EC Project

Team Members:-

1. Bachhav Aryan Kishor(210253)
2. Arpit Anand(210190)
3. Anjali Jangir(210144)

Section - A
Table No. - 12

Q1. What problem are you trying to solve, and why is it important/ interesting?

Ans. We are making a Heart Rate and SpO2 Measuring system which solved the following problems:-

1. **Convenience:** Users may find it inconvenient to visit healthcare facilities regularly for heart rate and SpO2 measurements. Our system could provide the convenience of monitoring these vital signs from the comfort of our own home, reducing the need for frequent clinic visits.
2. **Affordability:** Making the system affordable would ensure broader accessibility, allowing more people to benefit from monitoring their heart rate and SpO2 levels regularly.

Due to the afore-mentioned problems, it will be quite interesting for us to see how our project is able to replace the existing expensive methods of measuring Heart Rate and SpO2 values, as we need to go to the health centers for these measurements. Also with the help of this remote monitoring system, we can do long term monitoring which can help us identify any chronic changes in our health overtime and detect potential issues early on.
We are students of a technical institute who are interested in cutting-edge technology and innovation. Thus, making this project is of quite a relevance to us.

Q2. What are the existing solutions? Describe a few of them and list any shortcomings in them. Is your solution approach unique in some way?

Ans For instance, if we consider a scenario where the heart rate is reported as

28.8 beats per minute (bpm) and the oxygen saturation level (SpO2) is noted as 70%, such readings may be erroneous or inconsistent. However, by averaging the readings over time, say a minute or so, we obtain a more reliable assessment. In this case, the average heart rate might be around 84.5 bpm with an SpO2 of 97%, providing a more accurate representation of the individual's health status.

Despite its effectiveness in improving accuracy, this approach also has its shortcomings. One limitation is the potential delay in obtaining real-time insights, as averaging requires collecting data over a certain period. Additionally, sudden fluctuations or anomalies in vital signs may not be promptly detected, as they might be masked by the averaging process.

Q3. What resources do you require to complete the project? Give a breakup of tasks that you need to accomplish week by week to complete the project.

Ans. The components that we need are as follows:-

S. N	**Components Name**	**Quantity**
1	Arduino Nano	1
2	HC-05 Bluetooth Module	1
3	MAX30100 sensor	1
4	Resistor	3
5	Breadboard	1
6	Jumper Wires	8

Week wise breakup of tasks -

Week 1:

- Research and acquire the necessary components for the project.

- Familiarize ourselves with the basic concepts of Arduino programming, Bluetooth communication, HC05, and other components.
- Write and upload a simple Arduino program that blinks the LED light of Ardunio on and off.
- Measure and show the resistor value using a Arduino code

Week 2:

Build the circuit on a breadboard and test the MAX30102 sensor to make sure it works properly.

- Write and upload an Arduino program that reads the data from the sensor by putting a finger on it and sends it to a connected device over Bluetooth.
- Test the Bluetooth communication and make sure that the heart rate and oxygen level (SpO2) data is transmitted correctly.

Week 3:

- Connect the Bluetooth module to the mobile phone. MAX30102 sensor receiving data and displaying it on the mobile phone.
- Write the code to verify the heart rate and oxygen level data.
- Test the complete system by measuring heart rate and SpO2 level in the blood by MAX30102 sensor (Pulse Oximeter) and verifying that the system is working.

Project Description:-

We made a Heart rate and SpO2 measuring system. The system measures the person's heart rate and SpO2 content using the MAX30100 sensor, which collects the voltage values (analog input) and converts them to digital values (0/1) using its inbuilt Analog to Digital converter. The system sends the values from the Arduino board whose values can be seen on the serial monitor to the bluetooth terminal app on the mobile phone, using a Bluetooth module. The values are averaged out for every 10 inputs, so as to increase the accuracy of the system.

Components Used:-

1. **Arduino Nano-**

   It is the development board relying upon a microcontroller in it. It was used for measuring the voltage levels depending upon the sensor values.

2. **HC05 Bluetooth Module -**

   It is the Bluetooth module used for sending serial data to the Arduino board based on the sensor data.

3. **MAX30100 Sensor-**

   This sensor was used to measure the heart rate and SpO2 .

4. Jumper Wires-

Jumper wires were used for connecting different modules.

