

Three-Axis Level with WiFi Connection

Cristian-Mircea Muscai, Ion-Cornel Mituletu*, Tiberiu Manescu, Liviu-Bogdan Protea

The Three-Axial Level (TAL) device measures the angular inclination around the three axes of the orthogonal system. It consists of a triple-axis magnetometer sensor with I²C transmission and a NodeMCU V3 development board with WiFi module integrated. The device processes data from sensor, displays it on its screen and also transmits the angular values to various PADs (Portable Access Device). The experimental testing confirmed the initial expectation, showing the properly on-site operation of the designed device. This device is useful in the supervising process to avoid perturbing the worker that performs the leveling operation or when the working place is hard to reach.

Keywords: ESP8266, NodeMCU V3, OLED display, standalone web server, triple-axis magnetometer

1. Introduction

Wireless devices and systems create a smart environment by connecting sensors and transducers to develop various industrial and home automation / domotics applications. These equipments are usually embedded and can work independent or integrated in a larger system. Also, the internet provides the possibility to connect a large number of things (Internet of Things - IoT), devices and systems, resulting in smart virtual grids able to exchange important amount of data allover the world.

Based on low-cost development boards and sensors lately they were developed a large gamma of new gadgets, systems, algorithms and software.

As the first example, a cost-effective and efficient system designed to monitor drinking water quality which makes use of Internet of Things (IoT) technology is given. This system consists of several sensors to measure various water parameters and a microcontroller unit interfaced with these sensors, and further processing is

performed at personal computer. The obtained data is sent to the cloud by using IoT based ThinkSpeak application to monitor the quality of the water [1].

Also with the use of IoT platform and visual analytics, a real-time water consumption monitoring system has been developed for household and commercialized building applications. In order to get the water consumption data, a Hall Effect water flow sensor is attached to every faucet, which is then processed by a NodeMCU, then pushed to the Heroku cloud database [2].

In other paper are presented and analyzed the design and implementation aspects of open source Supervisory Control and Data Acquisition software. Case uses of the software are also discussed, mostly focusing in embedded computational platforms and microcontroller-based prototyping boards. The main objective of this software is to provide a low cost, open source and extensible tool for open source and scientific users [3].

Short Term Load Forecasting (STLF) is an essential aspect for smart grid operations such as power dispatch and load management. Therefore, cost effective Arduino Uno, Node MCU/ESP8266, PZEM 004T and DHT 11 sensors are used to achieve the hardware model that acquire the data for a novel online load forecasting, using supervised Machine Learning (ML), algorithms in Internet of Things (IOT) environment [4].

In this paper the accomplishment of a device which measures the angular values around three orthogonal axes is presented. Two of the axes define the absolute horizontal plane, and the third is along the gravitational acceleration direction. Also, the device features ensure the hold of real-time values, recalibration to zero at any desired angle value and wireless data transmission to PADs, mobile devices like phone, tablet and laptop. Zero calibration at any value offers an easier way to repetably achieve an desired angle value, like 13.7 degrees.

2. Design of the device

The device's structure shows the functional principle and does not intend to limit it at the use of these components, because their properly replace can produce higher performances. Thus, the block diagram is presented in figure 1.

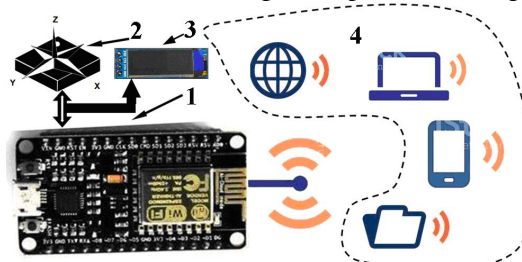


Figure 1. Device block diagram.

The structure from figure 1 is based on NodeMCU V3 development board (1), the values of the angles are given by the triple-axis magnetometer sensor (2), after the processing of the values they are shown on the display (3), and in area (4) the symbols of the PADs are presented, which are the mobile devices possible to connect.

