POLITEKNIK SULTAN SALAHUDDIN ABDUL AZIZ SHAH

JABATAN KEJURUTERAAN ELEKTRIK

DIPLOMA KEJURUTERAAN ELEKTRIK(PERUBATAN)

SESI 1 2022/2023

PROJECT: FINAL REPORT

PROJECT TITLE	**Heartbeat Monitoring System (Heart Rate Monitor)**
Student Name	**SUE LYN ANAK NYAMBONG**
NO. MATRIC	**08DEU20F1040**
COURSE'S CODE	**DEE50102**
LECTURER NAME	**PUAN NOR KHARUL AINA BINTI MAT DIN**

CONFIRMATION OF THE PROJECT

The project report titled " HEARTBEAT MONITORING SYSTEM (HEART RATE MONITOR) has been submitted, reviewed, and verified as a fulfills the conditions and requirements of the Project Writing as stipulated.

Checked by:

Supervisor's name : PUAN NOR KHARUL AINA BINTI MAT DIN

Supervisor's signature:

Date:

Verified by:

Project Coordinator name : SUE LYN ANAK NYAMBONG

Signature of Coordinator :

Date :

"I acknowledge this work is my own work except the excerpts I have alreadyexplained to our source"

1. Signature :

Name : SUE LYN ANAK NYAMBONG

Registration Number: 08DEU20F1040

Date:

DECLARATION OF ORIGINALITY AND OWNERSHIP

TITLE : HEARTBEAT MONITORING SYSTEM (HEART RATE MONITOR)

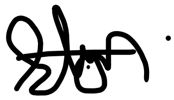
SESSION: SESI 1 2021/2022

SUE LYN ANAK NYAMBONG and my registrationnumber is 08DEU20F1040 is a final year student of **Diploma in Electrical Engineering, Department of Electrical, Polytechnic Sultan Salahuddin Abdul AzizShah**, which is located at **Persiaran Usahawan, 40150 Shah Alam, Selangor**. Hereinafter referred to as 'the Polytechnic').

1. I acknowledge that 'The Project above' and the intellectual property therein is theresult of our original creation /creations without taking or impersonating any intellectual property from the other parties.

2. I agree to release the 'Project' intellectual property to 'The Polytechnics' to meetthe requirements for awarding the **Diploma in Electrical Engineering** to me.

Made and in truth that is recognized by;
SUE LYN ANAK NYAMBONG
(Identification card No: 020212-13-0148)



....
SUE LYN ANAK NYAMBONG

In front of me, PUAN NOR KHARUL AINA BINTI MAT DIN

....
NOR KHARUL AINA BINTI MAT DIN

As a project supervisor, on the date:

APPROVAL PAGE FOR FINAL YEAR PROJECT

Submitted in partial full of requirements for diploma of Electronic Medical Engineering at Polytechnics Premier Sultan Salahuddin Abdul Aziz Shah by:

Candidate: SUE LYN ANAK NYAMBONG

Department/Field of Concentration: Electrical Engineering Department

TITLE: Heartbeat Monitoring System (Heart Rate Monitor)

APPROVED:

……………………………………………..   ……………………………………………….

(SUPERVISOR PROJECT)   (SIGNATURE)

ACKNOWLEDGEMENTS

First, I have taken efforts in this Project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them. I am highly indebted to PUAN NOR KHARUL AINA BINTI MAT DIN for their guidance and constant supervision as well as for providing necessary information regarding the Project & also for their support in completing the Project. I would like to express my gratitude towards my parents and my friends for their kind co-operation and encouragement which help me in completion of this Project. I would like to express my special gratitude and thanks to industry persons for giving me such attention and time. Finally, to the lecturers of the Department of Electrical Engineering, Faculty of Diploma Studies who are directly or indirectly involved and give ideas for the success of this project. Without your help, guidance and guidance, the project will not succeed very well. Thank you. My thanks and appreciation are also extended to my colleagues in developing the Project and those who are willing to help me with their abilities.

ABSTRACT

These days, we have an increasing number of people suffering from serious diseases such as heart problems, blood pressure, and other dangerous health problems regardless of age whether young or old. Some people are unaware of their state of health and may be careless about their health. In addition, there are some people who are lazy and might be afraid to do health check-ups at the hospital. The Heart Rate Monitoring system is proposed in this project to reduce the problems mentioned above. This project is to detect the heart rate beat using the Pulse sensor. It can help people easier to check pulse at home or at work and more convenience to check it on their own anytime. they will be able to check on their own and at any time. These canals ease the burden of the hospitals due to people can check on their own and can save a lot of time. Normal pulse rate for adult is 60 to 100 beats per minute while for a child is 70 to 100 beats per minute and for those who are active in sports can be 40 beats per minute. They only need to go to the hospital if their pulse is not in a normal state such as the pulse is in high rate or in lower rate and not as it should be. if the pulse is abnormal, it is very likely that the person has a dangerous disease and needs to get other hospital for further examination. This project also can change the perspective of typical humans that still lazy or afraid to the hospitals. Since a healthy body is the most important thing, we must have to continue our life, I hope that this project might iii open a whole new world towards community to be more alert of their health by always do a regular pulse rate check. Next, this project is to implement a heart rate monitoring system using Internet of things that use Microcontroller and Pulse sensor method. The result that going to be yielded in this project is a through the web that will display the output, this can help the user measure their pulse rate beat and solved their problems.

ABSTRAK

Zaman sekarang, kita mempunyai peningkatan jumlah orang yang menderita penyakit serius seperti masalah jantung, tekanan darah, dan masalah kesihatan berbahayayanglain tanpa mengira usia sama ada muda atau tua. Sebilangan orang tidak menyedari keadaan kesihatan mereka dan mungkin tidak peduli dengan kesihatan mereka. Di samping itu, ada sebilangan orang yang malas dan mungkin takut untuk melakukanpemeriksaan kesihatan di hospital. Sistem Pemantauan Denyut Nadi dicadangkandalam projek ini untuk mengurangkan masalah yang disebutkan di atas. Projekini adalah untuk mengesan denyut nadi dengan menggunakan sensor Pulse. Ia dapat membantu orang untuk memeriksa nadi dengan lebih mudah di rumah atau di tempat kerja dan lebih banyak kemudahan untuk memeriksanya sendiri bila-bila masa. mereka akan dapat memeriksa sendiri dan pada bila-bila masa. Ini juga dapat meringankan beban hospital kerana orang boleh memeriksa sendiri dandapat menjimatkan banyak masa. Denyutan nadi normal untuk orang dewasa ialah60hingga 100 denyutan seminit sementara bagi seorang kanak-kanak adalah 70hingga100 denyutan seminit dan bagi mereka yang aktif dalam sukan boleh menjadi 40denyut seminit. Mereka hanya perlu pergi ke hospital jika nadi mereka tidak dalam keadaan normal seperti nadi pada kadar tinggi atau pada kadar yang lebih rendah dan tidak seperti yang sepatutnya. jika nadi tidak normal, kemungkinan orang tersebut v mempunyai penyakit berbahaya dan perlu pergi ke hospital untuk pemeriksaan lanjut. Projek ini juga dapat mengubah perspektif manusia biasa yang masih malas atau takut ke hospital. Oleh kerana badan yang sihat adalah perkara terpenting yang harus kitamiliki untuk meneruskan kehidupan, saya berharap projek ini dapat membukaduniabaru kepada masyarakat agar lebih peka terhadap kesihatan mereka denganselalumelakukan pemeriksaan kadar nadi secara berkala. Seterusnya, projek ini adalah untuk melaksanakan sistem pemantauan kadar nadi menggunakan Internet of things (iot)yang menggunakan kaedah Mikrokontroler dan sensor nadi. Hasil yang akan dihasilkan dalam projek ini adalah melalui laman web yang akan menampilkan output, ini dapat membantu pengguna mengukur denyut nadi mereka dan menyelesaikan masalah mereka.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURES **TITTLE**

CHAPTER 1

1.1 Introduction

Over the last two decades, heart rate monitors (HRMs) have become a widely used training aid for a variety of sports. Also, this technology was developed rapidly to help many patients who suffer from cardiovascular diseases such as arrhythmias. Monitoring the heartbeats rate could be sufficient early indicator to these abnormalities and a non-invasive diagnostic tool.

1.2 Background of Project

A product has been created from the result of an idea discussed with my supervisor, which is a system for monitoring the heartbeat. After I am researching about this heart disease problem, I found that many patients suffer from heart problems because of not having early prevention and the object I want to create this is functional and applications that we can develop from this device. I, to make a heart rate monitor to ease the burden of heart problem patients and also suitable for repair body balance in doing exercise by monitoring range every pulse. I In addition, it can also improve the comfort and safety of users to do activities in daily life.

1.3 Problem Statements

Heart disease most often occurs when cholesterol accumulates and forms "plague" in a coronary artery. With blood flow impeded, the heart becomes starved for oxygen, causing chest pain (angina). Coronary artery disease, the restriction of blood flow to the heart, is the leading cause of death in Malaysia. If a blood clot forms and completely obstructs the artery, a heart attack (myocardial infarction) car occurs. The statistical death caused by heart attack in Malaysia has been increasing from one year to another. This shows that heart disease is one of the horror killers living in every human being. During a heart attack, heart muscle is depraved of oxygen and will literally die if the artery remains blocked. The first few hours are critical in saving much of dying muscles and preventing permanent heart damage.

1.4 Project Objective

- ✓ To study about detecting human heart rate.
- ✓ To design a heart rate tracking system that is safer to use in an easy -to -carry form.
- ✓ To design a system for doctor to monitor user's pulse rate and detect whether their pulse rate in normal rate or not.

1.5 Project Scope

Workers who consume a lot of high energy cause unstable heart rate and patients with unstable heart problems.

1.6 Important of research

The use of these two tools still has disadvantages because it is always done repeatedly and requires concentration to obtain accurate values. This method will inevitably cause some problems, such as the time required by medical personnel to determine the diagnosis, increase the burden of medical staff in conducting examinations, and increase the cost that must be incurred by the management in providing medical equipment.

1.7 PROJECT SIGNIFICANCE

Heart rate monitors can make people who have healthy lifestyle problems especially heart disease because of this they are able to keep their heart health from being seriously injured or ill and have body balance. This device can also make people aware of their symptom's lower limb problems such as abnormal pulse problems. Furthermore, it also recognizes the appropriate ones with the pulse of our body when doing our daily exercise. Finally, we can prevent the onset of a heart attack from a more dangerous level.

1.8 CHAPTER SUMMARY

In general, this section shows the thinking of the effort that needs to be done for reduce the occurrence of unhealthy body due to heart disease between Malaysians in particular. This section also contains more information about why we need to take care of our hearts. Take care of the good healthy is important, before it becomes a big problem in our lives, perhaps now the effect of having lower limb problems, not checking the normal pulse in a suitable day when doing work and an unbalanced body has not been seen, but over time the effect will be interesting.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

In this chapter is significance it will cover research of this project and information related the investigation. This part additionally will talk about research that comparative with this undertaking. A few articles and journals have been checked on furthermore, be references to this venture since it previously done to increase a few information. This section also is about anatomy research for developing this device.

2.2 Literature Review Topic 1

Tittle: Heartbeat Monitoring and Alert System Using GSM Technology.

Item/Title	Objectives	Problem Statement	Methodology	Sensor Used
PAPER 1	Heart rate monitoring is a vital aspect of maintaining heart health. People from different age groups have different ranges for maximum and minimum values of heart rate, the monitoring system must be compatible enough to tackle this scenario.	i. Heart attack patient dies in silent condition without anyone noticing such as sleeping. ii. Does not have constant monitoring of the heart condition.	It senses the heartbeat of a person and converts it in the form of electrical signals and pulses. The signals are amplified using a signal conditioning circuit and processed by a controller. The frequency of the signal depends on the heartbeat rate, this lays down the basic principle of the HB measuring system.	The **infrared sensor** is responsible for sending infrared light to the body. This sensor has a pair of transmitter and receiver. Using photodiode can also detect reflective light from the body and this signal is sent to the microcontroller to detect heartbeat.

2.3 Literature Review Topic 2

Tittle: Heartbeat and Temperature Monitoring System for Remote Patients using Arduino.

