

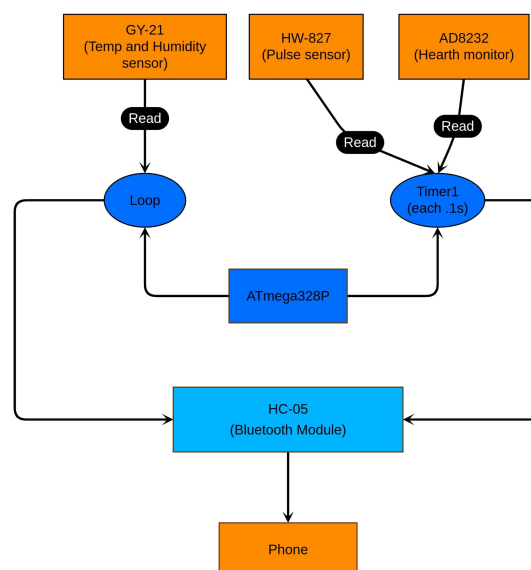
# Poligraph Test

## Introduction

A polygraph test is an instrument that measures and records physiological responses such as heart rate and electrical activity in the skin to assess whether or not someone is lying. With an Arduino Uno R3 ATmega328P, a heart rate sensor, an EKG/ECG sensor module, a precision temperature and humidity sensor SHT21, and a Bluetooth module (Master/Slave HC-05), you can build a custom polygraph system that can wirelessly transmit and record physiological data and analyze the results for deception detection or other purposes.

The concept arose from a basic curiosity about the mystery behind a lie detector, which led to a project in which, in addition to the device's already recognized utility, we have the possibility to study natural reactions of the human body in many scenarios that are difficult to manage.

## General description

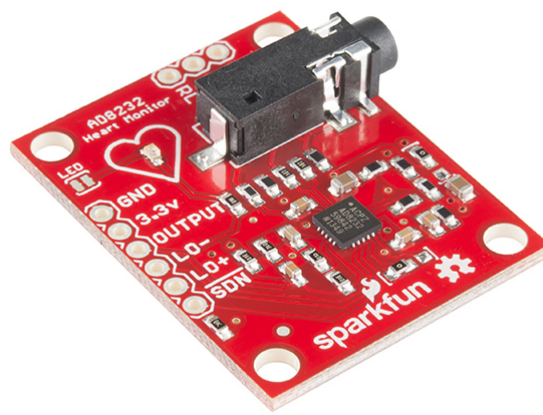


The device employs three sensors: GY-21, AD8232, and HW-827, which are used to analyze uncontrolled reactions of a person in specific situations. These are processed by an ATmega328P in the loop function and a timer set every 100 ms, both collecting and processing the information received by the sensors and then sending it via Bluetooth to the phone.

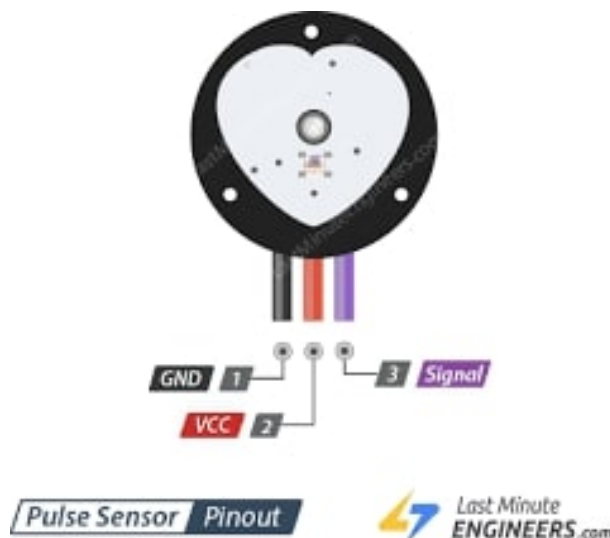
# Hardware Design

## List of parts:

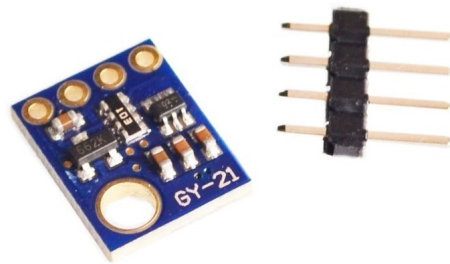
- **AD8232 (Hearth monitor)** - is a cost-effective board used to measure the electrical activity of the heart. This electrical activity can be charted as an ECG or Electrocardiogram and output as an analog reading. ECGs can be extremely noisy, the AD8232 Single Lead Heart Rate Monitor acts as an op amp to help obtain a clear signal from the PR and QT Intervals easily.



