IOT BASED BIOMEDICAL HEALTH MONITORING SYSTEM

MINOR PROJECT REPORT

A Report submitted in partial fulfilment of the requirement for the award of degree of

Bachelor of Technology

In

Electrical and Electronics Engineering

Under the Supervision of

Ms. Mamta Tholia

By

VIKRANT THAKUR (35515004915)

NAVEEN MEENA (02115004915)

CHETAN ATRAI (41915004915)

AMIT ARYAN (41015004915)

SAHIL VERMA (40815004915)

MAHARAJA SURAJMAL INSTITUTE OF TECHNOLOGY

C-4, Janakpuri, New Delhi-58

Affiliated to Guru Gobind Singh Indraprastha University, Delhi

DEC, 2018

DECLARATION

We, student of B.Tech (Electrical and Electronics Engineering) hereby declare that the project work done on "IOT BASED BIOMEDIACL HEALTH MONITORING SYSTEM" submitted to Maharaja Surajmal Institute of Technology, Janakpuri Delhi in partial fulfilment of the requirement for the award of degree of Bachelor of Technology comprises of our original work and has not been submitted anywhere else for any other degree to the best of our knowledge.

Vikrant thakur (35515004915)

Naveen meena (02115004915)

Chetan atrai (41915004915)

Amit Aryan (41015004915)

Sahil verma (40815004915)

CERTIFICATE

This is to certify that the project work done on "IOT BASED BIOMEDIACL HEALTH MONITORING SYSTEM" submitted to Maharaja Surajmal Institute of Technology, Janakpuri Delhi by "Vikrant thakur, Naveen meena, Chetan atrai, Amit aryan, Sahil verma" in partial fulfillment of the requirement for the award of degree of Bachelor of Technology, is a bonafide work carried out by him/her under my supervision and guidance. This project work comprises of original work and has not been submitted anywhere else for any other degree to the best of my knowledge.

Signature of Supervisor	Signature of HOD
(Project Supervisor)	(HOD, EEE)

ACKNOWLEDGEMENT

Team effort together with precious words of encouragement and guidance makes daunting tasks achievable. It is a pleasure to acknowledge the direct and implied help we have received at various stages in the task of developing the project. It would not have been possible to develop such a project without the furtherance on part of numerous individuals. We find it impossible to express our thanks to each one of them in words, for it seems too trivial when compare to the profound encouragement that they extended to us.

We are grateful to Dr.Meena Tushir, HOD, EEE, for having given us opportunity to do this project, which was of great interest to us.

Our sincere thanks to Ms. Mamta Tholia, Assistant Professor, EEE for believing in us and providing motivation all through. Without her guidance this project would not be such a success.

At last we thank the almighty, who had given the strength to complete this project on time.

Finally we would like to thank our project head Ms. Rakhi kamra our parents, all friends, and well wishers for their valuable help and encouragement throughout the project.

Vikrant thakur (35515004915)

Naveen meena (02115004915)

Chetan atrai (41915004915)

Amit Aryan (41015004915)

Sahil verma (40815004915)

Table of Contents

Abstract

Detection of pulse in a human body plays an important role for Health Monitoring and treatment of diseases. In this work we have streamlined our efforts towards obtaining a solution for long-term heart rate monitoring system that can be used at home as well as medical establishments. . In the proposed system, the patient's physiological signals are acquired by the sensors attached on the patient body, and are then transmitted to the remote base-station and used for storing and analysing data. The controller that is used to process the signal is 32bit low power high advanced architecture-based controller which is manufactured by STMicroelectronics. It is interfaced with different inbuilt features and data that is been recorded is processed or analysis through software's like MATLAB or Python. The data is further uploaded to IOT Based platform think we speak or server where it further analyses the data graphically.

Vikrant thakur (35515004915)

Naveen meena (02115004915)

Chetan atrai (41915004915)

Amit Aryan (41015004915)

Sahil verma (40815004915)

List of figures

FIGURE NO. **PAGE NO.**

List of tables:

CHAPTER-1

1.1 Introduction:

Body sensor network systems can help people by providing healthcare services such as medical monitoring, memory enhancement, medical data access, and communication with the healthcare provider in emergency situations. Continuous health monitoring with wearable or clothing embedded transducers and implantable body sensor networks will increase detection of emergency conditions in at risk patients. Not only the patient, but also their families will benefit from these. Also, these systems provide useful methods to remotely acquire and monitor the physiological signals without the need of interruption of the patient's normal life, thus improving life quality.

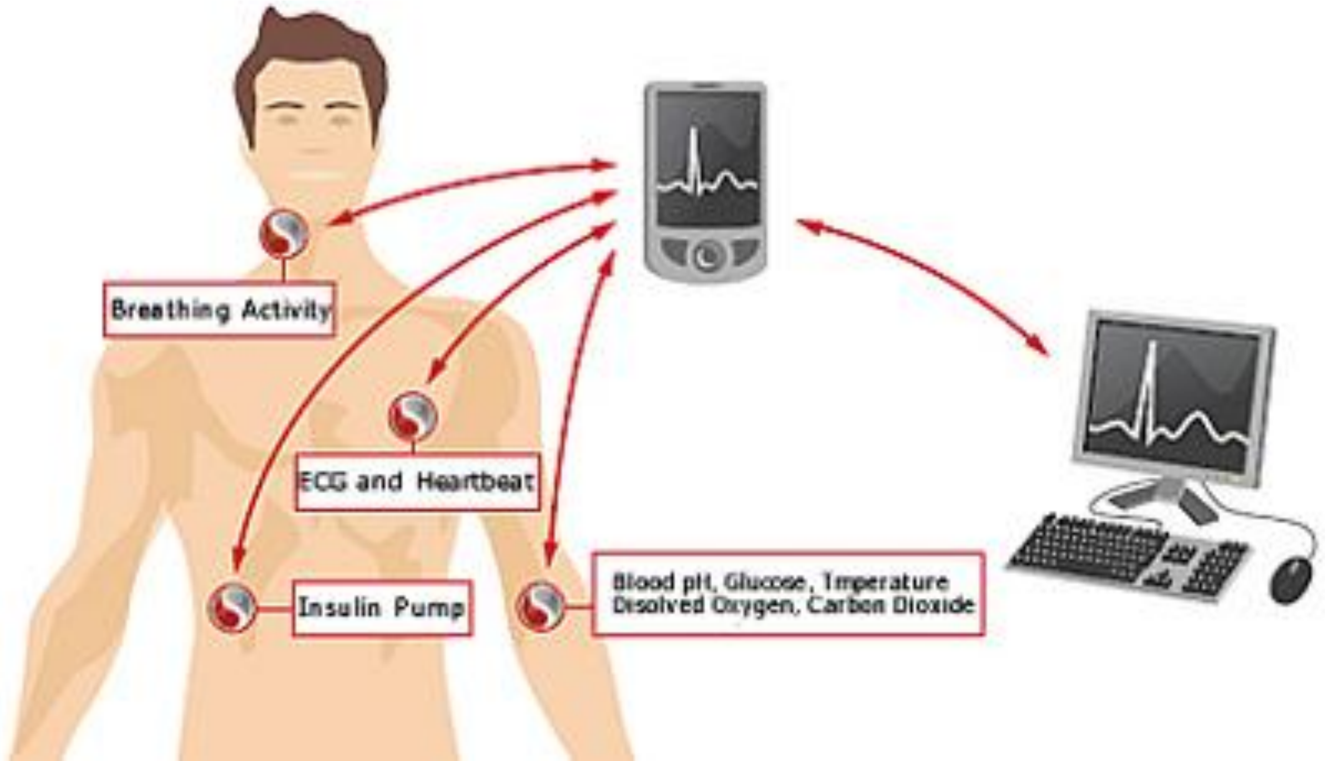

Figure 1

In a hospital health care monitoring system, it is necessary to constantly monitor the patient's physiological parameters. For example, a pregnant woman parameter such as blood pressure (BP) and heart rate of the woman and heart rate and movements of fatal to control their health condition. This project presents a monitoring system that has the capability to monitor physiological parameters such as temperature, heartbeat, ECG etc. In the proposed system, sensors are interfaced to the embedded world which constantly records data and the data is fetched through server and analyzed through graphical representation. The attached sensors on patient's body form a wireless body sensor network (WBSN) and they are able to sense the heart rate, blood pressure and so on. This system can detect the abnormal conditions, issue an alarm to the patient and present a graphical view to the physician.

Creating technologies for monitoring and creating body sensor networks has been the most active and successful research theme within pervasive healthcare – so far. One strand of work is dedicated to achieving reliable monitoring of health signs like blood pressure, ECG, heart rate, skin conductivity, blood sugar, and similar. The main challenge has been to design and develop reliable yet non-intrusive, wearable sensors, which can be used by us. The goal has been to create a platform for continuous monitoring, because substantial clinical evidence indicates that continuous monitoring of certain vital signs can work as early detector of different chronic diseases like hypertension, congestive heart failure, diabetes, dementia, and epilepsy. A related strand is to make sure that this sensor and monitoring technology works together in a distributed infrastructure. This research relates to research within general sensor networks, and has been dedicated to the design and development of health-specific body sensor networks. Resilience, fail-over, network topology, wireless communication, protocols, and real-time data management are important issues in this strand of research.

Body sensor network research seeks to move existing sampling technologies and their use from the laboratories inside hospitals out into the hands of the patients. In addition, a third strand of project seeks to devise technologies and approaches for monitoring and recognizing higher-order behavioral traits. For example, detecting activities of daily living (ADL) based on sensor networks and machine learning approaches has been the topic of several project groups. The goa is to reason about the functioning of an individual, and use this information for early warnings, safety, prevention, and assistance. For example, detecting early signs of dementia, monitoring falls in the home of elderly people, and helping people with dementia to wash hands or remember conversations. The state of the art is that the first prototypes have been developed, which can be used in larger clinical trials. For example, the My Heart technology is now being tried out with real patients and the Intel Mobile Sensing Platform (Intel MSP) is being used as a health monitoring infrastructure in different academic research projects. Also, the first types of products are being commercialized – even though many of these are advertised as “wellness technologies” and not as health monitoring devices. One example is the Pulsar monitoring equipment.

1.2 PROPOSED PROJECT:

Health is one of the global challenges for humanity. According to the constitutions of World Health Organization (WHO) the highest attainable standard of health is a fundamental right for an individual in hospitals there are provisions for continuous monitoring of patients. Their ECGs, heartbeat…are continuously monitored. There is no provision to check the parameters when they return to home. And hence there is a chance that the disease may return again. Patient's data (temperature, heart rate, ECG, position) will be frequently measured and sent to server. Period of sending (say every 1 min) can be set. Monitoring person learns patient specific threshold. Say the regular body temperature of a patient is 37□c whereas one person feels feverish if his body temperature is 37.0˚c. By employing an averaging technique over a relatively long time, Observer can learn these thresholds for patients. Using Android Application in doctor’s smart phone, doctor can view his patient’s health status. When any of the parameter goes beyond the threshold value he will get an alert notification. Using controller Application in patient’s or his caretaker’s smart phone the patient can view his health status. Early detection and diagnosis of potentially fatal physiological conditions such as heart attack require continuous monitoring of patients health following transfer from hospital to home. Studies have shown that 30% of patients with a discharge diagnosis of heart failure are readmitted at least once within 90 days with readmission rates ranging from 25 to 54% within 3 – 6 months. In response to these types of needs, health monitoring systems are being proposed as a low cost solution. Such a system consists of physiological data that stores, process and communicate through a local manner such as smart phones, personal computers. Such systems should satisfy strict safety, security, reliability, and long-term real-time operation requirement. In the proposed system we present a health monitoring system that uses the sensors for collecting data from patients, intelligently predicts patient's health status and provides feedback to doctors through their mobile devices having android application. The patients will participate in the health care process by their mobile devices and thus can access their health information from anywhere any time. Today Internet has become one of the important parts of our daily life. It has changed how people live, work, play and learn. Internet serves for many purpose educations, finance, Business, Industries, Entertainment, Social Networking, Shopping, E-Commerce etc. The next new mega trend of Internet is Internet of Things (IOT). Visualizing a world where several objects can sense, communicate and share information over a Private Internet Protocol (IP) or Public Networks. The interconnected objects collect the data at regular intervals, analyse and used to initiate required action, providing an intelligent network for analyzing, planning and decision making. This is the world of the Internet of Things (IOT).The IOT is generally considered as connecting objects to the Internet and using that connection for control of those objects or remote monitoring. But this definition was referred only to part of IOT evolution considering the machine to machine market today. But actual definition of IOT is creating a brilliant, invisible network which can be sensed, controlled and programmed. The products developed based on IOT include embedded technology which allows them to exchange information, with each other or the Internet. Since these devices come online, they provide better life style, create safer and more engaged communities and revolutionized healthcare. The entire concept of IOT stands on sensors, gateway and wireless network which enable users to communicate and access the application/information.

