IoT Based Remote Monitoring of Temperature and Pressure Using Arduino, BMP180 Sensor Module and Beebotte Cloud Server

[1]Adnya A. Shirgaonkar, [2]Dr. Vivek V. Bhide

Assistant Professor
R. E. Society's, R. P. Gogate College of Arts and Science & R. V. Jogalekar College of Commerce,
Ratnagiri, Ratnagiri -415612, Maharashtra, India

Abstract: **The complete working of this system is divided into two parts. First part deals with data acquisition of parameters, like, temperature and pressure, and in second part, this acquired data is uploaded to the cloud server. The system software involves two programs. One program is written in Arduino IDE and the other one is written in Python. In the first part, BMP180 sensor module is connected to Arduino UNO through I2C bus. Arduino UNO is programmed through Arduino IDE to acquire values of temperature and pressure. In the second part, Arduino UNO is python programmed in which the acquired data, i.e., values of temperature and pressure are accessed serially in a python program and these values are uploaded to the Beebotte cloud server. Data acquisition and data upload steps are repeated after time interval of 30 seconds and this will go on continuously. The main feature of this system is that the acquired data in available to the world almost in real time.**

Keywords: **Arduino UNO, BMP180, Data acquisition, IoT, Beebotte, Python, Arduino IDE**

I. INTRODUCTION:

Data acquisition is very much essential and important in almost all kind of research, technology, education, industries, etc. [1]. Along with data acquisition, the development of such kind of systems or devices which can acquire the remote data and which can provide the data to the world in real time, is also the need of today's world [2]. Along with development of such kind of systems or devices, their improvised versions, is also the topic of research nowadays. Along with having remote and real time data acquisition feature, the system should also involve basic features, like, it should acquire less space, it should cause less power consumption, it should be low cost, it should be compact and robust, and presence of this system shouldn't disturb the surrounding [3,4].

So, an attempt has been made to develop such kind of IoT based system or device, which is programmed in such a way that, it continuously gathers the data from sensors and quickly uploads the data to the cloud server, so that, acquired data is available to the world almost in real time. Also, this device can be installed at remote or difficult to reach locations. Once it is installed, it can continuously acquire the data and quickly upload that data to the cloud server, which allows continuous monitoring of the data from any corner of the world and also enables to take data-driven decisions, if needed. This device can further be used for the purpose of automation, as well as, to monitor environmental conditions, such as, weather parameters, which overall reduces the need of manual intervention.

BMP180 sensor module [5] is used to sense temperature and pressure. BMP180 is combined sensor module which includes temperature sensor and pressure sensor. Its operating voltage is 1.71 V to 3.6 V. Both these sensors have high accuracy, high resolution, low noise and faster response time. Also, this module can be easily mounted. Its small dimensions and low power consumption makes it suitable for this system. This sensor module provides I2C interface.

The intelligent system used to process the acquired data is Arduino UNO [6]. Arduino UNO is microcontroller board which uses 8-bit microcontroller ATmega328P. It is low cost, compact and open source microcontroller board. It has 32 KB of flash memory for storing user's program. The board also has 2 KB of SRAM which is used to store data while program execution. It has 1 KB of EEPROM. Its recommended input voltage requirement is 7-12V dc. It has 14 digital I/O pins, out of which 6 provides PWM output. It has 6 analog input pins, 16 MHz crystal oscillator, a USB connection, a power jack and a reset button. Also, it has UART, I2C and SPI peripherals. It can be programmed by connecting it to a computer with a USB cable. So for development purpose, the board is to be used with computer or laptop and once it is programmed, by powering this board up with AC-to-DC adapter, it can be used as stand-alone, i.e., without computer or laptop.

The Arduino board is programmed using Arduino Integrated Development Environment (IDE) [6] and Python Integrated Development and Learning Environment (IDLE) [7]. Arduino IDE is free software that allows developers to write and upload code to the Arduino board. The programming language of this IDE is very much similar to C, C++. Another programming language used to write system's program is Python. Python is open source, object oriented, high level and interpreted language. It provides GUI programming support. Also, it has large standard library collection. Because of all these reasons, it is very much popular language for data analysis, machine learning, automation, web development, scripting, IoT based applications, etc. Because of all these features, the second programming language, i.e., Python is used to process the acquired data further and to upload and store the data on Beebotte cloud server.