The most significant features of the components are given below:

1. NodeMCU V3: the development board equips the ESP-12E module containing ESP8266 chip having Tensilica Xtensa® 32-bit LX106 RISC microprocessor which operates at 80 to 160 MHz adjustable clock frequency and supports RTOS. There's also 128 KB RAM and 4 MB of Flash memory (for program and data storage) just enough to cope with the large strings that make up web pages, JSON/XML data, and everything we throw at IoT devices nowadays. The ESP8266 Integrates 802.11b/g/n HT40 Wi-Fi transceiver, so it can not only connect to a WiFi network and interact with the Internet, but it can also set up a network of its own, allowing other devices to connect directly to it. It has a total of 17 GPIO pins broken out to the pin headers on both sides of the development board. These pins can be assigned to all sorts of peripheral duties, including:

- A 10-bit ADC channel;
- UART interface is used to load code serially;
- PWM pins for dimming LEDs or controlling motors.
- SPI and I²C interface to link up all sorts of sensors and peripherals, and I²S interface if you want to add sound to your project.

2. Triple-axis magnetometer sensor QMC5883L is based on state-of-the-art, high resolution, magneto-resistive technology. Along with custom-designed 16-bit ADC ASIC, it offers the advantages of low noise, high accuracy, low power consumption, offset cancellation and temperature compensation. It enables 1° to 2° compass heading accuracy. The I²C serial bus allows for easy interface.

3. Display is a 128x32 pixel monochrome OLED. It is small, only about 1" diagonal, but very readable due to the high contrast and contains 128x32 individual white OLEDs, each one is turned on or off by the controller chip. Because the display makes its own light, no backlight is required, the power required to run the OLED is reduced.

Programming NodeMCU by ArduinoIDE requires a few steps. These are shortly presented below, more detailed aspects are given in the article "NodeMCU V3 For Fast IoT Application Development" [5]:

1. Installing the firmware;
2. Preparing the Arduino IDE;
3. Source code elaboration.

To easily achieve the source code, a flowchart must be realized based on the logical functionality of the device and also each component added to the develop-

ment board has to be tested separately, hardware and software. In the initial testing process a few important things have to be highlighted:

- variants of sensors are on the market that seem to be the same, but they are not, therefore the correct software must be identified, and sometimes this is not easy;

- attention has to be drawn to the component wiring, else ambiguous behavior could appear;

- I²C right addresses have to be identified, for this an I²C scanner can be used, it is easy to find on the internet.

A simplified flowchart is presented in figure 2, it defines just the important functional aspects.

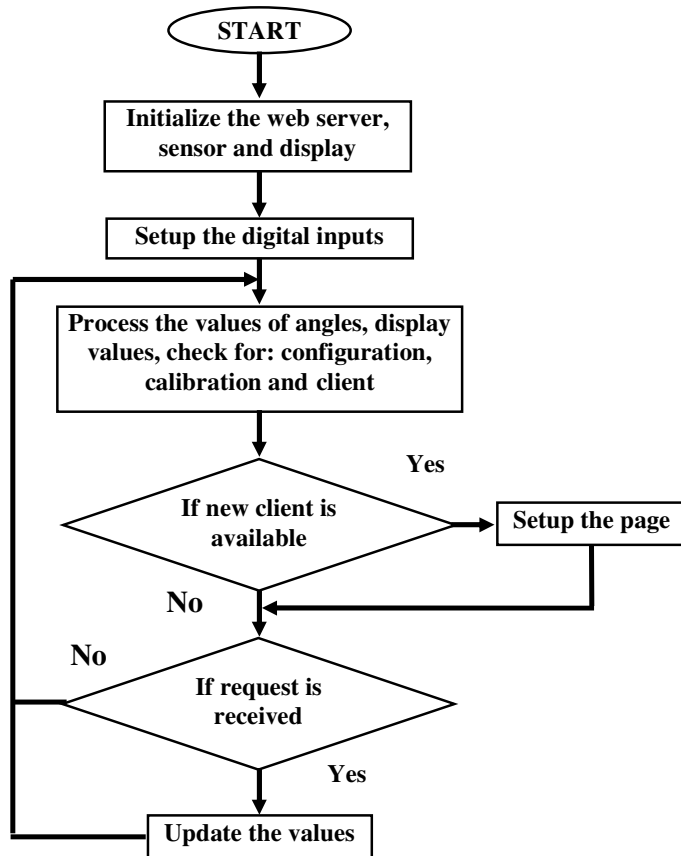


Figure 2. Operation flowchart.

As the flowchart shows the device operation is based on getting the values of angles, their processing and displaying, checking the request of operator for calibration or new configuration, and checking for new wireless network client or requests from the older. The sensor and display are linked on I²C bus, thus at initialization their serial addresses have to be firstly verified. Also, in the initialization stage the digital inputs of the two buttons must be defined.

