

***Plasmodium falciparum* DETECTION IN MICROSCOPIC BLOOD CELL IMAGES TO AVOID SUDDEN DEATH OF PEOPLE FROM MALARIA**

Submitted in partial fulfilment of the requirements for the award of
Bachelor of Technology degree in Biomedical Engineering

By

AISHWARYA J (37240002)

SOUNDHARYA B (37240072)



**DEPARTMENT OF BIOMEDICAL ENGINEERING
SCHOOL OF BIO AND CHEMICAL ENGINEERING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

MARCH-2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600 119

www.sathyabama.ac.in

DEPARTMENT OF BIOMEDICAL ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **AISHWARYA J (37240002)** and **SOUNDHARYA B (37240072)** who carried out the project entitled **"*Plasmodium falciparum* DETECTION IN MICROSCOPIC BLOOD CELL IMAGES TO AVOID SUDDEN DEATH OF PEOPLE FROM MALARIA"** under my supervision from September 2020 to March 2021.

MR. G. UMASHANKAR, M.Tech., (Ph.D.)

Internal Guide

Dr. T. SUDHAKAR, M.sc., Ph.D.,

Head of the Department

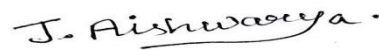
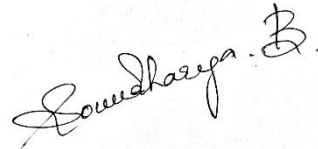
Submitted for Viva voce Examination held on **09.04.2021**

Internal Examiner

External Examiner

DECLARATION

We, **AISHWARYA J (37240002)** and **SOUNDHARYA B (37240072)** hereby Declare that the Project Report entitled “*Plasmodium falciparum* DETECTION IN MICROSCOPIC BLOOD CELL IMAGES TO AVOID SUDDEN DEATH OF PEOPLE FROM MALARIA” done by us under the guidance of **MR. G. UMASHANKAR, M.Tech., (Ph.D.)** is submitted in partial fulfilment of the requirements for the award of Bachelor of Technology degree in Biomedical Engineering.

Handwritten signature of J. Aishwarya in black ink.Handwritten signature of Soundharya B. in black ink.

DATE: 09.04.2021

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to **Board of Management of SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr. J. Premkumar**, M.Sc., Ph.D., and **Dr. T. SUDHAKAR**, M.Sc., Ph.D., **Head of the Department, Dept. of Biomedical Engineering** for providing us necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our Project Guide **Mr. G. UMASHANKAR**, M.Tech., (Ph.D.), **Department of Biomedical Engineering** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of our project work.

We wish to express our thanks to all **Teaching and Non-teaching** staff members of the **Department of Biomedical Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Malaria continues to be a significant threat to global health, with an estimated 200 million cases and over 400,000 deaths per year. Aside from scientific science and government activities, modern information technology is a critical component of many disease-fighting efforts. Inadequate malaria diagnosis has been one of the major roadblocks to effective mortality reduction. Image processing software and machine learning methods were used to classify *Plasmodium falciparum* in microscopic blood slides to improve diagnosis. This article addresses recent advances in image processing and machine learning for microscopic malaria diagnosis and provides an overview of these techniques. The various approaches published in the literature are organised according to image pre-processing, parasite identification and cell segmentation, feature computation, and automated cell classification are some of the imaging techniques used. Then, with the help of a microcontroller, the result is modified in IOT and an SMS is sent to the patient's designated individual via GSM.

Keywords – Plasmodium, IOT, GSM

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
-	ABSTRACT	i
-	LIST OF ABBREVIATION	v
-	LIST OF FIGURES	vi
-	LIST OF TABLES	vii
CHAPTER 1	INTRODUCTION	1
CHAPTER 2	LITERATURE SURVEY	1
CHAPTER 3	AIM AND OBJECTIVE	4
3.1	Aim	4
3.2	Objective	5
3.3	Existing system and proposed system	5
3.4	Working principle	6
CHAPTER 4	METHODOLOGY	6
4.1	Methodology block diagram	6
4.2	Hardware requirements	7
4.2.1	Arduino UNO	7
4.2.1.1	Introduction to Arduino	7
4.2.1.2	Programming	10
4.2.1.3	Applications	12
4.2.2	Heart beat sensor	12
4.2.2.1	Specifications	13
4.2.2.2	Features	13
4.2.3	Temperature sensor	15

4.2.3.1	LM35 temperature sensor circuit configuration	16
4.2.3.2	LM35 working principle	17
4.2.3.3	Features	17
4.2.3.4	Functional block diagram	18
4.2.3.5	LM35 temperature sensor applications	18
4.2.4	Respiratory sensor	18
4.2.4.1	Specifications	19
4.2.4.2	How to test	19
4.2.5	Liquid Crystal Display (LCD)	20
4.2.5.1	16X2 LCD pinout diagram	21
4.2.6	Global System for Mobile communication (GSM)	23
4.2.6.1	SIM900 GSM/GPRS module	24
4.2.6.2	Features	24
4.2.6.3	Working	25
4.2.6.4	AT commands	25
4.2.7	Buzzer	25
4.2.7.1	Introduction to Piezo buzzer	26
4.2.7.2	Specifications	26
4.2.8	Internet of Things (IoT)	27
4.2.8.1	Application areas	28
4.3	ESP-12E based node MCU	29
4.3.1	ESP-12E Architecture	30
4.3.2	Features	31
4.4	Hardware setup of our project	32

4.5	Software requirements	33
4.5.1	MATLAB	33
4.5.1.2	Features of MATLAB	34
4.5.2	Embedded C	35
4.5.2.1	Embedded system programming	36
4.5.2.2	Embedded system	36
4.5.2.3	Characteristics of embedded system	37
4.5.2.4	Advantages	38
4.5.2.5	Disadvantages	38
4.5.3	Arduino software (IDE)	38
4.5.3.1	Writing sketches	39
CHAPTER 5	RESULT AND DISCUSSION	39
5.1	Result	39
5.2	Discussion	49
CHAPTER 6	SUMMARY AND CONCLUSION	5
6.1	Summary	50
6.2	Conclusion	50
-	REFERENCES	51
-	APPENDIX	53
-	Hardware Program	53
-	Software Program	58

LIST OF ABBREVIATIONS

MATLAB	MATrix LABoratory
IDE	Integrated Development Environment
LCD	Liquid Crystal Display
IOT	Internet Of Things
MCU	Multi point Control Unit
LED	Light Emitting Diode
GSM	Global System for Mobile Communication

LIST OF FIGURES

FIGURE NO	NAME OF THE FIGURE	PAGE NO.
4.1	Whole process block diagram	6
4.2	Arduino Board	8
4.3	Pin configuration of Arduino board	10
4.4	Arduino programming	11
4.5	Heart beat sensor	12
4.6	Connection of KY-039	14
4.7	Temperature sensor (LM35)	15
4.8	Circuit configuration	16
4.9	Block diagram	18
4.10	Schematic connection diagram	20
4.11	Respiratory sensor	20
4.12	LCD	21
4.13	LCD pinout diagram	21
4.14	LCD pin configuration	22
4.15	GSM	24
4.16	Buzzer	26
4.17	Structure of buzzer	26
4.18	ESP-12E based node MCU	29
4.19	ESP-12E architecture	31
4.20	Hardware setup	32

4.21	Block diagram of embedded C	35
4.22	Progress in block diagram of embedded C	36
4.23	3 main elements of embedded device	37
4.24	Arduino software IDE	38
5.1	Output images	39 – 48
5.2	Output shown in IoT, LCD display and GSM	49

LIST OF TABLES

TABLE NO	NAME OF THE TABLE	PAGE NO.
4.1	Technical specifications	9

CHAPTER 1

INTRODUCTION

Malaria is a fatal, infectious disease caused by the Plasmodium parasite and spread by female Anopheles mosquito bites. According to the WHO's World Malaria Report 2019 [1] an estimated 405,000 people died from malaria in the previous year. The disease can be cured, but early detection is crucial. Malaria can be detected using a variety of techniques, including microscopic examinations. Infected cells are detected in a laboratory. The procedure is both costly and time-consuming. Malaria cases in the WHO African region accounted for 93 percent of all cases registered in 2018. In addition, the country has one of the lowest per capita incomes in the world. This model proposes a quicker, less expensive, and more accurate alternative to microscopic Malaria detection.

CHAPTER 2

LITERATURE SURVEY

Kishor Roy et al. (2018) proposed system model which is implemented by using two Segmentation that are HSV segmentation and watershed segmentation. The result of both the segmentations are mapped so that all the malarial parasites can be counted. This is done to improve accuracy.

Silvia et al. (2017) proposed a technique for estimating parasitemia. Template matching is used for detection of RBCs. Parasites are detected using variance-based technique from grayscale images and second approach is based on color co-occurrence matrix.

Ravi raja et al. (2018) the design takes the form of a standard pattern recognition and classification system. The system architecture of malaria parasite detection includes grey scale image conversion, thresholding, thinning labelling algorithm. To detect the infected cells, color range and image segmentation techniques are there from Giemsa-stained peripheral blood samples.

Aimi Salihah Abdul Nasar et al. (2016) this paper shows that the detection of malaria

parasite can be also done through color image segmentation which can be further applied on malaria images of *P. vivax* species. The use of partial contrast stretching is introduced here in order to obtain segmented RBC infected within the malaria parasite. After that an unsupervised segmentation technique is used which is known as k-means clustering. It segments the infected cell from background different color components of RGB, HSI and C-Y color modes needs to be checked to identify color components.

Rabbani et al. (2013) thin blood smears are used in this study and is mainly based on curve fitting to detect the infected cells in the blood smear. The method is composed of six phases- stain object extraction step, pre-processing phase which uses nonlinear diffusion filtering followed by detection of parasite nucleus from the result of previous step according to the image intensity, fourth step is to introduce an complete search process in which circle search step in defines the direction and initial points for direct least square ellipse fitting algorithm, further more ellipse searching process is applied finally a series of decision rules on the presence of infected cells in the blood smear.

Vipul Parkhi et al. (2014) mathematical Modelling technique is used by means of LP as an efficient tool to solve problems related to malaria parasites problems related to malaria diagnosis through microscopy imaging problems.

Bashir et al. (2017) the method made use of the intensity features of Plasmodium parasites and erythrocytes. Images of infected and non-infected erythrocytes were acquired, pre-processed, relevant features extracted from them and eventually diagnosis was made based on the features extracted from the images. A set of features based on intensity have been proposed, and the performance of these features on the red blood cell samples from the created database have been evaluated using an artificial neural network (ANN) classifier.

Matthias Elter et al. (2011) Malaria, caused by a blood parasite of the genus

plasmodium, kills millions of people each year. According to the World Health Organization, the standard for malaria diagnosis is microscopic examination of a stained blood film. We have developed a two-stage algorithm for the automatic detection of plasmodia in thick blood films. The focus of the first stage is on high detection sensitivity while accepting high numbers of false-positive detections per image. The second stage reduces the number of false-positive detections to an acceptable level while maintaining the detection sensitivity of the first stage. The algorithm can detect plasmodia at a sensitivity of 0.97 with a mean number of 0.8 false-positive detections per image. Our results indicate that the proposed algorithm is suitable for the development of an automated microscope for computer-aided malaria screening.

Seungjun Lee et al. (2011) to develop non-invasive diagnosis of malaria, we studied the light-scattering pattern of healthy and 2% *Plasmodium falciparum* malaria parasitized red blood cells. By measuring the wavelength-dependent scattering at discrete angles of both forward and backward directions, we found that the signal can clearly distinguish between healthy and ring stage infected red blood cells. In particular, we demonstrated the first backward scattering measurement, which enables elastic light scattering as a promising non-invasive diagnostic tool for malaria.

Yingjian Liu et al. (2017) Internet of Things (IoT) can provide interconnection of things, which can promote the achievement of the remote human health monitoring. In this paper, a human health monitoring platform is designed and developed under the application framework of body sensor network of IoT. The platform can collect the physical information of user by constructing the human state acquisition system. And then the collected human physiological data is transmitted to data processing platform through ZigBee wireless network for further data processing, preservation and display, with which real-time rescue and treatment can be sent out by guardian, doctors and healthcare caregivers.

Mohammad Salah Uddin et al. (2017) the popularity of Internet of Things is increasing day by day in the area of remote monitoring system. The remote monitoring systems include, vehicle or assets monitoring, kids/pets monitoring, fleet management, parking management, water and oil leakage, energy grid monitoring etc. In this paper, we have

proposed an intelligent patient monitoring system for monitoring the patients' health condition automatically through sensors based connected networks. Several sensors are used for gathering the biological behaviors of a patient. The meaningful biological information is then forwarded to the IoT cloud. The system is more intelligent that can able to detect the critical condition of a patient by processing sensors data and instantly provides push notification to doctors/nurses as well as hospital in-charge personal. The doctors and nurses get benefited from this system by observing their corresponding patients remotely without visiting in person. Patients' relatives can also get benefited from this system with limited access.

CHAPTER 3

AIM AND OBJECTIVE

3.1 AIM

We propose a model for detecting malaria-infected cells based on image processing. In thin smears on regular microscope slides, we use image processing techniques to detect parasite-infected red blood cells. Analysing thin blood smears under a microscope and visually scanning for infected cells is the most commonly used technique today. The number of parasitic red blotches is manually counted by a clinician. - up to 5,000 cells in some cases (according to WHO protocol). Malaria could be more effectively prevented, monitored, and relieved if more accurate and efficient symptomatic strategies were available. We used image processing techniques to determine the proximity of malaria-infected cells. And to classify the stage of malaria, whether it's falciparum, which is the most common, or vivax, which is the least common.

