



## مانیتورینگ سلامت بیماران و وضعیت بدنی ورزشکاران

گردآورندگان: زهرا انوریان - فاطمه توکلی - نازنین صرافزاده

شماره دانشجویی: ۹۵۵۲۱۰۵۴ - ۹۵۵۲۱۱۱۷ - ۹۵۵۲۱۲۵۲

استاد راهنما: دکتر امیرمهدی حسینی منزه

تابستان ۹۹

## تعریف مسئله

باتوجه به گسترش نیاز انسان به کنترل دغدغه‌های روزمره و افزایش روزافزون عوامل استرس و بیماری‌زا، نیاز به یک پلتفرم جهت سنجش سلامت انسان امروزی، به شدت حس می‌شود. در این سلامت سنج با اندازه‌گیری اطلاعات جسمانی فرد و عوامل محیطی موثر مثل ضربان قلب، دمای بدن و.. و ارسال این اطلاعات به دکتر، نه تنها رکورد کاملی از علایم جسمانی فرد به منظور تشخیص بیماری فراهم می‌کنیم، بلکه به سیستم این اجازه را می‌دهیم تا در صورت لزوم هشدار و پیشنهادات سلامتی را به فرد بدهد.

به محض در معرض استفاده قرار گرفتن دستگاه، اطلاعات جسمی و محیطی، از طریق سنسورهای تعبیه شده در سخت افزار این دستگاه که با بدن فرد در تماس است دریافت شده، سپس ریزپردازنده اطلاعات را از سنسورها خوانده و از طریق بلوتوث اطلاعات را به سیستم هوشمند فرد که تلفن همراه او می‌باشد، ارسال می‌کند. بنابر پروتکل‌های تعریف شده عملیات مورد نیاز کاربر اعم از هرگونه پردازش و نتیجه‌گیری با داده را انجام می‌دهد. کاری که ما انجام می‌دهیم صرفاً نمایش در یک اپلیکیشن گوشی همراه اندروید است.

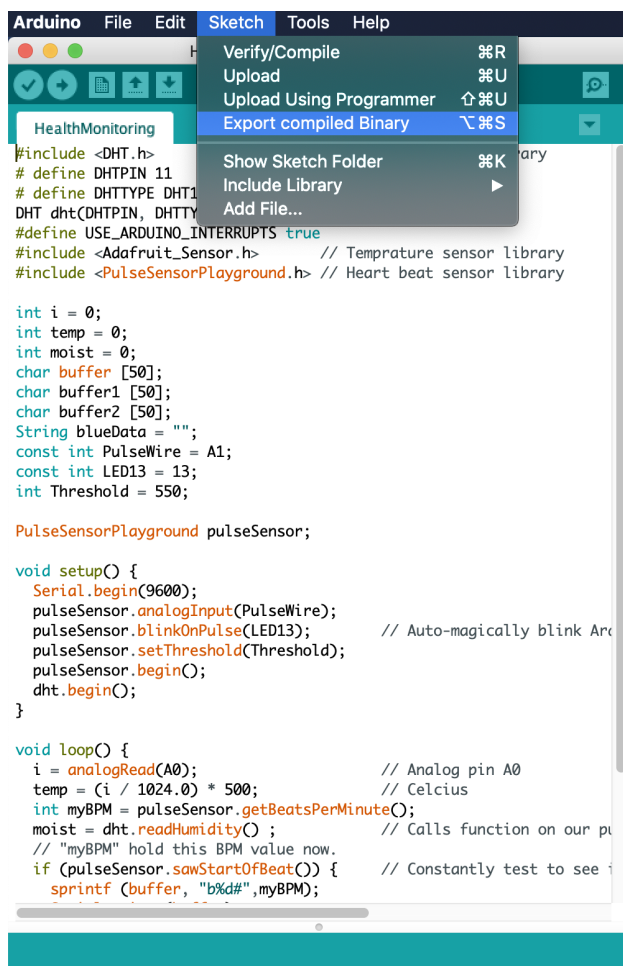
## وسایل مورد استفاده

- Arduino UNO R3
- Bluetooth HC-05
- DHT11
- Heart Beat sensor
- LM35
- POT-HG
- Android Studio