PAPER 2	A heartbeat sensor is a device that helps us to detect the heart rate of a person. It helps a person to get the heart rate checked up and thus helps us to monitor the heartbeat. It is mainly a heartbeat sensor framework that is connected with the database and there is also an emergency system that is associated with any hospital management.	The heartbeat sensor device with a database links to the hospital network is beneficial to patients and the community where the introduction of such a device can reduce the risk of the patient as well as save hospital bills, waiting time and reduce hospital traffic. Wireless sensors for heart rate and body temperature are incorporated in the proposed health monitoring program but this paper focuses only on heartbeat sensors for hospital management with a database communication system.	Sensor health monitoring is a list of techniques performed to keep a system in fully functional conditions that can be restricted to observe the current state of the system with these kinds of observations requiring maintenance and repair.	Heartbeat can be measured in two ways - either manually by checking the pulse rate of the finger, wrist, and neck or it can be measured through digitally used sensors but in that case, the patient must carry a sensor device like Arduino UNO.

2.4 Literature Review Topic 3

Tittle: Heartbeat Monitoring and Alert System Using Gsm Technology

PAPER 3	Heart rate is the number of heartbeats per unit of time, typically expressed as beats per minute. Heart rate can vary as the body's need to absorb oxygen and excrete carbon dioxide changes during exercise or sleep. The measurement of heart rate is used by medical professionals to assist in the diagnosis and tracking Of medical conditions. It is also used by individuals, such as athletes, who are interested in monitoring their heart rate to acquire maximum efficiency.	Changes in lifestyle and unhealthy eating habits have resulted in a dramatic increase in incidents of heart and vascular diseases. Furthermore, heart problems are being increasingly diagnosed on younger patients. Worldwide, Coronary heart disease is now the leading cause of death. Thus, any improvements in the diagnosis and treatment tools are welcomed by the medical community	Heart rate is measured under controlled conditions like blood measurement, heart beat measurement, and Electrocardiogram (ECG). However, there is a great need that patients are able to measure the heart rate in the home environment as well. The heart rate rises gradually during exercises and returns slowly to the rest value after exercise. The rate at which the pulse returns to normal is an indication of the fitness of the person.	Heart beat sensor will generate digital pulse corresponding to each beat. This pulse is counted by interfacing heart beat sensor to microcontroller to pin no. 15.

2.5 Literature Review Topic 4

Tittle: Heartbeat Sensor System for Remote Health Monitoring

PAPER 4	The heart is undeniably an essential part of the human body system. The heart's missions are to pump blood to the lungs to expel the waste carbon dioxide that resulted from the respiration and pump blood that will deliver oxygen throughout the human body's circulation system. While the blood that carries oxygen being pumped throughout the human body, the heart will provide the oxygen to all cells.	Heartbeat and body temperature are the major signs that are routinely measured by physicians after the arrival of a patient [1]. Heart rate refers to how many times a heart contracts and relaxes in a unit of time (usually per minute). Heart rate varies for different age groups. For a human adult of age 18 or more years, a normal resting heart rate is around 72 beats per minute (bpm). A lower heart rate at rest implies more efficient heart function and better cardiovascular fitness. Babies have a much higher rate than adults around 120 bpm and older children have heart rate around 90 bpm [2].	The proposed system is based on Heartbeat sensor (HBS). Sensors are connected to the hardware and attached to the patient"s body. Using sensor values of heart rate sensed and these values are sent to the android mobile phone. The Android application takes the values and these values are analyzed for predicting heart attack. This application is developed so that the patient is able to seek treatment even in the physical absence of the doctor and irrespective of the patient"s location.	Sensors are linked to the hardware and attached to the patient"s body. Using sensors values of heart rate sensed and these values are sent to the android mobile phone. Android application collects heart rate and these values are analyzed for Guessing heart attack.

2.6 Literature Review Topic 5

Tittle: Heartbeat rate monitoring system by pulse technique using HB sensor

PAPER 5	The heart is one of the most important organs of the human body. It acts as a pump that circulates the blood, carrying oxygen and nutrients in the body to keep it functioning. Today, several advanced methods have emerged in the field of medicine, which aim to improve the efficiency of the medical services. And these new methods are akin to a technical identification of the disease with high precision, a new radiological technology and a fashionable identification of medical health.	A heartbeat can be defined as a two-part pumping action of the heart which occurs for almost a second. It is produced due to the contraction of the heart. When blood collects in upper chambers, the SA(Sino Atrial) node sends out an electrical signal which in turn causes the atria to contract.	In this system we use the pulse sensor with Arduino Uno and Bluetooth HC-05 module, the pulse sensor is placed on the finger and it measures the heart rate and then sends the heart rate to android mobile application via Bluetooth. Early recognition of the disease is very vital in preventing more complications in the future.	An IoT-based human heartbeat rate monitoring And control system is developed. This system uses the capability of a heart pulse sensor for data acquisition. A human's heartbeat is captured as data signals and processed by the microcontroller.

2.7 CHAPTER SUMMARY

This section focuses on five different sections. This section reviews project software and hardware. Different components are used in different journals help to distinguish which one is more appropriate to use project. After that, for the anatomical study of heart disease to ensure that all objectives to develop this device is relevant to the facts about heart pressure. This part also made I better understand about heart stress and its effects for humans if any unhealthy heart condition.

CHAPTER 3

3.1 METHODOLOGY

This chapter will discuss about the methods and project flow from the beginning until the end of the project. The development of this project is carried out by applying the methodology of Agile Development. Among the sub-methodologies of Agile Development, Iterative and Incremental model is used to develop through repeated cycle which is iterative. The project can be proceeds if there are any changes in the middle of the project.

3.2 Project Design and Overview

In this project, we will design Heartbeat/Pulse/BPM Rate Monitor using Arduino & Pulse Sensor. You can interface the pulse sensor with Arduino for monitoring Heartbeat/Pulse/BPM Rate. I used a 204 LCD panel to display the pulse rate in BPM. You can even use a 162 LCD display. This sensor is quite easy to use and operate. Place your finger on top of the sensor and it will sense the heartbeat by measuring the change in light from the expansion of capillary blood vessels. You can use this sensor with ESP8266 to upload the BPM data to the Internet. One of such projects can be found here: BPM Monitoring Over Internet.

3.2.1 Block Diagram of the Project

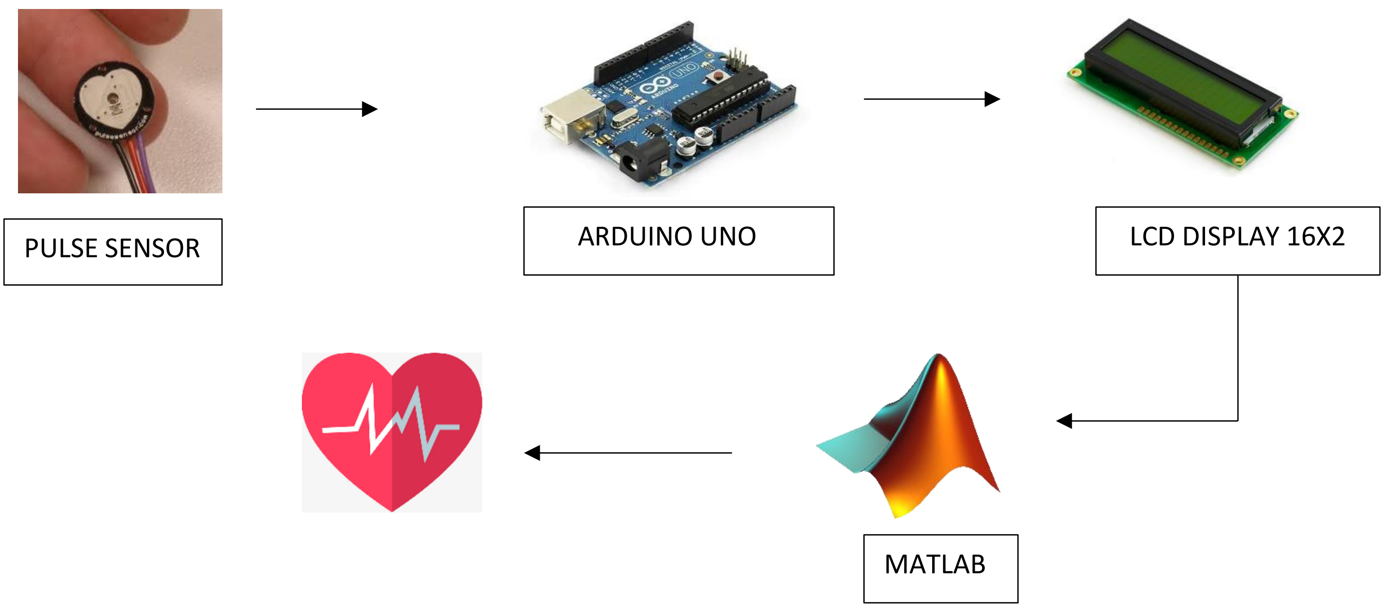


FIGURE 3.1 BLOCK DIAGARAM OF PROJECT

Figure 3.1 shows a block diagram of a Heartbeat Monitor. The block diagram shows how the process of checking our pulse. First, that the pulse sensor will detect our pulse from the source to provide a signal for us to check. The Arduino UNO turns on and executes programmed for functions barrier. After that it will enter the LCD Display to see our heart rate measurement signal. After that open the pc/laptop, so that we can open the MATLAB apps that will show the results of our heartbeat signal form system has been programmed in Arduino Software.

3.2.2 Flowchart of the Project

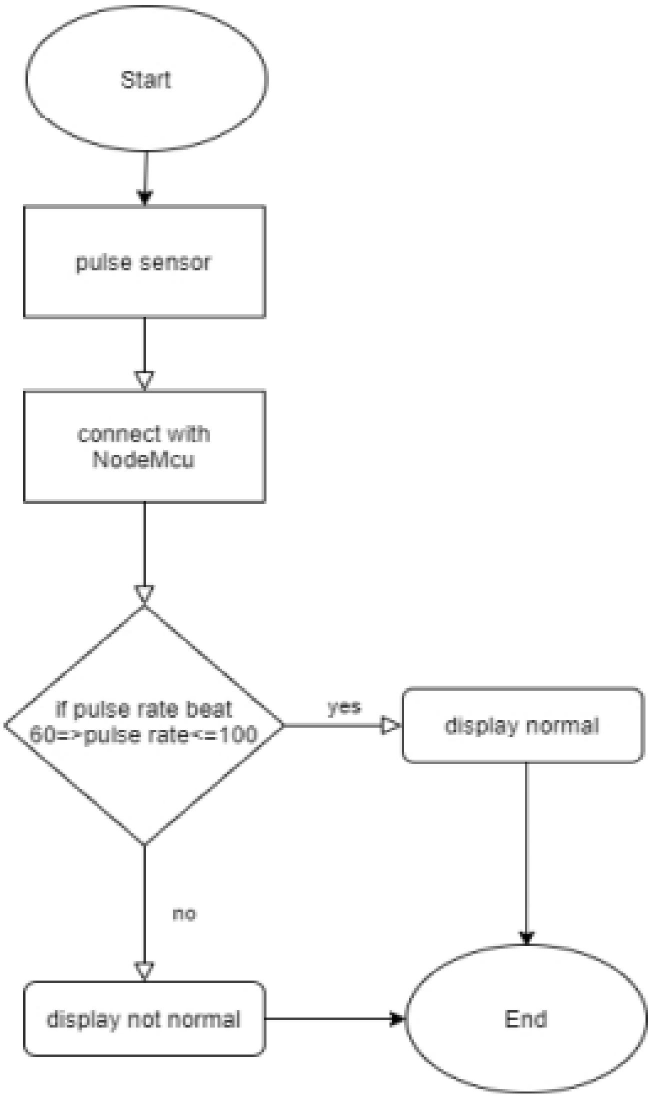


FIGURE 3.2 FLOWCHART OF THE PROJECT

Figure 3.2 shows the flow of the Heartbeat Monitoring System. For this device, the user will wear this heart rate monitor while doing their work or exercising and users can do their work daily life as usual. So, after that, the pulse sensor will detect if the user is sitting in weak condition or good condition. The LCD Display will show a signal to the user if the pulse rate is normal or abnormal.

3.2.3 Project Description

The pulse sensor helps in monitoring the heart rate depending on light intensity. In the serial plotter, the signal will remain approximately at the midpoint of ADC range which is 508 and it is equal nearly to 2.5 V. The Arduino will plot the output signal according to the threshold variable. It stimulates the Arduino to detect a valid signal. Its value can be any number between 0-1024. Moreover, decreasing the threshold increases the sensitivity of the sensor and vice versa. When the analogue signal exceeds the threshold value =509, the Arduino consider it as a valid pulse. Then the signal drops below the threshold and Arduino prepares to find new pulse till a wave formed as shown in Figure (4). Meanwhile, the volume across the capillaries during the heartbeat affects the reflection of light. The more the light is being reflected due to a surge of blood, the more the value goes up in the serial plotter. The signal that displays on serial plotter still has some noise. Thus, signal processing is used by MATLAB to eliminate the undesired small fluctuation. The function smooth data is used with specific smoothing method which is (' gaussian ') filter. Gaussian filter smooths the data which is varies rapidly. LCD displays beats per minute (BPM) by using the constant rate change. The microcontroller does not wait for a whole minute to count the beats per minute (BPM). Instead, it detects the pulse each 2ms and predicts the whole rhythm to calculate BPM swiftly as shown in Figure (6). BPM=88 considers as a normal reading for an adult has no health problems. The output signal has been analysed to the normal components of any heart signal (QRS wave and P-wave). The signal may seem a little bit different than the ECG signal. This is due to the pulse sensor precision and sensitivity.

