# Arduino – monitorizare locuinta

In acest tutorial vei descoperi cum se poate monitoriza o locuinta utilizand un senzor de temperatura, un senzor de efractie, o placa Arduino si un shield Wi-FI. Este o solutie simpla prin care poti observa daca cineva ti-a deschis usa de la intrare sau daca temperatura este normala.

Exista o gama variata de senzori care pot detecta efractia (senzor magnetic, inductiv, fotoelectric). In cazul de fata s-a utilizat, pe post de senzor de efractie, un buton brick. Daca butonul a fost apasat se considera ca usa a fost deschisa si senzorul a fost declansat.

Toata informatia o vei citi cu ajutorul unui browser deoarece placa Arduino se va comporta ca si un server web. Asta inseamna ca iti poti monitoriza locuinta chiar si de pe un smartphone cu conexiune la internet.

Vei avea nevoie de urmatoarele componente:

- O placa Arduino - http://www.robofun.ro/arduino
- Un Arduino Wifi Shield - http://www.robofun.ro/arduino_wifi_shield
- Un alimentator extern Arduino
  http://www.robofun.ro/surse_de_alimentare/alimentatoare/alimentator-extern-arduino
- Un senzor de presiune atmosferica si temperatura BMP085 Blue Edition
  http://www.robofun.ro/senzori/vreme/senzor-presiune-atmosferica-bmp085-blue
- Un buton mare brick - http://www.robofun.ro/electronice/butoane/buton-mare-brick
- Un breadboard - http://www.robofun.ro/breadboard/breadboard_mini
- Fire de conexiune mama-tata si tata-tata - http://www.robofun.ro/cabluri

## Cum se conecteaza senzorul si butonul brick ?

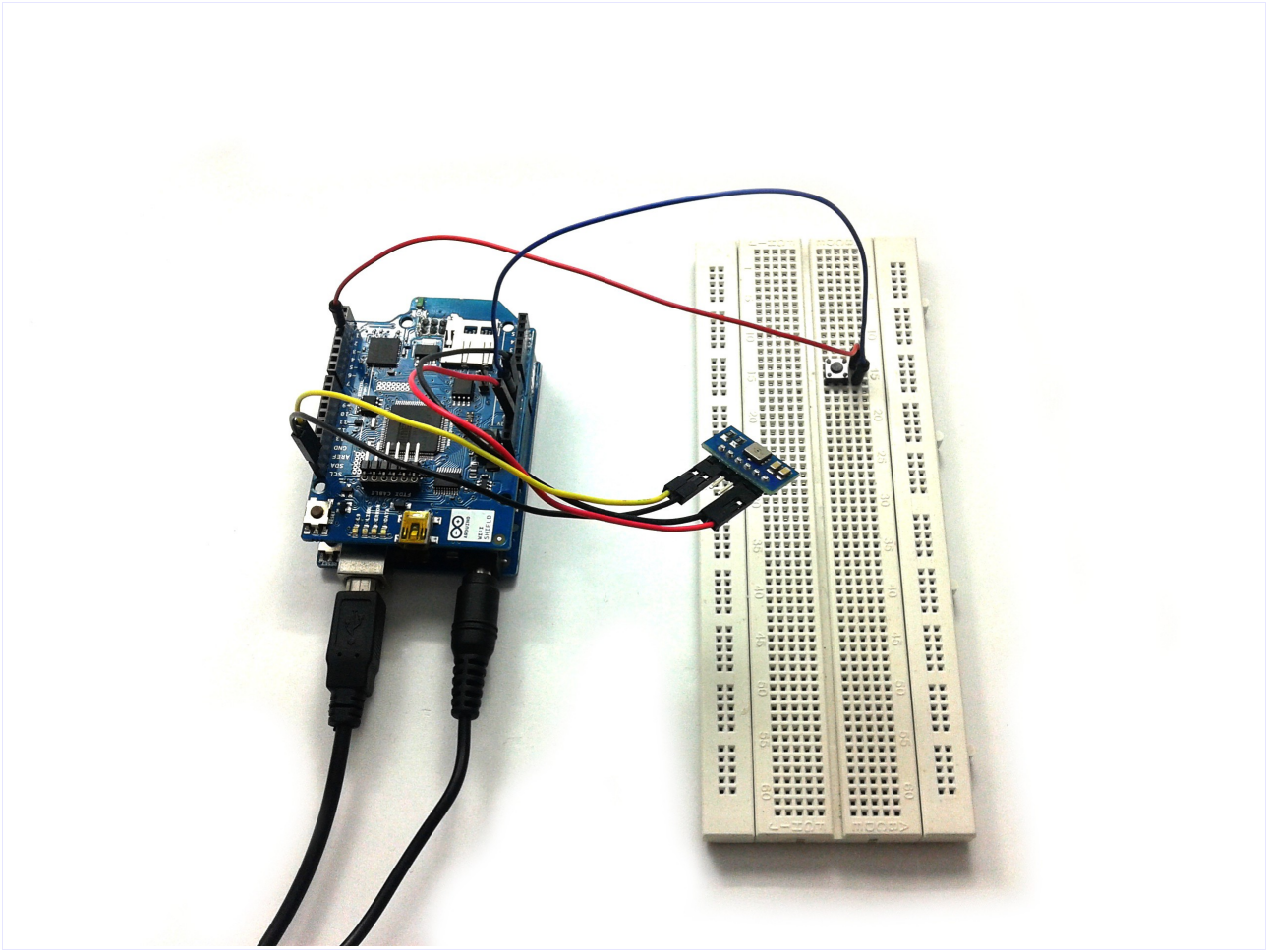
1. Infige shield-ul Wifi in pinii placii Arduino.
2. Conecteaza senzorul de presiune atmosferica si temperatura BMP085 Blue Edition dupa urmatorul tabel:

| Senzor pin VCC | Arduino pin 3.3V |
|---|---|
| Senzor pin GND | Arduino pin GND |
| Senzor pin SDA | Arduino pin SDA |
| Senzor pin SCL | Arduino pin SCL |

3. Conecteaza butonul brick astfel:

| Buton pin GND | Arduino pin GND |
|---|---|
| Buton pin VCC | Arduino pin 5V |
| Buton pin OUT | Arduino pin D2 |

4. Conecteaza alimentatorul Arduino in mufa Jack.



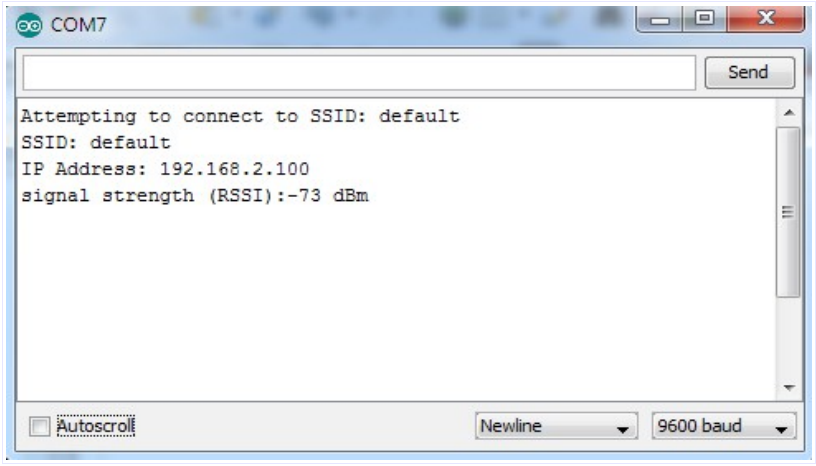
## Codul sursa.

Dupa ce ai realizat toate conexiunile fizice dintre senzori si placa Arduino, acum este momentul sa incarci sketch-ul de mai jos.