II. PROBLEM DEFINITION:

In today's social insurance framework for patients who stays in home during post operational days checking is done either via overseer/ medical caretaker. Ceaseless observing may not be accomplished by this system, on the grounds that anything can change in wellbeing parameter inside of part of seconds and amid that time if guardian/attendant is not in the premises causes more noteworthy harm. So with this innovation created period where web administers the world gives a thought to add to another keen health awareness framework where time to time constant checking of the patient is accomplished.

III. RELATED WORK:

The aim of the project is to automate the life easier in terms of medical advancements. In the proposed system, the patient's physiological signals are acquired by the sensors attached on the patient body, and are then transmitted to the remote base-station and used for storing and analysing data. The controller that is used to processed the signal is 32bit low power high advanced architecture-based controller which is manufactured by STMicroelectronics. It is interfaced with different inbuilt features and data that is been recorded is processed or analysis through software's like MATLAB or Python. The data is further uploaded to IOT Based platform think we speak or server where it further analyses the data graphically.

IV. SYSTEM AND OVERVIEW:

The Block diagram of the proposed system is shown in Fig2. The sensors Temperature, ECG, Heartbeat and ECG are connected to the STM discovery board. The values from the Microcontroller is given to the Web Server using Ethernet Shield. The parameter values can be viewed by the web server Application installed in doctors and patient's laptops.

V. CIRCUIT DIAGRAM:

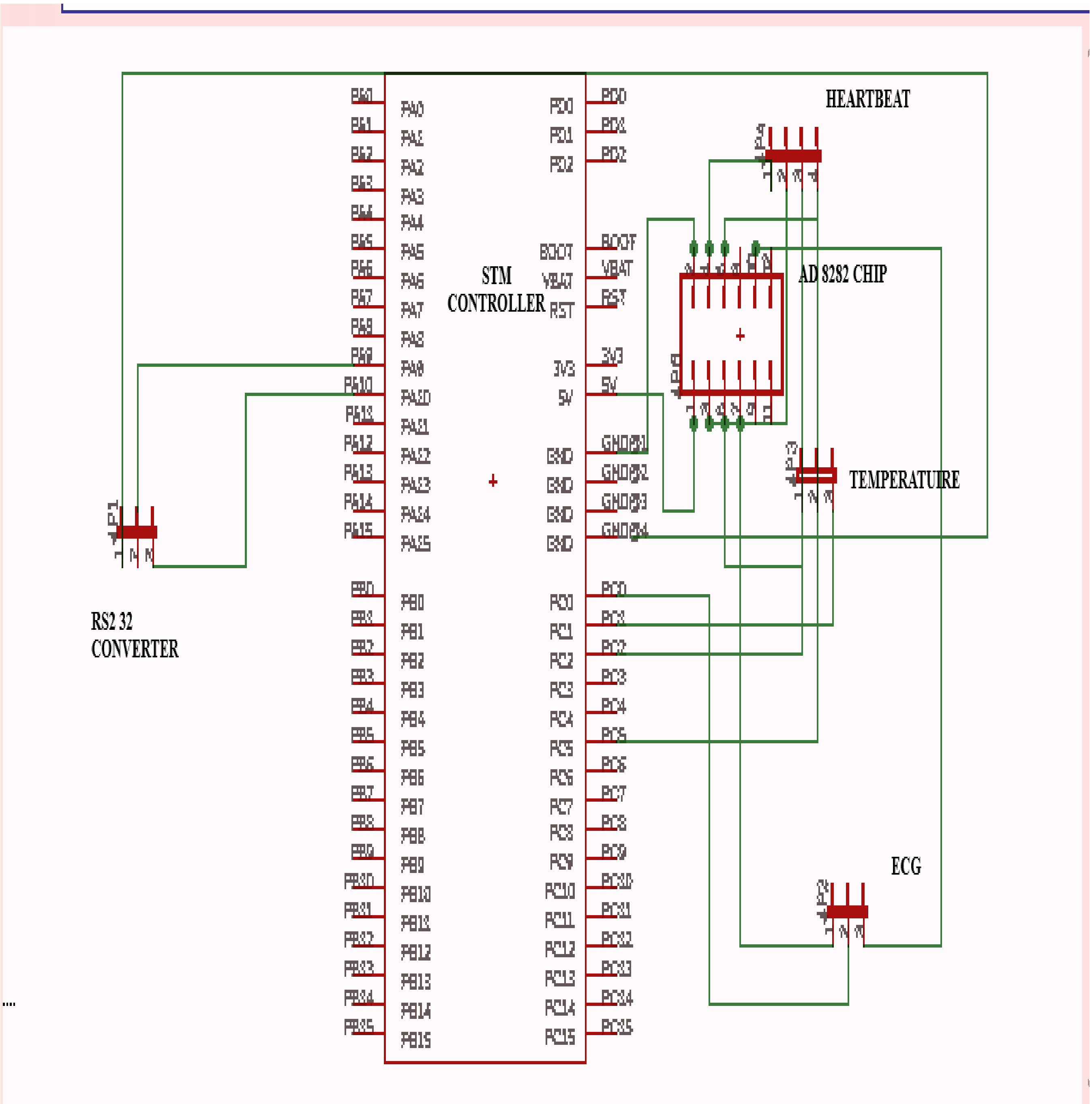

Figure 2

1.3 BLOCK DIAGRAM OF PROJECT:

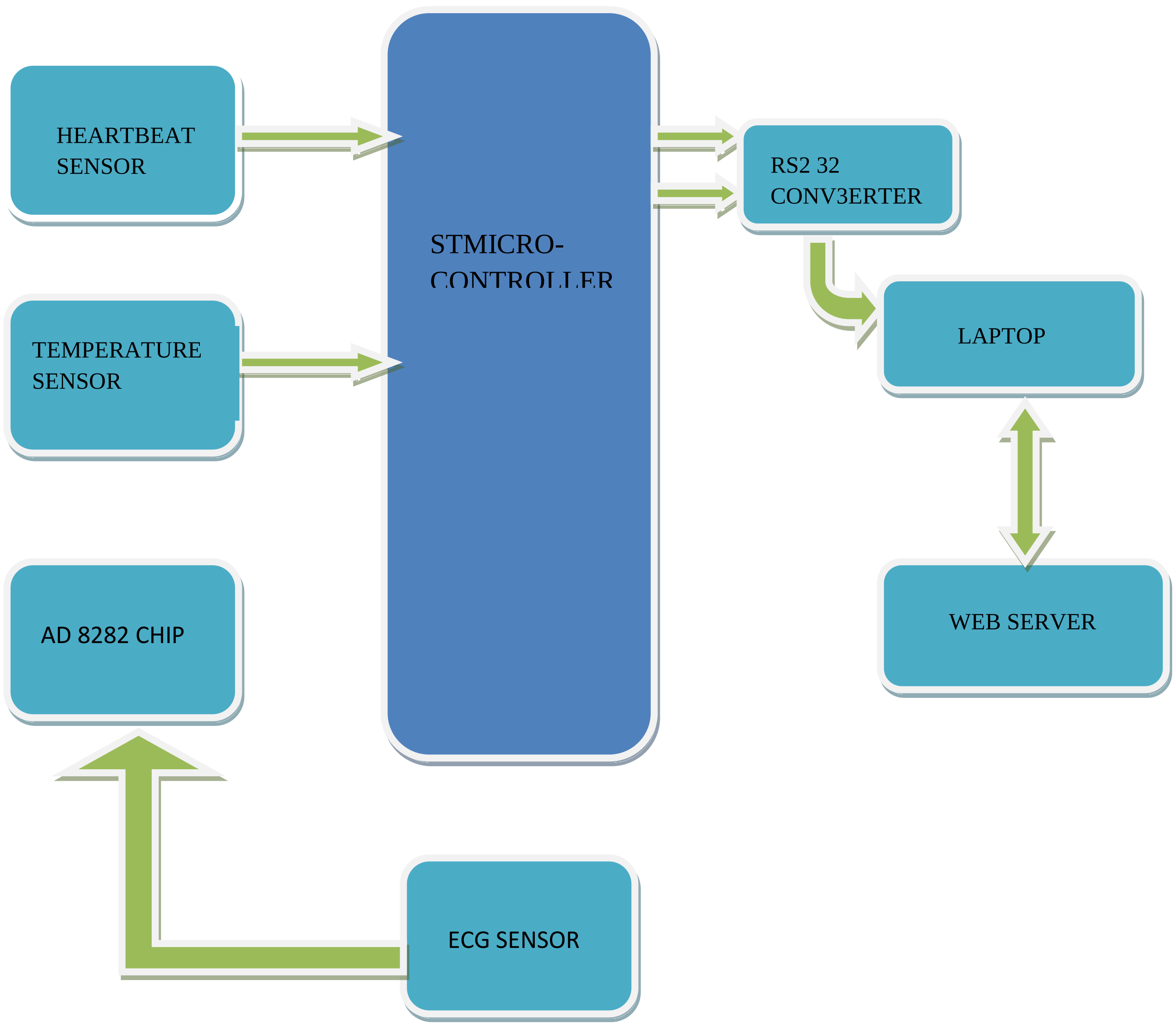

Figure 3

CHAPTER -2

2.1 PHYSICAL PARAMETERS:

EMBEDDDED WORLD: IOT WORLD:

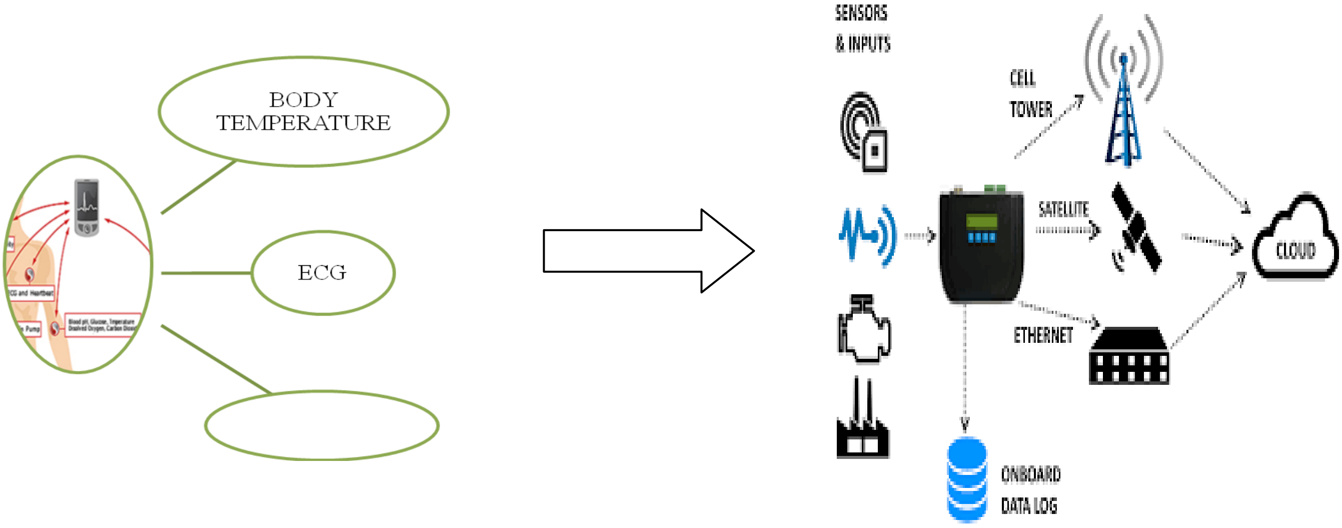

Figure 4

2.2 PROJECT COMPONENTS:

2.2.1 Temperature sensor LM355.

The LM335 temperature sensor is an easy to use, cost-effective sensor with decent accuracy (around +/- 3 degrees C calibrated). The sensor is essentially a zener diode whose reverse breakdown voltage is proportional to absolute temperature.