The cloud server that is used to store the acquired data is Beebotte. It provides visual dashboard. The dashboard provides variety of widgets, like, charts, maps, gauges, etc. It allows the users to visualize data in the form of these widgets and allows users to monitor the data in real time from any corner of the world with the help of dashboard link of the system.

Arduino board is programmed through Arduino IDE to read data from sensors and then Arduino board is python programmed to read the acquired data serially and to upload and to store this data to respective resources of Beebotte cloud server.

II. EXPERIMENTAL DETAILS

First of all, a user account is created on Beebotte cloud server, in order to upload and store data on Beebotte platform. This is followed by channel creation which represents the system, e.g., here the channel name is "Arduino_Based_Temperature_and_Pressure_Measurement_1". After that, different resources can be created under the channel which belongs to that channel. Here, resources are "Temperature1_Deg_Celcius" and "Pressure1_hPa". This is followed by dashboard creation for the channel, where uploaded data can be visualized. To visualize acquired data on Beebotte dashboard, two visualization widgets, i.e., "Timeline Chart" and "Table Chart" are created for every Beebotte resources, so that, data for temperature and pressure is visualized on dashboard in timeline graph format and in table format. Then dashboard is made "public", so that, by sharing this dashboard link, the uploaded data can be monitored from any corner of the world. Later, these resources, channel and account are accessed through python program, so that, the acquired data can be uploaded and stored to this server.

This system is made to acquire data like temperature and pressure of surrounding after interval of 30 seconds and to store this data on Beebotte cloud server. Arduino UNO is programmed through Arduino IDE to read data from BMP180 sensor module and then Arduino UNO is python programmed to read acquired data serially and to upload and store data on Beebotte cloud server. BMP180 sensor module supports I2C interface. In order to establish I2C interface between Arduino UNO and BMP180, the SDA pin of BMP180 sensor module is connected to SDA pin of Arduino UNO and the SCL pin of BMP180 sensor module is connected to SCL pin of Arduino UNO. Fig. (1) is the block diagram of the system.

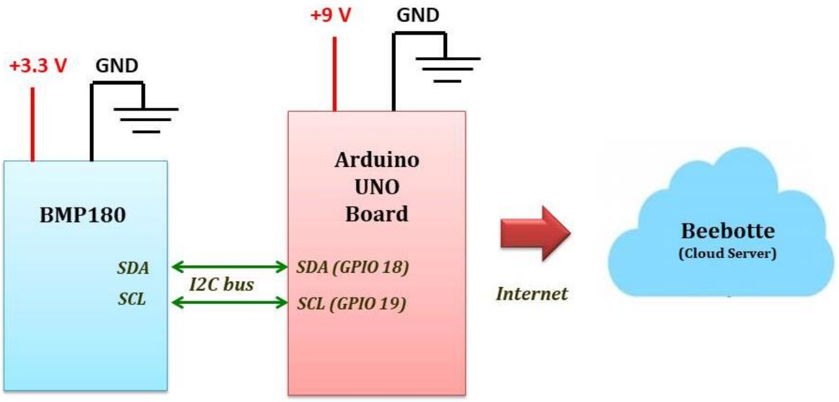


Fig. (1): Block diagram of System

Fig. (2) is the system photograph which shows BMP180 sensor module is connected to Arduino UNO board through I2C bus. Arduino UNO board is powered with computer through USB connector. BMP180 sensor module is provided with 3.3V dc power through on-board power lines of Arduino UNO board. Arduino UNO board is connected to internet over wifi connectivity, so that, acquired data, i.e., temperature and pressure will be uploaded to Beebotte cloud server.