3.3 Project Hardware

The project hardware to develop this project is Arduino Uno, pulse sensor, 10k potentiometer and 10Kohm resistor. All the hardware mentioned is essential hardware to develop this device.

3.3.1 Schematic Circuit

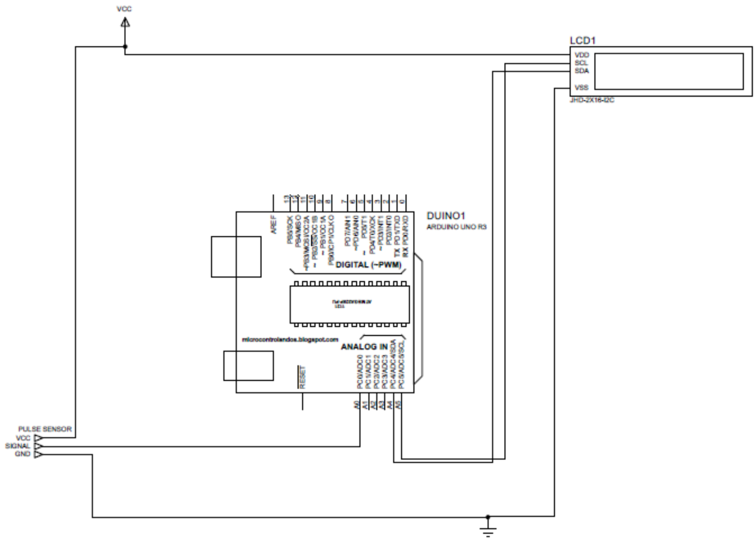
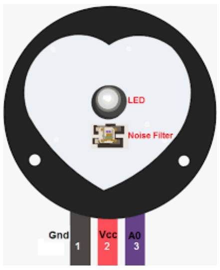


FIGURE 3.3: SCHEMATIC CIRCUIT OF PROJECT

Figure 3.3 shows the operation of a pulse sensor circuit using Arduino UNO. Current is sent to a variable resistor and then sent to the Arduino Uno to run the programmed program. From the Arduino Uno it will program the input by the sensor pulse to measure heart pressure. After that, the result of stress will be displayed on the LCD referring to pc/laptop.

3.3.2 Description of Main Component

3.3.2.1 Component 1



- ✓ **PULSE SENSOR**

  It is an Open-Source heart rate monitor which considered as a PPG device used to monitor the non-invasive heart rate. It measures the real time heart beats and calculates BPM with the aid of algorithms implemented by Arduino. The normal operating voltage is +5V or +3.3V and current consumption of 4mA. The sensor has two sides, one side consists of an LED with an ambient light sensor and the other side contains circuitry which amplifies the signals and filters the noise.

3.3.2.2 Component 2

- ✓ **Arduino uno**

  Arduino UNO is micro-controller based on Atmega328, having 14 digitals in/Out pins of which 6 are for PWM output, 6 are for analogue input. Operates at 16 MHz, with a USB, Power jack, Reset button.

3.3.2.3 Component 3

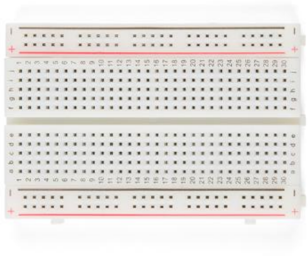
- ✓ **16 x 2 LCD Display**

  A liquid-crystal display (LCD) is a flat-panel display or other electronically modulated optical device that uses the light-modulating properties of liquid crystals combined with polarizers. Liquid crystals do not emit light directly,[1] instead using a backlight or reflector to produce images in colour or monochrome. [2] LCDs are available to display arbitrary images (as in a general-purpose computer display) or fixed images with low information content, which can be displayed or hidden, such as present words, digits, and seven segment displays, as in a digital clock. They use the same basic technology, except that arbitrary image is made from a matrix of small pixels, while other displays have larger elements. LCDs can either be normally on (positive) or off (negative), depending on the polarizer arrangement. For example, a character positive LCD with a backlight will have black lettering on a background that is the colour of the backlight, and a character negative LCD will have a black background with the letters being of the same colour as the backlight. Optical filters are added to white on blue LCDs to give them their characteristic appearance.

3.3.2.5 Component 4

- ✓ **USB cable**

  USB cable is used to connect computer to the Arduino UNO board.



3.3.2.6 Component 5

- ✓ **BREADBOARD**

  A breadboard is a solder less device for temporary prototype with electronics and test circuit designs. The breadboard has strips of the metal underneath the board and connect the holes on the top of the board.

3.3.2.7 Component 6

- ✓ **JUMPER WIRES**

  Jumper wires are simply wires that have connector pins at each end, allowing them to be used to connect two points to each other without soldering. Jumper wires are typically used with breadboards and other prototyping tools to make it easy to change a circuit as needed.

3.3.3 Circuit Operation

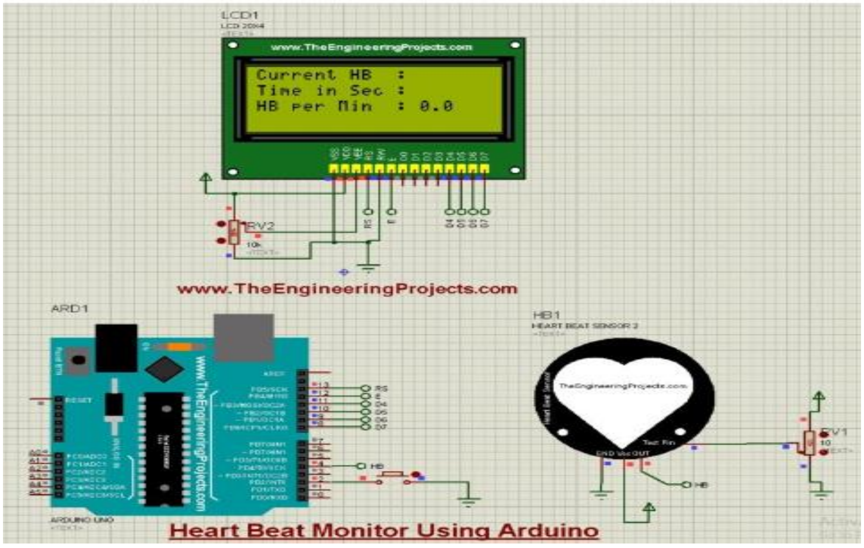


FIGURE 3.4: CIRCUIT OPERATION OF PROJECT

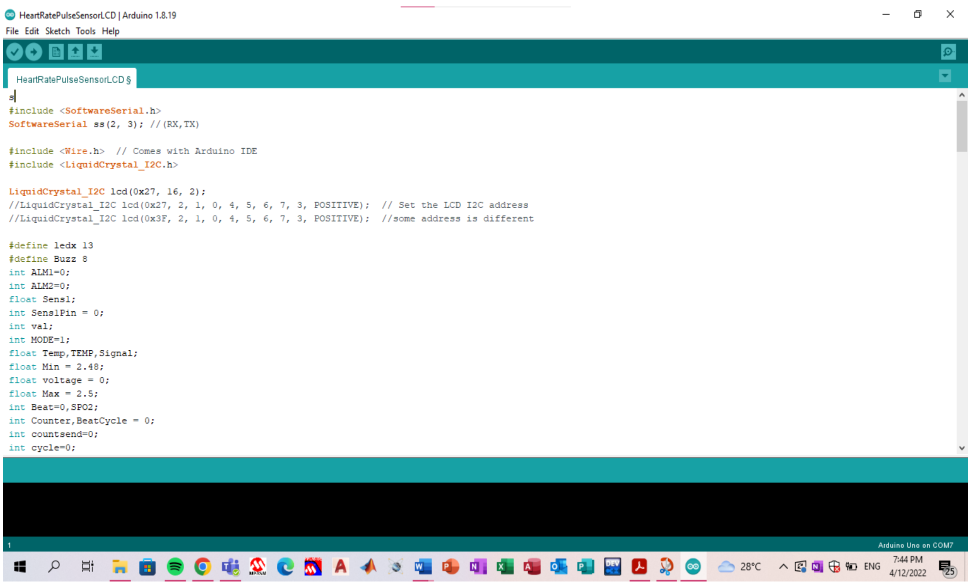


FIGURE 3.5: Coding Arduino Uno. This coding put in Arduino Uno to program the Heartrate monitoring to schedule time for the patient. It also for LCD to display the message.

3.3.4 PROJECT SOFTWARE

3.3.4. PROTEUS 8 PROFESSIONAL SOFTWARE

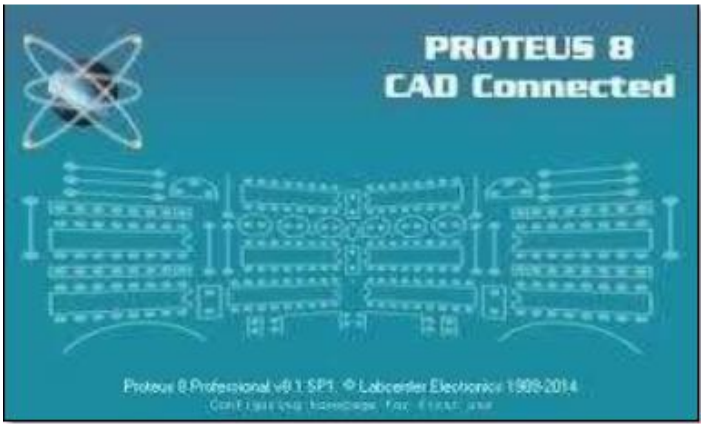


FIGURE 3.6: PROTEUS 8 PROFESSIONAL SOFTWARE

3.3.5 ARDUINO SOFTWARE

FIGURE 3.7: ARDUINO UNO SOFTWARE

3.3.6 PROTEUS SOFTWARE

FIGURE 3.8: TINKERCAD

3.3.7 PROTOTYPE DELOPMENT

Based on what I have studied about the color, shape and position of each component for the development of this prototype, I thought of sketching the LCD design so that the user can see the pulse rate is in a good state when detected to see the light on. While the pulse sensor can be placed on the tip of the finger in a calm state so that the LCD can be detected accurately. This prototype will be built in a medium size, neither large nor small so that it is easy to move and easy to install.

3.3.8 Mechanical Design/Product Layout

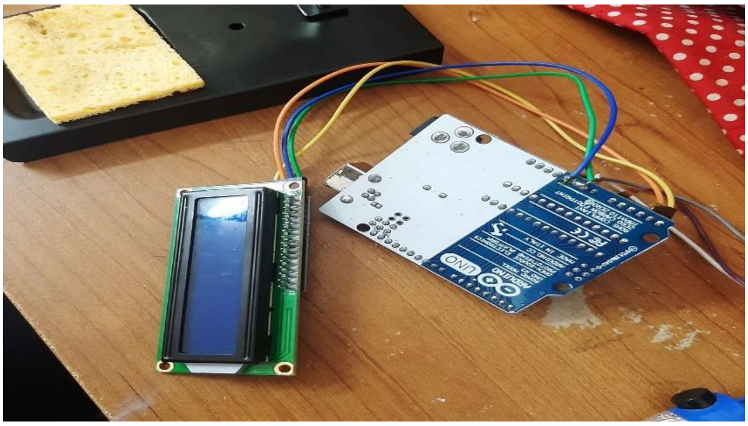


FIGURE 3.9: MECHANICAL DESIGN/PRODUCT LAYOUT

3.4 SUMMARY

My project is a heart rate monitoring system to know our pulse is normal". By using this device, patients can simply check their own pulse without the help of a doctor or any clinic in rehabilitation at home. By using a 16x2 LCD Display, we were able to improve the recovery process that could measure the rotational motion of the pulse rate which could provide more accurate analysis such as patient performance and limits on any recovery exercise. Once the project is completed, the result obtained is that all the objectives of the project have been achieved. With this project, patients with heart problems or the like no longer have trouble measuring the level of their affected pulse movement on their own without the help of others. I hope patients with heart problems are happy and welcome this project.

