Design of an Internet of Things — based multi-channel temperature monitoring system

B. Clerbaux, D. Gómez de Gracia, ∗ P. Petitjean and Y. Yang

Inter-University Institute for High Energies (IIHE VUB-ULB), VUB Campus Building G, on the first 2 floors, Pleinlaan 2, 1050 Brussels, Belgium

E-mail: dgomezde83@gmail.com

ABSTRACT: In the scope of the Jiangmen Underground Neutrino Observatory (JUNO) project, 6 back-end card (BEC) mezzanines connected to one BEC base board are in charge of compensating the attenuated incoming data from 48 front-end channels over 48 100-meter-long Ethernet cables. Each of the mezzanines has 16 equalizers that may be subject to overheating. It is important therefore to monitor their temperature in real time. However, collecting data from a relatively large (1080) number of mezzanines is not a trivial task. In this work we propose a solution based on Wi-Fi mesh. Both the technical details and the first test results performed are reported.

KEYWORDS: Control and monitor systems online; Front-end electronics for detector readout; Modular electronics

∗Corresponding author.
1 Introduction

The Jiangmen Underground Neutrino Observatory (JUNO [1]) is a multipurpose neutrino detector. Its main goal is to determine the neutrino mass hierarchy and to measure some of the neutrino oscillation parameters with unprecedented precision (<1%), using as a source electron anti-neutrinos coming from the Yangjiang and Taishan Nuclear Power Plants. JUNO is located at 700 m underground and its central detector consists of 20 ktons of liquid scintillator contained in a 35 m diameter acrylic sphere that will capture neutrino interactions from different sources. Among the numerous electronic components of the detector, a back-end card (BEC) is used as a concentrator to collect trigger requests. Each of the incoming trigger request signals passes through an equalizer placed on a Mezzanine, mounted on a BEC to compensate for the attenuation due to the long cables. There are 180 BECs, and a total of 17280 equalizers. Measuring the temperatures of the multiple equalizers placed on all the Mezzanines is necessary to make sure that they perform stably and don’t overheat.

2 JUNO electronics

The JUNO electronics system has two parts: the “wet electronics” system and the “dry electronics” system. Both systems are inter-connected with 100 m Ethernet cables [2]. Figure 1 shows how the components of both systems work together.

The wet electronics of JUNO performs basic tasks of data acquisition. The photomultiplier tubes (PMTs) collect (in groups of three) the light induced by the interaction of neutrinos with the liquid scintillator. This light is transformed into an electrical signal, which is usually called “PMT waveform” in the literature. This signal is digitized inside the global control unit (GCU), and then sent to the BEC through the synchronous link. The dry electronics system controls the trigger requests. The central trigger units (CTUs) compute signals to trigger data acquisition (called
trigger decisions). The BECs transmit the incoming trigger decisions, which arrive through the Trigger Timing Interface Mezzanine (TTIM), towards the GCUs [3]. A box is used to gather some of the elements of the dry electronics system, like the Mezzanines, the BEC and the TTIM. The exact placement of the different components is shown in figure 4.

3 Engineering analysis and development

In order to monitor the temperature of the 17280 equalizers present in the 180 BECs, three TMP100 temperature sensors are placed on the BEC (see figure 2(a)). The equalizers that need to be monitored lie on the mezzanines, which are 8 mm apart from the BEC (see figure 2(b)). This means that the temperature that will be measured by the sensors is the one of the hot air above the equalizers. TMP100s are digital temperature sensors [4] that measure the temperature of the BEC’s silicon on the spot the sensor is placed at. They offer a typical accuracy of ±1 °C without requiring calibration, and a maximum resolution of 0.0625 °C (which is the maximum degree to which a change in temperature can be detected), which are sufficient for our set-up as we look for a measurement precision to the degree level. The 180 BECs have to report the temperatures of their sensors to a central computer at a rate of once per minute at least. In JUNO, the three temperature sensors are read by the Trigger Timing Interface Mezzanine (TTIM) through an inter-integrated circuit (I2C) bus, and reported to the central computer via the White Rabbit Network [5]. In this paper, we explore an alternative way of reading and retrieving the temperatures using a development board and a Wi-Fi network, with the goal of achieving an information loss rate of less than 1%. 

Figure 1. Schematic view of JUNO electronic system.
4 Implementation

An ESP32 pico board [6] is placed on each BEC, and it is connected to the BEC’s $I^2C$ bus as a master device. ESP32s are development boards compatible with $I^2C$ and with Wi-Fi integration. They can communicate with the three sensors, which are connected as $I^2C$ slave devices, and read their registers at the desired rate. Each ESP32 board transmits its values at a frequency of 0.2 Hz, i.e. one message every 5 seconds, to the central computer in the form of messages that clearly separate the three temperature values, as well as the ID of the board where the temperatures were measured. Once they arrive to the central computer, they are logged into a database with the time and date they were received at. The status of all the motherboards is accessible both in real time and at any point in the monitoring timeline.

