XV International Conference on Durability of Building Materials and Components
DBMC 2020, Barcelona
C. Serrat, J.R. Casas and V. Gibert (Eds)

A Comprehensive Description of a Low-Cost Wireless Dynamic Real-Time Data Acquisition and Monitoring System

Seyedmilad Komarizadehasl[1], Behnam Mobaraki[2], Jose A. Lozano-Galant[2], and Jose Turmo[1]

[1]Department of Civil and Environment Engineering, Universitat Politècnica de Catalunya, BarcelonaTech. C/ Jordi Girona 1-3, 08034, Barcelona, Spain. Seyedmilad.komarizadehasl@upc.edu, jose.turmo@upc.edu

[2]Department of Civil Engineering, Univesidad de Castilla-La Mancha. Av. Camilo Jose Cela s/n, Ciudad Real, 13071, Spain. behnam.mobaraki@uclm.es, Joseantonio.lozano@uclm.es

Abstract. *Nowadays, low-cost sensors and acquisition devices have been emerging as an obvious solution to many innovative applications such as Structural Health Monitoring (SHM) systems. In this paper, issues regarding dynamic data acquisition, as well as their respective solutions, are presented. Moreover, a comprehensive description through-out an inexpensive sensor network system using open-source hardware for a real-time acceleration data acquisition has been presented. The platform consists of an accelerometer, an Arduino board, and a computer as a data recorder and presenter. Data is recorded through an efficient microcontroller code that can provide a considerable reading frequency (up to 300Hz). Using Python, instant acceleration recording for all the three axes has been done. It is shown how the performance of the proposed system is efficient for the field of SHM systems.*

Keywords: *Arduino Due, Structural Health Monitoring (SHM), Mpu9250, Internet Of Things (IoT).*

1 Introduction

While low-cost sensors are getting significant attention from scientists, how to use them correctly is another matter. Here, by using an Arduino Due and an accelerometer and a raspberry pi, a wireless application has been presented. The nobility of this application would be the chance to observe data capturing in real-time from any place with any device. Firstly here, an introduction through Arduino due, raspberry pi, MPU9250 (the accelerometer) has been presented. Secondly, a brief presentation regarding the coding of the Arduino and the raspberry pi has been given. Finally, the methodology and the result of this application has been shown.

2 State of the Art

In this section, a brief review of the used sensors and a microcontroller in the project has been given together with their technical descriptions.

2.1 Arduino Due

Arduino is an open-source electronics platform based on easy-to-use hardware and software. The Arduino Due (Fig.1) is a microcontroller board based on the Atmel SAM3X8E ARM

Cortex-M3 CPU. It is the first Arduino board based on a 32-bit ARM core microcontroller. It has 54 digital input/output pins (of which 12 can be used as PWM outputs), 12 analog inputs, 4 UARTs (hardware serial ports), an 84 MHz clock, an USB OTG capable connection, 2 DAC (digital to analog), 2 TWI, a power jack, an SPI header, a JTAG header, a reset button and an erase button. Unlike most Arduino boards, the Arduino Due board runs at 3.3V. The maximum voltage that the I/O pins can tolerate is 3.3V. Applying voltages higher than 3.3V to any I/O pin could damage the board (Blum, 2013).

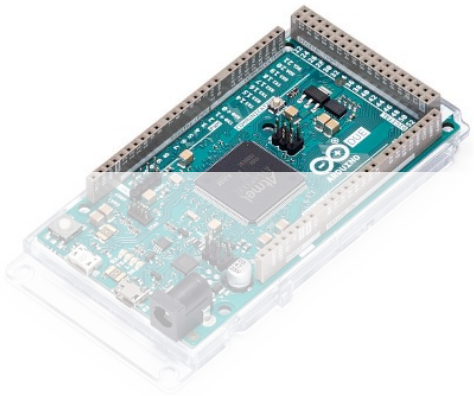

Figure 1. Schematic of an Arduino Due.

