USE OF AN IOT TECHNOLOGY TO ANALYSE THE INERTIAL MEASUREMENT OF SMART PING-PONG PADDLE

Rajeev Kanth[1], Tuomas Korpi[1] and Jukka Heikkonen[2]

[1]School of Engineering and Technology, Savonia University of Applied Sciences, Opistotie 2, 70101 Kuopio Finland
[2]Department of Future Technologies, University of Turku, Vesilinnantie 5, 20500 Turku Finland

ABSTRACT

In this article, a Smart Ping Pong Paddle has been introduced as an example of the use of sensor technology in sports. We have devised an accelerometer and a gyroscope sensor for the analyzing purpose and have gathered motion data of the game object to make a real-time 3D replica of the paddle to get an actual orientation of it. Technical details and principles of how to get the digital motion processing data from sensor to microcontroller and again transfer that wirelessly to a 3D modeling software are examined. Technical details are applied in practice, and the working demo of Smart Ping Pong paddle is built. Also, a couple of examples of other similar applications in the realm of object orientation sensing are overviewed.

KEYWORDS

Accelerometer Sensor, Gyroscope Sensor, Ping Pong Paddle, and Motion Analysis

1. INTRODUCTION

In recent years, the use of sensors has progressively utilized in analyzing several games. [1]-[3]. A game coach needs to instruct their trainees in real-time using a 3D replica of the ping pong paddle. In this article, we have succeeded in demonstrating a real-time analysis of the ping pong paddle. There is a massive potential to use IoT technology and embedded devices in sport to gather useful information and to make the sports training more efficient or gameplay more interesting. A game of table tennis is a familiar sport, and a six-axis (Gyro + Accelerometer) MEMS motion tracking sensor is cheaply available in the market. Therefore we came up with an idea to integrate the InvenSense MPU-6050 sensor inside a table tennis racket. Gyroscope and accelerometer data can be combined to calculate an orientation of the object. Orientation (angular position) and position should not be confused since it is not possible to get a position with 6-axis gyro/accelerometer. The position specifies the location and orientation provides the direction that the object is facing around an axis.

Table tennis racket cannot be hooked into a computer with a cable, therefore, it needs a wireless solution so that it seamlessly connects to a system and gather the vital orientation information at the real-time of gaming. An ESP8266 based microcontroller is used for wireless data transfer. With that same ‘wireless and no need to be next to computer’ -mentality, haptic feedback aka vibration motor is used in the handle of the Smart Ping Pong Paddle to inform the player when a connection between a wireless access point and receiver station is established.

David C. Wyld et al. (Eds): ITCSE, NLCA, ICAIT, CAIML, ICDIPV, CRYPIS, WiMo - 2020
pp. 241-248, 2020. CS & IT - CSCP 2020 DOI: 10.5121/csit.2020.100522

The receiver station is connected to a computer that handles 3D object manipulation based on data received. In principle many programs could be used to making and moving the 3D model like Prosessing3, Matlab, Blender or you could program software from scratch that reads serial port data and draws objects based on that. For this use case, the Unity3D game engine was chosen because it is a robust platform that specializes in moving game objects with C# code.

2. Related Works

Olli Koskenranta, in his thesis entitled "Manipulating 3D objects with gaze and hand gestures," [4] examines and creates a user interface for gesture-based 3D object handling. Gesture-based interaction is already used in playing games with Microsoft Kinect, PlayStation 3 Move and Nintendo Wii. For gaze tracking camera (Imaging Source DMK31AU03) was used. Nine-axis Accelerometer-gyroscope (ATR-Promotions WAA-010) with Bluetooth connection interpreted hand movements and sent the data to a laptop with JavaScript in the RealXten 3D platform. The user interface was expected to be able to manipulate objects in various ways, including object selection, gripping the object, releasing the purpose and rotating and moving it. Rotation and moving the camera was to be done with gaze and object manipulation with hand gestures. Regardless of inaccuracies in gaze function or limited hand gestures, the project displayed different user interaction methods and their future possibilities in a practical way. The article "Flex sensors and MPU-6050 sensors responses on a smart glove for sign language translation." written by A. Yudhana et al. [5] from Ahmad Dahlan University, Indonesia, goes over a design of a Smart Glove that translates hand sign language to a corresponding alphabet letter to a computer. Each finger has a flex sensor to detect a flex of the thumb, and on the top of the glove, there is a gyroscope-accelerometer sensor (MPU6050) to observe the tilt of the hand. Sensors are connected to Arduino microcontroller for processing and a Personal computer to monitor the value of sensor output value. Each gesture data (A-Z) is formed from seven different sensor values and each gesture data is taken 100 times to get a good sample size so that algorithms can predict the right sign by comparing samples.