The most important aspects considered in the hardware design were the position and the fixing of the sensor, the development board placing and the required quantity of energy for long operation. After tests, the continuous operation period was determined as exceeding five days. Power supply is ensured by two series rechargeable batteries, each one having the voltage range between 3.6 and 4.2 V.

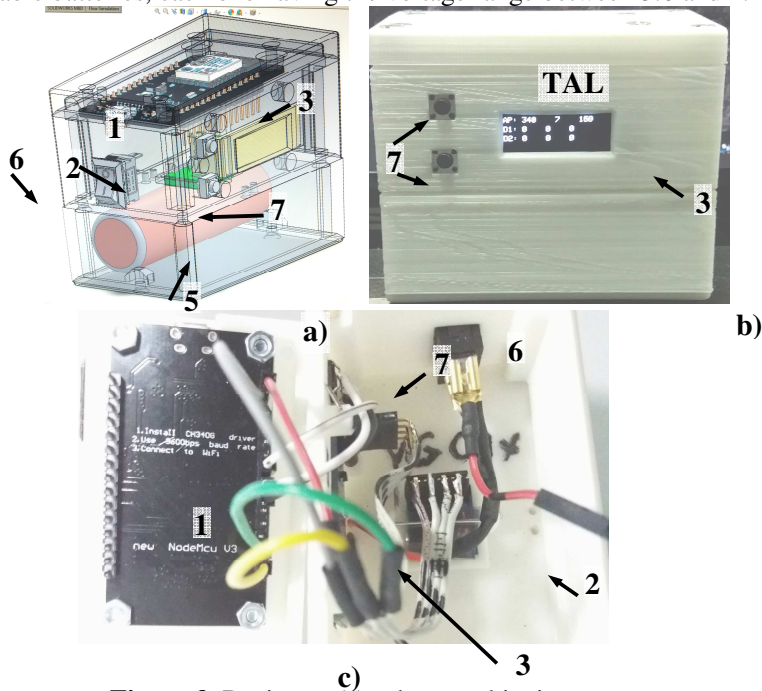


Figure 3. Design and hardware achieving aspects.

The design and hardware dimension were established in SolidWorks, resulting in the assembly form figure 3 a. Device box was presented in a transparent form in order to easily identify the place of the components. In this way, the design was completed by a power switch (6), two push buttons for manual operation (7), up button is assumed as Menu and down button as Select, and batteries (5).

Figure 3.b. depicts the final hardware version, the device box components were accomplished by 3D printing from PLA filament. How the internal connections have been made it is presented in figure 3.c.

Connection diagram is not presented because it is simple. It consists of connecting the SCL and SDA pins, from sensor and display, to the D1 and D2 pins of the development board. Buttons Menu and Select are connected to D14 and D12, acting when they are linked to ground, because the corresponding digital pins are set as digital inputs in pull up mode in the source code.

The most significant aspects in the source code achieving are presented below:

1. ESP8266 acts as Web Server, it can also set up a network by itself. This is possible because it can operate in three different modes: station, soft access point, and both in the same time. This last feature provides the possibility to build mesh networks. As station (STA), it gets IP and can setup a web server, delivering web pages to all connected devices. As soft access point (soft-AP), it creates its own network and acts as a hub for maximum five stations. It does not have interface to a wired network, the mode of operation is called soft access point [6].

2. In the main loop, as server it must handle clients, but more than that it has to do, in this case, the processing of the incoming angular values, displaying the results, and also checking the requests for calibration or configuration;

3. A good way to get some more efficiency is to optimize the number of page charging to the clients. Thus, the auto page refresh can be changed to dynamically load data by AJAX (Asynchronous Javascript And Xml) method. The method provides requesting data from the server asynchronously, in the background, without refreshing the page.

The device was set as soft-AP, so the network is created by itself, at first connection to a new client it sets the browser page and afterwards just updates the values by AJAX method.

AJAX uses a combination of a browser built-in XMLHttpRequest object, to request data from a web server, and JavaScript and HTML DOM, to display or use the data. AJAX applications might use XML to transport data, and they can transport data as plain text or JSON text. AJAX allows web pages to be updated asynchronously by exchanging data with a web server by updating parts of a web page, without reloading the entire page.

3. Testing the device

The testing of the device showed important aspects about how the network made by the device working as soft-AP and the other mobile devices as clients is working. In this regard, it must be specified the device can work stand alone, without any network, but waits for clients. At the connection moment a small gap was observed in the running of the device, it slows down. This gap increased if more

external devices were trying to connect at the same time. AJAX method drastically improved the running speed after connecting to an external device has been performed. Auto refreshing the page of the client produced slow running of the entire network, the number of updates decreasing to less than 10 per second, for three external device connected. AJAX increased this number to more than 20 per second, for five connected devices.