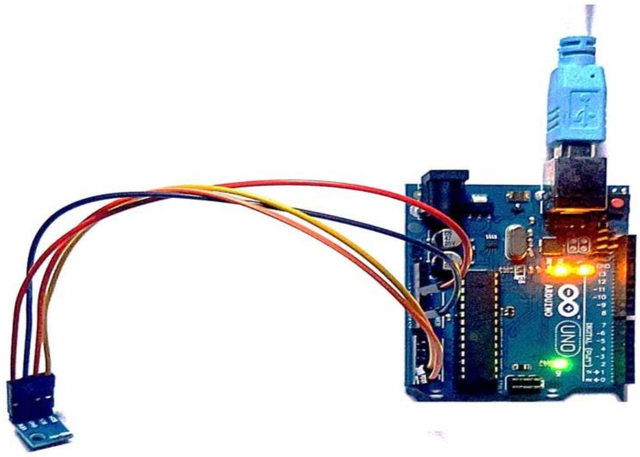


Fig. 2: System Photograph

Then, following python libraries are installed in order to achieve the given task.

- pyserial: This library allows Python program to communicate with devices over a serial port. Here, pyserial is used to carry out communication between Python code and Arduino UNO. With the help of this library, it is possible to open serial port, read and write data over serial port by configuring various communication parameters like baud rate, parity, etc.
- beebotte: This library allows Python program to communicate with Beebotte platform in order to send the data to the Beebotte dashboard.

Arduino UNO is first programmed through Arduino IDE and then Arduino board is python programmed in which-

A. Programming through Arduino IDE:

i. The SDA and SCL pins of Arduino UNO are configured to enable I2C interface between Arduino UNO and BMP180 sensor module.
ii. The sensor output of BMP180, i.e., values of temperature and pressure are read and these values are serially transmitted to Python code.
iii. Step A-ii) is repeated continuously.

B. Simultaneously following Python code executes which does following task:

i. The Beebotte account "saa" and channel created for the system "Arduino_Based_Temperature_and_Pressure_Measurement_1" are accessed through this program by providing necessary login credentials, like secret key and token.
ii. Baud rate is initialized as 9600 bps.
iii. Values of temperature and pressure which are serially transmitted by A – ii) are read through Python code
iv. These values are temporarily stored in respective variables, like, "temperature_r" and "pressure_r".
v. This acquired data is uploaded and stored to respective resources – "Temperature1_Deg_Celcius" and "Pressure1_hPa" of the Beebotte channel – "Arduino_Based_Temperature_and_Pressure_Measurement_1". This step takes almost 15 seconds to execute.
vi. Steps B-iii), B-iv) and B-v) are repeated after interval of 30 seconds by considering time taken to read data from sensors and time taken to upload data as mentioned in step B-v).

In this way, acquired data, i.e., values of temperature and pressure are uploaded to the cloud server. This uploaded data can be visualized on system's dashboard in the form of table chart and timeline chart as shown in Fig. (3).