2.2 MPU9250

The MPU-9250 (Fig.2), delivered in a 3x3x1mm QFN package, is the world's smallest 9-axis MotionTracking device and incorporates the latest InvenSense design innovations, enabling dramatically reduced chip size and power consumption, while at the same time improving performance and cost. The MPU-9250 MotionTracking device sets a new benchmark for 9-axis performance with a power consumption of only 9.3µA and a size that is 44% smaller than the company's first-generation device. Gyro noise performance is three times better, and full-scale compass range is over four times better than competitive offerings. The MPU-9250 is a System in Package (SiP) that combines two chips: the MPU-6500, which contains a 3-axis gyroscope, a 3-axis accelerometer, and an onboard Digital Motion Processor™ (DMP™) capable of processing complex MotionFusion algorithms; and the AK8963, the market-leading 3-axis digital compass. Improvements include supporting the accelerometer low power mode with as little as 6.4µA of, and it provides improved compass data resolution of 16-bits (0.15 µT per LSB). (InvenSense, 2014).

Figure2. Schematic of MPU9250 sensor.

3 Communication Ways

While many sensors use digital and Analog ports to provide the microcontroller the information which they have measured, some sensors use the inter-integrated circuit (I2C) protocol. This is a protocol that allows multiple “slave” digital integrated circuits (Sensors) to communicate with one or more “master” chips (Arduino). Like the Serial Peripheral Interface (SPI), which is only intended for short-distance communications within a single device(Ozdagli et al., 2018). The accelerometer sensor had to be connected to the I2C port (SCL, SDA) on the board. The code was written on the Arduino platform and uploaded to the board via a USB cable. For getting the main characteristics of these sensors, few tests have been done (Mobaraki et al., 2020) (Mobaraki and Vaghefi, 2016).

4 Methodology

After installing the last version of Python on the Computer. For communication of the computer with the Arduino, a library entitled pyserial had to be installed. With this done, the Arduino port could have been introduced to Python to show the captured data. With a python code, the received data were given their time of capture. For being able to use the exact time from the internet another library had to be added and used. The name of that library was DateTime.The given time to the received data had a resolution of 1 microsecond. For illustrating the data wirelessly, the computer had to be connected to the internet. The only power supply that this application needed was hocking up the computer. The Arduino gets its power from the USB port of the computer and initiates the accelerometer. In order to make this project wireless, another free application had to be installed on this computer (Komarizadehasl et al., 2020a) (Komarizadehasl et al., 2020b). The VNC is a remote application that lets users control their devices over the internet. By having the VCN application on any other internet-connected device, the functionality of this device would be observable. The written python code would save the data at the same time as it is illustrating them.

5 Experiment

For testing this sensor and its reliability, an experiment has been implemented. With a dynamic jack, a sinus signal has been programmed, and the vibrations had been saved by the accelerometer. In Fig.3, a picture of the jack can be seen. This jack can shake its bottom plate as was programmed. The instructions to the hydraulic jack were to make a wave with a fixed frequency of 5 hertz (5 complete waves in one second). The movement of the jack was to go up to 0.1 millimeters up and -0.1 millimeter down from its null axis to make a sinus wave. With a very simple two time differential, the acceleration equation could be calculated.

$$y = d * \sin(\omega * t + \varphi) \quad (1)$$

$$\omega = 2 * \pi * f \quad (2)$$

In the above equation, **y** is the displacement in time **t**, d is the maximum allowed movement of the jack in each cycle, $\boldsymbol{\omega}$ is the angular frequency, and $\boldsymbol{f}$ is the set frequency, which equals to 5Hz, and $\boldsymbol{\varphi}$ is the phase constant. On the equ.3, acceleration has been calculated from the Eq.1. This was done by getting the second-order derivative of the Eq.1. By putting all the data

in the Eq.3, the maximum acceleration was calculated as 10.4352 g*10^{-3} m/s^2.

$$a = \frac{d^2 * y}{dt^2} = \ddot{y} = -d * \omega^2 * \sin(\omega * t + \varphi) \quad (3)$$

Figure 3. The hydraulic jack subjected to the experiment.