Since the sensor is a zener diode, a bias current must be established in order to use the device. The spec sheet states that the diode should be biased between 400 uA and 5 mA; we'll bias it at 2 mA. It is important to note that self-heating can be a significant factor, which is why I'm not choosing a higher bias current. The bias circuit is as follows:

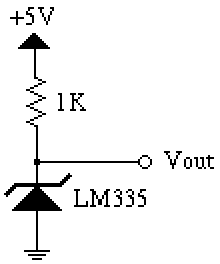

Figure 5

The temperature sensor's voltage output is related to absolute temperature by the following equation: Vout = VoutT0 * T / T0, where T0 is the known reference temperature where VoutT0 was measured. The nominal VoutT0 is equal to T0 * 10 mV/K. So, at 25 C, VoutT0 is nominally 298 K * 10 mV/K = 2.98 V (to be really accurate, we'd need a reference temperature and a voltmeter, but nominal values are OK for our purposes). Thus, the voltage dropped between +5 and the diode is 5V - 2.98V = 2.02V. In order to get 2 mA bias current, we need a 1 K resistor for R1.

Features of LM335 Temperature Sensor:

- Calibrated directly in ˚ Celsius (Centigrade)
- Rated for full l −55˚ to +150˚C range
- Suitable for remote applications.
- Low cost due to wafer-level trimming.

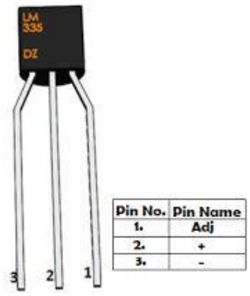

Figure 6

- Operates from 4 to 30 volts
- Low self-heating,
- ±1/4˚C of typical nonlinearity

Operation of LM335:

- The LM335 can be connected easily in the same way as other integrated circuit temperature sensors. It can be stuck or established to a surface and its temperature will be within around the range of 0.01˚C of the surface temperature.
- This presumes that the ambient air temperature is just about the same as the surface temperature; if the air temperature were much higher or lower than the surface temperature, the actual temperature of the LM335 die would be at an intermediate temperature between the surface temperature and the air temperature.
- The temperature sensors have well known applications in environmental and process control and also in test, measurement and communications. A digital temperature is a sensor, which provides 9-bit temperature readings. Digital temperature sensors offer excellent precise accuracy, these are designed to read from 0°C to 70°C and it is possible to achieve ±0.5°C accuracy. These sensors completely aligned with digital temperature readings in degree Celsius.

OUTPUT WAVEFORM:

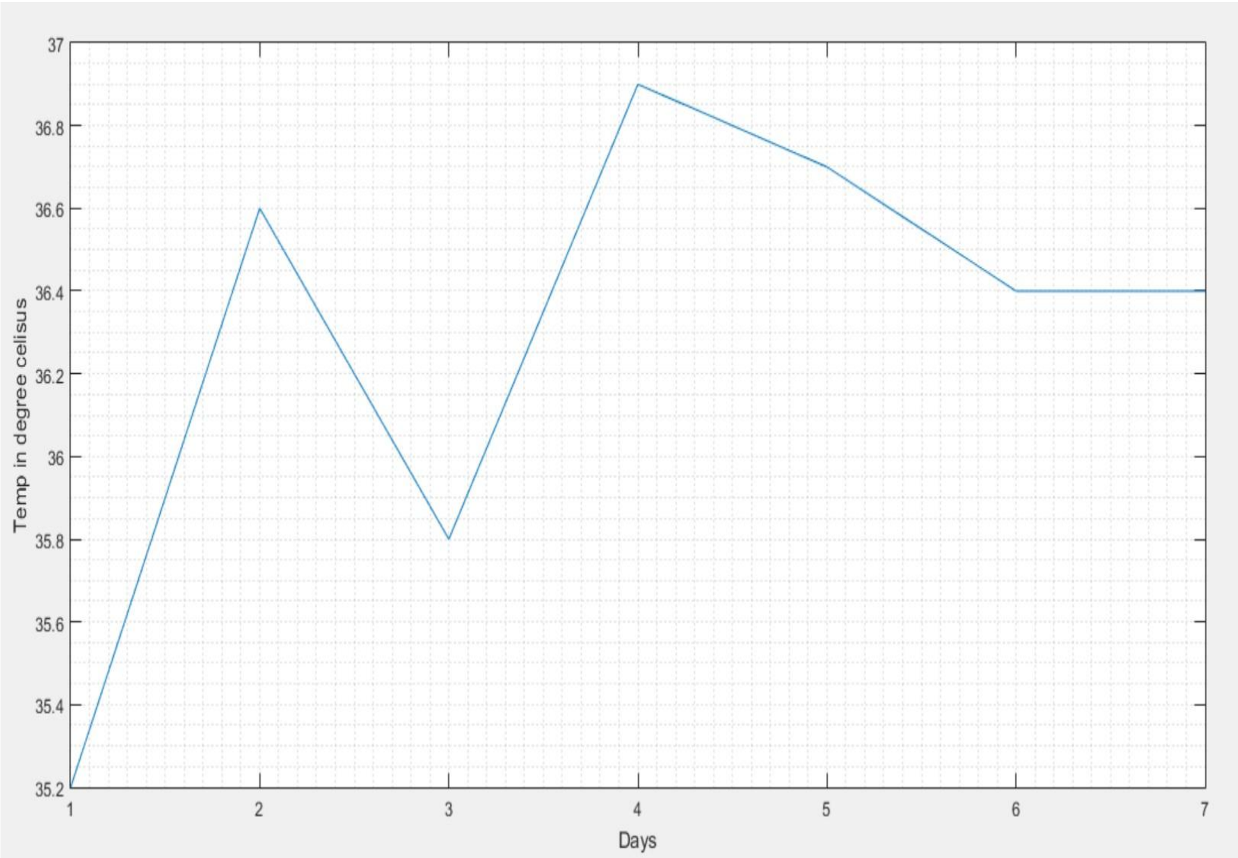

Figure 8

2.2.2 HEARTBEAT SENSOR:

Heart beat sensor is designed to give digital output of heat beat when a finger is placed inside it. This digital output can be connected to controller directly to measure the Beats per Minute (BPM) rate. It works on the principle of light modulation by blood flow through finger each pulse. IC LM358 is used for this sensor. Its dual low power operational amplifier consists of a super bright red LED and light detector. One will act as amplifiers and another will be used as comparator. LED needs to be super bright as the light must pass through finger and detected at other end. When heart pumps a pulse of blood through blood vessels, finger becomes slightly more opaque so less light reach at the detector. With each heart pulse, the detector signal varies which is converted to electrical pulse.

What do you mean by Heartbeat?

A person's heartbeat is the sound of the valves in his/her's heart contracting or expanding as they force blood from one region to another. The number of times the heart beats per minute (BPM), is the heart beat rate and the beat of the heart that can be felt in any artery that lies close to the skin is the pulse.

Two Ways to Measure a Heartbeat:

- **Manual Way**: Heart beat can be checked manually by checking one's pulses at two locations- wrist (the **radial pulse**) and the neck (**carotid pulse**). The procedure is to place the two fingers (index and middle finger) on the wrist (or neck below the windpipe) and count the number of pulses for 30 seconds and then multiplying that number by 2 to get the heart beat rate. However pressure should be applied minimum and also fingers should be moved up and down till the pulse is felt.
- **Using a sensor**: Heart Beat can be measured based on optical power variation as light is scattered or absorbed during its path through the blood as the heart beat changes.

Principle of Heartbeat Sensor:

The heartbeat sensor is based on the principle of photo phlethysmography. It measures the change in volume of blood through any organ of the body which causes a change in the light intensity through that organ (a vascular region). In case of applications where heart pulse rate is to be monitored, the timing of the pulses is more important. The flow of blood volume is decided by the rate of heart pulses and since light is absorbed by blood, the signal pulses are equivalent to the heart beat pulses.

There are two types of photophlethysmography:

Transmission: Light emitted from the light emitting device is transmitted through any vascular region of the body like earlobe and received by the detector.

Reflection: Light emitted from the light emitting device is reflected by the regions.

Figure 9

Working of a Heartbeat Sensor.

The basic heartbeat sensor consists of a light emitting diode and a detector like a light detecting resistor or a photodiode. The heart beat pulses causes a variation in the flow of blood to different regions of the body. When a tissue is illuminated with the light source, i.e. light emitted by the led, it either reflects (a finger tissue) or transmits the light (earlobe). Some of the light is absorbed by the blood and the transmitted or the reflected light is received by the light detector. The amount of light absorbed depends on the blood volume in that tissue. The detector output is in form of electrical signal and is proportional to the heart beat rate.

This signal is actually a DC signal relating to the tissues and the blood volume and the AC component synchronous with the heart beat and caused by pulsatile changes in arterial blood volume is superimposed on the DC signal. Thus the major requirement is to isolate that AC component as it is of prime importance.

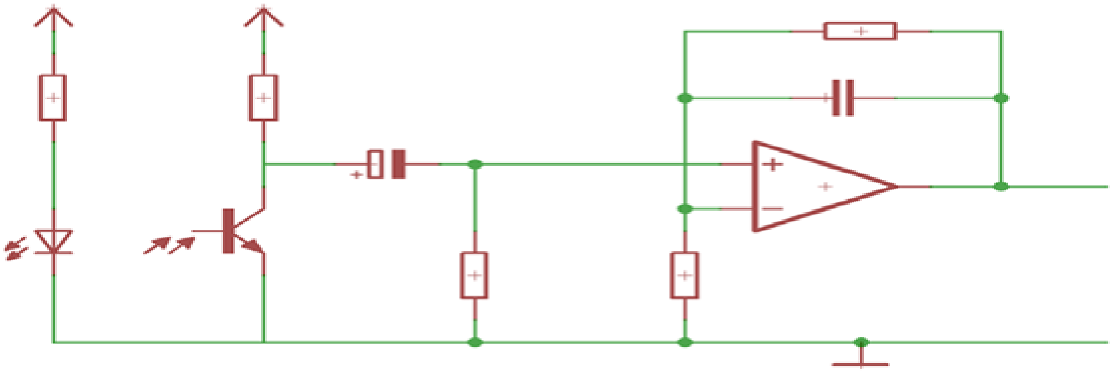

Figure 10

o achieve the task of getting the AC signal, the output from the detector is first filtered using a 2 stage HP-LP circuit and is then converted to digital pulses using a comparator circuit or using simple ADC. The digital pulses are given to a microcontroller for calculating the heat beat rate, given by the formula-

BPM (Beats per minute) = 60*f

Where f is the pulse frequency.