Inainte de a incarca sketch-ul in Arduino cauta cele 2 linii si modifica-le conform routerului tau:

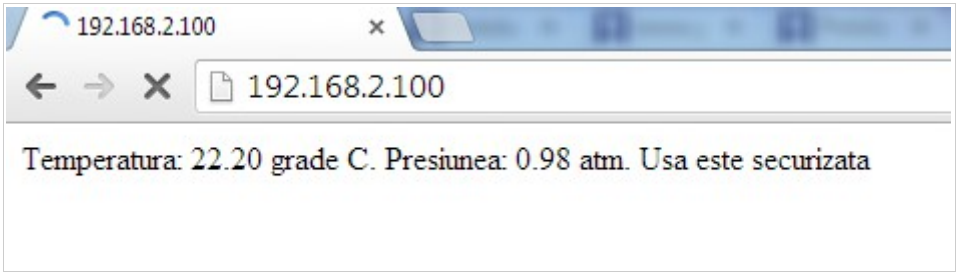
```
char ssid[] = "ssid";
char pass[] = "parola secreta";
```

Copiaza codul de mai jos cu copy/paste, incarca-l in placa Arduino si deschide Monitorul Serial. La cateva momente de la deschiderea Monitorului Serial iti va aparea urmatoarea informatie.



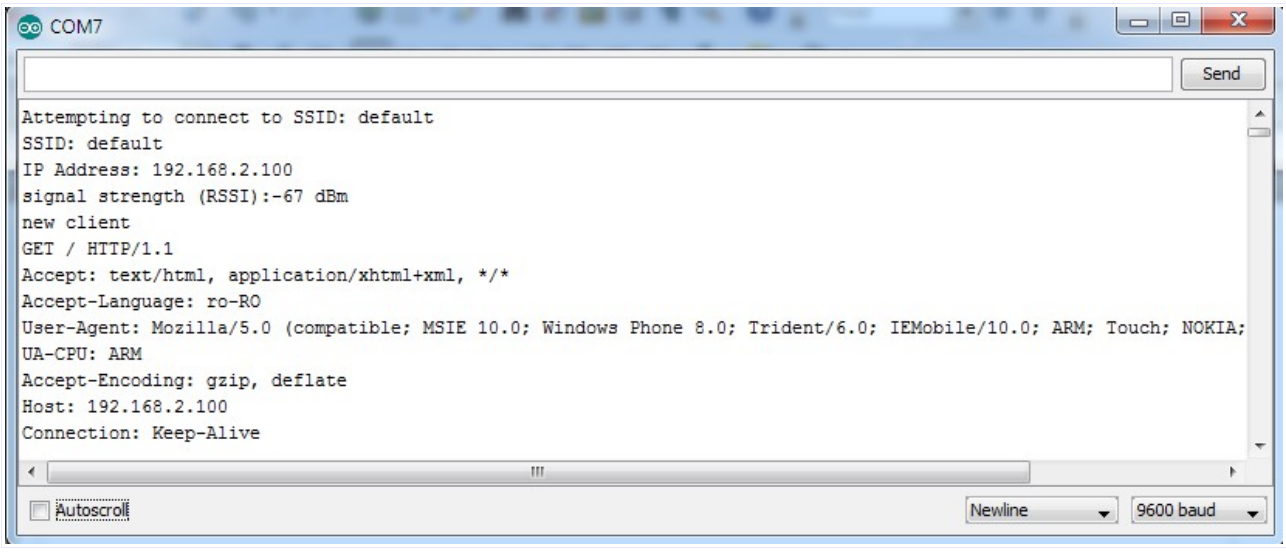

Informatia de mai sus iti arata ca placa Arduino s-a conectat cu succes la router si este pregatita sa primeasca cereri. IP-ul care apare in imagine este IP-ul pe care il tastezi in browser.

Poti sa deschizi orice browser si sa tastezi IP-ul in bara de link-uri. Asa iti va aparea informatia dupa ce te-ai conectat la placa Arduino.




Utilizand un smartphone iti va aparea asa.