CHAPTER 4
RESULT AND DISCUSSION

4.1 INTRODUCTION

For this project, when I wanted to connect the hardware with the Arduino Uno, I faced some difficulties because the pulse sensor could not detect well, and the inappropriate position made it difficult for the LCD to record accurate results and made the process of going through this project difficult. After some solid coding improvements, the project was finally able to interface with USB in an Arduino Uno. Results can also be made, and I have gotten good output showing the project is working well.

4.2 RESULT AND ANALYSIS

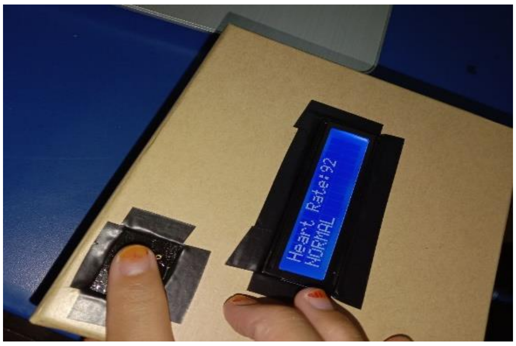
<table>
<tr><td>I.</td><td>
</td><td>If the pulse rate beat higher than 100 bpm it will display "NOR NORMAL RATE".</td></tr>
<tr><td>II.</td><td>
</td><td>If the pulse rate beat between 60 bpm to 100 bpm, it will display "NORMALRATE".</td></tr>
<tr><td>III.</td><td>
</td><td>If the pulse rate beat lower than 60 bpm it will display "NOR NORMAL RATE".</td></tr>
</table>

TABLE 4.1: SHOW THE RESULT AND ANALYSIS OF THE PROJECT

4.3 DISCUSSION

I managed to get a useful output because I was able to analyse the pulse rate level and the detected graph according to the detected pulse. In addition, this LCD is also useful for users because of its easy-to-read and easy-to-understand interface. With the help of this project, heart rate problems can be avoided and can give awareness about the importance of safety to users to be more careful even if they are healthy or not.

4.4 CHAPTER SUMMARY

In this chapter describes how the project works and production process. Among them are the arrangement of coding (programming), production of project design, program management of the project system produced as well as results and analysis from the community. Various ideas and challenges arose in the production of this project, and I managed to produce this project.

CHAPTER 5

5.1 INTRODUCTION

'HEART RATE MONITORING' is a tool to detect human heart rate. This detector function is usually used either on the neck, arm or fingertip only to calculate the pulse rate. However, as a society, we should have a tool to check the pulse rate at home as a detection tool to avoid unwanted things. At least there is one because the safety of household members is very important. Therefore, I chose to produce this project to help the community to ensure the health of household members.

5.2 CONCLUSION

Here I can summarize the entire report can be implemented successfully. All the chapters contained in this report can overcome the problems encountered and can be solved according to the developments required by the current HOSPITAL to achieve the development of industry 4.0. It is able to have a positive impact on the surrounding community because it can facilitate routine work that was previously done by humans. Next, "HEART RATE MONITORING" will be able to be expanded through the Ministry of Health to make it easier for people to do their daily routines either indoors or outdoors.

5.3 SUGGESTION FOR FUTURE WORK

The problem faced is to execute the program because the program that is run is problematic during compilation due to library duplication. In addition, the breadboard is damaged due to incorrect circuit connections. The solution, use a power bank at a total current of 1.3A and buy a new breadboard. In addition, the stability of the circuit connection is determined on the correct one. The solution, the detector pulse sensor will be installed correctly according to the circuit channel of the detector and determine the set number of scales.

5.4 CHAPTER SUMMARY

Most cases of death occur due to difficulty breathing or an unstable pulse. Even at a young age it is not necessary not to be affected by heart disease or respiratory problems, but if such things happen in the house, this can lead to facing a very high risk quickly. Therefore, I designed a system that can give early warning to users to prevent any accidents from happening. Prevention is better than cure.

CHAPTER 6

6.1 INTRODUCTION

This project involves the cost of purchasing components and materials for it implementation involving the cost of Arduino Uno, pulse sensor, LCD 16x2, and other equipment. All the components are purchased through the online purchase method to facilitate and save costs. The financial resources for this project are self-financed with some basic components and materials obtained in the project laboratory. Based on the projected cost, it is estimated at RM150 and below. It is flexible and can be achieved based on the investigation carried out.

6.2 GANT CHART AND ACTIVITIES OF THE PROJECT

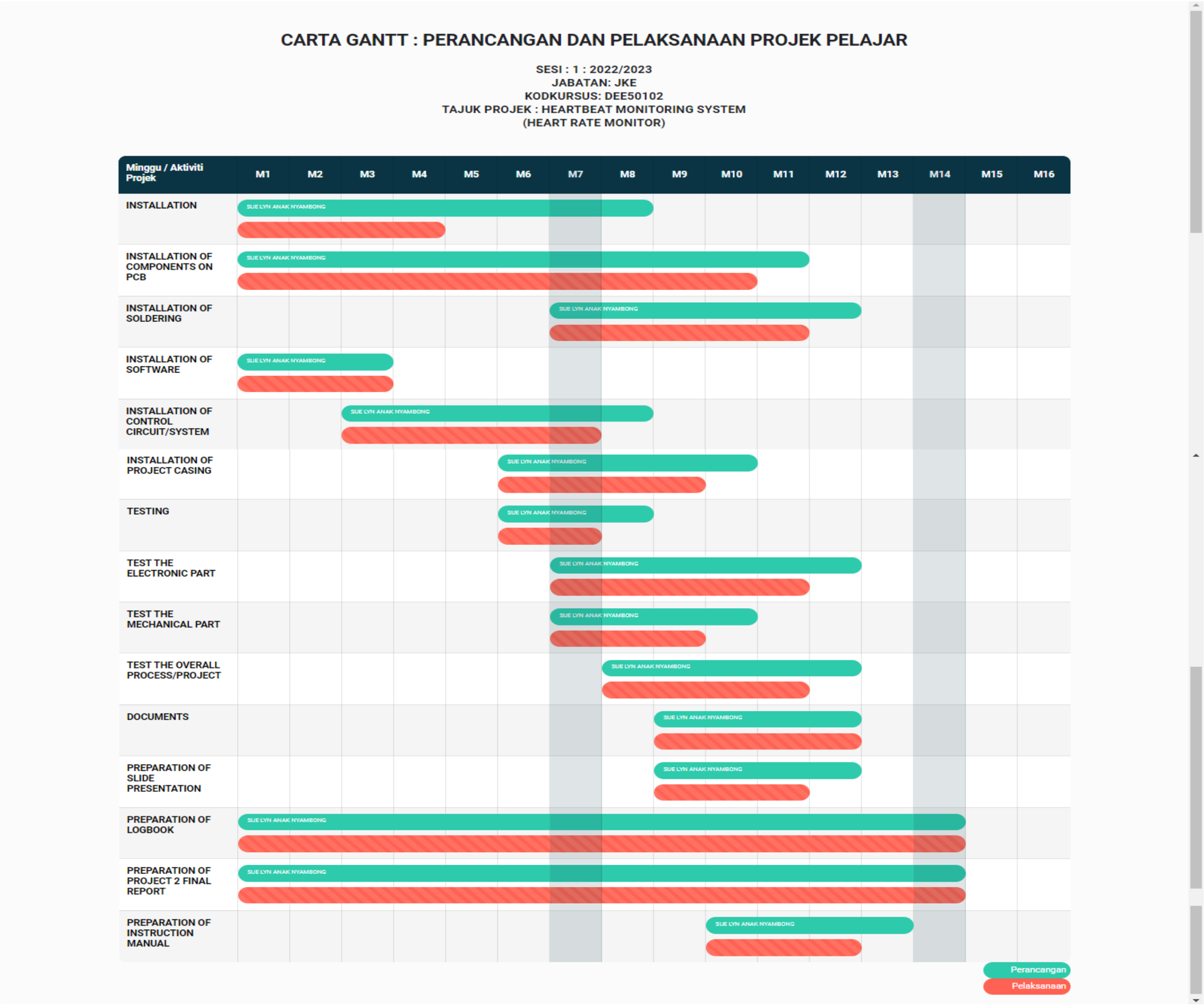


TABLE 6.1: SHOW THE GABT CHART OF THE PROJECT

6.3 MILESTONE

6.4 COST AND BUDGETING

No.	Component and materials	The unit price	Quantity	Total
1	ARDUINO UNO R3	RM 45.90	1	RM45.90
2	PULSE SENSOR	RM 17.50	1	RM 17.50
3	LCD DISPLAY 16X2	RM 16.99	1	RM 16.99
4	JUMPER WIRE	RM 4.60	4	RM 18.40
5	BREADBOARD	RM 3.90	1	RM 3.90
			TOTAL:	RM102.69

TABLE 6.2: SHOW THE PRICE OF THE PROJECT COMPONENT

6.5 CHAPTER SUMMARY

This chapter describes the costs associated with producing the HEARTRATE MONITORING project. I shop online for high quality items at affordable prices. The estimated cost is less than RM 150. Next, estimate the time to complete the project paper and step by step to complete the project.

REFERENCES:

[1]. A microcontroller based automatic heart rate counting system from fingertip Mamun AL, Ahmed N, AL Qahtani (JATIT) Journal OF Theory and Applied technology ISSN 1992-8645.

[2] Heartbeat and Temperature Monitoring System for remote patients using Arduino Vikram Singh, R. Parihar, Akash Y Tungurahua Gamora (IJAERS), International Journal of Advanced Engineering and Science eissn2349-6495.

[3] A GSM Enabled Real time simulated Heart Rate Monitoring and control system Suchinda F, Anna Rao S.J, (IJRET) International Journal of Research in Engineering And Technology, e ISSN 2319-3163.

[4] Heartbeat Sensing and Heart Attack Detection Using internet of things: IOT Aboobacker spideresque, Ardith Kumar, K. Sathish, (IJESCE) International Journal Of Engineering Science and Computing, April 2007.

[5] A Heartbeat and Temperature Measuring System for Remote Health

[6] Monitoring using Wireless Body Area Network Mohammad Wajih Alam, Tanin Sultana and Mohammad Sami Alam International Journal of Bio Science and Bio-Technology Vol.8, No.1 (2016).

[7] Heartbeat Monitoring Alert via SMS 2009 IEEE Symposium on Industrial Electronics and Applications October 4-6, 2009, Kuala Lumpur, Malaysia. Warsuzarina Mat Jubadi, Siti Faridatul Aisyah.

[8] Heartbeat Monitoring and Alert System Using GSM TechnologyBY R. KIssac D. RS SV. V. International Journal of Scientific Research in Computer Science, Engineering, and Information Technology (2019) 80-87

[9] Heartbeat and Temperature Monitoring System for Remote Patients using Arduino BY Parihar VTonge AGanorkar P. International Journal of Advanced Engineering Research and Science (2017) 4(5) 55-58.

[10] Heartbeat Monitoring and Alert System Using Gsm Technology
BY Ufoaroh SOranugo CUchechukwu M. International Journal of Engineering Research and General Science (2015) 3(4) 26-34.

[11] Heartbeat Sensor System for Remote Health Monitoring
BY Ahamed SEmu MSaleh W. International Journal of Computer Applications (2021) 174(15) 27-31

[12] Heartbeat rate monitoring system by pulse technique using HB sensor. BY Arora JGagandeepSingh A..International Conference on Information Communication and Embedded Systems, ICICES 2014 (2015)

APPENDICES

Appendix 1:

PROJECT MANUAL/PRODUCT CATALOGUE

POLITEKNIK MALAYSIA
SULTAN SALAHUDDIN ABDUL AZIZ SHAH

HEARTBEAT MONITORING SYSTEM
(HEART RATE MONITOR)

- SUE LYN ANAK NYAMBONG
- 08DEU20F1040
- PUAN NOR KHARUL AINA BINTI MAT DIN

Instruction Manual.

