Visvesvaraya Technological University, Belagavi

A project synopsis on

"PERVASIVE HEALTHCARE"

Submitted in partial fulfillment of the requirement for the award of the degree

Bachelor of Engineering

in

Electronics and Communication Engineering

Submitted by

HITHAISHI P (USN: 4MH13EC038)

KARTHIK M PRAKASH (USN: 4MH13EC041)

KARTHIK R (USN: 4MH13EC042)

MANU KASHYAP N (USN: 4MH13EC059)

Under the guidance of

Dr. MAHESH RAO

Professor, HOD & PG Coordinator,

Dept. of E&CE,

MIT MYSORE

Department of Electronics & Communication Engineering

Maharaja Institute of Technology Mysore

2016-2017

Visvesvaraya Technological University, Belagavi

Maharaja Institute of Technology Mysore

Department of Electronics and Communication Engineering

Certificate

Certified that the project work entitled **"Pervasive Healthcare"** carried out by **Hithaishi P(4MH13EC038), Karthik M Prakash(4MH13EC041), Karthik R(4MH13EC042), Manu Kashyap N(4MH13EC059)** are the bonafide students of **MIT Mysore** submitted in partial fulfillment for the award of **Bachelor of Engineering** in **Electronics and Communication Engineering** of **Visvesvaraya Technological University, Belgavi** during the year **2016-2017**. It is certified that all corrections and suggestions indicated for internal assessment have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said Degree.

Signature of the Guide	Signature of the HOD	Signature of the Principal
Dr. Mahesh Rao	**Dr. Mahesh Rao**	**Dr. B.G. Naresh Kumar**
Prof. & HOD	Prof. & HOD	Principal
Dept. of E&CE	Dept. of E&CE	MIT Mysore
MIT Mysore	MIT Mysore	

External Viva

Name of the examiners: Signature with date

1. ____________________ ____________________

2. ____________________ ____________________

Acknowledgement

The satisfaction and gratitude accompanying the success of any work would be incomplete unless extending our sincere regards to all who made it doable, with their constant guidance and encouragement making this effort fruitful.

We express our gratitude and respect to **Dr. B.G. Naresh Kumar**, Principal, MIT Mysore for granting permission to carry out the project work.

We express our deep sense of gratitude to our guide **Dr. Mahesh Rao.** Professor and Head, Department of Electronics and Communication Engineering, MIT Mysore for his valuable suggestions and inspiration. He has been the driving force behind this work and has constantly dedicated his precious time to successfully carry out the project work.

Last but not the least, we would like to thank our **Parents and Friends** for their moral support during critical phases of our work.

Hithaishi P

Karthik M Prakash

Karthik R

Manu Kashyap N

Abstract

The idea is to develop a compact and integrated embedded system for medical practitioners to remotely diagnose a patient over the WEB by providing access to the physiological data gathered by the basic diagnostic equipment such as Electrocardiogram (ECG), Digital Sphygmomanometer (BP), Pulse Oximeter and a Thermometer. This design will significantly reduce the cost incurred in providing basic diagnosis. The innovativeness of the project lies in using Digital Signal Processing to continuously monitor the health characteristics of the patient and wirelessly streaming the bio-telemetry to smartphones and computers. The microprocessor will continuously analyze the readings and send warning and the readings to the doctor when there is abnormality in patient's health.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Chapter 1

Preamble

1.1 Introduction

As the population of the world is increasing exponentially, the demand upon the healthcare system is skyrocketing, which costs more money and resources to provide for the infrastructure. Nearly 27% of the total deaths in India happen with no medical attention at the time of death, according to the 2013 ‘Civil Registration Data’ released by the Census Directorate. In the present scenario the patient needs to be admitted into a healthcare facility where a professional will record the readings of the diagnostic equipment and this recorded reading has to be analyzed by a doctor. The time delay involved here is the main cause for the high mortality rate shown by the census report. In the proposed design, a lay man with the most basic knowledge how to stick a few electrode stickers and put a clip on a finger will be able use the device right at the comfort of their homes.

1.2 Motivation

There are an estimated of 371 million smart phones connected to the internet in India today. There can’t be a bigger and better way of providing healthcare to the masses than catering it right to their fingertips.

Hospitals are to be highly committed to patient’s health. In spite of all the advancements in technology in the field of medical sciences, causalities occur due to delay in diagnosis. Our proposal prevents this by monitoring patients round the clock. It is not just designed to be used in hospitals but also in homes. But a doctor monitoring a patient round the clock is an impossible task. This is when the automation comes to rescue. All the data acquired by the integrated devices is continuously monitored by the MCU. It is programed to find any anomalies in the acquired data which would raise a flag and to notify the concerned doctor about the situation. This significantly reduces the casualty rate due to delayed diagnosis.

1.3 Objective and Scope

The device is intended to reduce the load on a medical practitioner who has to monitor many number of patients or to remotely get their health updates. The design uses Digital Signal Processing to monitor the patient's health characteristics. Thus the monitoring becomes inherently accurate, consistent and reliable which are the most necessary qualities of a good diagnosis. It completely changes the way how patients are diagnosed. The innovativeness of the proposal is using the web to wirelessly stream telemetry.

1.4 Block Diagram

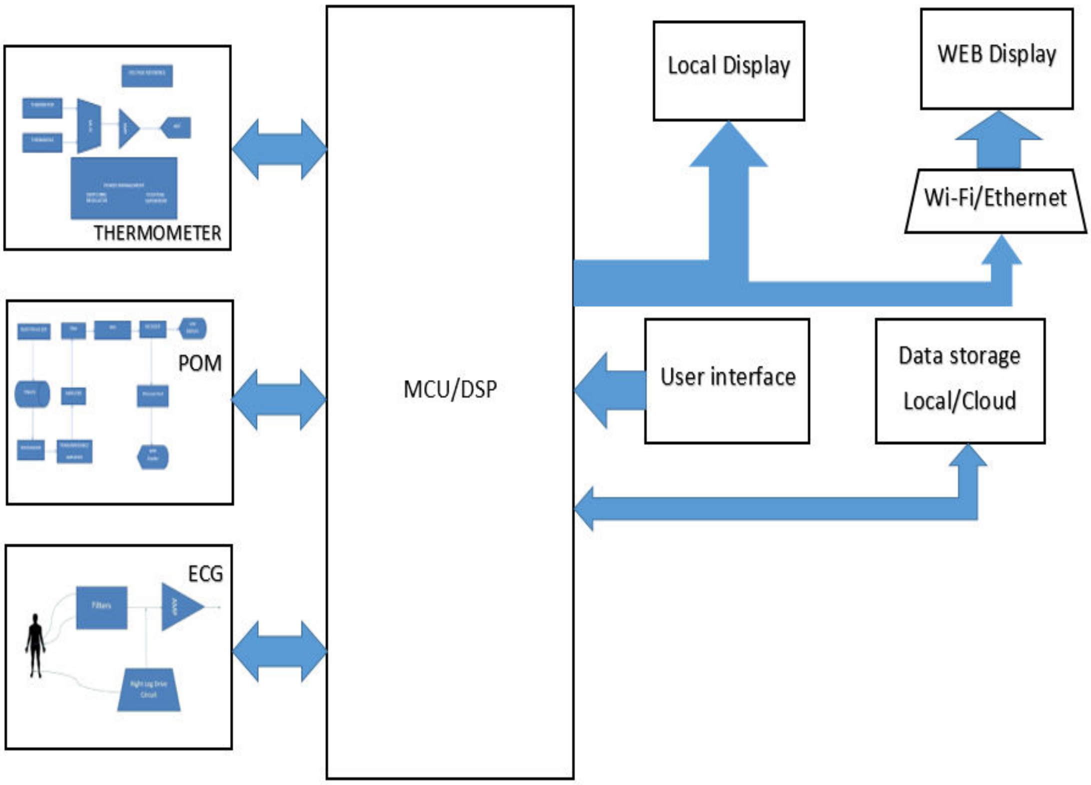


Fig. 1.1: Block Diagram

1.5 Report Organization

Chapter 1: This chapter gives a brief overview of the project, also discusses the motivation, objective and scope of the project and block diagram.

Chapter 2: This chapter provides a general overview of existing literature related to the project.

Chapter 3: This chapter discusses design and implementation of the proposed methodology. It briefly describes the hardware components used in the project.

Chapter 4: This chapter describes the software tools used in the implementation of the project and software flow.

Chapter 5: This chapter presents the results and analysis, the conclusion of the project and discusses the scope for future enhancements.

Chapter 2

Literature Survey

Andre Canha [1] deals with two systems namely Altcare and SmartCondo for pervasive healthcare in smart home. They monitor every movement of the patient using cameras and store the data. Additional infrastructure such as the installation of cameras everywhere in the home, maintaining data base of 2D and 3D pictures and most importantly maintaining the privacy of the patient are debatable issues here. Our design concentrates mainly on monitoring the health rather than the movement of the patient thus not impinging on the user's privacy.

Y. Ren, R. W. N. Pazzi, and A. Boukerche [2] have developed a highly customizable vital signs monitoring system based on a body area network (BAN) and a mobile-health (m-health) service platform utilizing next generation public wireless networks. The developed system allows the incorporation of diverse medical sensors via wireless connections, and the live transmission of the measured vital signs over public wireless networks to healthcare providers. Nine trials with different healthcare scenarios and patient groups in four different European countries have been conducted. These have been performed to test the service and the network infrastructure including its suitability for mobile healthcare applications. Preliminarily results have documented the feasibility of using the system, but also demonstrated logistical problems with use of the BANs and the infrastructure for transmitting mobile healthcare data.

A wireless body sensor network hardware has been designed and implemented [3] based on MICS (Medical Implant Communication Service) band. The sensor node can transmit data over the air to a remote Intelligent Medical Server (IMS) for further processing, monitoring and storage. The developed system offers medical staff to obtain patient's physiological data on demand basis via the Internet. Some preliminary performance data is presented in the paper.

Xiaohui Liang [4] introduced a Remote Health Monitoring (RHM) system which sends data from user's wearable sensors to a personal server. Then, through any wireless connection, the data are sent through an internet gateway and the RHM servers, to a remote medical doctor's site for real-time diagnosis, and then updated in database. They define the

RHM architecture and summarize the design considerations. They have presented a promising commercialized solution. They have also discussed regarding future research challenges for implementing RHM systems.