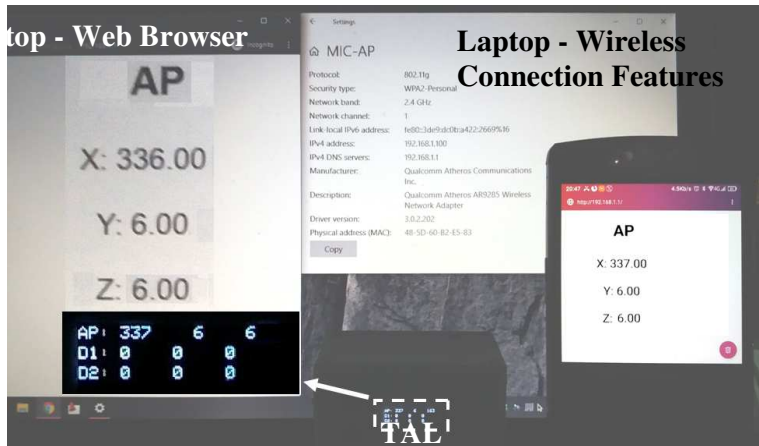


Figure 4. Device testing.

Figure 4 presents one of the testing way, which consisted of the device (TAL), laptop and mobile phone. The small difference between TAL's values and laptop is due to transmitting delay and vibration induced by the laptop to the TAL.

4. Conclusion

TAL device designed and presented can be useful in the supervising process to avoid the perturbation of the worker during operation, and helps when the working place is harder to reach by others.

As running frequency, TAL can transmit more than 20 updated values for each angle, being connected to maximum five PADs.

This device offers data protection and friendly connection to any PAD, and further it can easily be enriched with low-cost component to store data on memory stick, as .csv files.

Acknowledgement. The work has been funded by the Operational Programme Human Capital of the Ministry of European Funds through the Financial Agreement 51675/09.07.2019, SMIS code 125125.

References

- [1] Pasika S., Gandla S.T., Smart water quality monitoring system with cost-effective using IoT, *Heliyon - Contents lists available at ScienceDirect*, (<https://doi.org/10.1016/j.heliyon.2020.e04096>), 2020.
- [2] Tasonga A.C.,Abaob R.P., *Design and Development of an IoT Application with Visual Analytics for Water Consumption Monitoring*, 4th International Conference on Computer Science and Computational Intelligence, Sept. 12–13, 2019, pp. 205-213.
- [3] Scholl M.V., Rocha C.R., Embedded SCADA for Small Applications, *IFAC - PapersOnLine*, 49(21), 2016, pp. 246-253.
- [4] Pratapa Rajua M., Jaya Laxmib A., IOT based Online Load Forecasting using Machine Learning Algorithms, *Procedia Computer Science*, 171, 2020, pp. 551-560.
- [5] Al Dahoud A., Fezari M., NodeMCU V3 For Fast IoT Application Development, <https://www.researchgate.net/publication/328265730> (downloaded at August 21st, 2020).
- [6] ESP32 Web Server with Arduino IDE, <https://randomnerdtutorials.com/download/> (downloaded at August 22nd, 2020).
- [7] The World's Largest Web Developer Site, XML AJAX, https://www.w3schools.com/xml/ajax_intro.asp (downloaded at August 23rd, 2020).

Addresses:

- PhD. Stud. Eng. Cristian-Mircea Muscai, Babeş-Bolyai University, Faculty of Engineering, Piaţa Traian Vuia, nr. 1-4, 320085, Reşiţa, c.muscai@uem.ro
- Lect. PhD. Eng. Ion-Cornel Mituletu, Babeş-Bolyai University, Faculty of Engineering, Piaţa Traian Vuia, nr. 1-4, 320085, Reşiţa, mituic@yahoo.com
(* corresponding author)
- Assoc. Prof. Dr. Habil. Eng. Tiberiu Manescu, “Aurel Vlaicu” University of Arad, B-dul Revoluţiei, nr. 77, 310130, Arad, manescu.tiberiu@gmail.com
- PhD. Stud. Eng. Liviu-Bogdan Protea, Babeş-Bolyai University, Faculty of Engineering, Piaţa Traian Vuia, nr. 1-4, 320085, Reşiţa, b.protea@uem.ro