SRI VENKATACHALAPATHY POLYTECHNIC COLLEGE

VADAKUCHIPALAYAM, KAPPIYAMPULIYUR, VILLUPURAM - 605 601.

APRIL 2023

PROJECT WORK ON

IOT IV BAG MONITORING AND ALERT SYSTEM

This project submitted to the Directorate of Technical Education in practical fulfillment of the requirement for the award of the

DIPLOMA IN ELECTRICAL AND ELECTRONICS ENGINEERING

A PROJECT REPORT

SUBMITTED BY

21306953	SABARISH S
21306954	SANJAY R
21306955	SANTHOSH KUMAR S
21306957	SARAVANA KUMAR S
21306958	SARAVANAKUMAR S

GUIDED BY

Mr.S.UDAYAKUMAR M.Tech., HOD

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

APRIL 2023

SRI VENKATACHALAPATHY POLYTECHNIC COLLEGE

VADAKUCHIPALAYAM, KAPPIYAMPULIYUR, VILLUPURAM - 605 601.

APRIL 2023

BONAFIDE CERTIFICATE

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

Certified that this is Bonafide record of the project work done by

Mr/~~Ms~~.....Saravanakumar.S..... during the

academic year 2022-2023.

In practical fulfillment of the requirements for the **Diploma in Electrical and Electronics Engineering of the Board of Technical Education, Tamilnadu.**

Register No: ...2130695 8.....

GUIDE **HEAD OF THE DEPARTMENT**

Submitted for the Board Examination held on .....23/05/2023.....

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

Several people have directly or indirectly contributed towards the development and success of this project. It is indeed a great pleasure to acknowledgement the help of all these individuals.

Our sincere thanks to our beloved Principal **Mr.K.MANIKANDAN M.E.,** for his kind and consistent patronage.

We would like to express our deep felt thanks to **Mr.S.UDAYAKUMAR M.Tech.,** Head of the department of E EE for his advice and for having taken great pains in providing excellent laboratory facilities for the successful completion of the project.

To begin, we express our heart full gratitude and sincere thanks to our beloved guide **Mr.S.UDAYAKUMAR M.Tech.,** HOD department of EEE for her most gracious encouragement, very valuable suggestion and motivation throughout the development of this project.

We will be failing in our duty, if we do not acknowledge the timely help rendered by **Mr.U.KRISHNAMOORTHI B.E.,** Lecturer of EEE department for helping us in this project work.

We also express our thanks to **Mr.A.MANI** Lab assistant for their kind patience and helping us in this project work.

We thank our parents and friends for their continuous support towards a great academic endeavor. And above all we thank god for being with us for the success of our project, we would like to thanks all those whom we have failed to acknowledge…

TABLE OF CONTENT

I. LIST OF SYMBOLS

II. ABSTRACT

III. INTRODUCTION

IV. BLOCK DIAGRAM

V. CIRCUIT DIAGRAM

VI. MICROCONTROLLER

VII. SENSORS

VIII. POWER SUPPLY

IX. SOURCE CODE

X. CONCLUSION

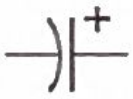
XI. REFERENCE

LIST OF SYMBOLS

LIST OF SYMBOLS

1) Capacitor

2) Capacitor, polarized



3) Capacitor, variable

4) Diode

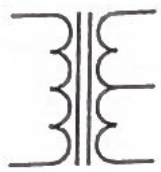
5) Zener diode

6) LED

7) Photodiode

8) Inductor

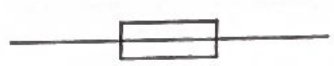
9) Transformer



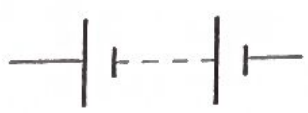
10) NPN transistor

11) Earth or ground

12) Fuse

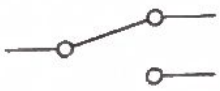


13) Battery

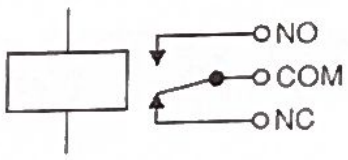


14) SPST (on-off switch)

15) SPDT (2-way switch)

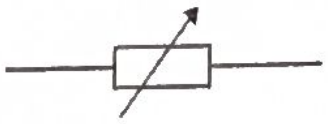


16) Relay



17) Resistor

18) Variable resistor



ABSTRACT

ABSTRACT:

During the peak of the Covid-19 Pandemic, healthcare professionals found themselves spread thin among the ever-increasing wave of incoming patients. In suchtimes, it is not possible for frontline workers to monitor and tend to each and every patient personally.Intravenous therapy is a medical technique used to deliver fluids, medications and nutrition directly into a person's vein. IV therapy is commonly used for rehydration and to provide nutrients and is crucial to help an individual with making a speedy recovery. However, IV drips need to be regularly monitored and replaced. The flow of the fluid also needs to be metered depending on the patient and their ailment. This IoT Intravenous Fluid Monitoring uses a weight sensor to detect as the fluid level in the IV Infusion bottle goes down and transmits the data over IoT. Once the system detects that the bottle has gone empty, it sends an alert over IoT.The system makes use of a Weight Sensor with an Atmega microcontroller and Wifi transmitter and LCD display to achieve this functionality. This allows for an automated and robust IV monitoring system. The Weight Sensor is attached to a small stand. The stand is fabricated with a cross section at bottom to balance it. A small rod stretching from the top allows user to suspend the weight sensor hook on the stand. The weight sensor is used to measure the weight of empty IV bag at first. This is considered as empty weight. When the IV bag is suspended onto the sensor stand, it keeps on dripping until the fluid runs out. The weight sensor value is constantly transmitted to atmega microcontroller. The controller constantly processes this data and processes it. The current level of IV bag is parallely displayed on an LCD display. Also this data is transmitted on IOT server via Wifi Module. This level is displayed on IOT server online. As soon as the level falls below certain level it LCD display as well as online dashboard displays as bag empty.