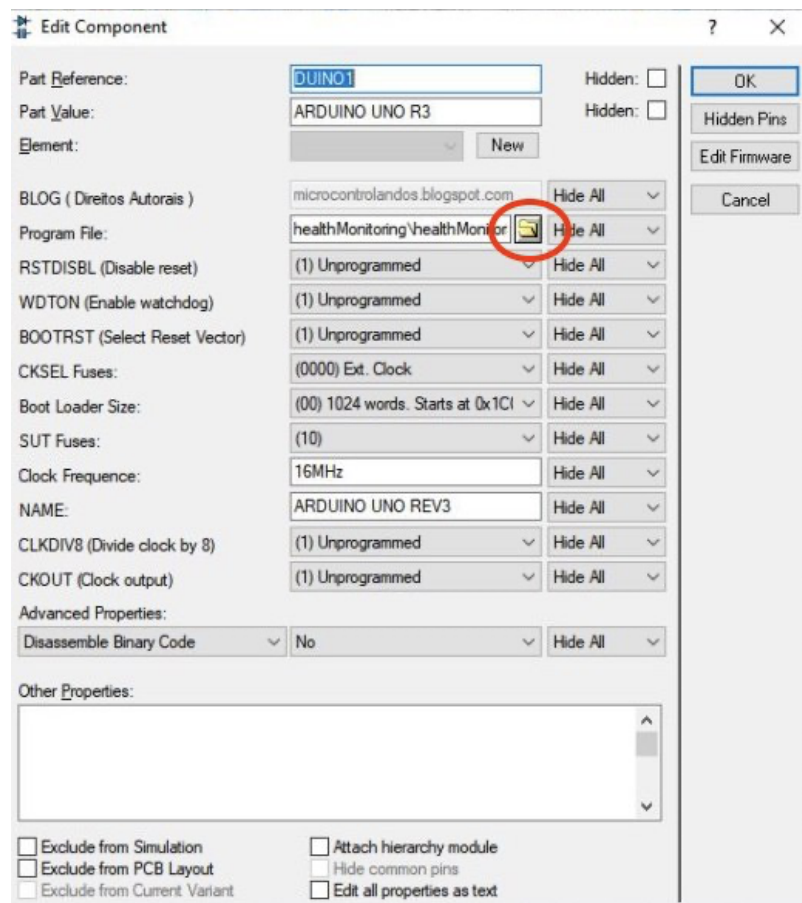
## اجرای برنامه

این پروژه دارای دو فایل نرم‌افزاری و سخت‌افزاری به ترتیب با نام‌های HealthMonitoring.ino و healthMonitoring.pdsprj است که برای تبدیل فایل

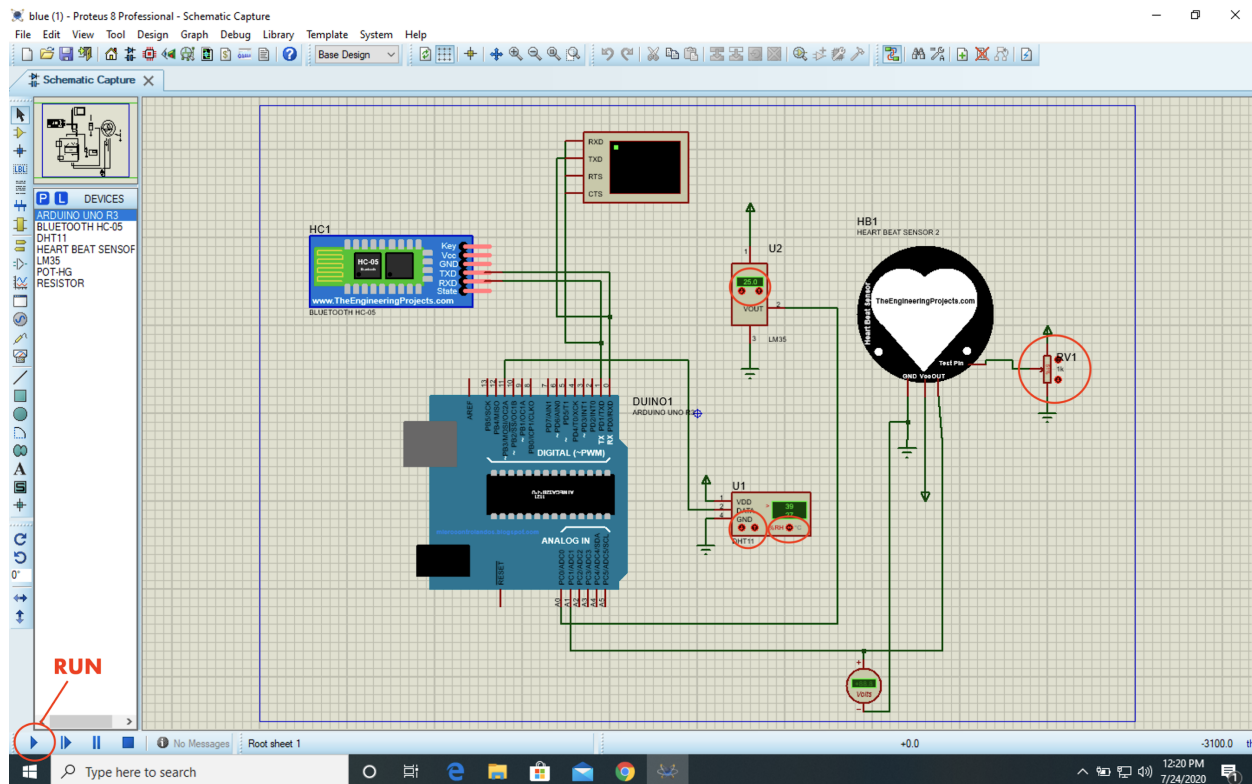
نرم‌افزاری به hex. آن را در Arduino IDE باز کنید و در گزینه‌ی Export Compiled Binary، Sketch را کلیک کنید. حال در فایل پروژه فایل hex را مشاهده می‌کنید.