1. When USB is connected to Arduino Uno it shows welcome message as"WELCOME" and follow "HEART RATE MONITORING".

2. Once switched on, the user must wait 1 minute for the user to start measuring the pulse.

3. After the sentence "HEART RATE:0" comes out, the user can place the fingertip for 2 minutes or 1 minute more.

4. After 2 minutes or 1 minute more the pulse results will come out.

5. The reading will tell the user whether it is "LOW","NORMAL" or HIGH.

6. Once the reading is released, the user can lift the fingertip again.

7. When the user wants to measure the pulse, he should wait for 1 minute for it to return to 0.

Safety Precautions

1. The user should sit quietly so that the reading can be read clearly.

2. The user should not be in a state of stress, should be in a calm state so that the pulse is not too high

3. The connection will be easily broken if the casing falls.

5. The pulse sensor wire will be disconnected if the user presses the pulse sensor strongly.

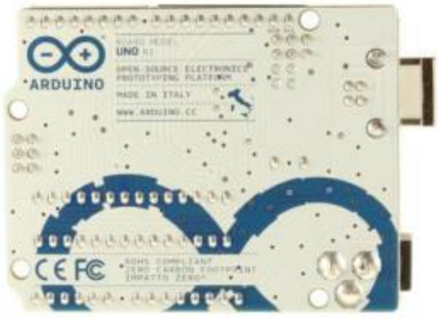
6. Installation of the wire cover so that it does not come into contact with the machine.

Appendix 2:

Datasheet of ARDUINO UNO

Arduino Uno

Arduino Uno R3 Front

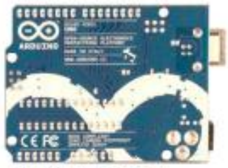


Arduino Uno R3 Back

Arduino Uno R2 Front

Arduino Uno SMD

Arduino Uno Front



Arduino Uno Back

Overview

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.
The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.
Revision 2 of the Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode.
Revision 3 of the board has the following new features:

- 1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the Arduino Due that operate with 3.3V. The second one is a not connected pin, that is reserved for future purposes.
- Stronger RESET circuit.
- Atmega 16U2 replace the 8U2.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the index of Arduino boards.

Summary

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

Schematic & Reference Design

EAGLE files: arduino-uno-Rev3-reference-design.zip (NOTE: works with Eagle 6.0 and newer)
Schematic: arduino-uno-Rev3-schematic.pdf
Note: The Arduino reference design can use an Atmega8, 168, or 328, Current models use an ATmega328, but an Atmega8 is shown in the schematic for reference. The pin configuration is identical on all three processors.

Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.
External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.
The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.
The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.**This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the analogWrite() function.

- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication using the SPI library.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function. Additionally, some pins have specialized functionality:

- **TWI: A4 or SDA pin and A5 or SCL pin.** Support TWI communication using the Wire library.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with analogReference().
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the mapping between Arduino pins and ATmega328 ports. The mapping for the Atmega8, 168, and 328 is identical.

Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).
A SoftwareSerial library allows for serial communication on any of the Uno's digital pins.
The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details. For SPI communication, use the SPI library.

Programming

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

- On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.
- On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

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

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.
This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.
The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread for details.

USB Overcurrent Protection

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Four screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

Data Sheet of PULSE SENSOR

WORLD FAMOUS ELECTRONICS llc.

www.pulsesensor.com

PULSE SENSOR
EASY TO USE HEART RATE SENSOR & KIT

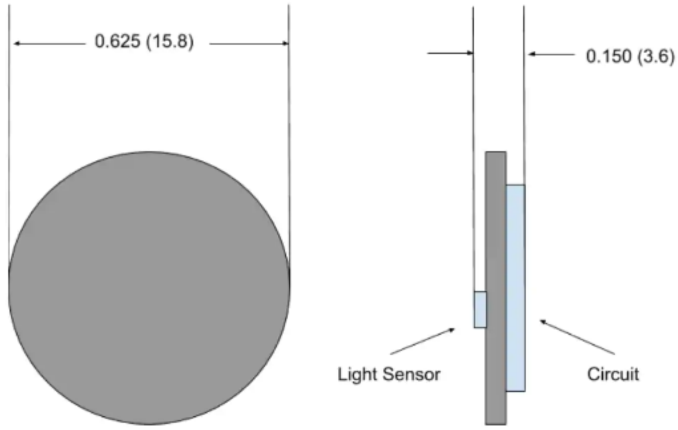
General Description	Features
The Pulse Sensor is the original low-cost optical heart rate sensor (PPG) for Arduino and other microcontrollers. It's designed and made by World Famous Electronics, who actively maintain extensive example projects and code at: www.pulsesensor.com	• Includes Kit accessories for high-quality sensor readings • Designed for Plug and Play • Small size and embeddable into wearables • Works with any MCU with an ADC • Works with 3 Volts or 5 Volts • Well-documented Arduino library

Absolute Maximum Ratings	Min	Typ	Max	Unit
Operating Temperature Range	-40		+85	°C
Input Voltage Range	3		5.5	V
Output Voltage Range	0.3	Vdd/2	Vdd	V
Supply Current	3		4	mA

Pulse Sensor Kit Contents

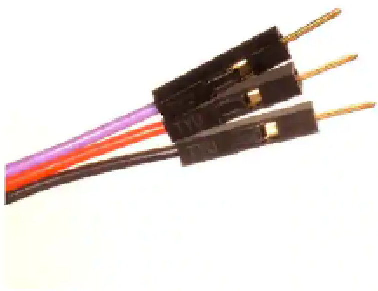
Pulse Sensor Optical Heart Rate Monitor

Physical Dimensions PCB inch(mm)



Cable Specs

- Length 610 mm (24 inches)
- 26 Gauge
- PVC Insulation, Ribbon Style
- Male Header Termination
  - Black Wire = GND
  - Red Wire = Vdd
  - Purple Wire = Pulse Signal

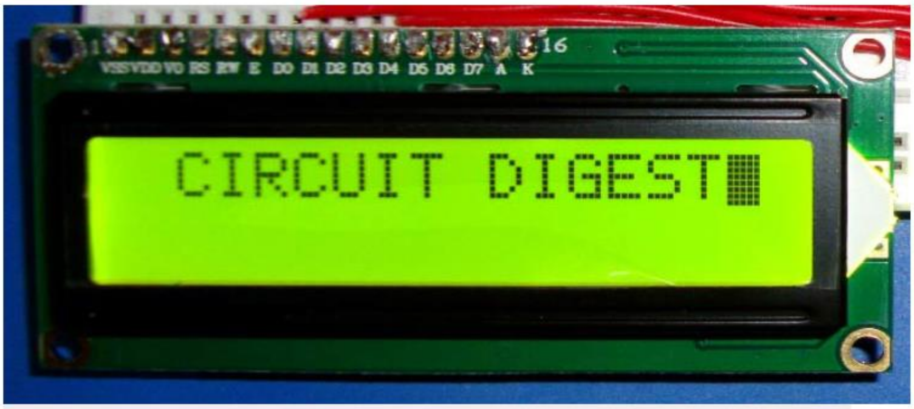


Data Sheet of LCD DISPLAY 16X2

· **16x2 LCD Display Module**

Published October 24, 2015 7

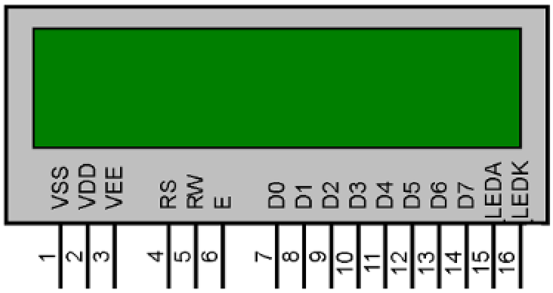
Aswinth Raj
Author



16x2 LCD Display Module with HD44780 Controller

16×2 LCD is named so because; it has 16 Columns and 2 Rows. There are a lot of combinations available like, 8×1, 8×2, 10×2, 16×1, etc. But the most used one is the 16*2 LCD, hence we are using it here.

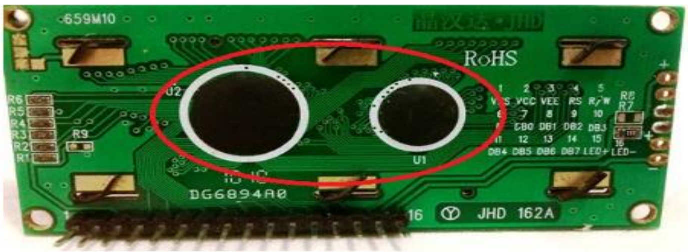
All the above mentioned LCD display will have 16 Pins and the programming approach is also the same and hence the choice is left to you. Below is the **Pinout and Pin Description of 16x2 LCD Module**:



Sr. No	Pin No.	Pin Name	Pin Type	Pin Description	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.	<u>In 4-Wire Mode</u> Only 4 pins (0-3) is connected to MCU <u>In 8-Wire Mode</u> 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

It is okay if you do not understand the function of all the pins, I will be explaining in detail below. Now, let us turn back our LCD:



Okay, what is this two black circle like things on the back of our LCD?

These black circles consist of an interface IC and its associated components to help us use this LCD with the MCU. Because our LCD is a 16*2 Dot matrix LCD and so it will have (16*2=32) 32 characters in total and each character will be made of 5*8 Pixel Dots. A Single character with all its Pixels enabled is shown in the below picture.

So Now, we know that each character has (5*8=40) 40 Pixels and for 32 Characters we will have (32*40) 1280 Pixels. Further, the LCD should also be instructed about the Position of the Pixels.

It will be a hectic task to handle everything with the help of MCU, hence an **Interface IC like HD44780** is used, which is mounted on LCD Module itself. The function of this IC is to get the **Commands and Data** from the MCU and process them to display meaningful information onto our LCD Screen.

Let's discuss the different type of mode and options available in our LCD that has to be controlled by our Control Pins.

4-bit and 8-bit Mode of LCD:

The LCD can work in two different modes, namely the 4-bit mode and the 8-bit mode. In **4 bit mode** we send the data nibble by nibble, first upper nibble and then lower nibble. For those of you who don't know what a nibble is: a nibble is a group of four bits, so the lower four bits (D0-D3) of a byte form the lower nibble while the upper four bits (D4-D7) of a byte form the higher nibble. This enables us to send 8 bit data.

Whereas **in 8 bit mode** we can send the 8-bit data directly in one stroke since we use all the 8 data lines.

Now you must have guessed it, Yes 8-bit mode is faster and flawless than 4-bit mode. But the major drawback is that it needs 8 data lines connected to the microcontroller. This will make us run out of I/O pins on our MCU, so 4-bit mode is widely used. No control pins are used to set these modes. It's just the way of programming that change.

Read and Write Mode of LCD:

As said, the LCD itself consists of an Interface IC. The MCU can either read or write to this interface IC. Most of the times we will be just writing to the IC, since reading will make it more complex and such scenarios are very rare. Information like position of cursor, status completion interrupts etc. can be read if required, but it is out of the scope of this tutorial.

The Interface IC present in most of the LCD is **HD44780U,** in order to program our LCD we should learn the complete datasheet of the IC. The datasheet is given here.

LCD Commands:

There are some preset commands instructions in LCD, which we need to send to LCD through some microcontroller. Some important command instructions are given below:

Hex Code	**Command to LCD Instruction Register**
0F	LCD ON, cursor ON
01	Clear display screen
02	Return home
04	Decrement cursor (shift cursor to left)
06	Increment cursor (shift cursor to right)
05	Shift display right
07	Shift display left

0E	Display ON, cursor blinking
80	Force cursor to beginning of first line
C0	Force cursor to beginning of second line
38	2 lines and 5×7 matrix
83	Cursor line 1 position 3
3C	Activate second line
08	Display OFF, cursor OFF
C1	Jump to second line, position 1
OC	Display ON, cursor OFF
C1	Jump to second line, position 1
C2	Jump to second line, position 2

Check our LCD interfacing Articles with different Microcontrollers:

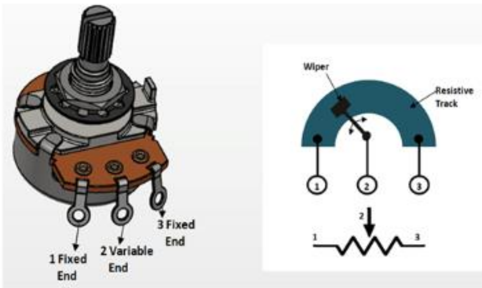
- LCD Interfacing with 8051 Microcontroller
- Interfacing LCD with ATmega32 Microcontroller
- LCD Interfacing with PIC Microcontroller
- Interfacing 16x2 LCD with Arduino
- 16x2 LCD Interfacing with Raspberry Pi using Python

Data Sheet of 10k POTENTIOMETER

Potentiometer

29 September 2017 - 0 Comments

10K Potentiometer



Potentiometer Pin Diagram

Potentiometer Pin Configuration

Pin No.	Pin Name	Description
1	Fixed End	This end is connected to one end of the resistive track
2	Variable End	This end is connected to the wiper, to provide variable voltage
3	Fixed End	This end is connected to another end of the resistive track

Features

- Type: Rotary a.k.a Radio POT
- Available in different resistance values like 500Ω, 1K, 2K, 5K, 10K, 22K, 47K, 50K, 100K, 220K, 470K, 500K, 1 M.
- Power Rating: 0.3W
- Maximum Input Voltage: 200Vdc
- Rotational Life: 2000K cycles

Selecting a Potentiometer

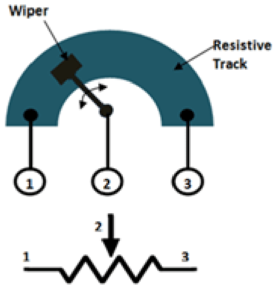
Potentiometers also known as **POT**, are nothing but **variable resistors**. They can provide a variable resistance by simply varying the knob on top of its head. It can be classified based on two main parameters. One is their **Resistance (R-ohms)** itself and the other is its **Power (P-Watts)** rating.