Chapter 3

Hardware Requirements

3.1 Electrocardiogram

Electrocardiogram (ECG) is the process of recording the electrical activity of the heart over a period of time using electrodes placed on the skin. It is a surface measurement of the electrical potential generated by electrical activity in cardiac tissue. Current flow, in the form of ions, signals the contraction of cardiac muscle fibers. This leads to the heart's pumping action.

The ECG is a valuable, non–invasive diagnostic tool which was first put to clinical use in 1913 with Einthoven's invention of the string galvanometer. Einthoven's recording is known as the "three lead" ECG, with measurements taken from three points on the body (defining the "Einthoven triangle" — an equilateral triangle with the heart at the center.) The difference between potential readings from L1 and L2 is what is used to produce the output ECG trace. The L3 connection establishes a common ground for the body and the recording device (oscilloscope.)

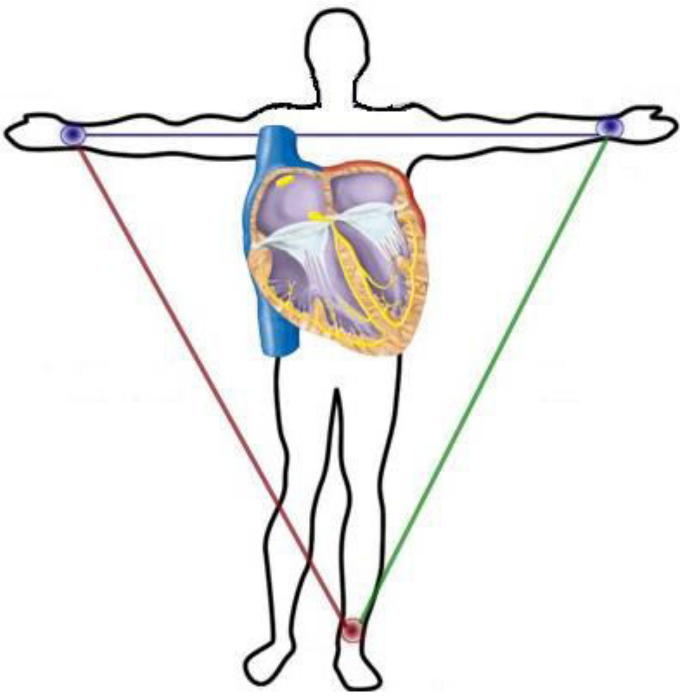


Fig. 3.1: Einthoven's Triangle

3.2 Electrical Activity of The Heart

The heart is the organ responsible for pumping blood throughout the body. It is located in the middle of the thorax, slightly offset to the left and surrounded by the lungs. The heart is composed of four chambers; two atriums and two ventricles. The right atrium receives blood returning to the heart from the whole body. This blood passes through the right ventricle and is pumped to the lungs where it is oxygenated and goes back to the heart through the left atrium, then the blood passes through the left ventricle and is pumped again to be distributed to the entire body through the network of arteries.

The events that occur in the heart on each Cardiac Cycle:

1. Atrium begins to depolarize.
2. Atrium depolarizes.
3. Ventricles begin to depolarize at apex. Atrium re polarizes.
4. Ventricles depolarize.
5. Ventricles begin to re polarize at apex.
6. Ventricles re polarize.

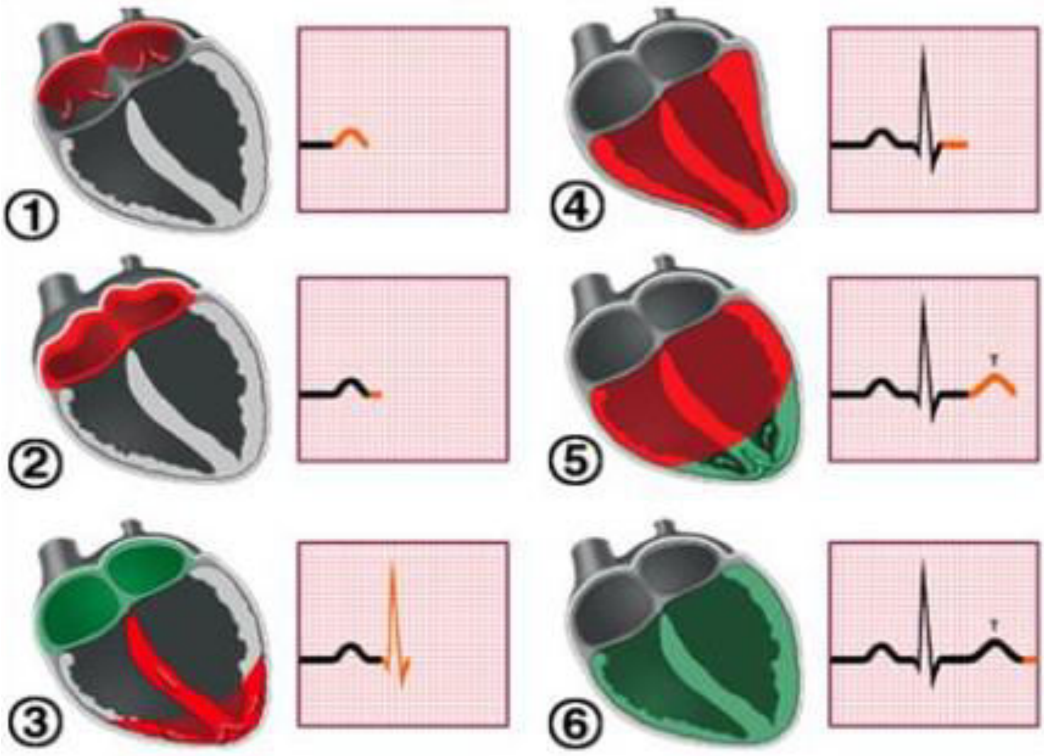


Fig. 3.2: Electrical Activity of the Heart

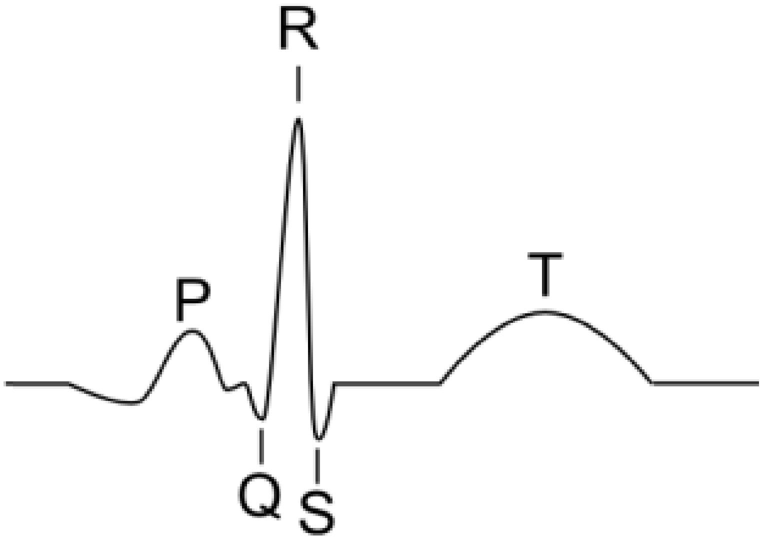


Fig. 3.3: Typical ECG Waveform

3.2.1 The P Wave

The first deflection of the heartbeat is a small upward wave called the P wave. It indicates atrial depolarization. The initial portion of the P wave is largely a reflection of right atrial depolarization and the terminal portion is largely a reflection of left atrial depolarization.

3.2.2 The Q Wave

The Q wave is an initial downward deflection after the P wave. The normal Q wave represents septal depolarization.

3.2.3 The R Wave

The R wave is the first upward deflection after the P wave (even when Q waves are absent). The R wave is normally the easiest waveform to identify on the ECG and represents early ventricular depolarization.

3.2.4 The S Wave

The S wave is the first negative deflection after the R wave. It represents the late ventricular depolarization.

3.2.5 The T Wave

The T wave represents repolarisation of the ventricles. It is normally upright, somewhat rounded, and slightly asymmetric.

