ARUNAI PUBLICATIONS

Asian Journal of Applied Research

DOI: http://dx.doi.org/10.20468/ajar.2018.10.10

Research Article

Internet of Things-Based Health Monitoring System Using Microelectromechanical Systems' Sensor

C. Gayathri[1], S. Venkatanarayanan[2], R. Karpaga Priya[3], M. Kannan[2]

[1]Department of EEE, Mount Zion College of Engineering and Technology, Pudukkottai, Tamil Nadu, India, [2]Department of EEE, K.L.N. College of Engineering, Sivaganga, Tamil Nadu, India, [3]Department of ECE, Vaigai College of Engineering, Madurai, Tamil Nadu, India

Address for correspondence: C. Gayathri, Department of EEE, Mount Zion College of Engineering and Technology, Pudukkottai, Tamil Nadu, India. E-mail: gayathrisbskpl@gmail.com

ABSTRACT

The development of newer technologies in health care has become an immediate necessity today for elderly people. Our project aims at designing an Internet of Things (IoT)-based wearable device for elderly people to monitor them. The fast development of IoT technology makes it possible for connecting different smart objects together through the use of Internet and providing more data interoperability methods for application purpose. Nowadays, most of the elderly people experience loneliness and psychological depressions, either as a result of living alone/abandonment or due to reduced connection with their children and relatives. To monitor and help the elderly people in emergency situation, we have developed IoT-based health-care monitoring system by the use of different types of sensors such as temperature sensor, heart rate sensor, blood pressure sensors, accelerometer for fall detection, and GPS for locating the person. The collected values by the sensors are sending to the cloud space through the Raspberry Pi for monitoring, and the values are compared with the threshold values. The alert message will be send to the doctors and nearby caretakers at the time of emergency through mobile push message as well as mail push message. The message contains the current health-care parameter values of the person. Hence, the doctors can know the conditions of the patients and can help them.

Keywords: Accelerometer, Blood pressure sensors, Data interoperability, Health-care monitoring, Heart rate sensor, Internet of Things, Raspberry Pi, Temperature sensor

Received: 05th February 2018
Accepted: 20th September 2018
Published: 13th October 2018

INTRODUCTION

Internet of things (IOT)

The IoT, also called the internet of objects, refers to a wireless network between objects. Short range mobile transceivers are embedded into a wide array of gadgets which enables a new form of communication between people and things.[1] The term "IoT" has come to describe a number of technologies and research disciplines that enable the Internet to reach out into the real world of physical objects.

"Things having identities and virtual personalities operating in smart spaces using intelligent interfaces to connect and communicate within social, environment, and user contexts."

The IoT is the network of physical objects or "things" embedded with electronics, software, sensor, and network connectivity, which enables these objects to collect and exchange data. IoT allows object to be sensed and controlled remotely across existing network infrastructure, creating opportunities for more integration between the physical world and computer-based system, and resulting in improved efficiency, accuracy, and economic benefit.[2]

"Things" in the IoT senses can refer to a wide variety of devices such as heart monitoring implants, biochip transponders on farm animals, electric clams in coastal water, automobiles with built-in sensor, and DNA analysis devices that assist firefighter in search and rescue operations. These devices collect useful data with the help of various existing technologies and then autonomously flow the data between other devices.

History of IoT

The concept of the IoT first became popular in 1999, through the auto-ID center at MIT and related market analysis publication. Radiofrequency identification (RFID) was seen as a prerequisite for the IoT at that point.[3] If all objects and people in daily life were equipped with identifiers, computers could manage and inventory them. Besides using RFID, the tagging of things may be achieved through such technologies as near field communication, barcodes, QR codes, Bluetooth, and digital watermarking.

Working of IoT

IoT is not the result of a single novel technology; instead, several complementary technical developments provide

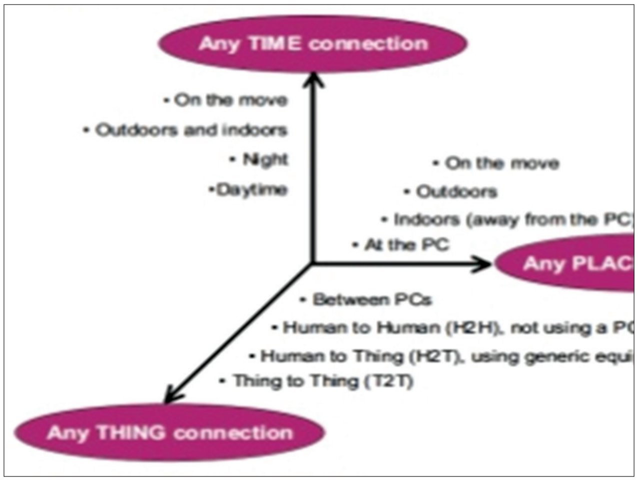

Figure 1: Internet of Things technology

capabilities that taken together help to bridge the gap between the virtual and physical world [Figure 1].

These capabilities include:

- Communication and cooperation
- Addressability
- Identification
- Sensing
- Actuation
- Embedded information processing
- Localization
- User interface.

Technological challenges of IoT

At present, IoT is faced with many challenges, such as:

- Scalability
- Technological standardization
- Interoperability
- Discovery
- Software complexity
- Data volumes interpretation
- Power supply
- Interaction and short-range communication
- Wireless communication.

IoT road map [Figure 2]

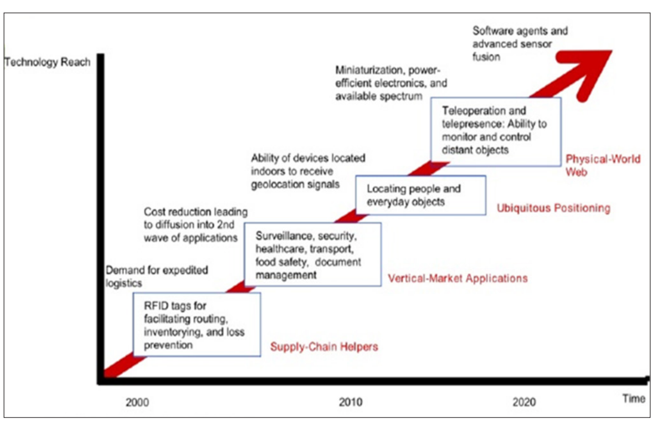

Figure 2: Internet of Things roadmap

Criticisms and controversies of IoT

Scholars and social observers and pessimists have doubts about the promises of the pervasive computing.