- **HW-827 (Pulse sensor)** - is a well-designed low-power plug-and-play heart-rate sensor for the Arduino. Anyone who wants to incorporate real-time heart-rate data into their work—students, artists, athletes, makers, and game and mobile developers—can benefit from it.



- **GY-21 (Temperature and humidity sensor)** - is a module that combines both a humidity sensor and a temperature sensor. Specifically, it uses the Sensirion SHT21 sensor to measure temperature and relative humidity. The GY-21 sensor module communicates with a microcontroller using the I2C protocol.



- **HC-05 (Bluetooth module)** - is a widely used Bluetooth module that allows wireless communication between devices. It uses the Bluetooth 2.0 protocol and is based on the CSR BC417 chip. The HC-05 can be configured as either a master or a slave device, making it versatile for use in a wide range of projects.



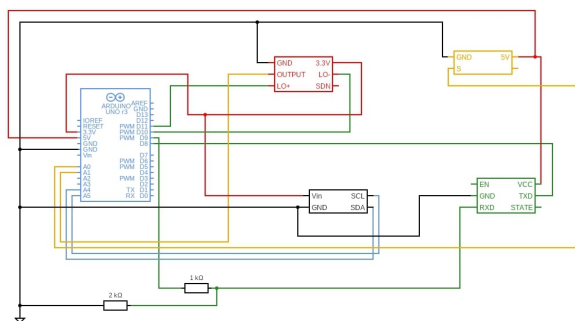
- **Arduino UNO** - is a microcontroller board based on the ATmega328P microcontroller chip. It is one of the most widely used Arduino boards and is designed to be an easy-to-use platform for creating a wide range of electronic projects.



## Additional components

- USB cable
- 1k & 2k resistors
- Jump wires

## Electrical Schematic



## Software Design

I utilized Visual Studio Code (Platform.io) for software development and only SoftwareSerial.h and Wire.h as third-party libraries. Several constants are defined in the code, including the GY21 sensor address, temperature and humidity reading commands, and the threshold value for detecting heartbeats. It also specifies variables for recording the number of heartbeats, the period between heartbeats, and the latest heartbeat value. The threshold value is constantly adjusting itself in response to background noise.

The code has an ISR (Interrupt Service Routine) that is triggered by a timer (every 100 milliseconds). Based on the signal threshold, the ISR reads the analog signal from the heart monitor and updates the heartbeat count and time between heartbeats. It also sends the analog signal and the last bpm value to the Bluetooth interface. If the Heart monitor does not detect anything and goes high on LO+ and LO-, the interrupt will be terminated.

```
ISR(TIMER1_COMPA_vect) {
    if((PINB & (1 << PB3)) || (PINB & (1 << PB2))) return;

    int signal = readADC(0);

    Threshold = (Threshold + signal) / 2;
    if(signal > Threshold && count == 0) {
        count = 1;
        if(beat == 0)
            time = micros() / 1000;
    } else if (signal < Threshold && count == 1) {
        beat++;
        count = 0;
    }
}
```

```

    if(beat == 6) {
        time = micros() / 1000 - time;
        lastBeat = !lastBeat ? 60000 * 2 / time : (lastBeat + 60000 * 2 / time)
/ 2;
        lastBeat = lastBeat <= 0 ? 0 : lastBeat;
        time = 0;
        count = 0;
        beat = 0;
    }
}
signal = readADC(1);
char result[60];
sprintf(result,"BMP %lu ECG %d",lastBeat,signal);
BTserial.println(result);
}

```

The loop() function uses I2C communication to read the temperature and humidity data from the GY21 sensor and displays them on the serial monitor and the Bluetooth terminal using the "USART0\_transmit\_string()" and "BTserial.println()" functions. It also incorporates a 5-second delay to prevent data overwhelming the Bluetooth interface.

```

void loop() {
    Wire.beginTransmission(GY21_ADDR);
    Wire.write(CMD_TEMP_HOLD);
    Wire.endTransmission();
    delay(100);

    Wire.requestFrom(GY21_ADDR, 3);

    if (Wire.available() == 3) {
        uint16_t rawTemp = (Wire.read() << 8) | Wire.read();
        rawTemp &= ~0x0003;
        tempC = -46.85 + 175.72 * (rawTemp / 65536.0);
    }
    Wire.beginTransmission(GY21_ADDR);
    Wire.write(CMD_HUMIDITY_HOLD);
    Wire.endTransmission();
    delay(100);

    Wire.requestFrom(GY21_ADDR, 3);

    if (Wire.available() == 3) {
        uint16_t rawHumidity = (Wire.read() << 8) | Wire.read();
        rawHumidity &= ~0x0003;
        humidity = -6.0 + 125.0 * (rawHumidity / 65536.0);
    }

    USART0_transmit_string(String("Temperature: " + String(tempC) + " C
Humidity: " + String(humidity) + "%").c_str());
}

```

```
BTserial.println(String("Temperature: " + String(tempC) + " C Humidity: "
+ String(humidity) + "%"));
delay(5000);
}
```

Following that, all of the data received in real time via bluetooth on a phone can be interpreted on a simple terminal or even more than that can be viewed in a demo application written in react native that can observe the received parameters and process the data if they exceed a certain threshold in order to identify certain reactions of the monitored person that can lead to the verification of the truth said.