5. Resistances-

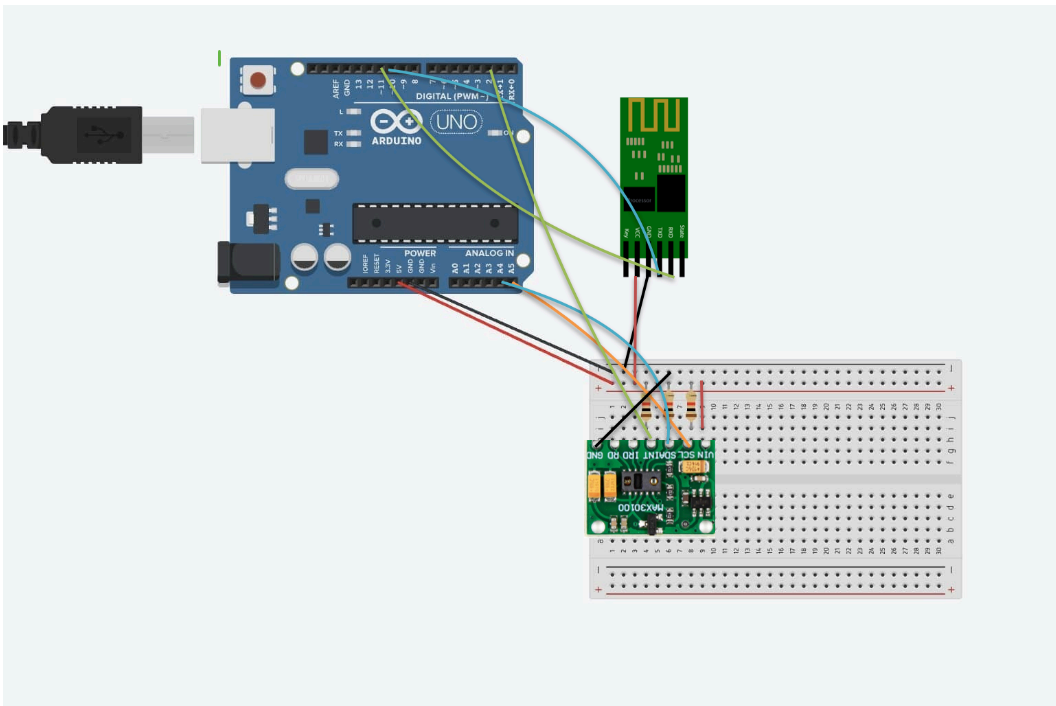
3 resistances each of 10K ohm are used.

6. Breadboard-

It was used for making the whole circuit.

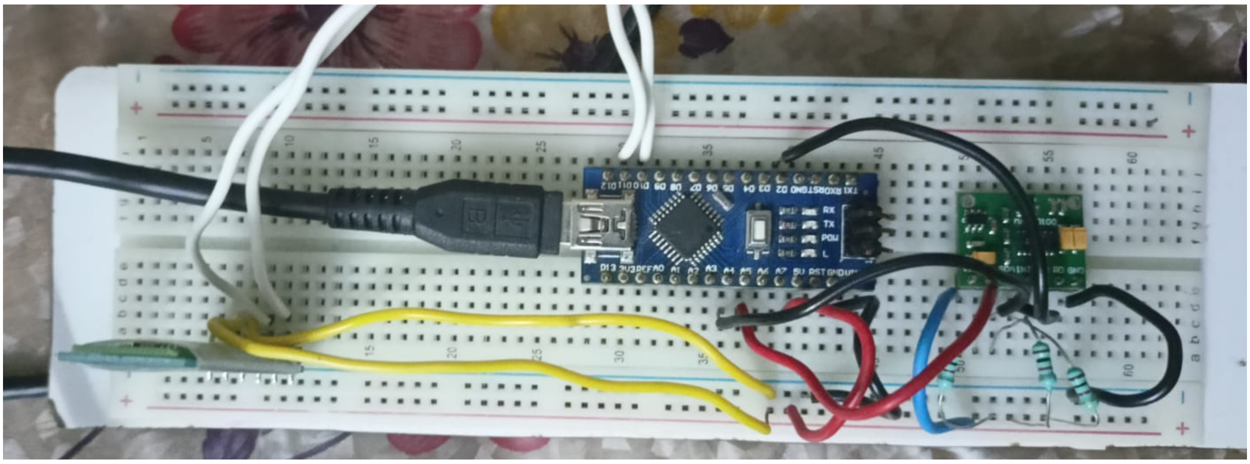
Circuit Diagram-

- The TX and RX pins of HC05 were connected to the RX and TX of the Arduino Nano. RX was directly connected to the TX of Arduino Nano, but the TX was connected to the RX of Arduino through a potential divider of $10k\Omega$ and $10k\Omega$.