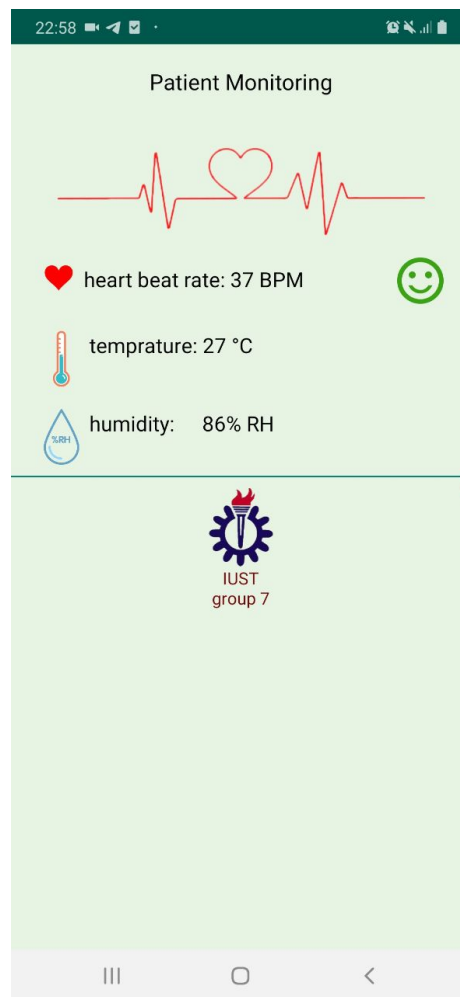
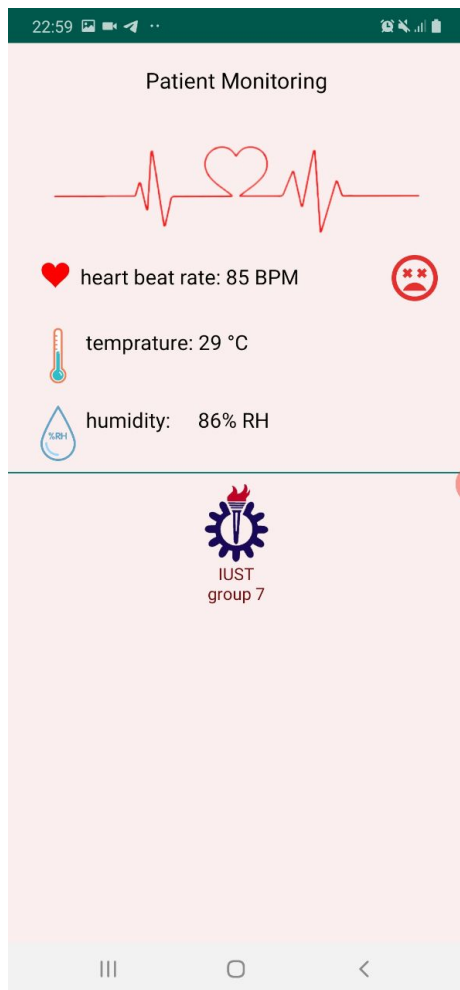
سپس فایل سخت‌افزاری را در پروتئوس باز کنید حال باید فایل hex. تولید شده را بر روی برد Arduino Uno بارگذاری کنید. برای این کار ابتدا بر روی برد دوبار کلیک کنید و سپس پنجره‌ی Edit Component برایتان باز می‌شود، آیکون پوشه‌ی مقابل Program File را فشار دهید تا بتوانید فایل hex. مورد نظر را قرار دهید.



