	Light Emitting Diode
UVC	USB Video Class
CNN	Convolutional Neural Network
IOT	Internet Of Things
DHT	Digital Humidity and Temperature
IDE	Integrated Development Environment
SPI	Serial Peripheral Interface
TTL	Transistor – Transistor Logic

CHAPTER 1

INTRODUCTION

Agriculture is essential to all progressive and developing nations; in fact, it serves as the foundation of these nations. Forestry, fruit cultivation, dairy, poultry, beekeeping, and other activities are currently recognised as being a part of contemporary agriculture in addition to farming. Due to market pressure and increased demand for milk products, dairy farming has transformed globally in recent years, moving towards larger, more intensive, profit-driven operations. The use of agricultural automation and cutting-edge technology has been leading in the market for such items. Automated animal health monitoring systems are a significant area of farm automation that have been the subject of research over the past 20 years. Cattle health monitoring is a crucial aspect of livestock management as it directly affects the health and productivity of cattle. The traditional methods of monitoring cattle health, such as visual observation, physical examination, and medical treatment, are time-consuming and labour-intensive. Moreover, early detection of diseases is challenging using these conventional methods, leading to high mortality rates and economic losses.

To address these challenges, there has been a growing interest in leveraging advanced technologies such as Artificial Intelligence (AI) to monitor cattle health in real-time. AI systems can analyse large amounts of data, identify patterns, and make predictions that would be challenging for humans to detect. Cattle health monitoring systems that use AI have the potential to revolutionize the livestock industry. These systems use image processing and AI algorithms to compare real-time images of healthy cattle with those of diseased cattle, enabling farmers to identify early signs of disease and take corrective action before it spreads to the entire herd. Additionally, these systems can also maintain the temperature in the cattle room using the count of cattle present.

The primary advantage of using AI for cattle health monitoring is the ability to detect diseases early and reduce the spread of infections, leading to improved herd health and productivity. These systems can help farmers save time and money by reducing labour costs and minimizing the need for medical treatment. Moreover, AI-based cattle health monitoring systems can provide farmers with valuable insights into cattle behaviour and performance, enabling them to make informed decisions about feed and medication administration. The data collected from these systems can also be used to improve breeding programs and genetic selection, leading to the development of more resilient and productive cattle breeds. One of the significant challenges in cattle health monitoring is early detection of diseases. Diseases in cattle can spread rapidly within the herd, leading to significant economic losses. Early detection of diseases is critical to prevent the spread of infections and reduce mortality rates. Traditional methods of disease detection in cattle involve visual observation and physical examination, which can be subjective and time-consuming.

AI-based cattle health monitoring systems can detect diseases in real-time, enabling farmers to take corrective action before the disease spreads. These systems use image processing algorithms to compare real-time images of healthy cattle with those of diseased cattle. By comparing the two sets of images, the system can identify early signs of disease, such as changes in the colour of the skin or eyes, changes in posture, and abnormal behaviour. Another challenge in cattle health monitoring is the maintenance of the ideal temperature in the cattle room. Cattle are sensitive to changes in temperature and humidity, and their productivity and health can be affected if the temperature is not maintained within the ideal range. Traditional methods of temperature monitoring in the cattle room involve manually checking the temperature using thermometers, which can be time-consuming and labour-intensive.

AI-based cattle health monitoring systems can maintain the ideal temperature in the cattle room by using the count of cattle present. The system can adjust the temperature and humidity levels in the room based on the number of cattle present, ensuring that the temperature is maintained within the ideal range. By maintaining the ideal temperature, the system can improve cattle productivity and health, reducing the need for medical treatment and improving the overall health of the herd.AI-based cattle health monitoring systems can also provide farmers with valuable insights into cattle behaviour and performance. The data collected from these systems can be used to monitor cattle behaviour, such as feeding habits, water intake, and sleeping patterns. By monitoring these behaviours, farmers can make informed decisions about feed and medication administration, leading to improved cattle health and productivity.

Moreover, the data collected from AI-based cattle health monitoring systems can be used to improve breeding programs and genetic selection. By analysing the data, farmers can identify patterns and trends in cattle behaviour and performance, enabling them to make informed decisions about breeding programs and genetic selection. This can lead to the development of more resilient and productive.

CHAPTER 2

LITERATURE REVIEW

Cattle health monitoring systems using AI that compare and check real-time cattle pictures with those of diseased cattle and maintain temperature using image processing are emerging as a promising approach to precision livestock farming. In this literature review, we will explore some of the key research and developments in this field.

- "Smart Cattle Farming: A Review on Technology Advancement and Its Prospects" by Muhammad Jawad Khalid et al. (2021)
  - The paper explores the use of technology in cattle farming, including the use of AI-based systems for monitoring cattle health. The authors discuss the potential benefits of such systems, including improved disease detection and prevention and increased efficiency in monitoring large herds.
- "Deep Learning-Based Recognition of Cattle Breeds Using Convolutional Neural Networks" by Lu Zhang et al. (2020)
  - This paper presents a system for cattle breed recognition using deep learning algorithms. The authors use convolutional neural networks to analyze images of cattle and identify the breed. The system could potentially be adapted to identify signs of disease or other health issues in cattle based on their visual appearance.
- "An Intelligent and Automatic Animal Identification System Based on Image Processing and Fuzzy Decision" by Jie Zhang et al. (2019)
  - This paper proposes an automatic identification system for cattle based on image processing and fuzzy decision-making. The authors use cameras to capture images of the cattle, which are then analyzed using an AI-based algorithm to identify

individual animals based on their unique features. The system could potentially be used to track individual animals and monitor their health over time.

- "Real-Time Cattle Health Monitoring System Based on Internet of Things and Deep Learning" by Yuxin Zhang et al. (2021)
    - This paper presents a real-time cattle health monitoring system based on the Internet of Things (IoT) and deep learning algorithms. The system includes sensors to monitor cattle behavior and health, as well as cameras for visual monitoring. The authors use deep learning algorithms to analyze the data collected by the sensors and cameras and identify signs of disease or other health issues in the cattle. The system could potentially be used to improve the overall health and productivity of cattle herds

Overall, these studies highlight the potential of AI-based cattle health monitoring systems in improving the efficiency and productivity of livestock farming. By providing real-time information on the health and well-being of cattle, such systems can help farmers to detect diseases at an early stage, provide timely treatment, and prevent the spread of diseases. Additionally, such systems can optimize the use of resources such as feed and water, improve animal welfare, and promote sustainable livestock farming practices.

CHAPTER 3

EXISTING SYSTEM

In the existing system If we take into account the dairy farm cattle's health monitoring, earlier years required dairy farms and farmers to employ a unique approach for the identification of animal health-related disorders. This technique required continuous or daily to daily observation, which again required excessive effort. Sometimes a technique like this produces inaccurate results that don't reflect the cattle's actual health status. The health of the cattle may suffer as a result of this..

One of the latest existing systems for cattle health monitoring is the use of wearable sensors to track the health and behaviour of individual cattle. These sensors can collect a range of data, including the animal's heart rate, body temperature, activity level, and feeding behaviour. This information can be transmitted to a central database where it can be analysed to identify signs of illness or stress.

However, this method has some problems that can be solved using AI technology. Firstly, the data collected by the sensors can be noisy and inconsistent, which can make it difficult to detect subtle changes in the animal's health and behaviour. AI algorithms can be used to pre-process the data, filter out noise and inconsistencies, and identify patterns and trends in the data that may indicate signs of illness or stress.

Secondly, the analysis of the data collected by the sensors is typically performed manually, which can be time-consuming and subjective. AI algorithms can be used to automate the analysis of the data, allowing for faster and more accurate diagnosis of diseases and other health problems. By analysing the data in real-time, AI algorithms can also provide early warning signs of potential health problems, allowing for timely intervention and treatment.

Thirdly, the use of wearable sensors can be limited by the number of sensors that can be attached to the animal's body. This can make it difficult to collect comprehensive data on the animal's health and behaviour. AI algorithms can be used to interpolate missing data points and to infer information about the animal's health and behaviour based on the available data.

Fourthly, the use of wearable sensors can be affected by external factors such as changes in weather conditions or the presence of other animals in the vicinity. AI algorithms can be trained to recognize and account for these external factors, providing more accurate and reliable analysis of the data.

Finally, the cost of implementing a wearable sensor system for a large herd of cattle can be prohibitive for many farmers. AI algorithms can be used to optimize the use of the sensors, reducing the number of sensors required while still providing comprehensive coverage of the herd. This can reduce the cost of implementation and make the technology more accessible to smaller farmers.

Overall, AI technology can address the problems of using wearable sensors for cattle health monitoring by providing a more efficient, accurate, and cost-effective method of monitoring. By using AI algorithms to pre-process and analyse the data collected by the sensors, farmers can detect signs of illness or stress early, allowing for timely treatment and reducing the risk of loss of life and income. AI can also interpolate missing data, account for external factors, and optimize the use of sensors, ensuring more comprehensive and reliable monitoring of the herd

3.1 DISADVANTAGES

- It having less efficient.
- Live monitoring will be not be possible.
- Highly expensive
- Data Inaccurate

CHAPTER 4
PROPOSED SYSTEM

The proposed system for cattle health monitoring using AI aims to leverage advanced technology to track the health of cattle in real-time. The system uses image processing and AI algorithms to compare real-time images of healthy cattle with those of diseased cattle, enabling farmers to identify early signs of disease and take corrective action before it spreads to the entire herd. Additionally, the system also maintains the temperature in the cattle room using the count of cattle present.

The system consists of a camera, a computer, and a software program. The camera is installed in the cattle room and captures images of the cattle periodically. The computer processes these images using image processing algorithms to identify individual cattle and count the number of cattle present in the room. The system then uses AI algorithms to compare the real-time image of each individual cattle with the reference image of a healthy cattle to determine if it's healthy or not.

In case a diseased cattle is detected, the system alerts the farmer immediately, allowing them to take the necessary steps to prevent the spread of disease to the rest of the herd. This can include quarantining the diseased cattle, providing appropriate medication, and consulting with a veterinarian.

The system also maintains the temperature in the cattle room using the count of cattle present. The temperature in the room is automatically adjusted based on the number of cattle present in the room, ensuring that the cattle are comfortable and not stressed due to excessive heat or cold.

Overall, this system can help farmers monitor the health of their cattle in real-time, enabling them to take corrective action before the disease spreads, reducing the risk of loss of life and income. Additionally, it can also help in reducing the manual effort required for monitoring and maintaining the temperature in the cattle room, freeing up farmers to focus on other important tasks. The system would involve installing a camera in the cattle's environment, such as in a barn or pasture. The camera would capture images of the cattle and transmit them to a microcontroller, such as an Arduino.

The microcontroller would use image processing algorithms to analyze the images captured by the camera. These algorithms could include object detection, motion detection, and color analysis to detect changes in the cattle's behavior and appearance that could indicate a health problem. The results of the image processing would be displayed on an LCD screen, which could show information such as the cattle's activity level, temperature, and any detected health problems. If a health problem is detected, a buzzer could be activated to alert the farmer or rancher.

4.1 ADVANTAGES

- Real monitoring system
  - It allows farmers to monitor the health of their cattle in real-time, enabling them to take corrective action before the disease spreads, reducing the risk of loss of life and income
- Reduced manual effort
  - It reduces the manual effort required for monitoring and maintaining the temperature in the cattle room, freeing up farmers to focus on other important tasks.
- High accuracy and reliability
  - It is highly accurate and reliable. The AI algorithms used in the system are trained on a large dataset of cattle images, making them highly accurate in detecting early signs of disease.
- Reduced use of medication
  - The system can help in reducing the use of antibiotics and other medications, as farmers can take preventive measures to ensure the

health of their cattle, rather than relying on medication after the disease has spread. This can lead to healthier cattle and safer food for consumers.

- Reduced cost
  - Furthermore, the system can also help in reducing costs for farmers. By detecting early signs of disease and taking corrective action, farmers can avoid the need for expensive medication and treatment later on. Additionally, by maintaining the temperature in the cattle room automatically, the system can help in reducing the energy costs associated with manually controlling the temperature.

CHAPTER 5

WORKING AND DIAGRAMS

5.1 WORKING

The proposed system for cattle health monitoring system using AI aims to leverage advanced technology to track the health of cattle in real-time. The system uses image processing and AI algorithms to compare real-time images of healthy cattle with those of diseased cattle, enabling farmers to identify early signs of disease and take corrective action before it spreads to the entire herd. Additionally, the system also maintains the temperature in the cattle room using the count of cattle present.

The system is composed of a camera or a set of cameras installed in the cattle room. The cameras capture real-time images of the cattle and transmit them to a computer system equipped with AI algorithms. The images are analyzed to identify any signs of disease, such as changes in skin color, lesions, or unusual behavior. The system compares these images with a database of healthy cattle images to determine if there are any deviations from the norm.

The AI algorithms used in the system are trained using machine learning techniques. The algorithms learn from a large dataset of cattle images, both healthy and diseased. The training process involves teaching the algorithms to recognize patterns in the images that are indicative of disease. Once the algorithms are trained, they can be used to analyze real-time images and detect any signs of disease.

In addition to disease detection, the system also monitors the temperature in the cattle room. The system uses the count of cattle present to determine the appropriate temperature for the room. The number of cattle present in the room affects the temperature and humidity levels, and maintaining the right conditions is crucial for the health and well-being of the animals. The system uses image processing to count the number of cattle present in the room and adjusts the temperature accordingly.