Revolution in the area such as:

- Privacy
- Security
- Autonomy and control
- Social control
- Political manipulation
- Design
- Environmental impact
- Influences human moral decision.

Few application of IoT

- Building and home automation
- Manufacturing
- Medical and health-care systems
- Media
- Environment monitoring
- Infrastructure management
- Energy management
- Transportation
- Better quality of life for elder.

EXISTING AND PROPOSED SYSTEM

Existing system

In the existing system, we identify status when sensors increased and have manual operation for transferring information. The system is controlling through man. Disadvantage is that patient should be in observations and failure in the updating of the status of patient to hospital.

Proposed system

The proposed system is effective and safety with systematic approach in monitoring and transferring messages through web technologies. Advantage of your system is that we can monitor system from anywhere [Figure 3].

IOT-BASED MICROELECTROMECHANICAL SYSTEMS (MEMS) SENSOR

Introduction of MEMS sensor

This report deals with the emerging field of MEMS. MEMS are a process technology used to create tiny integrated devices or systems that combine mechanical and electrical components. They are fabricated using integrated circuit (IC) batch processing techniques and can range in size from a few micrometers to millimeters.[2] These devices (or systems) have the ability to sense, control, and actuate on the microscale and generate effects on the macroscale.

The interdisciplinary nature of MEMS utilizes design, engineering, and manufacturing expertise from a wide and diverse range of technical areas including IC fabrication technology, mechanical engineering, materials science, electrical engineering, chemistry and chemical engineering, as well as fluid engineering, optics, instrumentation, and packaging. The complexity of MEMS is also shown in the extensive range of markets and applications that incorporate MEMS devices. MEMS can be found in systems ranging across automotive, medical electronics, communication, and defense applications.[4] Current MEMS devices include accelerometers for airbag sensors, Inkjet Printer Heads, computer disk drive read/write heads, projection display chips, blood pressure sensors, optical switches, micro valves, biosensors, and many other products that are all manufactured and shipped in high commercial volumes [Figure 4].

MEMS have been identified as one of the most promising technologies for the 21st century and have the potential to revolutionize both industrial and consumer products by combining silicon-based microelectronics with micromachining technology. This techniques and microsystem based devices have potential to dramatically affect our lives and the way we live. If semiconductor microfabrication was seen to be the first micro manufacturing revolution, MEMS is the second revolution.

This report introduces the field of MEMS and is divided into four main sections. In the first section, the reader is introduced to MEMS, its definitions, history, current and potential applications, as well as the state of the MEMS market and issues concerning miniaturization.

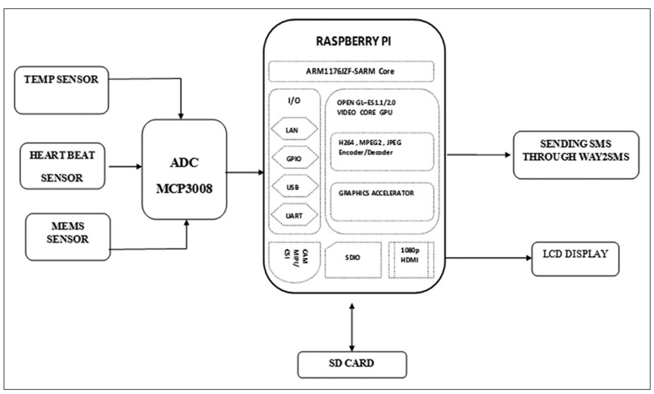

Figure 3: Block diagram

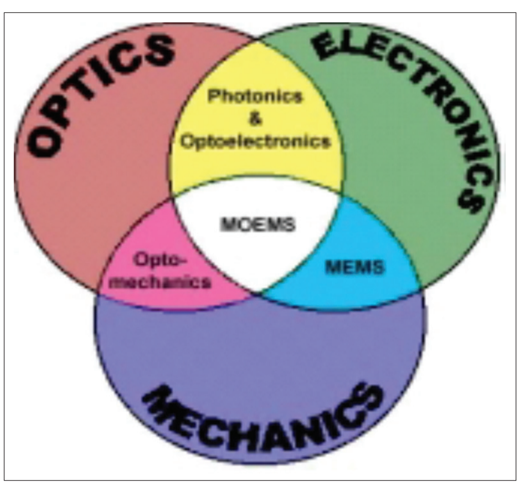

Figure 4: Microelectromechanical system technology

The second section deals with the fundamental fabrication methods of MEMS including photolithography, bulk micromachining, surface micromachining, and high-aspect-ratio micromachining; assembly, system integration, and packaging of MEMS devices are also described here. The third section reviews the range of MEMS sensors and actuators, the phenomena that can be sensed or acted on with MEMS devices, and a brief description of the basic sensing and actuation mechanisms.[4] The final section illustrates the challenges facing the MEMS industry for the common success of MEMS.

MEMS

MEMS are a process technology used to create tiny integrated devices or systems that combine mechanical and electrical components. They are fabricated using IC batch processing techniques and can range in size from a few micrometers to millimeters. These devices (or systems) have the ability to sense, control, and actuate on the microscale and generate effects on the macroscale.

Transducer

A transducer is a device that transforms one form of signal or energy into another form. The term transducer can, therefore, be used to include both sensors and actuators and is the most generic and widely used term in MEMS.[5]

Sensor

A sensor is a device that measures information from a surrounding environment and provides an electrical output signal in response to the parameter it measured. Over the years, this information (or phenomenon) has been categorized in terms of the type of energy domains, but MEMS devices generally overlap several domains or do not even belong in any one category.

These energy domains include:

- Mechanical - force, pressure, velocity, acceleration, and position
- Thermal - temperature, entropy, heat, and heat flow
- Chemical - concentration, composition, and reaction rate
- Radiant - electromagnetic wave intensity, phase, wavelength, and polarization
- Reflectance - refractive index and transmittance
- Magnetic - field intensity, flux density, magnetic moment, and permeability
- Electrical - voltage, current, charge, resistance, capacitance, and polarization.

Actuator

An actuator is a device that converts an electrical signal into an action. It can create a force to manipulate itself, other mechanical devices, or the surrounding environment to perform some useful function.