3. Block Diagram of the System

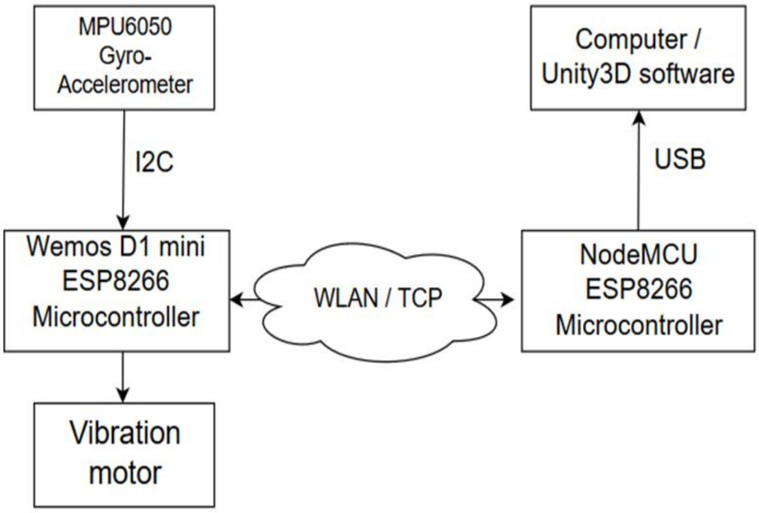
MPU6050 Gyro-Accelerometer
I2C
Wemos D1 mini ESP8266 Microcontroller
Vibration motor
WLAN / TCP
NodeMCU ESP8266 Microcontroller
USB
Computer / Unity3D software

Figure 1. Block Diagram of the System

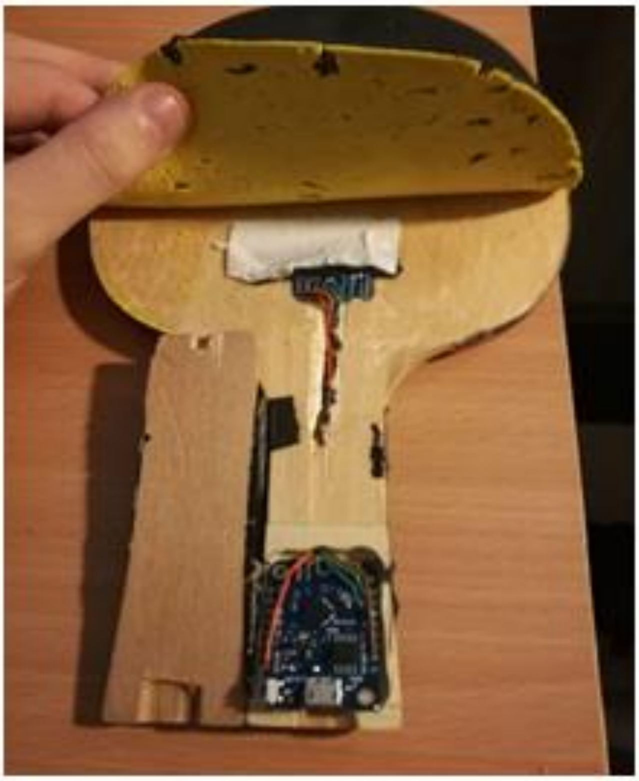

Figure 2. Embedded components

Components were chosen to be small so that they could be embedded inside a table tennis paddle and not affect the gameplay notably. MPU-6050 sensor was mounted under the rubber part of the paddle and handle was hollowed out to fit a small ‘Wemos D1 mini’ microcontroller to process sensor data via I2C bus, send haptic feedback to a small coin vibration motor (Parallax 28821) on the handle and use its built-in ESP8266 WiFi-module to transfer data wirelessly using TCP (Transmission Control Protocol). Components in the paddle get their power from 3xAA (4,5V) battery holder with an on/off switch. A battery holder is mounted on a player's wrist. Data receiving end of the system consist of another ESP8266 based microcontroller that collects data and sends it to the computer via USB, then serial monitoring is used to get the data in Unity3D, where a 3D model is ready to move based on the calculated orientation values. Receiving microcontroller does not need to be that small so NodeMCU is used instead of Wemos D1 mini to get more processing power and therefore smoother movement. The block diagram of the system and the embedded circuits within the paddle are shown respectively in Figures 1 and 2. Microcontroller D1 mini input voltage is 4.5 Volts trough a 5V pin. It has built-in 3.3 V voltage regulator so all of its digital pins output 3.3 Volts. The vibration coin motor has an operating voltage of 3.3 V and it is regulated through the D5 pin. Pins D1 and D2 act as SCL and SDA lines and connect to corresponding pins on the MPU-6050 sensor and it is powered through 3,3 V pin from D1 mini. The wiring diagram is shown in Figure 3.

