

DESIGN OF AN IOT SMART METEOROLOGICAL STATION COMPLETE WITH DATA ACQUISITION AND REALIZING TEST MEASUREMENTS IN CYBER-PHYSICAL ROBOTICS LAB

István FARAGÓ,¹ Timotei István ERDEI,² Sándor ZOLNAI,³ Géza HUSI⁴

University of Debrecen, Faculty of Engineering, Debrecen, Hungary

¹ istu.fg@gmail.com

² timoteierdei@eng.unideb.hu

³ zsanyi007@gmail.com

⁴ husigeza@eng.unideb.hu

Abstract

The presented project was completed using modern, widely used IoT based devices. These include the ESP-01 microcontroller, which is programmable using the Arduino IDE (with its integrated ESP8266 interpreter), and a Raspberry Pi Zero W single board computer, which requires a Raspbian Linux distribution. In this project, an original meteorological station was created (complete with data acquisition), employing various sensors.

Keywords: *IoT, ESP, Raspberry Pi, data acquisition, Linux.*

1. Introduction

The project was designed and assembled in the cyber-physical robot lab of the mechatronics research center, University of Debrecen [1].

The most important part of IoT networks are the sensors which provide us with data about changes in the surrounding world. The changes can be transmitted, stored, and processed into forms we desire. The relevance of this communication of data is ever growing in the current economy.

It's important for these sensor and transmitter devices to be low power, as an external power source might not be available, making them dependent on batteries for their operation. Wireless technology is likewise important for data acquisition.

2. IoT based remote access

It's important for the required data to be accessible from anywhere, anytime, even at a completely different geographical location. This functionality is provided by the internet. Using the web, a remotely accessible data acquisition device can be designed, and operated.



Figure 1. *The completed meteorological station*

This remote data acquisition solution can be achieved from any device, as long as it is connected to the internet. To achieve this, an Apache webserver was used, which contained a webpage written in PHP, with an included MySQL database and its necessary components. This server can be reached using the Linux SSH (secure shell) protocol, running on the Raspberry. Using Windows, a client program (such as Putty) is required to connect to this SSH server. By typing in the required commands, the required software can be installed, after which the webserver can be reached using the controller’s IP (Internet Protocol) with the correct username and password. **Figure 1** shows the completed meteorological station

3. Raspberry Pi & ESP 8266

3.1. The Raspberry Pi

At the beginning of 2017, on the occasion of the 5th anniversary of Raspberry Pi, the Raspberry Pi Zero W was released, equipped with a 1 GHz single core BCM2835 ARM based processor, and 521MB of RAM. It also has built in WiFi and Bluetooth 4.1 thanks to its integrated radio frequency antenna, designed by the manufacturer itself. The amount of GPIO ports are the same as its larger predecessor, and similarly it supports serial, IIC, and SPI interfaces. The unit is directly programmable using a Linux distribution, complete with GUI [2].

The project’s data acquisition system is built on this device. The device runs a Raspbian operating system, which is a Debian based system optimized for Raspberry Pi. Debian systems currently utilize a Linux or FreeBSD kernel [3].

3.2. ESP8266

For this project, the first model, the ESP-01 was chosen. Although it only has two outputs by default, with some basic soldering additional ports may be opened [4]. The device with its annotated ports is shown on **figure 2**.

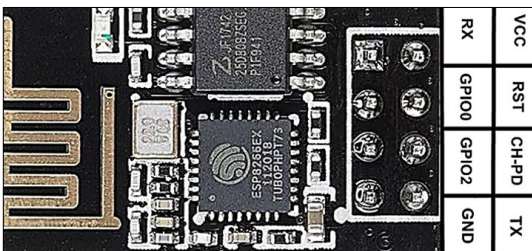


Figure 2. ESP8266-01

4. Sensors

4.1. BME280

One of the Bosch Sensortech sensors is the BME280, which is an integrated environmental sensor intended for mobile devices. During its design, the key elements were low energy consumption, high linearity, and high precision during the measurement of pressure, humidity and temperature. The BME280 is a flexible sensor with high linearity, high accuracy, and low energy consumption. It supports SPI interfaces, so it may freely communicate with the microcontroller.