3.3 ECG Circuit

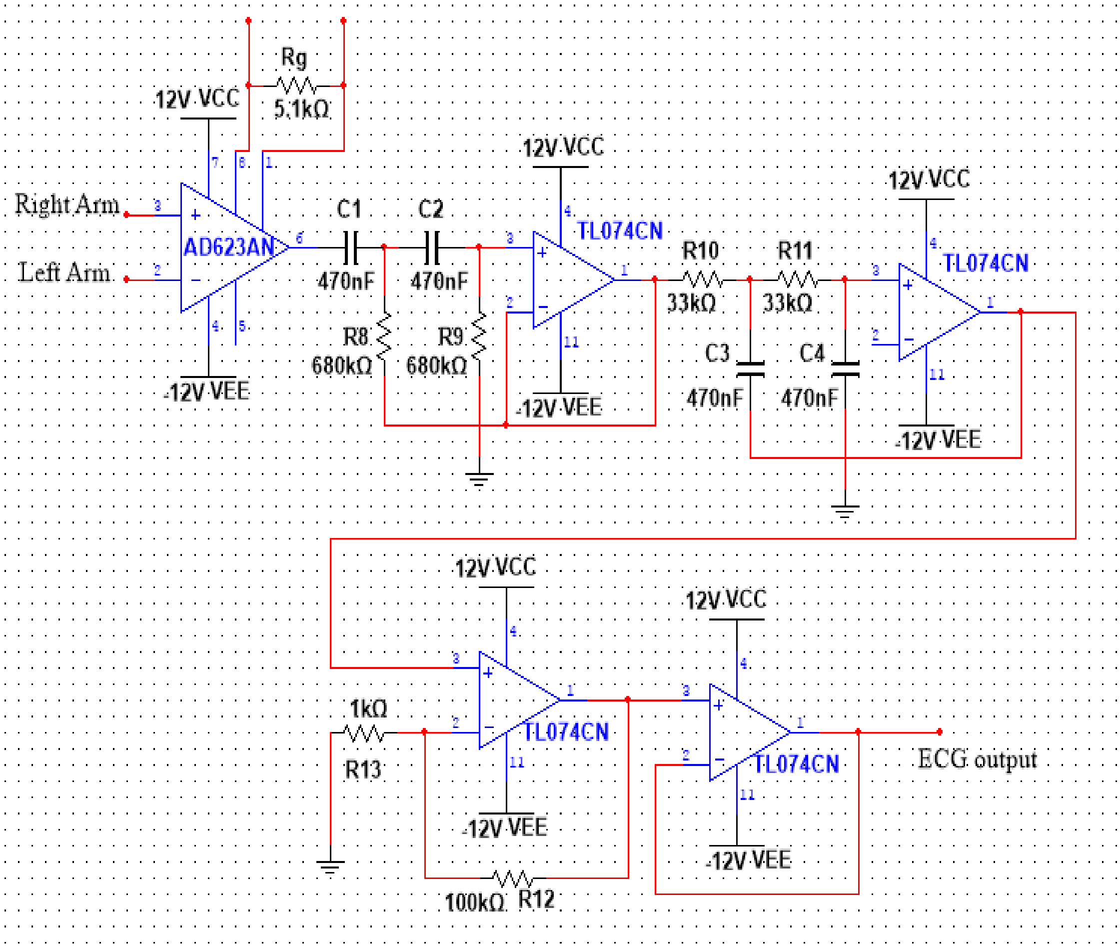


Fig. 3.4: ECG Circuit

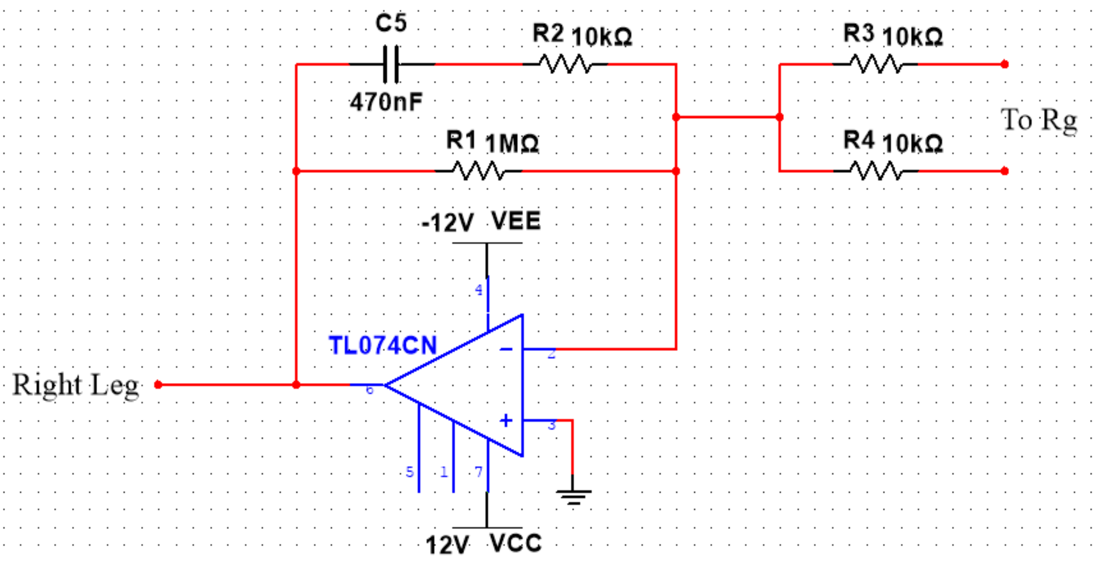


Fig. 3.5: Right Leg Drive Circuit

The circuit consists of an instrumentation amplifier, low-pass and high-pass filters and right leg drive circuit.

3.3.1 Instrumentation Amplifier

The Instrumentation amplifier is a cascade of differential amplifier and a summer. Instrumentation amplifier AD623AN from 'Analog Devices Inc.' is used. It is designed for gain G=10. The gain is set by varying resistor R_g.

The gain of the instrumentation amplifier is chosen as 10. The Rg corresponding to this gain is calculated from the formula specified by the manufacturer:

$$R_g = \frac{49.4\text{k}\Omega}{G-1}$$

$$= \frac{49.4\text{k}\Omega}{10-1}$$

$$=5.48\text{k}\Omega$$

3.3.2 High-Pass and Low-Pass Filter

Second order Butterworth active low pass and high pass filters are used to filter noise from the ECG signals. The high pass filter is designed with cut-off frequency f_h = 0.5Hz. The low pass filter is designed with cut-off frequency f_l = 10Hz.

The design follows the formula:

$$F = \frac{1}{2\pi RC}$$

3.3.3 Gain Stage

A non-inverting amplifier is cascaded with the filters to amplify the ECG Signals. The amplifier is designed for a gain 100.

The design follows the expression:

$$A_V = 1+\frac{R_f}{R_l}$$

The overall gain of the circuit is 1000.

3.3.4 Right leg drive circuit

Right leg acts as an electrical ground for the circuit. The common mode noise is cancelled by sending 180^{o} offset noise through the right leg.

Common mode noise from the body is picked up at R_g of the instrumentation amplifier. Right leg drive circuit is an inverting amplifier. Hence the inverted noise is supplied back to the body via the right leg lead, there by actively cancelling the noise.

3.4 ARDUINO UNO (ATmega 328P)

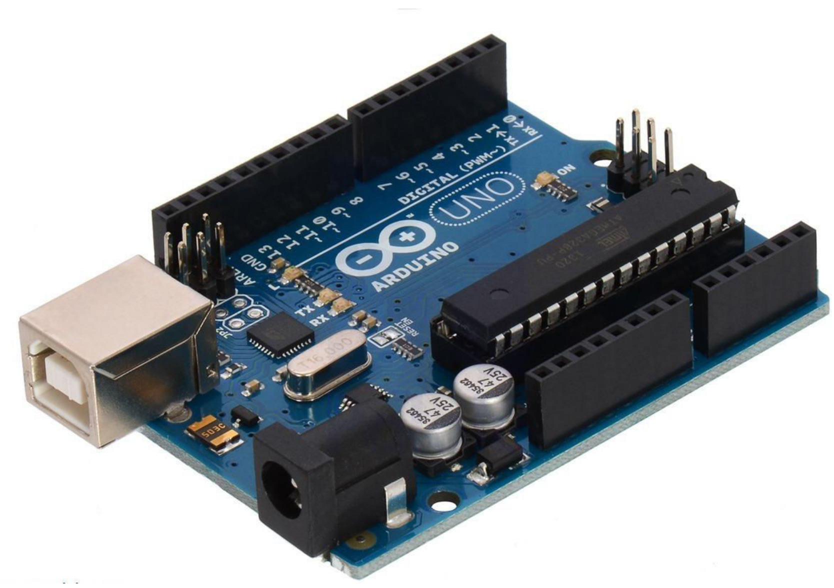


Figure 3.1: Arduino Uno Board

3.4.1 Overview

The Arduino UNO is the most used board in the family of Arduino boards. Arduino Uno is a microcontroller board based on the Atmega328P. It has 14 digital input/output pins of which 6 can be used as PWM outputs, 6 analog inputs, a 16MHz quartz crystal with 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. It has a resettable polyfuse that protects the 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 500mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Table 3.1: Technical specifications of Arduino UNO

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 of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm
Weight	25 g

3.4.2 Power

The Arduino Uno board 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 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 from 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may become 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:

1) **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.
2) **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 of 7 - 12V, the USB connector of 5V, or the VIN pin of the board of 7-12V. Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
3) **3V3**: A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
4) **GND**: Ground pins.
5) **IOREF**: This pin on the Arduino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs to work with the 5V or 3.3V.

3.4.3 Memory

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

3.4.4 Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output pins, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive 20mA as recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50k ohm.

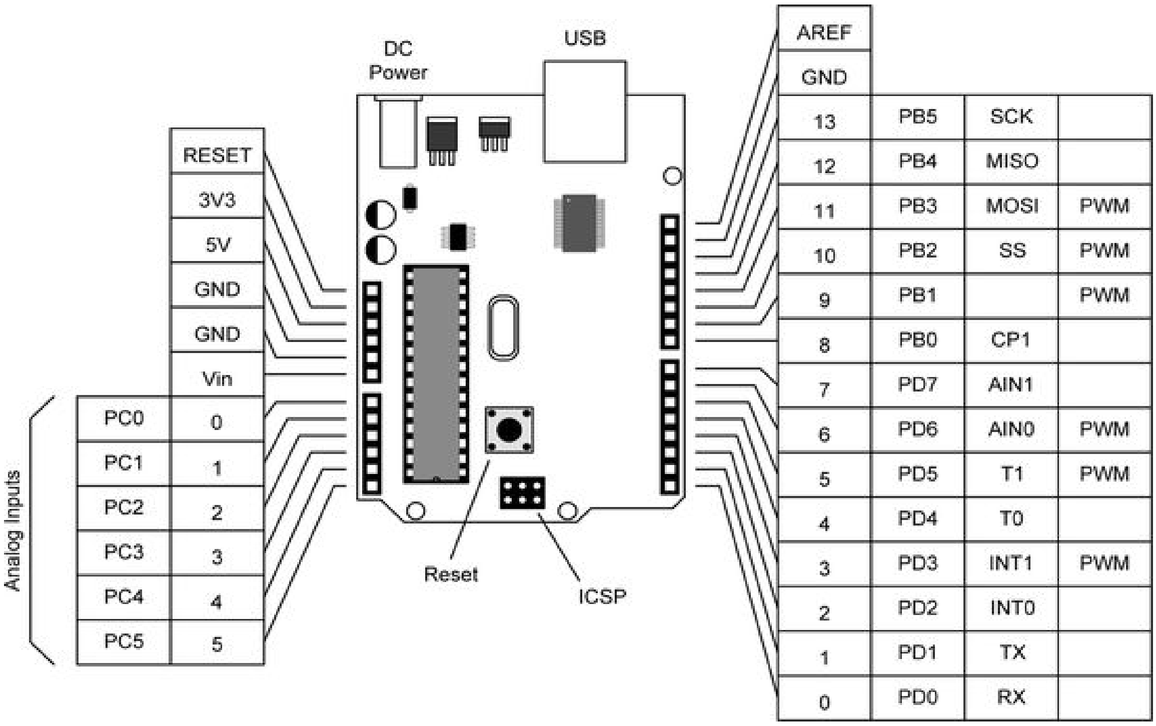


Figure 3.2: pin diagram of Arduino Uno

A maximum of 40mA is the value that must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller. In addition, some pins have specialized functions:

1) **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.
2) **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.

