

ITS5900/4900 – Internet of Things

Spring 2018

Project 02

1. Introduction:

We are assigned to create a project in which the temperature sensor (BMP-280) will sense the outside temperature and pressure and report them to serial port and to the AWS IoT platform.

2. Description:

This project uses BMP-280 sensor, particle photon, breadboard and jumper wires to sense the outside temperature and pressure. I used Fritzing software to create the schematics of photon circuit.

2.1 Hardware:

Pin D0: SDA (Black wire)

Pin D1: SCL (Brown wire)

VCC: 3.3v (Orange wire)

GND: GND (Green wire)

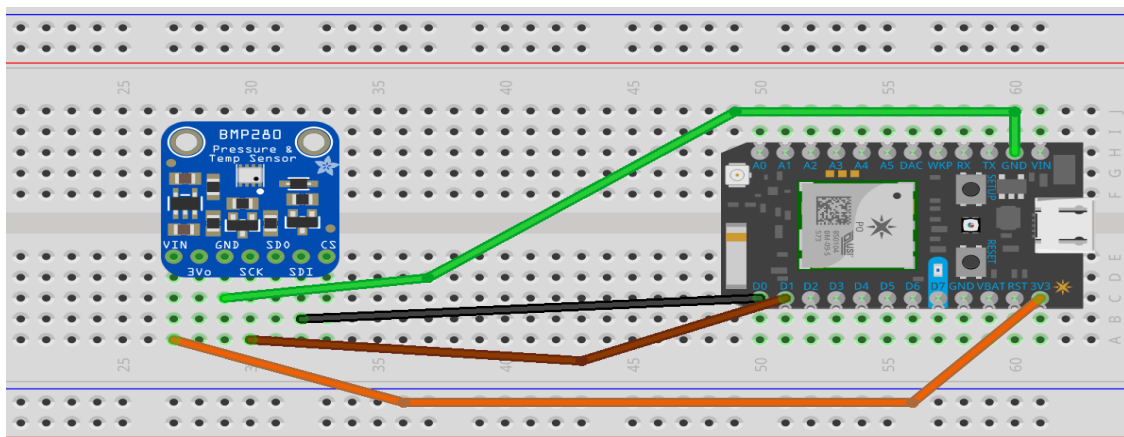


Fig 1. Schematics of Photon Circuit

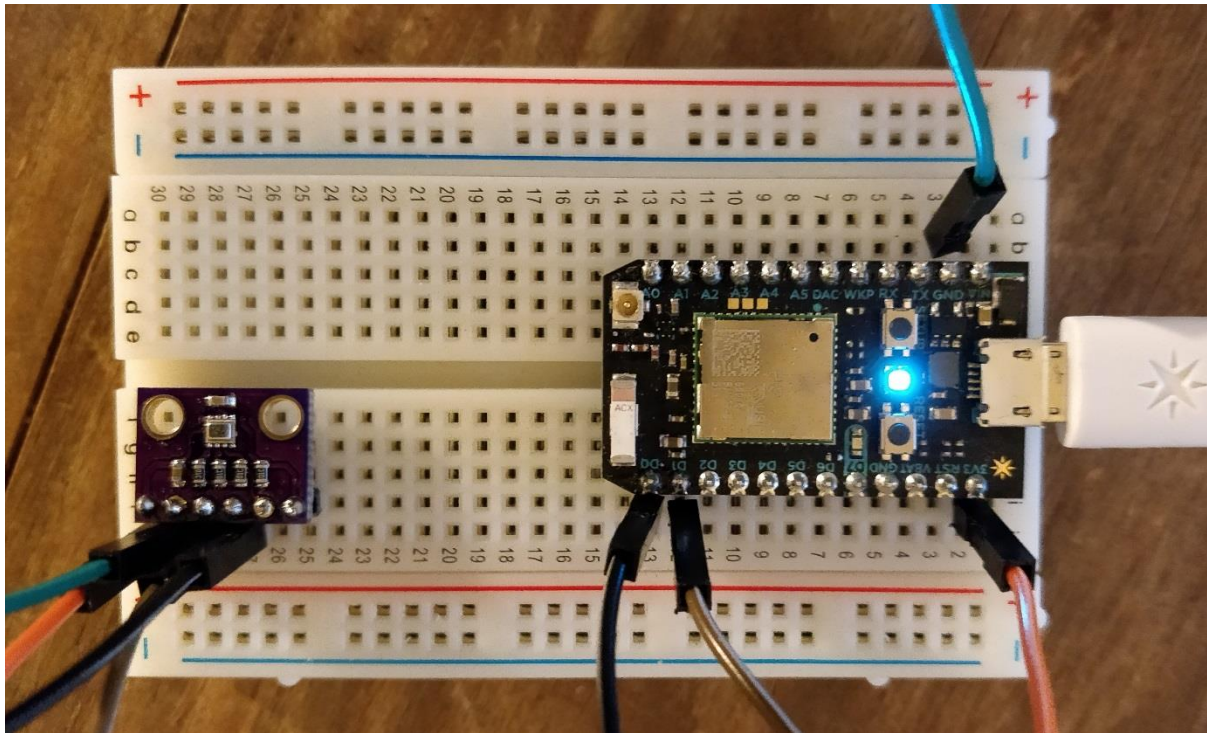


Fig 2. Photo of Particle photon circuit

2.2 Services:

The BMP280 sensor senses temperature and pressure from surroundings and publish it to the photon console, it uses 3.3v power from particle photon. The code uses **Adafruit_BMP280** and **Adafruit_Sensor** libraries from Particle photon libraries. There are total 3 debug levels in code which will help to understand the errors in the code if any. For Project1 tasks I used **Twilio** webhook for message and **IFTTT** for a voice call. The message sends a outside pressure and the voice call tells the temperature. I used **AWS IoT** webhook to send the data which is published on particle console to the **DynamoDB**. AWS IoT uses **lambda** function, which lets us run the code without managing servers. DynamoDB is a fully managed NOSQL database, which allows us to create database tables and store unlimited data which can serve any level of request Traffic. **IAM** is a AWS Identity and Access Management service which allows to control the authentication and authorization of data.

3. Code:

```
// This #include statement was automatically added by the Particle IDE.
#include <Adafruit_BMP280.h>

// This #include statement was automatically added by the Particle IDE.
#include <Adafruit_Sensor.h>

// Debug levels to print the error message

#define DBGLVL0 1

#define DBGLVL1 1

#define DBGLVL2 1

Adafruit_BMP280 bmp; // I2C


void setup() {
  // To initialise Serial communication
  Serial.begin(9600);

  if (DBGLVL0) Serial.println(F("Device is on"));

  if (bmp.begin((0x76))) {
    if (DBGLVL1) {
      Serial.println(F("Connection to BMP is successful"));
    }
  }
  else{
    if (DBGLVL1) Serial.println("Connection to BMP is not successful");
  }
}

void loop() {
  if (DBGLVL2) {
    // To print the data on serial port
```

```
Serial.print(F("Temperature = "));  
  
// Reads the temperature from the temperature sensor  
  
Serial.print(bmp.readTemperature());  
  
// To get the temperature in Celsius  
  
Serial.println(" *C");  
  
}  
  
// Store the temperature in String  
  
String data = String(bmp.readTemperature());  
  
// Publish the temperature to console  
  
Particle.publish("temperature", data, PRIVATE);  
  
if (DBG_LVL2) {  
    // Print the data on serial port  
  
    Serial.print(F("Pressure = "));  
  
    // Reads the pressure from sensor  
  
    Serial.print(bmp.readPressure());  
  
    // To get the pressure in Pascals  
  
    Serial.println(" Pa");  
  
}  
  
// Store the pressure in String  
  
String data1 = String(bmp.readPressure());  
  
//Publish the Pressure to console  
  
Particle.publish("pressure",data1, PRIVATE);  
  
// Publish a message for pressure using twilio webhook  
  
Particle.publish("twilio_sms", data1, PRIVATE);  
  
// Delay of 10000 milliseconds  
  
delay(10000);  
  
}
```