The system also generates alerts when any signs of disease are detected. The alerts can be sent to the farmer via email, text message, or a mobile app. The alerts contain information about the disease detected, the severity of the condition, and recommended actions to take. The system can also be integrated with other farm management systems to provide a comprehensive view of the health of the herd and enable farmers to make informed decisions about their operations.

To ensure the accuracy of the system, it is important to maintain a large and diverse dataset of cattle images. The dataset should include images of different breeds, ages, and genders, as well as images taken in different lighting and environmental conditions. This will help the AI algorithms to recognize patterns in the images and improve the accuracy of disease detection.

The system also requires regular maintenance and calibration to ensure that it is functioning properly. The cameras need to be cleaned and checked for any damage, and the AI algorithms need to be updated with new data. The system should also be tested regularly to ensure that it is providing accurate results.

In summary, the proposed system for cattle health monitoring using AI is a valuable tool for farmers to monitor the health of their cattle in real-time. The system uses image processing and AI algorithms to compare real-time images of healthy cattle with those of diseased cattle, enabling early detection of disease and timely intervention. The system also maintains the temperature in the cattle room using the count of cattle present, ensuring the health and well-being of the animals. The system generates alerts when any signs of disease are detected and can be integrated with other farm management systems to provide a comprehensive view of the health of the herd.

5.2 BLOCK DIAGRAM

5.2.1 Software Part

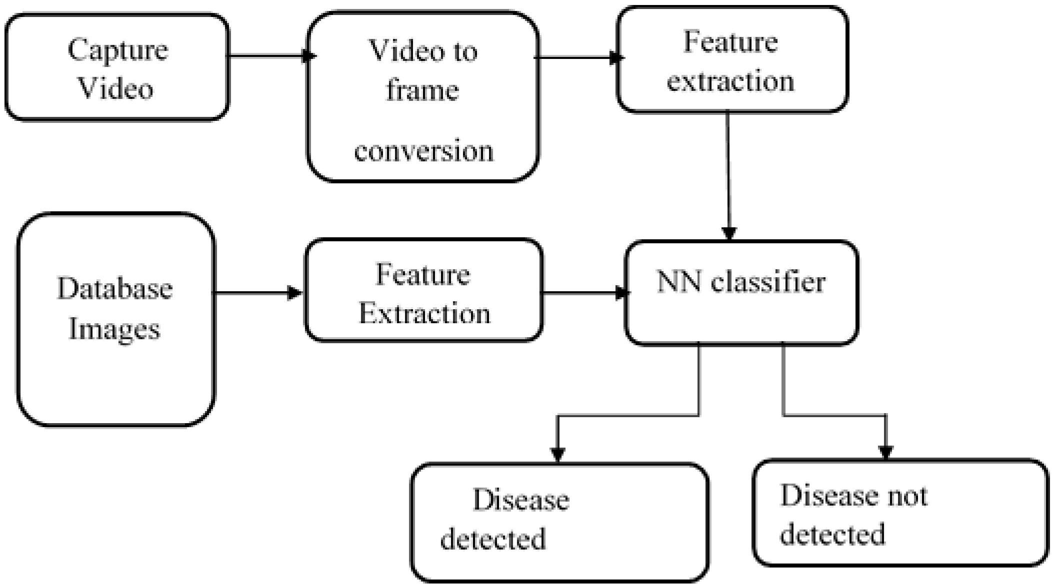


Fig 5.1 Software Block Diagram

5.2.2 Hardware Part

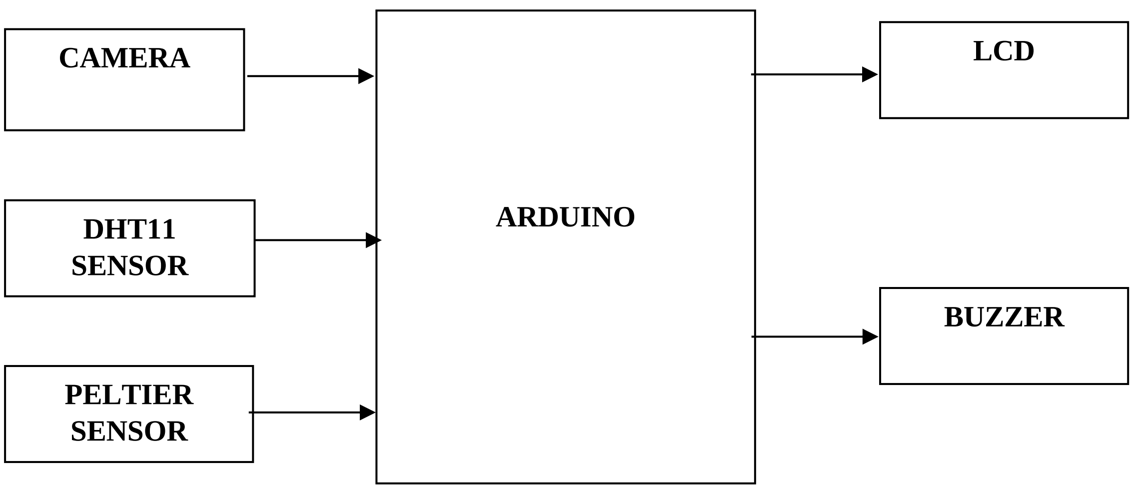
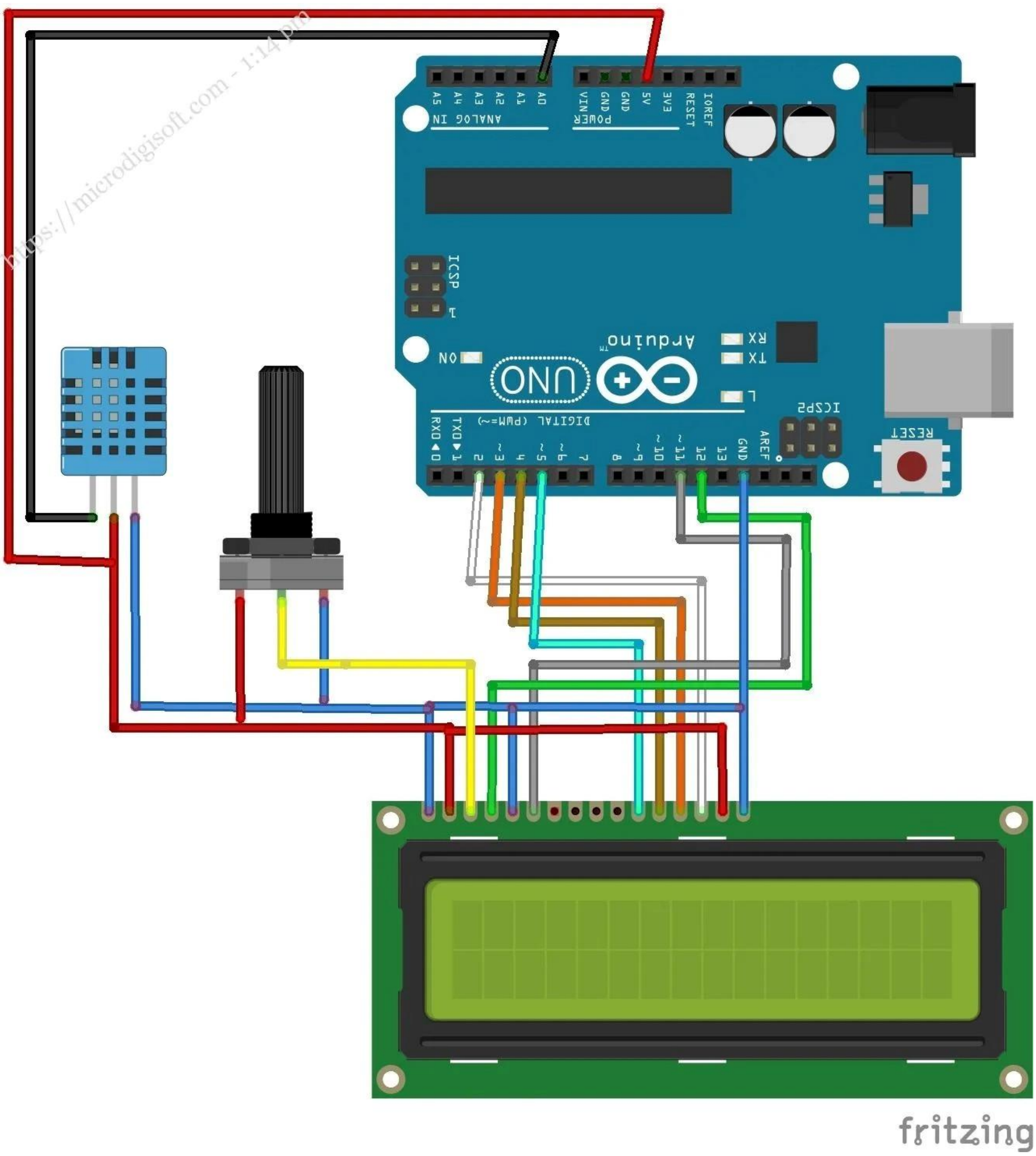


Fig 5.2 Hardware Block Diagram

5.3 CIRCUIT DIAGRAM



5.3 Circuit Diagram of the Prototype

CHAPTER 6

HARDWARE REQUIREMENT

The following hardware components are,

- Microcontroller
- LCD
- Camera
- Buzzer
- DHT11
- Peltier

6.1 MICROCONTROLLER

General-purpose input/output (GPIO) is a pin on an IC (Integrated Circuit). It can be either input pin or output pin, whose behaviour can be controlled at the run time.

6.1.1 Where To Use Microcontroller

The ESP32s is a very user friendly and low cost device to provide internet connectivity to your projects. The module can work both as a Access point (can create hotspot) and as a station (can connect to Wi-Fi), hence it can easily fetch data and upload it to the internet making Internet of Things as easy as possible. It can also fetch data from internet using API's hence your project could access any information that is available in the internet, thus making it smarter. Another exciting feature of this module is that it can be programmed using the Arduino IDE which makes it a lot more user friendly. However this version of the module has only 2 GPIO pins (you can hack it to use upto 4 so you have to use it along with another microcontroller like Arduino, else you can look onto the more standalone ESP-12 or ESP-32 versions. So if you are looking for a module to get started with IOT or to provide internet connectivity to your project then this module is the right choice for you.

6.1.2 How To Use Microcontroller

There are so many methods and IDEs available to with ESP modules, but the most commonly used on is the Arduino IDE. So let us discuss only about that further below. The ESP32s module works with 3.3V only, anything more than 3.7V would kill the module hence be cautions with your circuits. The best way to program an ESP-01 is by using the FTDI board that supports 3.3V programming. If you don't have one it is recommended to buy one or for time being you can also use an Arduino board. One commonly problem that every one faces with ESP-01 is the powering up problem. The module is a bit power hungry while programming and hence you can power it with a 3.3V pin on Arduino or just use a potential divider. So it is important to make a small voltage regulator for 3.31v that could supply a minimum of 500mA. One recommended regulator is the LM317 which could handle the job easily. A simplified circuit diagram for using the ESP32s-01 module is given below,

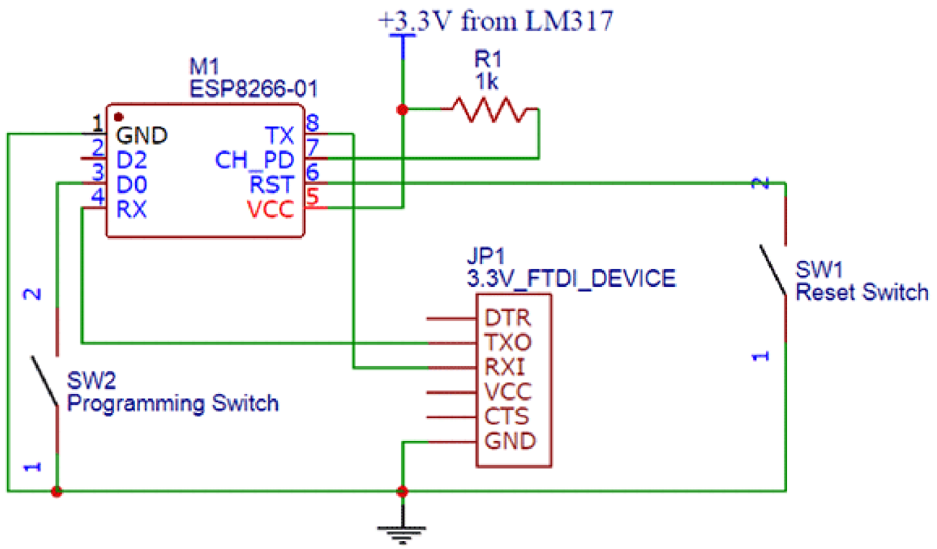


Fig 6.1 Circuit Diagram of Microcontroller

The switch SW2 (Programming Switch) should be held pressed to hold the GPIO-0 pin to ground. This way we can enter into the programming mode and upload the code. Once the code is released the switch can be released.

6.1.3 Power Requirement

As the operating voltage range of ESP32S is 3V to 3.6V, the board comes with a LDO voltage regulator to keep the voltage steady at 3.3V. It can reliably supply up to 600mA, which should be more than enough when ESP32S pulls as much as 80mA during RF transmissions. The output of the regulator is also broken out to one of the sides of the board and labelled as 3V3. This pin can be used to supply power to external components.