3) **PWM**: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.
4) **SPI**: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.
5) **LED**: 13. There is a built-in LED driven by digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
6) **TWI**: A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.

   The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution. 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. There are a couple of other pins on the board:

1) **AREF**: Reference voltage for the analog inputs. Used with analogReference().
2) **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.

3.4.5 Communication

Arduino Uno has a number of facilities for communicating with a computer, another Arduino board, or other microcontrollers. The ATmega328P 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. The Arduino Software (IDE) includes a serial monitor which allows simple textual data to be sent to and from the 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 serial communication on any of the Uno's digital pins. The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino Software (IDE) includes a Wire library to simplify use of the I2C bus; see the documentation for details. For SPI communication, use the SPI library.

3.5 ESP8266 (Wi-Fi Module)

Fig 3.3: ESP8266

3.5.1 Overview

ESP8266EX delivers highly integrated Wi-Fi SoC solution to meet the continuous demands for efficient power usage, compact design and reliable performance in the Internet of Things industry.

With the complete and self-contained Wi-Fi networking capabilities, ESP8266EX can perform either as a standalone application or as the slave to a host MCU. When ESP8266EX hosts the application, it promptly boots up from the flash. The integrated highspeed cache helps to increase the system performance and optimize the system memory. Also, ESP8266EX can be applied to any micro-controller design as a Wi-Fi adaptor through SPI / SDIO or I2C / UART interfaces.

ESP8266EX integrates antenna switches, RF balun, power amplifier, low noise receive amplifier, filters and power management modules. The compact design minimizes the PCB size and requires minimal external circuitries.

Besides the Wi-Fi functionalities, ESP8266EX also integrates an enhanced version of Tensilica's L106 Diamond series 32-bit processor and on-chip SRAM. It can be interfaced with external sensors and other devices through the GPIOs. Software Development Kit.

3.5.2 Wi-Fi Protocols

- 802.11 b/g/n/e/i support.
- Wi-Fi Direct (P2P) support.
- P2P Discovery, P2P GO (Group Owner) mode, GC(Group Client) mode and P2P Power Management.
- Infrastructure BSS Station mode / P2P mode / SoftAP mode support.
- Hardware accelerators for CCMP (CBC-MAC, counter mode), TKIP (MIC, RC4), WAPI (SMS4), WEP (RC4), CRC.
- WPA/WPA2 PSK, and WPS driver.
- Additional 802.11i security features such as pre-authentication, and TSN.
- Open Interface for various upper layer authentication schemes over EAP such as TLS, PEAP, LEAP, SIM, AKA, or customer specific.
- 802.11n support (2.4 GHz).
- Supports MIMO 1×1 and 2×1, STBC, A-MPDU and A-MSDU frame aggregation and 0.4µs guard interval.

3.5.3 Main Technical Specifications

Table 3.2. Main Technical Specifications

Categories	Items	Parameters
Wi-Fi	Standards	FCC/CE/TELEC/SRRC
	Protocols	802.11 b/g/n/e/i
	Frequency Range	2.4G ~ 2.5G (2400M ~ 2483.5M)
	Tx Power	802.11 b: +20 dBm
		802.11 g: +17 dBm
		802.11 n: +14 dBm
	Rx Sensitivity	802.11 b: -91 dbm (11 Mbps)
		802.11 g: -75 dbm (54 Mbps)
		802.11 n: -72 dbm (MCS7)
	Antenna	PCB Trace, External, IPEX Connector, Ceramic Chip
Hardware	CPU	Tensilica L106 32-bit micro controller
	Peripheral Interface	UART/SDIO/SPI/I2C/I2S/IR Remote Control
		GPIO/ADC/PWM/LED Light & Button
	Operating Voltage	2.5V – 3.6V
	Operating Current	Average value: 80 mA
	Operating Temperature Range	-40°C – 125°C
	Storage Temperature Range	-40°C – 125°C
	Package Size	QFN32-pin (5 mm x 5 mm)
	External Interface	-
Software	Wi-Fi Mode	Station/SoftAP/SoftAP+Station
	Security	WPA/WPA2
	Encryption	WEP/TKIP/AES
	Firmware Upgrade	UART Download / OTA (via network)
	Software Development	Supports Cloud Server Development / Firmware and SDK for fast on-chip programming
	Network Protocols	IPv4, TCP/UDP/HTTP/FTP
	User Configuration	AT Instruction Set, Cloud Server, Android/iOS App

3.5.4 Functional Description

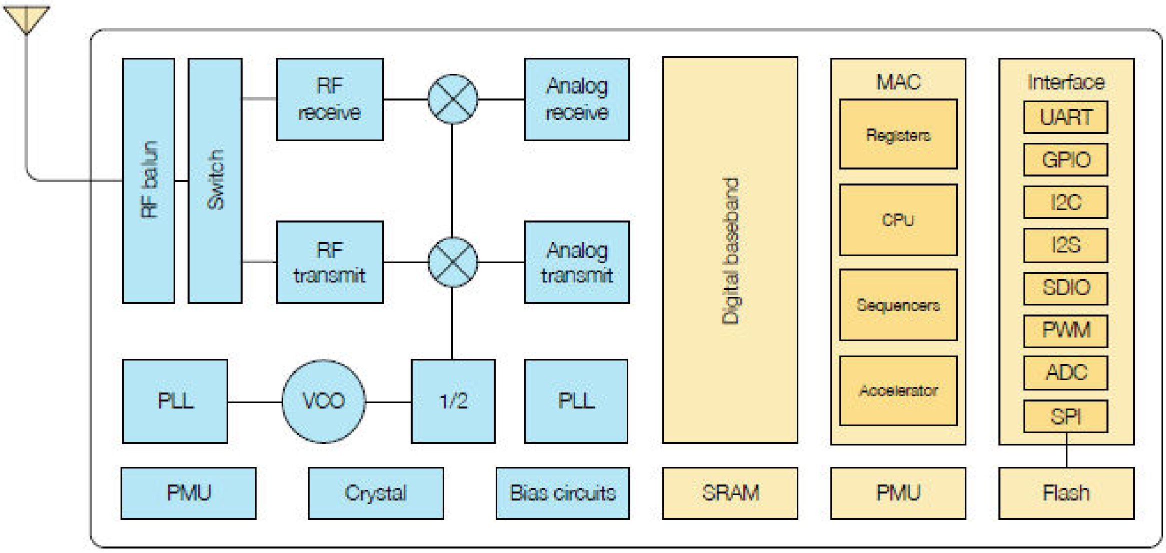


Fig 3.4: Functional Description

3.5.4.1 CPU

ESP8266EX integrates Tensilica L106 32-bit micro controller (MCU) and ultra-low-power 16-bit RSIC. The CPU clock speed is 80 MHz. It can also reach a maximum value of 160 MHz. Real Time Operation System (RTOS) is enabled. Currently, only 20% of MIPS has been occupied by the Wi-Fi stack, the rest can all be used for user application programming and development. The CPU includes the interfaces as below.

- Programmable RAM/ROM interfaces (iBus), which can be connected with memory
- controller, and can also be used to visit flash.
- Data RAM interface (dBus), which can be connected with memory controller.
- AHB interface which can be used to visit the register.

3.5.4.2 Memory

ESP8266EX Wi-Fi SoC integrates memory controller and memory units including SRAM and ROM. MCU can access the memory units through iBus, dBus, and AHB interfaces. All memory units can be accessed upon request, while a memory arbiter will decide the running sequence according to the time when these requests are received by the processor. According to the current version of SDK, SRAM space available to users is assigned as below.

- RAM size < 50 kB, that is, when ESP8266EX is working under the Station mode and connects to the router, programmable space accessible in heap + data section is around 50 kB.
- There is no programmable ROM in the SoC, therefore, user program must be stored in an external SPI flash.

3.5.4.3 External Flash

ESP8266EX uses external SPI flash to store user programs, and supports up to 16 MB memory capacity theoretically.

3.5.5 Radio

ESP8266EX radio consists of the following blocks.

- 2.4 GHz receiver
- 2.4 GHz transmitter
- High speed clock generators and crystal oscillator
- Real time clock
- Bias and regulators
- Power management

3.5.5.1 Channel Frequencies

The RF transceiver supports the following channels according to IEEE802.11b/g/n standards.

Table 3.3: Channel Frequencies

Channel No.	Frequency (MHz)	Channel No.	Frequency (MHz)
1	2412	8	2447
2	2417	9	2452
3	2422	10	2457
4	2427	11	2462
5	2432	12	2467
6	2437	13	2472
7	2442	14	2484

the appropriate registers. Each GPIO can be configured with internal pull-up or pull-down, or set to high impedance, and when configured as an input, the data are stored in software registers; the input can also be set to edge-trigger or level trigger CPU interrupts. In short, the IO pads are bidirectional, non-inverting and tristate, which includes input and output buffer with tristate control inputs.

These pins can be multiplexed with other functions such as I2C, I2S, UART, PWM, IR Remote Control, LED Light and Button etc. For low power operations, the GPIOs can also be set to hold their state. For instance, when the chip is powered down, all output enable signals can be set to hold low. Optional hold functionality can be built into the IO if requested. When the IO is not driven by the internal or external circuitry, the hold functionality can be used to hold the state to the last used state. The hold functionality introduces some positive feedback into the pad.

Hence, the external driver that drives the pad must be stronger than the positive feedback. The required drive strength is small — in the range of 5 µA to pull apart the latch.

3.5.6 I2C Interface

ESP8266EX has one I2C used to connect with micro-controller and other peripheral equipment such as sensors. The pin definition of I2C is as below.

Table 3.4: Pin Definitions of I2C

Pin Name	Pin Num	IO	Function Name
MTMS	9	IO14	I2C_SCL
GPIO2	14	IO2	I2C_SDA

Both I2C Master and I2C Slave are supported. I2C interface functionality can be realized via software programming, and the clock frequency is 100 kHz at a maximum. It should be noted that I2C clock frequency should be higher than the slowest clock frequency of the slave device.

3.5.7 I2S Interface

ESP8266EX has one I2S data input interface and one I2S data output interface. I2S interfaces are mainly used in applications such as data collection, processing, and transmission of audio data, as well as the input and output of serial data. For example, LED lights (WS2812 series) are supported. The pin definition of I2S is shown in Table 4-5. I2S functionality can be enabled via software programming by using multiplexed GPIOs, and linked list DMA is supported.

