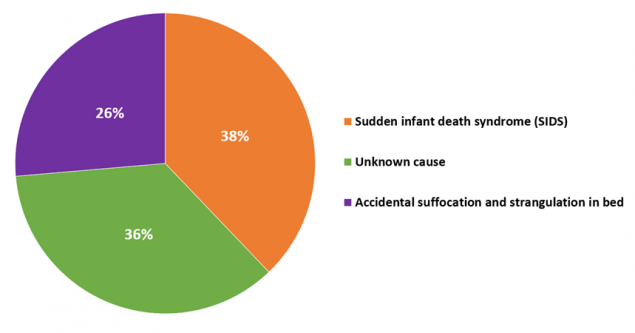
***Cloud Integrated Sudden Infant Death Syndrome Monitoring***

***Abstract:***

* *Sudden infant death syndrome (SIDS) is a disease of unknown cause. Despite recent decreases in the incidence of SIDS, SIDS is still responsible for more infant deaths in the United States than any other cause of death during infancy beyond the neonatal period.*[*1*](https://pediatrics.aappublications.org/content/105/3/650)
* *SIDS is defined as:*
* *“The sudden death of an infant under 1 year of age, which remains unexplained after a thorough case investigation, including performance of a complete autopsy, examination of the death scene, and review of the clinical history.”*

***Project Brief Description:***

* *Sudden Infant Death Syndrome is one of the major causes of death among infants during their sleep. The Wearable IoT Device acts as a wireless sensor node integrated with a Chest Belt and it has the capacity to monitor the following parameters:*
* *Body temperature*
* *Heartbeat rate*
* *Body position*

****

***PROBLEMS:***

*1.Age*

*2. Tobacco Smoke*

*3.Sleeping*

*4.Breastfeeding*

*5.Pregnancy and infant factors*

*6.Genetics*

*7.Alcohol*

*8.other*

***Problem Explanation:***

*The exact cause of SIDS is unknown. The requirement of a combination of factors including a specific underlying susceptibility. a specific time in development, and an environmental stressor has been proposed. These environmental stressors may include sleeping on the stomach or side, overheating, and exposure to tobacco smoke.*

*Accidental suffocation from bed sharing(also known as co-sleeping) or soft objects may also play a role. Another risk factor is being born before 39 weeks of gestation SIDS makes up about 80% of sudden and unexpected infant deaths (SUIDs)The other 20% of cases are often caused by infections, genetic disorders and heart problems. While child abuse  in the form of intentional suffocation may be misdiagnosed as SIDS, this is believed to make up less than 5% of cases.*

***About Components used (their Specifications and working principles)***

**Arduino Uno Board**:

* with Real-Time Application Projects. ... The **Arduino Uno board** is a microcontroller based on the



* ATmega328. It has 14 digital input/output pins in which 6 can be used as PWM outputs, a 16 MHz ceramic resonator, an ICSP header, a USB connection, 6 analog inputs, a power jack and a reset button.

***Bluetooth HC-05:***



**HC**‐**05** module is an easy to use **Bluetooth** SPP (Serial Port Protocol) module , designed for transparent wireless serial connection setup.

The **HC**-**05 Bluetooth** Module can be used in a Master or Slave configuration, making it a great solution for wireless communication.

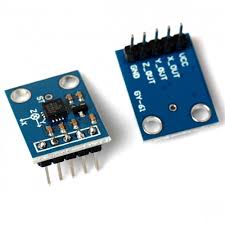
This serial port  **Bluetooth** module is fully qualified ...

***Pulse Sensor:***



***Pulse Sensor****is a well-designed plug-and-play****heart-rate sensor****for Arduino . The****sensor****clips onto a fingertip or earlobe and plugs right into Ardunio with some jumper cables. It also includes an open-source monitoring app that graphs your* ***pulse****in real time.*

***Accelerometer Sensor:***



*It can measure the static acceleration of gravity in tilt-****sensing****applications, as well as dynamic acceleration resulting from motion, shock, or vibration. The user selects the bandwidth of the****accelerometer****using the CX, CY, and CZ capacitors at the XOUT, YOUT, and ZOUT****pins****.*

*Wearable chest Sensor belt*

* *This paper presents the design and development of a wearable ubiquitous healthcare monitoring system using integrated electrocardiogram (ECG), accelerometer and oxygen saturation sensors. In this design, non-intrusive healthcare system was designed based on wireless sensor network (WSN) for wide area coverage with minimum battery power to support RF transmission. We have developed various devices such as wearable ubiquitous sensor network (USN) node, wearable chest sensor belt and wrist pulse oximeter for this system. Low power ECG, accelerometer and SpO(2) sensors board was integrated to the wearable USN node for user's health monitoring. The wearable ubiquitous healthcare monitoring system allows physiological data to be transmitted in wireless sensor network using IEEE 802.15.4 from on-body wearable sensor devices to a base-station which is connected to a server PC. Physiological data can be displayed and stored in the server PC continuously.*

***Code:***

#include <SoftwareSerial.h>

SoftwareSerial mySerial(8,9);//rx,tx

// VARIABLES

int pulsePin = A1; // Pulse Sensor purple wire connected to analog pin 0

int tempPin = A0; // Temperature Sensor connected to analog pin 1

int blinkPin = 13; // pin to blink led at each beat

int sensorValue[5] = {0,0,0,0,0};

char inbyte = 0;

String data;

// these variables are volatile because they are used during the interrupt service routine!

volatile int BPM; // used to hold the pulse rate

volatile int Signal; // holds the incoming raw data

volatile int IBI = 600; // holds the time between beats, must be seeded!

volatile boolean Pulse = false; // true when pulse wave is high, false when it's low

volatile boolean QS = false; // becomes true when Arduoino finds a beat.

void setup(){

pinMode(blinkPin,OUTPUT); // pin that will blink to your heartbeat!

Serial.begin(9600); // we agree to talk fast!

mySerial.begin(9600); //

interruptSetup(); // sets up to read Pulse Sensor signal every 2mS

// UN-COMMENT THE NEXT LINE IF YOU ARE POWERING The Pulse Sensor AT LOW VOLTAGE,

// AND APPLY THAT VOLTAGE TO THE A-REF PIN

//analogReference(EXTERNAL)

void loop(){

getSensorValues();

sendAndroidValues();

delay(20); // take a break

}

void getSensorValues()

{

// read the analog in value to the sensor array

sensorValue[0] = ( 5 \* analogRead(tempPin) \* 100.0) / 1024.0;

Serial.println(sensorValue[0]);

if (QS == true){ // Quantified Self flag is true when arduino finds a heartbeat

Serial.print("Signal=");

Serial.println(Signal); // Print pulse sensor raw data

Serial.print("HeartBeat=");

Serial.println(BPM); // Print BPM (beats per Minut)

Serial.print("Time Interval=");

Serial.println(IBI); // Print IBI (Interval between beats)

sensorValue[1] = BPM; // read the BPM to the sensor array

QS = false; // reset the Quantified Self flag for next time

}

sensorValue[2]=analogRead(A2);

sensorValue[3]=analogRead(A3);

sensorValue[4]=analogRead(A4);

}

void sendAndroidValues()

{

data="#";

data+= sensorValue[0];

data+=",";

data+= sensorValue[1];

data+=",";

data+= sensorValue[2];

data+=",";

data+= sensorValue[3];

data+=",";

data+= sensorValue[4];

data+="+";

Serial.println("data is");

Serial.println(data);

mySerial.println(data);

***Softwares:***

* Aurdino IDE
* Android Studio/MIT app inventor