The value or resistance decides how much opposition it provides to the flow of current. The greater the resistor value the smaller the current will flow. Some standard values for a potentiometer are 500Ω, 1K, 2K, 5K, 10K, 22K, 47K, 50K, 100K, 220K, 470K, 500K, 1 M.

Resistors are also classified based on how much current it can allow; this is called Power (wattage) rating. The higher the power rating the bigger the resistor gets and it can also more current. For potentiometers the power rating is 0.3W and hence can be used only for low current circuits.

How to Use a Potentiometer

As far as we know resistors should always have two terminals but, why a **potentiometer** has three terminals and how to we use these terminals. It is very easy to understand the purpose of these terminals by looking at the diagram below.



Applications

- Voltage and Current Control Circuits
- Used as volume control knobs in radios
- Tuning or controlling circuits
- Analog input control knobs

The diagram shows the parts present inside a potentiometer. We have a resistive track whose complete resistance will be equal to the rated resistance value of the POT.

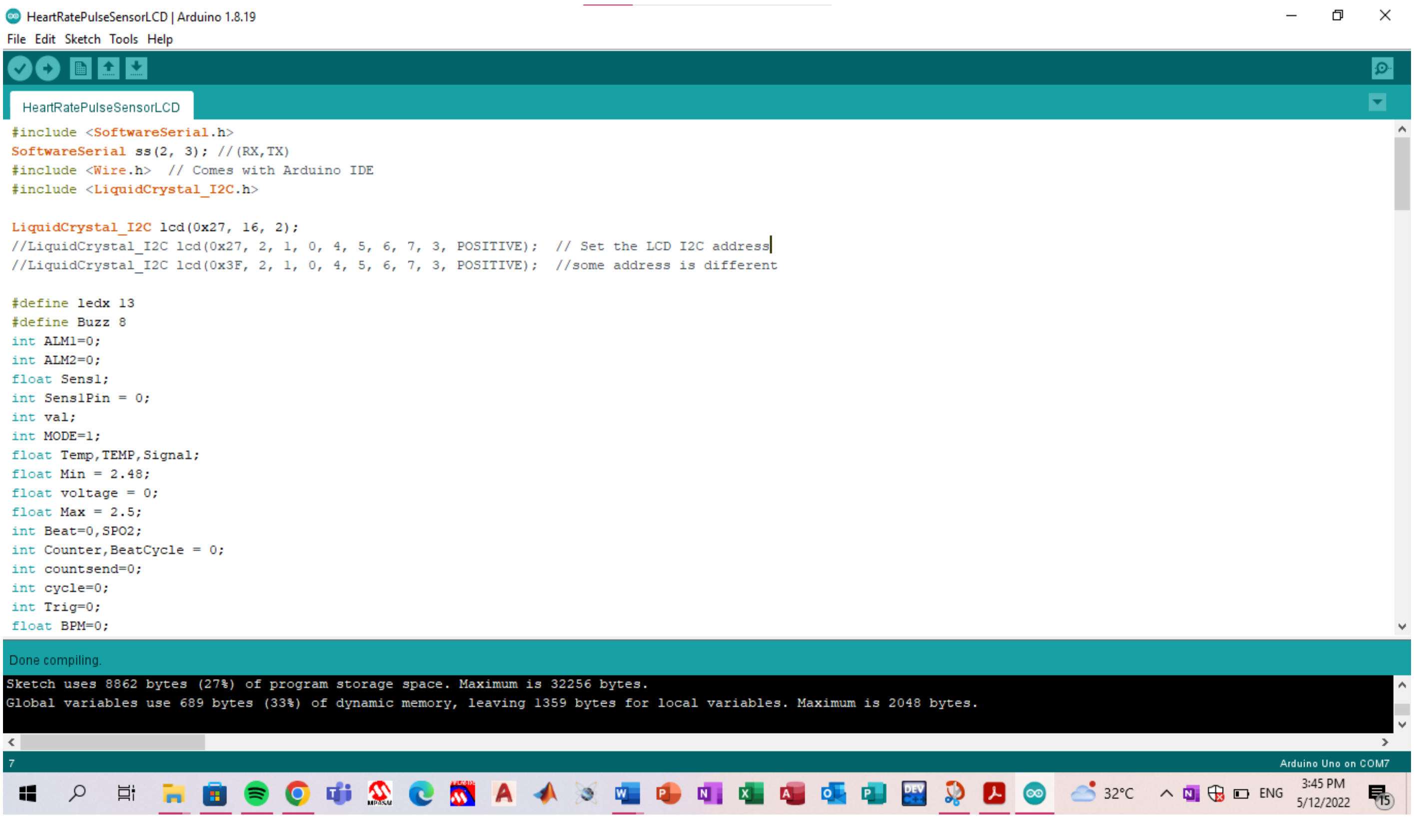
As the symbol suggests a potentiometer is nothing but a resistor with one variable end. Let us assume a 10k potentiometer, here if we measure the resistance between terminal 1 and terminal 3 we will get a value of 10k because both the terminals are fixed ends of the potentiometer. Now, let us place the wiper exactly at 25% from terminal 1 as shown above and if we measure the resistance between 1 and 2 we will get 25% of 10k which is 2.5K and measuring across terminal 2 and 3 will give a resistance of 7.5K.

So the terminals 1 and 2 or terminals 2 and 3 can be used to obtain the variable resistance and the knob can be used to vary the resistance and set the required value.

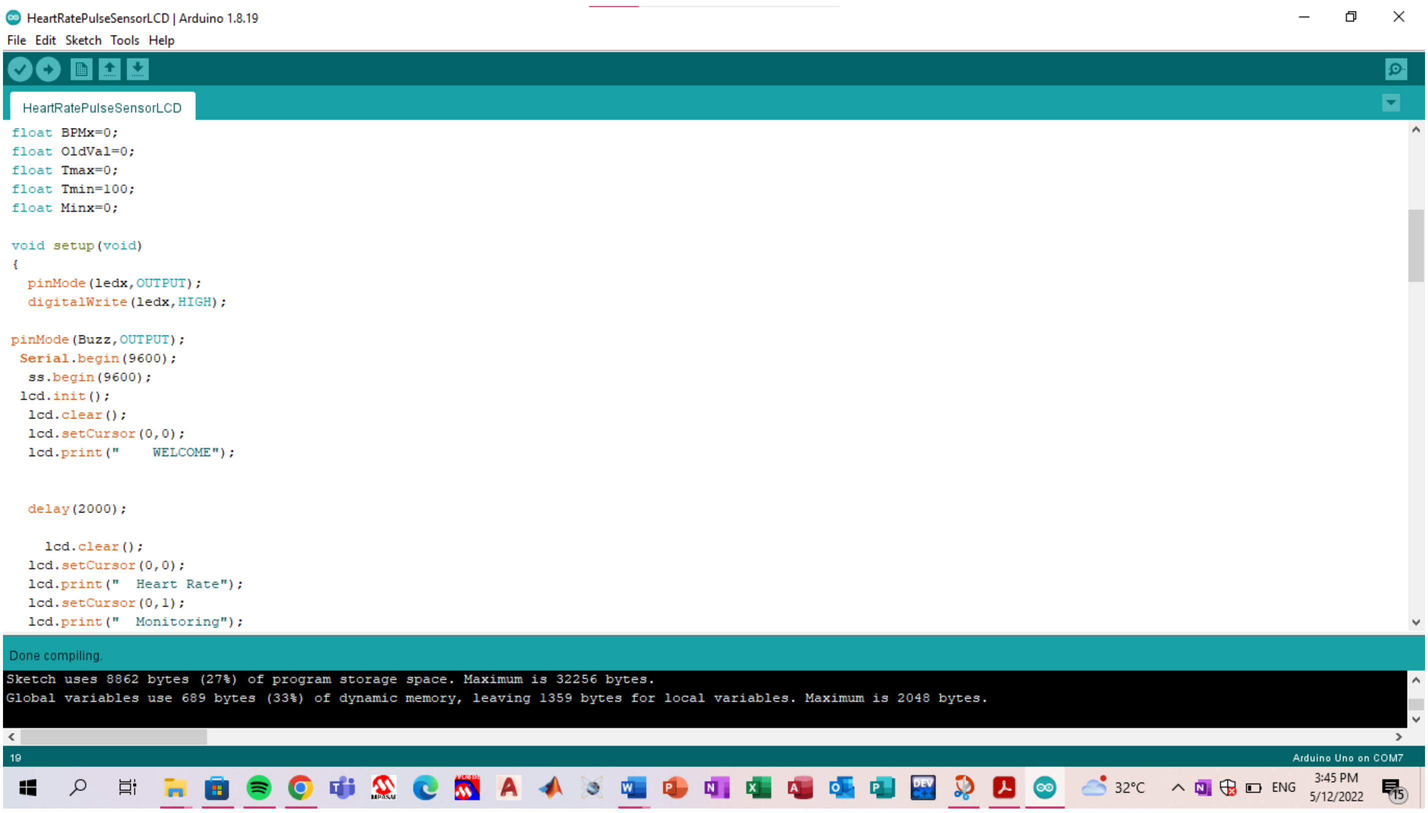
Appendix 3:

PROGRAMMING OF PROJECT

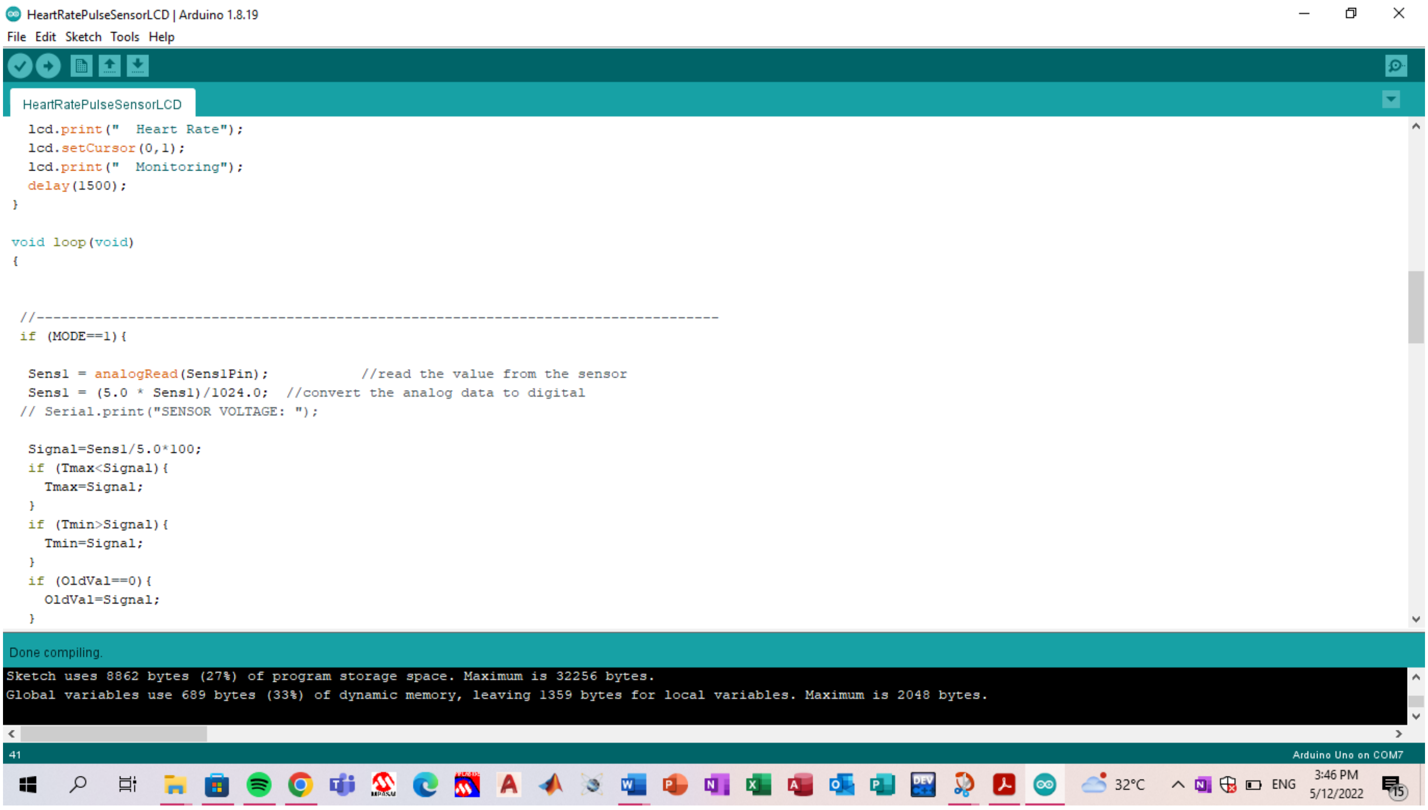
1.