3.2 OBJECTIVE

Malaria is a parasitic disease that affects humans. Detection from thin film blood smear images necessitates segmentation of single blood cells from microscopic blood slide images obtained from a pathologist, and the dataset which contain un segmented cell images. As a result, the proposed approach uses a number of image processing methods for segmentation. Techniques for edge detection and segmentation were used. This method solves the problem of cell overlapping by filtering out noise and identifying cell discontinuities. Using morphological segmentation, it distinguishes each cell and detects infection within the cell. Furthermore, since all of the images are raw and have varying intensities, detecting cells and infection is extremely difficult. To address this problem, the proposed approach employs histogram matching, in which all images are uniform in size and intensity, thus increasing accuracy.

3.3 EXISTING SYSTEM AND PROPOSED SYSTEM

EXISTING SYSTEM	PROPOSED SYSTEM
In existing system, there is no method to identify the malaria affected people. Drawbacks of existing system • Loss of life of people due to no awareness. • No proper treatment provides to the affected people.	In proposed system, there is a method to identify and classify the malaria affected people. Advantages of proposed system • Loss of life of people is prevented using image processing. • Technology helped to provide proper treatment to the affected people.

3.4 WORKING PRINCIPLE

We used an Arduino UNO microcontroller in this design, which serves as the system's brain and stores the entire system software. The blood cell image, which is a predefined dataset image, is used to identify and classify malaria patients using image processing. Using a heart rate, the malaria-affected patient's heart rate and body temperature are tracked at the same time temperature monitor, etc. If any abnormalities are discovered, the buzzer emits a warning sound and sends a message to the appropriate individual, as well as updating the IOT. The LCD displays all of the system's details.

CHAPTER 4

METHODOLOGY

4.1 METHODOLOGY BLOCK DIAGRAM:

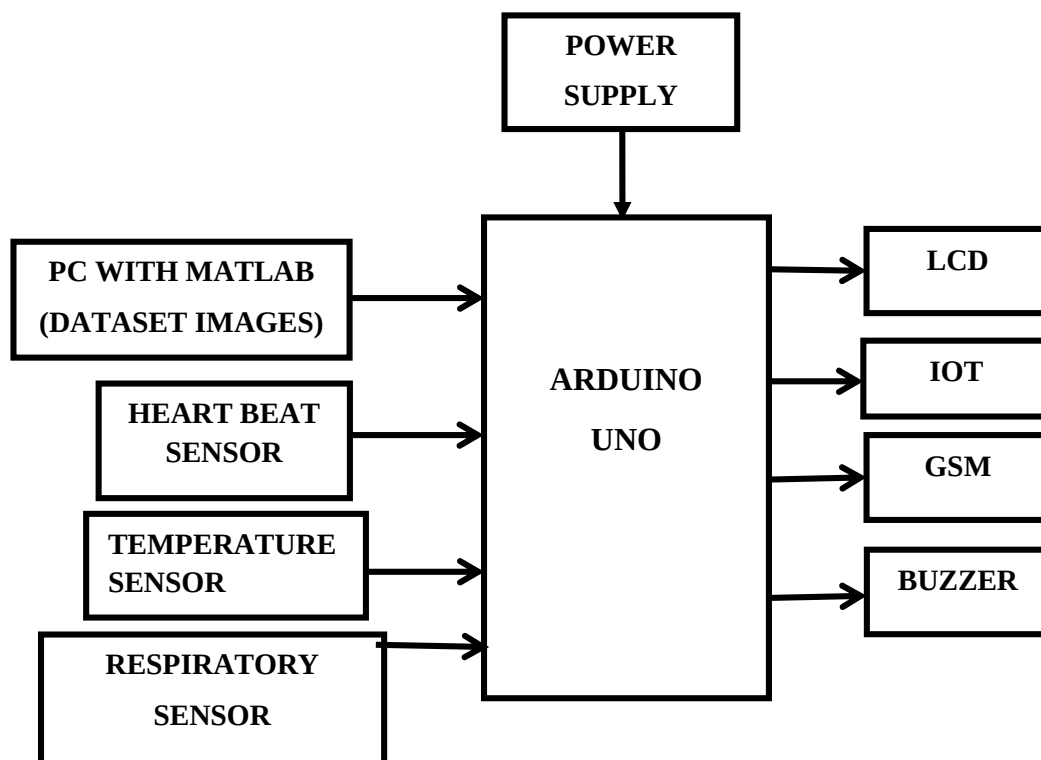


FIG 4.1 WHOLE PROCESS BLOCK DIAGRAM

4.2 HARDWARE REQUIREMENTS:

- Arduino UNO
- Heart Rate Sensor
- Temperature Sensor
- Respiratory Sensor
- LCD
- GSM
- Buzzer
- IOT

HARDWARE DESCRIPTIONS:

4.2.1 ARDUINO UNO

4.2.1.1 Introduction to Arduino

Arduino is an open-source electronics platform that uses simple hardware and software to make it easy to use. Arduino boards can read inputs - such as light from a sensor, a finger on a button, or a Twitter message - and convert them to outputs - such as turning on an LED, triggering a motor, or publishing something online. By sending a series of instructions to the board's microcontroller, you can tell it what to do.

Thousands of projects have used Arduino over the years, ranging from simple household items to sophisticated scientific instruments. This open-source platform has gathered a global group of creators - students, hobbyists, musicians, programmers, and professionals - whose efforts have added up to an enormous amount of usable information that can be of great benefit to novices and experts alike.

Arduino was created at the Ivrea Interaction Design Institute as a simple tool for rapid prototyping targeted at students with no previous experience with electronics or programming. As soon as it gained a larger following, the Arduino board began to evolve in order to meet new requirements and challenges, evolving from basic 8-bit boards to products for IoT, wearables, 3D printing, and embedding the climate All

Arduino boards are open-source, allowing users to create them on their own and customize them to meet their specific needs. The software is also open-source, and it is evolving thanks to contributions from users all over the world.



FIG 4.2 ARDUINO BOARD

The Arduino Uno is a microcontroller board that uses the ATmega328P microcontroller (datasheet). There are 14 digital input/output pins (six of which can be used as PWM outputs), six analogue inputs, a 16 MHz quartz crystal, a USB link, a power jack, an ICSP header, and a reset button on the board. It comes with everything you'll need to help the microcontroller; simply plug it into a device through USB or power it with an AC-to-DC adapter or a battery to get going. You can experiment with your UNO without fear of making a mistake; in the worst-case scenario, you can replace the chip for a few dollars and start over.

In Italian, the word "Uno" means "one," and it was chosen to commemorate the release of Arduino Software (IDE) 1.0. The Uno board and Arduino Software (IDE) version 1.0 were the reference versions of Arduino, which have since been superseded by newer updates. The Arduino Uno board is the first in a series of USB Arduino boards and the platform's reference model; see the Arduino website for a complete list of new, past, and obsolete boards the boarder Arduino index.

TABLE 4.1 TECHNICAL SPECIFICATIONS

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 Ma
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by boot loader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm
Weight	25 g

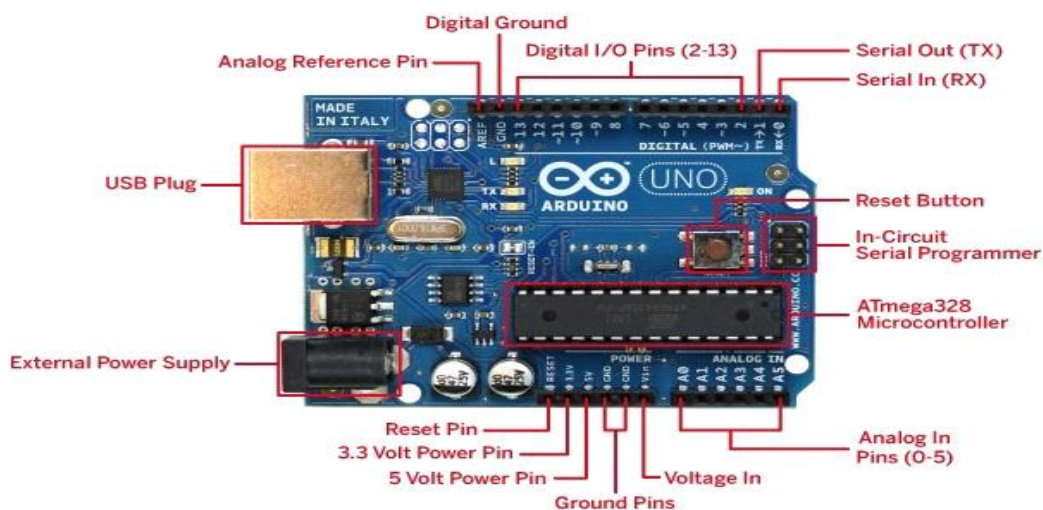


FIG 4.3 PIN CONFIGURATION OF ARDUINO BOARD

4.2.1.2 Programming

The (Arduino Software (IDE)) can be used to programme the Arduino Uno. From the Tools > Board menu, choose "Arduino/Genuino Uno" (according to the microcontroller on your board). See the guide and tutorials for more information.

The Arduino Uno's ATmega328 comes preprogrammed with a boot loader, allowing you to upload new code without the use of an external hardware programmer. It uses the original STK500 protocol to communicate (reference, C header files).

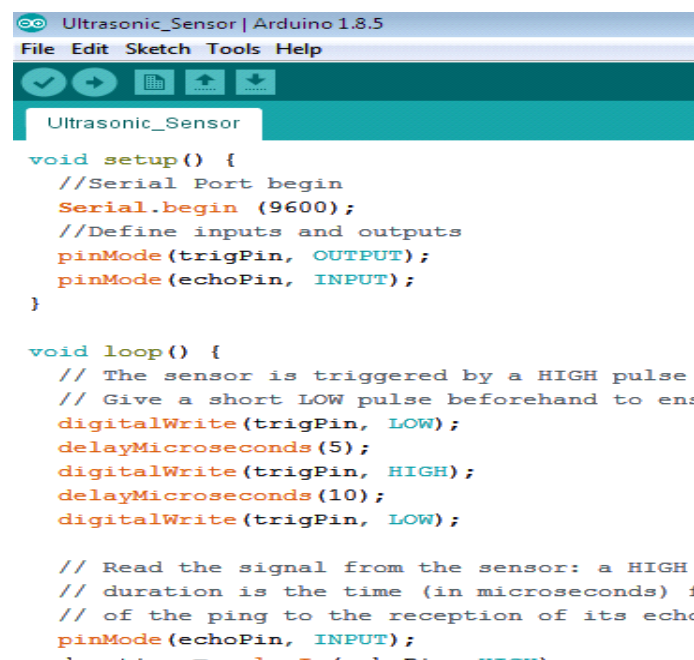


FIG 4.4 ARDUINO PROGRAMMING

You may also use Arduino ISP or similar to bypass the boot loader and programme the microcontroller directly via the ICSP (In-Circuit Serial Programming) header; see these instructions for more details.

The firmware source code for the ATmega16U2 (or 8U2 in the rev1 and rev2 boards) is available in the Arduino repository. The ATmega16U2/8U2 comes with a DFU boot loader that can be triggered using the following commands:

- Attach the solder jumper on the back of the board (near the map of Italy) and then rese the 8U2 on Rev1 boards.
- A resistor pulls the 8U2/16U2 HWB line to ground on Rev2 or later boards, making it easier to bring into DFU mode.

After that, you can load a new firmware using Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux). You may also use an external programmer with the ISP header (overwriting the DFU boot loader). For more detail, see this user-contributed tutorial.

4.2.1.3 Applications

- Arduboy is an Arduino-based handheld game console.
- The Arduinome is a MIDI controller that looks like the Monome.
- Drone software and hardware from Ardupilot.
- ArduSat is an Arduino-based cubesat.
- C-STEM Studio is a robotics-based forum for hands-on learning in computing, science, technology, engineering, and mathematics (C-STEM).
- Data loggers are used in scientific research. OBDuino is a trip machine that works with most modern cars' on-board diagnostics interface.
- OpenEVSE is a free and open-source electric vehicle charging station.
- XOD is an Arduino optical programming language.

4.2.2 HEART BEAT SENSOR



FIG 4.5 HEARTBEAT SENSOR KY-039

This project detects the pulse of the finger using a phototransistor and a bright infrared (IR) LED, for a red LED flashing with each pulse. The LED is on the light side of the finger, and the phototransistor is on the other side of the finger; the phototransistor is used to obtain the flux released when the blood pressure pulses by the finger, and the resistance of the photo transistor is low.

Will be altered slightly We selected a very high resistance resistor R1 for the project's schematic circuit because much of the light passing through the finger is absorbed, so the phototransistor must be sensitive enough. To get the best results, resistance can be chosen by experimentation. The most important thing is to prevent stray light from entering the phototransistor. For lighting in the house, Since the lights at home are often dependent on 50HZ or 60HZ fluctuation, a slight heartbeat will add a significant amount of noise.

The calculated values are printed when the programme is run. It could be difficult to get a real pulse from this. Since much of the light that passes through the finger is absorbed, the phototransistor in the KY-039 Heartbeat Detector has a very high resistance resistor R1. The most important thing is to prevent stray light from entering the phototransistor. Lighting for the house This is especially important because the lights in most homes fluctuate between 50 and 60 Hz, so a slight heartbeat can create a lot of noise.