INTRODUCTION

INTRODUCTION

The Internet of Things is considered now as one of the feasible solutions for any remote value tracking especially in the field of health monitoring. Internet of Things (IoT) development brings new opportunities in many applications, including smart cities and smart healthcare. Currently, the primary usage of the IoT in healthcare can be categorized as remote monitoring and real-time health systems. Controlling and managing dire situations, such as the one in 2020 when the coronavirus dis-ease (COVID-19) took over the world, can be achieved with the help of IoT systems, without imposing severe restrictions on people and industries. COVID-19 causes respiratory symptoms and appears to be more contagious in comparison to SARS in 2003. Health is always a major concern in every growth the human race is advancing in terms of technology. Like the recent corona virus attack that has ruined the economy of China to an extent is an example how health care has become of major importance.

Remote patient monitoring allows you to stay connected with the patient beyond an initial virtual visit or screening.It allows you to monitor changes in symptoms or underlying health conditions and risk factors with surveys and connected devices, and to provide educational content. It enables your care team to follow protocols to intervene timely and appropriately as the patient's condition changes. The system involves a mobile app or web browser. People who show symptoms of the viral infection can get enrolled with AIIMS-Rishikesh. The hospital will assess the complaints and, if found necessary, will provide the patient a health monitoring kit to monitor his or her parameters regularly. Patient health parameters, along with patient location, are uploaded on a regular basis on to a centralized command and control center on cloud using either the patient's mobile phone or an integral GSM SIM card. The use of cloud will facilitate seamless scaling of the database of COVID-19 suspects/patients, BEL said. When the health parameters exceed the threshold, an alertis sent to medical officers and healthcare workers. It will also show the severity ofthe patient's condition using color codes.

The Internet of things is the inter-connection of devices, apps, sensors and network connectivity that enhances these entities to gather and exchange data. The distinguishing characteristic of Internet of Things in the healthcare system is the constant monitoring a patient through checking various parameters and also infers a good result from the history of such constant monitoring.

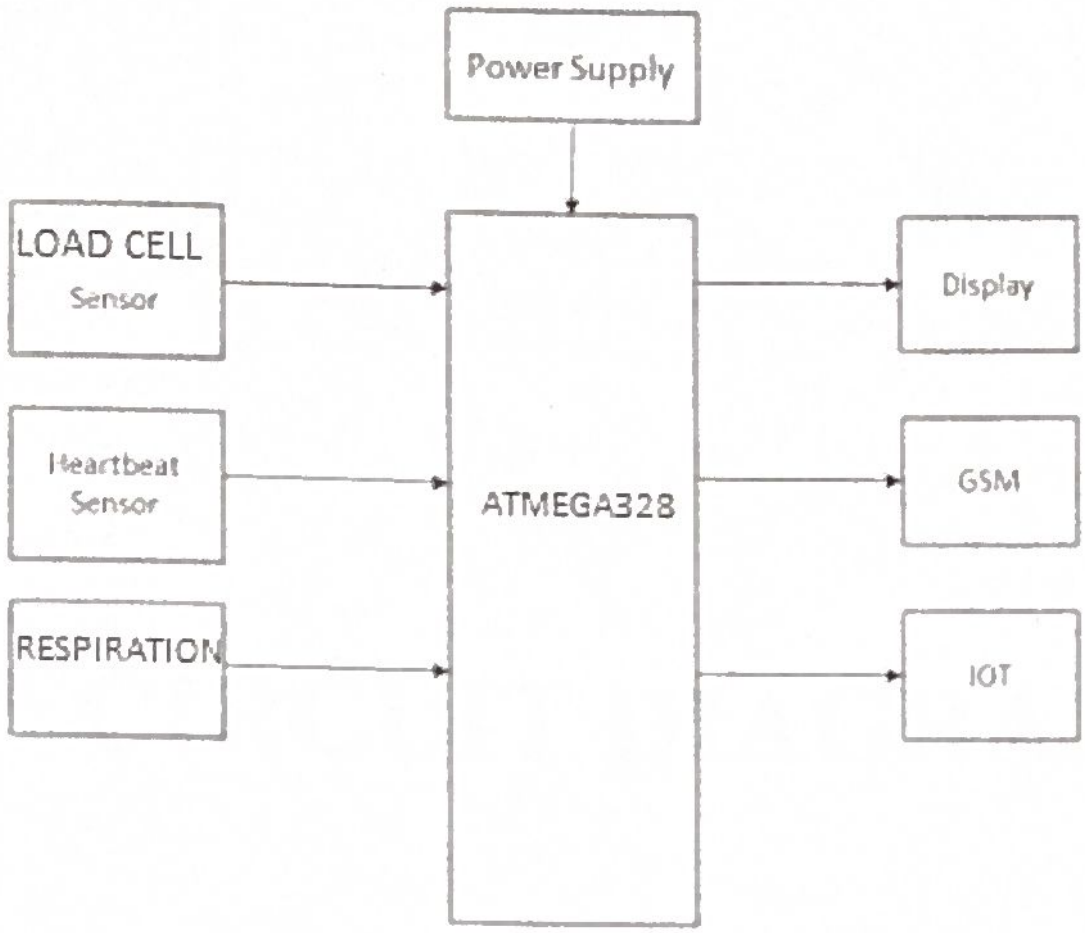
Many such devices equipped with medical sensors are present in the ICUs now-a-days. There could be instances where the doctor couldn't be alerted in time when there is an emergency, despite of 24 hours of monitoring. Also there might be hurdles in sharing the data and information with the specialist doctors and the concerned family members and relatives. The technology that enhances these features is already available but is not accessible and affordable by most of the people in developing countries such as India. Hence these solutions to these problems can be just a simple extension to the current devices which don't have these facilities. This paper demonstrates a Remote Health Monitoring System controlled by ATMEGA328 .328 ATMEGA is a small single-board microcontroller made to enhance the basic computer science education in colleges and developing nations. In this paper, a system is designed to continuously monitor the vital parameters such as heart rate, blood pressure and body temperature. The information is stored on a cloud server database and can be displayed through an online website or mobile application by authorized personnel only. The idea might not be very new, but we propose an absolute and cheap method for the system using 328 The main objective of this system is to update the data online and send an alert to the doctors for any abnormality and also predict if the patient is having any disease. The former is accomplished by using MySQL db module to link 328 to the database whereas the latter is achieved by the combination of 328 and GSM module and the web interface. This system has much future scope as the data collected by monitoring is so valuable and can be used for any kind of research by the medical community.