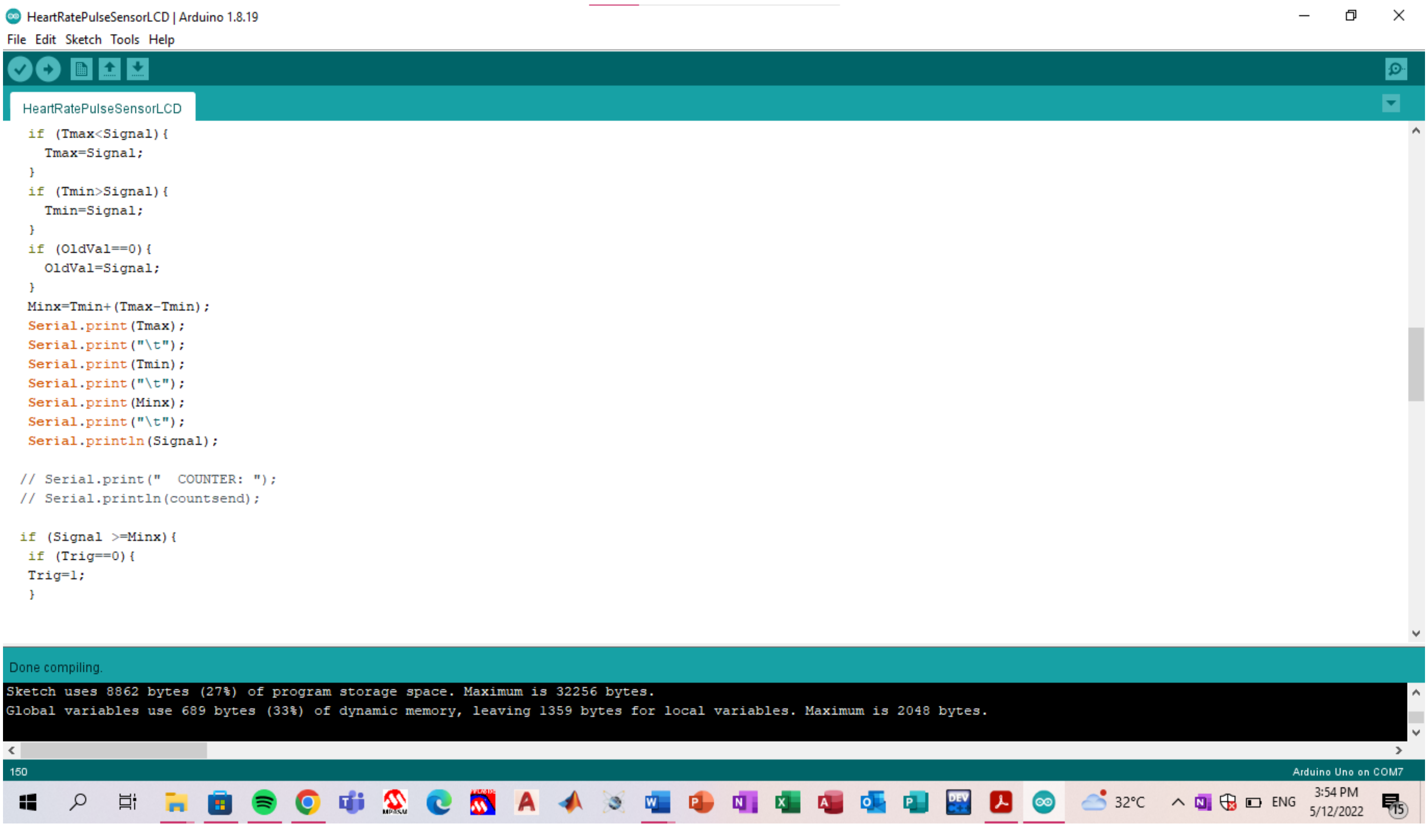
2.



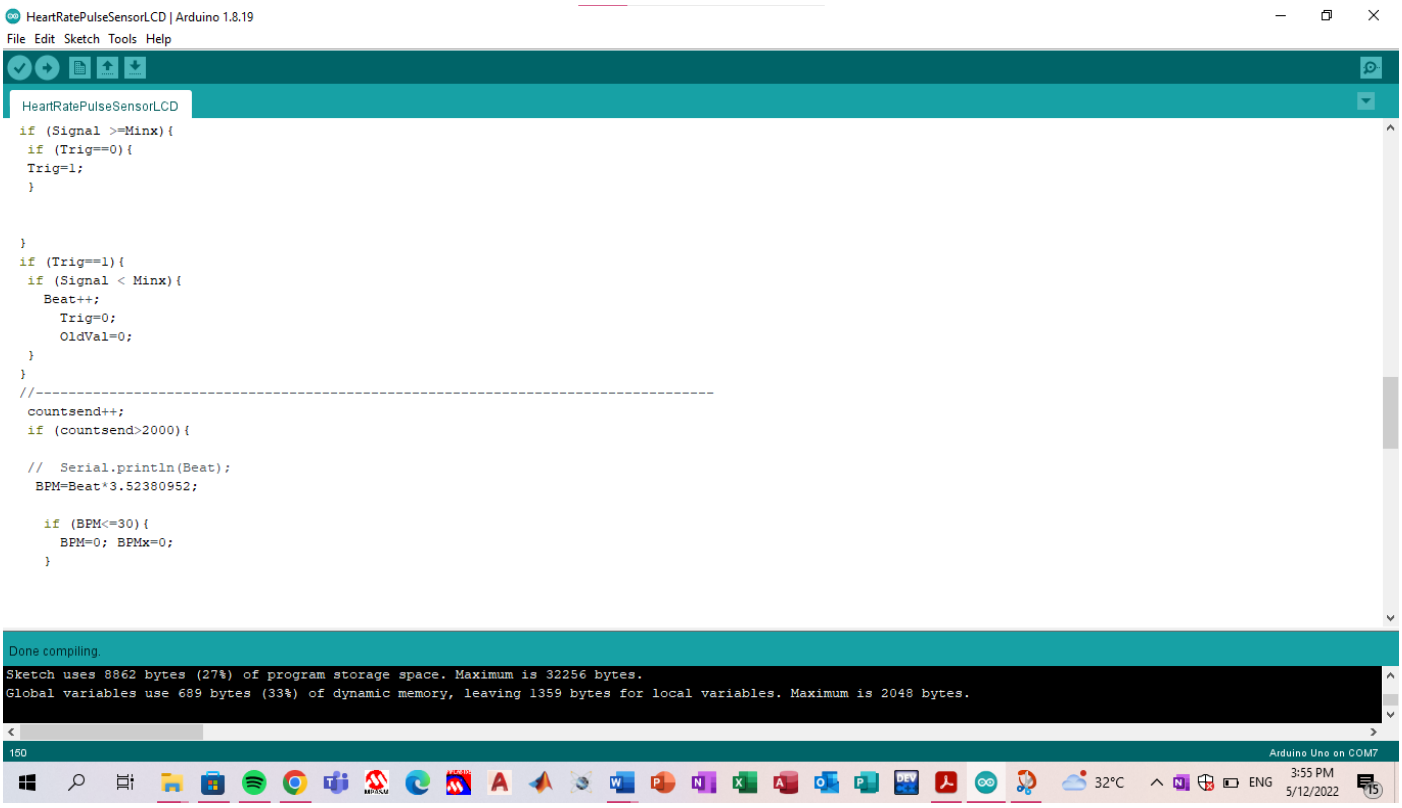
3.


HeartRatePulseSensorLCD | Arduino 1.8.19
File Edit Sketch Tools Help
HeartRatePulseSensorLCD
lcd.print(" Heart Rate");
lcd.setCursor(0,1);
lcd.print(" Monitoring");
delay(1500);
}
void loop(void)
{
//---
if (MODE==1){
Sens1 = analogRead(Sens1Pin);   //read the value from the sensor
Sens1 = (5.0 * Sens1)/1024.0; //convert the analog data to digital
// Serial.print("SENSOR VOLTAGE: ");
Signal=Sens1/5.0*100;
if (Tmax<Signal){
Tmax=Signal;
}
if (Tmin>Signal){
Tmin=Signal;
}
if (OldVal==0){
OldVal=Signal;
}
Done compiling.
Sketch uses 8862 bytes (27%) of program storage space. Maximum is 32256 bytes.
Global variables use 689 bytes (33%) of dynamic memory, leaving 1359 bytes for local variables. Maximum is 2048 bytes.
41
Arduino Uno on COM7
32°C
ENG
3:46 PM
5/12/2022

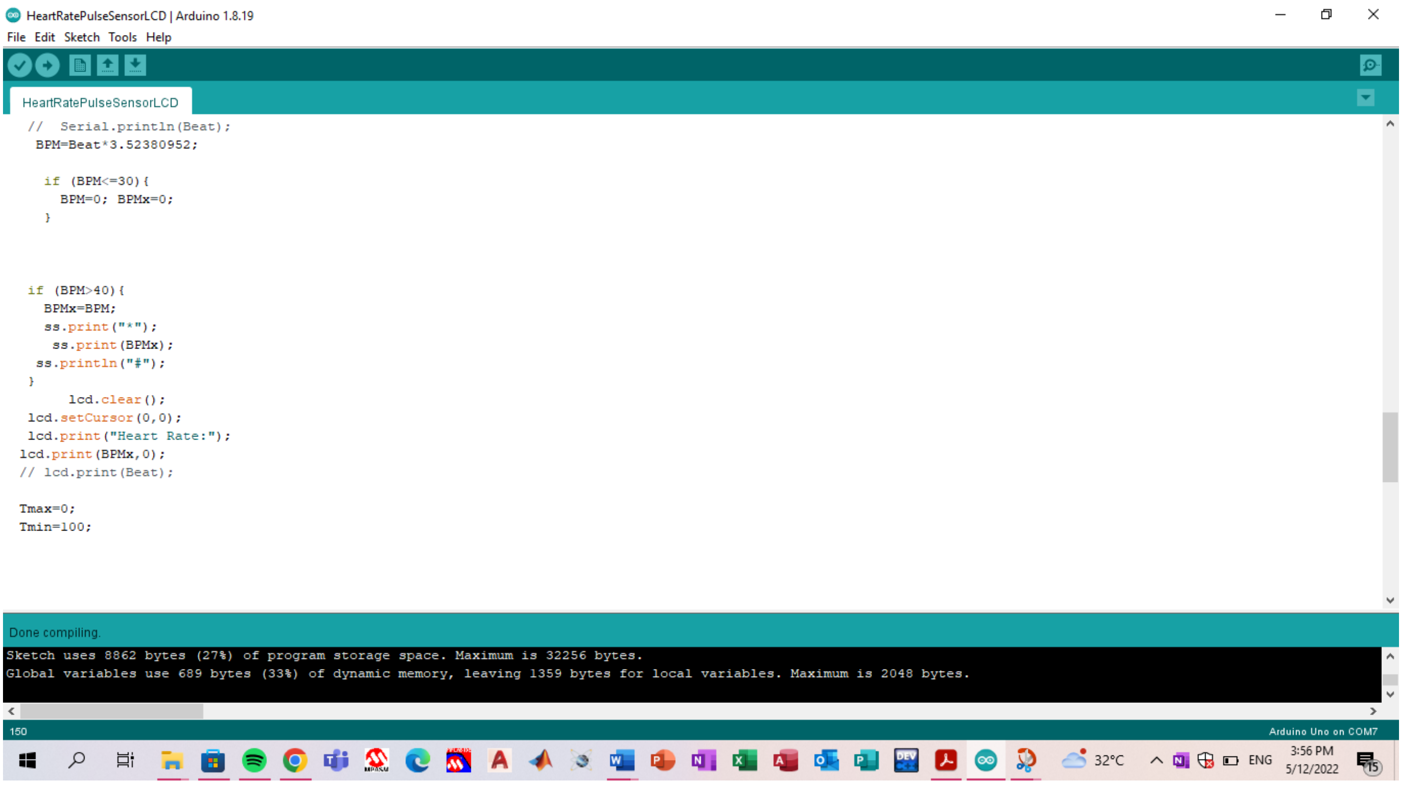
4.