- Operating Voltage: 2.5V to 3.6V
- On-board 3.3V 600mA regulator
- 80mA Operating Current
- 20 µA during Sleep Mode

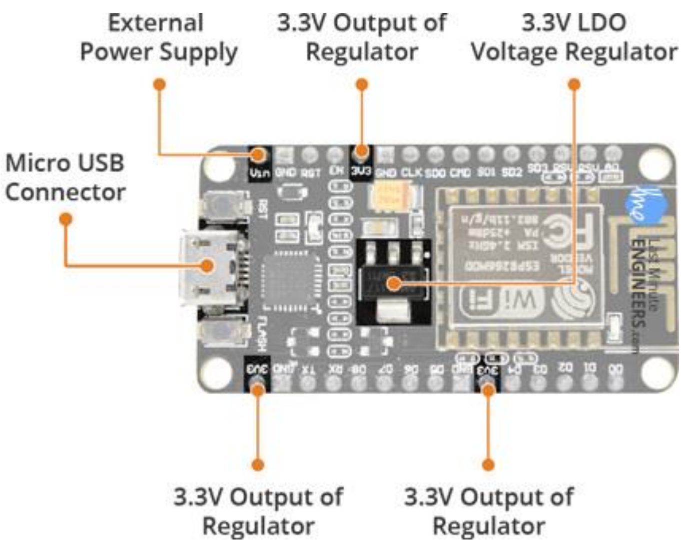


Fig 6.2 ESP32s Module

Power to the ESP32S NodeMCU is supplied via the on-board Micro-B USB connector. Alternatively, if you have a regulated 5V voltage source, the Vin pin can be used to directly supply the ESP32S and its peripherals. The ESP32S requires a 3.3V power supply and 3.3V logic levels for communication. The GPIO pins are not 5V-tolerant! If you want to interface the board with 5V (or higher) components, you'll need to do some level shifting.

6.1.4 Peripherals and I/O

The ESP32S NodeMCU has total 17 GPIO pins broken out to the pin headers on both sides of the development board. These pins can be assigned to all sorts of peripheral duties, including:

- ADC channel – A 10-bit ADC channel.
- UART interface – UART interface is used to load code serially.
- PWM outputs – PWM pins for dimming LEDs or controlling motors. SPI, I2C & I2S interface – SPI and I2C interface to hook up all sorts of sensors and peripherals.
- I2S interface – I2S interface if you want to add sound to your project.

6.1.5 Multiplexed I/O

- 1 ADC channels
- 2 UART interfaces
- 4 PWM outputs
- SPI, I2C & I2S interface

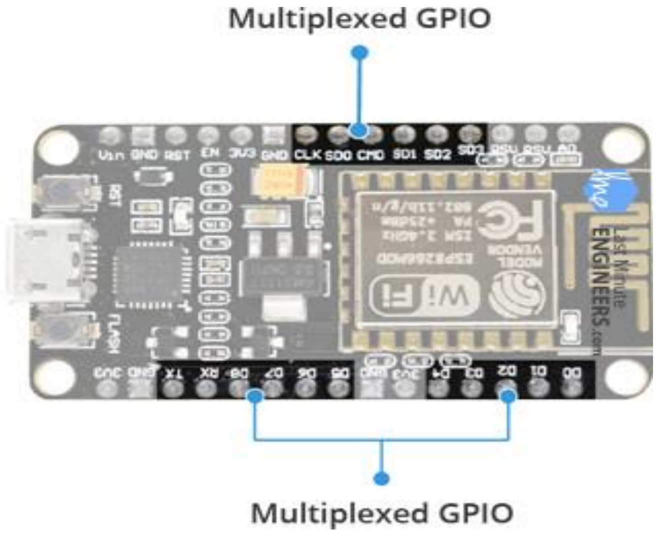


Fig 6.3 Multiplexed GPIO

The ESP32S's pin multiplexing feature (Multiple peripherals multiplexed on a single GPIO pin). Meaning a single GPIO pin can act as PWM/UART/SPI.

6.1.6 On Board switches and LED Indicator

The ESP32S NodeMCU features two buttons. One marked as RST located on the top left corner is the Reset button, used of course to reset the ESP32S chip. The other FLASH button on the bottom left corner is the download button used while upgrading firmware.

- RST – Reset the ESP32S chip
- FLASH – Download new programs
- Blue LED – User Programmable

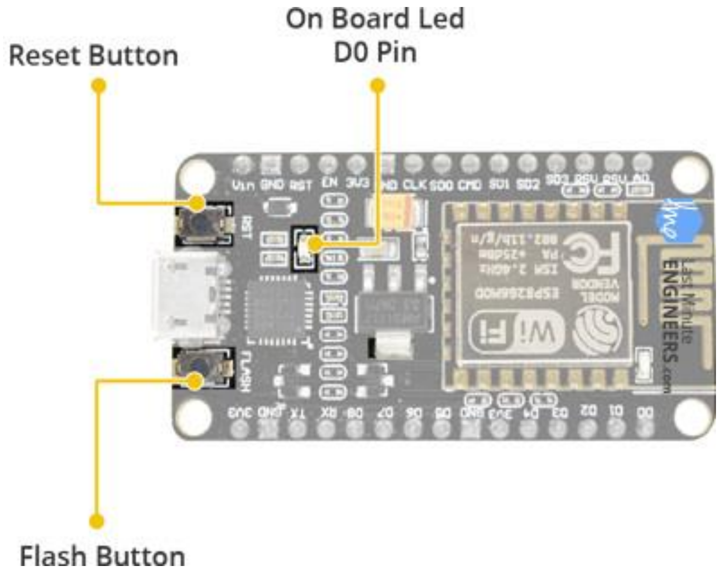


Fig 6.4 On board Switches and LED

The board also has a LED indicator which is user programmable and is connected to the D0 pin of the board.

6.1.7 Serial Communication

The board includes CP2102 USB-to-UART Bridge Controller from Silicon Labs, which converts USB signal to serial and allows your computer to program and communicate with the ESP32S chip.

- CP2102 USB-to-UART converter
- 4.5 Mbps communication speed
- Flow Control support

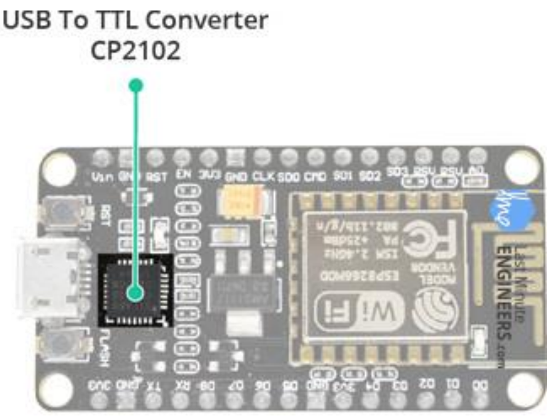


Fig 6.5 USB to TTL Converter

If you have an older version of CP2102 driver installed on your PC, we recommend upgrading now.

6.1.8 Pinout Diagram

The ESP32S NodeMCU has total 30 pins that interface it to the outside world. The connections are as follows:

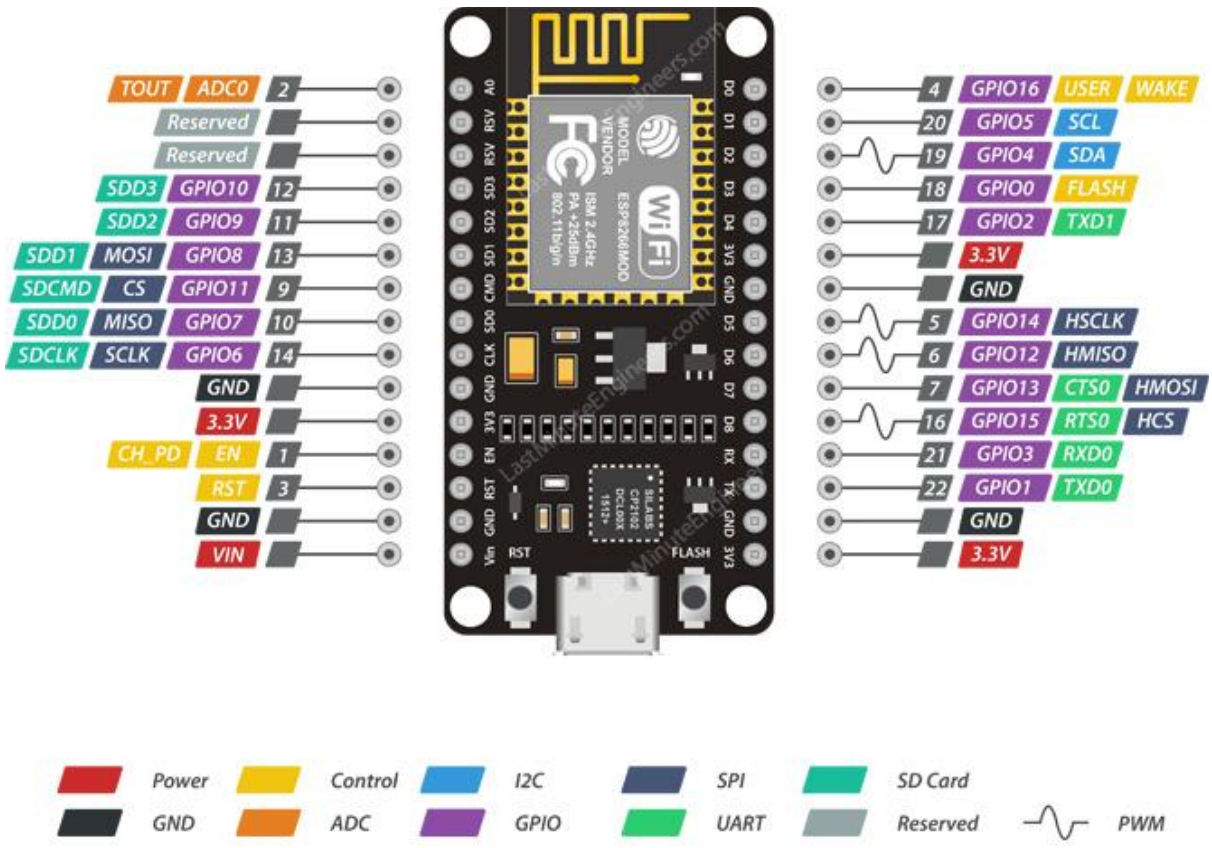


Fig 6.6 Pin Diagram of Microcontroller

For the sake of simplicity, we will make groups of pins with similar functionalities.

- Power Pins
  - There are four power pins viz. one VIN pin & three 3.3V pins. The VIN pin can be used to directly supply the ESP32S and its peripherals, if you have a regulated 5V voltage source. The 3.3V

pins are the output of an on-board voltage regulator. These pins can be used to supply power to external components.

- GND
  - GND is a ground pin of ESP32S NodeMCU development board.
- I2C Pins
  - I2C Pins are used to hook up all sorts of I2C sensors and peripherals in your project. Both I2C Master and I2C Slave are supported. I2C interface functionality can be realized programmatically, and the clock frequency is 100 kHz at a maximum. It should be noted that I2C clock frequency should be higher than the slowest clock frequency of the slave device.
- GPIO Pins
  - ESP32S NodeMCU has 17 GPIO pins which can be assigned to various functions such as I2C, I2S, UART, PWM, IR Remote Control, LED Light and Button programmatically. Each digital enabled GPIO can be configured to internal pull-up or pull-down, or set to high impedance. When configured as an input, it can also be set to edge-trigger or level-trigger to generate CPU interrupts.
- ADC Channel
  - The NodeMCU is embedded with a 10-bit precision SAR ADC. The two functions can be implemented using ADC viz. Testing power supply voltage of VDD3P3 pin and testing input voltage of TOUT pin. However, they cannot be implemented at the same time.
- UART Pins
  - ESP32S NodeMCU has 2 UART interfaces, i.e. UART0 and UART1, which provide asynchronous communication (RS232 and RS485), and can communicate at up to 4.5 Mbps. UART0 (TXD0, RXD0, RST0 & CTS0 pins) can be used for communication. It

supports fluid control. However, UART1 (TXD1 pin) features only data transmit signal so, it is usually used for printing log.

- SPI Pins
    - ESP32S features two SPIs (SPI and HSPI) in slave and master modes. These SPIs also support the following general-purpose SPI features:
    - 4 timing modes of the SPI format transfer
    - Up to 80 MHz and the divided clocks of 80 MHz
    - Up to 64-Byte FIFO
- SDIO Pins
    - ESP32S features Secure Digital Input/Output Interface (SDIO) which is used to directly interface SD cards. 4-bit 25 MHz SDIO v1.1 and 4-bit 50 MHz SDIO v2.0 are supported.
- PWM Pins
    - The board has 4 channels of Pulse Width Modulation (PWM). The PWM output can be implemented programmatically and used for driving digital motors and LEDs. PWM frequency range is adjustable from 1000 μs to 10000 μs, i.e., between 100 Hz and 1 kHz.
- Control Pins
    - Control Pins are used to control ESP32S. These pins include Chip Enable pin (EN), Reset pin (RST) and WAKE pin.
    - EN pin – The ESP32S chip is enabled when EN pin is pulled HIGH. When pulled LOW the chip works at minimum power.
    - RST pin – RST pin is used to reset the ESP32S chip.
    - WAKE pin – Wake pin is used to wake the chip from deep-sleep.

6.1.9 Features

- Low cost, compact and powerful Wi-Fi Module.
- Power Supply: +3.3V only.
- Current Consumption: 100mA.
- I/O Voltage: 3.6V (max).
- I/O source current: 12mA (max).
- Built-in low power 32-bit MCU @ 80MHz.
- 512kB Flash Memory.
- Can be used as Station or Access Point or both combined.
- Supports Deep sleep (<10uA).
- Supports serial communication hence compatible with many development platform like Arduino.
- Can be programmed using Arduino IDE or AT-commands or Lua Script .

