EVALUATION OF LOW-COST ANGULAR MEASURING SENSORS

Seyedmilad Komarizadehasl*[1], Behnam Mobaraki[2], Jose Antonio 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.
[2]Department of Civil Engineering, Univesidad de Castilla-La Mancha. Av. Camilo Jose Cela s/n, Ciudad Real, 13071, Spain.

ABSTRACT

One of the main issues for performing Structural Health Monitoring (SHM) is the high cost of metering devices. In order to make it applicable to the conventional structures with low defined budgets, low-cost sensors have been widely utilized. In this paper, the characteristics of a low-cost circuit (MPU9250) with low power consumption for measuring angles are studied. This circuit is composed of an accelerometer, a gyroscope, and a magnetometer. There are two ways of coding and using this sensor for angular measurements. In the first application, the accelerometer and the gyroscope of the circuit are only used to get angle around X and Y-axis. In the second application, the gyroscope is going to be added to the other two sensors in order to get angular measurements of all axis. The data accuracy plus the advantages and disadvantages of the response of this circuit regarding each code has been studied in this paper by using the codded sensor in some experiments. Although the second application showed less error from the expected results, it was less stable than the first application.

Keywords: Internet of the Things, Arduino Due, Low-cost sensors, Structural Health Monitoring, Angular measurement

1 INTRODUCTION

Although the low-cost sensors are becoming more and more popular, the correct way to use their features and potentialities is another matter (Sumitro and Wang, 2005). With the technology progress, new low-cost sensors are being introduced every day for different applications and various needs. Each one of these has advantages as well as disadvantages in their aimed application, which should be dealt with appropriately (Aguero et al., 2019). Using Arduino instead of other microcontrollers for controlling these sensors is getting much attention. Firstly, it is widely available at an affordable price. Secondly, there are many online tutorials for learning its software and hardware. These features have made this microcontroller very interesting (Bńlský et al., n.d.). In this investigation, 2 different ways of using the MPU9250 circuit have been introduced. The first one can work in any place but only can get angular measurements around the X and Y-axis. The second one can get angular measurements in X, Y, and Z but only can work in environments where there is no big metallic field. It has to be mentioned that the available 2 directional sensors in the market have a price of 179$ (“2-Axis Evaluation Kit,” n.d.) while the proposed MPU9250 costs about 8$. For doing these tests, a tool has been designed by the Autodesk Fusion 360, and it has been printed by a 3D printer. This tool was to place the sensor on different angles for reading correctly and comparing the 2 applications.

In this paper, firstly, an introduction through Arduino due, and the used sensors have been given. Secondly, the applications, as well as their essential part of their codes, have been illustrated. In the end, the conclusion through performed experiments has been given.

2 STATE OF THE ART

In this section, a brief review of the used sensors and the used microcontroller in this project have 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, which a schematic of it can be seen in Figure 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 (Ozdagli et al., 2019). 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 Mpu 9250

The MPU-9250 circuit (Figure 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 3 times better, and full-scale compass range is over 4 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).

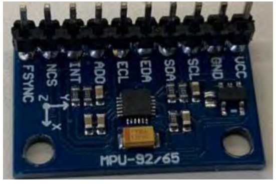

Figure2. Schematic of MPU9250 sensor

2.3 Communication ways

While many sensors use digital or 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 (Serial Clock Line (SCL), Serial Data Line (SDA)) on the board. The codes were to be 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.

3 METHODOLOGY

The way of connecting the mpu9250 to the Arduino on both applications is precisely the same. The only difference between these applications is the uploaded codes into the Arduino platform. In this paper, 2 applications have been studied for the same introduced equipment. These applications are to investigate the importance of magnetometry in the equations and their disadvantages. The first application is the easy one in which data of magnetometry are not included. The second includes those data as well.

Connecting the MPU9250 circuit is quite accessible since it is working with the I2C communication port. The pins of SCL and SDA should be connected to their representatives on Arduino due (SCL and SDA). The VCC to the 3.3V port and the GND to the GND of the Arduino. The data that MPU9250 provides are acceleration, angular speed, and the speed of a change in the magnetic field. For using this device as an angular quantification, on the captured data, some mathematical work should have been performed. Firstly, the generated data by the MPU has been received by the Arduino. Secondly, Arduino has done some mathematical operations on the raw data to provide usable outputs.

Since the chosen sensitivity was +-2g for accelerometer and 250 degrees per second range for gyroscope, the provided raw data from the sensor had to be divided based on the given data of the datasheet. These sensitivities are the basic ones. As it has been declared in the datasheet, this way provided data have the least amount of noise.

3.1 First application

In the first application, only the data of accelerometer and the gyroscope chipset of the MPU9250 is used. In Table 1 and Table 2, the FS_SEL refers to the corresponding library code for activating each of the sensitivities. LSB means the Least significant bit. On the first table. It is per degree-second, and on table 2, it is based on per g. The reported data by the sensor has to be divided by these converters in order to illustrate acceleration data per 9.81 meter per second squared (per g) and gyro speed in degree-second.