6 Analysis

The very first faced problem in this experiment was that the sensor could not record data or, if recorded, the data were messy. It was deducted that the sensor had to be glued to the bottom plate of the jack for getting accurate information. The second problem was that the written python code could save only 120 data per second while the sensor was reporting more than 300 data per second. Although, by using serial port commercial software on the computer, it could have been possible to save data at the same speed as their production. Since here, getting the accurate time of capture was vital, it was obligatory to use Python to attach the provided data with their corresponding time. To tackle this problem, the speed of data capture had to be dialed down, so the Python could get and save them. To be on the safe side, the speed of capture had been set on 84Hz.

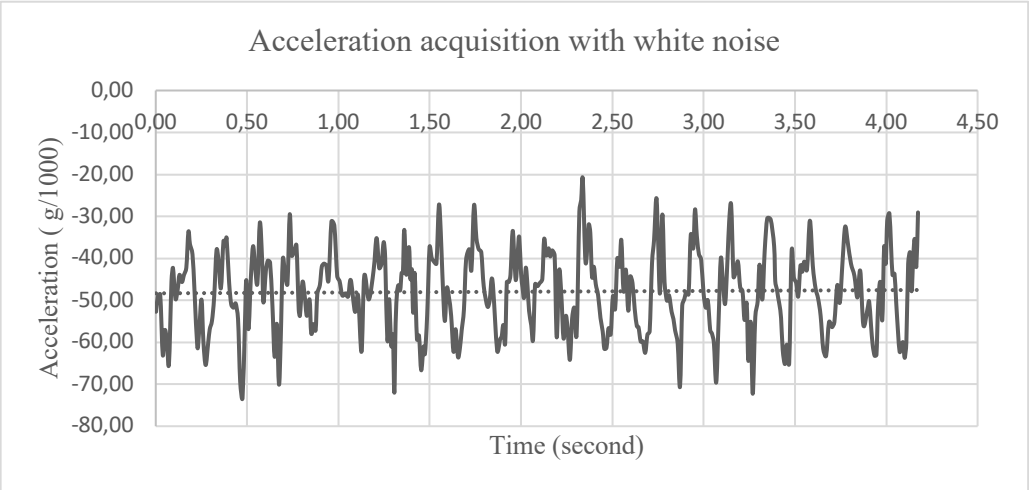

Figure 4. Acceleration, Time diagram with white noise.

The other unexpected issue that this project faced was that, though this sensor had been calibrated in the company, it had a constant number added to all provided data, which from nowhere it would be named as the white noise. As in the Fig.4, it has been illustrated, and the averaged data is around -50milli-g while they had to fluctuate around zero. This -50milli-g was considered as the white noise of this sensor. In order to measure this correctly, the average of 10000 sets of data in a vibration-free test has been calculated. For this sensor, the white noise had been calculated as -49.8535 milli-g.

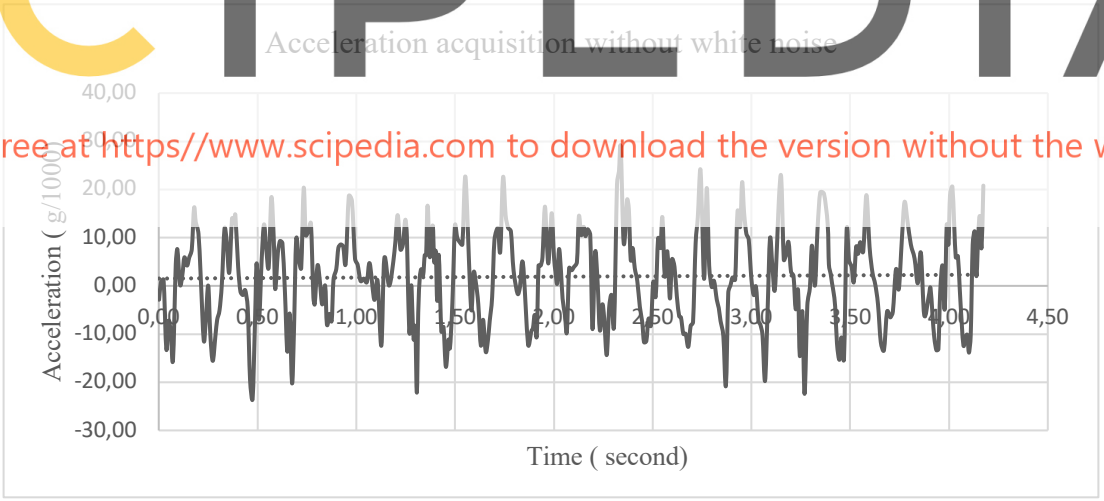

Figure 5. Acceleration, Time diagram without the white noise.