This, and any other mentioned sensor can be seen on **Figure 3**.

The sensor’s basic specifications are as follows:

- SPI interfaces,
- three integrated sensors,
- Industrial temperature range of $-40...+85\text{ }^{\circ}\text{C}$.

4.2. GY-30 (BH1750)

The GY-30 precision light sensor, similar to the BME280, has a bus interface and also analog outputs. Its colour sensitivity is about the same as the human eye, able to provide a wide measurement range and high resolution at a low power consumption.

The sensor’s basic specifications are as follows:

- interface,
- 50/60 Hz noise filtering,
- industrial temperature range of $-40\text{ }^{\circ}\text{C}...+85\text{ }^{\circ}\text{C}$.

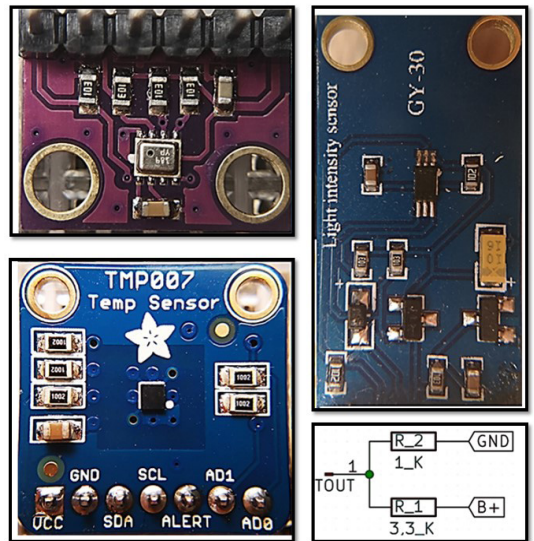


Figure 3. GY-30 sensor, BME280, TMP007 sensor, voltage divider

4.3. TMP 007 contactless temperature sensor

The TMP007 is Texas instrument’s latest contactless temperature sensor and is the successor to the TMP 006. The internal mathematical unit is responsible for all the temperature calculations, so the IIC interface can easily read this provided data. The sensor is able to measure an object’s temperature without coming into contact with it or measure the sensor cell’s temperature.

- IIC interface,
- 14-Bit (0,03125 °C) resolution,
- industrial temperature range of -40...+125 °C,
- low power consumption,
- small size DSBGA package.

5. Calculating dew point and perceived temperature

Thanks to the data provided by the BME280, dew point can be calculated. This can be done with a simple equation using relative humidity, and current temperature, with 1% precision, assuming that relative humidity is above 30%.

By careful placement of the device, it’s possible to find critical points in a building part, and temperatures dropping below dew point due to thermal bridges [5].

6. Connections

The BME280, TMP007, and GY-30 sensors must be connected to the SDA and SCL pins [GPIO 0, GPIO 2], and also must be provided with 3,3 V power. The two serially connected resistors’ connection point must be connected to TOUT. The

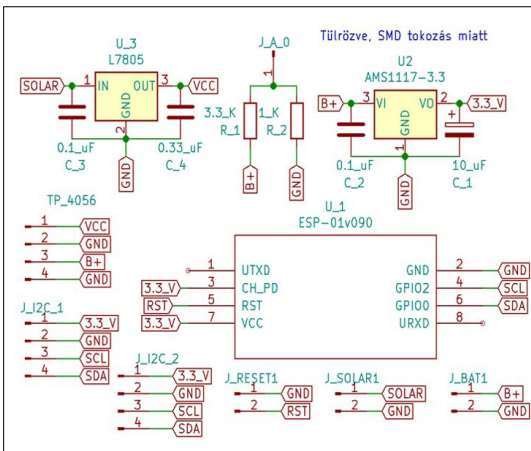


Figure 4. Schematic wiring diagram

entire system uses an 18650 battery as a power source, which is charged by a 10 V solar panel producing up to 1 W.