The generated raw values for the X and Y axis have been translated to angular speed based on the datasheet instruction in table 1. In equations 1 and 2, the transformation equation could be seen. As can be figured out, rawGyroX and rawGyroY are data that the sensor had provided. The gyroX and gyroY are being calculated by Arduino to change their unit to degree-second.

Table 1, Gyroscope sensitivity

FS_SEL	Full-scale range	LSB sensitivity
0	+- 250 °/s	131 LSB/°/s
1	+- 500 °/s	65.5 LSB/°/s
2	+- 1000 °/s	32.8 LSB/°/s
3	+- 2000 °/s	16.4 LSB/°/s

$$gyroX = \frac{(rawGyroX)}{65.5} \tag{1}$$

$$gyroY = \frac{(rawGyroY)}{65.5} \tag{2}$$

For getting the angular measurement, the interval time had to be multiplied to the above equations. In a simple word with equations 3 and 4, the angle value has been calculated by taking the integral of the angular speed. On the blow equations, angleGyroX and angleGyroY are the integrated ones from gyroX and gyroY, respectively, with the unit of degree.

$$angleGyroX += gyroX * interval \tag{3}$$

$$angleGyroX += gyroX * interval \tag{4}$$

It is also possible to get angle value from acceleration data. Firstly, like the gyroscope values, the acceleration data must be treated as instructed in the datasheet in Table 2. The acceleration values of all the X, Y, and Z-axis has been calculated by equations 5, 6, and 7. On these mentioned equations, rawAcc data had been provided by the MPU9250, and accX has been converted by Arduino microcontroller for having usable data per g.

Table 2, acceleration sensitivity

AFS_SEL	Full-scale range	LSB sensitivity
0	+-2g	16384 LSB/g
1	+-4g	8192 LSB/g
2	+-8g	4096 LSB/g
3	+-16g	2048 LSB/g

$$accX = \frac{(rawAccX)}{16384.0} \tag{5}$$

$$accY = \frac{(rawAccY)}{16384.0} \tag{6}$$

$$accZ = \frac{(rawAccZ)}{16384.0} \tag{7}$$

For accurate measurements of tilt in the X and Y planes, the received acceleration on all of the 3 axes would be need. The reference position is taken as the typical orientation of a device, with the x- and y-axes in the plane of the horizon (0 g field) and the z-axis orthogonal to the horizon (1 g field). On the equations 8 and 9, formulas for obtaining angle from recorded vibrations have been shown.

$$angleaccX = 360 * 2 * \pi * tan\,\hat{}(-1)\frac{accY}{(accZ + abs(accX))\rrbracket} \tag{8}$$

$$angleaccY = 360 * 2 * \pi * \tan\hat{}(-1) \frac{accX}{(accZ+abs(accY))〗} \quad (9)$$

3.2 Second application

On the second application, the third part of the circuit MPU9250 has been used to overcome the problem of not being able to measure the Z plane, which was observed on the first application.

Using a magnetometer chipset in order to use the magnetic field of the earth as a reference for tilting around the Z-axis, had resolved the mentioned problem. The reason why it was not possible to get the angle around the Z-axis on the first application was the fact that there was no reference for it. On the other hand, X and Y axis on the first application was using the gravity of the earth as their reference point; consequently, they were working smoothly, correct and through time their data remained untacked respectively.

Since the raw data provided from this method contains a lot of noises a filter on the out-put of the sensor has been set to convert the data to quaternions (Brown and Hwang, 1992). To describe it simply this converting will use data of gyroscope, accelerometer and the magnetometer for returning quaternions values. In mathematics, the quaternions are a number system that extends the complex numbers [5]. For extracting the yaw pitch and roll, these quaternions values (q[0], q[1], q[2], q[3]) will be converted to angles with the unit of degree. The Yaw would be angle around Z axis, pitch is around X axis and roll is around Y axis.

```
yaw   = atan2(2.0f * (q[1] * q[2] + q[0] * q[3]), q[0] * q[
0] + q[1] * q[1] – q[2] * q[2] – q[3] * q[3]);

pitch = -asin(2.0f * (q[1] * q[3] – q[0] * q[2]));

roll = atan2(2.0f * (q[0] * q[1] + q[2] * q[3]), q[0] * q[0]
– q[1] * q[1] – q[2] * q[2] + q[3] * q[3]);

pitch *= 180.0f / PI;

yaw   *= 180.0f / PI;

yaw   -= 1.087; /* Declination at Barcelona, Spain Mode
l Used:    IGRF12

Latitude:    41.3876° N

Longitude:    2.1107° E

Date    Declination

2020-01-16    1.087° East changing by 0.121° E per year
(- for East)*/

roll *= 180.0f / PI;
```

Figure 3, calibrating the magnetometer and transformation from quaternions data.