HeartRatePulseSensorLCD | Arduino 1.8.19
File Edit Sketch Tools Help
HeartRatePulseSensorLCD
if (Tmax<Signal){
Tmax=Signal;
}
if (Tmin>Signal){
Tmin=Signal;
}
if (OldVal==0){
OldVal=Signal;
}
Minx=Tmin+(Tmax-Tmin);
Serial.print(Tmax);
Serial.print("\t");
Serial.print(Tmin);
Serial.print("\t");
Serial.print(Minx);
Serial.print("\t");
Serial.println(Signal);
// Serial.print(" COUNTER: ");
// Serial.println(countsend);
if (Signal >=Minx){
if (Trig==0){
Trig=1;
}
Done compiling.
Sketch uses 8862 bytes (27%) of program storage space. Maximum is 32256 bytes.
Global variables use 689 bytes (33%) of dynamic memory, leaving 1359 bytes for local variables. Maximum is 2048 bytes.
150
Arduino Uno on COM7
32°C
ENG
3:54 PM
5/12/2022

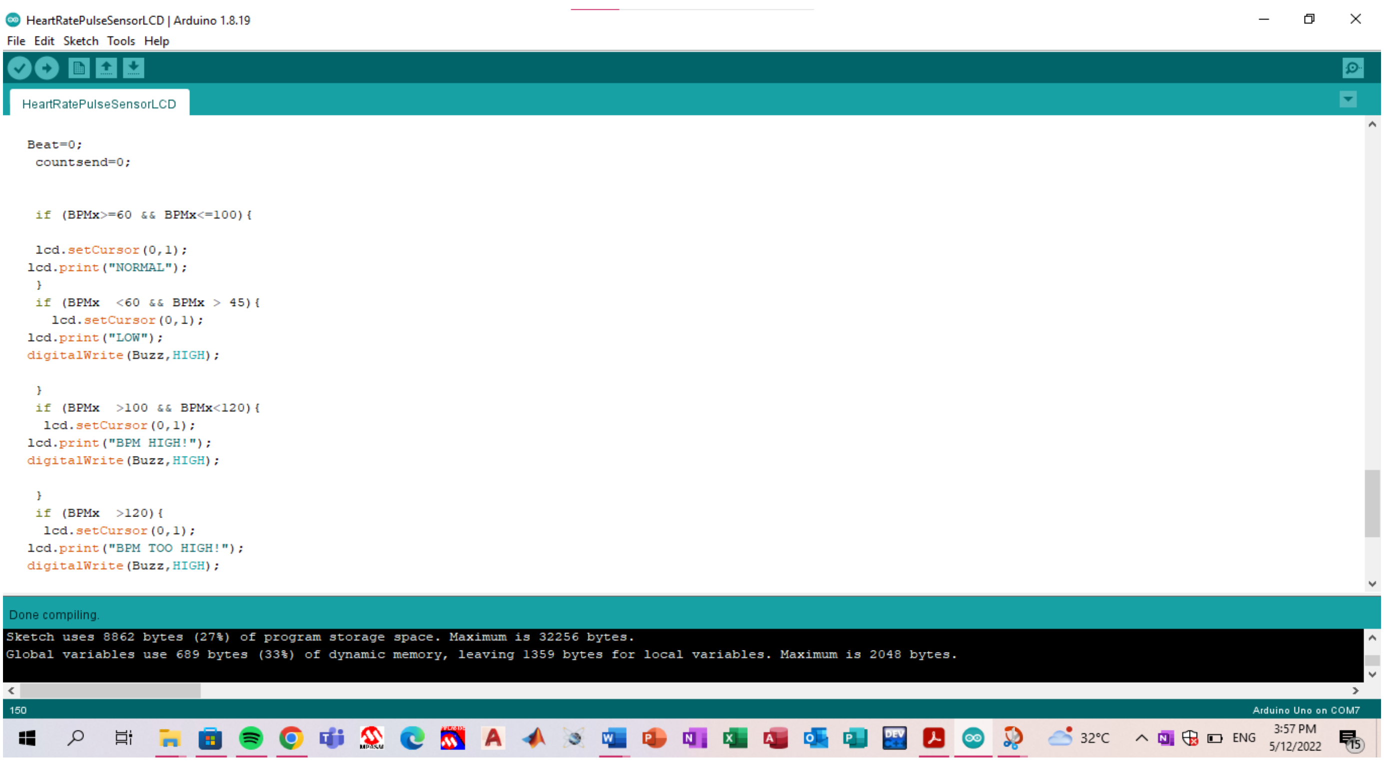
5.


HeartRatePulseSensorLCD | Arduino 1.8.19
File Edit Sketch Tools Help
HeartRatePulseSensorLCD
if (Signal >=Minx){
if (Trig==0){
Trig=1;
}
}
if (Trig==1){
if (Signal < Minx){
Beat++;
Trig=0;
OldVal=0;
}
}
//--
countsend++;
if (countsend>2000){
// Serial.println(Beat);
BPM=Beat*3.52380952;
if (BPM<=30){
BPM=0; BPMx=0;
}
Done compiling.
Sketch uses 8862 bytes (27%) of program storage space. Maximum is 32256 bytes.
Global variables use 689 bytes (33%) of dynamic memory, leaving 1359 bytes for local variables. Maximum is 2048 bytes.
150
Arduino Uno on COM7
32°C
ENG
3:55 PM
5/12/2022

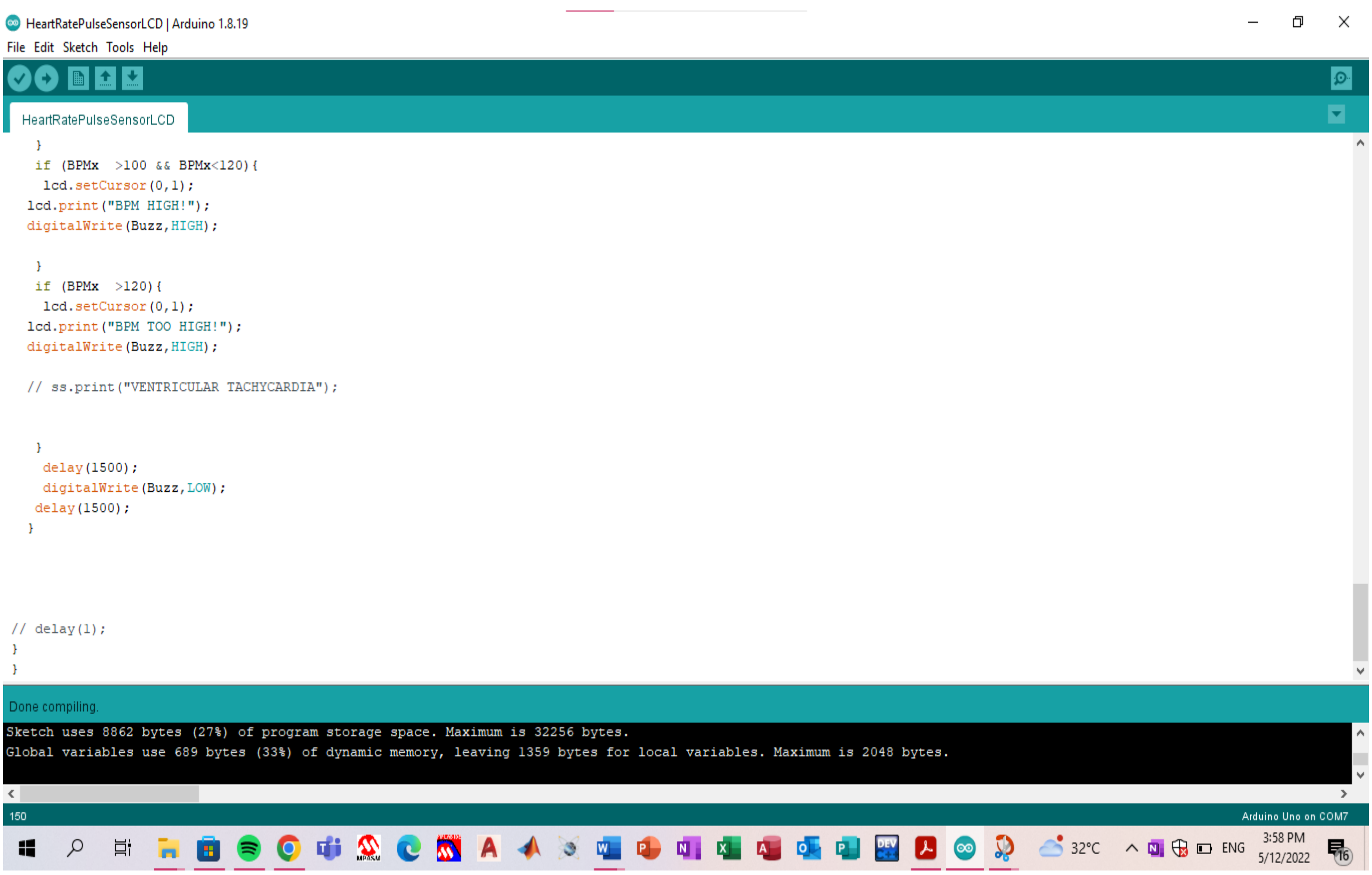
6.


HeartRatePulseSensorLCD | Arduino 1.8.19
File Edit Sketch Tools Help
HeartRatePulseSensorLCD
// Serial.println(Beat);
BPM=Beat*3.52380952;
if (BPM<=30){
BPM=0; BPMx=0;
}
if (BPM>40){
BPMx=BPM;
ss.print("*");
ss.print(BPMx);
ss.println("#");
}
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Heart Rate:");
lcd.print(BPMx,0);
// lcd.print(Beat);
Tmax=0;
Tmin=100;
Done compiling.
Sketch uses 8862 bytes (27%) of program storage space. Maximum is 32256 bytes.
Global variables use 689 bytes (33%) of dynamic memory, leaving 1359 bytes for local variables. Maximum is 2048 bytes.
150
Arduino Uno on COM7
32°C
ENG
3:56 PM
5/12/2022

7.


HeartRatePulseSensorLCD | Arduino 1.8.19
File Edit Sketch Tools Help
HeartRatePulseSensorLCD
Beat=0;
countsend=0;
if (BPMx>=60 && BPMx<=100){
lcd.setCursor(0,1);
lcd.print("NORMAL");
}
if (BPMx <60 && BPMx > 45){
lcd.setCursor(0,1);
lcd.print("LOW");
digitalWrite(Buzz,HIGH);
}
if (BPMx >100 && BPMx<120){
lcd.setCursor(0,1);
lcd.print("BPM HIGH!");
digitalWrite(Buzz,HIGH);
}
if (BPMx >120){
lcd.setCursor(0,1);
lcd.print("BPM TOO HIGH!");
digitalWrite(Buzz,HIGH);
Done compiling.
Sketch uses 8862 bytes (27%) of program storage space. Maximum is 32256 bytes.
Global variables use 689 bytes (33%) of dynamic memory, leaving 1359 bytes for local variables. Maximum is 2048 bytes.
150
Arduino Uno on COM7
32°C
ENG
3:57 PM
5/12/2022

8.


HeartRatePulseSensorLCD | Arduino 1.8.19
File Edit Sketch Tools Help
HeartRatePulseSensorLCD
}
if (BPMx >100 && BPMx<120){
lcd.setCursor(0,1);
lcd.print("BPM HIGH!");
digitalWrite(Buzz,HIGH);
}
if (BPMx >120){
lcd.setCursor(0,1);
lcd.print("BPM TOO HIGH!");
digitalWrite(Buzz,HIGH);
// ss.print("VENTRICULAR TACHYCARDIA");
}
delay(1500);
digitalWrite(Buzz,LOW);
delay(1500);
}
// delay(1);
}
}
Done compiling.
Sketch uses 8862 bytes (27%) of program storage space. Maximum is 32256 bytes.
Global variables use 689 bytes (33%) of dynamic memory, leaving 1359 bytes for local variables. Maximum is 2048 bytes.
150
Arduino Uno on COM7
32°C
ENG
3:58 PM
5/12/2022