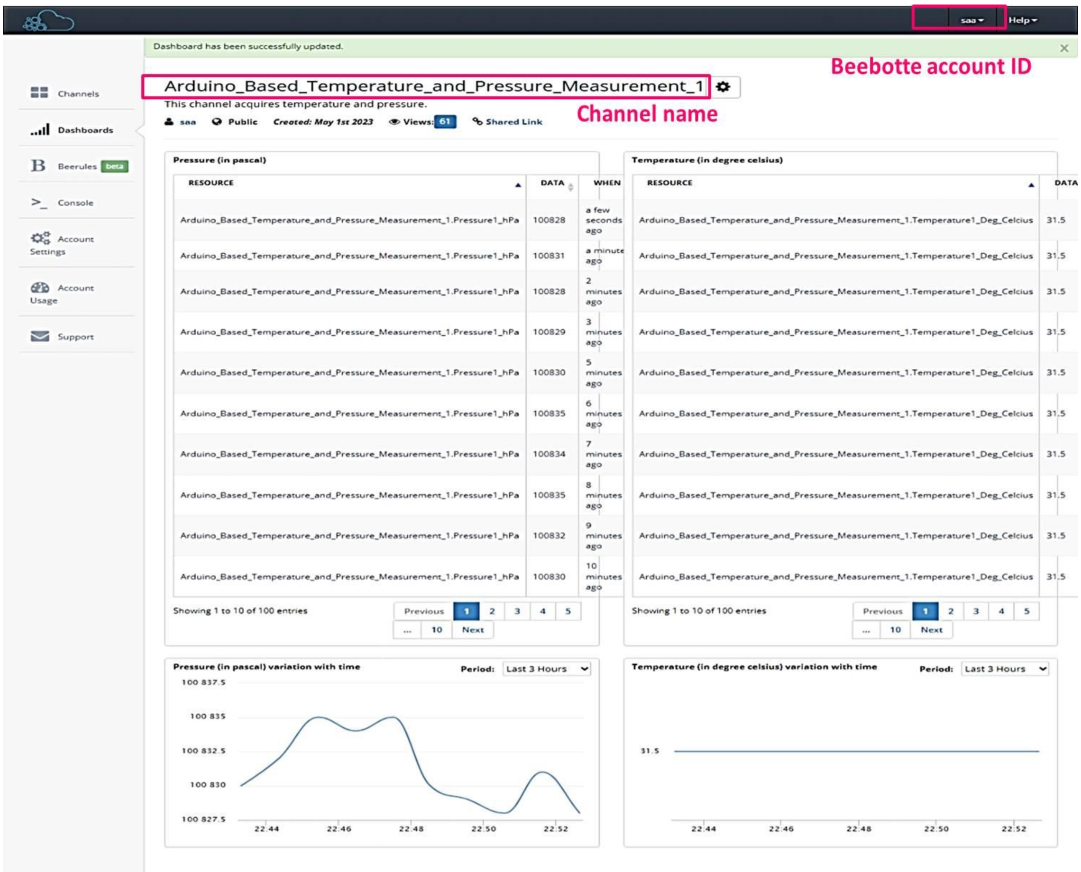


Fig. 3: Screenshot of System's dashboard

This dashboard link of this channel can be shared through any message sharing platform and as this channel is made "public", the uploaded data can be observed from any corner of the world at real time. Data modification will only be allowed to the Beebotte account holder and not to the observer.

III. RESULT AND ANALYSIS

Fig. (3), Fig. (4), Fig. (5), Fig. (6) and Fig. (7) are screenshots of system's dashboard. After every data acquisition process, the acquired data, i.e., the values of temperature and pressure are uploaded to the respective resources created for the Beebotte channel "Arduino_Based_Temperature_and_Pressure_Measurement_1". The uploaded data to the Beebotte cloud server is available on system's dashboard in table chart format and in timeline graph format as shown in Fig. (4), Fig. (5), Fig. (6) and Fig. (7).

Fig. (4) is the screenshot of system's dashboard which shows timeline chart for temperature resource, where graphical variation of temperature in °C with time is shown. Following data is the latest collected data. Previously collected data is also available on dashboard. By selecting appropriate option from dropdown list of 'Period' option, previously uploaded data can be observed.

