

# گزارش پروژه پایانی سیستم های نهفته و بی درنگ

اعضای گروه:

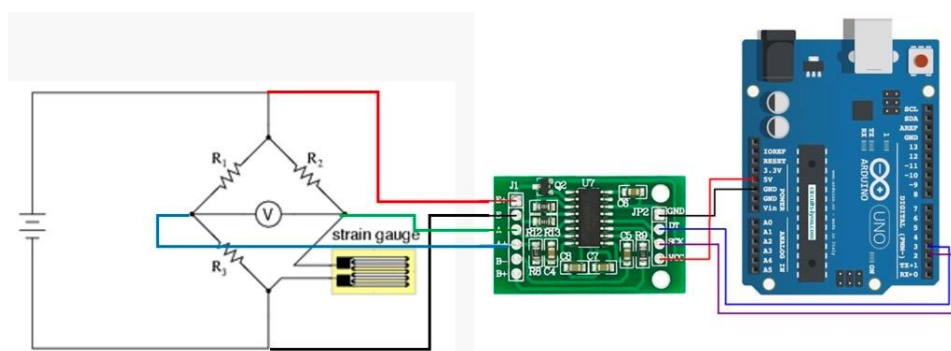
کیوان بوشهری، عابد عبادی، سینا ضیایی

بخش اول) ابتدا به توضیح سخت افزار می پردازیم:

قطعات مورد نیاز:

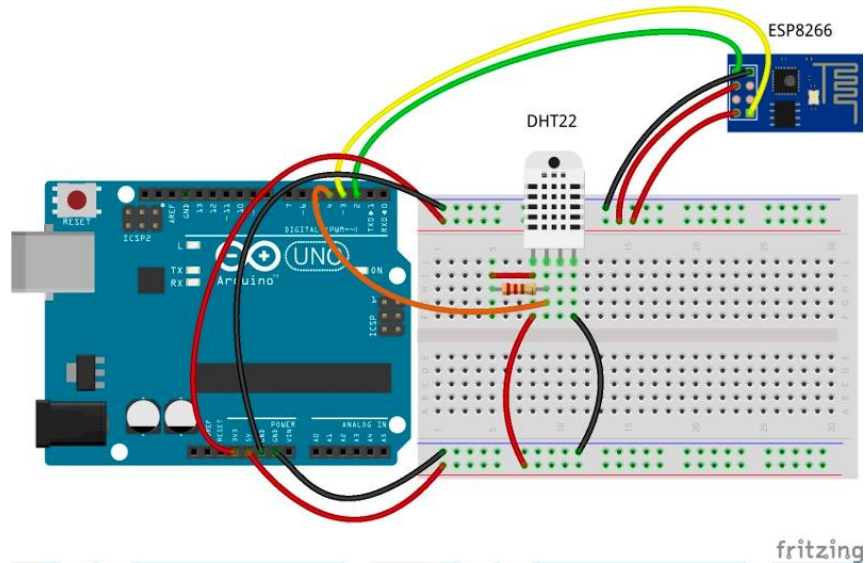
شماره قطعه	نام قطعه	تعداد	قیمت هر واحد (به تومان)
۱	Arduino UNO	1	290
۲	Amplifier	2	50
۳	Strain gauge	3	30
۴	DHT11	1	50
۵	ESP8266	1	75
۶	مقاومت ... اهمی	100	50 جمعا
۷	سیم	2 دسته	27 جمعا
۸	برد برد	1	40
۹	باتری	1	80
۱۰	پیک	۳ سری	200 جمعا

هزینه کل: 1 میلیون تومان ناقابل 😞



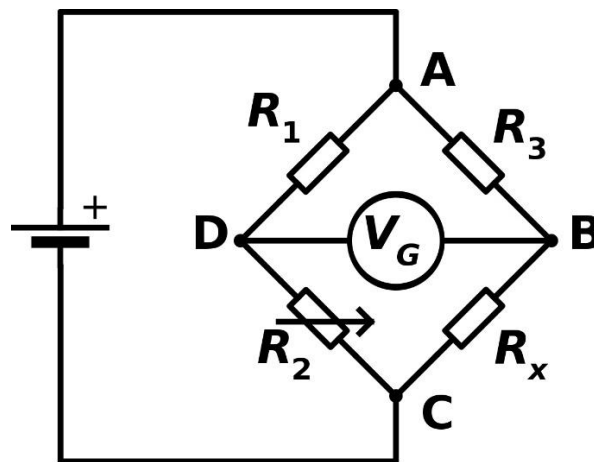
شکل 1) بخش های اتصال Strain Gauge

بخش اصلی مدار که برای محاسبه ی تعداد تنفس کاربر استفاده می شود به صورت بالا می باشد. همان طور که در شکل ۳ مشخص است پل وتسون معادله ۱ را دارد. شایان ذکر است که برای اندازه گیری تعداد تنفس باید مقدار strain gauge را به صورت سنجش تفاوت با مقدار اولیه در نظر گرفت. به این معنا که ممکن است مقدار اولیه هر سری از کار با این ماژول متفاوت باشد، پس باید تفاوت ولتاژ معیار اندازه گیری تعداد تنفس باشد. به این نکته نیز باید توجه کرد که برای مشاهده ی مقادیر strain gauge باید از آمپلیفایر نیز استفاده کرد.