The major aim of the paper can be summarized as following:

- To obtain the real-time medical information about a patient via IoT.
- Processing and classification of information gathered about the patient.
- To interpret and predict any disease or disorder in preliminary stage itself using the data mining techniques that will also provide the approach advantageous for decision making.
- To provide Internet of Things based healthcare solutions at anytime and anywhere

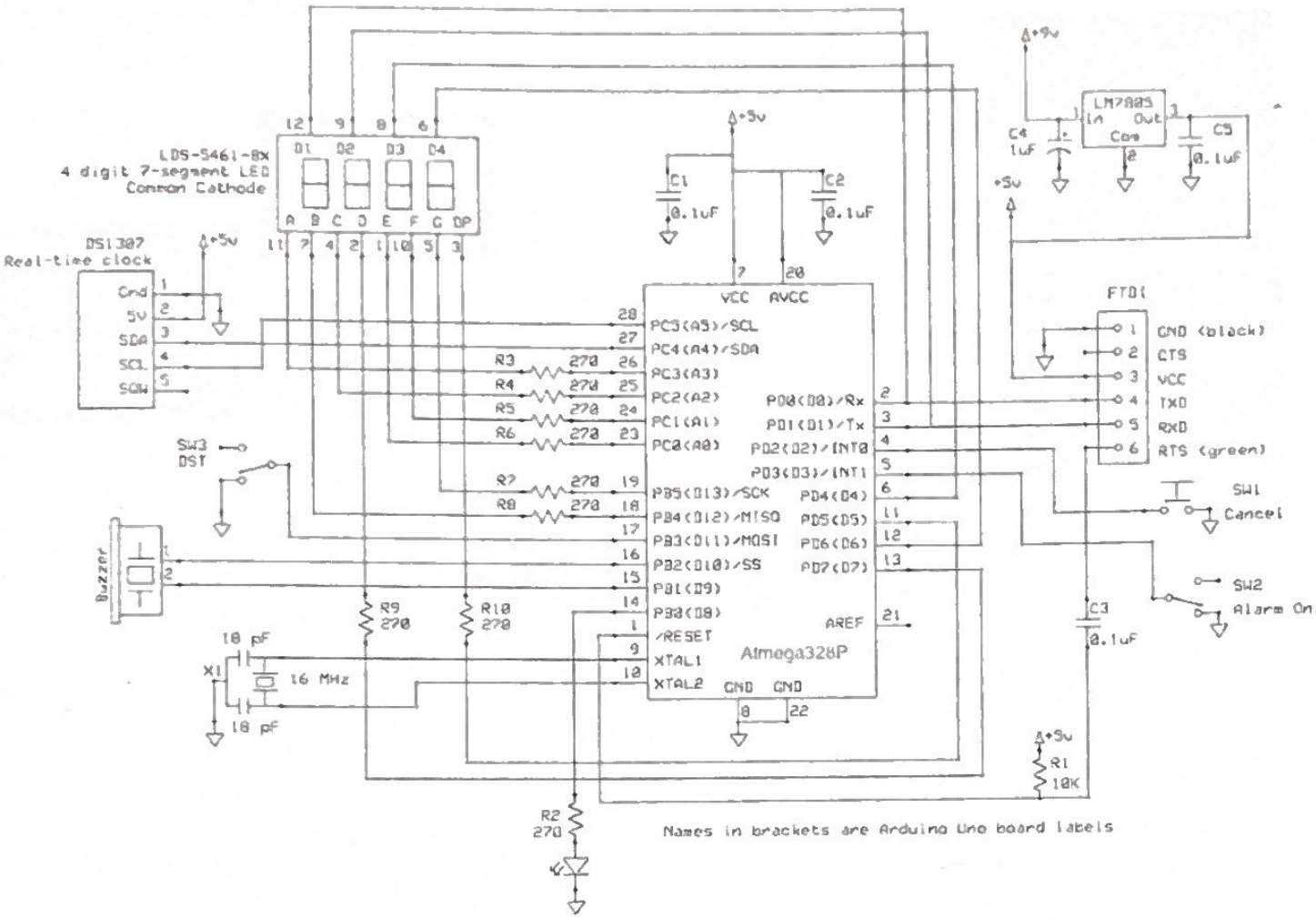
BLOCK DIAGRAM

BLOCK DIAGRAM:



CIRCUIT DIAGRAM

CIRCUIT DIAGRAM:



WORKING

WORKING

In this paper, we propose an automatic system to monitor patient's body temperature, heart rate, body movements and blood pressure. Further we extend the existing system to predict if the patient is suffering from any chronic disorder or disease using the various health parameter and various other symptoms that are obtained by the system.

In level-1, unprocessed data from various IoT devices is obtained and stored on the server. These devices include various sensors such as temperature sensor, vibration sensor, BP sensor and pulse sensor. Since some of the sensors give analog output which cannot be used by raspberry pi, we first convert the analog values into digital form and using convertor IC. Then using the raspberry pi on which Linux OS is installed, we write the code in python that reads the values from the sensors and update them into the database at regular intervals.

In level-2, the relevant information is obtained as a result from the data stored by filtering, classifying and categorizing it. This information is nothing but the patient's real-time health data and symptoms that the patient has. This information will be further used in the next level to predict if the patient is suffering from any kind of disease. This helps to make the system smart and efficient.