## Obtained results

Following the completion of the project, I was able to create, first and foremost, a device that can be used to monitor people's heart rates by analyzing the electrical activity of the heart as well as the temperature and humidity at the skin level, with the monitoring being able to be local directly on the serial or from a distance using the bluetooth module by communicating with a phone, which can save the received data locally. Second, an algorithm that can operate on a smartphone and assess the data received, generating probabilities about the watched person's truthfulness.

## Conclusion

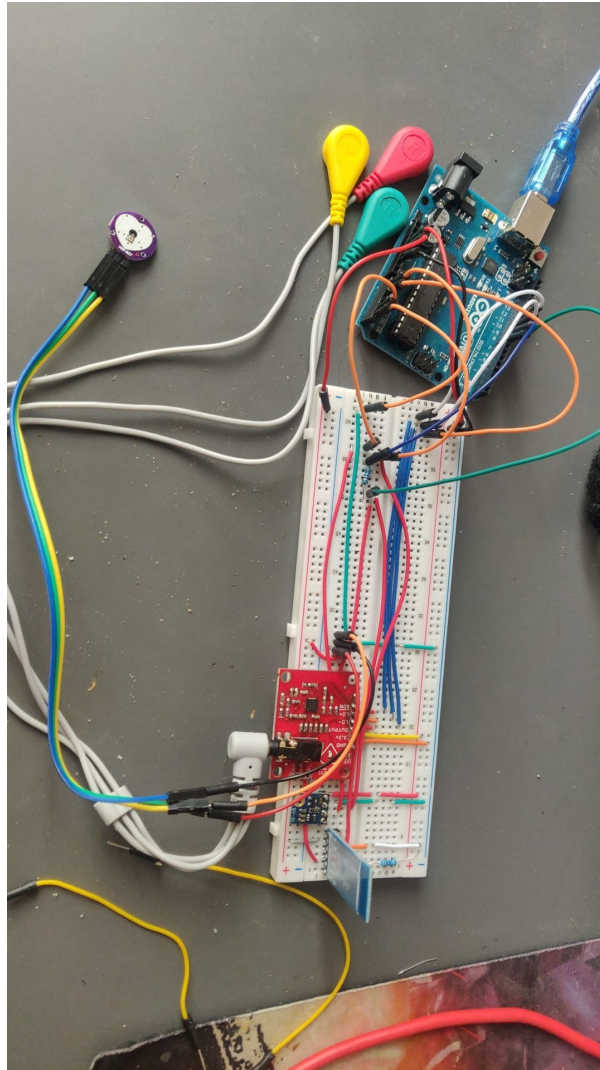
In conclusion, through the successful completion of this project, I have accomplished several significant outcomes. Firstly, I have developed a device capable of monitoring individuals' heart rates by analyzing the electrical activity of their hearts, as well as measuring the temperature and humidity at the skin level. This monitoring can be conducted either locally, directly through the serial connection, or remotely using the Bluetooth module by communicating with a smartphone. Additionally, the smartphone is equipped with an algorithm that can process the received data and generate probabilities regarding the truthfulness of the individual being monitored. These achievements pave the way for enhanced monitoring capabilities and contribute to the advancement of data-driven assessments in various contexts.

## Download

- [poligraph.7z](#)

## Journal

- 03.05.2023 - So far, I've assembled all of the components, processing the received signals and testing serial data transmission over Bluetooth and locally.