History of MEMS

The history of MEMS is useful to illustrate its diversity, challenges, and applications. The following list summarizes some of the key MEMS milestones.

1950's

1958: Silicon strain gauges commercially available

1960's

1961: First silicon pressure sensor demonstrated

1967: Invention of surface micromachining. Westinghouse creates the Resonant Gate Field Effect Transistor. Description of use of sacrificial material to free micromechanical devices from the silicon substrate.

1970's

1970: First silicon accelerometer demonstrated

1979: First micromachined inkjet nozzle

1980'

1980: First experiment was done using surface micro machined silicon.

1980: Micromachining leverage microelectronics industry and widespread experimentation and documentation increase public interest.

1982: Disposable blood pressure transducer

1982: "Silicon as a mechanical material."

Instrumental paper to entice the scientific community - reference for material properties.

1982: LIGA process

1988: First MEMS conference

1990's

Methods of micromachining aimed toward improving sensors.

1992: MCNC starts the Multi-User MEMS Process (MUMPS) sponsored by the Defense Advanced Research Projects Agency (DARPA)

1992: First micromachined hinge

1993: First surface micromachined accelerometer sold (Analog Devices, ADXL50)

1994: Deep reactive-ion etching is patented

1995: Bio-MEMS rapidly develops

2000: MEMS optical-networking components become bigness.

Few MEMS applications

Automotive airbag sensor

Automotive airbag sensors were one of the first commercial devices using MEMS. They are in widespread use today in the form of a single chip containing a smart sensor or accelerometer, which measures the rapid deceleration of a vehicle on hitting an object. The deceleration is sensed by a change in voltage. An electronic control unit subsequently sends a signal to trigger and explosively fill the airbag.

Medical pressure sensor

Another example of an extremely successful MEMS application is the miniature disposable pressure sensor used to monitor blood pressure in hospitals. These sensors connect to a patient's intravenous (IV) line and monitor the blood pressure through the IV solution. For a fraction of their cost ($10), they replace the early external blood pressure sensors that cost over $600 and had to be sterilized and recalibrated for reuse. These expensive devices measure blood pressure with a saline-filled tube and diaphragm arrangement that has to be connected to an artery with a needle.

IoT-Based MEMS sensor

The MEMS sensor is a small, thin, low-power, complete 3-axis accelerometer with signal-conditioned voltage outputs, all on a single monolithic IC. The product measures acceleration with a minimum full-scale range of ±3 g.[5] It can measure the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion, shock, or vibration.

Some of the unique features of MEMS sensors are as follows[6]:

- A unique sensor portfolio, from discrete to fully-integrated solutions, to meet all design
- High-volume manufacturing capacity to provide cost-competitive solutions, fast time-to-market, and security of supply
- High-performance sensor fusion to improve the accuracy of multiaxis sensor systems to enable new emerging and highly demanding applications, such as indoor navigation and location-based services such as IoT, indoor navigation, and location-based services
- High-quality products, already tested in different application fields, including mobile, portable, gaming, consumer, automotive, and health-care segments
- Multiple sites dedicated to MEMS, with full in-house dual sourcing, guaranteeing 100% security of supply

As an emerging technology, MEMS products are center around technology-product paradigms rather than product-market paradigms. Consequently, a MEMS device may find numerous applications across a diversity of industries. For example, the MEMS inkjet printer head nozzle in widespread use today has developed from a nozzle originally used in nuclear separation [Table 1].

Application of MEMS

Table 2 explains the MEMS sensors used in various fields such as automotive, medical, communication, and defense.

HARDWARE AND SOFTWARE ARCHITECTURE

Raspberry pi

The raspberry Pi is a low-cost, credit card-sized computer that plugs into a computer monitor or TV and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing and to learn how to program in languages such as Scratch and Python. An interesting aspect of the Raspbian project is unofficial. The Raspberry Pi 3 is a $35 computer that is on the cusp of challenging the modern PC [Figure 5].

Table 1: Commercialization of products

Product	Discovery	Evolution	Cost reduction/application expansion	Full commercialization
Pressure sensor	1954-1980	1960-1975	1975-1990	1990 – Present
Accelerometers	1974-1985	1985-1990	1990-1998	1996
Gas sensors	1988-1994	1994-1998	1998-2005	2005
Valves	1980-1988	1988-1998	1996-2002	2002
Nozzles	1972-1984	1984-1990	1990-1998	1996
Photonic displays	1980-1986	1986-1998	1998-2004	2004
Bio/Chemical sensors	1980-1994	1994-1999	1999-2004	2004
RF Switches	1994-1998	1998-2001	2001-2005	2005
Rate sensors	1982-1990	1990-1998	1996-2002	2002
Micro relays	1977-1982	1993-1998	1998-2006	2006

Table 2: Application of MEMS

Product	Discovery	Evolution	Cost reduction/application expansion	Full commercialization
Pressure sensor	1954-1980	1960-1975	1975-1990	1990 – Present
Accelerometers	1974-1985	1985-1990	1990-1998	1996
Gas sensors	1988-1994	1994-1998	1998-2005	2005
Valves	1980-1988	1988-1998	1996-2002	2002
Nozzles	1972-1984	1984-1990	1990-1998	1996
Photonic displays	1980-1986	1986-1998	1998-2004	2004
Bio/Chemical sensors	1980-1994	1994-1999	1999-2004	2004
RF Switches	1994-1998	1998-2001	2001-2005	2005
Rate sensors	1982-1990	1990-1998	1996-2002	2002
Micro relays	1977-1982	1993-1998	1998-2006	2006

General purpose input/output (GPIO) pin diagram

The bump to the processing power of the latest machine has, according to its cocreator, elevated its performance to a point where it can comfortably be used as a desktop computer produced by enthusiastic users. Arch Linux ARM is a fork of Arch Linux built for ARM processors. This has a long history of being used on a wide range of products, including the Pogo plug, and on the Raspberry Pi.[7] It is fast and stable. In a Shutdown State. 5.23 V, 0.05 Amps = 0.26 W. Hence, we can see that the new Raspberry Pi B+ uses 1.21 W with just keyboard dongle versus 1.89 W for the old model B. The difference, 0.68 W might not sound like a lot, but it is 36% less power usage [Figure 6].