4.2.2.1 Specifications

- Working voltage: 5V
- detect pulsation in fingers using an IR LED and an optical transistor; used for a synthesized teaching experiment.
- Weight: 2g
- Dimensions: 25 x 22 x 17mm

4.2.2.2 Features

This project detects the pulse of the finger using a bright infrared (IR) LED and a phototransistor, for a red LED flashing with each pulse. The following is how a pulse sensor works:

- The LED is on the light side of the finger, and the phototransistor is on the dark side. The phototransistor is used to obtain the flux released as the blood pressure pulses through the finger and the phototransistor resistance changes slightly.
- Since much of the light passing through the finger is absorbed, it is preferable to use a phototransistor that is sensitive enough. To get the best results, resistance can be chosen by experimentation. The most important thing is to prevent stray light from entering the phototransistor. This is especially important for home lighting since most lights in homes operate at 50 or 60 Hz. fluctuate, because even a slight heartbeat can add a lot of noise.

Connecting to Arduino

- The Arduino pin Analog 0 / A0 is connected to the sensor pin S.
- Link sensor pin + (middle pin) to Arduino pin 5+.
- Link the sensor pin to GND on the Arduino.

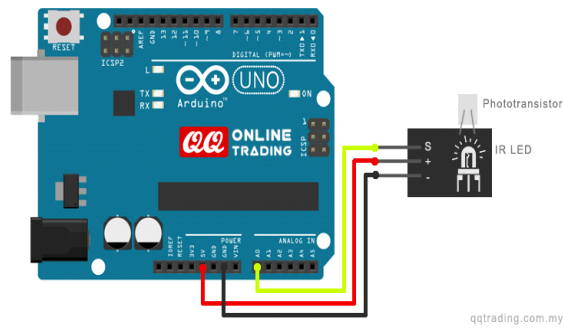


FIG 4.6 CONNECTION OF KY-039

4.2.3 TEMPERATURE SENSOR

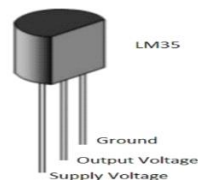


FIG 4.7 TEMPERATURE SENSOR (LM35)

The LM35 series are precision integrated-circuit temperature devices with a linearly proportional output voltage to the temperature in degrees Celsius. In comparison to linear temperature sensors tuned in Kelvin, the LM35 unit has the advantage of not requiring the consumer to subtract a significant constant voltage from the output to obtain convenient Centigrade scaling. The LM35 computer is self-contained and does not need any external calibration or trimming to achieve average accuracies of 14°C at room temperature and 34°C over a maximum temperature range of 55°C to 150°C. Trimming and calibration at the water level ensure lower costs. The LM35 device's low

output impedance, linear output, and precise inherent calibration render interfacing to readout or control circuitry a breeze. With a single power supply, the unit is used, or a combination of the two. The LM35 unit has a very low self-heating of less than 0.1°C in still air since it only draws 60 A from the supply.

The LM35 device is rated for a temperature range of 55°C to 150°C , whereas the LM35C device is rated for a temperature range of 40°C to 110°C (with increased accuracy of 10°). The LM35-series devices are available in hermetically sealed TO transistors. The LM35C, LM35CA, and LM35D devices are available in plastic TO-92 transistor packages, while the LM35C, LM35CA, and LM35D devices are available in metal TO-92 transistor packages. The LM35D is available in a plastic TO-220 package and an 8-lead surface-mount small-outline package.

4.2.3.1 LM35 temperature sensor circuit configuration

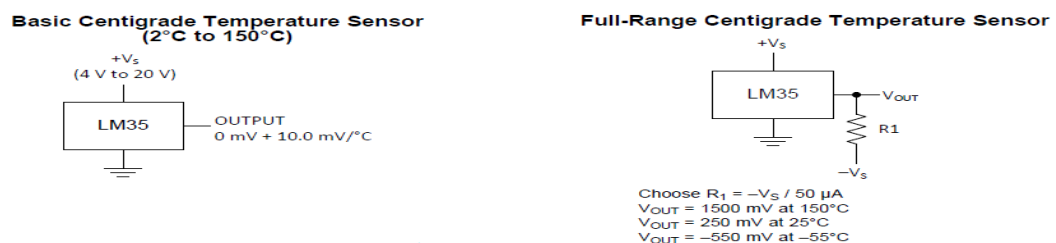


FIG 4.8 CIRCUIT CONFIGURATION

The Lm35 can be used in two different circuits. Both produce different outcomes. Only positive temperatures between 2 and 150 degrees Celsius can be measured in the first configuration. We simply power the lm35 and link the output directly to analogue to digital converters in this first configuration. We can use all of the sensor tools in the second configuration and measure the entire region. Temperatures range from -55 degrees to 150 degrees centigrade. This setup is a little more complicated, but it produces excellent results. In this situation, we'll need to use an external resistor to raise the level of negative voltage. The value of the external resistor can be determined

using the formula given below the configuration circuit. The second configuration circuit can be built in a number of different ways. Visit the Texas Instruments lm35 datasheet to learn more about the second configuration circuits. The circuit is listed on the Texas Instruments data sheet with simple component values. Despite the fact that the first configuration did not need a resistor on the output side, I suggest connecting an 80 k to 100 k resistor between V_{out} and GND pin. When I did some tests, I found that the readings fluctuated from time to time. The V_{out} pin floats as the voltages fluctuate. As a result, a resistor between V_{out} and GND keeps the V_{out} pin low and stops it from floating.

For both configurations, the accuracy parameters are different. For both configurations, the average accuracy level is ± 1 degree Celsius. When the temperature is between 2 and 25 degrees Celsius, however, the accuracy level drops. Now that we've covered the pin out, structure, linear scale factor, and accuracy level of the lm35 temperature sensor, it's time to go through the steps for measuring it. The temperature is measured with a Lm35 temperature sensor.

Formula to convert voltage to temperature

Centigrade Temperature = Voltage is the formula for converting voltage to centigrade temperature for Lm35. I divided by ten millivolts since the linear scale factor for the Lm35 is ten millivolts.

4.2.3.2 LM35 Working Principle

To comprehend the working theory of the lm35 temperature sensor, we must first comprehend the linear scale factor. It is mentioned in the features of the lm35 that it is +10 millivolts per degree centigrade. It means that for every 10 millivolts increase in sensor V_{out} pin output, the temperature value increases by one. For instance, if the sensor outputs 100 millivolts at the V_{out} pin, the temperature would be 10 degrees Celsius. The negative temperature reading is the same. The temperature would be -10 degrees Celsius if the sensor outputs -100 millivolts.

4.2.3.3 Features

- Directly calibrated in Celsius (Centigrade)
- + 10-mV/°C Linear 0.5°C Scale Factor Accuracy Guaranteed (at 25°C)

- Temperature Range: 55°C to 150°C
- Suitable for Use in Remote Locations
- Low-Cost Wafer-Level Trimming Operates from 4 to 30 V Less than 60- μ A Current Drain Low Self-Heating, 0.08°C in Still Air Non-Linearity Just 14°C This is typical.
- for 1-mA Low-Impedance Output
- Since it's tiny, it's ideal for remote applications.
- TO-92, TO-220, TO-CAN, and SOIC packages are available.
- The output voltage is directly proportional (Linear) to temperature (i.e., for every 1°C increase in temperature, the output voltage rises by 10mV (0.01V).

4.2.3.4 Functional block diagram

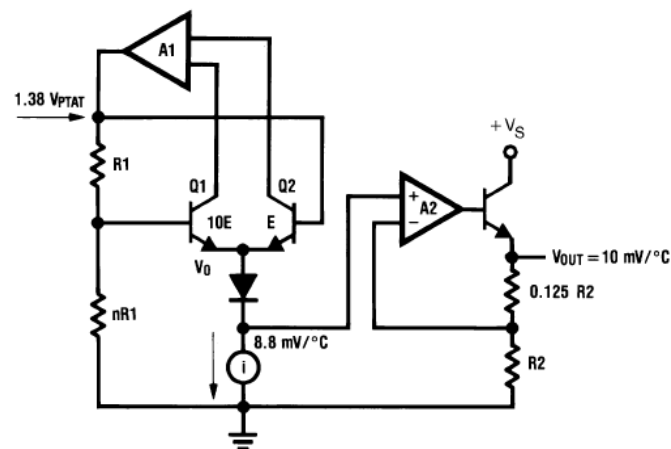


FIG 4.9 BLOCK DIAGRAM

An amplifier buffers the temperature-sensing portion before sending it to the VOUT pin. As seen in the Functional Block Diagram, the amplifier has a simple class A output stage with a standard 0.5-output impedance. As a result, the LM35 can only source current and has a sinking capability of 1A.

4.2.3.5 LM35 Temperature Sensor Applications

Providing thermal shut down for a circuit/component, monitoring battery temperature, and measuring temperatures for HVAC applications are all things that can be done with a thermometer.

4.2.4 RESPIRATORY SENSOR

The Respiration Sensor is used in biofeedback applications such as stress management and relaxation training to track breathing. The sensor should be tightened enough to avoid tension loss. This sensor is mounted near the nose and resembles a face mask. The sound sensor module is commonly used to detect sound intensity and provides a convenient way to detect sound. This module is suitable for use. for applications such as protection, switching, and monitoring Its precision can be easily tailored to fit your needs. It makes use of a microphone as the source of data for an amplifier, peak detector, and buffer.

When a sound is detected, the sensor generates an output signal voltage, which is sent to a microcontroller, which then performs the necessary processing. The sound sensor module makes it simple to detect sounds. It is used to detect sound and is most often used to detect sound strength. This module can be used for applications such as security, switching, and monitoring. Its precision can be easily tailored to fit your needs. It makes use of a microphone as the source of data for an amplifier, peak detector, and buffer. When a sound is detected, the sensor generates an output signal voltage that is sent to a microcontroller. After that, it performs the required processing.

4.2.4.1 Specifications

1. Voltage of operation 3.3V to 5V
2. Model of output: digital switch outputs (0 and 1, high or low level)
3. With a hole for a mounting screw

4. PCB Dimensions: 3.4 * 1.6 cm

4.2.4.2 How to test

The following components will be used:

- Microcontroller (any Arduino compatible)
- Sound sensor module
- 1 Pin M-M connectors
- Breadboard
- USB cable

1. Using an M-M pin connector, link the components according to the wiring diagram's figure. The VCC pin is attached to the 3.3V or 5V power supply, the GND pin to the GND, the DO pin to a digital I/O pin, and the AO pin to an analogue pin. The pin number will be determined by the software code.
2. After the hardware is connected, open the Arduino IDE and load the sample sketch.
3. Attach the microcontroller's ports to the device with a USB cable.
4. Put the software on the computer.
5. Use the serial monitor to display the data.

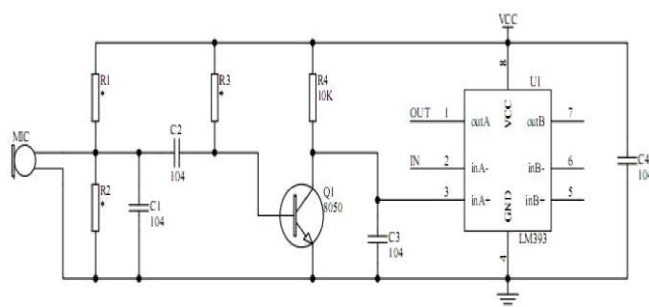


FIG 4.10 SCHEMATIC CONNECTION DIAGRAM



FIG 4.11 RESPIRATORY SENSOR

4.2.5 LIQUID CRYSTAL DISPLAY (LCD)

An LCD screen is an electronic display module that can be used in a variety of ways. A 16x2 LCD monitor is a very simple module that can be used in a variety of devices and circuits. Seven-segment and other multi-segment LEDs are favored over these modules. The reasons for this are as follows: LCDs are inexpensive, easy to programme, and have no restrictions on displaying unique or even custom characters (unlike some other displays). Animations, for example, are used in seven segments.

A 16x2 LCD can display 16 characters per line on each of its two sides. Each character is represented in a 5x7 pixel matrix on this LCD. Command and Data are the two registers on this LCD. The LCD command instructions are stored in the command register. A command is a collection of instructions provided to an LCD to perform a specific task, such as initializing it, clearing its screen, or positioning the cursor. controlling the show, and so on. The data to be displayed on the LCD is stored in the data register. The ASCII value of the character to be displayed on the LCD is the data.