6.1.10 Applications

- IOT Projects.
- Access Point Portals.
- Wireless Data logging.
- Smart Home Automation.
- Learn basics of networking.
- Portable Electronics.
- Smart bulbs and Sockets.

6.2 LCD

LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. A 16×2 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred over seven segments and other multi segment LEDs. The reasons being: LCDs are economical; easily programmable; have no limitation of displaying special & even custom characters (unlike in seven segments), animation and so on.

A 16×2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5×7 pixel matrix. This LCD has two registers, namely, Command and Data.

The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD. Click to learn more about internal structure of a LCD.

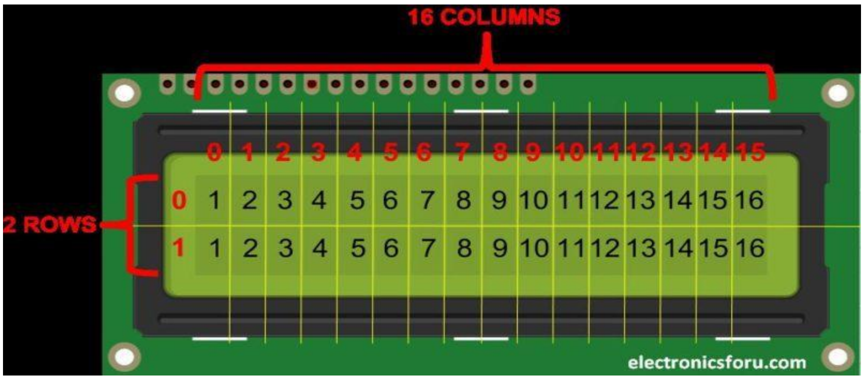


Fig 6.7 LCD

The basic structure of LCD should be controlled by changing the applied current. We must use a polarized light. Liquid crystal should able be to control both of the operation to transmit or can also able to change the polarized light. As mentioned above that we need to take two polarized glass pieces filter in the making of the liquid crystal. The glass which does not have a polarized film on the surface of it must be rubbed with a special polymer which will create microscopic grooves on the surface of the polarized glass filter. The grooves must be in the same direction of the polarized film. Now we have to add a coating of pneumatic liquid phase crystal on one of the polarized filter of the polarized glass. The microscopic channel cause the first layer molecule to align with filter orientation. When the right angle appears at the first layer piece, we should add a second piece of glass with the polarized film. The first filter will be naturally polarized as the light strikes it at the starting stage.

Thus the light travels through each layer and guided on the next with the help of molecule. The molecule tends to change its plane of vibration of the light in order to match their angle. When the light reaches to the far end of the liquid crystal substance, it vibrates at the same angle as that of the final layer of the molecule vibrates. The light is allowed to enter into the device only if the second layer of the polarized glass matches with the final layer of the molecule.

6.2.1 Working

The principle behind the LCD's is that when an electrical current is applied to the liquid crystal molecule, the molecule tends to untwist. This causes the angle of light which is passing through the molecule of the polarized glass and also cause a change in the angle of the top polarizing filter. As a result a little light is allowed to pass the polarized glass through a particular area of the LCD. Thus that particular area will become dark compared to other. The LCD works on the principle of blocking light. While constructing the LCD's, a reflected mirror is arranged at the back. An electrode plane is made of indium-tin oxide which is kept on top and a polarized glass with a polarizing film is also added on the bottom of the device. The complete region of the LCD has to be enclosed by a common electrode and above it should be the liquid crystal matter.

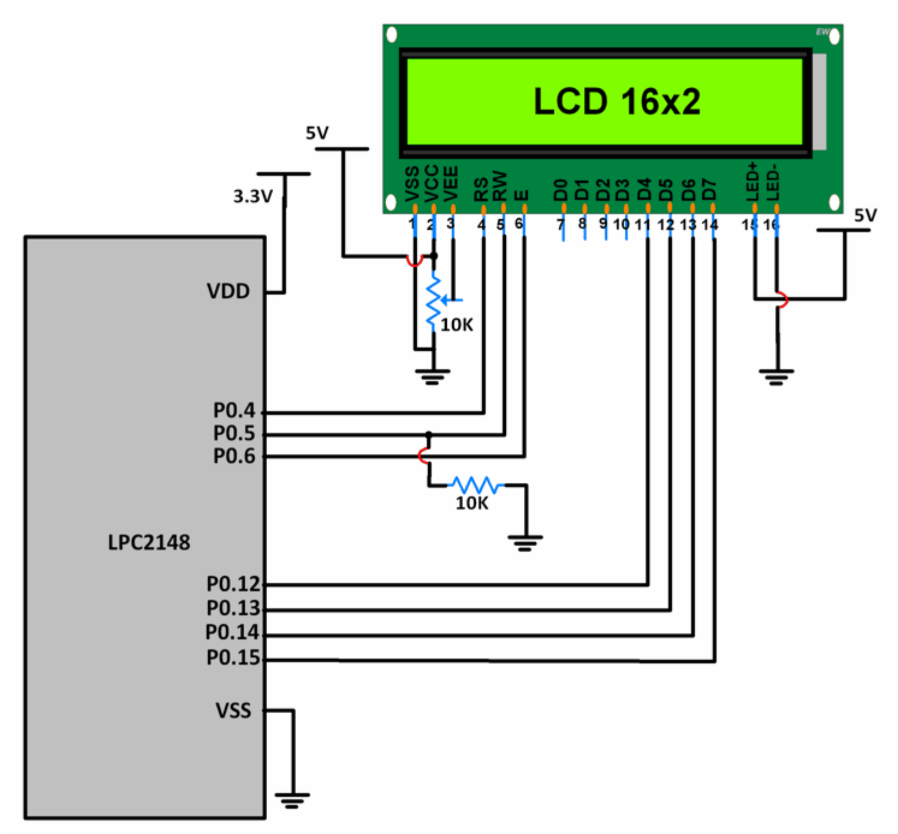


Fig 6.8 Circuit Diagram of LCD

Next comes to the second piece of glass with an electrode in the form of the rectangle on the bottom and, on top, another polarizing film. It must be considered that both the pieces are kept at right angles. When there is no current, the light passes through the front of the LCD it will be reflected by the mirror and bounced back. As the electrode is connected to a battery the current from it will cause the liquid crystals between the common-plane electrode and the electrode shaped like a rectangle to untwist. Thus the light is blocked from passing through. That particular rectangular area appears blank.

6.2.2 Pinout Diagram

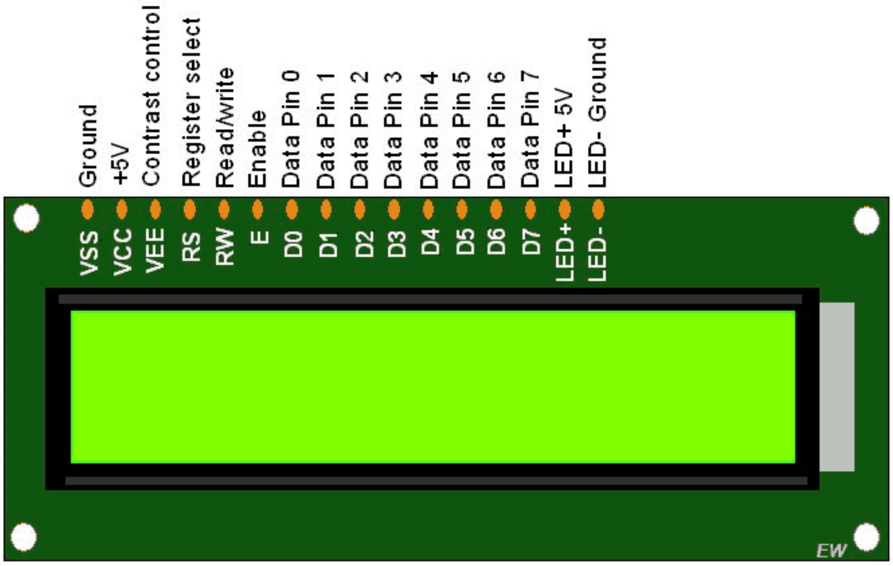


Fig 6.9 Pin Diagram of LCD(16x2)

Table 6.1 LCD Pin Description

S. NO	PIN NO.	PIN NAME	PIN TYPE	PIN DESCRIPTON	PIN CONNECTION
1	Pin 1	Ground	Source Pin	This is a ground pin of LCD	Connected to the ground of the MCU/ Power source
2	Pin 2	VCC	Source Pin	This is the supply voltage pin of LCD	Connected to the supply pin of Power source

3	Pin 3	V0/VEE	Control Pin	Adjusts the contrast of the LCD.	Connected to a variable POT that can source 0-5V
4	Pin 4	Register Select	Control Pin	Toggles between Command/Data Register	Connected to a MCU pin and gets either 0 or 1. 0 -> Command Mode 1-> Data Mode
5	Pin 5	Read/Write	Control Pin	Toggles the LCD between Read/Write Operation	Connected to a MCU pin and gets either 0 or 1. 0 -> Write Operation 1-> Read Operation
6	Pin 6	Enable	Control Pin	Must be held high to perform Read/Write Operation	Connected to MCU and always held high.
7	Pin 7-14	Data Bits (0-7)	Data/Command Pin	Pins used to send Command or data to the LCD.	In 4-Wire Mode Only 4 pins (0-3) is connected to MCU

					In 8-Wire Mode All 8 pins(0-7) are connected to MCU
8	Pin 15	LED Positive	LED Pin	Normal LED like operation to illuminate the LCD	Connected to +5V
9	Pin 16	LED Negative	LED Pin	Normal LED like operation to illuminate the LCD connected with GND.	Connected to ground

6.2.3 Features

- Operating Voltage is 4.7V to 5.3V.
- Current consumption is 1mA without backlight.
- Alphanumeric LCD display module, meaning can display alphabets and numbers.
- Consists of two rows and each row can print 16 characters.
- Each character is build by a 5×8 pixel box.
- Can work on both 8-bit and 4-bit mode.
- It can also display any custom generated characters.
- Available in Green and Blue Backlight.

6.2.5 Advantages

- LCD’s consumes less amount of power compared to CRT and LED.
- LCD’s are consist of some microwatts for display in comparison to some mill watts for LED’s.
- LCDs are of low cost.
- Provides excellent contrast.
- LCD’s are thinner and lighter when compared to cathode ray tube and LED.

6.2.6 Disadvantages

- Require additional light sources.
- Range of temperature is limited for operation.
- Low reliability.
- Speed is very low.
- LCD’s need an AC drive.

6.2.4 Applications

- Liquid crystal technology has major applications in the field of science and engineering as well on electronic devices.
- Liquid crystal thermometer.
- Optical imaging.
- The liquid crystal display technique is also applicable in visualization of the radio frequency waves in the waveguide.
- Used in the medical applications.

6.3 CAMERA

A webcam is a video camera that feeds or streams an image or video in real time to or through a computer to a computer network, such as the Internet. Webcams are typically small cameras that sit on a desk, attach to a user's monitor, or are built into the hardware. Webcams can be used during a video chat session involving two or more people, with conversations that include live audio and video. For example, Apple's iSight camera, which is built into Apple laptops, iMacs and a number of iPhones, can be used for video chat sessions, using the iChat instant messaging program (now called Messages). Webcam software enables users to record a video or stream the video on the Internet. As video streaming over the Internet requires a lot of bandwidth, such streams usually use compressed formats. The maximum resolution of a webcam is also lower than most handheld video cameras, as higher resolutions would be reduced during transmission. The lower resolution enables webcams to be relatively inexpensive compared to most video cameras, but the effect is adequate for video chat sessions.

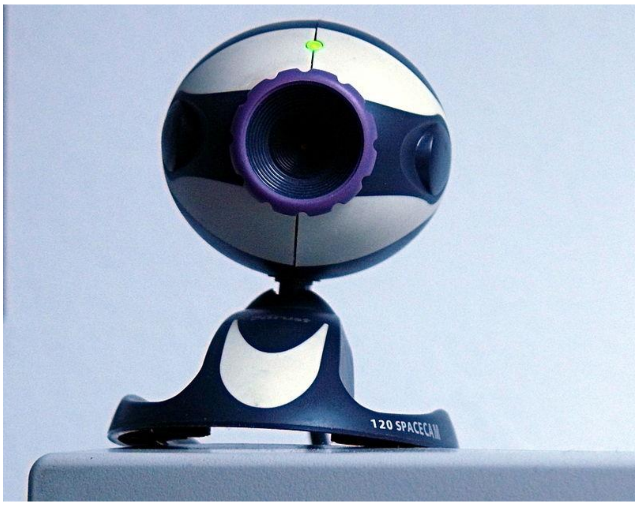


Fig 6.10 Camera

6.3.1 Optics

Various lenses are available, the most common in consumer-grade webcams being a plastic lens that can be manually moved in and out to focus the camera. Fixed-focus lenses, which have no provision for adjustment, are also available. As a camera system's depth of field is greater for small image formats and is greater for lenses with a large f-number (small aperture), the systems used in webcams have a sufficiently large depth of field that the use of a fixed-focus lens does not impact image sharpness to a great extent.

Most models use simple, focal-free optics (fixed focus, factory-set for the usual distance from the monitor to which it is fastened to the user) or manual focus.