AVERAGE HEART-RATE	
AGE	**AVERAGE HEART RATE**
NEW BORN	**140**
7 YEAR	**85-90**
14 YEAR	**80-85**
ADULTS	**70-80**

TABLE-1

Waveform of heart-rate:

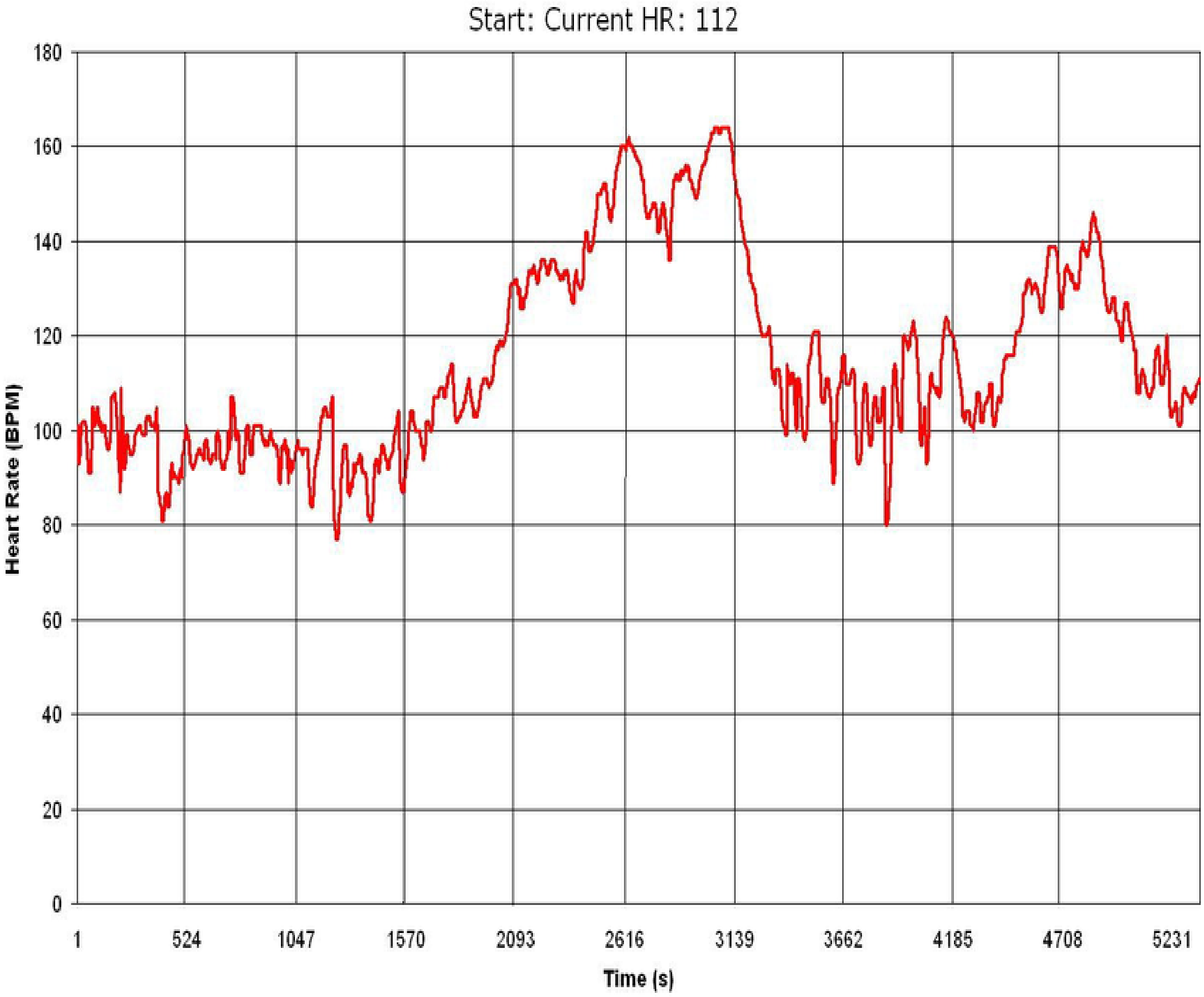

Figure 11

2.2.3 ECG SENSOR:

ECG is primarily a tool for examination of cardiac diseases. An ECG sensing device commonly consists of a group of electrodes to detect electrical events of a heart. The ECG is the electrical manifestation of the contractile activity of the heart, and can be recorded fairly easily with surface electrodes on the limbs or chest. The rhythm of the heart in terms of beats per minute (BPM) may be easily estimated by counting the readily identifiable waves .The amplifier takes the input from 3 electrodes which are connected to the patient.

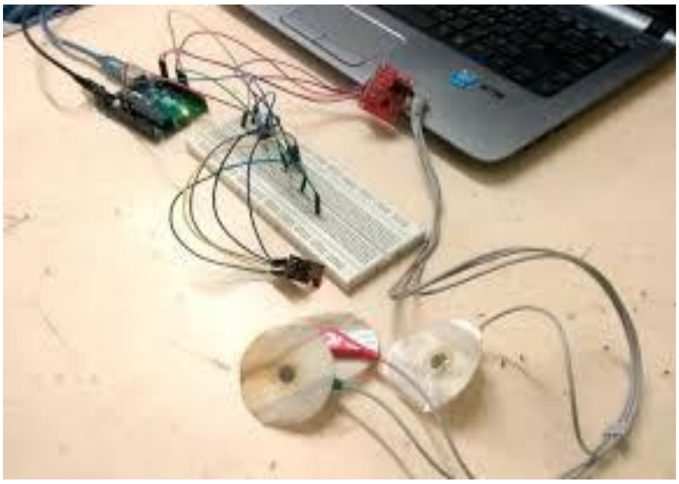

Figure 12

Physiology and function of the heart:

Before we dig deeper into the fundamentals of ECG, let's briefly recap on heart physiology and function:

- The heart has four chambers. The upper two chambers (left/right atria) are entry-points into the heart, while the lower two chambers (left/right ventricles) are contraction chambers sending blood through the circulation. The circulation is split into a "loop" through the lungs (pulmonary) and another "loop" through the body (systemic).
- The cardiac cycle refers to a complete heartbeat from its generation to the beginning of the next beat, comprising several stages of filling and emptying of the chambers. The frequency of the cardiac cycle is reflected as heart rate (beats per minute, bpm).

- The heart operates automatically – it is self-exciting (other muscles in the body require nervous stimuli for excitation). The rhythmic contractions of the heart occur spontaneously, but are sensitive to nervous or hormonal influences, particularly to sympathetic (arousing) and parasympathetic (decelerating) activity.

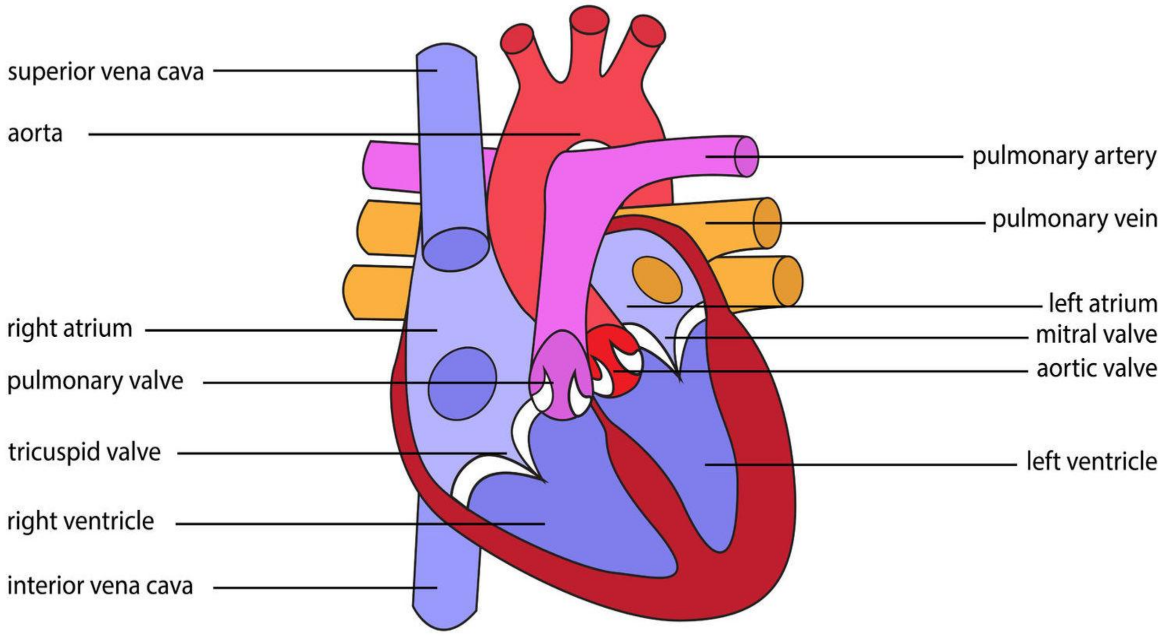

Figure-13

How to measure heart activity?

Heart activity can be recorded in two ways:

1. Electrocardiography (ECG, EKG)

- ECG records the electrical activity generated by heart muscle depolarisations, which propagate in pulsating electrical waves towards the skin. Although the electricity amount is in fact very small,

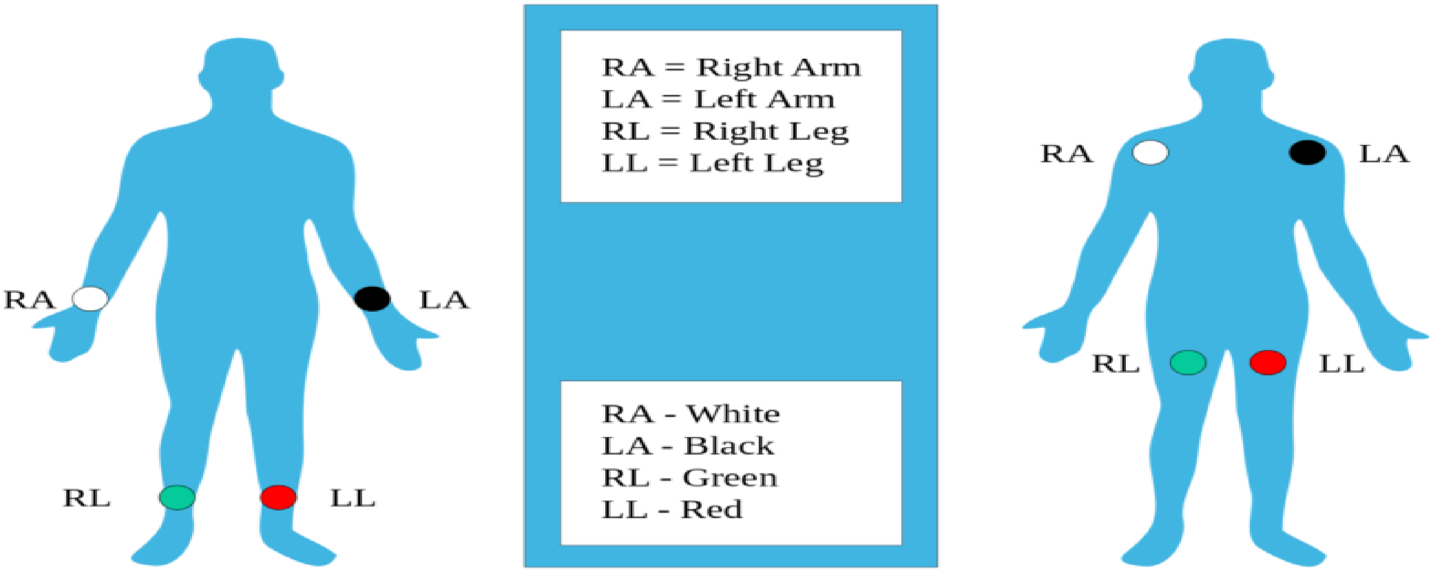

Figure-14

- Can be picked up reliably with ECG electrodes attached to the skin (data unit: microvolt, uV). The full ECG setup comprises at least four electrodes which are placed on the chest or at the four extremities according to standard nomenclature (RA = right arm; LA = left arm; RL = right leg; LL = left leg). Of course, variations of this setup exist in order to allow more flexible and less intrusive recordings, for example, by attaching the electrodes to the forearms and legs. ECG electrodes are typically wet sensors, requiring the use of a conductive gel to increase conductivity between skin and electrodes.

2. Photo-Plethysmography (PPG).

- Throughout the cardiac cycle, blood pressure throughout the body increases and decreases – even in the outer layers and small vessels of the skin. Peripheral blood flow can be measured using optical sensors attached to the fingertip, the ear lobe or other capillary tissue. The device has an LED that sends light into the tissue and records how much light is either absorbed or reflected to the photodiode. While not as accurate as ECG recordings, PPG clips use dry sensors and can be attached much quicker compared to ECG setups, making their use easier and less bothersome for participants.