In acelasi timp poti observa in Monitorul Serial cateva detalii despre dispozitivul care se conecteaza la placa Arduino.



```
#include <SPI.h>
#include <WiFi.h>
#include <Wire.h>

#define BMP085_ADDRESS 0x77  // I2C address of BMP085

const unsigned char OSS = 0;  // Oversampling Setting

// Calibration values
int ac1;
int ac2;
int ac3;
unsigned int ac4;
unsigned int ac5;
unsigned int ac6;
int b1;
int b2;
int mb;
int mc;
int md;

long b5;

char ssid[] = "ssid";
char pass[] = "parola secreta";

int status = WL_IDLE_STATUS;

boolean door_status = false;

WiFiServer server(80);


void setup() {
  pinMode(2, INPUT);
  // start serial port for debugging purposes
  Serial.begin(9600);
  Wire.begin();
  bmp085Calibration();

  // Attach interrupt to pin 2
  attachInterrupt(0, setDoorStatus, FALLING);


  // attempt to connect to Wifi network:
  while ( status != WL_CONNECTED) {
    Serial.print("Attempting to connect to SSID: ");
    Serial.println(ssid);
```

```
    status = WiFi.begin(ssid, pass);
    // wait 10 seconds for connection:
    delay(10000);
  }
  server.begin();
  // you're connected now, so print out the status:
  printWifiStatus();
}


void loop() {

  // listen for incoming clients
  WiFiClient client = server.available();
  if (client) {
    Serial.println("new client");
    // an http request ends with a blank line
    boolean currentLineIsBlank = true;
    while (client.connected()) {
      if (client.available()) {
        char c = client.read();
        Serial.write(c);
        // if you've gotten to the end of the line (received a
newline
        // character) and the line is blank, the http request has
ended,
        // so you can send a reply
        if (c == '\n' && currentLineIsBlank) {
          // send a standard http response header
          client.println("HTTP/1.1 200 OK");
          client.println("Content-Type: text/html");
          client.println("Connnection: close");
          client.println();
          client.println("<!DOCTYPE HTML>");
          client.println("<html>");
          client.println("<meta http-equiv=\"refresh\"
content=\"5\">");

          float temperature = bmp085GetTemperature(bmp085ReadUT());
          float pressure = bmp085GetPressure(bmp085ReadUP());
          float atm = pressure / 101325; // "standard atmosphere"
          float altitude = calcAltitude(pressure);
          client.print("Temperatura: ");
          client.print(temperature);
          client.print(" grade C.");
          client.print(" Presiunea: ");
          client.print(atm);
          client.print(" atm.");
```

```
          if (door_status == false){
            client.println(" Usa este securizata");
          }
          else {
            client.println(" Atentie! Usa a fost deschisa.");
          }
          client.println("<br />");
          client.println("</html>");
          break;
        }
        if (c == '\n') {
          // you're starting a new line
          currentLineIsBlank = true;
        }
        else if (c != '\r') {
          // you've gotten a character on the current line
          currentLineIsBlank = false;
        }
      }
    }
    // give the web browser time to receive the data
    delay(1);
    // close the connection:
    client.stop();
    Serial.println("client disonnected");
  }
}


void printWifiStatus() {
  // print the SSID of the network you're attached to:
  Serial.print("SSID: ");
  Serial.println(WiFi.SSID());

  // print your WiFi shield's IP address:
  IPAddress ip = WiFi.localIP();
  Serial.print("IP Address: ");
  Serial.println(ip);

  // print the received signal strength:
  long rssi = WiFi.RSSI();
  Serial.print("signal strength (RSSI):");
  Serial.print(rssi);
  Serial.println(" dBm");
}
```

```
void setDoorStatus() {
  door_status = true;
}

void bmp085Calibration()