USB ports

The Raspberry Pi 3 has four ports, allowing you connect it to keyboards, mouse, wifi dongles, and USB sticks containing all your files. Since the ports do not provide much power, if you want to add a USB hub to the Pi, you will need to find one that comes with an external power supply [Figure 7].

GPIO header

This comprises the GPIO pins. They are a set of connections that have various functions, but their main one is to allow you to connect to the Raspberry Pi with an electronic circuit. You can then program theP to control the circuit and do some amazing things with it [Figure 8].

Ethernet port

The traditional way to the internet is through a wire called an Ethernet cable. You will find a few similar ports like this at the rear of your router at home that will let you connect the Raspberry Pi directly into it. This method is easier to set up than wifi and may provide faster internet but then limited by the length of the cable [Figure 9].

Micro SD card slot

A little SD card is used as the Raspberry Pi's hard drive. This is where the operating system will live once you are put it on there. Most computers would not be able to directly connect to a micro SD card, but you can get an adaptor that plugs into normal SD card slots [Figure 10].

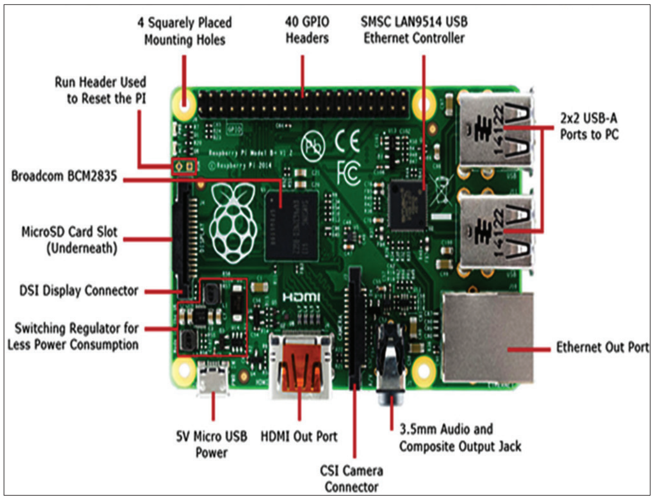

Figure 5: Raspberry Pi

Figure 6: General purpose input/output pin diagram

Figure 7: USB port

Power port

This is the kind of small charging port you might find in your smartphone. This micro USB port means you power the pi with the right kind of mobile phone charger or directly from your PC; however, it is best to use the official Raspberry Pi power supply to make sure that the Pi is getting enough power [Figure 11].

Hypertext markup language (HTML) port

This is an HTML port, the kind you will find on the back of most modern TVs and computer monitors. Use a standard HTML cable to connect your Raspberry Pi to your chosen screen, to see (and hear) whatever it is doing. You will definitely need to plug it in to set up the Pi [Figure 12].

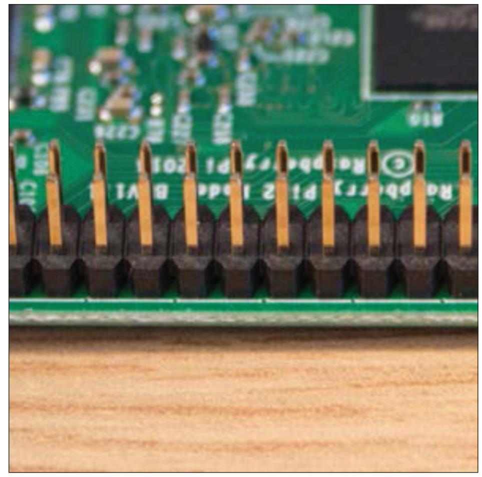

Figure 8: General purpose input/output header

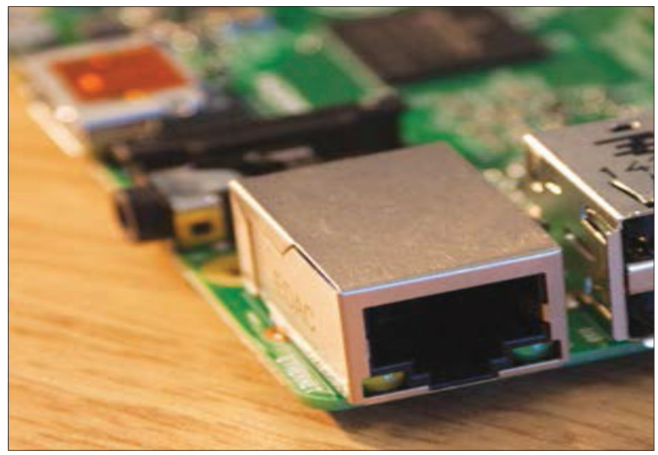

Figure 9: Ethernet port

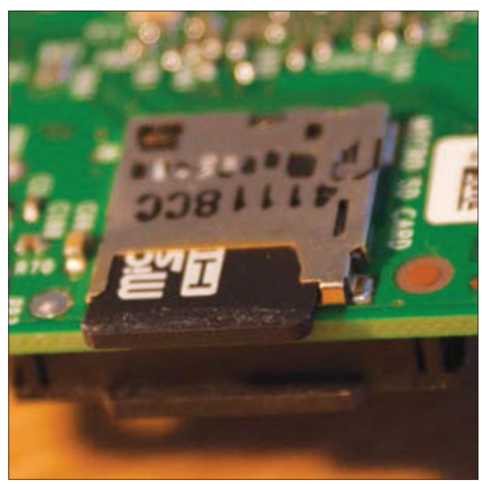

Figure 10: Micro SD card slot

Audio out

This looks like a headphone socket because that is exactly what it is. A 35 mm jack to be precise, this allows you to connect the Pi to computer speakers, or you could even plug in your favorite headphones and have a Raspberry jam [Figure 13].

Raspberry PI 3 modules

The chip of raspberry is Broadcom BCM2837 SOC. The core architecture is ARM11. CPU: 1.2 GHz low-power ARM1176JZFS applications processor, GPU: provides OpenGL ES 2.0, hardware-accelerated OpenVG, and 1080p30 H.264 high-profile decode capable of 1 gigapixel/s, 1.5 gigatexel/s or 24 GFLOPS with texture filtering and DMA infrastructure or more. The Raspberry Pi memory is 1 GB SDRAM.[8]

The operating system is booted from Micro SD card, running a version of the Linux operating system. The Ethernet is 10/100 base-T Ethernet socket, and video output is HDMI (rev 1.3 and 1.4).