Figure 3. Wi-Fi mesh network with central computer, an arbitrary number of “Bridge” nodes, and “Mesh” nodes.

In order to achieve this, a Wi-Fi mesh network was designed (see figure 3). This architecture features two kind of nodes that can connect to a single device, and accept at most 10 connections.
from other devices (which is the maximum amount of connections the ESP32 hardware can manage): *bridge nodes* and *mesh nodes*. The former are directly connected to an access point, and the latter connect to other nodes of the network. Previous studies (such as the one described in [7]) show that transmitting messages over this kind of network is reliable and relatively fast (less than 100 ms delay). The architecture was implemented using the PainlessMesh library [8], which automatically builds and maintains the mesh network, while allowing to incorporate new boards on the go, message sending, message broadcasting, and identification of boards.

5 Validation and results

A small scale test was performed in the lab with 27 mesh nodes and 2 bridge nodes in order to study the reliability and performance of the mesh network by measuring the percentage of retrieved messages and the eventual issues with the mesh formation or maintenance. The result showed that for a transmission rate of 0.2 Hz the reception rate was of 99.06% (which means that less than 1% of the sent messages were lost) over a period of 11 days. Experiments were done to assess the reason for the loss of messages. It was found that some of the ESP32s spontaneously disconnected from the mesh and never reconnected back, producing the loss of the messages that should have been relayed since the disconnection. This problem was partially solved by implementing a routine that reconnected the ESP32 to the network in case it had disconnected, and by having two bridge nodes instead of one, but 0.94% of the messages were still lost. Further investigation is needed, particularly on the functioning of PainlessMesh’s engine, to determine the reason for these disconnections.

![Box with electronic components including the BEC, the mezzanines, the ESP32, the TMP100s (IC5 (center), IC3 (near fan 2), IC4 (near fan 1)) and 5 numbered fans.](image)

Figure 4. Box with electronic components including the BEC, the mezzanines, the ESP32, the TMP100s (IC5 (center), IC3 (near fan 2), IC4 (near fan 1)) and 5 numbered fans.

To determine the difference between the measured temperatures and the real temperatures of the equalizers, we placed four thermocouples on the box. Two of them were placed on fans 1 and 2,
to measure the air entering and exiting the box, respectively. The other two were placed on mezzanines 1 and 6 (which are next to fans 1 and 2, respectively, as seen in figure 4), to measure their real temperatures. Figure 5 shows the temperatures read by the thermocouples (left) and by one ESP32 (right) during a period of 2 hours. We found that the equalizers are 12–13 °C warmer than the hot air above them (which is measured by the TMP100s). The temperature at the fans is about 3–5 °C lower than the temperature at the TMP100s.

![Figure 5. Temperature measurement for a period of 2 h. Left: real temperatures measured with a thermocouple at fans 1 and 2, and at mezzanines 1 and 6. Right: temperatures measured by the three TMP100 sensors connected to one ESP32.](image)

6 Conclusion

Given the results presented in this short article, we can conclude that a Wi-Fi mesh architecture of 180 boards can be implemented if a sufficient number of bridge nodes is chosen. This network can reliably transmit to a central computer more than 99% of the messages each board produces if the frequency of transmission is kept at one message every five seconds or more.

Acknowledgments

Funded by FNRS (Fonds de la Recherche Scientifique).

References

[1] Jiangmen Underground Neutrino Observatory, http://juno.ihep.cas.cn/.

[2] B. Clerbaux, S. Hang, P.-A. Petitjean, P. Wang and Y. Yang, Automatic test system of the back-end card for the JUNO experiment, arXiv:2011.06823 (2020).

[3] Y. Yang and B. Clerbaux, Design of a common verification board for different back-end electronics options of the JUNO experiment, arXiv:1806.09698 (2018).

[4] TMP100 data sheet, product information and support|TI.com, https://www.ti.com/product/TMP100#design-development#all.
The White Rabbit Project is a multilaboratory, multicompant and multinational collaboration to develop new technology that provides a versatile solution for control and data acquisition systems, https://white-rabbit.web.cern.ch/.

ESP32-PICO-KIT V4/V4.1 getting started guide — ESP32 — ESP-IDF programming guide latest documentation, https://docs.espressif.com/projects/esp-idf/en/latest/esp32/hw-reference/esp32/get-started-pico-kit.html.

Y. Chia, R.H. Arjadi, E. Setyaningsih, P. Wibowo and M.I. Sudrajat, Performance evaluation of ESP8266 mesh networks, J. Phys.: Conf. Ser. 1230 (2019) 012023.

A painless way to setup a mesh with ESP8266 and ESP32 devices, https://gitlab.com/painlessMesh/painlessMesh/-/wikis/home.