6.3.2 Compression

Digital video streams are represented by huge amounts of data, burdening its transmission (from the image sensor, where the data is continuously created) and storage alike. Most if not all cheap webcams come with built-it ASIC to do video compression in real-time.

Support electronics read the image from the sensor and transmit it to the host computer. The camera pictured to the right, for example, uses a Sonix SN9C101 to transmit its image over USB. Typically, each frame is transmitted uncompressed in RGB or YUV or compressed as JPEG. Some cameras, such as mobile-phone cameras, use a CMOS sensor with supporting electronics "on die", i.e. the sensor and the support electronics are built on a single silicon chip to save space and manufacturing costs. Most webcams feature built-in microphones to make video calling and videoconferencing more convenient.

6.3.3 Interface

Typical interfaces used by articles marketed as a "webcam" are USB, Ethernet and IEEE (denominated as IP camera). Further interfaces such as e.g. Composite video or S-Video are also available.

The USB video device class (UVC) specification allows inter-connectivity of webcams to computers without the need for proprietary device drivers.

6.3.4 Software

Various proprietary as well as free and open-source software is available to handle the UVC stream. One could use Guvcview or GStreamer and GStreamer-based software to handle the UVC stream.

6.3.5 Characteristics

Webcams are known for their low manufacturing cost and their high flexibility, making them the lowest-cost form of video telephony. As webcams evolved simultaneously with display technologies, USB interface speeds and broadband internet speeds, the resolution went up from gradually 320×240, to 640×480, and some even offering 1280×720 (aka 720p) or 1920×1080 (aka 1080p) resolution.

Despite the low cost, the resolution offered as of 2019 is impressive, with now the low-end webcams offering resolutions of 720p, mid-range webcams offering 1080p resolution, and high-end webcams offering 4K resolution at 60 fps.

Webcams have become a source of security and privacy issues, as some built-in webcams can be remotely activated by spyware. To address this concern, many webcams come with a physical lens cover.

6.3.6 Uses

The most popular use of webcams is the establishment of video links, permitting computers to act as videophones or videoconference stations. Other popular uses include security surveillance, computer vision, video broadcasting, and for recording social videos.

The video streams provided by webcams can be used for a number of purposes.

- Health Care
  - Most modern webcams are capable of capturing arterial pulse rate by the use of a simple algorithmic trick. Researchers claim that this method is accurate to ±5 bpm.
- Video Monitoring
  - Webcams may be installed at places such as childcare centres, offices, shops and private areas to monitor security and general activity.
- Commerce
  - Webcams have been used for augmented reality experiences online. One such function has the webcam act as a "magic mirror" to allow an online shopper to view a virtual item on themselves. The Webcam Social Shopper is one example of software that utilizes the webcam in this manner.
- Video Calling and Video Conferencing

  Further information: Videophone, Videoconferencing, and Video telephony.

- Webcam can be added to instant messaging, text chat services such as AOL Instant Messenger, and VoIP services such as Skype, one-to-one live video communication over the Internet has now reached millions of mainstream PC users worldwide. Improved video quality has helped webcams encroach on traditional video conferencing systems.
- New features such as automatic lighting controls, real-time enhancements (retouching, wrinkle smoothing and vertical stretch), automatic face tracking and autofocus, assist users by providing substantial ease-of-use, further increasing the popularity of webcams. Webcam features and performance can vary by program, computer operating system, and also by the computer's processor capabilities. Video calling support has also been added to several popular instant messaging programs.

➢ Video Security

- Webcams can be used as security cameras. Software is available to allow PC-connected cameras to watch for movement and sound, recording both when they are detected. These recordings can then be saved to the computer, e-mailed, or uploaded to the Internet. In one well-publicised case, a computer e-mailed images of the burglar during the theft of the computer, enabling the owner to give police a clear picture of the burglar's face even after the computer had been stolen.
- Unauthorized access of webcams can present significant privacy issues (see "Privacy" section below).

➢ Video Clips and Stills

- Webcams can be used to take video clips and still pictures. Various software tools in wide use can be employed for this, such as PicMaster (for use with Windows operating systems), Photo Booth (Mac), or Cheese (with Unix systems). For a more complete list see Comparison of webcam software.

- Input Control Devices

  - Special software can use the video stream from a webcam to assist or enhance a user's control of applications and games. Video features, including faces, shapes, models and colours can be observed and tracked to produce a corresponding form of control. For example, the position of a single light source can be tracked and used to emulate a mouse pointer, a head-mounted light would enable hands-free computing and would greatly improve computer accessibility. This can be applied to games, providing additional control, improved interactivity and immersiveness.
  - Freetrack is a free webcam motion-tracking application for Microsoft Windows that can track a special head-mounted model in up to six degrees of freedom and output data to mouse, keyboard, joystick and FreeTrack-supported games. By removing the IR filter of the webcam, IR LEDs can be used, which has the advantage of being invisible to the naked eye, removing a distraction from the user. TrackIR is a commercial version of this technology.
  - The EyeToy for the PlayStation 2, PlayStation Eye for the PlayStation 3, and the Xbox Live Vision camera and Kinect motion sensor for the Xbox 360 and are colour digital cameras that have been used as control input devices by some games. Small webcam-based PC games are available as either standalone executables or inside web browser windows using Adobe Flash.

6.4 BUZZER

A buzzer or beeper is an audio signalling device, which may be mechanical, electromechanical, or piezoelectric (piezo for short). Typical uses of buzzers and beepers include alarm devices, timers, and confirmation of user input such as a mouse click or keystroke.

Fig 6.11 Buzzer

6.4.1 How To Use Buzzer

A buzzer is a small yet efficient component to add sound features to our project/system. It is very small and compact 2-pin structure hence can be easily used on breadboard, Perf Board and even on PCBs which makes this a widely used component in most electronic applications.

There are two types are buzzers that are commonly available. The one shown here is a simple buzzer which when powered will make a Continuous Beep sound, the other type is called a readymade buzzer which will look bulkier than this and will produce a Beep Sound due to the internal oscillating circuit present inside it. But, the one shown here is most widely used because it can be customised with help of other circuits to fit easily in our application.

This buzzer can be used by simply powering it using a DC power supply ranging from 4V to 9V. A simple 9V battery can also be used, but it is recommended to use a regulated +5V or +6V DC supply. The buzzer is normally associated with a switching circuit to turn ON or turn OFF the buzzer at required time and require interval.

6.4.2 Types

- Electromechanical
  - Early devices were based on an electromechanical system identical to an electric bell without the metal gong.
  - Similarly, a relay may be connected to interrupt its own actuating current, causing the contacts to buzz. Often these units were anchored to a wall or ceiling to use it as a sounding board. The word "buzzer" comes from the rasping noise that electromechanical buzzers made.
- Mechanical
  - A joy buzzer is an example of a purely mechanical buzzer and they require drivers. Other examples of them are doorbells.
- Piezoelectric

Fig 6.12 Piezoelectric buzzer

  - 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.

Fig 6.13 Piezoelectric Disk Beeper

- Interior of a readymade loudspeaker, showing a piezoelectric-disk-beeper (With 3 electrodes ... including 1 feedback-electrode (the central, small electrode joined with red wire in this photo), and an oscillator to self-drive the buzzer.
- A piezoelectric buzzer/beeper also depends on acoustic cavity resonance or Helmholtz resonance to produce an audible beep.

6.4.3 Pin Configuration

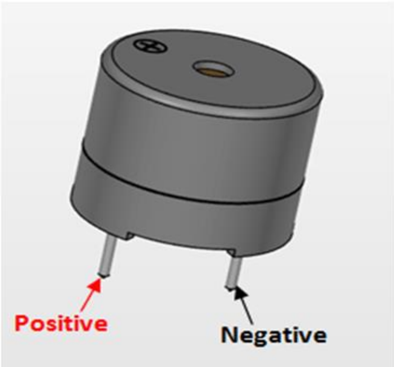


Fig 6.14 Buzzer Pin Configuration

Table 6.2 Buzzer Pin Description

PIN NUMBER	PIN NAME	DESCRIPTION
1	Positive	Identified by (+) symbol or longer terminal lead. Can be powered by 6V DC
2	Negative	Identified by short terminal lead. Typically connected to the ground of the circuit

6.4.4 Specifications

- Rated Voltage: 6V DC.
- Operating Voltage: 4-8V DC.
- Rated current: <30mA.
- Sound Type: Continuous Beep.
- Resonant Frequency: ~2300 Hz.
- Small and neat sealed package.
- Breadboard and Perf board friendly.

6.4.5 Applications

- Alarming Circuits, where the user has to be alarmed about something.
- Communication equipment.
- Automobile electronics.
- Portable equipment, due to its compact size.

6.5 DHT-11

DHT11 humidity and temperature sensor is available as a sensor and as a module. The difference between this sensor and module is the pull-up resistor and a power-on LED. DHT11 is a relative humidity sensor. To measure the surrounding air this sensor uses a thermistor and a capacitive humidity sensor.

This sensor is used here to monitor the humidity variation of the environment where the crops are cultivated. This is a digital sensor and measures the humidity value in percentage format.

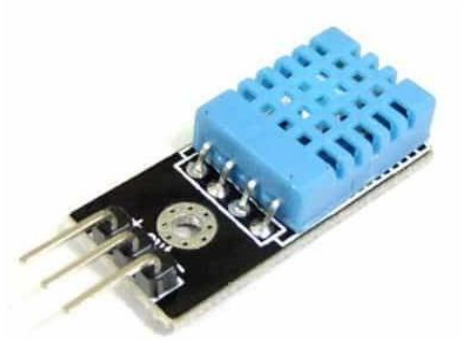


Fig 6.15 DHT11 Sensor

DHT11 is a low-cost digital sensor for sensing temperature and humidity. This sensor can be easily interfaced with any micro-controller such as Arduino, Raspberry Pi etc… to measure humidity and temperature instantaneously.

6.5.1 Working

DHT11 sensor consists of a capacitive humidity sensing element and a thermistor for sensing temperature. The humidity sensing capacitor has two electrodes with a moisture holding substrate as a dielectric between them. Change in the capacitance value occurs with the change in humidity levels. The IC measure, process this changed resistance values and change them into digital form. For measuring temperature this sensor uses a Negative Temperature coefficient thermistor, which causes a decrease in its resistance value with increase in temperature. To get larger resistance value even for the smallest change in temperature, this sensor is usually made up of semiconductor ceramics or polymers.

The temperature range of DHT11 is from 0 to 50 degree Celsius with a 2-degree accuracy. Humidity range of this sensor is from 20 to 80% with 5% accuracy. The sampling rate of this sensor is 1Hz .i.e. it gives one reading for every second. DHT11 is small in size with operating voltage from 3 to 5 volts. The maximum current used while measuring is 2.5mA.

DHT11 sensor has four pins- VCC, GND, Data Pin and a not connected pin. A pull-up resistor of 5k to 10k ohms is provided for communication between sensor and micro-controller.

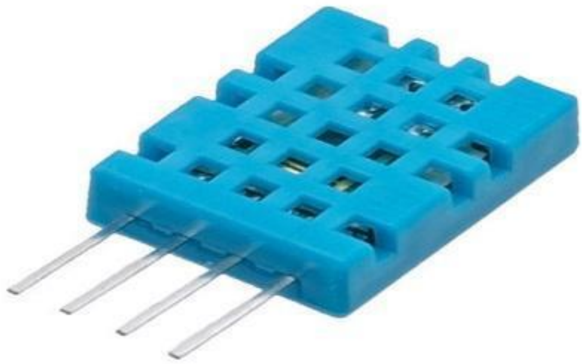


Fig 6.16 DHT11 Pins

DHT11 uses only one wire for communication. The voltage levels with certain time value defines the logic one or logic zero on this pin. The communication process is divided in three steps, first is to send request to DHT11 sensor then sensor will send response pulse and then it starts sending data of total 40 bits to the microcontroller.

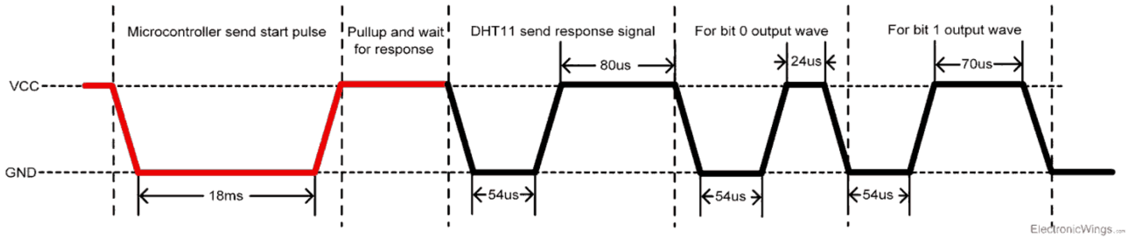


Fig 6.17 Waveform of DHT11

6.5.2 Communication Process

- Start pulse (Request)

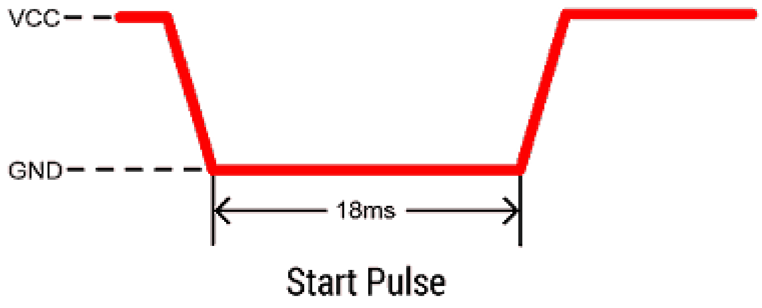


Fig 6.18 Start Pulse for DHT11

  - To start communication with DHT11, first we should send the start pulse to the DHT11 sensor.
  - To provide start pulse, pull down (low) the data pin minimum 18ms and then pull up, as shown in diag.