Heartbeat sensor

Heartbeat sensor is designed to give digital output of heat beat when a finger is placed on it. When the heartbeat detector is working, the beat LED flashes in unison with each heartbeat.[9] This digital output can be connected to microcontroller directly to measure the beats per minute rate. It works on the principle of light modulation by blood flow through finger at each pulse. One will act as amplifiers, and another will be used as comparator. LED needs to be super bright as the light must pass through the finger and detected at other end. When the heart pumps a pulse of blood through blood vessels, finger becomes slightly more opaque so less light reaches the detector [Figure 14].

The operating voltage is +5V DC regulated and the operating current is 100 MA. The output data level is 5 V TTL level. Heartbeat detection indicated by LED and output high pulse and light source is 660 nm Super Red LED

Pins of heartbeat sensor

- 1 - +5V power supply positive input
- 2 - OUT active high output
- 3 - power supply ground

Using a sensor

Heartbeat can be measured based on optical power variation as light is scattered or absorbed during its path through the blood as the heartbeat changes.

Principle of heartbeat sensor

The heartbeat sensor is based on the principle of photo plethysmography. It measures the change in the value of blood

Processor	**Broadcom BCM2387 chipset. 1.2GHz Quad-Core ARM Cortex-A53 802.11 b/g/n Wireless LAN and Bluetooth 4.1 (Bluetooth Classic and LE)**
GPU	Dual Core VideoCore IV® Multimedia Co-Processor. Provides Open GL ES 2.0, hardware-accelerated OpenVG, and 1080p30 H.264 high-profile decode. Capable of 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs with texture filtering and DMA infrastructure
Memory	1GB LPDDR2
Operating System	Boots from Micro SD card, running a version of the Linux operating system or Windows 10 IoT
Dimensions	85 x 56 x 17mm
Power	Micro USB socket 5V1, 2.5A
Ethernet	10/100 BaseT Ethernet socket
Video Output	HDMI (rev 1.3 & 1.4 Composite RCA (PAL and NTSC)
Audio Output	Audio Output 3.5mm jack, HDMI USB 4 x USB 2.0 Connector
GPIO Connector	40-pin 2.54 mm (100 mil) expansion header: 2×20 strip
	Providing 27 GPIO pins as well as+3.3 V, +5 V and GND supply lines
Camera Connector	15-pin MIPI Camera Serial Interface (CSI-2)
Display Connector	Display Serial Interface (DSI) 15 way flat flex cable connector with two data lanes and a clock lane
Memory Card	Slot Push/pull Micro SDIO

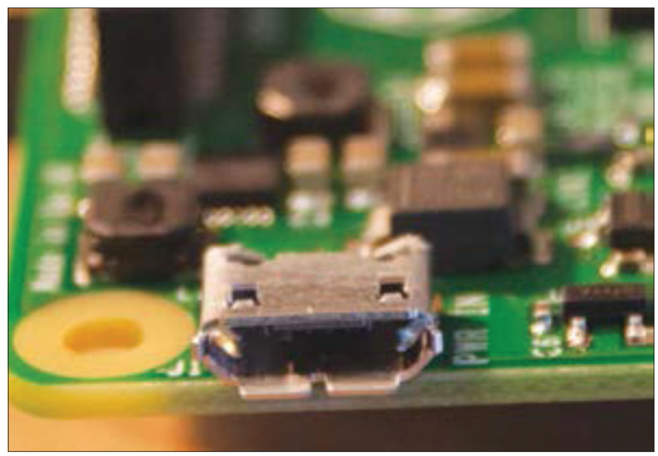

Figure 11: Power port

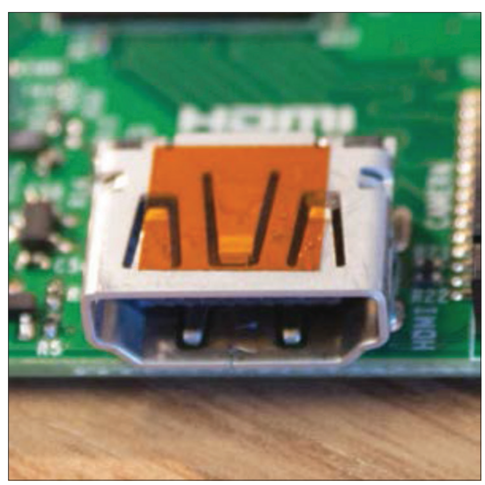

Figure 12: HTML port

through any organ of the body which causes a change in the light intensity through that organ (a vascular region). In case of application 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 heartbeat pulses.

Types

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-emitted device is reelected by the regions.

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 heartbeat pulses cause a variation in the body. When a tissue is illuminated with the light source, i.e., light emitted by the led, it either reflects the light or transmits the light and the reflected light is received by the light detector. Some of the light absorbed by the body and transmitted or the reflected is received in the light detector. The amount of light absorbed by the body depends on the blood volume in that tissue. The detector output is in the electrical signal and is proportional to heartbeat rate.

Applications of heartbeat sensor

- Two operating amplifiers are compensated internally
- Two internally compensated op amps
- Removes the necessity of dual supplies
- Permits direct sensing close to GND and VOUT
- Well suited with all method of logics
- Power drain appropriate for the operation of the battery.

Temperature sensor

The LM35 series are precision-integrated circuit temperature devices with an output voltage linearly proportional to the centigrade temperature. The LM35 device has an advantage over linear temperature sensors calibrated in Kelvin, as the user is not required to subtract a large constant voltage from the output to obtain convenient centigrade scaling. The LM35 device does not require any external calibration or trimming to provide typical accuracies of ±¼°C at room temperature and ±¾°C at −55°C to 150°C temperature range.