حال می‌توانید OK را کلیک کنید و پروژه را اجرا کنید. برای تغییر ضربان قلب از یک مقاومت متغییر (POT-HG) استفاده می‌کنیم که با زیاد و کم کردن آن به صورت دستی مقدار پالس ضربان قلب تغییر می‌کند و همچنین برای تغییر دما همانطور که روی سنسور LM35 مشاهده می‌کنید، می‌توانید با فشار دادن + و - دما را تغییر دهید و همچنین برای تغییر رطوبت، باید سنسور DHT11 را روی حالت %RH قرار دهید و همانند سنسور LM35، با فشار دادن + و - می‌توان به راحتی مقدار رطوبت را تغییر داد.



حال برای نمایش داده‌ها روی اپلیکیشن موبایل، کد جاوای پوشه‌ی اندروید (MainActivity.java) را باز و اجرا کنید سپس اپلیکیشن Health Monitoring روی گوشی شما اجرا می‌شود و داده‌های حس شده توسط سنسورها را روی موبایل نمایش می‌دهد.

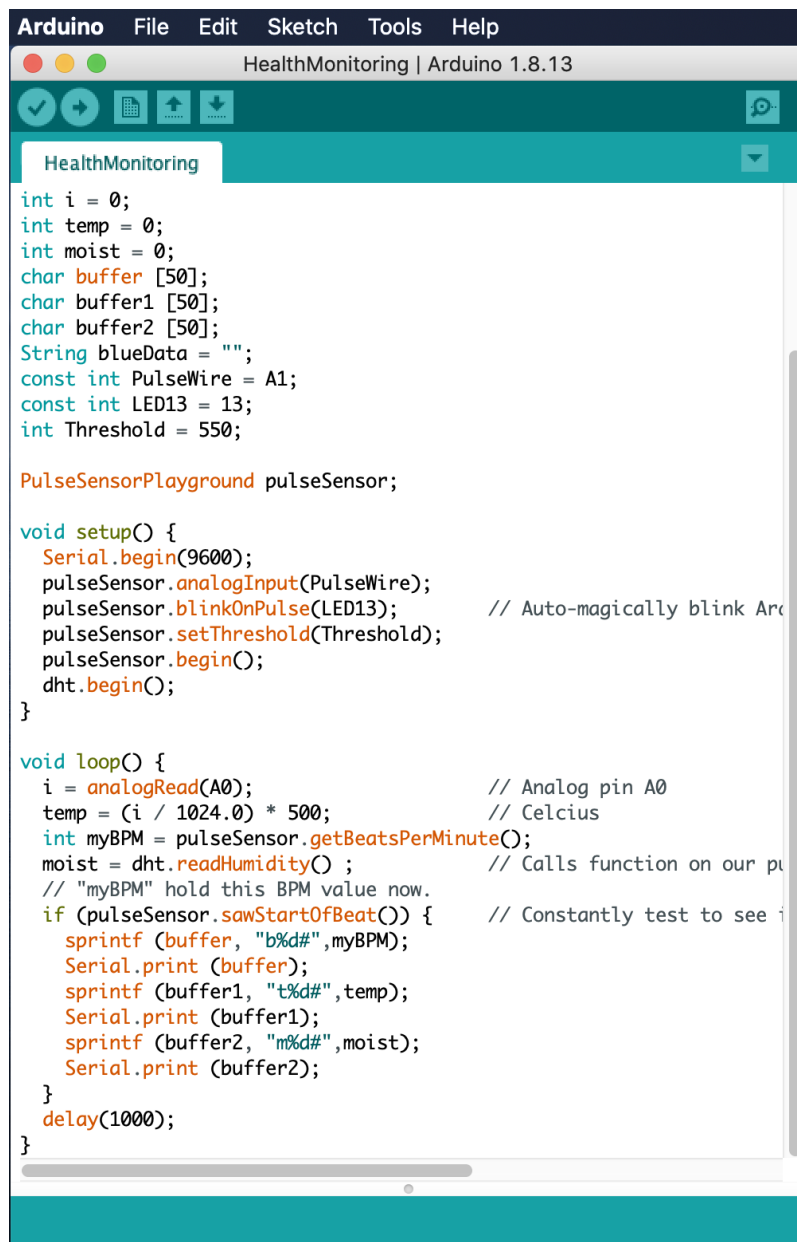


## نحوه‌ی پیاده‌سازی کد

توضیح کد Arduino:

همانطور که در کد آردوینو مشاهده می‌کنید ابتدا کتابخانه‌های سنسورها را اینکلود می‌کنیم. از کتابخانه‌ی DHT برای کار با سنسور رطوبت استفاده شده است. ۳ بافر به منظور فرستادن اطلاعات از آردوینو به backend اپلیکیشن از طریق بلوتوث تعریف شده است. در `setup` ابتدا ( `serial` ، `pulse Sensor` (Heart Rate sensor و `dht` یا همان سنسور رطوبت تعریف شده‌اند.

در **loop** برنامه اطلاعات سنسورها خوانده می‌شود و زمانی که هر سه مقدار ها برای ارسال آماده بود، داده‌های هر سنسور به همراه یک حرف در ابتدای داده به عنوان **label** در بافر ریخته می‌شود تا داده‌های **int** و **char** ماهیت خود را در کنار هم حفظ کنند. سپس هر کدام با **serial.print** ارسال می‌شود (در اینجا از طریق **bluetooth**) مقادیر ارسالی را میتوان هنگام اجرا بر روی **terminal** در **proteus** مشاهده کرد. در انتهای **loop** یک **delay** وجود دارد تا هر ۱ ثانیه اطلاعات سنسور خوانده شود.



```
Arduino File Edit Sketch Tools Help
HealthMonitoring | Arduino 1.8.13

HealthMonitoring

int i = 0;
int temp = 0;
int moist = 0;
char buffer [50];
char buffer1 [50];
char buffer2 [50];
String blueData = "";
const int PulseWire = A1;
const int LED13 = 13;
int Threshold = 550;

PulseSensorPlayground pulseSensor;

void setup() {
  Serial.begin(9600);
  pulseSensor.analogInput(PulseWire);
  pulseSensor.blinkOnPulse(LED13); // Auto-magically blink Arduino
  pulseSensor.setThreshold(Threshold);
  pulseSensor.begin();
  dht.begin();
}

void loop() {
  i = analogRead(A0); // Analog pin A0
  temp = (i / 1024.0) * 500; // Celcius
  int myBPM = pulseSensor.getBeatsPerMinute();
  moist = dht.readHumidity(); // Calls function on our pu
  // "myBPM" hold this BPM value now.
  if (pulseSensor.sawStartOfBeat()) { // Constantly test to see if
    sprintf (buffer, "b%d#", myBPM);
    Serial.print (buffer);
    sprintf (buffer1, "t%d#", temp);
    Serial.print (buffer1);
    sprintf (buffer2, "m%d#", moist);
    Serial.print (buffer2);
  }
  delay(1000);
}
```

توضیح کد Java اندروید:

به طور کلی در این قسمت از برنامه اطلاعات ما از module بلوتوث به صورت stream گرفته و پردازش می شود تا داده ها را به صورتی که می خواهیم داشته باشیم. در انتها پارامترهایی که قرار است در UI اپلیکیشن برای کاربر نشان داده شود، مشخص شده و قرار داده می شود.

در مرحله ی اول داده های stream شده باید دریافت شود. برای این کار ابتدا یک buffer reader تعریف کردیم تا input stream را دریافت کند و در بافر بریزد. محتوای بافر را تا انتها (زمانی که به ۱- برسیم) می خوانیم و اگر به # رسیدیم یعنی مقدار یکی از سنسورها را دریافت کرده ایم. این مقدار را در msg میریزیم. در انتها msg را به صورت یک MESSAGE به h که همان handler برای تحلیل پیام است می فرستیم.

```
public void run() {
    // Keep listening to the InputStream until an exception occurs
    while (true) {
        try {
            String msg = "";
            BufferedReader buffer=new BufferedReader(new InputStreamReader(mmInStream));
            int r;
            while ((r = buffer.read()) != -1) {
                char c = (char) r;

                if (c == '#')
                    break;

                msg += c;
            }
            h.obtainMessage(RECEIVE_MESSAGE, -1, -1, msg).sendToTarget();
        } catch (IOException e) {
            break;
        }
    }
}
```

در قسمت بعد پیام از سنسور در msg به handler فرستاده شده و Msg را به string تبدیل می کنیم و در متغیر textFromBT ذخیره می کنیم. در ادامه labelهایی که در کد arduino گذاشته بودیم را چک می کنیم. اگر ابتدای عددمان t باشد که از temperature می آید یعنی این



داده از سنسور دما است و در textArduino عددش را ذخیره می‌کنیم. این String در کد Android در مقابل دما داخل اپلیکیشن در گوشی نشان داده می‌شود. به همین صورت labelهای b برای heart rate و m برای moisture چک می‌شود و در fieldهای مربوطه در اپلیکیشن نوشته و به روزرسانی می‌شوند.