In level-3 the analysis/predication phase, we use data mining techniques to predict the type and nature of the disease or the disorders for which the system was designed. Using artificial intelligence can further improvise the system by making it smarter. Hence we can infer the disease or disorder by using the existing knowledge base and categorize the result in various categories such as Ideal, Normal, and With Symptoms

MICROCONTROLLER

ATMEGA 328

Features

- **High-performance, Low-power AVR®**

  - 8-bit Microcontroller

- **Advanced RISC Architecture**

– 130 Powerful Instructions – Most Single-clock Cycle Execution

– 32 x 8 General Purpose Working Registers

– Fully Static Operation

– Up to 16 MIPS Throughput at 16 MHz

– On-chip 2-cycle Multiplier

- **High Endurance Non-volatile Memory segments**

– 8K Bytes of In-System Self-programmable Flash program memory

– 512 Bytes EEPROM

– 1K Byte Internal SRAM

– Write/Erase Cycles: 10,000 Flash/100,000 EEPROM

– Data retention: 20 years at 85°C/100 years at 25°C (1)

– Optional Boot Code Section with Independent Lock Bits

• **In-System Programming by On-chip Boot Program**

– True Read-While-Write Operation

– Programming Lock for Software Security

• **Peripheral Features**

– Two 8-bit Timer/Counters with Separate Prescaler, one compare

•**Six Channels 10-bit Accuracy**

– Byte-oriented Two-wire Serial Interface

– Programmable Serial USART

– Master/Slave SPI Serial Interface

– Programmable Watchdog Timer with Separate On-chip

•**Oscillator**

– On-chip Analog Comparator

•**Special Microcontroller Features**

– Power-on Reset and Programmable Brown-out Detection

– Internal Calibrated RC Oscillator

– External and Internal Interrupt Sources

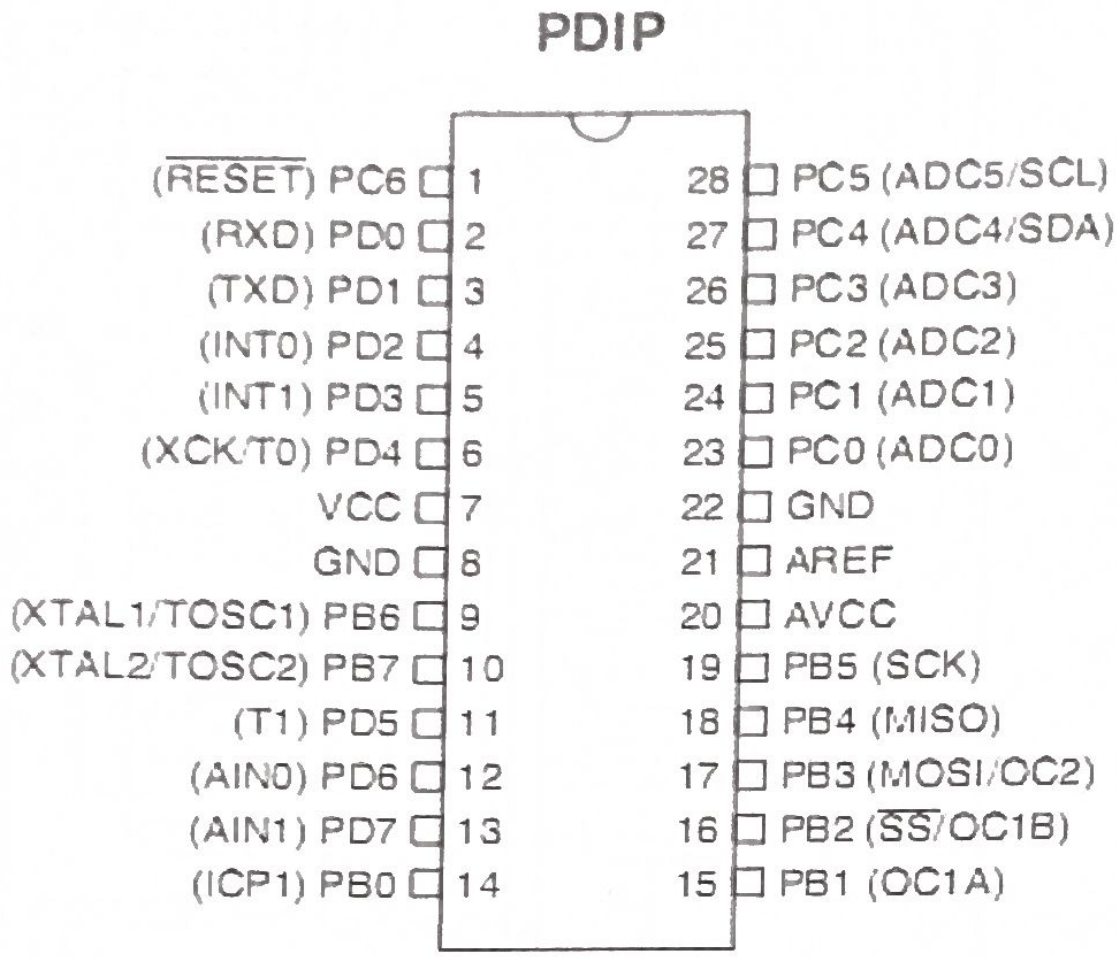
– Five Sleep Modes: Idle, ADC Noise Reduction, Power-save