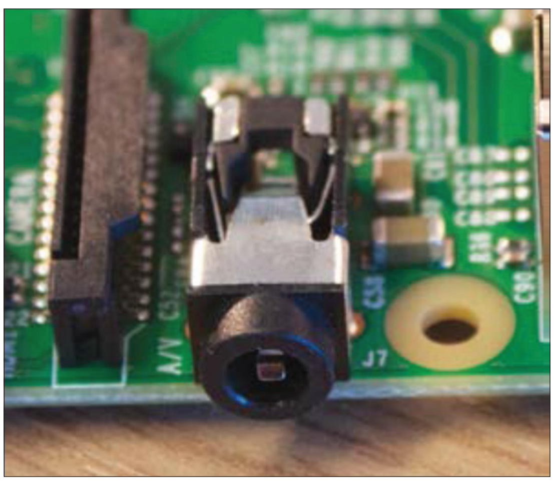

Figure 13: Audio port

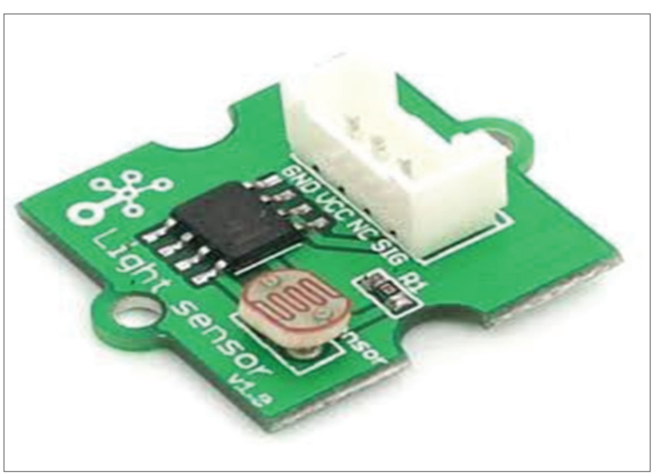

Figure 14: Heartbeat sensor

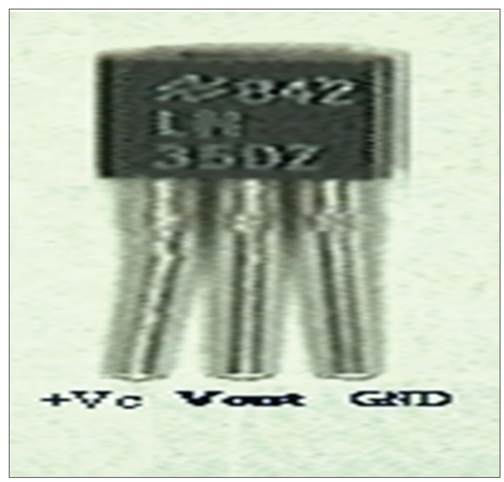

Figure 15: Temperature sensor

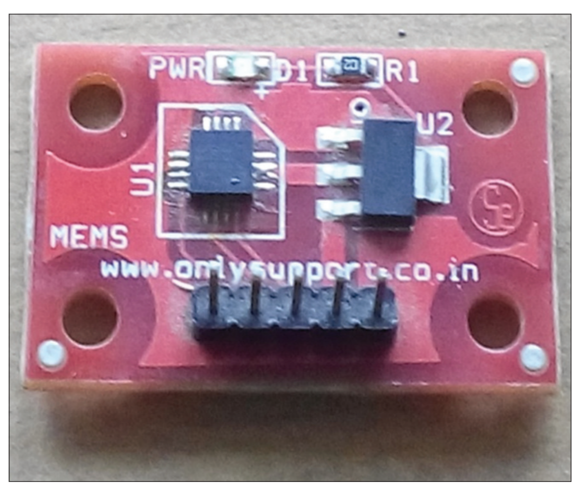

Figure 16: Microelectromechanical systems sensor

Lower cost is assured by trimming and calibration at the wafer level. The low-output impedance, linear output, and precise inherent calibration of the LM35 device make interfacing to readout or control circuitry, especially easy. The device is used with single power supplies or with plus and minus supplies. As the LM35 device draws only 60 μA from the supply, it has very low self-heating of <0.1°C in still air. LM35 is calibrated directly in °celsius (Centigrade) and is 0.5°C accuracy at +25°C and full accuracy for −55° to +150°C range. Operates from 4 to 30 V and <60 μA current drain. The LM35 is an IC sensor that can be used to measure temperature with an electrical output proportional to the temperature [Figure 15].

Features of LM35

- Calibrated directly in degree celsius (centigrade)
- Linear + 10.0 mV/°C scale factor
- 0.5°C accuracy guaranteeable (at +25°C)
- Rated for full −55°–+150°C range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts
- <60 μA current drain
- Low self-heating, 0.08°C in still air
- Nonlinearity only ±1/4°C typical
- Low impedance output, 0.1 W for 1 mA load

Applications of LM35

The LM35 can be applied easily in the same way as other integrated circuit temperature sensors. It can be glued or cemented to a surface, and its temperature will be within about 0.01°C of the surface temperature. This presumes that the ambient air temperature is almost the same as the surface temperature; if the air temperature was much higher or lower than the surface temperature, the actual temperature of the LM35 die would be at an intermediate temperature between the surface temperature and the air temperature.

MEMS sensor

The ADXL330 is a small, thin, low-power, complete 3-axis accelerometer with signal-conditioned voltage outputs, all on a single monolithic IC. The product measures acceleration with a minimum full-scale range of ±3 g.[10] It can measure the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion, shock, or vibration. The 3-axis sensing, small and low-profile package, 4 mm × 4 mm × 1.45 mm LFCSP , Low-power , 180 μA at VS = 1.8 V (typical) , single-supply operation, 1.8–3.6 V, 10,000 g shock survival, Excellent temperature stability, BW adjustment with a single capacitor per axis and ROHS/WEEE lead-free complaints [Figure 16].

The pressure range is 0–258 mm Hg and also the maximum pressure without permanent damage is 1550 mm Hg. The typical accuracy is ±1 mm Hg and temperature compensated is −20–85°C; furthermore, sensing element is SSCMRRN005PGAA5. The combined linearity and hysteresis typical ±0.25% , response time: 1 ms. The 500 μA max active current at 5V and industrial temperature range is −40°C to +85°C also available in PDIP, SOIC and TSSOP packages.

Microcontroller: Mcp3008

The Raspberry Pi is an excellent small board computer that you can use to control digital inputs and outputs. Use a simple MCP3008 analog-to-digital converter (ADC) to read up to 8 channels of analog input with 10-bit precision. Or use a fancier ADS1x15 series ADC to read 4 channels with 12–16 bit precision and a programmable gain [Figure 17].