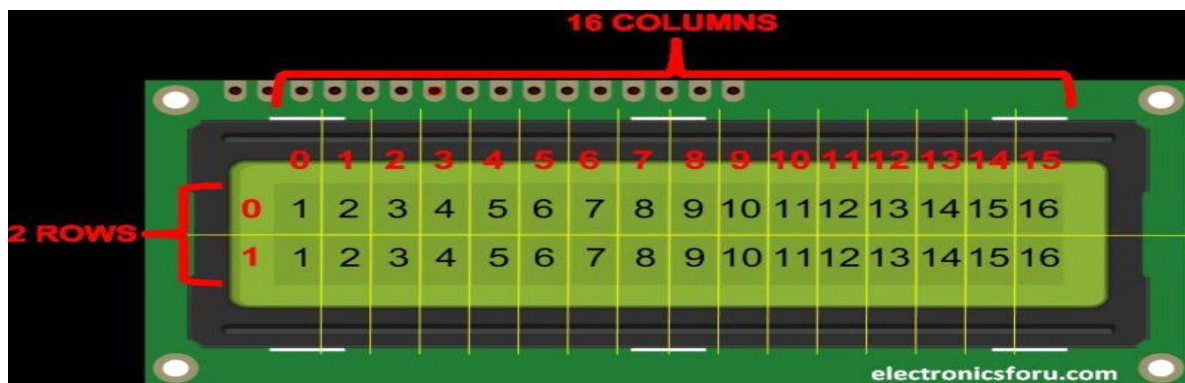


FIG 4.12 LCD

LCD displays can be seen all over the place. The time is shown on computers, calculators, television sets, cell phones, and digital watches using some kind of display. A liquid crystal display (LCD) is an electronic display module that produces a visible image using liquid crystal. The 162 LCD monitor is a very simple module that

is often used in projects. The 162 refers to a display of 16 characters per line in two lines. Each character is represented in a 5x7 pixel matrix on this LCD.

4.2.5.1 16X2 LCD Pinout diagram

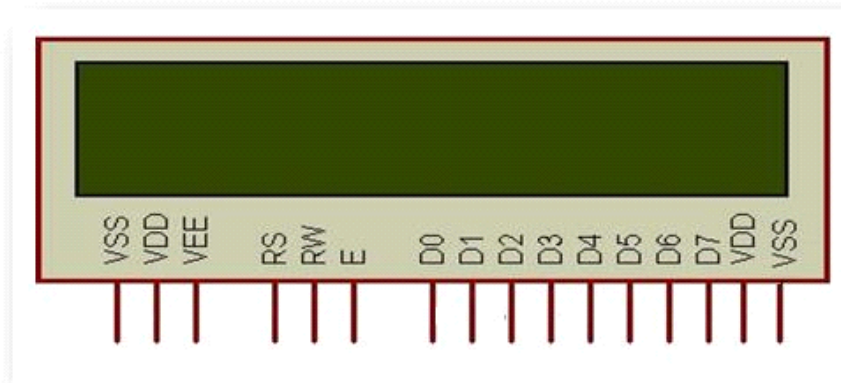


FIG 4.13 LCD PINOUT DIAGRAM

A 16X2 LCD has two registers, command and data.

REGISTER SELECT (RS)

Switching from one register to another is done with the register pick. The command register has the value RS=0, while the data register has the value RS=1.

COMMAND REGISTER

The command register keeps track of the commands sent to the LCD. A command tells the LCD to perform a specific task, such as initializing it, clearing its screen, setting the cursor location, controlling the monitor, and so on. The command register is where commands are processed.

DATA REGISTER

Stores the data that will be displayed on the LCD. The ASCII value of the character to be displayed on the LCD is the data. When we send data to the LCD, it is sent to the data register, where it is processed. The data register selected when RS=1.

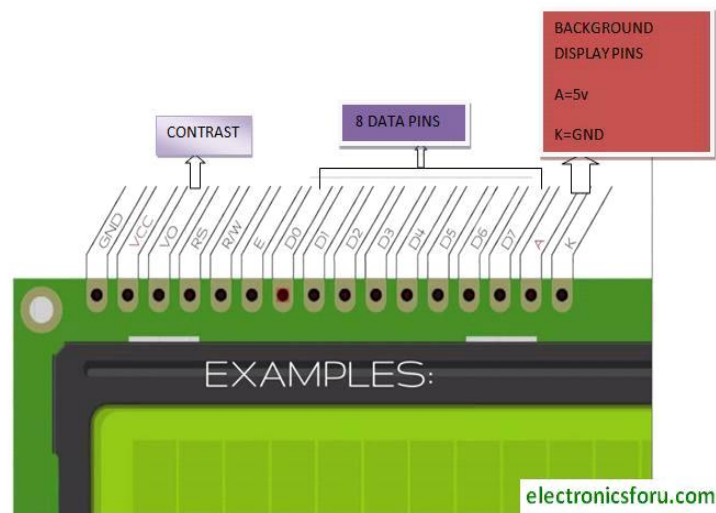


FIG 4.14 LCD PIN CONFIGURATION

GENERATING CUSTOM CHARACTERS FOR A 16X2 LCD

Creating custom characters for a 16X2 LCD is not difficult. It necessitates an understanding of LCD custom created random access memory (CG-RAM) and the LCD chip controller. Hitachi HD4478 controller is used in most LCDs. The key component in creating custom characters is CG-RAM. Once declared in the code, it saves the custom characters. The size of the CG-RAM is 64 bytes, allowing for the development of eight characters at once the size of each character is eight bytes.

CG-RAM addresses begin at 0x40 in hexadecimal or 64 in decimal. At these addresses, we can create custom characters. We can now print our characters on the LCD at any time by sending simple commands to the LCD after we create them at these addresses. Below are the character addresses and printing commands.

Each character's starting address, as well as their printing commands, are mentioned in the table above. The first character is created at 0x40 to 0x47 and printed on the LCD by simply sending command 0 to the LCD. The second character is formed at addresses 0x48 to 0x55 and printed on the LCD by sending 1 to it.

4.2.6 GLOBAL SYSTEM FOR MOBILE COMMUNICATION (GSM)

Definition

GSM (Global System for Mobile Communication) is an internationally recognized digital cellular communication standard. GSM stands for Global System for Mobile Communications, and it was established in 1982 with the aim of developing a popular European mobile telephone standard that would define requirements for a Pan-European mobile cellular radio system operating at 900 MHz. Many countries outside of Europe are expected to have a population of over a billion people. will become a member of the GSM alliance.



FIG 4.15 GSM

4.2.6.1 SIM 900 GSM/GPRS Module

The GSM/GPRS Modem-RS232 is equipped with a Dual Band GSM/GPRS engine (SIM900) that operates at frequencies of 900 and 1800 MHz the modem has an RS232 interface, which allows you to connect it to a PC or a microcontroller that has an RS232 chip (MAX232). The baud rate can be changed from 9600 to 115200 using the AT button. The internal TCP/IP stack on the GSM/GPRS Modem allows you to connect to the internet through GPRS. In M2M gui, it is suitable for SMS, Voice, and DATA transfer applications. You can link a wide range of unregulated power supplies to the onboard Controlled Power Supply. You can use this modem to make audio calls, send and receive SMS, answer incoming calls, and access the internet using easy AT commands.

4.2.6.2 Features

- RS232 interface for direct communication with device or MCU package.
- Dual-Band GSM/GPRS 900/ 1800 MHz
- The baud rate can be modified.
- The 29302WU IC is used to regulate the power.
- Compliance with ESD.
- MIC and Speaker socket should be allowed. With slid in SIM card tray.
- With Stub antenna and SMA socket, and 12V DC input voltage.

4.2.6.3 Working

A GSM modem, unlike a cell phone, does not have a keypad or monitor in which to communicate. It simply accepts and acknowledges those commands through a serial interface. AT commands are the abbreviation for these commands. There is a list of AT commands that can be used to tell the modem what to do. Any order begins with the letters "AT." This is why they're referred to as AT orders. AT is an acronym for "attention." The software in our simple project waits for the mobile number to be entered through the keyboard. The software instructs the modem to send the text message using a sequence of AT commands when a ten-digit mobile number is given.

4.2.6.4 AT Commands

- Set the SIM900 to text mode: AT+CMGF=1\r.
- Send SMS to a number: AT+CMGS=PHONE_NUMBER (in international format).
- Read the first SMS from the inbox: AT+CMGR=1\r.
- Read the second SMS from the inbox: AT+CMGR=2\r.
- Read all SMS from the inbox: AT+CMGR=ALL\r.
- Call to a number: ATDP+PHONE_NUMBER (in international format).
- Hang up a call: ATH.
- Receive an incoming call: ATA

4.2.7 BUZZER

A buzzer or beeper is a mechanical, electromechanical, or piezoelectric audio signalling system (piezo for short). Alarm clocks, timers, and confirmation of user input such as a mouse click or keystroke are all popular uses for buzzers and beepers.



FIG.4.16 BUZZER

4.2.7.1 Introduction to Piezo buzzer

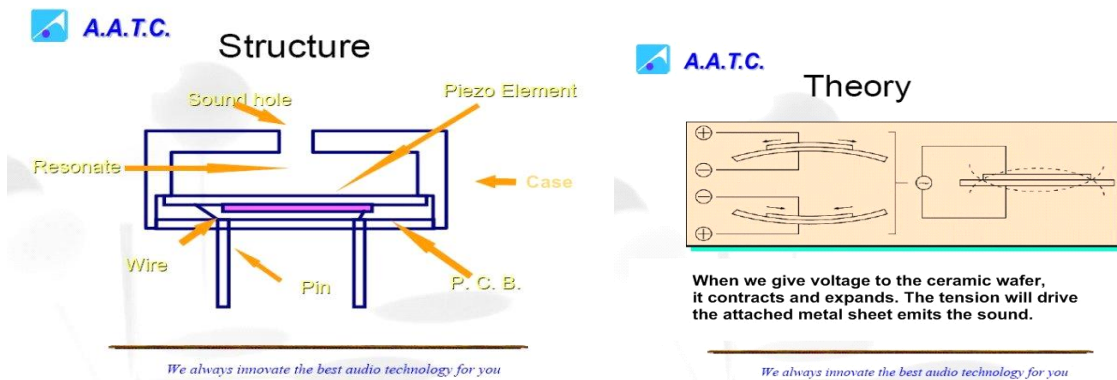


FIG 4.17 STRUCTURE OF BUZZER

4.2.7.2 Specifications

Rated Voltage: Square waves are used to power a piezo buzzer (V p-p).

Operating Voltage: To allow for regular operation. However, making the minimum SPL under the rated voltage is not guaranteed.

Consumption Current: Under normal service, the current is absorbed steadily. However, it usually takes three times the current amount when you first start working.

Capacitance: With a higher capacitance, a piezo buzzer can produce higher SPL, but it also consumes more electricity. The decibel meter is used to calculate the sound quality. Using rated voltage and square waves over a distance of ten centimetres.

Rated Frequency: A buzzer can produce sound at any frequency, but we believe the graded frequency produces the highest and most stable SPL.

Operating Temp: Maintain good working conditions between -30°C and $+70^{\circ}\text{C}$.

Driving methods: When we can't provide a buzzer with a frequency signal and it just requires direct current, the AZ-xxxxS-x series with built-in drive circuit is the best option. You may also choose from a variety of tone types, including continuous, quick pulse, and slow pulse.

Dimension: Frequency is affected by size; small sizes result in high frequency.

Voltage: The higher the voltage, which is powered by square waves (V p-p), the higher the SPL.

Pin Pitch: Due to the large number of piezo buzzer specifications, finding a facsimile is difficult; therefore, we recommend that you start by selecting a specification with a similar pitch and frequency. Micro Speakers are being introduced.

4.2.8 INTERNET OF THINGS (IoT)

The internet of things (IoT) is a network of physical devices, cars, houses, and other items that are equipped with electronics, software, sensors, actuators, and network communication to capture and share data. The Internet of Things (IoT) was described as "the infrastructure of the information society" by the Global Standards Initiative on Internet of Things (IoT-GSI) in 2013. The Internet of Things (IoT) enables objects to communicate with each other to be sensed and operated remotely over existing network infrastructure, allowing for more direct incorporation of the physical environment into computer-based systems, improving performance, precision, and economic gain.

As IoT is enhanced with sensors and actuators, it becomes a subset of the larger category of cyber-physical devices, which also includes cyber-security. Smart grids, smart houses, intelligent transportation, and smart cities are all examples of smart technology. Each thing is uniquely recognizable thanks to its embedded computing system, but it can also communicate with other things on the Internet. Experts predict that by 2020, the Internet of Things would have nearly 50 billion items.

4.2.8.1 Application areas

Cities

Smart Parking: The number of parking spaces in the city is monitored.

Vibrations and material conditions in buildings, bridges, and historical landmarks are monitored for structural health.

Noise Urban Maps: Real-time sound tracking in bars and centric zones.

Traffic Congestion: Car and pedestrian levels are monitored in order to improve driving and walking paths.

Intelligent and weather-adaptive lighting in street lights is known as smart lighting.

Waste Management: Detection of garbage levels in containers in order to maximize performance.

Routes for garbage collection

Smart Roads and Intelligent Highways are two types of intelligent transportation systems.

with weather-related warnings and detours, as well as unusual incidents such as collisions or traffic jams.

Security & Emergencies

Perimeter Access Control: Controlling access to restricted areas and detecting people in areas where they are not allowed.

Liquid Presence: Identification of liquid in data centres, warehouses, and sensitive building grounds to avoid breakdowns and corrosion.

Radiation Levels: Calculation of radiation levels in the vicinity of nuclear power plants to produce leakage warnings. *Perimeter Access Control:* Controlling access to restricted areas and detecting people in areas where they are not allowed.

Liquid Presence: Identification of liquid in data centres, warehouses, and sensitive building grounds to avoid breakdowns and corrosion.

Radiation Levels: Calculation of radiation levels in the vicinity of nuclear power plants to produce leakage warnings.

Explosive and Hazardous Gases: Detection of gas levels and leaks in industrial settings, chemical plant surroundings, and my interiors.

eHealth

Fall Detection: Assisting elderly or disabled people who are self-sufficient.

Medical Fridges: Temperature control in freezers that store vaccines, medications, and organic materials.