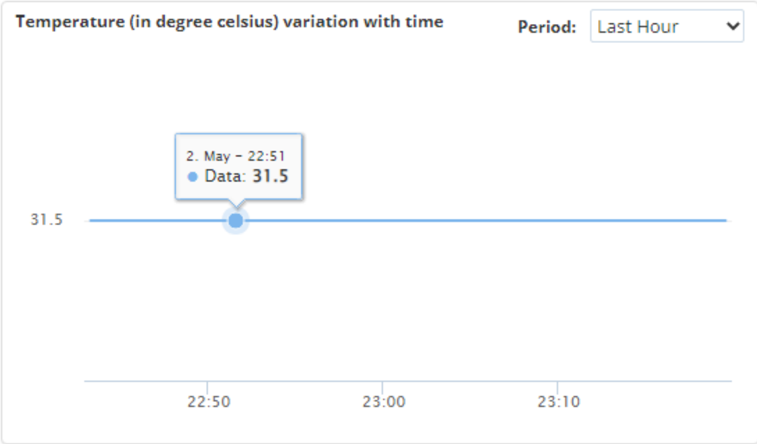
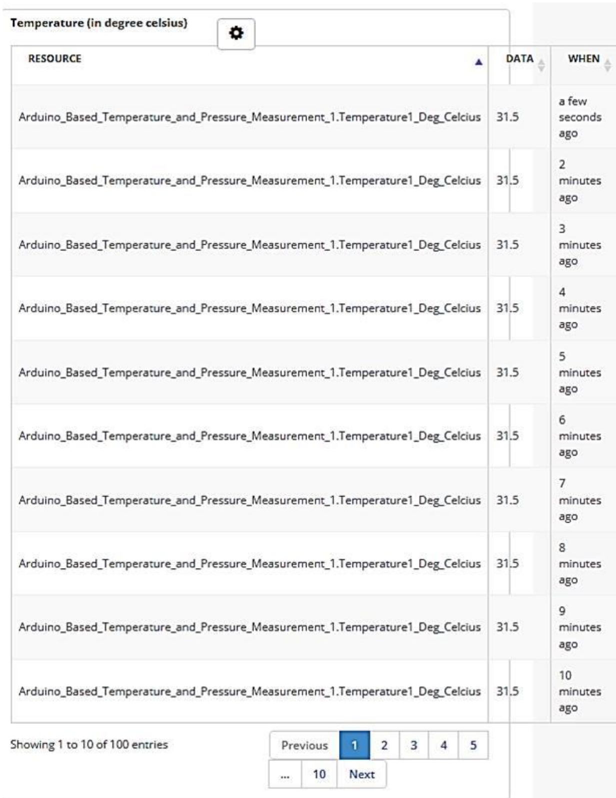


Fig.4: Screenshot of system's dashboard: Graphical variation of temperature in °C with time

Fig. (5) is the screenshot of system's dashboard. This tabular chart shows acquired values for temperature resource in °C with time.



Temperature (in degree celsius)

RESOURCE	DATA	WHEN
Arduino_Based_Temperature_and_Pressure_Measurement_1.Temperature1_Deg_Celcius	31.5	a few seconds ago
Arduino_Based_Temperature_and_Pressure_Measurement_1.Temperature1_Deg_Celcius	31.5	2 minutes ago
Arduino_Based_Temperature_and_Pressure_Measurement_1.Temperature1_Deg_Celcius	31.5	3 minutes ago
Arduino_Based_Temperature_and_Pressure_Measurement_1.Temperature1_Deg_Celcius	31.5	4 minutes ago
Arduino_Based_Temperature_and_Pressure_Measurement_1.Temperature1_Deg_Celcius	31.5	5 minutes ago
Arduino_Based_Temperature_and_Pressure_Measurement_1.Temperature1_Deg_Celcius	31.5	6 minutes ago
Arduino_Based_Temperature_and_Pressure_Measurement_1.Temperature1_Deg_Celcius	31.5	7 minutes ago
Arduino_Based_Temperature_and_Pressure_Measurement_1.Temperature1_Deg_Celcius	31.5	8 minutes ago
Arduino_Based_Temperature_and_Pressure_Measurement_1.Temperature1_Deg_Celcius	31.5	9 minutes ago
Arduino_Based_Temperature_and_Pressure_Measurement_1.Temperature1_Deg_Celcius	31.5	10 minutes ago

Showing 1 to 10 of 100 entries

Previous 1 2 3 4 5 ... 10 Next

Fig.5: Screenshot of system's dashboard: Acquired data for temperature in °C with time in tabular format

Fig. (6) is the screenshot of system's dashboard. This timeline graph shows acquired values for pressure resource, where graphical variation of pressure in pascal with time is shown.