WAVEFORM OF ECG OUTPUT:

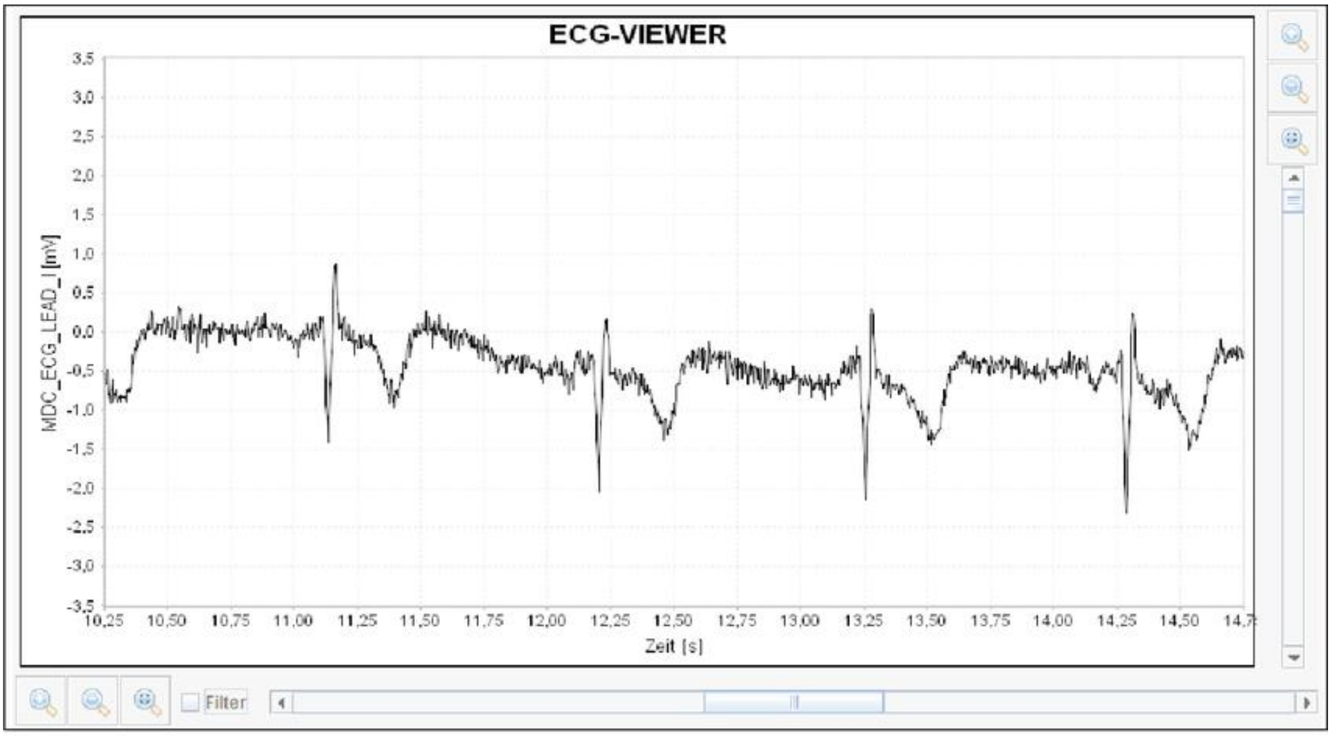

Figure 15

2.2.4 STMICRO-CONTROLLER:

STM32 is a family of 32-bit microcontroller integrated circuits by STMicroelectronics. The STM32 chips are grouped into related series that are based around the same 32-bit ARM processor core, such as the Cortex-M7F, Cortex-M4F, Cortex-M3, Cortex-M0+, or Cortex-M0. Internally, each microcontroller consists of the processor core, static RAM memory, flash memory, debugging interface, and various peripherals.

The **STM32** is a family of microcontroller ICs based on the 32-bit RISC ARM Cortex-M7F, Cortex-M4F, Cortex-M3, Cortex-M0+, and Cortex-M0 cores. STMicroelectronics licenses the ARM Processor IP from ARM Holdings. The ARM core designs have numerous configurable options, and ST chooses the individual configuration to use for each design. ST attaches their own peripherals to the core before converting the design into a silicon die. The following tables summarize the STM32 microcontroller families.

STM32 Series	ARM CPU Core
L5	Cortex-M33
F7, H7	Cortex-M7F
F4, F3, L4, J	Cortex-M4F
F2, F1, L1, W, J	Cortex-M3
L0, J	Cortex-M0+
F0, J	Cortex-M0

Table -2

STM32 F0:

STM32 F0 series	
Produced	From 2012 to current
Max. CPU clock rate	48 MHz
Instruction set	Thumb-1 (most), Thumb-2 (some)
Micro architecture	ARM Cortex-M0[6]

Table-3

The STM32 F0-series are the first group of ARM Cortex-M0 chips in the STM32 family. The summary for this series is:

- Core:
  - ARM Cortex-M0 core at a maximum clock rate of 48 MHz.
  - Cortex-M0 options include the SysTick Timer.
- Memory:
  - Static RAM consists of 4 / 6 / 8 / 16 / 32 KB general purpose with hardware parity checking.
  - Flash consists of 16 / 32 / 64 / 128 / 256 KB general purpose.
  - Each chip has a factory-programmed 96-bit unique device identifier number. (except STM32F030x4/6/8/C and STM32F070x6/B,)
- Peripherals:
- Each F0-series includes various peripherals that vary from line to line.
- Oscillators consists of internal (8 MHz, 40 kHz), optional external (1 to 32 MHz, 32.768 to 1000 kHz).
- IC packages: TSSOP20, UFQFPN32, LQFP/UFQFN48, LQFP64, LQFP/UFBGA100.
- Operating voltage range is 2.0 to 3.6 volt with the possibility to go down to 1.65 V

STM32F0 Discovery Board:

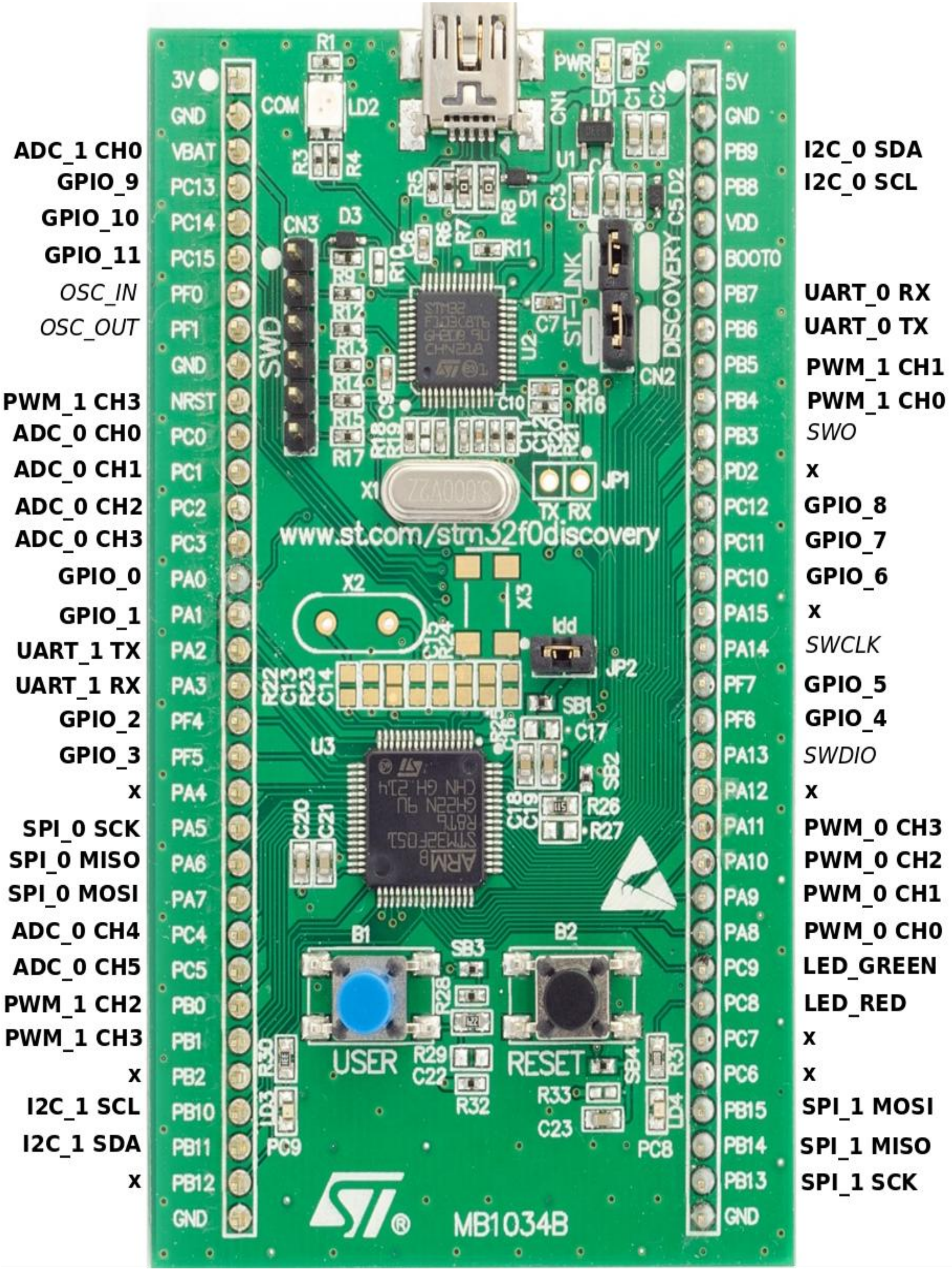

Figure 16

MCU:

MCU	**STM32F051R8T6**
Family	ARM Cortex-M0
Vendor	ST Microelectronics
RAM	8Kb
Flash	64Kb
Frequency	48MHz (using the on-board 8MHz Oscillator of the ST-Link)
FPU	no
Timers	9 (8x 16-bit, 1x 32-bit [TIM2])
ADCs	1x 16-channel 12-bit
UARTs	2
SPIs	2
I2Cs	2
Vcc	2.0V - 3.6V
Datasheet	Datasheet
Reference Manual	Reference Manual
Programming Manual	Programming Manual
Board Manual	Board Manual

TABLE-4

Pin mapping in RIOT:

Please refer to this document for RIOTs static pin mapping chosen for this board. This mapping is completely arbitrary; it can be adjusted in boards/stm32f0discovery/include/periph_conf.h

User Interface

2 Buttons:

NAME	USER	RESET
Pin	PA0 (IN)	NRST

TABLE-5

2 LEDs:

NAME	LD3	LD4
Color	green	blue
Pin	PC9	PC8

TABLE-6

CHAPTER-3

3.1 INTERNET OF THINGS:

Figure 17

The **Internet of things** (**IoT**) is the network of physical devices, vehicles, home appliances, and other items embedded with electronics, software, sensors, actuators, and connectivity which enables these things to connect, collect and exchange data.

IoT involves extending Internet connectivity beyond standard devices, such as desktops, laptops, smartphones and tablets, to any range of traditionally *dumb* or non-internet-enabled physical devices and everyday objects. Embedded with technology, these devices can communicate and interact over the Internet, and they can be remotely monitored and controlled. With the arrival of driverless vehicles, a branch of IoT, i.e. the Internet of Vehicles starts to gain more attention.

The definition of the Internet of things has evolved due to convergence of multiple technologies, real-time analytics, machine learning, commodity sensors, and embedded systems. Traditional fields of embedded systems, wireless sensor networks, control systems, automation (including home and building automation), and others all contribute to enabling the Internet of things.

The concept of a network of smart devices was discussed as early as 1982, with a modified Coke machine at Carnegie Mellon University becoming the first Internet-connected appliance, able to report its inventory and whether newly loaded drinks were cold. Mark Weiser's 1991 paper on ubiquitous computing, "The Computer of the 21st Century", as well as academic venues such as UbiComp and PerCom produced the contemporary vision of IoT. In 1994, Reza Raji described the concept in *IEEE Spectrum* as "[moving] small packets of data to a large set of nodes, so as to integrate and automate everything from home appliances to entire factories". Between 1993 and 1997, several companies proposed solutions like Microsoft's at Work or Novell's NEST. The field gained momentum when Bill Joy envisioned Device to Device (D2D) communication as a part of his "Six Webs" framework, presented at the World Economic Forum at Davos in 1999.

The term "Internet of things" was likely coined by Kevin Ashton of Procter & Gamble, later MIT's Auto-ID Center, in 1999, though he prefers the phrase "Internet *for* things".At that point, he viewed Radio-frequency identification (RFID) as essential to the Internet of things,which would allow computers to manage all individual things.

A research article mentioning the Internet of things was submitted to the conference for Nordic Researchers in Logistics, Norway, in June 2002, which was preceded by an article published in Finnish in January 2002. The implementation described there was developed by Kary Främling and his team at Helsinki

University of Technology and more closely matches the modern one, i.e. an information system infrastructure for implementing smart, connected objects.

Benefits:

Simultaneous reporting and monitoring:

Real-time monitoring via connected devices can save lives in event of a medical emergency like heart failure, diabetes, asthma attacks, etc. With real-time monitoring of the condition in place by means of a smart medical device connected to a Smartphone app, connected devices can collect medical and other required health data and use the data connection of the Smartphone to transfer collected information to a physician.

Center of Connected Health Policy conducted a study that indicates that there was a 50% reduction in 30-day readmission rate because of remote patient monitoring on heart failure patients.