Why We Need an ADC

The Raspberry Pi computer does not have a way to read analog inputs. It is a digital-only computer. Compare this to the Arduino, AVR or PIC microcontrollers that often have 6 or more analog inputs! Analog inputs are handy because many sensors are analog outputs, so we need a way to make the Pi analog friendly. Advanced users may note that the Raspberry Pi does have a hardware SPI interface (the Cobbler pins are labeled MISO/MOSI/SCLK/CE0/CE1). The hardware SPI interface is superfast but not included in all distributions. For that reason, we are using a bit-banged SPI implementation, so the SPI pins can be any of the Raspberry Pi's GPIOs (assuming you update the script).

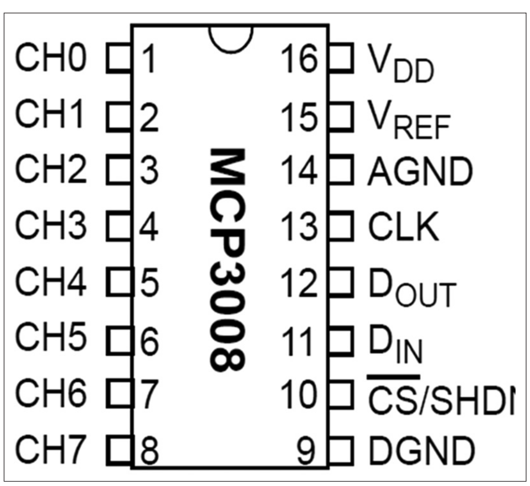

Figure 17: Microcontroller pin diagram

PHP programming

Background

Personal Home Page – C kind of scripts written in Perl language, by Ramses Lerdorf in 1995. He called the language as PHP/FI - Personal Home Page/Forms Interpreter. 1997 Version 2.0 was released and then came 3.0 and 4.0. They were called PHP simply. The recent version is PHP 5.2.3.[3]

Installation

Apache server needs to be installed first, and the configuration file for apache server will be present in the cone folder under the Apache installed directory. Set the document root to the path where the PHP files will be stored. Install PHP. Point Apache cone directory when it asks so and select the appropriate web server. Move the PHP.ini file to C: \WNDOWS directory.

Functions and parameters

- PHP functions need to be defined with keyword function
- It can have zero or more values (parameters)
- Functions may or may not return values

- If a function needs to return value, the last statement of the function should be return-return value.

Variables in PHP

- PHP variables must begin with a “$” sign
- Case sensitive ($Foo != $foo != $fOo)
- Global and locally scoped variables
  - Global variables can be used anywhere
  - Local variables restricted to a function or class
- Certain variable names reserved by PHP.

HTML

Intended to be maximally portable. HTML is logical markup and is graceful degradation of presentation. An idea promoted by early WWW and used to be more honored and it is getting better now.

Markup language

The computer language for describing syntax of tags may be used with other tools to specify rendering.

Logical markup

It described parts of the document and does not specify how to render.

Why HTML became XHTML

HTML was originally a SGML application and tags described the syntax. A DTD could check the syntax. Informal mapping from syntax to render. Multiple incompatible versions arose. IETF moves at ’net speed not web $peed. Tag abuse was rampant in the ’net. They were a plague unto the users. Big vendors agreed. To start over. To use extensible Markup Language. A re-write of SGML emphasized simplicity.

XHTML basics

It has very few changes from HTML. But very strict. All tags are in lowercase. All tags must be closed.

Paired tags

3 parts to an XHTML or HTML document

Doc type

Type of DTD used

Head

Meta information

Only <title> is required

Linux

Objectives

The objective are to explore the history of the UNIX operating system from which Linux is derived and the principles which Linux is designed on, to examine the Linux process model, to illustrate how Linux schedules processes and provides interposes communication, to look at memory management in Linux, and to explore how Linux implements file systems and manages I/O devices.

History

Linux is a modern, free operating system based on UNIX standards. First developed as a small but self-contained kernel in 1991 by Lines Thorvaldsen, with the major design goal of UNIX compatibility. Its history has been one of collaborations by many users from all around the world, corresponding almost exclusively over the Internet. It not only has been designed to run efficiently and reliably on common PC hardware but also runs on a variety of other platforms. The core Linux operating system kernel is entirely original, but it can run much existing free UNIX software, resulting in an entire UNIX-compatible operating system free from proprietary code. Many varying Linux Distributions include the kernel, applications, and management tools.

The Linux Kernel

Version 0.01 (May 1991) had no networking, ran only on 80386-compatible Intel processors and on PC hardware, had extremely limited device-drive support, and supported only the Minx file system.

Linux 1.0 (March 1994) included these new features:

Support for UNIX’s standard TCP/IP networking protocols. BSD - Compatible socket interface for networking programming driver support for running over an Ethernet. Enhanced file system. Support for a range of SCSI controllers for high-performance disk access. Extra hardware support Version 1.2 (March 1995) was the final PC-only Linux kernel.

Linux system

Linux uses many tools developed as part of Berkeley’s BSD operating system, MIT’s X Window System, and the Free Software Foundation’s GNU project. The main system libraries were started by the GNU project, with improvements provided by the Linux community. Linux networking-administration tools were derived from 4.3BSD code; recent BSD derivatives such as Free BSD have borrowed code from Linux in return. The Linux system is maintained by a loose network of developers.

Linux distributions

Standard, precompiled sets of packages, or distributions include the basic Linux system, system installation and management utilities, and ready-to-install packages of common UNIX tools. The first distributions managed these packages by simply providing a means of unpacking all the files into the

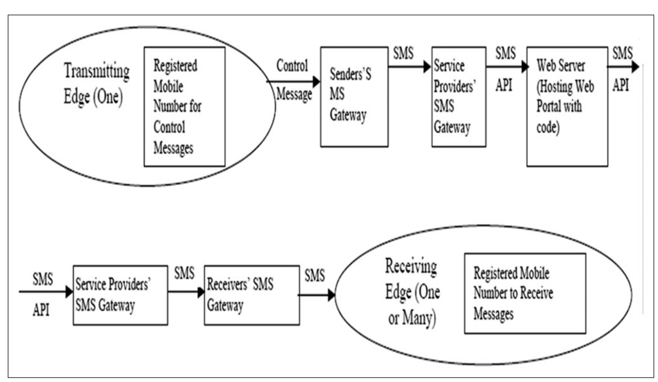

Figure 18: Web page block diagram

appropriate places; modern distributions include advanced package management. Early distributions included SLS and Slack ware. Red Hat and Debi are popular distributions from commercial and non-commercial sources, respectively, the RPM Package file format permits compatibility among the various Linux distributions.