Vital Signs Tracking in High Performance Centres and Fields

Patients Surveillance: Observation of patients' conditions in hospitals and nursing homes.

UV Radiation: UV sun rays are measured in order to warn people not to be exposed at those hours

4.3 ESP-12E BASED NODE MCU

The Espressif ESP8266 is a microcontroller created by Espressif Systems. The ESP8266 is a self-contained Wi-Fi networking system that can function as a bridge between existing microcontrollers and Wi-Fi, as well as run standalone applications. This module has a built-in USB connector as well as a large number of pin-outs. You can link the Node MCU devkit with a micro-USB cable to your laptop and flash it in the same way that Arduino does. It's also breadboard-friendly right away.

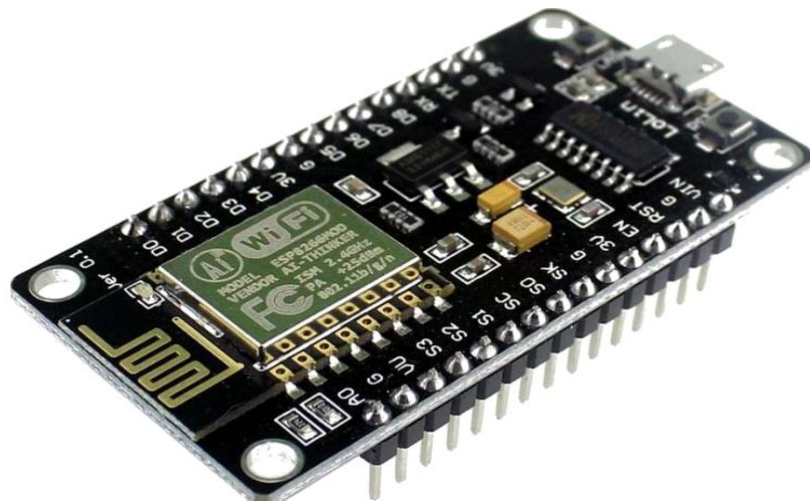


FIG 4.18 ESP-12E BASED NODE MCU

Ai-thinker Team created the ESP-12E Wi-Fi module. Tensilica L106 combines industry-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/LNA, on-board antenna in smaller sizes of the module.

The module fully supports IEEE802.11 b/g/n Protocol stack for TCP/IP. Users can connect modules to an existing device's networking or create their own network controller. ESP8266 is a high-integration wireless SOC designed for mobile platform designers with limited space and power. It offers unrivalled ability to integrate Wi-Fi capabilities into other systems or to operate as a standalone programme at the lowest cost and with the least amount of effort. a need for space

The ESP8266EX is a complete and self-contained Wi-Fi networking solution that can either host or offload Wi-Fi networking functions from another application processor. As the ESP8266EX runs the programme, it does so from an external flash drive. In order to increase the system's performance in such applications, it has an integrated cache. Alternatively, it can be used as a Wi-Fi adapter.

With easy connectivity (SPI/SDIO or I2C/UART interface), wireless internet access can be added to any microcontroller-based design. The ESP8266EX is one of the industry's most integrated Wi-Fi chips; it includes antenna switches, RF baluns, power amplifiers, low noise receive amplifiers, filters, and power management modules, and it requires minimal external circuitry. The front-end module is planned to take up the least amount of space on the PCB.

4.3.1 ESP-12E Architecture

Aside from Wi-Fi, the ESP8266EX includes an upgraded version of Tensilica's L106 Diamond series 32-bit processor with on-chip SRAM. Via its GPIOs, the ESP8266EX is often connected to external sensors and other application-specific devices; codes for such applications are included in the SDK's examples. The Smart Connectivity Platform (ESCP) from Espressif Systems demonstrates sophisticated connectivity. Quick sleep/wake context switching for energy-efficient VoIP and adaptive radio biasing are system-level features. Low-power service, advanced signal processing, spur cancellation, and radio coexistence functionality for cellular, Bluetooth, DDR, LVDS, and LCD interference mitigation.

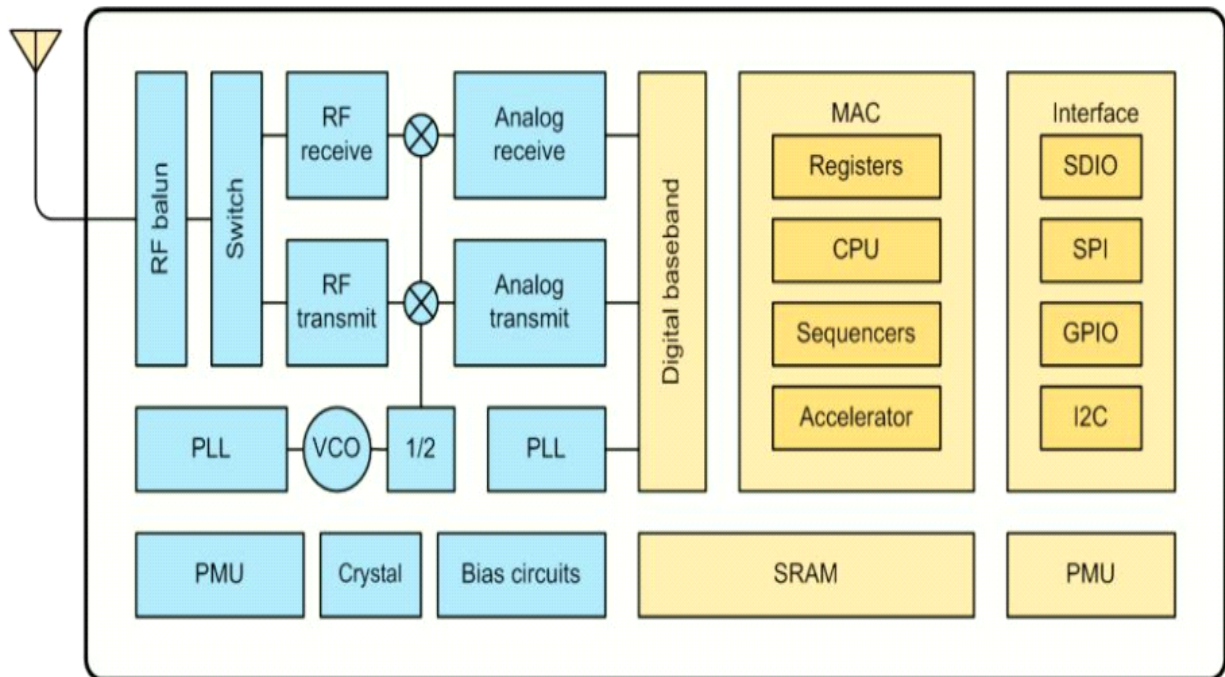


FIG 4.19 ESP-12E ARCHITECTURE

4.3.2 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
- All Android and iOS devices can use the Smart Link feature.
- SDIO 2.0, SPI, UART, I2C, I2S, IRDA, PWM, and GPIO (H)
- 1x1 MIMO, 2x1 MIMO STBC
- Aggregation of A-MPDU and A-MSDU data, with a 0.4s guard interval
- 5uA deep sleep power

- 2ms wake up and relay packets
- 1.0mW standby power consumption (DTIM3)
- In 802.11b mode, the output power is +20dBm; the operating temperature range is -40C to 125C.

4.4 HARDWARE SETUP OF OUR PROJECT:

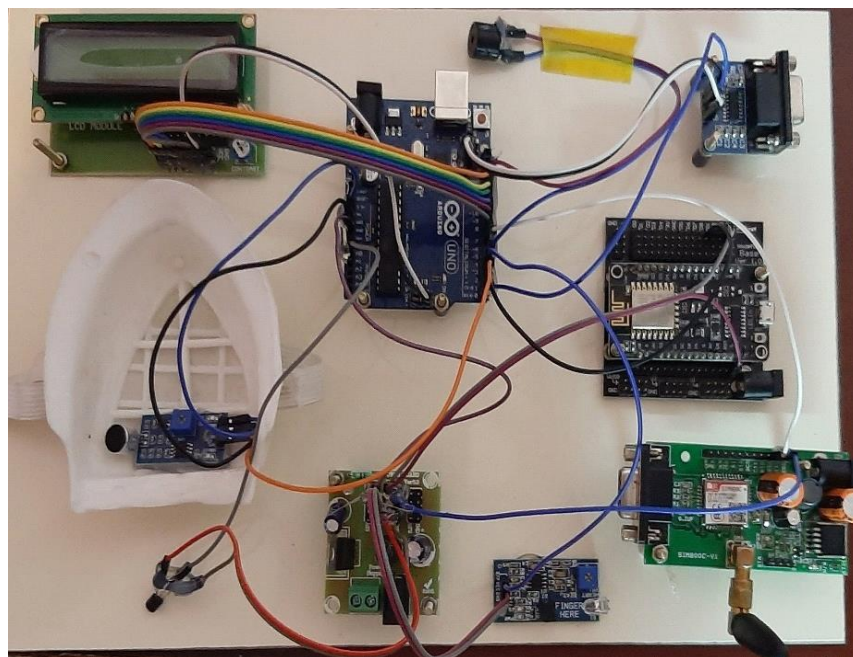


FIG 4.20 HARDWARE SETUP

4.5 SOFTWARE REQUIREMENTS:

- MATLAB
- EMBEDDED C
- ARDUINO IDE

SOFTWARE DESCRIPTIONS

4.5.1 MATLAB

MATLAB (MATrix LABoratory) is a fourth-generation programming language and numerical computing environment. Matrix manipulations, plotting of functions and data, implementation of algorithms, design of user interfaces, and interfacing with programs written in other languages, such as C, C++, Java, and Fortran, are all possible with MATLAB, which was developed by Math Works.

MATLAB is mainly designed for numerical computation, but an optional toolbox uses the MuPADsymbolic engine to provide symbolic computing capabilities. Simulink, a separate kit for dynamic and embedded systems, adds graphical multi-domain simulation and Model-Based Design.

Around one million people used MATLAB in industry and academia in 2004. Users of MATLAB come from a variety of engineering, science, and economics backgrounds. In both academic and industrial settings, MATLAB is commonly used.

Little's specialty, control engineering, was the first to introduce MATLAB, but it soon spread to a variety of other fields. It is now widely used in education, especially in the teaching of linear algebra and numerical analysis, and among image processing scientists. The MATLAB framework is based on the MATLAB programming language. The most straightforward method of executing. To enter MATLAB code, use the Command Window, which is one of the MATLAB Desktop's features. MATLAB can be used as an interactive mathematical shell when code is inserted in the Command Window. Sequences of commands may be saved as a script or encapsulated into a feature in a text file, usually using the MATLAB Editor, expanding the commands accessible.

A variety of features in MATLAB allow you to log and share your work. Your MATLAB code can be integrated with other languages and applications, and your MATLAB algorithms and applications can be distributed.

4.5.1.2 FEATURES OF MATLAB

- Interactive methods for iterative experimentation, design, and problem solving. High-level language for technical computing. Development environment for managing code, files, and data.
- Linear algebra, statistics, Fourier analysis, filtering, optimization, and numerical integration are among the mathematical functions available.
- Graphics functions in 2D and 3D for visualizing
- Functions for integrating MATLAB-based algorithms with external programmes and languages such as C, C++, Fortran, Java™, COM, and Microsoft Excel. Tools for creating custom graphical user interfaces.

Signal and image processing, communications, control design, test and calculation, financial modelling and analysis, and computational all use MATLAB. Add-on toolboxes (groups of MATLAB functions with a specific purpose) extend the MATLAB environment to solve specific types of problems in these application areas.

MATLAB is a programming language that can be used on both personal computers and efficient server systems, such as the Cheaha compute cluster. The Parallel Computing Toolbox allows the language to be expanded with parallel implementations for common tasks.

Unrolling for-loops is one of the computational features. Offloading computationally intensive workloads to Cheaha, the campus compute cluster, is also supported by this toolbox. MATLAB is one of only a few programming languages in which each vector is a matrix (in the broadest sense) and "knows" how large it is. Furthermore, when r is large, the fundamental operators (such as addition and multiplication) are programmed to deal with matrices. And the MATLAB environment takes care of a lot of the tedious housekeeping that allows all of this to happen. Since matrices are used in so many of the Macro-Investment Analysis procedures, MATLAB proves to be a very efficient language for both communication and execution.

4.5.2 EMBEDDED C

Embedded C is the most widely used programming language in the software industry for creating electronic devices. Embedded software is connected with each processor used in an electronic device.

Embedded C programming is important for the processor to perform specific tasks. We use a variety of electronic devices in our daily lives, such as a cell phone, a washing machine, a digital camera, and so on. Both of these gadgets are ineffective. The system runs on a microcontroller that is programmed in embedded C. Let's look at how embedded machine programming is represented in a block diagram.