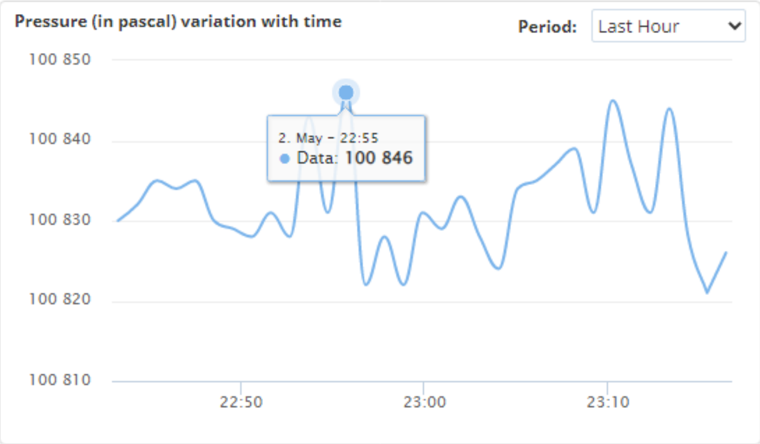


Fig.6: Screenshot of system's dashboard: Graphical variation of pressure in pascal with time

Fig. (7) is the screenshot of system's dashboard. This tabular chart shows acquired values for pressure resource in pascal with time.

Pressure (in pascal)

RESOURCE	DATA	WHEN
Arduino_Based_Temperature_and_Pressure_Measurement_1.Pressure1_hPa	100828	9 minutes ago
Arduino_Based_Temperature_and_Pressure_Measurement_1.Pressure1_hPa	100831	10 minutes ago
Arduino_Based_Temperature_and_Pressure_Measurement_1.Pressure1_hPa	100828	11 minutes ago
Arduino_Based_Temperature_and_Pressure_Measurement_1.Pressure1_hPa	100829	12 minutes ago
Arduino_Based_Temperature_and_Pressure_Measurement_1.Pressure1_hPa	100830	13 minutes ago
Arduino_Based_Temperature_and_Pressure_Measurement_1.Pressure1_hPa	100835	14 minutes ago
Arduino_Based_Temperature_and_Pressure_Measurement_1.Pressure1_hPa	100834	16 minutes ago
Arduino_Based_Temperature_and_Pressure_Measurement_1.Pressure1_hPa	100835	17 minutes ago
Arduino_Based_Temperature_and_Pressure_Measurement_1.Pressure1_hPa	100832	18 minutes ago
Arduino_Based_Temperature_and_Pressure_Measurement_1.Pressure1_hPa	100830	19 minutes ago

Showing 1 to 10 of 100 entries   Previous 1 2 3 4 5 … 10 Next

Fig.7: Screenshot of system's dashboard: Acquired data for pressure in pascal with time in tabular format

This system is installed at a certain place in Ratnagiri. The acquired values for temperature and pressure are verified with weather station's data.

IV. Conclusion

The "IoT Based Remote Monitoring of Temperature and Pressure Using Arduino, BMP180 Sensor Module and Beebotte Cloud server" is designed, built, programmed and tested successfully. The above system is installed at a certain place in Ratnagiri and values of temperature and pressure are acquired. These acquired values of temperature and pressure are verified with weather station's data at Ratnagiri for the same span of time. The acquired data is found to be in accordance with weather station's data.

REFERENCES:

1. Vladinir Vujovic, Mirjana Maksimovic, 11th International conference "Interactive Systems: Problems of Human-Computer Interactions", PP 57-62, Ulyanovsk, Russia 2015.
2. Mr. Nilesh Magare, Ms. Pooja Bharajkar, Mr. Saurabh Bansod, International Research Journal of Engineering and Technology (IRJET), Volume: 06 Issue: 07, July 2019, PP 372-374.

3. Ioan Lita, Ion Bogdan Cioc, Ilie Popa, IEEE Conference, 2006, 29th International Spring Seminar on Electronics Technology DOI: 10.1109/ISSE.2006.365130
4. Girish Bekaroo, Aditya Santokhee, 2016 IEEE International Conference on Emerging Technologies and Innovative Business Practices for the Transformation of Societies (EmergiTech), DOI:10.1109/EmergiTech.2016.7737367
5. BMP180 datasheet
6. Arduino UNO – Product reference manual
7. Dhruv A., Patel, R. and Doshi N., Proceedings of the International Conference on Culture Heritage, Education, Sustainable Tourism, and Innovation Technologies (CESIT 2020), PP 292-299, DOI: 10.5220/0010307900003051