- Response

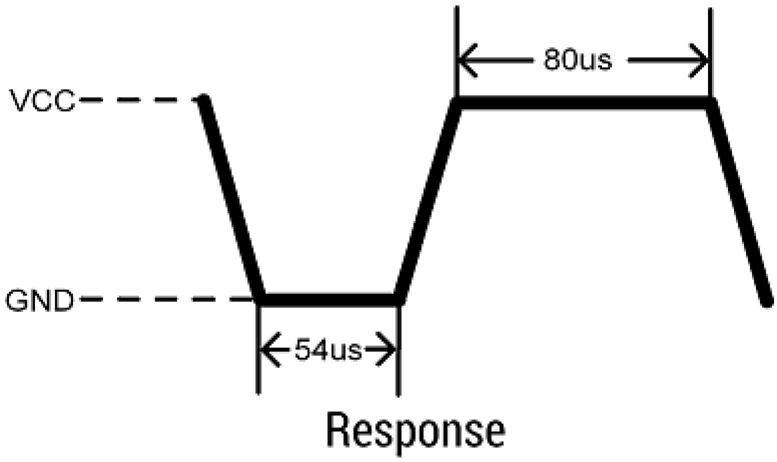


Fig 6.19 Response Pulse for DHT11

  - After getting start pulse from, DHT11 sensor sends the response pulse which indicates that DHT11 received start pulse.
  - The response pulse is low for 54us and then goes high for 80us.

➢ Data

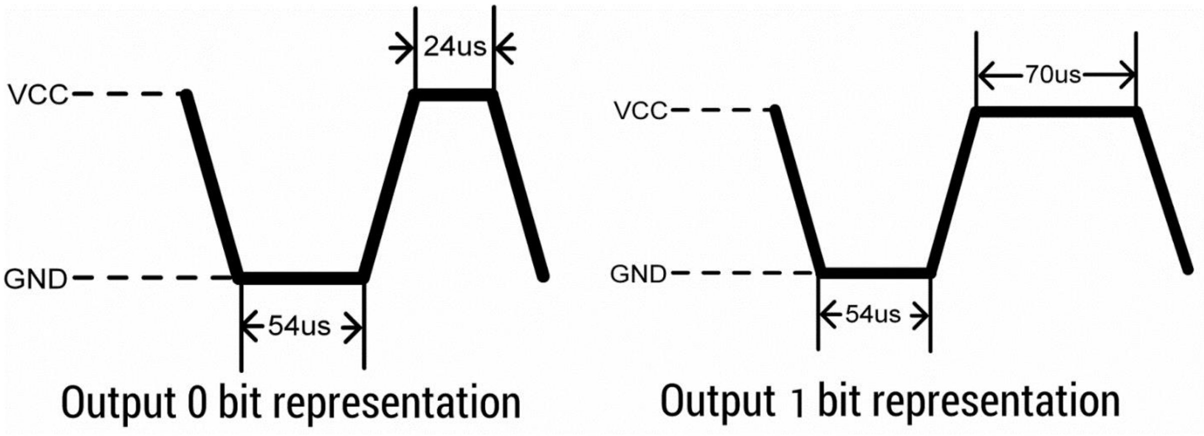


Fig 6.20 Data Representation for DHT11

- After sending the response pulse, DHT11 sensor sends the data, which contains humidity and temperature value along with checksum.
- The data frame is of total 40 bits long, it contains 5 segments (byte) and each segment is 8-bit long.
- In these 5 segments, first two segments contain humidity value in decimal integer form. This value gives us Relative Percentage Humidity. 1st 8-bits are integer part and next 8 bits are fractional part.
- Next two segments contain temperature value in decimal integer form. This value gives us temperature in Celsius form.
- Last segment is the checksum which holds checksum of first four segments.
- Here checksum byte is direct addition of humidity and temperature value. And we can verify it, whether it is same as checksum value or not. If it is not equal, then there is some error in the received data.
- Once data received, DHT11 pin goes in low power consumption mode till next start pulse.

➢ End of frame

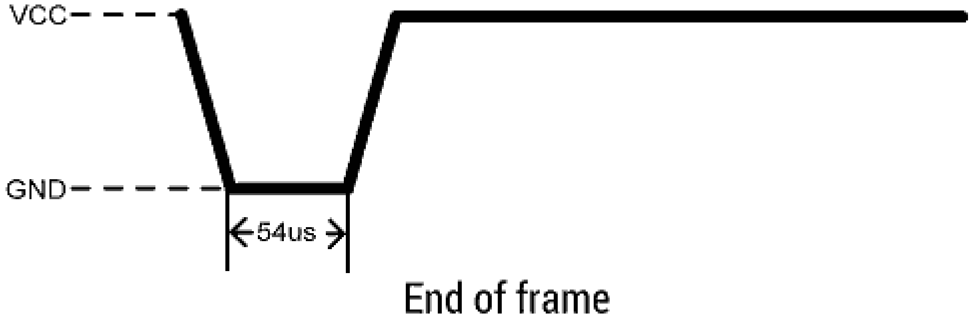


Fig 6.21 End of Frame for DHT11

- After sending 40-bit data, DHT11 sensor sends 54us low level and then goes high. After this DHT11 goes in sleep mode.

6.5.3 Features

- Ultra-low cost.
- 3 to 5V power and I/O.
- 2.5mA max current use during conversion (while requesting data).
- Good for 20-80% humidity readings with 5% accuracy.
- Good for 0-50°C temperature readings ±2°C accuracy.
- No more than 1 Hz sampling rate (once every second).
- Body size 15.5mm x 12mm x 5.5mm.
- 4 pins with 0.1" spacing.

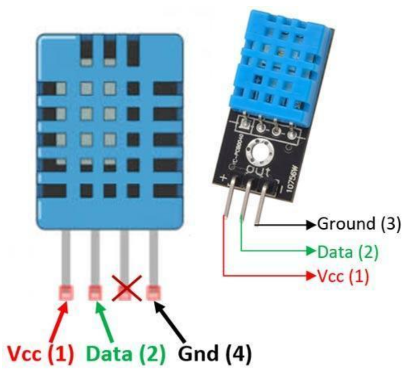
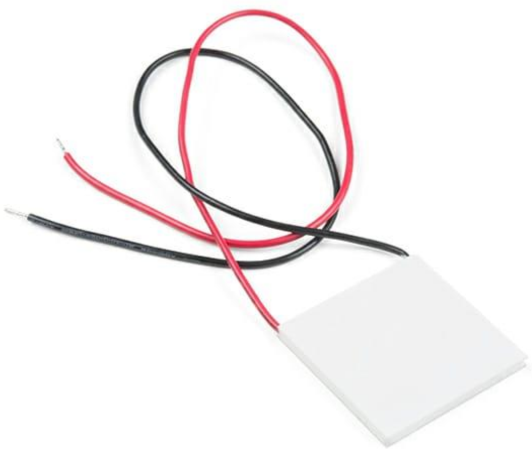


Fig 6.22 Pin Configuration for DHT11

6.5.4 Applications

- This sensor is used in various applications such as measuring humidity and temperature values in heating, ventilation and air conditioning systems.
- Weather stations also use these sensors to predict weather conditions.
- The humidity sensor is used as a preventive measure in homes where people are affected by humidity.
- Offices, cars, museums, greenhouses and industries use this sensor for measuring humidity values and as a safety measure.

6.6 PELTIER SENSOR



6.23 Peltier Module

To start with, let's define what a thermoelectric device is. A thermoelectric device is a type of device that can convert heat into electricity, or vice versa. It does this by exploiting the thermoelectric effect, which is the phenomenon where a temperature difference between two materials creates a voltage difference between them. This voltage difference can be used to generate electricity, or it can be used to create a temperature difference between the two materials.

Peltier devices are a type of thermoelectric device that use the Peltier effect, which is the inverse of the Seebeck effect. The Seebeck effect is the phenomenon where a voltage difference between two materials creates a temperature difference between them. The Peltier effect, on the other hand, is the phenomenon where a temperature difference between two materials creates a voltage difference between them.

Peltier devices are made up of two different materials that are joined together. These materials are typically n-type and p-type semiconductors. When a current is applied to the device, electrons move from the n-type material to the p-type material, creating a temperature difference between the two materials. This temperature difference can be used to cool or heat the device, depending on the direction of the current.

6.6.1 Advantages

- Fast response time,
    - Because they use the Seebeck effect, they can respond quickly to changes in temperature, making them useful in applications where rapid temperature changes need to be monitored.
- Ability to cool or heat small spaces with high precision,
    - Because the temperature difference created by the device is proportional to the current applied to it, the temperature can be controlled very precisely by varying the current.

6.6.2 Disadvantages

- Low efficiency,
  - Because they are made up of two different materials that have different thermal conductivities, a lot of heat is generated at the junction between the two materials. This heat can reduce the efficiency of the device, making it less effective at cooling or heating.
- Limited temperature range,
  - They are typically only effective at temperatures up to a few hundred degrees Celsius. Above this temperature, the materials used in the device can begin to break down, reducing its effectiveness.

6.6.3 Applications

- One of the most common applications of Peltier devices is in thermoelectric cooling,
  - In this application, the Peltier device is used to cool a small space, such as a computer processor or a wine cooler.
  - The device is placed between the hot side and the cold side of the space, and a current is applied to it.
  - This current creates a temperature difference between the two sides of the device, causing heat to be absorbed from the hot side and released to the cold side. This cools the space.
- Peltier devices can also be used for temperature sensing,
  - In this application, the device is used as a thermocouple, which is a type of temperature sensor that uses the Seebeck effect.
  - The Peltier device is placed between two conductors, and a current is applied to it.

- The resulting voltage difference is measured, and the temperature can be calculated from the voltage difference using a calibration curve.

- They are particularly useful in situations where small spaces need to be cooled or heated with high.

CHAPTER 7

SOFTWARE REQUIREMNT

The following software components are,

- Arduino Software
- Embedded C

7.1 ARDUINO SOFTWARE (IDE)

- Writing Sketches
    - File
    - Edit
    - Sketch
    - Tools
    - Help
- Sketchbook
- Tabs, Multiple Files and Compilation
- Uploading
- Libraries
- Third Party Hardware
- Serial Monitor
- Preferences
- Language Support
- Boards

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuino hardware to upload programs and communicate with them.

7.1.1 Writing Sketches

Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension .ino. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom righthand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

NB: Versions of the Arduino Software (IDE) prior to 1.0 saved sketches with the extension .pde. It is possible to open these files with version 1.0, you will be prompted to save the sketch with the .ino extension on save.

Additional commands are found within the five menus: File, Edit, Sketch, Tools, Help. The menus are context sensitive, which means only those items relevant to the work currently being carried out are available..

7.1.2 Sketchbook

The Arduino Software (IDE) uses the concept of a sketchbook: a standard place to store your programs (or sketches). The sketches in your sketchbook can be opened from the File > Sketchbook menu or from the Open button on the toolbar. The first time you run the Arduino software, it will automatically create a directory for your sketchbook. You can view or change the location of the sketchbook location from with the Preferences dialog.

Beginning with version 1.0, files are saved with a .ino file extension. Previous versions use the .pde extension. You may still open .pde named files in version 1.0 and later, the software will automatically rename the extension to .ino.

7.1.3 Libraries

Libraries provide extra functionality for use in sketches, e.g. working with hardware or manipulating data. To use a library in a sketch, select it from the Sketch > Import Library menu. This will insert one or more #include statements at the top of the sketch and compile the library with your sketch. Because libraries are uploaded to the board with your sketch, they increase the amount of space it takes up. If a sketch no longer needs a library, simply delete its #includestatements from the top of your code.

There is a list of libraries in the reference. Some libraries are included with the Arduino software. Others can be downloaded from a variety of sources or through the Library Manager. Starting with version 1.0.5 of the IDE, you do can import a library from a zip file and use it in an open sketch. See these instructions for installing a third-party library.

7.1.4 Language Support

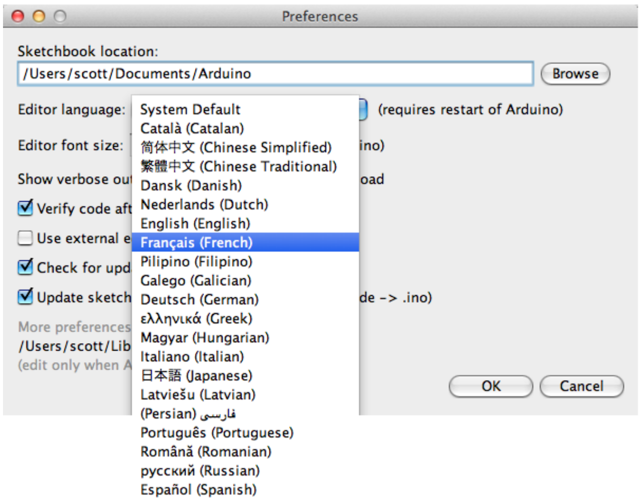


Fig 7.1 Language Support

Since version 1.0.1 , the Arduino Software (IDE) has been translated into 30+ different languages. By default, the IDE loads in the language selected by your operating system. (Note: on Windows and possibly Linux, this is determined by the locale setting which controls currency and date formats, not by the language the operating system is displayed)

If you would like to change the language manually, start the Arduino Software (IDE) and open the Preferences window. Next to the Editor Language there is a dropdown menu of currently supported languages. Select your preferred language from the menu, and restart the software to use the selected language. If your operating system language is not supported, the Arduino Software (IDE) will default to English.