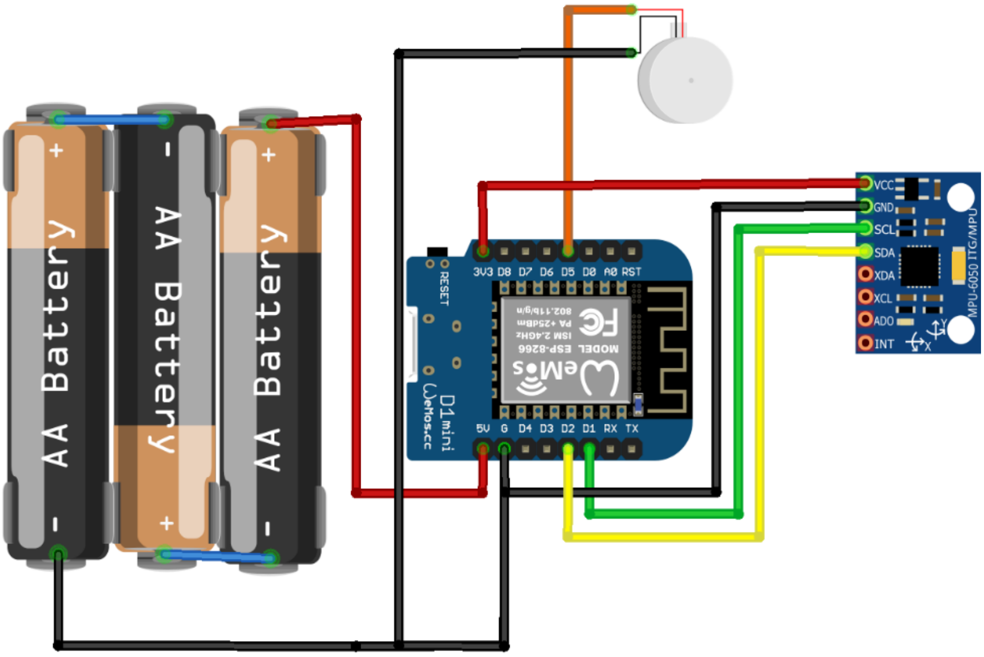

Figure 3. Wiring diagram made using Fritzing.

4. Accelerometer and Gyroscope

The InvenSense MPU-6050 sensor contains a MEMS (Micro-Electro-Mechanical Systems) accelerometer and a MEMS gyro in a single chip. It is the IMU (inertial measurement unit) sensor that is used to get the orientation of the object attached to it in three-dimensional space. MPU-6050 is a six-axis IMU sensor meaning that it gives out three gyro values and three accelerometer values as output. Each output has a 16-bit analog-to-digital converter to digitalize the output. There are many applications for orientation sensing like smartphone screen rotation, self-balancing robots, and quadcopters.

An accelerometer measures the direction of inclination and its magnitude. Gyroscopes output the rate of rotation. Both accelerometer and gyro use MEMS technology meaning that they are microscopic devices, particularly with moving parts. They can convert mechanical movement to electrical current that derives to orientation values.

By themselves, accelerometer nor gyroscope data is not accurate enough for smooth digital motion processing because accelerometer data is too noisy and gyroscope data starts to drift with time, but by combining those data sets, it is possible to get a precise angular position of the object. For the data merger, an algorithm called ‘complementary filter’ is used. Formula of the filter is: *“angle = 0,98 * (angle + gyroData * dt) + 0,02 * (accelData)”,* where *“dt”* is sample rate in seconds and other values are constants or sensor data. That angle defines the rotation around the axis. With MPU-6050, we can calculate rotation around X-axis (pitch) and Y-axis (roll) since gravity measured by accelerometer defines down for them. Z-axis rotation (yaw) cannot be calculated without a reference point for down, for example, a magnetometer to determine north.