The IoT device collects and transfers health data: blood pressure, oxygen and blood sugar levels, weight, and ECGs. These data are stored in the cloud and can be shared with an authorized person, who could be a physician, your insurance company, a participating health firm or an external consultant, to allow them to look at the collected data regardless of their place, time, or device.

End-to-end connectivity and affordability:

IoT can automate patient care workflow with the help healthcare mobility solution and other new technologies, and next-gen healthcare facilities. IoT enables interoperability, machine-to-machine communication, information exchange, and data movement that make healthcare service delivery effective.

Connectivity protocols: Bluetooth LE, Wi-Fi, Z-wave, ZigBee, and other modern protocols, healthcare personnel can change the way they spot illness and ailments in patients and can also innovate revolutionary ways of treatment.

Consequently, technology-driven setup brings down the cost, by cutting down unnecessary visits, utilizing better quality resources, and improving the allocation and planning.

Remote medical assistance:

In event of an emergency, patients can contact a doctor who is many kilometers away with smart mobile apps. With mobility solutions in healthcare, the medics can instantly check the patients and identify the ailments on-the-go.

Also, numerous healthcare delivery chains that are forecasting to build machines that can distribute drugs on the basis of patient's prescription and ailment-related data available via linked devices.

IoT will improve the patient's care in hospital. This in turn, will cut on people's expanse on healthcare.

3.2 Applications:

Consumer applications:

A growing portion of IoT devices are created for consumer use, including connected vehicles, home automation, wearable technology, connected health, and appliances with remote monitoring capabilities.

Smart home:

IoT devices are a part of the larger concept of home automation, which can include lighting, heating and air conditioning, media and security systems. Long term benefits could include energy savings by automatically ensuring lights and electronics are turned off.

A smart home or automated home could be based on a platform or hubs that control smart devices and appliances. For instance, using Apple's HomeKit, manufacturers can get their home products and accessories be controlled by an application in iOS devices such as the iPhone and the Apple Watch. This could be a dedicated app or iOS native applications such as Siri. This can be demonstrated in the case of Lenovo's Smart Home Essentials, which is a line of smart home devices that are controlled through Apple's Home app or Siri without the need for a Wi-Fi bridge.There are also dedicated smart home hubs that are offered as standalone platforms to connect different smart home products and these include the Amazon Echo, Apple's HomePod, and Samsung's SmartThings Hub.

Elder care:

One key application of smart home is to provide assistance for those with disabilities and elderly individuals. These home systems use assistive technology to accommodate an owner's specific disabilities.[35] Voice control can assist users with sight and mobility limitations while alert systems can be connected directly to cochlear implants worn by hearing impaired users.] They can also be equipped with additional safety features. These features can include sensors that monitor for medical emergencies such as falls or seizures.[37] Smart home technology applied in this way can provide users with more freedom and a higher quality of life.[38]

The term "Enterprise IoT" refers to devices used in business and corporate settings. By 2019, it is estimated that EIoT will account for 9.1 billion devices.

Medical and healthcare:

The **Internet of Medical Things** (also called the **internet of health things**) is an application of the IoT for medical and health related purposes, data collection and analysis for research, and monitoring. This ‘Smart Healthcare’, as it can also be called, led to the creation of a digitized healthcare system, connecting available medical resources and healthcare services.

IoT devices can be used to enable remote health monitoring and emergency notification systems. These health monitoring devices can range from blood pressure and heart rate monitors to advanced devices capable of monitoring specialized implants, such as pacemakers, Fitbit electronic wristbands, or advanced hearing aids. Some hospitals have begun implementing "smart beds" that can detect when they are occupied and when a patient is attempting to get up. It can also adjust itself to ensure appropriate pressure and support is applied to the patient without the manual interaction of nurses. A 2015 Goldman Sachs report indicated that healthcare IoT devices "can save the United States more than $300 billion in annual healthcare expenditures by increasing revenue and decreasing cost."Moreover, the use of mobile devices to support medical follow-up led to the creation of ‘m-health’, used “to analyze, capture, transmit and store health statistics from multiple resources, including sensors and other biomedical acquisition systems”.

Specialized sensors can also be equipped within living spaces to monitor the health and general well-being of senior citizens, while also ensuring that proper treatment is being administered and assisting people regain lost mobility via therapy as well. These sensors create a network of intelligent sensors that are able to collect, process, transfer and analyse valuable information in different environments, such as connecting in-home monitoring devices to hospital-based systems. Other consumer devices to encourage healthy living, such as connected scales or wearable heart monitors, are also a possibility with the IoT. End-to-end health monitoring IoT platforms are also available for antenatal and chronic patients, helping one manage health vitals and recurring medication requirements.

As of 2018 IoMT was not only being applied in the clinical laboratory industry, but also in the healthcare and health insurance industries. IoMT in the healthcare industry is now permitting doctors, patients and others involved (i.e. guardians of patients, nurses, families, etc.) to be part of a system, where patient records are saved in a database, allowing doctors and the rest of the medical staff to have access to the patient's information. Moreover, IoT-based systems are patient-centered, which involves being flexible to the patient's medical conditions. IoMT in the insurance industry provides access to better and new types of dynamic information. This includes sensor-based solutions such as biosensors, wearables, connected health devices and mobile apps to track customer behaviour. This can lead to more accurate underwriting and new pricing models.

3.3 Trends and characteristics:

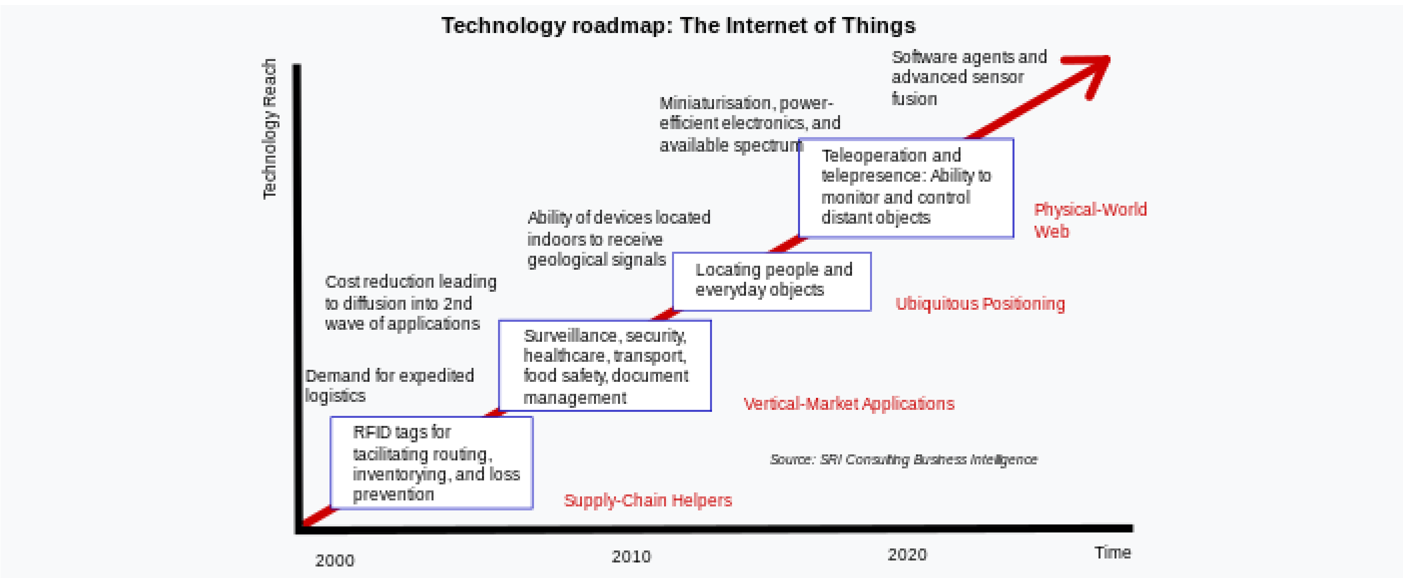

Figure 18

Technology roadmap: Internet of things.

The IoT's major significant trend in recent years is the explosive growths of devices connected and controlled by the Internet.The wide range of applications for IoT technology mean that the specifics can be very different from one device to the next but there are basic characteristics shared by most.

IoT creates opportunities for more direct integration of the physical world into computer-based systems, resulting in efficiency improvements, economic benefits, and reduced human exertions.

The number of IoT devices increased 31% year-over-year to 8.4 billion in the year 2017 and it is estimated that there will be 30 billion devices by 2020.The global market value of IoT is projected to reach $7.1 trillion by 2020.

CHAPTER-4

WEB SERVER

4.1 INTRODUCTION:

Web server refers to server software, or hardware dedicated to running said software, that can serve contents to the World Wide Web. A web server processes incoming network requests over HTTP and several other related protocols.

The primary function of a web server is to store, process and deliver web pages to clients. The communication between client and server takes place using the Hypertext Transfer Protocol (HTTP). Pages delivered are most frequently HTML documents, which may include images, style sheets and scripts in addition to the text content.

A user agent, commonly a web browser or web crawler, initiates communication by making a request for a specific resource using HTTP and the server responds with the content of that resource or an error message if unable to do so. The resource is typically a real file on the server's secondary storage, but this is not necessarily the case and depends on how the web server is implemented.

While the primary function is to serve content, a full implementation of HTTP also includes ways of receiving content from clients. This feature is used for submitting web forms, including uploading of files.

Many generic web servers also support server-side scripting using Active Server Pages (ASP), PHP (Hypertext Preprocessor), or other scripting languages. This means that the behaviour of the web server can be scripted in separate files, while the actual server software remains unchanged. Usually, this function is used to generate HTML documents dynamically ("on-the-fly") as opposed to returning static documents. The former is primarily used for retrieving or modifying information from databases. The latter is typically much faster and more easily cached but cannot deliver dynamic content.

Web servers are not only used for serving the World Wide Web. They can also be found embedded in devices such as printers, routers, webcams and serving only a local network. The web server may then be used as a part of a system for monitoring or administering the device in question. This usually means that no additional software has to be installed on the client computer since only a web browser is required (which now is included with most operating systems).

NETWORK CONNECTIONS AND PORTS:

To connect to a server, the client must be able to communicate with it over the network. Computers connected to the Internet typically communicate using TCP/IP (Transmission Control Protocol and the Internet Protocol). TCP/IP allows different types of computers to communicate at a low level; it is up to applications, however, to determine how client and server software talk to each other. Applications such as e-mail, ftp, and Web browsers use their own protocols (SMTP, HTTP, etc.) to communicate on the application level while using TCP/IP at the network level.

TCP/IP uses IP addresses to communicate between computers. Each computer on the Internet has its own unique IP address. When a computer wants to send a message to another machine on the Internet, it specifies the address of the other machine and the message finds its way through the network. This is similar to how a letter finds its way through the postal system. The destination computer may have many different ser-vices running on it, so to specify which service we want to communicate with, we must use a port number. Each service has a unique number as-signed to it known as a *port number.* Most of the services have standard port numbers.

SERVERS AND BROWSERS:

The main goal of any Web server is to provide documents to clients. The first Web servers were very simple and did little more than this. Today's Web servers are full of features that allow them to do more than just respond to simple requests for static documents, and many provide easy-to-use graphical user interfaces for administration and customization. Today's servers support options that allow the creation of dynamic documents—documents that are generated on the fly, not stored on disk.

The purpose of a Web browser is to retrieve and display information from a Web server by using HTTP. A browser allows any user to access a server easily. Without even knowing what a Web server is, a user can easily obtain information from one just by entering a URL. Browsers have evolved.

The primary function of a Web browser:

The primary function of a Web browser is to display HTML documents. Al-though it can be used to view local documents on a hard drive, it is normally used as a client to retrieve documents from an HTTP server. Although browser software has expanded over the past few years to include such services as e-mail and news, its primary function is to format HTML documents for display.

4.2 Functions of web server:

1. Stores and secures website data:

In web hosting services, a web server stores all website data and secures it from unauthorized users when it is properly configured.

2. Provides web database access:

A web server’s responsibility is to provide access to websites that are hosted. Web hosting service providers own some web servers that are used in variable ways to provide different web hosting services, such as backend database servers.

3. Serve the end user requests:

Web servers accept requests from different users connected over the internet and serve them accordingly.

4. Bandwidth controlling to regulate network traffic:

It is a feature available in web server to minimize excess network traffic. Web Hosts can set bandwidth values to regulate the rate of data transmission over the internet. This feature avoids the down time caused by high web traffic.