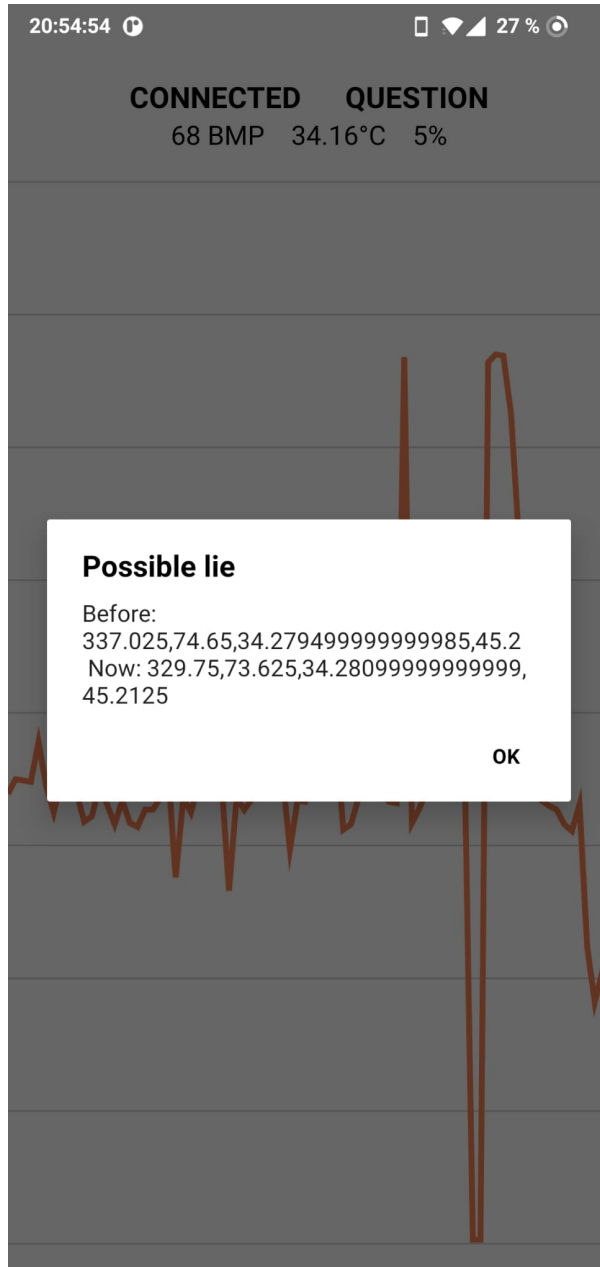


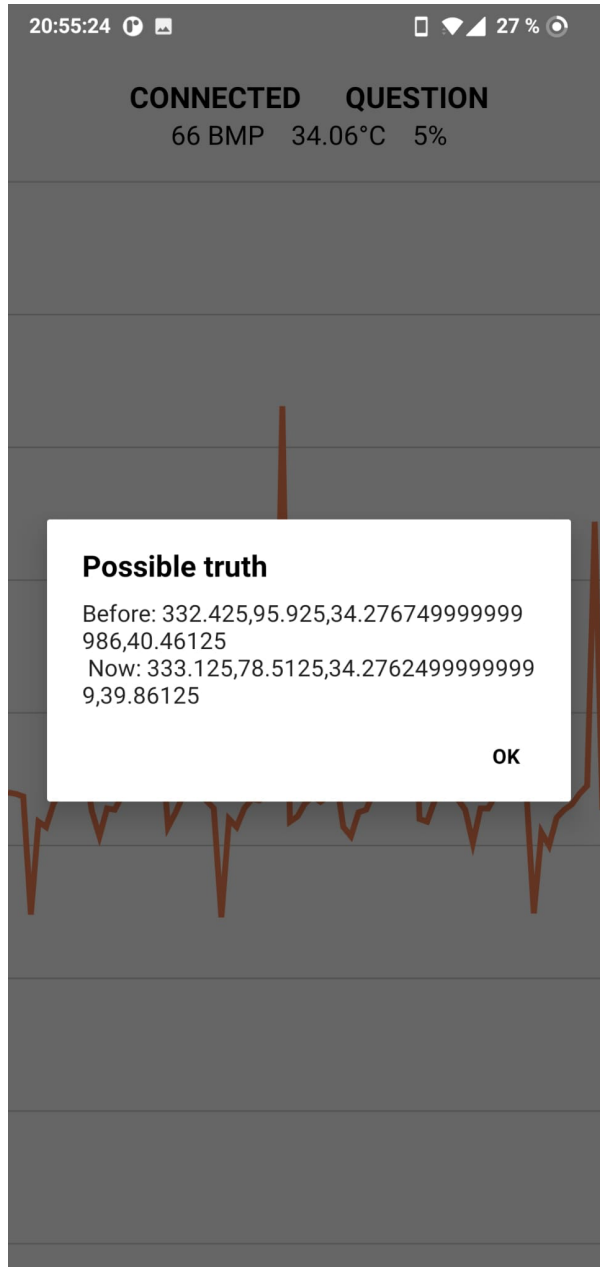
- 20.05.2023 - I have successfully completed the project and have also developed an Android application that connects an Android device to the Bluetooth module and processes the received data, thus becoming a monitoring interface.

**CONNECTED QUESTION**

76 BMP 34.12°C 56.79%







### CONNECTED QUESTION

74 BMP 34.03°C 56%



## Bibliography / References

- [PDF](#)

From:  
<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:  
<http://ocw.cs.pub.ro/courses/pm/prj2023/apredescu/poligraph>

Last update: **2023/05/28 19:40**