شکل ۲) بخش های اتصالی سنسور دما و رطوبت و ماژول esp32

این بخش از مدار نمایانگر محل های اتصال ماژول وای فای و سنسور دما / رطوبت است.



شکل 3) مدار پل وتسون استفاده شده در بخش strain gauge مدار

$$V_G = \left( \frac{R_2}{R_1 + R_2} - \frac{R_x}{R_x + R_3} \right) V_s$$

$$R_x = \frac{R_2 \cdot V_s - (R_1 + R_2) \cdot V_G}{R_1 \cdot V_s + (R_1 + R_2) \cdot V_G} R_3$$

فرمول (۱) فرمول مدار پل وتسون (Rx مفاومت Strain gauge است)

فرمول محاسبه ی تعداد تنفس های کاربر: نکته ی مهم این هست که ما باید یک بازه را مشخص کنیم و در آن بازه سَمپل گیری کنیم تا تشخیص بدیم شخص در حال نفس کشیدن هست یا خیر. به این باید توجه کنیم که اگر تعداد سَمپل هامون زیاد باشه عملاً تفاوت خاصی رو نمی بینیم.

$$\frac{\Delta V_{\text{strain gauge}}}{\Delta T} = \frac{dbreath}{dtime}$$

$$\Delta T = 15 \text{ s} \rightarrow \text{BPS} = N \left( \frac{dv}{dt} \geq \varepsilon \right) \times 4 \text{ bps (breaths per second)}$$

$$\varepsilon = 100 \text{ mVps}$$

فرمول (۲) فرمول های محاسبه تعداد تنفس (ساخته ی ذهن ما 😊)

**بخش دوم) بررسی کد ها:**

اتصال به سرور MQTT با ماژول esp32 :

```
// Initialize the Ethernet client object
```

```
WiFiEspClient espClient;
```

```
// Setup the MQTT client class by passing in the WiFi client and MQTT server and login details.
```

```
Adafruit_MQTT_Client mqtt(&espClient, AIO_SERVER, AIO_SERVERPORT, AIO_USERNAME, AIO_KEY);
```

```
Adafruit_MQTT_Publish Light = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/Keyvan");
```

مشخص کردن پین ها :

```
// HX711 circuit wiring
```

```
const int LOADCELL_DOUT_PIN = 6;
```

```
const int LOADCELL_SCK_PIN = 7;
```

```
HX711 scale;
```

```
#define WIFI_AP "K1-Rooms"
```

```
#define WIFI_PASSWORD "09212041269"
```

```
// DHT
```

```
#define DHTPIN 4
```

```
#define DHTTYPE DHT11
```

اتصال به وای فای برای ارتباط با اینترنت:

```
// Connect to WPA/WPA2 network
```

```
status = WiFi.begin(WIFI_AP, WIFI_PASSWORD);
```

پینگ سرور برای جلوگیری از قطع ارتباط:

```
// ping the server to keep the mqtt connection alive
```

```
if (! mqtt.ping()) {
```

```
    mqtt.disconnect();
```

```
}
```

```
    delay(5000);
```

چک کردن وجود ماژول esp8266 برای اتصال به وای فای:

```
void InitWiFi()
```

```
{
```

```
    // initialize serial for ESP module
```

```
    soft.begin(9600);
```

```
    // initialize ESP module
```

```
    WiFi.init(&soft);
```

```
    // check for the presence of the shield
```

```
    if (WiFi.status() == WL_NO_SHIELD) {
```

```
        Serial.println("WiFi shield not present");
```

```
        // don't continue
```

```
        while (true);
```

```
    }
```

```
    Serial.println("Connecting to AP ...");
```

```
    // attempt to connect to WiFi network
```

```

while ( status != WL_CONNECTED) {

  Serial.print("Attempting to connect to WPA SSID: ");

  Serial.println(WIFI_AP);

  // Connect to WPA/WPA2 network

  status = WiFi.begin(WIFI_AP, WIFI_PASSWORD);

  delay(500);

}

Serial.println("Connected to AP");
}

```

بخش مربوط به اتصال MQTT:

```

void MQTT_connect() {

  int8_t ret;

  // Stop if already connected.

  if (mqtt.connected()) {

    return;

  }

  Serial.print("Connecting to MQTT... ");

  uint8_t retries = 3;

  while ((ret = mqtt.connect()) != 0) { // connect will return 0 for connected

    Serial.println(mqtt.connectErrorString(ret));

    Serial.println("Retrying MQTT connection in 5 seconds...");

    mqtt.disconnect();

    delay(5000); // wait 5 seconds

    retries--;

    if (retries == 0) {

      // basically die and wait for WDT to reset me

      while (1);

    }

  }

  Serial.println("MQTT Connected!");

}

```