The system is provided by 3,3V power using a TS1117 voltage stabilizer. By default, the ESP8266-01 the XPD_DCDC pin is not connected to the RESET pin, so we’re unable to use the “deep sleep” function. Soldering these pins together solves the aforementioned issue. These connections can be observed on Figure 4.

A TP4056 Lithium-ion battery charger is also incorporated into the system, to supply the battery with the correct charging voltage, and also to protect against overcurrent. This device is sensitive to the input voltage fluctuations, so an LM7805 LDO is used to reduce the fluctuating 10 V output to a stable 5 V, as the TP4056 only accepts charging voltages as high as 5.5 V.

7. Making of the printed circuit board

After completing the schematic diagram, the PCB plan was also made in KiCad. KiCad is made in an open source integrated software package, and is used for creating circuit diagrams and PCB layouts. While designing the PCB layout, the two most important aspects were small size, and suitable placement of the devices, as seen on Figure 5.

The program is also able to review the plans in 3D, providing an excellent opportunity to find design errors.

The PCB was created using a subtractive method. Its main principle is that the copper-foil coated board is covered with a chemically resistant layer at places where the desired circuit is. Then later, the excess copper is removed chemically. The completed PCB is then usually covered by a protective layer, which is most often a specialized

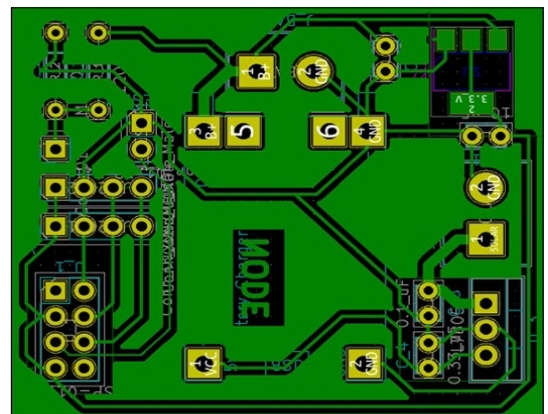


Figure 5. PCB

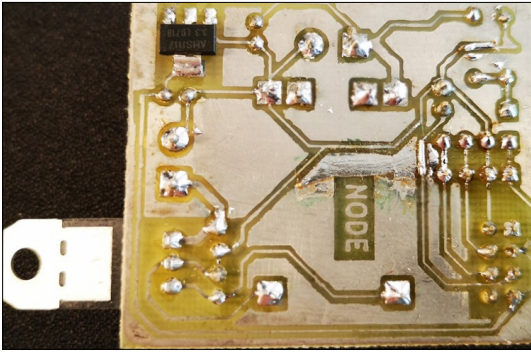


Figure 6. Finished, tinned PCB

lacquer, or chemical tin surface finish, which also increases conductivity. After tinning, a lacquer layer is still used however, mainly to improve electrical separation in cases of high humidity.

The final result is shown on Figure 6.

8. Programs

The project’s controlling program was written in Arduino IDE (1.6.11), using its C/C++ based programming language. The PHP code running on the Raspberry Pi was written in Notepad++ text editor.

The meteorological station program is written in such a way that at specific intervals the it sends ESP POST requests using the sensors’ data, which gives the API keys (security code) and sensor data.

Then, if the API key matches with the one “burnt” into the PHP code, then the data is stored in a pre-defined table (INSERT command). This is achieved by built-in MySQL commands. Later, the PhpMyAdmin web-based client can be used to access the database.

POST data is not shown in the browser, unlike GET requests. However, this is not much safer, as

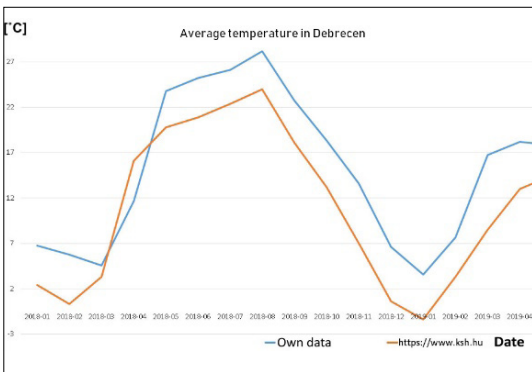


Figure 7. Comparison of measured data [6]

the sent packages may be snooped on the sensors’ network (where the ESP connects to the Raspberry).