Table 3.5: Pin Definitions of I2S

I2S Data Input			
Pin Name	Pin Num	IO	Function Name
MTDI	10	IO12	I2SI_DATA
MTCK	12	IO13	I2SI_BCK
MTMS	9	IO14	I2SI_WS
MTDO	13	IO15	I2SO_BCK
U0RXD	25	IO3	I2SO_DATA
GPIO2	14	IO2	I2SO_WS

3.6 Pulse-Oximeter

Pulse oximetry is a noninvasive method for monitoring a person's oxygen saturation (SO2). Its reading of SpO2 (peripheral oxygen saturation) is not always identical to the reading of SaO2 (arterial oxygen saturation) from arterial blood gas analysis, but the two are correlated well enough that the safe, convenient, noninvasive, inexpensive pulse oximetry method is valuable for measuring oxygen saturation in clinical use. The device being used to measure is known as Pulse Oximeter and the graph obtained is called plethysmography.

A pulse oximeter is a medical device that indirectly monitors the oxygen saturation of a patient's blood and changes in blood volume in the skin, producing a photoplethysmogram. The pulse oximeter may be incorporated into a multi-parameter patient monitor. Most monitors also display the pulse rate. Portable, battery-operated pulse oximeters are also available for transport or home blood-oxygen monitoring.

Pulse oximeters for home use are small, lightweight monitors that painlessly attach to a fingertip, to monitor the amount of oxygen carried in the body. An oxygen level of greater than 95% is generally considered to be a normal oxygen level. An oxygen level of 92% or less (at sea level) suggests low blood oxygen. In addition to oxygen level, pulse rate is also displayed. Normal pulse rate values for adults 60-80 beats per minute2. Heart rate can be expected to increase some with exercise and oxygen saturation may slightly decrease (it should still remain at 90% or greater though).

3.6.1 Function

A blood-oxygen monitor displays the percentage of blood that is loaded with oxygen. More specifically, it measures what percentage of hemoglobin, the protein in blood that carries oxygen, is loaded. Acceptable normal ranges for patients without pulmonary pathology are from 95 to 99 percent. For a patient breathing room air at or near sea level, an estimate of arterial pO2 can be made from the blood-oxygen monitor "saturation of peripheral oxygen" (SpO2) reading.

A typical pulse oximeter utilizes an electronic processor and a pair of small light-emitting diodes (LEDs) facing a photodiode through a translucent part of the patient's body, usually a fingertip or an earlobe. One LED is red, with wavelength of 660 nm, and the other is infrared with a wavelength of 940 nm. Absorption of light at these wavelengths differs significantly between blood loaded with oxygen and blood lacking oxygen. Oxygenated hemoglobin absorbs more infrared light and allows more red light to pass through. Deoxygenated hemoglobin allows more infrared light to pass through and absorbs more red light. The LEDs sequence through their cycle of one on, then the other, then both off about thirty times per second which allows the photodiode to respond to the red and infrared light separately and also adjust for the ambient light baseline. The amount of light that is transmitted is measured, and separate normalized signals are produced for each wavelength. These signals fluctuate in time because the amount of arterial blood that is present increases with each heartbeat. By subtracting the minimum transmitted light from the peak transmitted light in each wavelength, the effects of other tissues is corrected for. The ratio of the red light measurement to the infrared light measurement is then calculated by the processor (which represents the ratio of oxygenated hemoglobin to deoxygenated hemoglobin), and this ratio is then converted to SpO2 by the processor via a lookup table based on the Beer–Lambert law.

3.6.2 Advantages

Pulse oximetry is particularly convenient for noninvasive continuous measurement of blood oxygen saturation. Pulse oximetry is useful in any setting where a patient's oxygenation is unstable, including intensive care, operating, recovery, emergency and hospital ward settings, pilots in unpressurized aircraft, for assessment of any patient's oxygenation, and determining the effectiveness of or need for supplemental oxygen.

Because of their simplicity of use and the ability to provide continuous and immediate oxygen saturation values, pulse oximeters are of critical importance in emergency medicine and are also very useful for patients with respiratory or cardiac problems, especially COPD, or for diagnosis of some sleep disorders such as apnea and hypopnea. Portable battery-operated pulse oximeters are useful for pilots operating in a non-pressurized aircraft above 10,000 where supplemental oxygen is required. Portable pulse oximeters are also useful for mountain climbers and athletes whose oxygen levels may decrease at high altitudes or with exercise.

3.6.3 Limitations

Pulse oximetry measures solely hemoglobin saturation, not ventilation and is not a complete measure of respiratory sufficiency. Erroneously low readings may be caused by hypo perfusion of the extremity being used for monitoring incorrect sensor application; highly calloused skin; or movement, especially during hypo perfusion.

To ensure accuracy, the sensor should return a steady pulse and pulse waveform. Pulse oximetry technologies differ in their abilities to provide accurate data during conditions of motion and low perfusion. Pulse oximetry also is not a complete measure of circulatory sufficiency. If there is insufficient blood flow or insufficient hemoglobin in the blood, tissues can suffer hypoxia despite high oxygen saturation in the blood that does arrive.

3.6.4 Pulse-Oximeter Circuit

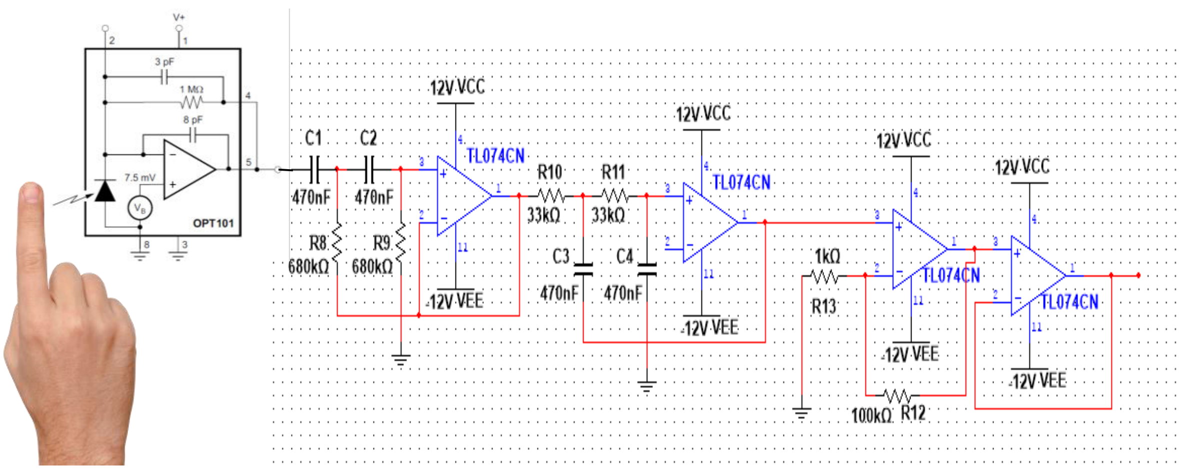


Fig 3.5: Pulse-Oximeter Circuit

3.6.5 OPT101

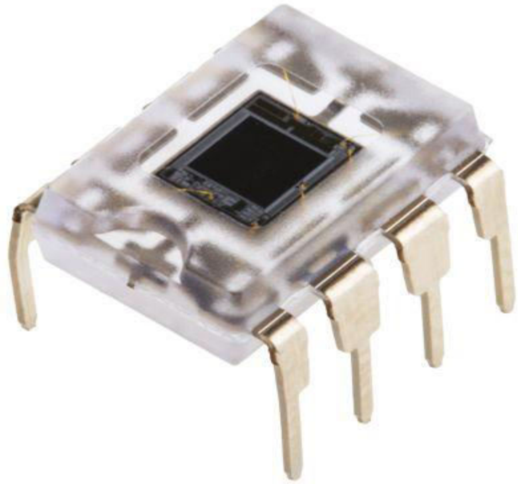


Fig 3.6: OPT101

The OPT101 is a monolithic photodiode with on-chip transimpedance amplifier. The integrated combination of photodiode and transimpedance amplifier on a single chip eliminates the problems commonly encountered in discrete designs, such as leakage current errors, noise pick-up, and gain peaking as a result of stray capacitance. Output voltage increases linearly with light intensity. The amplifier is designed for single or dual power-supply operation.

The 0.09 inch × 0.09 inch (2.29 mm × 2.29 mm) SOP photodiode operates in the photoconductive mode for excellent linearity and low dark current. The OPT101 operates from 2.7 V to 36 V supplies and quiescent current is only 120 μA. This device is available in clear plastic 8-pin PDIP, and J-lead SOP mounting. The temperature range is 0°C to 70°C.

3.6.6 Features

- Single Supply: 2.7 to 36 V
- Photodiode Size: 0.090 inch × 0.090 inch (2.29 mm × 2.29 mm)
- Internal 1-MΩ Feedback Resistor
- High Responsivity: 0.45 A/W (650 nm)
- Bandwidth: 14 kHz at RF = 1 MΩ
- Low Quiescent Current: 120 μA
- Packages: Clear Plastic 8-pin PDIP and J-Lead SOP

3.6.7 Overview

The OPT101 is a large-area photodiode integrated with an optimized operational amplifier that makes the OPT101 a small, easy-to-use, light-to-voltage device. The photodiode has a very large measurement area that collects a significant amount of light, and thus allows for high-sensitivity measurements. The photodiode has a wide spectral response with a maximum peak in the infrared spectrum, and a useable range from 300 nm to 1100 nm. The wide power-supply range of 2.7 V to 36 V makes this device useful in a variety of architectures; from all-analog circuits to data conversion base circuits. The on-chip voltage source keeps the amplifier in a good operating region, even at low light levels.

The OPT101 voltage output is the product of the photodiode current times the feedback resistor, (IDRF), plus a pedestal voltage, VB, of approximately 7.5 mV introduced for single-supply operation. Output is 7.5 mV dc with no light, and increases with increasing illumination. Photodiode current, ID, is proportional to the radiant power, or flux, (in watts) falling on the photodiode. At a wavelength of 650 nm (visible red) the photodiode responsivity, RI, is approximately 0.45 A/W. Responsivity at other wavelengths is shown in Figure 1. The internal feedback resistor is laser trimmed to 1 MΩ. Using this resistor, the output voltage responsivity, RV, is approximately 0.45 V/μW at 650-nm wavelength.