By removing this amount from the provided accelerations, the values had been pulled up where they needed to be. The data were clearer and more understandable when this white noise had been removed (Fig.5). Different accelerometer boards had been tested, and it was concluded that each circuit has its own different amount of white noise that must be dealt with. It should have been assured that this white noise is constant. Moreover, the circumstances and time

cannot change it. For this reason, the jack was programmed with many more different frequencies and displacements. In all of those experiments, the white noise stayed the same.

7 Conclusions

As in Fig.5, it is visible the sinus wave conducted from the accelerometer is quite close to the expected behavior. As it is observable, the sinus wave is fluctuating about 10.5 milli-g from its average as it was calculated in the last section, the graph should have had a 10.453 milli-g fluctuation. It is quite notable to see that they have worked out almost the same. In Fig.5, the filtered data from the 5Hz experiment has been shown. As can be seen, the result is not so accurate, for the fluctuation has other unexpected data or noises as well. In the future works, filters must be applied to delete the unwanted data and ambient noises that may have entered into this experiment unwantedly.

Acknowledgments

The authors are indebted to the Spanish Ministry of Economy and Competitiveness for the funding provided through the research project BIA2017-86811-C2-1-R directed by José Turmo and BIA2017-86811-C2-2-R, directed by Jose Antonio Lozano-Galant. All these projects are funded with FEDER funds. Authors are also indebted to the Secretaria d' Universitats i Recerca de la Generalitat de Catalunya for the funding provided through Agaur (2017 SGR 1481). It is also to be noted that funding for this research has been provided for MR. SEYEDMILAD KOMARIZADEHASL by Agencia Estatal de Investigación del Ministerio de Ciencia Innovación y Universidades grant and the Fondo Social Europeo grant (PRE2018-083238).

ORCID

Seyedmilad Komarizadehasl: https://orcid.org/0000-0002-9010-2611
Behnam Mobaraki: https://orcid.org/0000-0002-2924-643X
Jose Antonio Lozano Galant: http://orcid.org/0000-0003-0741-0566
Jose Turmo: https://orcid.org/0000-0001-5001-2438

References

Blum, J. (2013). Exploring Arduino: tools and techniques for engineering wizardry.

InvenSense, T., 2014. MPU-9250, Nine-Axis (Gyro+ Accelerometer+ Compass) MEMS MotionTracking™ Device.

Komarizadehasl, S., Mobaraki, B., Lozano-Galant, J.A., Turmo, J. (2020)a. Detailed evaluation of low-cost ranging sensors for structural health monitoring applications, in: International Conference of Recent Trends in Geotechnical and Geo-Environmental Engineering and Education. "RTCEE/RTGEE 2020, 8–12.

Komarizadehasl, S., Mobaraki, B., Lozano-Galant, J.A., Turmo, J., (2020)b. Evaluation of low-cost angular measuring sensors, in: International Conference of Recent Trends in Geotechnical and Geo-Environmental Engineering and Education. "RTCEE/RTGEE 2020, 17–21.

Mobaraki, B., Komarizadehasl, S., Castilla-Pascual, F.J., Lozano-Galant, J.A. (2020). Determination of Enviromental Parameters Based onArduino Based Low-Cost Sensors, in: International Conference of Recent Trends in Geotechnical and Geo-Environmental Engineering and Education. "RTCEE/RTGEE 2020.

Mobaraki, B., Vaghefi, M. (2016). Effect of the Soil Type on the Dynamic Response of a Tunnel under Surface Detonation. Combust. Explos. *Shock Waves 52,* 119–127. https://doi.org/10.1134/S0010508216030175

Ozdagli, A.I., Liu, B., and Moreu, F. (2018). Low-cost, efficient wireless intelligent sensors (LEWIS) measuring real-time reference-free dynamic displacements. Mech. Syst. *Signal Process. 107*, 343–356. https://doi.org/10.1016/j.ymssp.2018.01.034