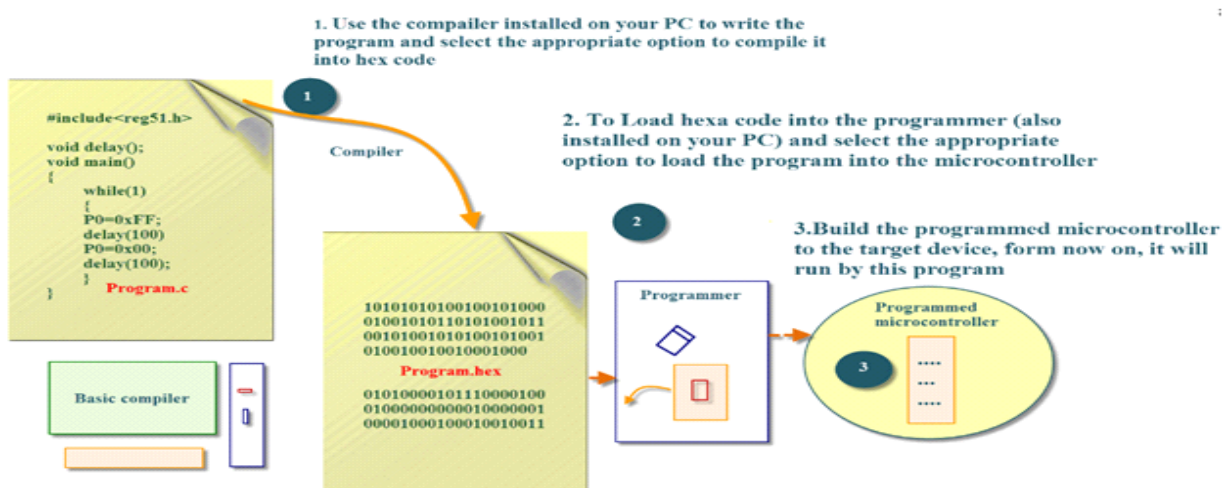


FIG 4.21 BLOCK DIAGRAM OF EMBEDDED C

The Embedded C code in the block diagram above is used to blink the LED connected to the microcontroller's Port0.

C code is favoured over other languages in embedded machine programming. Because of the following factors: ease of use, high reliability, portability, and scalability

4.5.2.1 Embedded system programming

Basic Declaration

Let's look at the progress of Embedded C Programming in a block diagram.

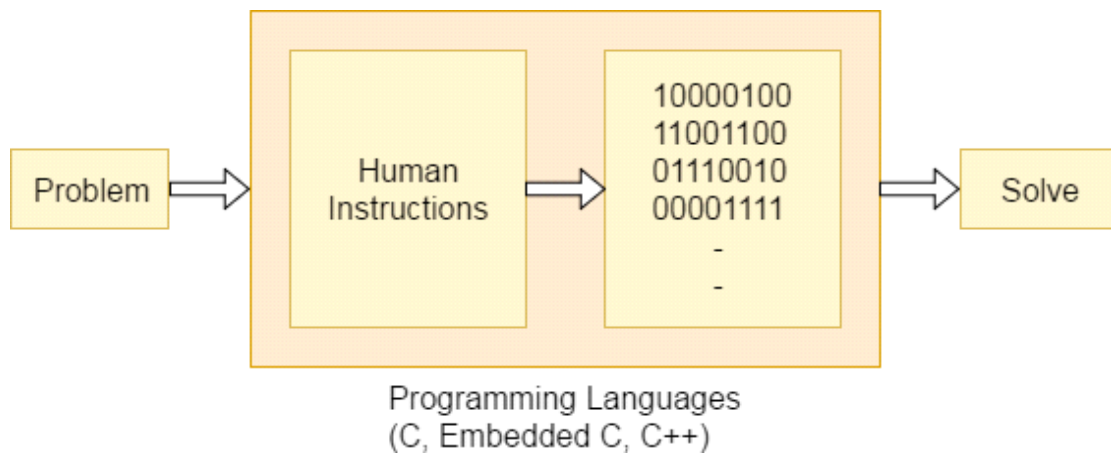


FIG 4.22 PROGRESS IN BLOCK DIAGRAM OF EMBEDDED C

A programming language is a set of one or more functions that are used to perform a specific task. A function is a collection of statements that are used to perform a specific task. Basic elements and grammatical rules exist in every language. Variables, character set, data types, keywords, expressions, and other features of the C programming language are used to write C programmes. The embedded C programming language is a C language extension. In addition to the features mentioned above, embedded programming in C includes data types, keywords, and a header file, which is defined by `#includemicrocontroller name.h`.

4.5.2.2 Embedded system

An Embedded System is a computer-hardware system that has software embedded in it, allowing it to be used for a variety of applications or a particular part of an application or product, or as part of a larger system.

An embedded system may be a small stand-alone system or a large multi-system setup. It is a microcontroller-based control device that is used to carry out a particular task.

An embedded device is made up of three main elements:

- Hardware is a component that is physically attached to an embedded device and is used in its operation. It includes a microcontroller-based integrated circuit, a power supply, and an LCD display, among other things.
- Application software: Application software enables a user to perform a variety of tasks on an embedded device by modifying the code installed in the system.
- The real-time operating system (RTOS) controls how an embedded system operates. It serves as a bridge between hardware and application software, supervising the latter and providing a framework for allowing the processor to operate on the basis of scheduling in order to monitor the impact of latencies.

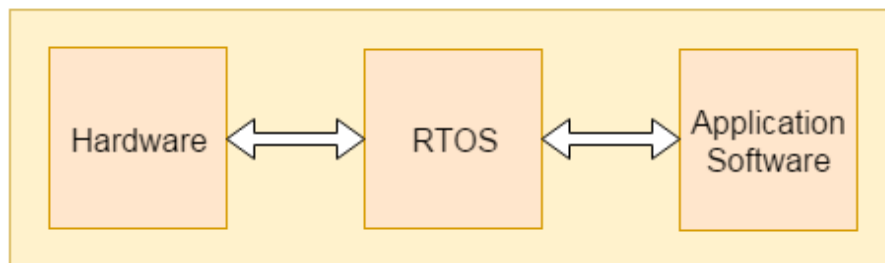


FIG 4.23 3 MAIN ELEMENTS OF EMBEDDED DEVICE

4.5.2.3 Characteristics of Embedded system

- An embedded system is software that is integrated into computer hardware to create a system that can be used for a number of purposes.
- Embedded systems are typically used to perform particular tasks and provide real-time output based on the embedded system's various characteristics.
- A smaller component of a larger device may be used for serving a more specific application to perform a variety of tasks using a hardware-software intermixing configuration in an embedded system.
- It has a high level of reliability and can perform calculations in real time.

4.5.2.4 Advantages

- The same hardware may be used for a number of purposes.
- Less power is required.
- The system's maintenance costs are reduced; and the system's performance and efficiency are improved.

4.5.2.5 Disadvantages

It took longer to develop a scheme. Because of the functional complexity, skilled engineers are needed because a single error may cause the entire project to be destroyed.

4.5.3 ARDUINO SOFTWARE (IDE)

The Arduino Software (IDE) includes a text editor for writing code, a message field, a text console, a toolbar with buttons for common functions, and a set of menus. It communicates with the Arduino and Genuino hardware by connecting to them and uploading programmes.

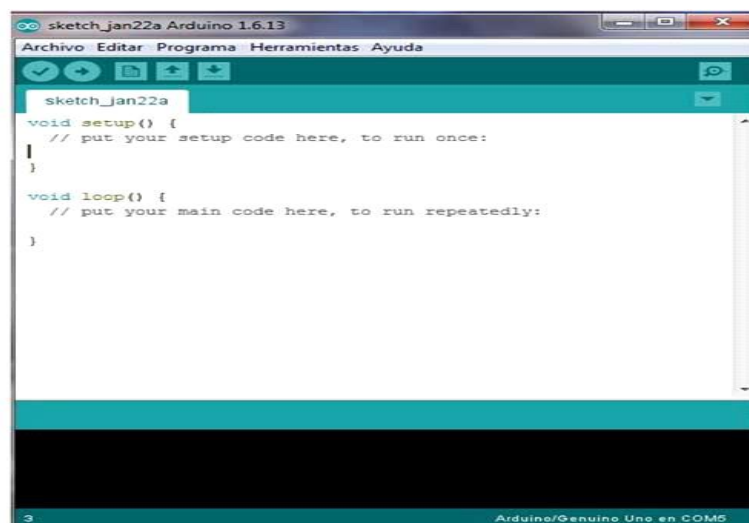


FIG 4.24 ARDUINO SOFTWARE IDE

4.5.3.1 WRITING SKETCHES

Sketches are programmes created with the Arduino Software (IDE). These sketches were created in a text editor and saved with the .ino file extension. Cutting/pasting, as

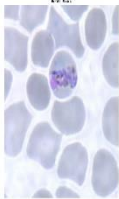
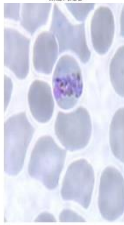
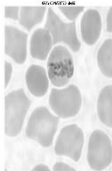
well as searching/replacing text, are all available in the editor. The message area shows errors and provides input when saving and exporting. The Arduino Software (IDE) outputs text to the console, which includes complete Error messages and other data can be found here. The designed board and serial port are shown in the window's bottom right corner. You can verify and upload programmes, make, open, and save sketches, and open the serial monitor using the toolbar buttons.

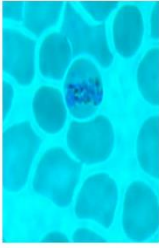
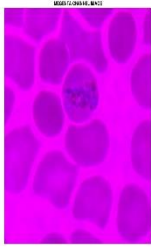
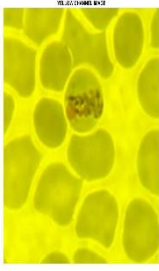
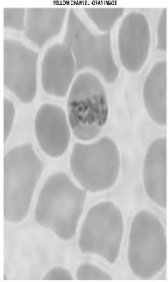
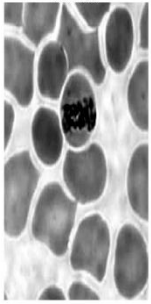
CHAPTER 5

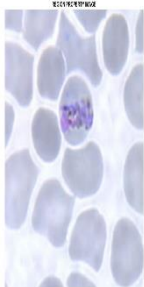
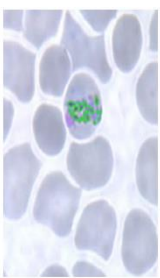
RESULT AND DISCUSSION

5.1 RESULT

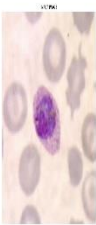
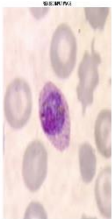
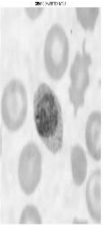
1. *Plasmodium malariae*

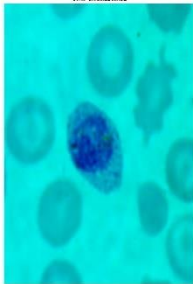
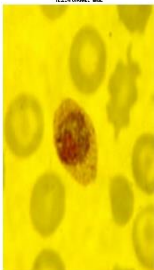
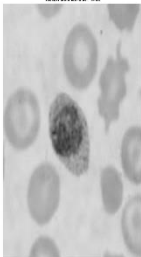
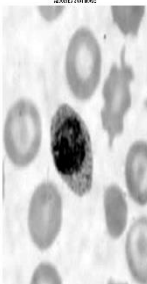
PREPROCESSING	1. INPUT IMAGE	2. RESIZED INPUT IMAGE
		
	3. GRAY CONVERTED	
		

SEGMENTATION	4. BLACK AND WHITE CONVERTED IMAGE 	5. CYAN CHANNEL IMAGE 
	6. MEGENTA CHANNEL IMAGE 	
	7. YELLOW CHANNEL IMAGE 	8. YELLOW CHANNEL GRAY IMAGE 
	9. ADJUSTED GRAY IMAGE 	

FEATURE EXTRACTION AND CLASSIFICATION	10. BLACK AND WHITE IMAGE	11. COMPLEMENT IMAGE
		
	12. REGION PROPERTY IMAGE	13. OUTPUT IMAGE
		

2. *Plasmodium ovale*

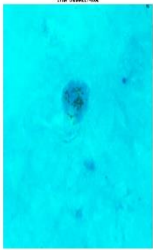
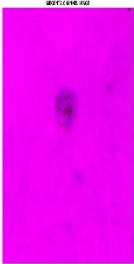
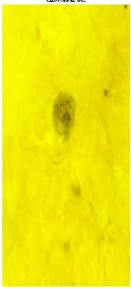
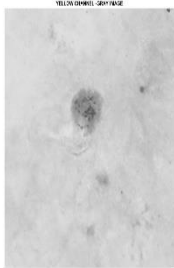
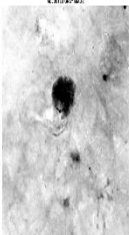
PREPROCESSING	1. INPUT IMAGE	2. RESIZED INPUT IMAGE
		
	3. GRAY CONVERTED	
		

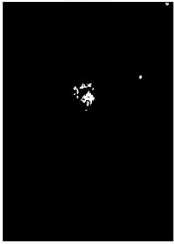
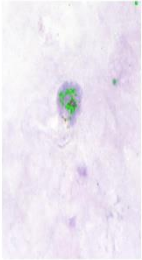
SEGMENTATION	4. BLACK AND WHITE CONVERTED IMAGE 	5. CYAN CHANNEL IMAGE 
	6. MEGENTA CHANNEL IMAGE 	
	7. YELLOW CHANNEL IMAGE 	8. YELLOW CHANNEL GRAY IMAGE 
	9. ADJUSTED GRAY IMAGE 	

FEATURE EXTRACTION AND CLASSIFICATION	10. BLACK AND WHITE IMAGE	11. COMPLEMENT IMAGE
	12. REGION PROPERTY IMAGE	13. OUTPUT IMAGE