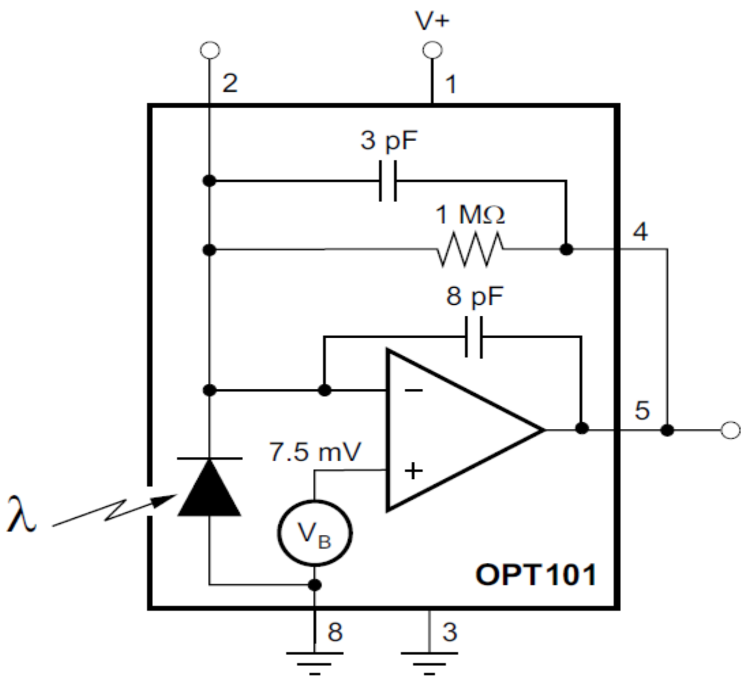


Fig 3.7: Functional Block Diagram

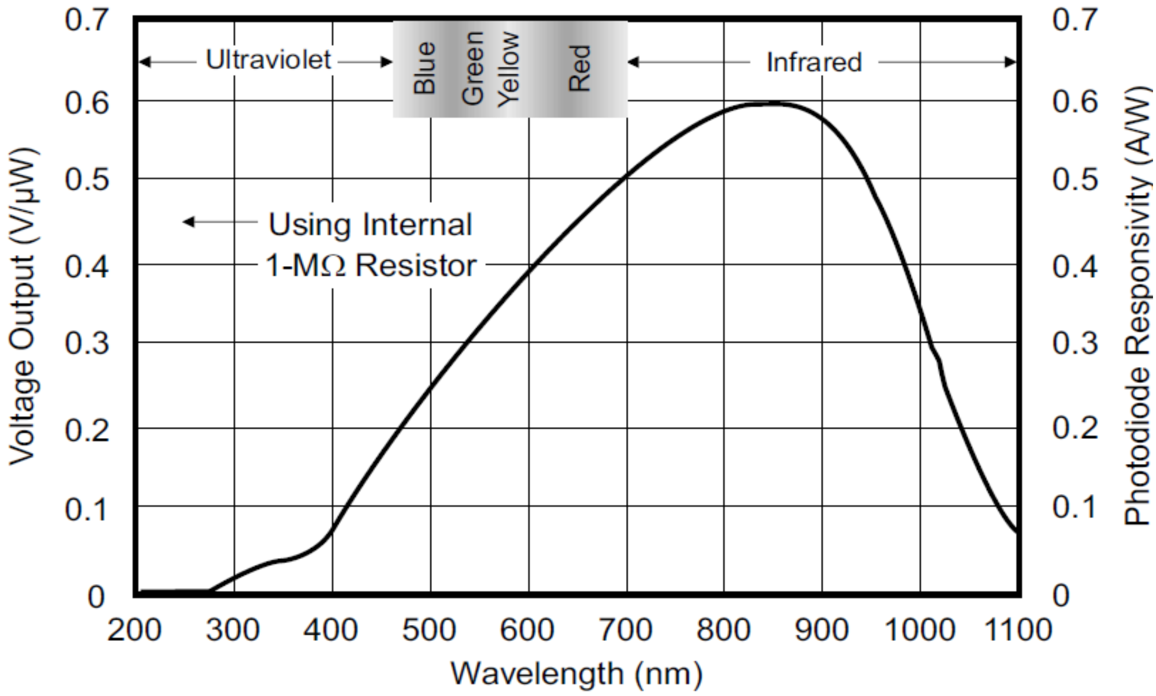


Fig 3.8: Normalized Spectral Responsivity

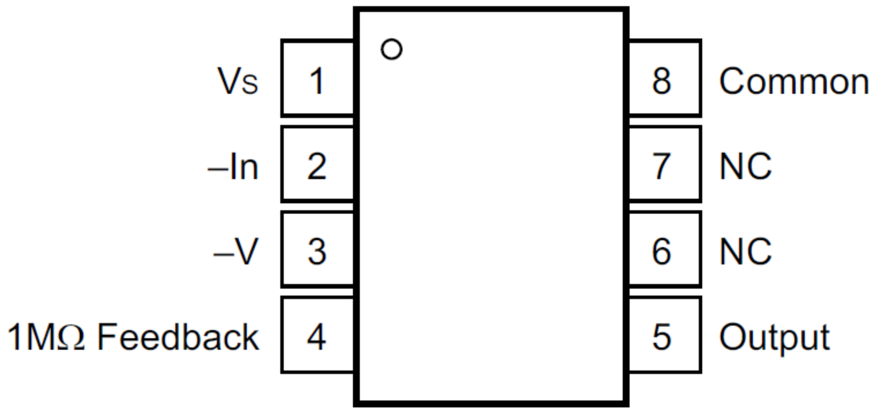


Fig 3.9: Pin Configuration

Table 3.6: Pin Functions

PIN NO.	PIN NAME	I/O	DESCRIPTION
1	V_S	Power	Power supply of device. Apply 2.7 V to 36 V relative to –V pin.
2	–In	Input	Negative input of op amp and the cathode of the photodiode. Either do not connect, or apply additional op amp feedback.
3	–V	Power	Most negative power supply. Connect to ground or a negative voltage that meets the recommended operating conditions.
4	1MΩ Feedback	Input	Connection to internal feedback network. Typically connect to Output, pin 5.
5	Output	Output	Output of device.
6	NC	—	Do not connect
7	NC	—	Do not connect
8	Common	Input	Anode of the photodiode. Typically, connect to ground.

Chapter 4

Software Requirements

4.1 Arduino IDE

An integrated development environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of a source code editor, build automation tools and a debugger. Most modern IDEs have intelligent code completion.

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

4.1.1 Writing Sketches

Fig. 4.1: Arduino IDE

Figure 4.1 shows the window when we open the Arduino IDE software. Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension .ino. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom right hand corner of the window displays the configured board and serial port. The toolbar buttons allow to verify and upload programs, create, open, and save sketches, and open the serial monitor.

The sketch involves following options:

1) Verify/Compile

This step checks the sketch for errors compiling it. It will report memory usage for code and variables in the console area.

2) Upload

Compiles and loads the binary file onto the configured board through the configured Port.

3) Include Library

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

There are many libraries in the reference. Some libraries are included with the Arduino software. Others can be downloaded from a variety of sources or through the Library Manager.

4.1.2 Tools

Tools that are needed while testing the codes are:

1) Serial Monitor

Serial monitor window initiates the exchange of data with any connected board on the currently selected Port. This usually resets the board, if the board supports Reset over serial port opening.

It displays serial data being sent from the Arduino or Genuino board (USB or serial board). To send data to the board, enter text and click on the "send" button or press enter. Choose the baud rate from the drop-down that matches the rate passed to Serial.begin in your sketch.

2) Board

Select the board as esp8266 while dumping the code to Wi-Fi module and select arduino board while connected to arduino uno.

The board selection sets the parameters like CPU speed and baud rate used when compiling and uploading sketches. Some of the board definitions differ only in the latter, so even if you've been uploading successfully with a particular selection you'll want to check it before burning the bootloader.

Arduino Software (IDE) includes the built in support for the boards in the following list, all based on the AVR Core. The Boards Manager included in the standard installation allows to add support for the growing number of new boards based on different cores like Arduino Due, Arduino Zero, Edison, Galileo and so on.

3) Port

This menu contains all the serial devices (real or virtual) on your machine. It will automatically refresh every time you open the top-level tools menu.

4.1.3 Help

Here you find easy access to a number of documents that come with the Arduino Software (IDE). You have access to Getting Started, Reference, this guide to the IDE and other documents locally, without an internet connection. The documents are a local copy of the online ones and may link back to our online website.

4.1.4 Uploading

Before uploading sketch, we need to select the correct items from the Tools like Board and Port. The boards are described as COM1 or COM2 for a serial board and COM4, COM5 and COM or higher for a USB board. To find out this, we have to look for USB serial device in the ports section of the Windows Device Manager. Once selected the correct serial port and board, press the upload button in the toolbar or select the Upload item from the Sketch menu. Current Arduino boards will reset automatically and begin the upload. On most boards, contains the RX and TX LEDs blink as the sketch is uploaded. The Arduino Software (IDE) will display a message when the upload is complete, or show an error.

When we upload a sketch, a small program that has been loaded on to the microcontroller on to the board. It allows us to upload code without using any additional hardware. The bootloader is active for a few seconds when the board resets, then it starts whichever sketch was most recently uploaded to the microcontroller. The bootloader will blink the on-board (pin 13) LED when it starts (i.e. when the board resets).

4.2 MATLAB

MATLAB (matrix laboratoty) is a multi-paradigm numerical computing environment and fourth-generation programming language. A proprietary programming language developed by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, C#, Java, Fortran and Python.

Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the MuPADsymbolic engine, allowing access to symbolic computing abilities. An additional package, Simulink, adds graphical multi-domain simulation and model-based design for dynamic and embedded systems.

4.2.1 Syntax

The MATLAB application is built around the MATLAB scripting language. Common usage of the MATLAB application involves using the Command Window as an interactive mathematical shell or executing text files containing MATLAB code.