•**I/O and Packages**

– 23 Programmable I/O Lines

– 28-lead PDIP, 32-lead TQFP, and 32-pad QFN/MLF

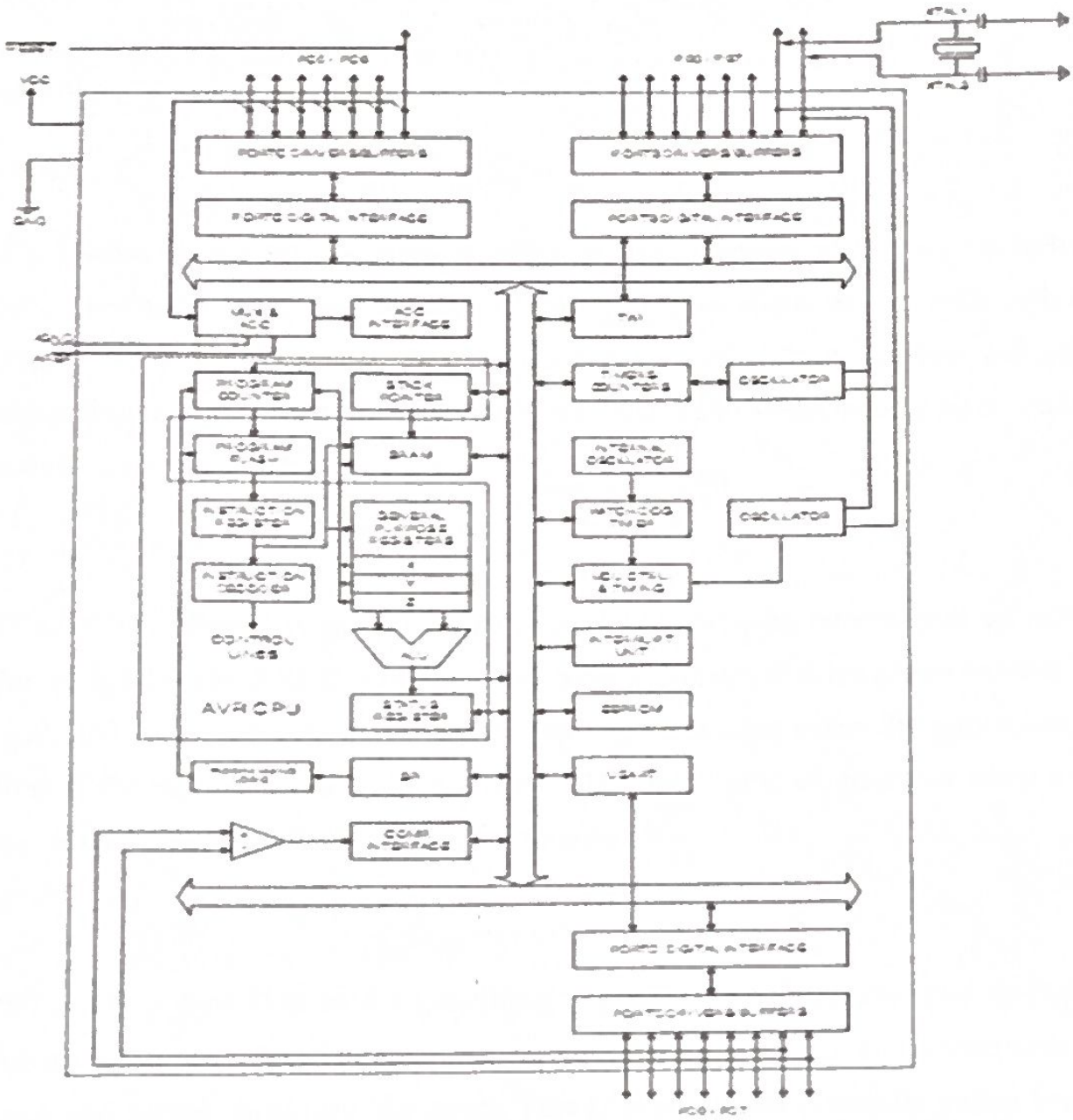
Configurations



The ATmega8 is a low-power CMOS 8-bit microcontroller based on the AVR RISC architecture.

By executing powerful instructions in a single clock cycle, the ATmega8 achieves throughputs approaching 1 MIPS per MHz, allowing the system designer to optimize power consumption versus processing speed.

BLOCK DIAGRAM



Pin Descriptions

VCC :-Digital supply voltage.

GND:- Ground.

Port B(PC7..PB0) :- is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active,even if the clock is not running.

Port C (PC5..PC0) :- Port C is an 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port D (PD7..PD0) :- Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active,even if the clock is not running.

RESET (Reset input):- A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The Shorter pulses are not guaranteed to generate a reset.

AVCC:- AVCC is the supply voltage pin for the A/D Converter, Port C (3..0), and ADC (7..6). It should be externally connected to VCC, even if the ADC is not used. If the ADC is used, it should be connected to VCC through a low-pass filter.

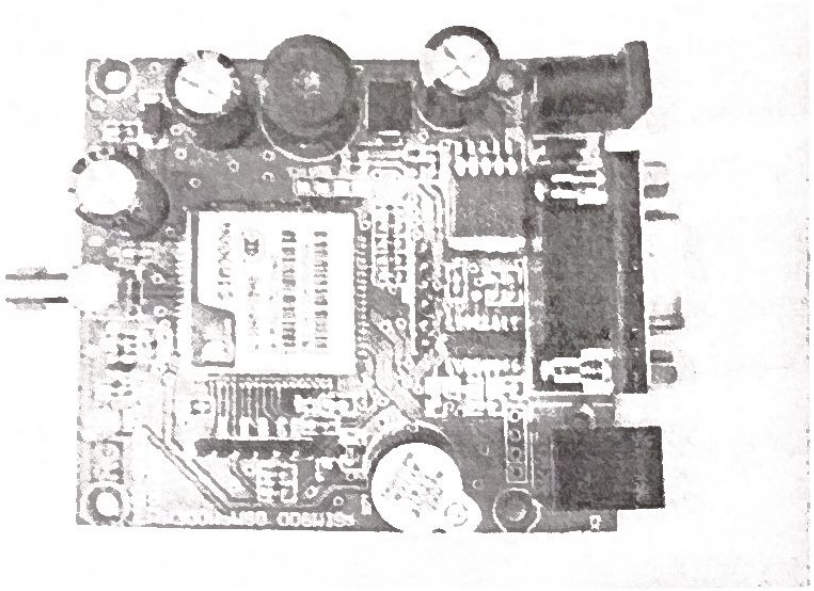
AREF:- AREF is the analog reference pin for the A/D Converter These pins are powered from the analog supply and serve as 10-bit ADC channels.

Basic Function:- The main function of the CPU core is to ensure correct program execution. The CPU must therefore be able to access memories,

perform calculations, control peripherals, and handle interrupts.

SENSORS

GSM:



This GSM Modem can accept any GSM network operator SIM card and act just like a mobile phone with its own unique phone number. Advantage of using this modem will be that you can use its RS232 port to communicate and develop embedded applications. Applications like SMS Control, data transfer, remote control and logging can be developed easily.