3. *Plasmodium vivax*

PREPROCESSING	1. INPUT IMAGE	2. RESIZED INPUT IMAGE
	3. GRAY CONVERTED	

SEGMENTATION	4. BLACK AND WHITE CONVERTED IMAGE	5. CYAN CHANNEL IMAGE
		
	6. MEGENTA CHANNEL IMAGE	
		
	7. YELLOW CHANNEL IMAGE	8. YELLOW CHANNEL GRAY IMAGE
		
		9. ADJUSTED GRAY IMAGE

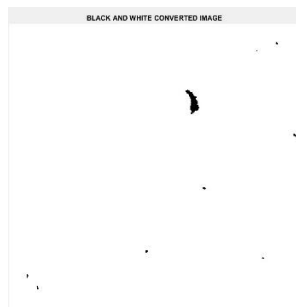
FEATURE EXTRACTION AND CLASSIFICATION	10. BLACK AND WHITE IMAGE 	11. COMPLEMENT IMAGE 
	12. REGION PROPERTY IMAGE 	13. OUTPUT IMAGE

4. *Plasmodium falciparum*

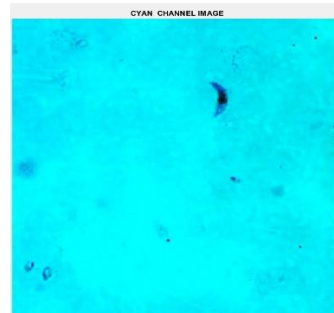
PREPROCESSING	1.INPUT IMAGE 	2. RESIZED INPUT IMAGE 
	3. GRAY CONVERTED 	

SEGMENTATION

4. BLACK AND WHITE
CONVERTED IMAGE



5. CYAN CHANNEL IMAGE



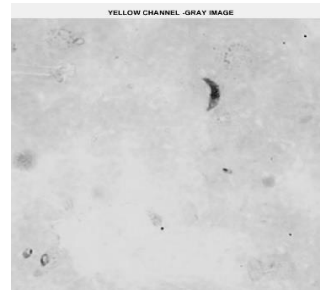
6. MEGENTA CHANNEL IMAGE



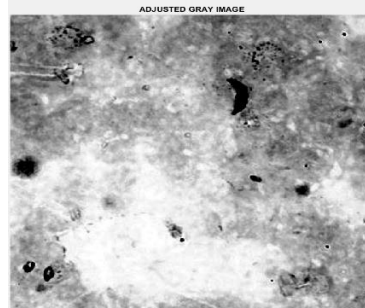
7. YELLOW CHANNEL IMAGE



8. YELLOW CHANNEL
GRAY IMAGE



9. ADJUSTED GRAY IMAGE

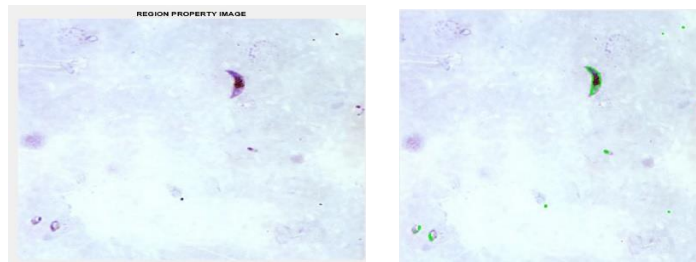


FEATURE
EXTRACTION
AND
CLASSIFICATION

10. BLACK AND WHITE IMAGE 11. COMPLEMENT IMAGE



12. REGION PROPERTY IMAGE 13. OUTPUT IMAGE



5. NO MALARIA

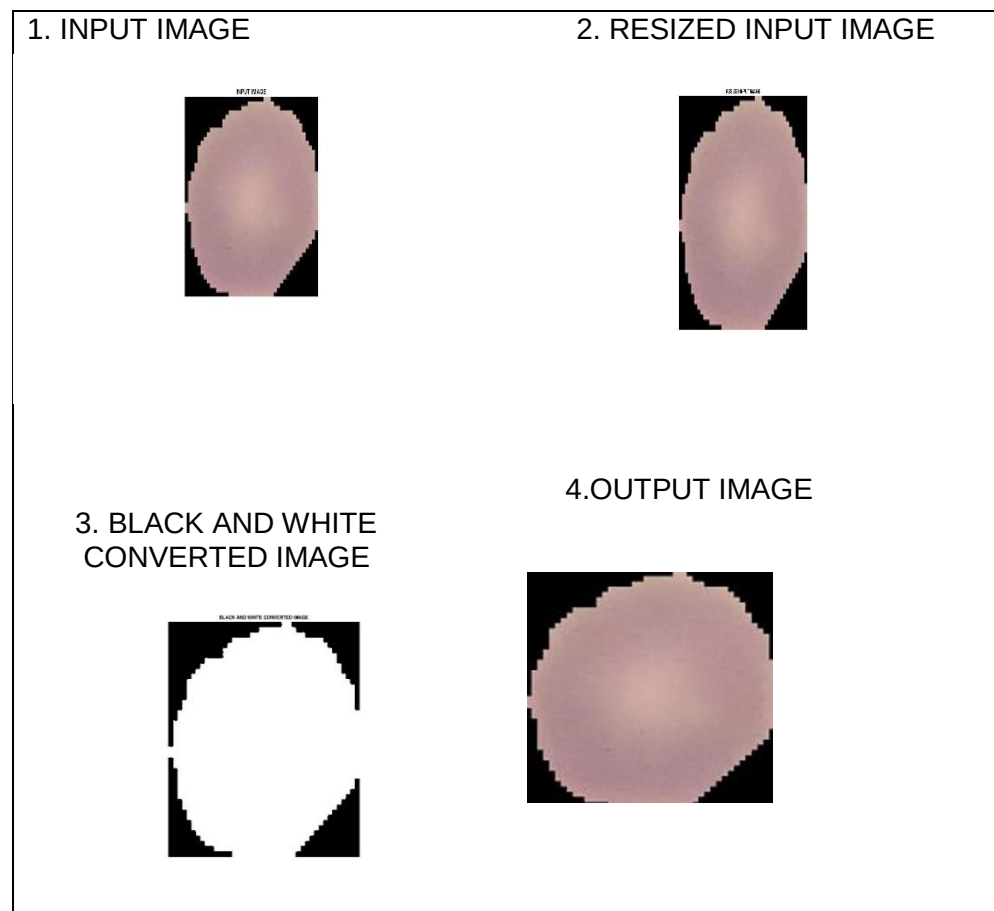


FIG 5.1 OUTPUT IMAGES

← → 🔍 Not secure | iotclouddata.tech/201/readtable/readtable.php ☆

IOT MONITORING SECTION

[Click here for home](#)

LogID	DATA	DATE	TIME
1	PLASMODIUM_FALCIPARUM	10/3/2021	14:13:30
2	PLASMODIUM_FALCIPARUM	1/1/1970	5:30:11
3	PLASMODIUM_FALCIPARUM	1/1/1970	5:30:33
4	PLASMODIUM_FALCIPARUM	10/3/2021	14:31:47
5	PLASMODIUM_FALCIPARUM	10/3/2021	14:36:3
6	PLASMODIUM_FALCIPARUM	10/3/2021	14:39:14
7	PLASMODIUM_FALCIPARUM	10/3/2021	15:9:23
8	PLASMODIUM_FALCIPARUM	10/3/2021	15:11:48

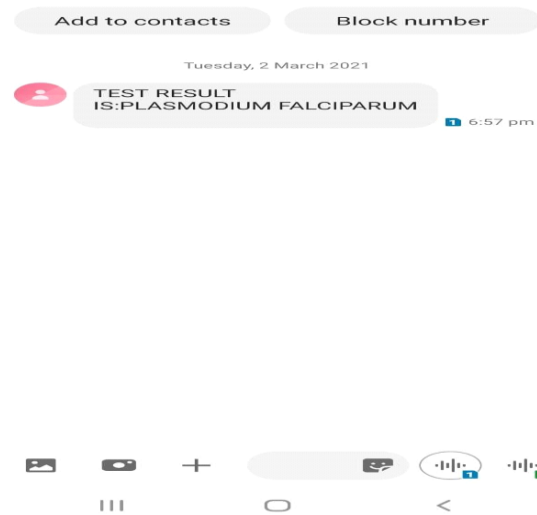


FIGURE 5.2 OUTPUT SHOWN IN IOT, LCD DISPLAY AND GSM

5.2 DISCUSSION

Currently, creating a single blood smear image takes anywhere from thirty seconds to several minutes to complete. This is largely due to the sliding mask circle detection algorithm, which was used to try to fit circles into the outline mask. The above-mentioned improvements in parasite mask processing and circle detection, as well as refactoring of existing MATLAB code, should have a significant impact. reduce the amount of time it takes to process one picture

CHAPTER 6

SUMMARY AND CONCLUSION

6.1 SUMMARY

As we could observe from the results, some of the blood cells overlap due to improper thin blood film preparation and some false positives because of artefacts. In the proposed method since we deal with species differentiation thin blood film is used. We suppose that, as it has been done for thin blood film, developing automated system for the thick blood film and integrating it with the proposed method can produce a good result. To decrease the human involvement, it is also better to have automatic image acquisition set up. This can be done by using stepper motors to slide the position of the microscope stage to focus the camera on a different field of view in the sample, and acquire the image from the microscope directly to the MATLAB. The proposed method was developed by using MATLAB software running on a windows platform. In addition to the benefit obtained from this thesis work, it will decrease cost, improve portability and accessibility because smart phones are widely used throughout the world, even in developing countries like Ethiopia where malaria disease is prevalent.

6.2 CONCLUSIONS

We've presented a method and created an algorithm for detecting malaria, as well as automatic malaria detection and quantification. In addition, we created a machine-learning-based training technique that is adaptable to the identification of malaria with other parasites, and we discussed how to improve the predictive value with the findings.

REFERENCES

1. A. Dohr, R. Modre-Opsrian, M. Drobits, D. Hayn, and G. Schreier, "The internet of things for ambient assisted living," published in Seventh International Conference on Information Technology (2010).
2. E.C.W. Lau, "Simple Bus Tracking System," published in Journal of Advanced Computer Science and Technology Research (2017) Vol. 3 (Issue no.1).
3. F. Boray Tek, Andrew Dempster and Izzet Kale. "Computer vision for microscopy diagnosis of malaria", published in malaria journal (2009) Vol 8 (Issue no.1) pp.153.
4. Gloria Diaz, Fabio A. Gonzalez, and Eduardo Romero "semiautomatic method for quantification and classification of erythrocytes infected with malaria parasites in microscopic images", published in journal of Biomedical Informatics (2017) Vol 42 (Issue no.2) pp.296.
5. H. Mansor, M.H.A. Shukor, S.S. Meskam, N.Q.A.M. Rusli, and N.S. Zamery, "Body temperature measurement for remote health monitoring system", published in IEEE International Conference on Smart Instrumentation, Measurement and Applications (2017).
6. J. Mohammed, C.H. Lung, A. Ocneanu, A. Thakral, C. Jones, and A. Adler, "Internet of things: Remote patient monitoring using web services and cloud computing", published in IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (Green Com) and

- IEEE Cyber, Physical and Social Computing (CPS Com) (2014).
7. K.M. Kumar and R.S. Venkatesan, "A Design Approach to Smart Health Monitoring Using Android Mobile Devices", published in IEEE International Conference on Advanced Communications, Control and Computing Technologies (2014).
 8. L. Salman, S. Salman, S. Jahangirian, M. Abraham, F. German, C. Blair, and P. Krenz "Energy Efficient IoT-Based Smart Home", published in 3rd World Forum on IEEE (2016).
 9. M.S. Jassas, Qasem, and Q.H. Mahmoud, "A smart system connecting e-health sensors and the cloud", published in IEEE 28th Canadian Conference on Electrical and Computer Engineering (2015).
 10. P.N. Jain, and T.P. Agarkar, "An embedded, GSM based, multi-parameter, real-time patient monitoring system and control; an implementation for ICU patients", published in World Congress on Information and Communication Technologies (2012).
 11. R. Bhilare and S. Mali "IoT based smart home with real time e-metering using e-controller", published in India Council International Conference (2017).
 12. R. F. Rahmat, Athmanathan, M. F. Syahputra and M. S. Lydia "Real time monitoring system for water pollution in Lake Toba", published in International Conference on Informatics and Computing (2016).
 13. Russell E Coleman, Nongnuj Maneechai, Nattawan Rachaphaew, Chalernpol Kumpitak, R. Scott Miller, Virat Soyseng, Krongthong Thimasarn, and Jetsumon Sattabongkot "Comparison of field and expert laboratory microscopy for active surveillance for asymptomatic plasmodium falciparum and plasmodium vivax in western thailand" (2012) pp 141–144,
 14. S. Greene, H. Thapliyal, and D. Carpenter, "IoT based Fall Detection for Smart Home Environments", published in Nano electronic and Information Systems (2016).
 15. S. Li, L. D. Xu, and S. Zhao "The Internet of Things: A Survey", published in Information Systems Frontiers (2015) Vol.17 (Issue no.2) pp. 243–259.
 16. Wendy Prudhomme O'Meara, F. Ellis McKenzie, Alan J Magill, J. Russ Forney, Barnyen Permpaich, Carmen Lucas, Robert A Gasser, and Chansuda

Wongsrichanalai "Sources of variability in determining malaria parasite density by microscopy", published in Am J Trop Med Hyg (2005) Vol 73 (Issue no.3) pp:593–598.