5. Virtual hosting:

Virtual Hosting is a type of web hosting service in which a web server is used to host other software based virtual web-servers web sites, data, applications and other services. Virtualized Web servers do possess this feature to provide virtual hosting.

6. Server side web scripting:

This feature of web server enables the user to create dynamic web pages. The popular server side scripting languages include Perl, Ruby, Python, PHP and ASP etc.

4.3 Application server:

An **application server** is a software framework that provides both facilities to create web applications and a server environment to run them.

Application Server Frameworks contain a comprehensive service layer model. An application server acts as a set of components accessible to the software developer through a standard API defined for the platform itself. For Web applications, these components are usually performed in the same running environment as their web server(s), and their main job is to support the construction of dynamic pages. However, many application servers target much more than just Web page generation: they implement services like clustering, fail-over, and load-balancing, so developers can focus on implementing the business logic.

In the case of Java application servers, the server behaves like an extended virtual machine for running applications, transparently handling connections to the database on one side, and, often, connections to the Web client on the other.

Other uses of the term may refer to the services that a server makes available or the computer hardware on which the services run.

Application servers are system software upon which web applications or desktop applications run. Application servers consist of web server connectors, computer programming languages, runtime libraries, database connectors, and the administration code needed to deploy, configure, manage, and connect these components on a web host. An application server runs behind a web Server (e.g. Apache or Microsoft Internet Information Services (IIS)) and (almost always) in front of an SQL database (e.g. PostgreSQL, MySQL, or Oracle). Web applications are computer code which run atop application servers and are written in the language(s) the application server supports and call the runtime libraries and components the application server offers.

Many application servers exist. The choice impacts the cost, performance, reliability, scalability, and maintainability of a web application.

Proprietary application servers provide system services in a well-defined but proprietary manner. The application developers develop programs according to the specification of the application server. Dependence on a particular vendor is the drawback of this approach.

An opposite but analogous case is the Java EE platform. Java EE application servers provide system services in a well-defined, open, industry standard. The application developers develop programs according to the Java EE specification and not according to the application server. A Java EE application developed according to Java EE standard can be deployed in any Java EE application server making it vendor independent.

Java application servers:

Java Platform, Enterprise Edition or Java EE (was J2EE) defines the core set of API and features of Java Application Servers.

The Java EE infrastructure is partitioned into logical containers.

- EJB container: Enterprise JavaBeans (EJB) are used to manage transactions. According to the J2EE blueprints, the business logic of an application resides in Enterprise JavaBeans—a modular server component providing many features, including declarative transaction management, and improving application scalability.
- Web container: The Web modules include servlets and JavaServer Pages (JSP).
- JCA container (Java EE Connector Architecture)

- JMS provider (Java Message Service)

Some Java Application Servers leave off many Java EE features like EJB and Java Message Service (JMS) including Jetty from Eclipse Foundation. Their focus is more on Java Servlets and JavaServer Pages.

There are many open source Java application servers that support Java EE including JOnAS from Object Web, WildFly (formerly JBoss AS) from JBoss (division of Red Hat), Geronimo from Apache, TomEE from Apache, Resin Java Application Server from Caucho Technology, Blazix from Desiderata Software, Enhydra Server from Enhydra.org,GlassFish from Oracle and Payara Server from C2B2.

Commercial Java application servers have been dominated by WebLogic Application Server by Oracle, Web Sphere Application Server from IBM and the open source JBoss Enterprise Application Platform (JBoss EAP) by Red Hat, but there are also new platforms, like JLupin Next Server, which present new approach for architecture focusing on distributed computing and microservices environments.

A Java Server Page (JSP) executes in a web container. JSPs provide a way to create HTML pages by embedding references to the server logic within the page. HTML coders and Java programmers can work side by side by referencing each other's code from within their own.

The application servers mentioned above mainly serve web applications, and services via RMI, EJB, JMS and SOAP. Some application servers target networks other than web-based ones: Session Initiation Protocol servers, for instance, target telephony networks

CHAPTER-5

CONCLUSION AND FUTURE WORK:

By using the system the healthcare professionals can monitor, diagnose, and advice their patients all the time. The health parameters data are stored and published online. Hence, the healthcare professional can monitor their patients from a remote location at any time. Our system is simple.

In the designed system the enhancement would be connecting more sensors to internet which measures various other health parameters and would be beneficial for patient monitoring i.e. connecting all the objects to internet for quick and easy access. Establishing a Wi-Fi mesh type network to increase in the communication range.

References:

1. Stanford V (2002) using pervasive computing to deliver elder care. IEEE Pervasive Computing 1: 10-13.

2. Mcfadden T, Indulska J (2004) Context-aware environments for independent living, In Proceedings of the 3rd National Conference of Emerging Researchers in Ageing, Brisbane, Australia.

3. Anliker U, Ward JA, Lukowicz P, Troster G, Dolveck F, et al. (2004) AMON: a wearable multi-parameter medical monitoring and alert system. IEEE Trans Inf Technol Biomed 8: 415–427.

4. Cho G, Yoo1 SK (2009) Wearable ECG Monitoring System Using Conductive Fabrics and Active Electrodes, Proceedings of the 13th International Conference on HumanComputer Interaction, Berlin, Heidelberg.

5. Darwish A, Hassanien AE (2012) Wearable and Implantable Wireless Sensor Network Solutions for Healthcare Monitoring. Sensors 12: 12375-12376.

6. Shnayder V, Chen B, Lorincz K, FulfordJones TRF, Welsh M (2005) Sensor Networks for Medical Care, Proceedings of the 3rd international conference on embedded networked sensor systems, New York, USA.

7. Alemdar H, Ersoy C (2010) Wireless sensor networks for healthcare: A survey. Computer Networks 54: 2688-2710.

8. Varshney U (2005) Using wireless networks for enhanced monitoring of patients. International Journal of HealthCare Technology and Management 6: 489-499.

9. Varshney U (2003) Pervasive healthcare. IEEE Computer 12: 138-140.

10. Real time wireless health monitoring application using mobile devices, International Journal of Computer Networks & Communications (IJCNC) Vol.7, No.3, May 2015, Amna Abdullah, Asma Ismael, Aisha Rashid, Ali Abou-ElNour, and Mohammed Tarique.

11. Secured Smart Healthcare Monitoring System Based on Iot, International Journal on Recent and Innovation Trends in Computing and Communication Volume: 3 Issue: 7, Bhoomika.B.K, Dr. K N Muralidhara.

12. Home Based Health Monitoring System Using Android Smartphone, International Journal of Electical, Electronics and Data Communication, Vol-2, Issue-2, Feb-2014, Sushama Pawar, P.W.Kulkarni.

13. Chung WY, Mo an S, Lee SC (2012) Real Time Multi-hop Routing Protocol for Healthcare System Based on WSN, 14th International Meeting on Chemical Sensors, Nuremberg, Germany.

14. Kyriacou E, Pattichis C, Hoplaros D, Kounoudes A, Milis M, et al, (2010) “A system for Monitoring Children with Suspected Cardiac Arrhythmias Technical Optimizations and Evaluation”. IFMBE Proceedings 29: 924-927.

15. Lowe SA, Brown MA, Dekker G, Gatt S, McLintock C, et al. (2008) “Guidelines for the Management of Hypertensive Disoders of Pregnancy”. Society of Obstetric medicine of Australia and New Zealand.

16. Fahim M, Fatima I, Lee S, Lee YK (2012) Daily life activity tracking application for Smart homes using android smartphone. Proceedings of 14th international conference on Advances Communication Technology (ICACT), Pyeong Chang.

17. Yuce MR (2010) “Implementation of wireless body area networks for healthcare systems”. Sensors and Actuators A: Physical 162: 116-129.

18. Keong HC, Yuce MR (2008) Low data rate ultra-wideband ECG monitoring system. Proceedings of IEEE Engineering in Medicine and Biology Society Conference, University of Newcastle, Callaghan, Australia.

Appendices:

CODE FOR EMBEDDED SYSTEM:

```
/**
  ******************************************************************************
  * @file           : main.c
  * @brief          : Main program body
  ******************************************************************************
  ** This notice applies to any and all portions of this file
  * that are not between comment pairs USER CODE BEGIN and
  * USER CODE END. Other portions of this file, whether
  * inserted by the user or by software development tools
  * are owned by their respective copyright owners.
  *
  * COPYRIGHT(c) 2018 STMicroelectronics
  *
  * Redistribution and use in source and binary forms, with or without modification,
  * are permitted provided that the following conditions are met:
  *   1. Redistributions of source code must retain the above copyright notice,
  *      this list of conditions and the following disclaimer.
  *   2. Redistributions in binary form must reproduce the above copyright notice,
  *      this list of conditions and the following disclaimer in the documentation
  *      and/or other materials provided with the distribution.
  *   3. Neither the name of STMicroelectronics nor the names of its contributors
  *      may be used to endorse or promote products derived from this software
  *      without specific prior written permission.
  *
  * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
```

```
  * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
  * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE ARE
  * DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE
LIABLE
  * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
  * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR
  * SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
  * CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY,
  * OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE
USE
  * OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
  *
  ******************************************************************************
  */
```

CODE FOR TEMPERATURE SENSOR:

```
 /*@ author:vikrant thakur
/* Includes ------------------------------------------------------------------*/
#include "main.h"
#include "stm32f0xx_hal.h"


/* USER CODE BEGIN Includes */
#include "math.h"
#include "stdbool.h"
/* USER CODE END Includes */


/* Private variables ---------------------------------------------------------*/
ADC_HandleTypeDef hadc;
```

```
TIM_HandleTypeDef htim6;

UART_HandleTypeDef huart1;

/* USER CODE BEGIN PV */
/* Private variables ---------------------------------------------------------*/
uint32_t v[2], analog_data;
float temperature;
unsigned char i,counter;
volatile bool trigger=true;
char temp1[5],ecg1[5];
unsigned char rece[0];
/* USER CODE END PV */

/* Private function prototypes -----------------------------------------------*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_ADC_Init(void);
static void MX_USART1_UART_Init(void);
static void MX_TIM6_Init(void);

/* USER CODE BEGIN PFP */
void reverse(char *str, int len);
int intToStr(int x, char str[], int d);
/* Private function prototypes -----------------------------------------------*/
/* USER CODE END PFP */

/* USER CODE BEGIN 0 */
void reverse(char *str, int len)
```

```
{
    int i=0, j=len-1, temp;
    while (i<j)
    {
        temp = str[i];
        str[i] = str[j];
        str[j] = temp;
        i++; j--;
    }
}
int intToStr(int x, char str[], int d)
{
    int i = 0;
    while (x)
    {
        str[i++] = (x%10) + '0';
        x = x/10;
    }
    // If number of digits required is more, then
    // add 0s at the beginning
    while (i < d)
        str[i++] = '0';

    reverse(str, i);
    str[i] = '\0';
    return i;
}
// Converts a floating point number to string.
void temp(char * res, int afterpoint)
```

```
{
    // Extract integer part
    int ipart = (int)temperature;
    // Extract floating part
    float fpart = temperature - (float)ipart;
    // convert integer part to string
    int i = intToStr(ipart, res, 0);
    // check for display option after point
    if (afterpoint != 0)
    {
        res[i] = '.';
        // add dot
        // Get the value of fraction part upto given no.
        // of points after dot. The third parameter is needed
        // to handle cases like 233.007
        fpart = fpart * pow(10, afterpoint);
        intToStr((int)fpart, res + i + 1, afterpoint);
    }
}
void ecg(char * res, int afterpoint)
  {
      // Extract integer part
      int ipart = (int)v[1];
      // Extract floating part
      float fpart = v[1] - (float)ipart;
      // convert integer part to string
      int i = intToStr(ipart, res, 0);
      // check for display option after point
      if (afterpoint != 0)
```

```
        {
            res[i] = '.';
            // add dot
            // Get the value of fraction part upto given no.
            // of points after doet. The third parameter is needed
            // to handle cases like 233.007
            fpart = fpart * pow(10, afterpoint);
            intToStr((int)fpart, res + i + 1, afterpoint);
        }
    }

  void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef* hadc)
  {
    if (__HAL_ADC_GET_FLAG(hadc, ADC_FLAG_EOC))
      {
      v[i] = HAL_ADC_GetValue(hadc); // analog data for temperature ecg and heartbeat
      i++;
      }
    if (__HAL_ADC_GET_FLAG(hadc, ADC_FLAG_EOS))
     {
     i=0;
     temperature = (3.3 * v[0] * 100.0) / 4096.0;
     }
  }
/* USER CODE END 0 */

/**
  * @brief  The application entry point.
  *
```

```
  * @retval None
  */
int main(void)
{
  /* USER CODE BEGIN 1 */
   unsigned char blank=0;
  /* USER CODE END 1 */

  /* MCU Configuration----------------------------------------------------------*/

  /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
  HAL_Init();

  /* USER CODE BEGIN Init */
  /* USER CODE END Init */

  /* Configure the system clock */
  SystemClock_Config();

  /* USER CODE BEGIN SysInit */
  /* USER CODE END SysInit */

  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  MX_ADC_Init();
  MX_USART1_UART_Init();
  MX_TIM6_Init();
  /* USER CODE BEGIN 2 */
  /* USER CODE END 2 */
```

```
  /* Infinite loop */

  /* USER CODE BEGIN WHILE */

  while (1)

{

  /* USER CODE END WHILE */


  /* USER CODE BEGIN 3 */

        //matlab send data

       HAL_UART_Receive(&huart1,rece,1,HAL_MAX_DELAY);

       if(rece[0]=='1')

       {

              // temperature sensor data

          HAL_GPIO_WritePin(GPIOC,GPIO_PIN_6,1);

          while(counter<10)

          {

    HAL_ADC_Start_IT(&hadc);

          HAL_Delay(10);

          temp(temp1,2);

          HAL_UART_Transmit(&huart1,temp1,sizeof(temp1),HAL_MAX_DELAY);

          counter++;

          }

       }

       else if(rece[0]=='2')

        {
```