Sending SMS to hospital through website

Application of incoming SMS to a website

I have discussed an idea to control sending of web SMS through SMS sent from cell phone. A new technique of controlling websites using SMS will provide remote control to web applications. In this technique, message as a command will be sent from a cell phone to web application. The command will generate another customized message for one or more receivers and sent through bulk SMSAPI. This concept is based on programming language conditional statements.[10] As there are commands to operate computer with different functionalities, control message will work in the similar fashion to initiate the information transfer. The control message (sent from a mobile device) consists of some parameters through which the single control message will transfer customized information to defined user or group through SMS anytime anywhere even if the sender does not have multimedia cell phone [Figure 18].

Cost factor and benefits

Cost of communication is never negligible even if single SMS is required to be sent from one cell phone to another. Here, I discussed about receiving SMS through a website and sending SMS to cell phones based on the above said criteria.[7] Therefore, it is required to calculate the cost for the system to verify the expensiveness of the system with other existing systems for the same communication.

This system can be easily embedded into the existing website offering to web SMS; otherwise, the cost of designing a fresh website and buying an API to send SMS through the website will also be included in the overall cost of the system.[6] After surveying some of the websites providing the SMS API, I have generated a table, considering the requirement of 10,000 SMSs per month for an organization.

Command base code

It is a command-based code used with parameters that are validated by the program. Therefore, according to the code defined, the control message sent from authorized mobile number must look this.

cMsg: -t <tablename> -g <groupname> -m <message>

cMsg: It is a command name from where the web site after receiving the message can easily judge that it is a control message and not a simple incoming message.-t: It is a parameter to activate particular table to access group or user mobile number(s). This helps to reduce searching time for a mobile number or mobile numbers in a particular group. The table can differentiated based on students, faculties, staff members, peers, subordinates, management, etc. It also depends on different organizations.

Coding

```
#!/usr/bin/python
importRPi.GPIO as GPIO
importspidev
import serial
import time
import decimal
importoss
importsubprocess
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(16,GPIO.IN,pull_up_down=GPIO.PUD_UP)
# Open SPI bus
spi = spidev.SpiDev()
spi.open(0,0)
# Function to read SPI data from MCP3008 chip
# Channel must be an integer 0-7
defReadChannel(channel):
adc = spi.xfer2([1,(8+channel)<<4,0])
data = ((adc[1]&3) << 8) + adc[2]
return data
# Function to convert data to voltage level,
# rounded to specified number of decimal places.
defConvertVolts(data,places):
volts = (data * 3.3)/float(1023)
volts = round(volts,places)
return volts
# Function to calculate temperature from
# TMP36 data, rounded to specified
# number of decimal places.
defConvertTemp(data,places):
temp = ((data * 330)/float(1023))
temp = round(temp,places)
temp = temp *1.5
return temp
# Define sensor channels
temp_channel = 0
pot = 1
```

```
# Define delay between readings
delay = 3
HB = 0
count = 0
while True:
# Read the temperature sensor data
temp_level=ReadChannel(temp_channel)
temp_volts=ConvertVolts(temp_level,2)
temp= ConvertTemp(temp_level,2)
 X = ReadChannel(channel)
 Y = ReadChannel(channe2)
 Z = ReadChannel(channe3)
 P = ReadChannel(pot)
if(GPIO.input(16)==1):
count = count+1
else:
count = 0
# Print out results
print "----------------------------------"
print("Temp: {}".format(temp))
print("Heartbeat: {} ".format(count))
print("X-Direction: {}".format(X))
print("Y-Direction: {} ".format(Y))
print("Z-Direction: {} ".format(Z))
print "-----------------------------------"
time.sleep(1)
f = open('log.txt','w')
f.write('Temp:')
f.write(str(temp))
f.write('\n')
f.write('Heartbeat:')
f.write('\n')
f.write(str(count)
f.write('X-Direction:')
f.write(str(X))
f.write('\n')
f.write('\n')
f.write('Y-Direction:')
f.write(str(Y))
f.write('\n')
f.write('\n')
f.write('Z-Direction:')
f.write(str(Z))
f.close()
```

CONCLUSIONS

The developed solution accomplishes the objective of controlling a website to send SMS through a control SMS sent from mobile phone. As the third party, SMS gateway is required for this application, so the efficiency not only depends on how fast web application processes the command sent through mobile phone or cell phone but also depends on the services provided by the gateway.

REFERENCES

1. Beck RL, Bradley ML, Wood BL. Remote sensing and human health: New sensors and new opportunities. Emerg Infect Dis 2000;6:217.
2. Gubbi J, Buyya R, Marusic S, Palaniswami M. Internet of things (IoT): A vision, architectural elements, and future directions. Future Gen Comput Syst 2013;29:1645-60.
3. Chenetal M. Body Area Networks: A survey. Mobile Netw Appl 2011;16:171-93.
4. Alemdar H, Ersoy C. Wireless sensor networks for healthcare: A survey. Comput Netw 2010;54:2688-710.
5. Dharetal SK. Enabling smartphone as gateway to wireless sensor network. Recent Advances in Information Technology. Switzerland: Springer; 2014. p. 19-26.
6. Rodrguez CC, Riveill M. e-Health Monitoring Applications: What about Data Quality? In Proceedings of the Health Ambiant Information Systems Workshop; 2010.
7. Paschouetal M. Health internet of things: Metrics and methods for efficient data transfer. Simul Model Pract Theory 2013;34:186-99.
8. Faludi R. Building Wireless Sensor Networks: With ZigBee, XBee, Arduino and Processing. O'Reilly Media Inc.; 2010.
9. Paul SH. Modified protothreads for embedded systems. J Comput Sci Coll 2012;28:177-84.
10. Rohokale VM, Neeli RP, Prasad R. A cooperative Internet of Things (IoT) for rural healthcare monitoring and control. IEEE 2nd International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace and Electronics Systems Technology (Wireless VITAE); 2011.

Cite this article: Gayathri C, Venkatanarayanan S, Priya RK, Kannan M. Internet of Things-Based Health Monitoring System Using Microelectromechanical Systems' Sensor. Asian J Appl Res 2018;4(2):44-55.

Source of Support: Nil,
Conflict of Interest: None declared.