در این مرحله یک چک ساده انجام داده این تا اگر شرایط فرد مناسب نبود در اپلیکیشن هشدار بدهد و توجه فرد را جلب کند.

برای این منظور heart rate فرد با مقدار ۴۴ مقایسه می‌شود و اگر بیشتر بود یعنی فرد تپش قلب بیشتر از حالت نرمال دارد و صفحه‌ی برنامه قرمز می‌شود. اگر مقدار heart rate عادی باشد صفحه سبز می‌ماند.

بقیه ی کد جاوا مربوط به اتصالات بلوتوث است.

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    page = findViewById(R.id.page);
    situBtn = findViewById(R.id.situation);
    txtArduino = findViewById(R.id.txtArduino);
    txtArduino2 = findViewById(R.id.txtArduino2); // for display the received data from the Arduino
    txtArduino3 = findViewById(R.id.txtArduino3);
    h = new Handler() {
        public void handleMessage(android.os.Message msg) {
            switch (msg.what) {
                case RECIEVE_MESSAGE: // if receive massa
                    String textFromBT = (String) msg.obj;

                    if (textFromBT.charAt(0) == 't') {
                        txtArduino.setText(textFromBT.substring(1));
                    } else if (textFromBT.charAt(0) == 'b') {
                        int alert = Integer.parseInt(textFromBT.substring(1));
                        txtArduino2.setText(textFromBT.substring(1));
                        if(alert >= 44){
                            situBtn.setBackgroundResource(R.drawable.ic_sentiment_very_dissatisfied_black_24dp);
                            page.setBackgroundColor(Color.parseColor("#faeeee"));
                        } else{
                            situBtn.setBackgroundResource(R.drawable.ic_sentiment_satisfied_black_24dp);
                            page.setBackgroundColor(Color.parseColor("#e7f4e3"));
                        }
                    } else if (textFromBT.charAt(0) == 'm') {
                        txtArduino3.setText(textFromBT.substring(1));
                    }
                break;
            }
        }
    };

    btAdapter = BluetoothAdapter.getDefaultAdapter(); // get Bluetooth adapter
    checkBTState();
}
```

## چگونگی تنظیم و کار با بلوتوث در پروتئوس

برای کار با بلوتوث در پروتئوس نیاز به library بلوتوث داریم. آن را دانلود کرده و در فولدر libraryها در پروتئوس قرار می‌دهیم. برای قرار دادن این سخت افزار در پروتئوس در قسمت deviceها bluetooth را جست‌وجو کرده و BLUETOOTH-HC05 را انتخاب می‌کنیم و به پین‌های ۰ و ۱ آردوینو وصل می‌کنیم. این پین‌های سریال همزمان به virtual terminal وصل‌اند تا داده‌هایی که به بلوتوث فرستاده می‌شود را در ترمینال ببینیم.

سپس به تنظیمات بلوتوث لپ‌تاپ می‌رویم و آن را روشن می‌کنیم. سپس در bluetooth setting در tab مربوط به COM ports COM4 رفته و COM4 را می‌سازیم.

در پروتئوس روی ماژول بلوتوث کلیک کرده و physical port را COM4 انتخاب می‌کنیم. از این به بعد با پورت‌های سریال هم می‌توانیم داده از بلوتوث در کد آردوینو بگیریم و هم ارسال کنیم.

از طرفی در کد Java در قسمت MAC address آدرس لپ‌تاپ را می‌دهیم و بلوتوث گوشی را به لپ‌تاپ وصل می‌کنیم تا لپ‌تاپ حکم سرور را داشته باشد که داده‌ها را از بلوتوث پروتئوس گرفته، بررسی می‌کند و از طریق بلوتوث به گوشی ارسال می‌کند.

## کتابخانه‌های مورد استفاده

- HeartBeatSensor
  - [link](#)
- DHT11
  - [link](#)
- Adafruit Unified Sensor
  - [link](#)