4.2.2 Variables

Variables are defined using the assignment operator, =. MATLAB is a weakly typed programming language because types are implicitly converted. It is an inferred typed language because variables can be assigned without declaring their type, except if they are to be treated as symbolic objects, and that their type can change. Values can come from constants, from computation involving values of other variables, or from the output of a function. For example:

```
>> x = 17
x =
 17

>> x = 'hat'
x =
hat

>> y = x + 0
y =
       104         97        116

>> x = [3*4, pi/2]
x =
   12.0000    1.5708

>> y = 3*sin(x)
y =
   -1.6097    3.0000
```

4.2.3 Vectors and matrices

A simple array is defined using the colon syntax: init : increment : terminator.

```
>> array = 1:2:9
array =
 1 3 5 7 9
```

Vector defines a variable named array (or assigns a new value to an existing variable with the name array) which is an array consisting of the values 1, 3, 5, 7, and 9. That is, the array starts at 1 (the init value), increments with each step from the previous value by 2 (the increment value), and stops once it reaches (or to avoid exceeding) 9 (the terminator value).

Matrices can be defined by separating the elements of a row with blank space or comma and using a semicolon to terminate each row. The list of elements should be surrounded by square brackets: []. Parentheses: () are used to access elements and subarrays. They are also used to denote a function argument list.

```
>> A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]
A =
 16   3   2  13
  5  10  11   8
  9   6   7  12
  4  15  14   1

>> A(2,3)
ans =
 11
```

Sets of indices can be specified by expressions such as "2:4", which evaluates to [2, 3, 4]. For example, a submatrix taken from rows 2 through 4 and columns 3 through 4 can be written as:

```
>> A(2:4,3:4)
ans =
 11  8
 7  12
 14  1
```

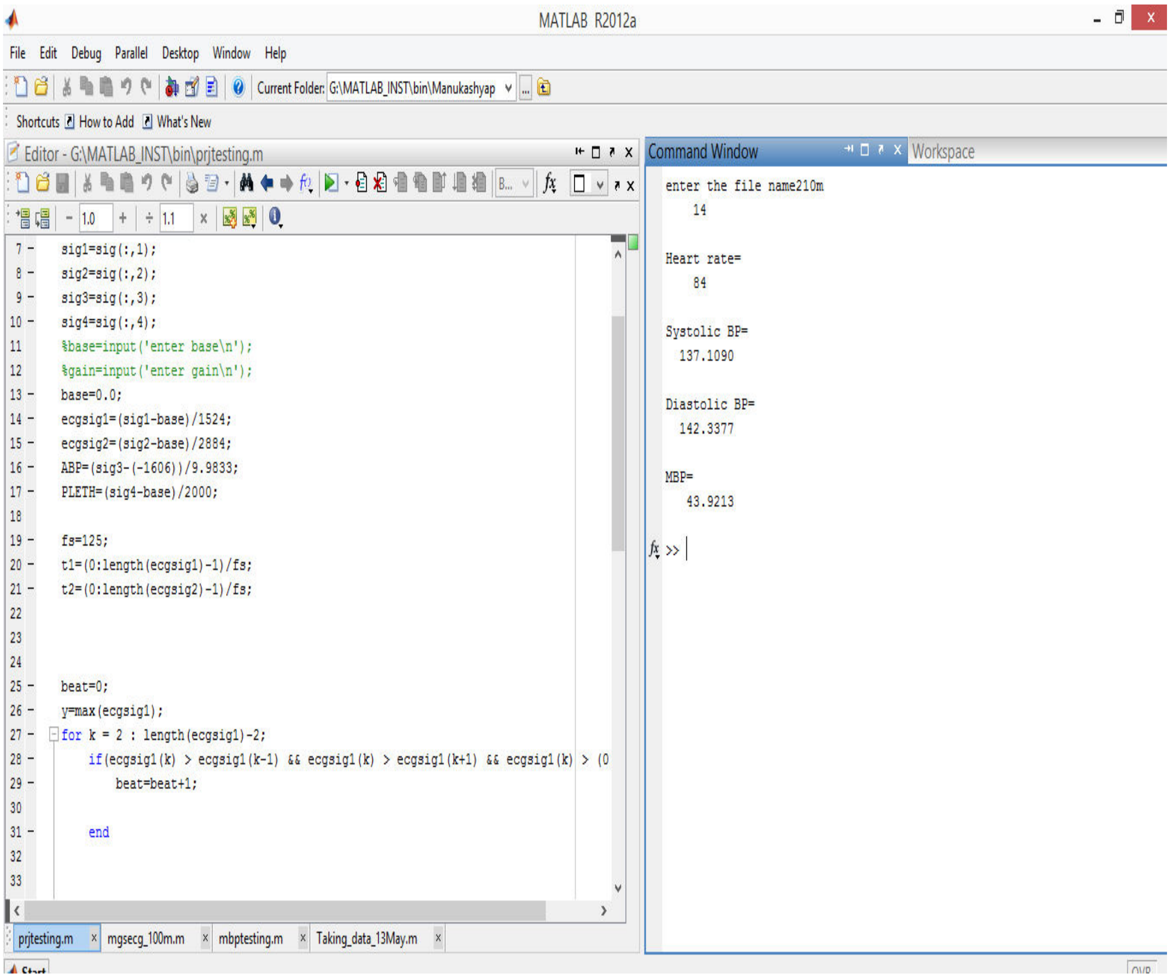
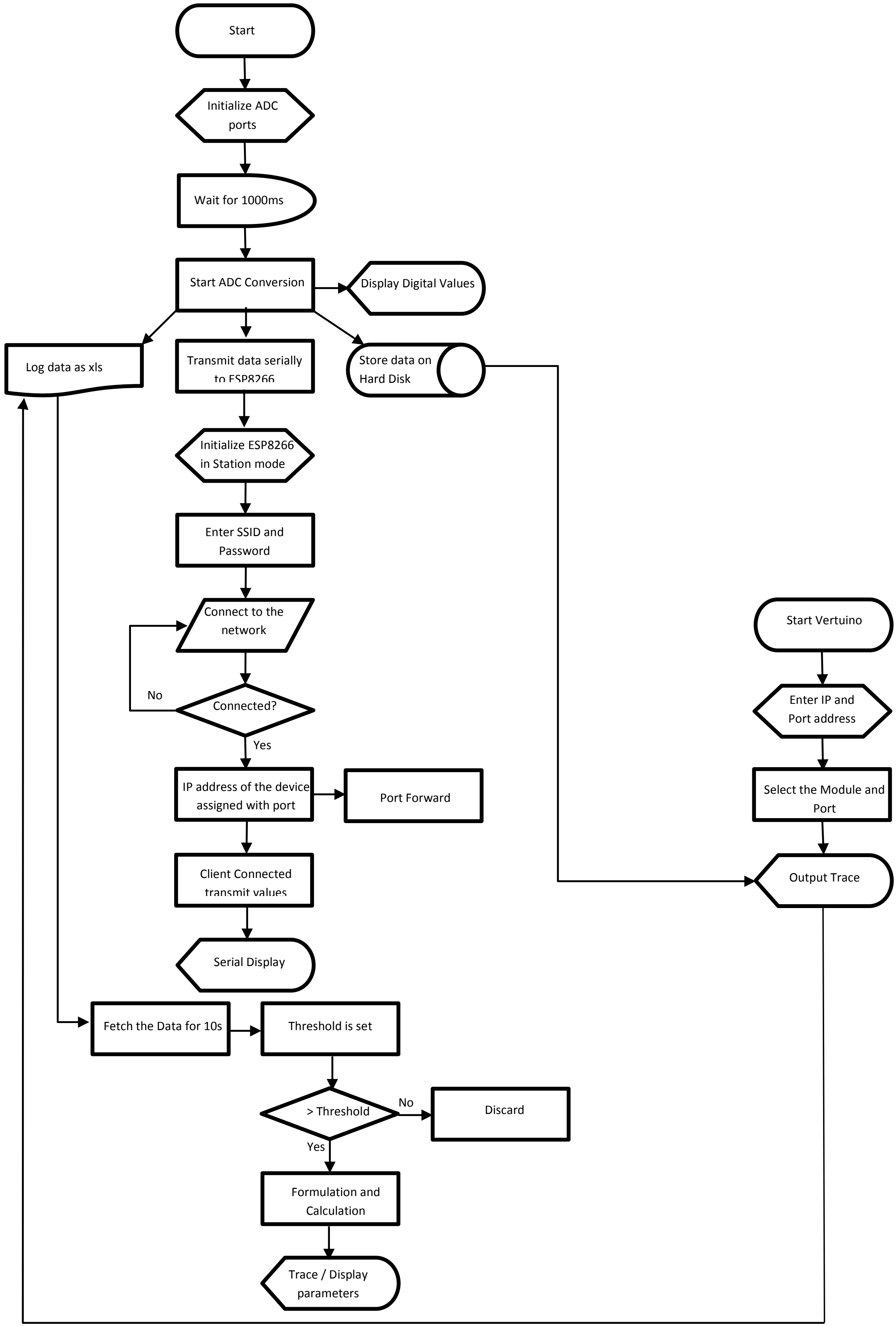


Fig. 4.2: MATLAB Editor Window and Command Window

4.3 Software Implementation


Start
Initialize ADC ports
Wait for 1000ms
Start ADC Conversion
Display Digital Values
Log data as xls
Transmit data serially to ESP8266
Store data on Hard Disk
Initialize ESP8266 in Station mode
Enter SSID and Password
Connect to the network
No
Connected?
Yes
IP address of the device assigned with port
Port Forward
Client Connected transmit values
Serial Display
Start Vertuino
Enter IP and Port address
Select the Module and Port
Output Trace
Fetch the Data for 10s
Threshold is set
> Threshold
No
Discard
Yes
Formulation and Calculation
Trace / Display parameters