{
  ac1 = bmp085ReadInt(0xAA);
  ac2 = bmp085ReadInt(0xAC);
  ac3 = bmp085ReadInt(0xAE);
  ac4 = bmp085ReadInt(0xB0);
  ac5 = bmp085ReadInt(0xB2);
  ac6 = bmp085ReadInt(0xB4);
  b1 = bmp085ReadInt(0xB6);
  b2 = bmp085ReadInt(0xB8);
  mb = bmp085ReadInt(0xBA);
  mc = bmp085ReadInt(0xBC);
  md = bmp085ReadInt(0xBE);
}

// Calculate temperature in deg C
float bmp085GetTemperature(unsigned int ut){
  long x1, x2;

  x1 = (((long)ut - (long)ac6)*(long)ac5) >> 15;
  x2 = ((long)mc << 11)/(x1 + md);
  b5 = x1 + x2;

  float temp = ((b5 + 8)>>4);
  temp = temp /10;

  return temp;
}



long bmp085GetPressure(unsigned long up){
  long x1, x2, x3, b3, b6, p;
  unsigned long b4, b7;

  b6 = b5 - 4000;
  // Calculate B3
  x1 = (b2 * (b6 * b6)>>12)>>11;
  x2 = (ac2 * b6)>>11;
  x3 = x1 + x2;
  b3 = (((((long)ac1)*4 + x3)<<OSS) + 2)>>2;

  // Calculate B4
  x1 = (ac3 * b6)>>13;
```

```
  x2 = (b1 * ((b6 * b6)>>12))>>16;
  x3 = ((x1 + x2) + 2)>>2;
  b4 = (ac4 * (unsigned long)(x3 + 32768))>>15;

  b7 = ((unsigned long)(up - b3) * (50000>>OSS));
  if (b7 < 0x80000000)
    p = (b7<<1)/b4;
  else
    p = (b7/b4)<<1;

  x1 = (p>>8) * (p>>8);
  x1 = (x1 * 3038)>>16;
  x2 = (-7357 * p)>>16;
  p += (x1 + x2 + 3791)>>4;

  long temp = p;
  return temp;
}


char bmp085Read(unsigned char address)
{
  unsigned char data;

  Wire.beginTransmission(BMP085_ADDRESS);
  Wire.write(address);
  Wire.endTransmission();

  Wire.requestFrom(BMP085_ADDRESS, 1);
  while(!Wire.available())
    ;

  return Wire.read();
```

```
}

// Read 2 bytes from the BMP085
// First byte will be from 'address'
// Second byte will be from 'address'+1
int bmp085ReadInt(unsigned char address)
{
  unsigned char msb, lsb;

  Wire.beginTransmission(BMP085_ADDRESS);
  Wire.write(address);
  Wire.endTransmission();

  Wire.requestFrom(BMP085_ADDRESS, 2);
  while(Wire.available()<2)
    ;
  msb = Wire.read();
  lsb = Wire.read();

  return (int) msb<<8 | lsb;
}

// Read the uncompensated temperature value
unsigned int bmp085ReadUT(){
  unsigned int ut;

  // Write 0x2E into Register 0xF4
  // This requests a temperature reading
  Wire.beginTransmission(BMP085_ADDRESS);
  Wire.write(0xF4);
  Wire.write(0x2E);
  Wire.endTransmission();

  // Wait at least 4.5ms
  delay(5);


  // Read two bytes from registers 0xF6 and 0xF7
  ut = bmp085ReadInt(0xF6);
  return ut;
}

// Read the uncompensated pressure value
unsigned long bmp085ReadUP(){

  unsigned char msb, lsb, xlsb;
  unsigned long up = 0;
```

```
  // Write 0x34+(OSS<<6) into register 0xF4
  // Request a pressure reading w/ oversampling setting
  Wire.beginTransmission(BMP085_ADDRESS);
  Wire.write(0xF4);
  Wire.write(0x34 + (OSS<<6));
  Wire.endTransmission();

  // Wait for conversion, delay time dependent on OSS
  delay(2 + (3<<OSS));

  // Read register 0xF6 (MSB), 0xF7 (LSB), and 0xF8 (XLSB)
  msb = bmp085Read(0xF6);
  lsb = bmp085Read(0xF7);
  xlsb = bmp085Read(0xF8);

  up = (((unsigned long) msb << 16) | ((unsigned long) lsb << 8) |
(unsigned long) xlsb) >> (8-OSS);

  return up;
}

void writeRegister(int deviceAddress, byte address, byte val) {
  Wire.beginTransmission(deviceAddress); // start transmission to
device
  Wire.write(address);       // send register address
  Wire.write(val);         // send value to write
  Wire.endTransmission();     // end transmission
}

int readRegister(int deviceAddress, byte address){

  int v;
  Wire.beginTransmission(deviceAddress);
  Wire.write(address); // register to read
  Wire.endTransmission();

  Wire.requestFrom(deviceAddress, 1); // read a byte

  while(!Wire.available()) {
    // waiting
  }
```

```
  v = Wire.read();
  return v;
}

float calcAltitude(float pressure){

  float A = pressure/101325;
  float B = 1/5.25588;
  float C = pow(A,B);
  C = 1 - C;
  C = C /0.0000225577;

  return C;
}
```