17. Wendy Prudhomme O'Meara, Mazie Barcus, Chansuda Wongsrichanalai, Sinuon Muth, Jason D Maguire, Robert G Jordan, William R Prescott, and F. Ellis McKenzie "Reader technique as a source of variability in determining malaria parasite density by microscopy" (2006).
18. Whitmore, A. Agarwal, and L. Xu "The Internet of Things– A Survey of Topics and trends", published in bio information.
19. X. Luo, T. Liu, J. Liu, X. Guo, and G. Wang "Design and implementation of a distributed fall detection system based on wireless sensor networks", published in EURASIP Journal on Wireless Communications and Networking (2012) Vol. 12 (Issue no.1) pp. 118.

APPENDIX

HARDWARE PROGRAM

```
#include<LiquidCrystal.h>           //HEADER FILE FOR LCD
LiquidCrystal lcd(8, 9, 10, 11, 12, 13); //(RS=8, EN=9, D4=10, D5=11, D6=12, D7=13)
#include<SoftwareSerial.h>          //HEADER FILE FOR GSM
SoftwareSerial gsm(6, 7);           //RX=6, TX=7

#define buzzer 5
#define heart 4
#define temperature A0
#define respiratory 3

int heartbeat();
void sms(String message);
void iot(String stringdata);        //iot function

void setup()
{
```

```

// put your setup code here, to run once
Serial.begin(9600); //for iot
gsm.begin(9600);    //for gsm
lcd.begin(16, 2);   //for lcd
pinMode(buzzer, OUTPUT);
pinMode(heart, INPUT);
pinMode(temperature, INPUT);
pinMode(respiratory, INPUT);
digitalWrite(buzzer, LOW);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("PLASMODIUM");
lcd.setCursor(5, 0);
lcd.print("FALCIPARUM");
delay(1000);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print(" DETECTION IN ");
lcd.setCursor(0, 0);
lcd.print("BLOOD CELL IMAGS");
delay(1000);
}

void loop ()
{
// put your main code here, to run repeatedly
int temp = analogRead(temperature) / 2;
int heartvalue = heartbeat();
int breath = digitalRead(respiratory);
while (Serial.available() > 0)

```

```

{
  char b = Serial.read();
  if (b == 'A')
  {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("PLASMODIUM");
    lcd.setCursor(0, 1);
    lcd.print("  MALARIAE ");
    delay(1500);
  }
  else if (b == 'B')
  {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("PLASMODIUM");
    lcd.setCursor(0, 1);
    lcd.print("  OVALE ");
    delay(1500);
  }
  else if (b == 'C')
  {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("PLASMODIUM");
    lcd.setCursor(0, 1);
    lcd.print("  VIVAX ");
    delay(1500);
  }
  else if (b == 'D')

```

```

{
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("PLASMODIUM");
  lcd.setCursor(0, 1);
  lcd.print("FALCIPARUM");
  digitalWrite(buzzer, HIGH);
  sms("9884071809", "PLASMODIUM FALCIPARUM");
  digitalWrite(buzzer, LOW);
  iot("*PLASMODIUM FALCIPARUM#");
}
else if (b == 'E')
{
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("NO MALARIA");
  lcd.setCursor(0, 1);
  lcd.print(" DETECTED ");
  delay(1500);
}
}

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("BT:");
lcd.print(temp);
lcd.setCursor(8, 0);
lcd.print("HB:");
lcd.print(heartvalue);
if (breath == 0)
{

```

```

    lcd.setCursor(0, 1);
    lcd.print("PATIENT BREATHING");
}
else
{
    lcd.setCursor(0, 1);
    lcd.print("NOT BREATHING");
}
delay(1000);
}

int heartbeat()
{
    int j = 120;
    int i = 0;
    while (j > 0)
    {
        if (digitalRead(heart) == LOW)
            i++;
        j--;
        delay(14);
    }
    return i;
}

void sms(String phno, String message)
{
    gsm.println("AT");
    delay(200);
    gsm.println("AT+CMGF=1");
    delay(200);
    gsm.println("AT+CMGS=\"+91\" + phno + \"\\r\\n\"");
    delay(200);
    gsm.println("TEST RESULT IS:" + message);
}

```

```

    delay(200);
    gsm.println((char)26);
    delay(1200);
}
void iot(String stringdata)    //iot function
{
    for (int i = 0; i < stringdata.length(); i++)
    {
        Serial.write(stringdata[i]);
    }
    delay(4000);
}

```

SOFTWARE PROGRAM

%%%%%%%%%%**MALARIA CLASSIFICATION USING MACHINE LEARNING** %%%%%%%%%%

clc

close all;

clear all;

warning off;

%%%%%%%%%**Get the input image** %%%%%%%%%%
[filename,pathname]=uigetfile('*.jpg');

%%%%%%%% **Read Input Image** %%%%%%%%%%
I=imread([pathname,filename]);

figure,imshow(I),title('INPUT IMAGE');

%%%%%%%% **Resize The input image** %%%%%%%%%%
I=imresize(I,[512 512]);

figure,imshow(I);title('RESIZED INPUT IMAGE');

%%%%%%%% **GRAY CONVERSION** %%%%%%%%%%

It=rgb2gray(I);

```

figure,imshow(It),title('GRAY CONVERTED IMAGE');

%%%%%%%%%BLACK AND WHITE CONVERSION %%%%%%%%%%
It=im2bw(It);

figure,imshow(It),title('BLACK AND WHITE CONVERTED IMAGE');

%%%%%%%%% REMOVE SMALL OBJECTS %%%%%%%%%%

It=bwareaopen(It,2000);

stats=regionprops(It,'ALL');

den=[stats.Solidity];

area=[stats.Area];
if (area<210000)

ccf=5;
figure,imshow(I);
hold on

msgbox('NO MALARIA CELL DETECTED');

else

%%%%%%%%%CONVERT RGB2CMYK COLOUR SPACE CONVERSIONS
%%%%%%%%%

cyan=I;
magenta=I;
yellow=I;

cyan(:,:,1)=0;
cyan(:,:,1)=0;
magenta(:,:,2)=0;
magenta(:,:,2)=0;
yellow(:,:,3)=0;
yellow(:,:,3)=0;

figure,imshow(cyan),title('CYAN CHANNEL IMAGE');
figure,imshow(magenta),title('MEGENTA CHANNEL IMAGE');
figure,imshow(yellow),title('YELLOW CHANNEL IMAGE');

figure,
subplot(1,3,1),imshow(cyan),title('CYAN');
subplot(1,3,2),imshow(magenta),title('MEGENTA');
subplot(1,3,3),imshow(yellow),title('YELLOW');

```

**%%%%%%%%SELECT THE YELLOW CHANNEL AND PERFORM RGB TO GRAY
CONVERSION %%%%%%%%%**

%% RGB to Gray conversion

```
[m n o]=size(yellow);
if o==3
    gray=rgb2gray(yellow);
else
    gray=yellow;
end
figure,imshow(gray);title('YELLOW CHANNEL -GRAY IMAGE');
```

%%%%%%%%ADJUST THE CONTRAST OF THE GRAY CHANNEL IMAGE%%%%%%%%

```
ad=imadjust(gray);
figure,imshow(ad);title('ADJUSTED GRAY IMAGE');
```

**%%%%%%%%TO PERFORM BINARY CONVERSION ON THE ADJUSTED GRAY
IMAGE%%%%%%%%**

```
bw=im2bw(gray,0.5);
figure,imshow(bw);title('BLACK AND WHITE IMAGE');
```

%%%%%%%%TAKE COMPLEMENT TO THE BLACK AND WHITE IMAGE %%%%%%%%%

```
bw=imcomplement(bw);
figure,imshow(bw);title('COMPLEMENT IMAGE');
```

```
bww=bw;
```

%%%%%%%%TO LABEL THE SMALL OBJECTS REMOVED IMAGE%%%%%%%%

```
label=bwlabel(bww);
```

%%%%%%%%APPLY REGION SEGMENTATION (REGION PROPERTIES)%%%%%%%%

```
S=regionprops(label,'ALL');
figure,imshow(I),title('REGION PROPERTY IMAGE');
```

%%%%%%%%FIND THE PERIMETER FOR THE BOUNDARY%%%%%%%%

```
u=bwperim(bww);
```

%%%%%%%%CONVERT TO DOUBLE FOEMAT%%%%%%%%

```
u=im2double(u);
```

**% % %%%TO ASSIGN THE BOUNDARIES FOR THE DETECTION OF NUMBER
OF WBC CELLS %%%%%%%%%**

```
[B,L] = bwboundaries(bww,'noholes');
imshow(label2rgb(L, @jet, [.5 .5 .5]))
figure,imshow(I,[]);
hold on
for k = 1:length(B)
    boundary = B{k};
    plot(boundary(:,2), boundary(:,1), 'g', 'LineWidth', 2)
```

```

end
% %%%FEATURE EXTRACTION%%%%%%%%

%%%%%%%%TO TAKE COLOUR FEATURES%%%%%%%%
R = mean2(I(:,:,1));
G = mean2(I(:,:,2));
B = mean2(I(:,:,3));
Co_Fea = [R G B];

%%%%%%%%TO TAKE GEOMETRICAL FEATURES%%%%%%%%
Area = mean([S.Area]);

for i = 1:size(S,1)
    diameters(i) = mean([S(i).MajorAxisLength S(i).MinorAxisLength])/2;
    radii(i) = diameters(i)./2;
end
perimeter = mean([S.Perimeter]);
ecc = mean([S.Eccentricity]);
elg = mean([S.Solidity]);
Elongation = mean(Elong);
dia = mean(diameters);
rad = mean(radii);
Geome_Fea = [Area dia rad perimeter ecc elg];

%%%%%%%%TAKE IMPORTANT TEXTURE FEATURES %%%%%%%%%

g = graycomatrix(bww);
stats = graycoprops(g,'Contrast Correlation Energy Homogeneity');
Contrast = stats.Contrast;
Correlation = stats.Correlation;
Energy = stats.Energy;
Homogeneity = stats.Homogeneity;
Tex_feat= [Contrast,Correlation,Energy,Homogeneity];

feat_tot=[Co_Fea Geome_Fea Tex_feat];

% % %%%LOAD ALL THE FEATURES%%%%%%%%

load featuremalaria.mat

%%%%%%%%CLASSIFICATION%%%%%%%%
%%%%%%%%K-NEAREST NEIGHBOUR CLASSIFICATION
%%%%%%%%

label=ones(1,128);
label(1:30)=1;
label(31:57)=2;
label(58:93)=3;

```

```

label(94:128)=4;

model=fitcknn(fea_malaria1,label);
result=predict(model,feat_tot);

if result==1

    ccf=1;
    msgbox('plasmodium malariae');

elseif result==2
    ccf=2;
    msgbox('plasmodium ovale');

elseif result==3
    ccf=3;
    msgbox('plasmodium vivax ');

elseif result==4
    ccf=4;
    msgbox('plasmodium falciparam');
end

%%%%%%%%% ACCURACY %%%%%%%%%%%

ACTUAL=[1,1,1,1,1,1,2,2,2,2,2,3,3,3,3,3,4,4,4,4,4,4,5,5,5,5,5,5];

PREDICTED=[1,1,1,2,1,1,2,2,2,1,2,2,3,0,3,3,3,2,4,4,4,3,4,4,5,5,5,5,5,5];
% %%%%%%%%%%knn%%%%%%%%%
EVAL=Evaluate(ACTUAL,PREDICTED);

idx = (ACTUAL()==1);

p = length(ACTUAL(idx));
n = length(ACTUAL(~idx));
N = p+n;

tp = sum(ACTUAL(idx)==PREDICTED(idx));
tn = sum(ACTUAL(~idx)==PREDICTED(~idx));
fp = n-tn;
fn = p-tp;

tp_rate = tp/p;
tn_rate = tn/n;

accuracy = (tp+tn)/N;
sensitivity = tp_rate;
specificity = tn_rate;

```

```

precision = tp/(tp+fp);
recall = sensitivity;
f_measure = 2*((precision*recall)/(precision + recall));
gmean = sqrt(tp_rate*tn_rate)
disp(' accuracy=');
disp(accuracy*100);
disp('sensitivity=');
disp(sensitivity*100);
disp('specificity=');
disp(specificity*100);
disp('precision=');
disp(precision*100);
disp('recall=');
disp(recall*100);
disp('f_measure=');
disp(f_measure*100);
disp('gmean=');
disp(gmean*100);

```

```

end

```

%%%%%%%%HARDWARE CONNECTED OUTPUT FOR LCD DISPLAY%%%%%%%%

```

instrumentObjects=instrfind; % don't pass it anything - find all of them.
delete(instrumentObjects)
a=serial('COM5','BaudRate',9600);
fopen(a);
if ccf==1
    fprintf(a, '%c','A');%%%%%%%%A=plasmodium malariae%%%%%%%%
elseif ccf == 2
    fprintf(a, '%c','B'); %%%%B=plasmodium ovale %%%%%%%%%%
elseif ccf == 3
    fprintf(a, '%c','C'); %%%%%%%%%C=plasmodium vivax%%%%%%%%
elseif ccf == 4
    fprintf(a, '%c','D'); %%%%%%%%%D=plasmodium falciparam%%%%%%%%
elseif ccf == 5
    fprintf(a, '%c','E'); %%%%%%%%%F=no malaria detected %%%%%%%%%
end

fclose(a);

```

%%% END THE PROGRAM %%%%