You can return the software to its default setting of selecting its language based on your operating system by selecting System Default from the Editor Language drop-down. This setting will take effect when you restart the Arduino Software (IDE). Similarly, after changing your operating system's settings, you must restart the Arduino Software (IDE) to update it to the new default language.

7.1.5 Boards

The board selection has two effects: it sets the parameters (e.g. CPU speed and baud rate) used when compiling and uploading sketches; and sets and the file and fuse settings used by the burn bootloader command. Some of the board definitions differ only in the latter, so even if you've been uploading successfully with a particular selection you'll want to check it before burning the bootloader. You can find a comparison table between the various boards here. Arduino Software (IDE) includes the built in support for the boards in the following list, all based on the AVR Core. The Boards Manager included in the standard installation allows to add support for the growing number of new boards based on different cores like Arduino Due, Arduino Zero, Edison, Galileo and so on.

- Arduino Yùn
  - An ATmega32u4 running at 16 MHz with auto-reset, 12 Analog In, 20 Digital I/O and 7 PWM.
- Arduino/Genuino Uno
  - An ATmega328 running at 16 MHz with auto-reset, 6 Analog In, 14 Digital I/O and 6 PWM.
- Arduino Diecimila or Duemilanove w/ ATmega168
  - An ATmega168 running at 16 MHz with auto-reset.
- Arduino Nano w/ ATmega328
  - An ATmega328 running at 16 MHz with auto-reset. Has eight analog inputs.
- Arduino/Genuino Mega 2560
  - An ATmega2560 running at 16 MHz with auto-reset, 16 Analog In, 54 Digital I/O and 15 PWM.
- Arduino Mega
  - An ATmega1280 running at 16 MHz with auto-reset, 16 Analog In, 54 Digital I/O and 15 PWM.
- Arduino Mega ADK
  - An ATmega2560 running at 16 MHz with auto-reset, 16 Analog In, 54 Digital I/O and 15 PWM.
- Arduino Leonardo
  - An ATmega32u4 running at 16 MHz with auto-reset, 12 Analog In, 20 Digital I/O and 7 PWM.
- Arduino/Genuino Micro
  - An ATmega32u4 running at 16 MHz with auto-reset, 12 Analog In, 20 Digital I/O and 7 PWM.

- Arduino Esplora
  - An ATmega32u4 running at 16 MHz with auto-reset.
- Arduino Mini w/ ATmega328
  - An ATmega328 running at 16 MHz with auto-reset, 8 Analog In, 14 Digital I/O and 6 PWM.
- Arduino Ethernet
  - Equivalent to Arduino UNO with an Ethernet shield: An ATmega328 running at 16 MHz with auto-reset, 6 Analog In, 14 Digital I/O and 6 PWM.
- Arduino Fio
  - An ATmega328 running at 8 MHz with auto-reset. Equivalent to Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega328, 6 Analog In, 14 Digital I/O and 6 PWM.
- Arduino BT w/ ATmega328
  - ATmega328 running at 16 MHz. The bootloader burned (4 KB) includes codes to initialize the on-board bluetooth module, 6 Analog In, 14 Digital I/O and 6 PWM..
- LilyPad Arduino USB
  - An ATmega32u4 running at 8 MHz with auto-reset, 4 Analog In, 9 Digital I/O and 4 PWM.
- LilyPad Arduino
  - An ATmega168 or ATmega132 running at 8 MHz with auto-reset, 6 Analog In, 14 Digital I/O and 6 PWM.
- Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega328
  - An ATmega328 running at 16 MHz with auto-reset. Equivalent to Arduino Duemilanove or Nano w/ ATmega328; 6 Analog In, 14 Digital I/O and 6 PWM.
- Arduino NG or older w/ ATmega168

    - An ATmega168 running at 16 MHz without auto-reset. Compilation and upload is equivalent to Arduino Diecimila or Duemilanove w/ ATmega168, but the bootloader burned has a slower timeout (and blinks the pin 13 LED three times on reset); 6 Analog In, 14 Digital I/O and 6 PWM.
- Arduino Robot Control
    - An ATmega328 running at 16 MHz with auto-reset.
- Arduino Robot Motor
    - An ATmega328 running at 16 MHz with auto-reset.
- Arduino Gemma
    - An ATtiny85 running at 8 MHz with auto-reset, 1 Analog In, 3 Digital I/O and 2 PWM.

7.2 EMBEDDED C

Embedded C is a programming language that is widely used in the field of embedded systems development. It is a variant of the C programming language that is specifically designed for use in embedded systems. An embedded system is a computer system that is designed to perform a specific task within a larger system. Examples of embedded systems include medical devices, automotive systems, and consumer electronics devices.

Embedded C is a high-level language that is easy to learn and use. It provides the ability to write programs for microcontrollers and other embedded systems in a high-level language, which makes the development process faster and more efficient. In this article, we will provide a brief explanation of Embedded C, its features, and how it is used in embedded systems development.

7.2.1 Features

Embedded C has a number of features that make it an ideal choice for programming embedded systems. These features include:

- Low-level Access: Embedded C provides access to low-level hardware resources such as memory-mapped I/O ports, registers, and interrupts. This allows developers to write code that interacts directly with the hardware, which is essential for developing efficient and reliable embedded systems.
- Efficient Code Execution: Embedded C is designed to generate efficient code that can run on resource-constrained devices such as microcontrollers. This makes it possible to write programs that consume minimal resources and execute quickly.
- Small Memory Footprint: Embedded C is designed to generate code that has a small memory footprint. This is important for embedded systems, where memory is often limited and expensive.
- Portability: Embedded C is portable across different platforms and architectures. This makes it possible to write code that can be easily ported to different hardware platforms.

7.2.2 Applications

Embedded C is used extensively in embedded systems development. It is used to write software for microcontrollers and other embedded devices. Some of the common uses of Embedded C in embedded systems development include:

- Device Drivers: Device drivers are software programs that allow the operating system to communicate with hardware devices. Embedded C is used to write device drivers for microcontrollers and other embedded devices.

- Real-Time Operating Systems: Real-time operating systems (RTOS) are used in embedded systems to manage the execution of tasks in real-time. Embedded C is used to write code for RTOS kernels.

- Control Systems: Embedded C is used to write code for control systems in embedded devices such as automotive systems and medical devices.

- Networking: Embedded C is used to write code for networking protocols such as TCP/IP, which are used in embedded systems to communicate with other devices.

Embedded C is a powerful programming language that is widely used in embedded systems development. Its features, such as low-level access, efficient code execution, small memory footprint, portability, and ease of use, make it an ideal choice for developing software for microcontrollers and other embedded devices. As the demand for embedded systems continues to grow, Embedded C will continue to be a key tool in the development of efficient and reliable embedded systems.