- V_{CC} of the Bluetooth was connected to V_{CC} (5 V) of the Nano. The GND of the Bluetooth was connected to the GND of the Nano.
- The SCL and SDA ports of the MAX30100 sensor are connected to the A4 and A5 terminal of the

- Similarly, in the Dual Channel Relay module, the V_{CC} and GND were connected to V_{CC} (5 V) and GND of the Nano, respectively. The IN1 pin of the relay was connected to digital pin 9 (D9) of the nano.
- The Common terminal of the relay was a module connected to the ground of the 12 V supply. The NC1 terminal was connected to the negative terminal of the solenoid lock.
- The positive terminal of the lock was connected directly to the 12 V supply.
-



Working Explained-

- Software Serial Configuration: Initializes a software serial port named bt on Arduino pins 10 (RX) and 11 (TX) for communication with a Bluetooth module.
- PulseOximeter Initialization:
  PulseOximeter object named pox, which interfaces with the MAX30100 sensor.
  Sets a callback function onBeatDetected() to be executed when a beat is detected.
- Setup Function: Initializes serial communication with baud rates of 9600. Initializes the PulseOximeter sensor and checks for successful initialization. Sets the beat detection callback function.
- Loop Function: Updates sensor readings and accumulates heart rate and SpO2 values over 10 readings. Every 500 milliseconds, averages the values and prints them to the serial monitor. Transmits the averaged values via Bluetooth.

Code Used -

```
#include <Wire.h>
#include "MAX30100_PulseOximeter.h"
#include <SoftwareSerial.h>

SoftwareSerial bt(10, 11); // RX | TX
#define REPORTING_PERIOD_MS     500
int i=0;
float hp=0,sp=0;


// PulseOximeter is the higher level interface to the sensor
// it offers:
//  * beat detection reporting
//  * heart rate calculation
//  * SpO2 (oxidation level) calculation
PulseOximeter pox;

uint32_t tsLastReport = 0;

void onBeatDetected()
{
   Serial.println("Beat!");
}

void setup()
{
   bt.begin(9600);
   Serial.begin(9600);

   Serial.print("Initializing pulse oximeter..");

   if (!pox.begin()) {
       Serial.println("FAILED");
       for(;;);
   } else {
       Serial.println("SUCCESS");
   }

    pox.setOnBeatDetectedCallback(onBeatDetected);
}

void loop()
{
    i=10;
    hp=0;
    sp=0;
    while(i>0){
     pox.update();
```

```
  if (millis() - tsLastReport > REPORTING_PERIOD_MS ) {

      hp+=pox.getHeartRate();

      sp+=pox.getSpO2();



      tsLastReport = millis();
      i--;

  }

   }hp/=10;
   sp/=10;
    Serial.print("Heart rate:");
      Serial.print(hp);

   Serial.print("bpm / SpO2:");

      Serial.print(sp);
         Serial.println("%");

      bt.print(hp);
      bt.print("bpm / SpO2:");
      bt.print(sp);
      bt.println("%");

}
```

About the App -

The project utilized the Serial Bluetooth Terminal app for establishing a connection between an Arduino and a mobile device. The HC-05 Bluetooth module facilitated this communication. SPO2 and heart rate readings were transmitted from the Arduino to the mobile device via the HC-05 module. The app displayed these readings on the mobile device interfac

Workflow-

- First of all, we tested the working of Arduino Nano by running a blink code given as an example in Arduino IDE.

```
void setup() {
    // initialize digital pin LED_BUILTIN as an
  output. pinMode(LED_BUILTIN, OUTPUT);
}


// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage
level)
    delay(1000);
  digitalWrite(LED_BUILTIN, LOW);   // wait for a second
LOW                                 // turn the LED off by making the voltage
    delay(1000);
}                                   // wait for a second
```

- After that, we did connect the potentiometer to the Arduino and displayed the value of resistance.
- To verify the functionality of the MAX3001 sensor, perform a signal integrity test by monitoring its analog output using an oscilloscope. Check for the presence of ECG or PPG waveforms in the signal. This method directly confirms whether the sensor is successfully detecting physiological signals.

- The next step was to check the HC05 Bluetooth module. The LED would blink ON and OFF depending upon the data available from the Bluetooth. One thing that we had to take care of was while compiling and uploading, we had to remove the serial pin connections and connect them only after the code had compiled and uploaded correctly.
- With all individual modules confirmed operational, we proceeded to assemble the entire circuit on a breadboard. The code was then uploaded to the Arduino Nano. Next, we established a connection between the Bluetooth module and a mobile device. Utilizing the first 10 values of heart rate and SpO2 readings to compute an average enhanced the accuracy of our measurement.

Final Result -

The final project can be seen in the following video - [VID20240408162329.mp4]