// **CODE FOR ECG SENSOR:**

```
    if((HAL_GPIO_ReadPin(GPIOC,GPIO_PIN_5)==1) &&
(HAL_GPIO_ReadPin(GPIOC,GPIO_PIN_6)==1))

          {

           HAL_UART_Transmit(&huart1,blank,sizeof(blank),HAL_MAX_DELAY);
```

```
                }
        else
        {
           HAL_GPIO_WritePin(GPIOC,GPIO_PIN_7,1);
           while(counter<20)
                        {
                    HAL_ADC_Start_IT(&hadc);
                        HAL_Delay(10);
                        ecg(ecg1,2);
                        HAL_UART_Transmit(&huart1,temp1,sizeof(temp1),HAL_MAX_DELAY);
                        counter++;
                        }
        }
            }
}
  /* USER CODE END 3 */

}

/**
  * @brief System Clock Configuration
  * @retval None
  */
void SystemClock_Config(void)
{

  RCC_OscInitTypeDef RCC_OscInitStruct;
  RCC_ClkInitTypeDef RCC_ClkInitStruct;
  RCC_PeriphCLKInitTypeDef PeriphClkInit;
```

```
  /**Initializes the CPU, AHB and APB busses clocks
  */
 RCC_OscInitStruct.OscillatorType =
RCC_OSCILLATORTYPE_HSI|RCC_OSCILLATORTYPE_HSI14;
 RCC_OscInitStruct.HSIState = RCC_HSI_ON;
 RCC_OscInitStruct.HSI14State = RCC_HSI14_ON;
 RCC_OscInitStruct.HSICalibrationValue = 16;
 RCC_OscInitStruct.HSI14CalibrationValue = 16;
 RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
 RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSI;
 RCC_OscInitStruct.PLL.PLLMUL = RCC_PLL_MUL6;
 RCC_OscInitStruct.PLL.PREDIV = RCC_PREDIV_DIV1;
 if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
 {
  _Error_Handler(__FILE__, __LINE__);
 }

  /**Initializes the CPU, AHB and APB busses clocks
  */
 RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
                   |RCC_CLOCKTYPE_PCLK1;
 RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
 RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
 RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV1;

 if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_1) != HAL_OK)
 {
  _Error_Handler(__FILE__, __LINE__);
 }
```

```
  PeriphClkInit.PeriphClockSelection = RCC_PERIPHCLK_USART1;
  PeriphClkInit.Usart1ClockSelection = RCC_USART1CLKSOURCE_PCLK1;
  if (HAL_RCCEx_PeriphCLKConfig(&PeriphClkInit) != HAL_OK)
  {
    _Error_Handler(__FILE__, __LINE__);
  }

    /**Configure the Systick interrupt time
    */
  HAL_SYSTICK_Config(HAL_RCC_GetHCLKFreq()/1000);

    /**Configure the Systick
    */
  HAL_SYSTICK_CLKSourceConfig(SYSTICK_CLKSOURCE_HCLK);

  /* SysTick_IRQn interrupt configuration */
  HAL_NVIC_SetPriority(SysTick_IRQn, 0, 0);
}

/* ADC init function */
static void MX_ADC_Init(void)
{

  ADC_ChannelConfTypeDef sConfig;

    /**Configure the global features of the ADC (Clock, Resolution, Data Alignment and number of
conversion)
    */
  hadc.Instance = ADC1;
```

```
hadc.Init.ClockPrescaler = ADC_CLOCK_ASYNC_DIV1;
hadc.Init.Resolution = ADC_RESOLUTION_12B;
hadc.Init.DataAlign = ADC_DATAALIGN_RIGHT;
hadc.Init.ScanConvMode = ADC_SCAN_DIRECTION_FORWARD;
hadc.Init.EOCSelection = ADC_EOC_SINGLE_CONV;
hadc.Init.LowPowerAutoWait = DISABLE;
hadc.Init.LowPowerAutoPowerOff = DISABLE;
hadc.Init.ContinuousConvMode = DISABLE;
hadc.Init.DiscontinuousConvMode = DISABLE;
hadc.Init.ExternalTrigConv = ADC_SOFTWARE_START;
hadc.Init.ExternalTrigConvEdge = ADC_EXTERNALTRIGCONVEDGE_NONE;
hadc.Init.DMAContinuousRequests = DISABLE;
hadc.Init.Overrun = ADC_OVR_DATA_PRESERVED;
if (HAL_ADC_Init(&hadc) != HAL_OK)
{
  _Error_Handler(__FILE__, __LINE__);
}

  /**Configure for the selected ADC regular channel to be converted.
  */
sConfig.Channel = ADC_CHANNEL_1;
sConfig.Rank = ADC_RANK_CHANNEL_NUMBER;
sConfig.SamplingTime = ADC_SAMPLETIME_239CYCLES_5;
if (HAL_ADC_ConfigChannel(&hadc, &sConfig) != HAL_OK)
{
  _Error_Handler(__FILE__, __LINE__);
}

  /**Configure for the selected ADC regular channel to be converted.
```

```
  */
  sConfig.Channel = ADC_CHANNEL_2;
  if (HAL_ADC_ConfigChannel(&hadc, &sConfig) != HAL_OK)
  {
    _Error_Handler(__FILE__, __LINE__);
  }

    /**Configure for the selected ADC regular channel to be converted.
    */
  sConfig.Channel = ADC_CHANNEL_3;
  if (HAL_ADC_ConfigChannel(&hadc, &sConfig) != HAL_OK)
  {
    _Error_Handler(__FILE__, __LINE__);
  }

}

/* TIM6 init function */
static void MX_TIM6_Init(void)
{

  TIM_MasterConfigTypeDef sMasterConfig;

  htim6.Instance = TIM6;
  htim6.Init.Prescaler = 0;
  htim6.Init.CounterMode = TIM_COUNTERMODE_UP;
  htim6.Init.Period = 0;
  htim6.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_DISABLE;
  if (HAL_TIM_Base_Init(&htim6) != HAL_OK)
```

```
  {
    _Error_Handler(__FILE__, __LINE__);
  }

  sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
  sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
  if (HAL_TIMEx_MasterConfigSynchronization(&htim6, &sMasterConfig) != HAL_OK)
  {
    _Error_Handler(__FILE__, __LINE__);
  }

}

/* USART1 init function */
static void MX_USART1_UART_Init(void)
{

  huart1.Instance = USART1;
  huart1.Init.BaudRate = 9600;
  huart1.Init.WordLength = UART_WORDLENGTH_8B;
  huart1.Init.StopBits = UART_STOPBITS_1;
  huart1.Init.Parity = UART_PARITY_NONE;
  huart1.Init.Mode = UART_MODE_TX_RX;
  huart1.Init.HwFlowCtl = UART_HWCONTROL_NONE;
  huart1.Init.OverSampling = UART_OVERSAMPLING_16;
  huart1.Init.OneBitSampling = UART_ONE_BIT_SAMPLE_DISABLE;
  huart1.AdvancedInit.AdvFeatureInit = UART_ADVFEATURE_NO_INIT;
  if (HAL_UART_Init(&huart1) != HAL_OK)
  {
```

```
    _Error_Handler(__FILE__, __LINE__);
  }


}
```

CODE FOR HEARTBEAT SENSOR:

```
/** Configure pins as
        * Analog
        * Input
        * Output
        * EVENT_OUT
        * EXTI
     PA1   ------> SharedAnalog_PA1
     PA2   ------> SharedAnalog_PA2
     PA3   ------> SharedAnalog_PA3
*/
static void MX_GPIO_Init(void)
{


  GPIO_InitTypeDef GPIO_InitStruct;


  /* GPIO Ports Clock Enable */
  __HAL_RCC_GPIOA_CLK_ENABLE();
  __HAL_RCC_GPIOC_CLK_ENABLE();
  __HAL_RCC_GPIOB_CLK_ENABLE();


  /*Configure GPIO pin Output Level */
  HAL_GPIO_WritePin(GPIOC, GPIO_PIN_6|GPIO_PIN_7|GPIO_PIN_8|GPIO_PIN_9, 
GPIO_PIN_RESET);


  /*Configure GPIO pin : PA0 */
```

```
GPIO_InitStruct.Pin = GPIO_PIN_0;
GPIO_InitStruct.Mode = GPIO_MODE_INPUT;
GPIO_InitStruct.Pull = GPIO_NOPULL;
HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);

/*Configure GPIO pins : PA1 PA2 PA3 */
GPIO_InitStruct.Pin = GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3;
GPIO_InitStruct.Mode = GPIO_MODE_ANALOG;
GPIO_InitStruct.Pull = GPIO_NOPULL;
HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);

/*Configure GPIO pins : PC4 PC5 */
GPIO_InitStruct.Pin = GPIO_PIN_4|GPIO_PIN_5;
GPIO_InitStruct.Mode = GPIO_MODE_INPUT;
GPIO_InitStruct.Pull = GPIO_NOPULL;
HAL_GPIO_Init(GPIOC, &GPIO_InitStruct);

/*Configure GPIO pins : PB14 PB15 */
GPIO_InitStruct.Pin = GPIO_PIN_14|GPIO_PIN_15;
GPIO_InitStruct.Mode = GPIO_MODE_INPUT;
GPIO_InitStruct.Pull = GPIO_NOPULL;
HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);

/*Configure GPIO pins : PC6 PC7 PC8 PC9 */
GPIO_InitStruct.Pin = GPIO_PIN_6|GPIO_PIN_7|GPIO_PIN_8|GPIO_PIN_9;
GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
HAL_GPIO_Init(GPIOC, &GPIO_InitStruct);
```

```
}

/* USER CODE BEGIN 4 */

/* USER CODE END 4 */

/**
  * @brief  This function is executed in case of error occurrence.
  * @param  file: The file name as string.
  * @param  line: The line in file as a number.
  * @retval None
  */
void _Error_Handler(char *file, int line)
{
  /* USER CODE BEGIN Error_Handler_Debug */
  /* User can add his own implementation to report the HAL error return state */
  while(1)
  {

  }
  /* USER CODE END Error_Handler_Debug */
}

#ifdef  USE_FULL_ASSERT
/**
  * @brief  Reports the name of the source file and the source line number
  *         where the assert_param error has occurred.
  * @param  file: pointer to the source file name
```

```
  * @param  line: assert_param error line source number
  * @retval None
  */
void assert_failed(uint8_t* file, uint32_t line)
{
  /* USER CODE BEGIN 6 */
  /* User can add his own implementation to report the file name and line number,
     tex: printf("Wrong parameters value: file %s on line %d\r\n", file, line) */
  /* USER CODE END 6 */
}
#endif /* USE_FULL_ASSERT */

/**
  * @}
  */

/**
  * @}
  */

/************************ (C) COPYRIGHT STMicroelectronics *****END OF FILE****/
```