CHAPTER 8

RESULT OF THE PROTOTYPE

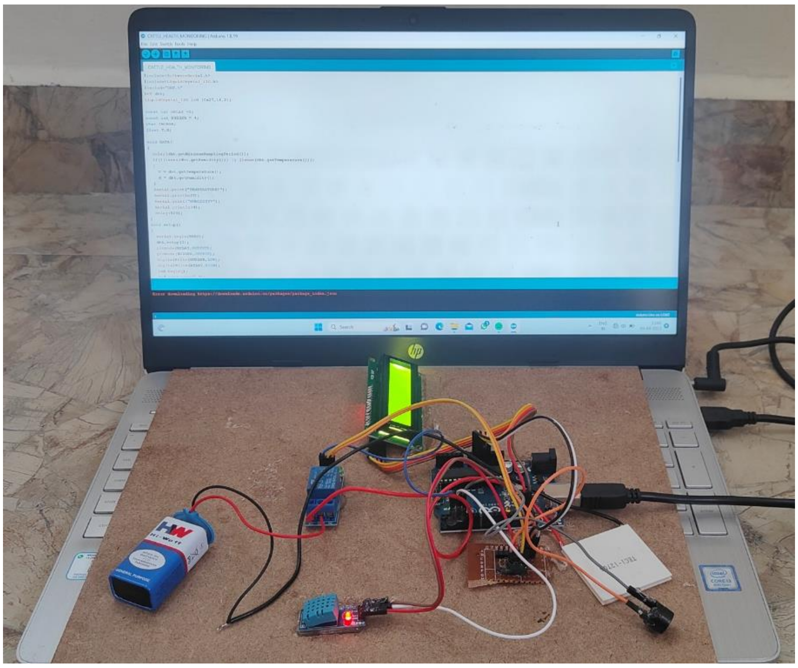


Fig 8.1 Image of the Prototype

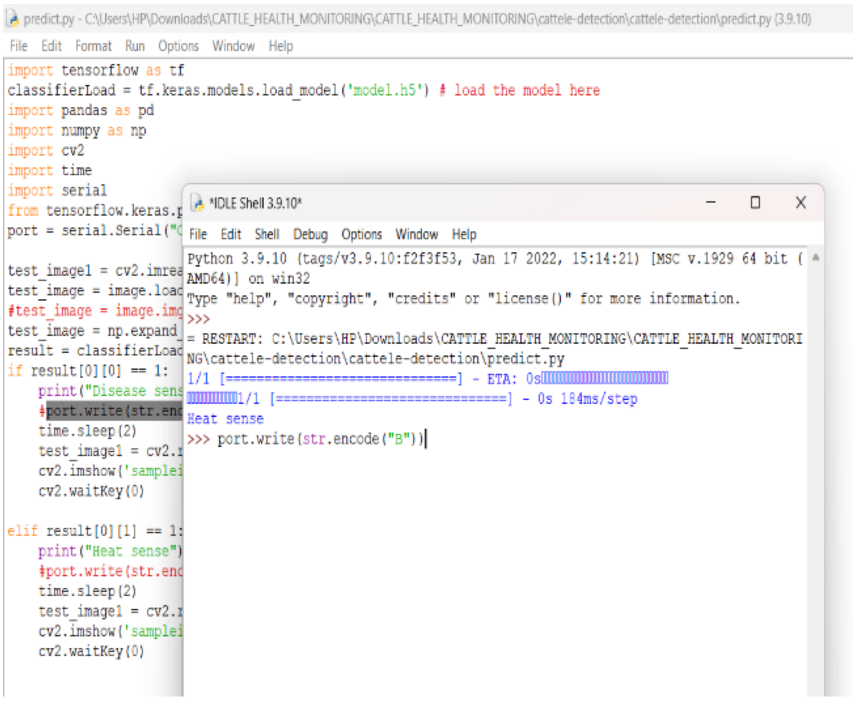


Fig 8.2 Input for Heat Sense

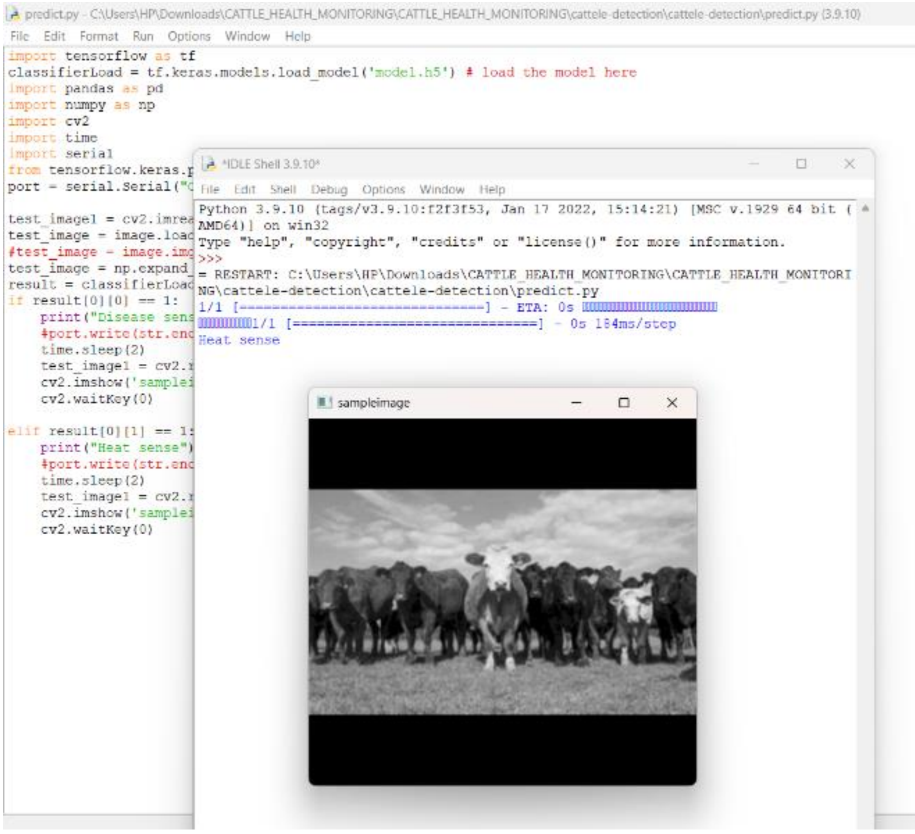


Fig 8.3 Loading Sample Image

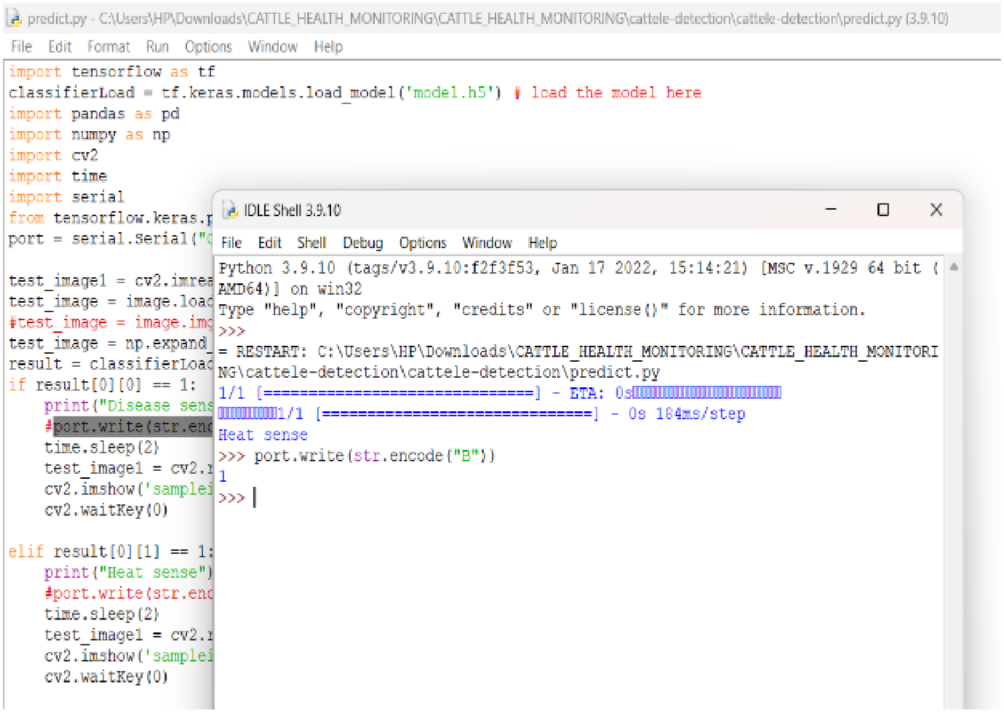


Fig 8.4 Final Output for Heat Sense

8.1 OUTPUT

This code is using a trained model to predict whether an image is showing symptoms of a disease or heat sense. The code imports the necessary libraries, including TensorFlow, which is used for loading the trained model. The trained model is loaded from the file 'model.h5' using the Keras load_model function, and stored in the variable 'classifierLoad'. Then, the code imports Pandas and NumPy for data handling, OpenCV (cv2) for image processing, time for waiting, and serial for serial communication. The code also imports the image preprocessing module from TensorFlow Keras. The code then loads an image '1.png' using cv2.imread() to show it later as a sample image. It also loads the same image using image.load_img() and resizes it to 300x300 pixels. Then it expands the dimensions of the image to match the input shape of the model.

The image is passed through the loaded model using predict() function to get a prediction result. If the model predicts that the image is showing symptoms of a disease, the code prints "Disease sense" and displays the image using cv2.imshow(). It waits for a key press with cv2.waitKey(0) and then exits. If the model predicts that the image is showing heat sense, the code prints "Heat sense" and shows the image with cv2.imshow(). Additionally, the code includes comments to explain each step of the process, including loading the model, loading and preprocessing the image, and displaying the result. The code also includes commented out code to send a signal to a serial port based on the prediction result.

CHAPTER 9

CONCLUSION

In conclusion, the proposed system for cattle health monitoring using AI is a game-changing solution that has the potential to revolutionize the livestock industry. By leveraging the power of machine learning and image processing technologies, the system can accurately detect signs of illness or disease in cattle in real-time, allowing farmers and livestock managers to take immediate action to prevent the spread of disease and maintain optimal health conditions for the cattle. The system's automated temperature maintenance, based on the count of cattle in the room, ensures that the cattle are kept in optimal temperature conditions, which is essential for their health and well-being. The system's ability to provide real-time monitoring of the cattle's health status, combined with its automated temperature maintenance, offers significant advantages over traditional livestock management practices.

Overall, the proposed system for cattle health monitoring using AI offers a promising solution for improving livestock management practices and enhancing the health and welfare of cattle. Its ability to detect signs of illness in real-time and automate temperature maintenance offers significant advantages over traditional livestock management practices. Therefore, the proposed system has the potential to transform the livestock industry, promoting better animal health and improving the profitability of livestock management practices.

9.1 FUTURE SCOPE

One of the future works for the proposed system is the integration of advanced machine learning techniques. Currently, the system uses image processing algorithms to identify and compare the real-time cattle picture and diseased cattle picture. However, the integration of advanced machine learning techniques such as deep learning algorithms can significantly improve the accuracy of the system. Deep learning algorithms can help the system to detect patterns in the data and make more accurate predictions, which will help in identifying the early signs of disease in cattle and prevent the spread of the disease.

In addition, the proposed system can be expanded to include other types of livestock such as pigs, sheep, and goats. By expanding the system to include other types of livestock, farmers can have a comprehensive solution to monitor the health of their livestock.

Furthermore, the proposed system can be integrated with a mobile application that allows farmers to monitor the health status of their cattle remotely. The mobile application can provide real-time updates on the health of the cattle, and farmers can use this information to make informed decisions about their livestock.

Lastly, the proposed system can be integrated with automatic feeding and watering systems. These systems can automatically provide feed and water to the cattle, and the system can monitor the consumption of feed and water by the cattle. This can help farmers to monitor the health of the cattle more effectively and prevent overfeeding or underfeeding.

In conclusion, the proposed system of cattle health monitoring using AI is a promising solution for cattle farmers. However, there are still areas where the system can be further improved to make it more effective and efficient. The integration of advanced machine learning techniques, the expansion of the system to include other types of livestock, integration with a mobile application, and integration with automatic feeding and watering systems are some of the future works for the proposed system that can make it more effective and efficient.

REFERENCES

1. Shukla, R., Aggarwal, A., & Singh, S. (2021). Cattle health monitoring using deep learning and internet of things. International Journal of Computer Science and Mobile Computing, 10(8), 46-53.
2. Kaur, P., & Jain, N. (2020). Automated cattle monitoring and healthcare system using image processing and IoT. Journal of Ambient Intelligence and Humanized Computing, 11(11), 5047-5055.
3. Parvin, S., Haque, M. A., & Hasan, M. R. (2020). Real-time health monitoring system for cattle using image processing and machine learning. In 2020 IEEE International Conference on Informatics, Electronics and Vision (ICIEV) (pp. 1536-1541). IEEE.
4. Wu, Y., Yu, H., Tan, X., & Chen, G. (2021). A smart cattle health monitoring system based on image processing and artificial intelligence. Computers and Electronics in Agriculture, 186, 106005.
5. Zhang, Y., Du, Q., Wei, J., & Liu, Q. (2020). Research on cattle health monitoring system based on image processing and IoT technology. In Proceedings of the 2020 International Conference on Intelligent Autonomous Systems (pp. 49-54). ACM.
6. Zhang, Y., Du, Q., Wei, J., & Liu, Q. (2021). Design of a smart cattle health monitoring system based on image processing and IoT technology. IEEE Internet of Things Journal, 8(8), 6764-6774.
7. Tadesse, M., & Nega, A. (2021). Machine learning-based real-time cattle health monitoring system using thermal imaging and IoT. In 2021 International Conference on Advancements in Computing and Management (ICACM) (pp. 259-265). IEEE.

8. Huang, H., Sun, Y., & Wang, Y. (2021). A cattle health monitoring system based on convolutional neural network and support vector machine. Journal of Ambient Intelligence and Humanized Computing, 12(8), 8421-8431.
9. Arya, S., & Goel, S. (2020). Real-time health monitoring of cattle using image processing and deep learning techniques. In 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1-5). IEEE.
10. Liu, Q., Du, Q., Zhang, Y., & Wang, K. (2021). Cattle health monitoring system based on IoT and image processing. Journal of Sensors, 2021, 6611941.

APPENDIX

1.Code for the Prototype

```
# Part 1 - Building the CNN

import matplotlib.pyplot as plt

# Importing the Keras libraries and packages

from keras.models import Sequential

from keras.layers import Convolution2D

from keras.layers import MaxPooling2D

from keras.layers import Flatten

from keras.layers import Dense

from keras.models import model_from_json


batch_size = 1


from tensorflow.keras.preprocessing.image import ImageDataGenerator


# All images will be rescaled by 1./255

train_datagen = ImageDataGenerator(rescale=1/255)


# Flow training images in batches of 128 using train_datagen generator

train_generator = train_datagen.flow_from_directory(

        'dataset',# This is the source directory for training images
```

```
    target_size=(300,300), # All images will be resized to 200 x 200
    batch_size = batch_size,
    # Specify the classes explicitly
    classes = ['cattle_diseased','cattle_group'],
    # Since we use categorical_crossentropy loss, we need categorical labels
    class_mode='categorical')


import tensorflow as tf


model = tf.keras.models.Sequential([
  # Note the input shape is the desired size of the image 200x 200 with 3 bytes
color
  # The first convolution
  tf.keras.layers.Conv2D(16, (3,3), activation='relu', input_shape=(300,300, 3)),
  tf.keras.layers.MaxPooling2D(2, 2),
  # The second convolution
  tf.keras.layers.Conv2D(32, (3,3), activation='relu'),
  tf.keras.layers.MaxPooling2D(2,2),
  # The third convolution
  tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
  tf.keras.layers.MaxPooling2D(2,2),
  # The fourth convolution
```

```
    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    # The fifth convolution
    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    # Flatten the results to feed into a dense layer
    tf.keras.layers.Flatten(),
    # 128 neuron in the fully-connected layer
    tf.keras.layers.Dense(128, activation='relu'),
    # 2 output neurons for 2 classes with the softmax activation
    tf.keras.layers.Dense(2, activation='softmax')
])

model.summary()

from tensorflow.keras.optimizers import Adam

model.compile(loss='categorical_crossentropy',
              optimizer=Adam(lr=0.001),
              metrics=['acc'])

total_sample = train_generator.n
```

```
print(total_sample)

n_epochs = 30

hist = model.fit(
    train_generator,
    steps_per_epoch=int(total_sample/batch_size),
    epochs=n_epochs,
    verbose=1)

model.save('model.h5')

history_dict = hist.history
print(history_dict.keys())

plt.figure()
plt.title("Accuracy")
plt.plot(hist.history['acc'], 'r', label='Training')
plt.legend()
plt.show()

plt.figure()
```

```
plt.title("Loss")

plt.plot(hist.history['loss'], 'r', label='Training')

plt.legend()

plt.show()

# For Image Processing

import tensorflow as tf

classifierLoad = tf.keras.models.load_model('model.h5') # load the model here

import pandas as pd

import numpy as np

import cv2

import time

import serial

from tensorflow.keras.preprocessing import image

port = serial.Serial("COM19",9600,timeout = 0.1)


test_image1 = cv2.imread('1.png',0)

test_image = image.load_img('1.png', target_size = (300,300))  # load the
sample image here

#test_image = image.img_to_array(test_image)

test_image = np.expand_dims(test_image, axis=0)

result = classifierLoad.predict(test_image)

if result[0][0] == 1:
```

```
    print("Disease sense")
    #port.write(str.encode("B"))
    time.sleep(2)
    test_image1 = cv2.resize(test_image1, (380,360))
    cv2.imshow('sampleimage',test_image1)
    cv2.waitKey(0)

elif result[0][1] == 1:
    print("Heat sense")
    #port.write(str.encode("A"))
    time.sleep(2)
    test_image1 = cv2.resize(test_image1, (380,360))
    cv2.imshow('sampleimage',test_image1)
    cv2.waitKey(0)
# For Arduino
#include<SoftwareSerial.h>
#include<LiquidCrystal_I2C.h>
#include"DHT.h"
DHT dht;
LiquidCrystal_I2C lcd (0x27,16,2);

const int RELAY =2;
```

```
const int BUZZER = 4;
char INCHAR;
float T,H;
void DATA()
{
  delay(dht.getMinimumSamplingPeriod());
  if(!(isnan(dht.getHumidity())) || (isnan(dht.getTemperature())))
  {
    T = dht.getTemperature();
    H = dht.getHumidity();
  }
  Serial.print("TEMPERATURE=");
  Serial.println(T);
  Serial.print("HUMIDITY=");
  Serial.println(H);
  delay(500);
}
void setup()
{
  Serial.begin(9600);
  dht.setup(3);
  pinMode(RELAY,OUTPUT);
```

```
  pinMode(BUZZER,OUTPUT);
  digitalWrite(BUZZER,LOW);
  digitalWrite(RELAY,HIGH);
  lcd.begin();
  lcd.setCursor(0,0);
  lcd.print("CATTEL HEALTH");
  lcd.setCursor(0,1);
  lcd.print("MONITORING");
  delay(2000);
  lcd.clear();
}

void loop()
{
  DATA();
  if (Serial.available    {
    INCHAR = Serial.read();
    if (INCHAR == 'A')
    {
      if (T>60)
      {
        lcd.clear();
```

```
      lcd.setCursor(0,0);
      lcd.print("TOO HEAT...");
      digitalWrite(RELAY,HIGH);
     }
     else
     {
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("TOO COOL..");
      digitalWrite(RELAY,LOW);
     }
    }
    if (INCHAR == 'B')
    {
     lcd.clear();
     lcd.setCursor(0,0);
     lcd.print("DISEASED CATTEL");
     //Serial.println("DISEASE CATTLE");
     digitalWrite(BUZZER,HIGH);
    }
  }
}
```