Chapter 5

Result and Conclusion

5.1 Result

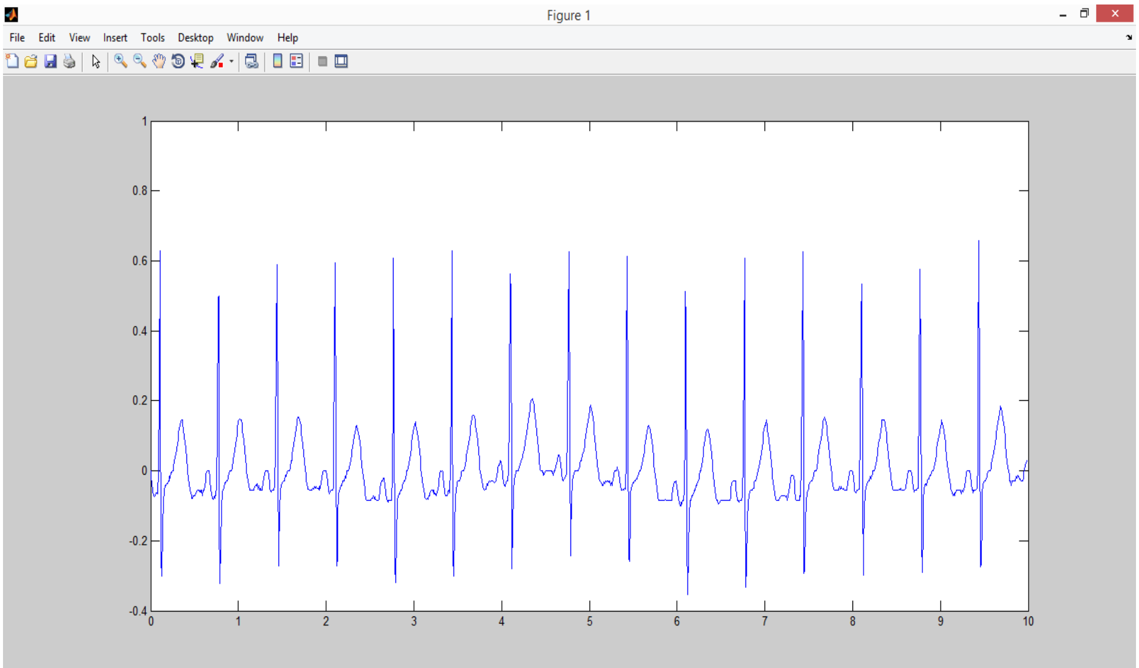


Fig 5.1: ECG Waveform

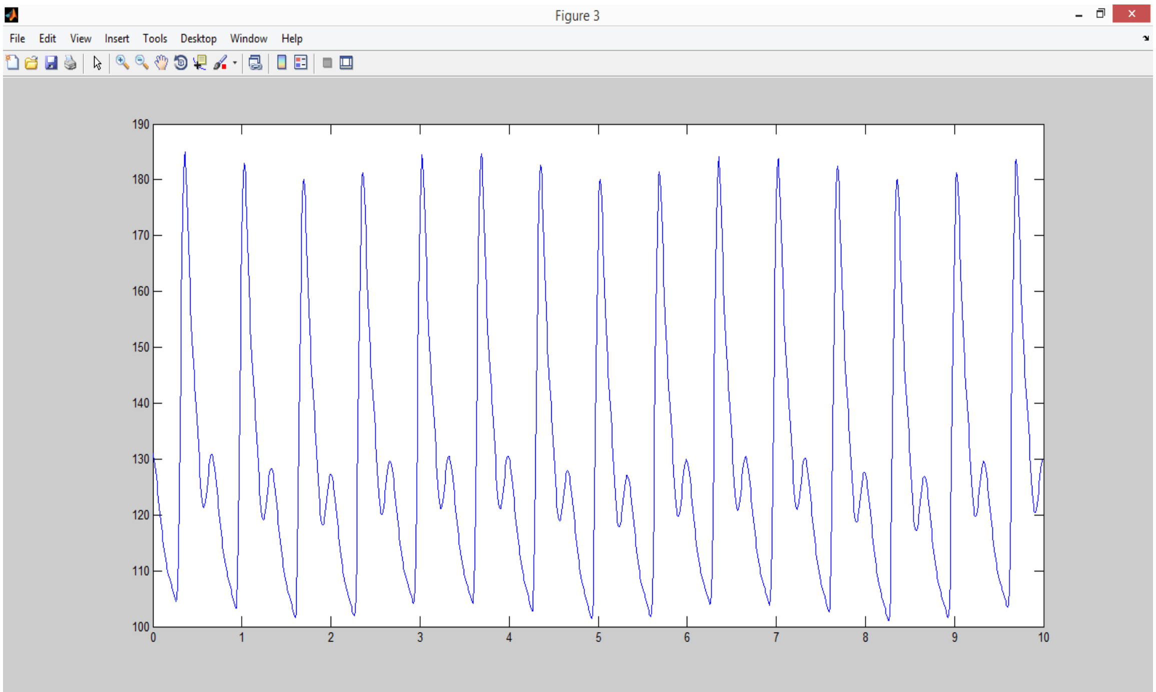


Fig 5.2: Pulse Waveform

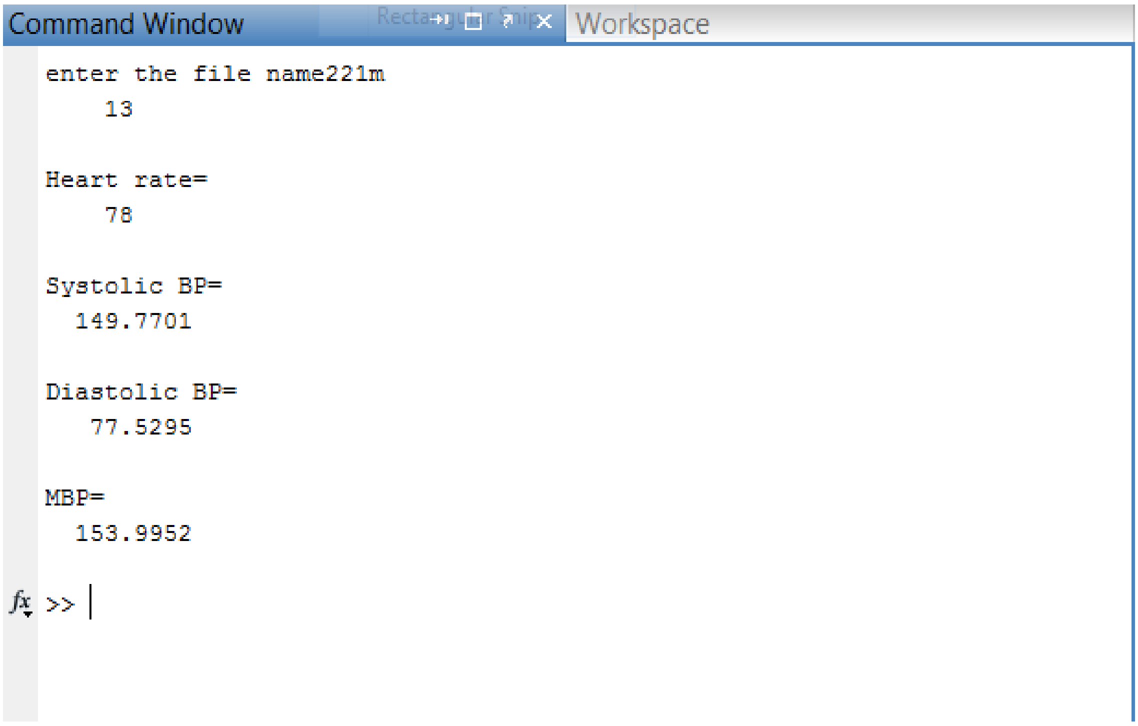


Fig 5.3: Blood Pressure and Instantaneous Heart Rate

5.2 Conclusion

Nearly 27% of the total deaths in India happen with no medical attention at the time of death, according to the 2013 ‘Civil Registration Data’ released by the Census Directorate and it will have increased by a greater percentage by now. This project has been developed with the vision to mitigate mortality due to lack of timely diagnosis. It completely changes the way how patients are diagnosed. In the present scenario the patient needs to be admitted into a healthcare facility where a professional will monitor the readings of the equipment. But in the proposed design any lay man with the most basic knowledge about, how to stick a few electrode stickers and how to clip a pulse oximeter to a finger will be able use the device right at the comfort of their homes.

5.3 Future Scope

This project has a vast scope and can be developed in numerous ways for effective and efficient use. A Real-time-tracking system can be implemented by fixing a GPS to the module, so that it can be tracked if stolen. Also, a camera can be fixed so that the medical practitioner can assist the care taker over the video call if an immediate assistance is needed. The data being transmitted can be stored if there is no deviation from the standard value instead of deleting it. The stored data can be used in future for other purposes. Security features can also be added to the system to avoid misuse of the system. The data that needs to be transmitted should be sent to a particular doctor and shouldn't be an open source and stored in the patient's profile. Also, the mobile application developed can be protected using a password or an option of login using One Time Password can be designed.

References

[1]. André Canha, Mauro Gama, Rui Monteiro, Pervasive Healthcare in Smart Homes,

[2]. Y. Ren, R. W. N. Pazzi, and A. Boukerche, “Monitoring patients via a secure and mobile healthcare system,” IEEE Wireless Communications, vol. 17, pp. 59–65, 2010

[3]. Rifat Shahriyar, Md. Faizul Bari, Gourab Kundu, Sheikh Iqbal Ahamed, and Md. Mostofa Akbar, “Intelligent Mobile Health Monitoring System (IMHMS)”

[4]. Enable Pervasive Healthcare through Continuous Remote Health Monitoring Xiaohui Liang, Xu Liy, Mrinmoy Barua, Le Chen, Rongxing Lu, Xuemin (Sherman) Shen, and Henry Y. Luoz

[5]. Davies JI, Struthers AD (2003) Pulse wave analysis and pulse wave velocity: a critical review of their strengths and weaknesses. J Hypertens 21:463–472

[6]. Moha'meed O. Al-Jaafreh and Adel A. Al-Jumaily “New model to estimate Mean Blood Pressure by Heart Rate with Stroke Volume changing influence” IEEE EMBS Annual International Conference.

[7]. M.Satyanarayanan, Pervasive Computing Vision and Challenges, IEEE Personal Communications,2001

[8]. Andreas Butz, Antonio Kr¨uger, User-centered development of a pervasive healthcare application, " Pervasive Health Conference and Workshops, 2006 , vol., no., pp.1-8, Nov. 29 2006-Dec. 1 2006

[9]. Jurgen Bohn, Felix G¨artner, and Harald Vogt, Dependability Issues of Pervasive Computing in a Healthcare Environment, Proceedings of the First International Conference on Security in Pervasive Computing, volume 2082 of Lecture Notes in Computer Science,2003, pages53-70, Springer-Verlag

[10]. Elham Rastegari, Amirmasood Rahmani, Saeed Setayeshi, Pervasive Computing in Healthcare Systems, World Academy of Science, Engineering and Technology 59, 2011

[11]. Xiaohui Liang, Xu Li, Mrinmoy Barua, Le Chen, Rongxing Lu, Enable Pervasive Healthcare through Continuous Remote Health Monitoring, IEEE Wireless Communications, vol. 19,no.6, 2012, pages 10-18.