Accessing the database on a local network is completely user friendly, only the server’s IP address has to be typed into a web browser of our choice. This leads us first to a login window where we a valid username–password combination is required to access the tables. The data found here are ordered by their unique ID (based on when they were sent), which may be deleted, or modified. Downloading from the database is possible in various formats (for example .csv, or .clsx file formats. The database entries can later be processed to create reports, or graphs similar to that shown on Figure 7.

9. Packaging

The processed data was later compared to data shared by “ksh.hu”. Based on this comparison, it can be said that based on the tendencies, temperature measurement is good, but due to non-optimal design, temperatures were much higher.

Battery charge was also unsatisfactory, especially in the colder months, as seen from the amount of sent data. To avoid low battery levels, the original 6 V solar panel was replaced with a 10 V one.

After several other designs and analyses a 3D design was made to scale in fusion 360, where the box was designed in such a way that it can be produced using 3D printing.

After all parts were ready, assembly and placement could proceed. The outer unit was placed further from the building. Thanks to its white color, the sun does not warm it up as much, thus, more precise temperature measurement can be obtained.

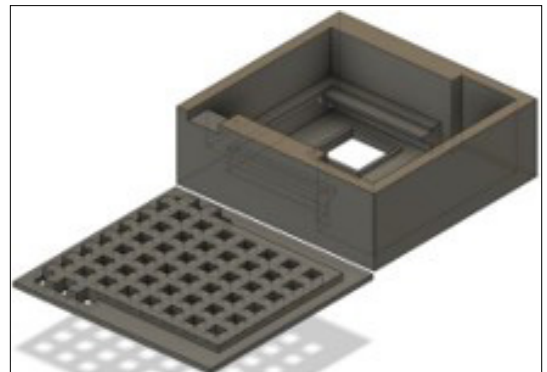


Figure 8. Designed internal box

The unit's internal design is entirely different. Under the circuit the TMO007 is located, and through a window a flat surface's temperature can be measured. The BME280 is located in the box itself, and to ensure proper ventilation, the top of the box has a mesh structure, as seen on [Figure 8](#).

10. Conclusions

The meteorological station and data acquisition system was designed, the circuit was assembled, and the program was written. The IoT based network communication is suitable for the assembled system, the remote supervision method also suits the project's needs.

The Linux based peripheral performs well, in spite of its low hardware specs, is perfectly suitable for the data base's needs, and during its test operation, no slowdowns, freezes, or network issues were observed.

The query of data is possible on both PC and mobile phones, displaying data was made possible. Later, the system can also be equipped with radio frequency modules, thus, data collection became possible even at locations, where there's no network connection. Moreover, the amount of col-

lected information can also be increased by adding additional sensors. The project was designed and assembled in the cyber-physical robot lab of the mechatronics research center, University of Debrecen.

Acknowledgement

This research was supported by the Doctoral School of Information Technology, University of Debrecen.

References

- [1] T. I. Erdei, Zs. Molnár, N. C. Obinna, G. Husi: *Cyber physical systems in mechatronic research centre*. MATEC Web Conf. Volume 126, 2017.
- [2] Raspberry Pi Family (accessed on: 2019.11.20) <https://www.raspberrypi.org/blog/raspberry-pi-zero-w-joins-family/>
- [3] Linux Debian (accessed on: 2019.11.21) <https://www.debian.org/intro/about>
- [4] ESP8266 (accessed on: 2019.11.23) https://butykoltem.blog.hu/2016/07/27/csinald_magad_iot_esp8266
- [5] Szász G., Tőke L.: *Meteorológia mezőgazdáknek, kertészeknek, erdészeknek*. Mezőgazda Kiadó, Budapest, 1997.
- [6] KSH (accessed on: 2019.11.23) https://www.ksh.hu/docs/hun/xstadat/xstadat_evkozi/e_met002.html