The modem can either be connected to PC serial port directly or to any microcontroller through MAX232. It can be used to send and receive SMS or make/receive voice calls. It can also be used in GPRS mode to connect to internet and do many applications for data logging and control. In GPRS mode you can also connect to any remote FTP server and upload files for data logging.

This GSM modem is a highly flexible plug and play quad band **SIM900A** GSM modem for direct and easy integration to RS232 applications. Supports features like Voice, SMS, Data/Fax, GPRS and integrated TCP/IP stack.

Description:

This is a GSM Modem with a simple to implement RS232, TTL Serial,Use it to send SMS, make and receive calls, and do other GSM operations by simple AT commands through a serial interface from microcontrollers and computers. It uses the highly popular SIM900A module for all its GSM operations. It comes with a standard RS232 interface which can be used to easily interface the modem to microcontrollers and computers. The modem also features a serial TTL interface option. All modem operations(sending and receiving messages, calls, etc) can be carried out by sending AT commands to this virtual serial port though a serial terminal program. Most programming languages allow sending and receiving serial commands to a serial port and can be used to write softwares that can operate the modem without the need to implement any complex interface. This makes the modem a great choice if you plan to use GSM features on a computer.SIM900A.

Applications

SMS based Remote Control & Alerts

Security Applications

Sensor Monitoring

GPRS Mode Remote Data Logging

Features

Status of Modem Indicated by LED

Simple to Use & Low Cost

On board switching type power supply regulator

RS232 output

Pinouts

Network LED

The Network LED indicates the various status of GSM module eg. Power on, Network registration & GPRS connectivity. When the modem is powered up, this NETWORK LED will blink every second. After the Modem registers in the network (takes between 10-60 seconds), this LED will blink in step of 3 seconds at slow rate. At this stage you can start using Modem for your application,showing that modem is registered with the network.

BOOTING UP THE GSM:

1. Insert the SIM card to module and lock it.

2. Connect the adapter to module and turn it ON!

3. Now wait for some time (say 1 minute) and see the blinking rate of 'status LED' (GSM module will take some time to establish connection with mobile network)

4. Once the connection is established successfully, the status LED will blink continuously every 3 seconds.

POWER SUPPLY

Block Diagram

The ac voltage, typically 220V rms, is connected to a transformer, which steps that ac voltage down to the level of the desired dc output. A diode rectifier then provides a full-wave rectified voltage that is initially filtered by a simple capacitor filter to produce a dc voltage. This resulting dc voltage usually has some ripple or ac voltage variation.

A regulator circuit removes the ripples and also remains the same dc value even if the input dc voltage varies, or the load connected to the output dc voltage changes. This voltage regulation is usually obtained using one of the popular voltage regulator IC units.

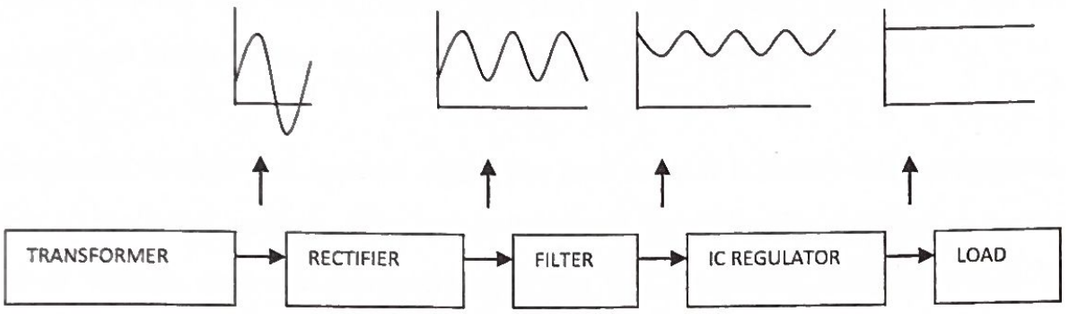


Fig 2.1 Block Diagram of Power supply

Working principle

Transformer

The potential transformer will step down the power supply voltage (0-230V) to (0-6V) level. Then the secondary of the potential transformer will be connected to the precision rectifier, which is constructed with the help of op–amp. The advantages of using precision rectifier are it will give peak voltage output as DC, rest of the circuits will give only RMS output.

Bridge rectifier

When four diodes are connected as shown in figure, the circuit is called as bridge rectifier. The input to the circuit is applied to the diagonally opposite corners of the network, and the output is taken from the remaining two corners.

Let us assume that the transformer is working properly and there is a positive potential, at point A and a negative potential at point B. the positive potential at point A will forward bias D3 and reverse bias D4.

The negative potential at point B will forward bias D1 and reverse D2. At this time D3 and D1 are forward biased and will allow current flow to pass through them; D4 and D2 are reverse biased and will block current flow.

The maximum voltage that appears across the load resistor is nearly-but never exceeds-500 v0lts, as result of the small voltage drop across the diode. In the bridge rectifier shown in view B, the maximum voltage that can be rectified is the full secondary voltage, which is 1000 volts. Therefore, the peak output voltage across the load resistor is nearly 1000 volts. With both circuits using the same transformer, the bridge rectifier circuit produces a higher output voltage than the conventional full-wave rectifier circuit.

IC voltage regulators

Voltage regulators comprise a class of widely used ICs. Regulator IC units contain the circuitry for reference source, comparator amplifier, control device, and overload protection all in a single IC. IC units provide regulation of either a fixed positive voltage, a fixed negative voltage, or an adjustably set voltage. The regulators can be selected for operation with load currents from hundreds of milli amperes to tens of amperes, corresponding to power ratings from milli watts to tens of watts.