This application depends on the location of the test. The magnetometer is required to compensate for magnetic declination. As a result, the variable Yaw needs to be calibrated. By going to this website (“magnetic declination,” n.d.) and entering the longitude and latitude, the magnetic declination, as well as its annual change ratio, could be obtained. In Figure 3, the implantation of the calibration had been illustrated.

4 EXPERIMENT

For examining these applications, a tool has been created by a 3D printer. With the help of this tool, the sensor could be checked on 90-degree changing positions. The aim was to put the sensor on different locations of this tool and check the measured data. The tool and the positioning of the sensor have been shown in Figure 5.

For comparing these applications, first, the sensor was uploaded with the first code from the first explained application. After that, it has been set on all the locations and angles were saved. Secondly, the same sensor has been uploaded with the second code from the second explained application, which responded to the second application. After calibrating the magnetometer of the second application, the sensor again has been set in all the different locations, and measured data had been saved. The downside of this Application is that around metallic ambient, it loses its efficiency. Moreover, calibrating and preparing the sensor for measuring is harder than the first application. It is to be told for each time of setting up the sensor calibration is needed. The sensor has to be moved for about a minute on a route like a Figure 4.

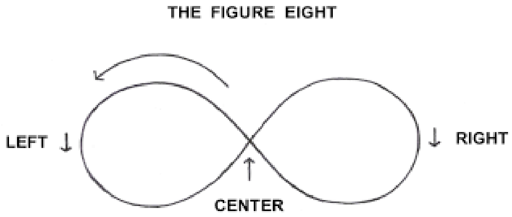

Figure 4. Sensor calibration movement route

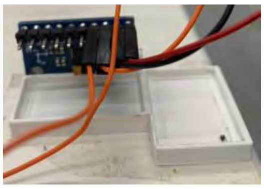

Figure 5. Positioning of the sensor (z=0, z=90, y=90, x=90)

In Table 3, the results of the first application can be seen. In this Table, X, Y, and Z are the illustrated data by the applications. Directions of the tool regard the positioning of the sensor on the tool (Figure 5). As it can be deducted, the first application had about 8.33 degree of error on Y axes and 10.59 degree On X axes.

Table 3. Result of the first application (degrees)

The direction of the tool	X	Y	Z
Z=X=Y=0	81.34 °	81.19 °	21.45 °
Z=90	81.34 °	81.19 °	65.97 °
Y=90	0.24 °	179.52 °	106.7 °
X=90	1.93 °	2.10 °	248 °

In Table 4, the result of the second application has been shown. On this table, the 90-degree change of positioning on each plane had errors as 3.1 degrees on X-axes, 0.28 degrees for Y axes, and 4.99 degrees for Z axes.

Table 4. Result of second application (degrees)

The direction of the tool	X	Y	Z
Z=X=Y=0	7.23 °	-2.58 °	-108.0 °
Z=90	-0.12	3.65	-13.01 °
Y=90	78.92	87.14	-113.3 °
X=90	94.13	-1	-117.96 °

5 CONCLUSIONS

As has been mentioned, the first application cannot measure angles on the Z-axis. It has been observed that throughout time, the value of Z-direction increases. Moreover, the first application is showing about 10 degrees of an error on the Y-axis and X-axis.

The second application, which every time needed the magnetometer calibration. Although it can be understood that this application had less than 5 degrees of error for each direction, lunching it was not as easy as the first application.

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).

REFERENCES

2-Axis Evaluation Kit [WWW Document], n.d. URL https://www.bendlabs.com/products/2-axis-evaluation-kit/

Aguero, M., Ozdagli, A., Moreu, F., 2019. Measuring Reference-Free Total Displacements of Piles and Columns Using Low-Cost, Battery-Powered, Efficient Wireless Intelligent Sensors (LEWIS2). mdpi.com. https://doi.org/10.3390/s19071549

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

Bńlský, M., Kozlok, M., of, R.B.-2018 V.L.C., 2018, undefined, n.d. Application of Arduino Platform for Light Field Analysis. ieeexplore.ieee.org.

Brown, R., Hwang, P., 1992. Introduction to random signals and applied Kalman filtering.

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

magnetic declination [WWW Document], n.d. URL http://geomag.nrcan.gc.ca/calc/mdcal-en.php

MPU6050 data-sheet [WWW Document], n.d. URL https://www.invensense.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf%0A

Ozdagli, A., Liu, B., Moreu, F., 2019. Real-time low-cost wireless reference-free displacement sensing of railroad bridges, in: Conference Proceedings of the Society for Experimental Mechanics Series. Springer New York LLC, pp. 103–109. https://doi.org/10.1007/978-3-319-74642-5_12

Ozdagli, A.I., Liu, B., 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

Sumitro, S., Wang, M.L., 2005. Sustainable structural health monitoring system, in: Structural Control and Health Monitoring. John Wiley & Sons, Ltd, pp. 445–467. https://doi.org/10.1002/stc.79