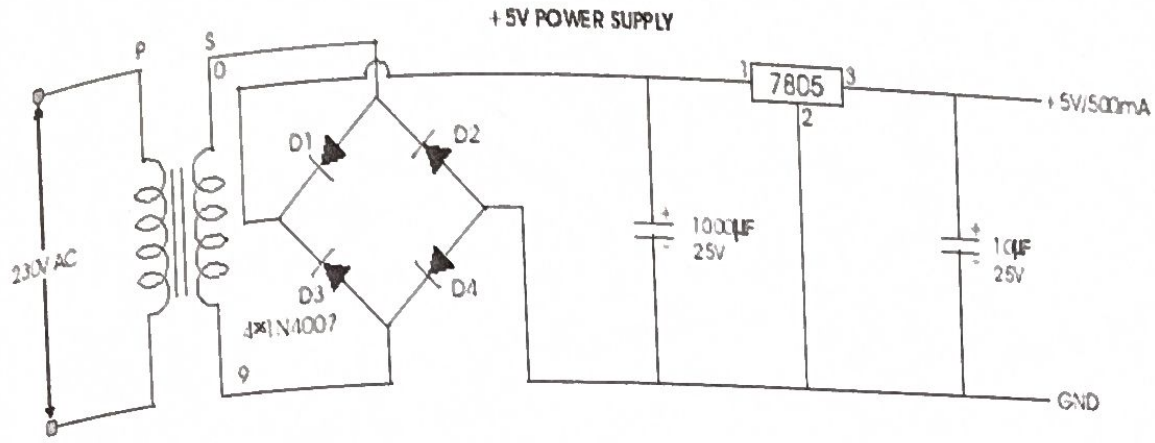


Fig 2.2 Circuit Diagram of Power Supply

A fixed three-terminal voltage regulator has an unregulated dc input voltage, Vi, applied to one input terminal, a regulated dc output voltage, Vo, from a second terminal, with the third terminal connected to ground.

The series 78 regulators provide fixed positive regulated voltages from 5 to 24 volts. Similarly, the series 79 regulators provide fixed negative regulated voltages from 5 to 24 volts.

- For ICs, microcontroller, LCD --------- 5 volts
- For alarm circuit, op-amp, relay circuits ---------- 12 volts

SOURCE CODE

```
#include<avr/io.h>
#include<util/delay.h>
#include <avr/interrupt.h>
#include <avr/eeprom.h>

#define F_CPU 8000000UL
#define USART_BAUDRATE 9600 // Baud Rate value
#define BAUD_PRESCALE (((F_CPU / (USART_BAUDRATE *
16UL))) - 1)

void UART_Transmit_string( char *string );
void UART_Transmit_char( unsigned char data );
void gsm_AT_cmd();
void master_num();
void USART_Init();
void user_num();
void check_mobilenumber();
void check_mobilenumber_sms();
unsigned int
a,b,i,f,t,d,r,s=0,b,Q,A,B,C,D=0,K=0,check=0,correct=0,n=0;
unsigned char
*sms="+CMT",*call="+CLI",*signal="+CRE",ready,item,count,rec,star
t=0,stop=0,msg=0,res=0,correctno=0,master=0,user=0,value;
unsigned char info[65],Mob_no[10],user_no[10];

void UART_Transmit_char( unsigned char data )
{
 /* Wait for empty transmit buffer */
 while ( !( UCSRA & (1<<UDRE)) )
 ;/* Put data into buffer, sends the data */
```

```
 UDR = data;
}
void UART_Transmit_string( char string[] )
{
 int i=0;
 while ( string[i] > 0)
 UART_Transmit_char(string[i++]);
}
ISR (USART_RXC_vect)
{
   char value ;
    value = UDR;                    // Read data from the RX buffer
            if(value==*signal)      //READ SIGNAL
            {
                    ready++;
                    *signal++;
                    if(ready==4)
                    {
                    for(i=0;i<ready;i++)
                    {       *signal--;      }
                            check=0;
                            start=1;
                            res=1;
                            s=0;
                            ready=0;
                    }
            }

            if(value==*sms)         //READ SMS
            {
```

```
            item++;
            *sms++;
            if(item==4)
            {
            for(i=0;i<item;i++)
            {       *sms--;         }
                    check=0;
                    start=1;
                    msg=1;
                    s=0;
                    item=0;
            }
        }

            if(value==*call)        //READ INCOMING CALL
        {
            count++;
            *call++;

            if(count==4)
            {
                for(i=0;i<count;i++)
                {
                        *call--;
                }
                check=0;
                n=0;
                s=s+1;
                start=1;
                count=0;
```

```
                }
        }
            if(start==1)
            {
            info[check++]=value; //BUFFER DATA++
            }

                if(correctno==1)
                {
                n=1;
                if(s==4&&r==0)
                {
                UART_Transmit_string("ATH\r");
                _delay_ms(4000);
                A=1;eeprom_write_byte ((uint8_t*) 2, 0xEE);
                UART_Transmit_string("ATH\r");
                _delay_ms(4000);
                r=1;s=0;s=0;correctno=0;
                _delay_ms(1000);
                check=0;

        info[8]=0;info[9]=0;info[10]=0;info[11]=0;info[12]=0;
                }
                if(s==2&&r==1)
                {
                UART_Transmit_string("ATH\r");
                _delay_ms(4000);
                C=1;eeprom_write_byte ((uint8_t*) 2, 0xCC);
                UART_Transmit_string("ATH\r");
                _delay_ms(4000);
```

```
                    PORTC &= ~(1<<5);
                    r=0;s=0;s=0;correctno=0;
                    _delay_ms(1000);
                    check=0;

        info[8]=0;info[9]=0;info[10]=0;info[11]=0;info[12]=0;
                    }
                    }
                    if(correctno==0&&s>=1&&n==0)
                    {
                    UART_Transmit_string("ATH\r");
                    _delay_ms(4000);
                    UART_Transmit_string("ATH\r");
                    _delay_ms(4000);
            s=0;s=0;d=0;
                    _delay_ms(1000);
                    }
}
void master_num()
{
UART_Transmit_string("AT+CMGS=\""); _delay_ms(100);
UART_Transmit_char(Mob_no[0]);      _delay_ms(100);
UART_Transmit_char(Mob_no[1]);      _delay_ms(100);
UART_Transmit_char(Mob_no[2]);      _delay_ms(100);
UART_Transmit_char(Mob_no[3]);      _delay_ms(100);
UART_Transmit_char(Mob_no[4]);      _delay_ms(100);
UART_Transmit_char(Mob_no[5]);      _delay_ms(100);
UART_Transmit_char(Mob_no[6]);      _delay_ms(100);
UART_Transmit_char(Mob_no[7]);      _delay_ms(100);
UART_Transmit_char(Mob_no[8]);      _delay_ms(100);
```

```
UART_Transmit_char(Mob_no[9]);          _delay_ms(100);
UART_Transmit_string("\"\r");       _delay_ms(100);
}
void user_num()
{
UART_Transmit_string("AT+CMGS=\""); _delay_ms(100);
UART_Transmit_char(user_no[0]);          _delay_ms(100);
UART_Transmit_char(user_no[1]);          _delay_ms(100);
UART_Transmit_char(user_no[2]);          _delay_ms(100);
UART_Transmit_char(user_no[3]);          _delay_ms(100);
UART_Transmit_char(user_no[4]);          _delay_ms(100);
UART_Transmit_char(user_no[5]);          _delay_ms(100);
UART_Transmit_char(user_no[6]);          _delay_ms(100);
UART_Transmit_char(user_no[7]);          _delay_ms(100);
UART_Transmit_char(user_no[8]);          _delay_ms(100);
UART_Transmit_char(user_no[9]);          _delay_ms(100);
UART_Transmit_string("\"\r");       _delay_ms(100);
}
void check_mobilenumber_sms()
{
if
 ( Mob_no[0]==info[7]&&          //read master number in call
     Mob_no[1]==info[8]&&
  Mob_no[2]==info[9]&&
  Mob_no[3]==info[10]&&
     Mob_no[4]==info[11]&&
     Mob_no[5]==info[12]&&
     Mob_no[6]==info[13]&&
     Mob_no[7]==info[14]&&
     Mob_no[8]==info[15]&&
```

```
  Mob_no[9]==info[16])
      {
      master=1;
      }
      else
      {
      master=0;
      }
if
 ( user_no[0]==info[7]&&          //read master number in call
      user_no[1]==info[8]&&
   user_no[2]==info[9]&&
   user_no[3]==info[10]&&
      user_no[4]==info[11]&&
      user_no[5]==info[12]&&
      user_no[6]==info[13]&&
      user_no[7]==info[14]&&
      user_no[8]==info[15]&&
   user_no[9]==info[16])
      {
      user=1;
      }
      else
      {
      user=0;
      }
}
void gsm_AT_cmd()
{
      UART_Transmit_string("AT\r");
```

```
    _delay_ms(8000);
  UART_Transmit_string("ATE0\r");
    _delay_ms(8000);
    UART_Transmit_string("AT+CLIP=1\r");
    _delay_ms(8000);
    UART_Transmit_string("AT+CMGF=1\r");
    _delay_ms(8000);
    UART_Transmit_string("AT+CMGF=1\r");
    _delay_ms(8000);
    UART_Transmit_string("AT+CNMI=2,2,0,0,0\r");
    _delay_ms(9000);
}
```

CONCLUSION

CONCLUSION

This System allows:

- Doctors to monitor patients remotely without risk of infection
- A single doctor over 500 patients at a time.
- Doctor gets instant alert in case of health fluctuations of emergency.

The system is mounted at patient bedside and constantly transmits patient health data over the internet so that doctors can monitor multiple patients remotely and attend the desired patient urgently when needed This project fulfills the aims to significantly reduce the risk of exposure in healthcare workers. It is also expected to reduce the increasing demand of PPE (personnel protection equipment) and logistics. stays in quarantine period reduced for conventional routine examinations and most important that the health can be monitored and disease diagnosed by any doctor at any distance.

The weight of empty IV bag at first. This is considered as empty weight. When the IV bag is suspended onto the sensor stand, it keeps on dripping until the fluid runs out. The weight sensor value is constantly transmitted to atmega microcontroller. The controller constantly processes this data and processes it. The current level of IV bag is parallely displayed on an LCD display. Also this data is transmitted on IOT server via Wifi Module. This level is displayed on IOT server online. As soon as the level falls below certain level it LCD display as well as Online dashboard displays as bag empty.

REFERENCE

REFERENCE:

[1] B. G. Ahn, Y. H. Noh, and D. U. Jeong. Smart chair based on multi heart rate detection system. In 2015 IEEE SENSORS, pages 1–4, Nov 2015.

[2] S. H. Almotiri, M. A. Khan, and M. A. Alghamdi. Mobile health (m-health) system in the context of iot. In 2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW), pages 39–42, Aug 2016.

[3] T. S. Barger, D. E. Brown, and M. Alwan. Healthstatus monitoring through analysis of behavioral patterns. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 5(1):22–27, Jan 2005. ISSN 1083-4427.

[4] I. Chiuchisan, H. N. Costin, and O. Geman. Adopting the internet of things technologies in health care systems. In 2014 International Conference and Exposition on Electrical and Power Engineering (EPE), pages 532– 535, Oct 2014.

[5] A. Dwivedi, R. K. Bali, M. A. Belsis, R. N. G. Naguib,P. Every, and N. S. Nassar. Towards a practical healthcare information security model for healthcare institutions. In 4th International IEEE EMBS Special Topic Conference on Information Technology Applications in Biomedicine, 2003., pages 114–117, April 2003.

[6] M. S. D. Gupta, V. Patchava, and V. Menezes. Healthcare based on iot using raspberry pi. In 2015 International Conference on Green Computing and Internet of Things (ICGCIoT), pages 796–799, Oct 2015.

[7] P. Gupta, D. Agrawal, J. Chhabra, and P. K. Dhir. Iot based smart healthcare kit. In 2016 International Con- ference on Computational Techniques in Information and Communication Technologies (ICCTICT), pages 237– 242, March 2016.

[8] N. V. Lopes, F. Pinto, P. Furtado, and J. Silva. Iot architecture proposal for disabled people. In 2014 IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), pages 152–158, Oct 2014.