4.1. I2C Protocol

The MPU-6050 communicates to a system processor using the I2C serial interface that consists of a serial data line (SDA) and Serial clock line (CKL). The MPU-6050 always acts as a slave when communicating with the system processor. Microcontroller (Master) will send a 7-bit device address plus one bit of reading data frame to a sensor with a specified address. If the address matches, the microcontroller reads sensor data. Because of specified addresses microcontroller can read both accelerometer data and gyro data using only two wires. At InvenSense's webpage [6], there is a Register map for MPU-6050, where registers for each accelerometer/gyro axis are mentioned. Each axis has two registers for LSB and MSB. In that register document, there are also lots of other useful information that is used in Arduino IDE code for microcontroller, like full-scale range and instrument sensitivity.

4.2. TCP Protocol

ESP (NodeMCU) acts as an Access Point (Server) and another ESP (D1 mini) as a Station (Client). Then they'll establish wireless communication and the client sends a message to the server. The server is continuously listening for a connection when it successfully establishes a connection and receives a message it prints that string on the serial monitor. The relationship between ESPs means that the Local Area Network (LAN) has been generated with its own Service Set Identifier (SSID) and password. Every device on the network has a personal Internet Protocol (IP) address but alone, it can't guarantee that communication is received in the same order it was sent. That is where Transmission Control Protocol (TCP) comes in. TCP makes data transmission reliable by ordering and re-sending corrupted packages. ESP8266 boards have a TCP library for Arduino IDE.

4.3. Arduino IDE

Both server and client microcontrollers are programmed using Arduino IDE with ESP8266 add-on. The client microcontroller utilizes the 'Wire' library to enable I2C communication between sensors and microcontrollers. 'ESP8266WiFi' and 'ESP8266WebServer' libraries come with ESP8266 add-on and they enable TCP communication. The final library needed is 'ArduinoJson' library for message decoding. All the tasks from reading sensor data, sending haptic feedback when TCP connection is created, and sending and receiving data is done in Arduino IDE and uploaded to microcontrollers. Unity's code is only left to handle data merging and object movement. Figures 4 and Figure 5 display flowcharts of Arduino IDE code for client and server, respectively.

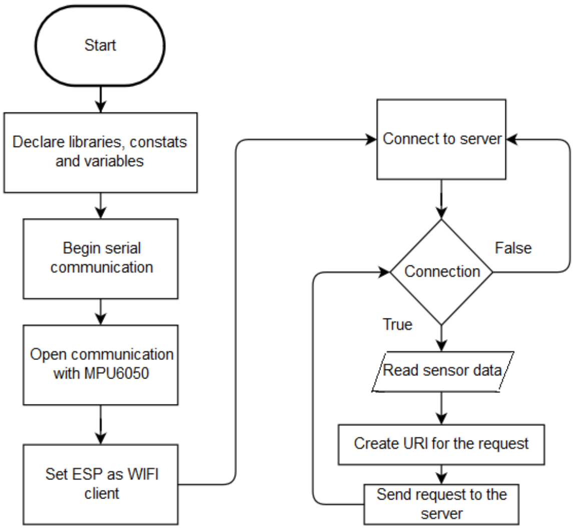
Start
Declare libraries, constats and variables
Begin serial communication
Open communication with MPU6050
Set ESP as WIFI client
Connect to server
Connection
False
True
Read sensor data
Create URI for the request
Send request to the server

Figure 4. Flowchart of the client Microcontroller

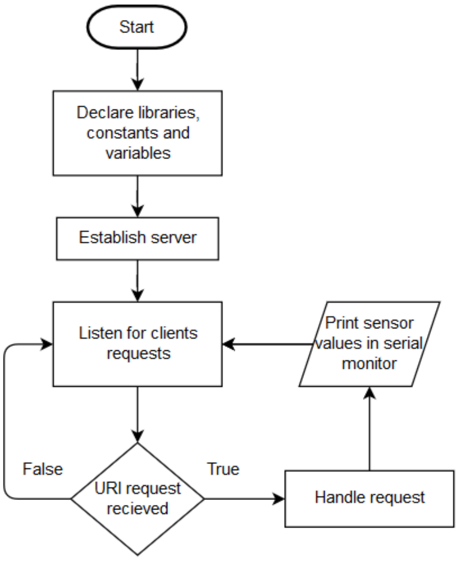
Start
Declare libraries, constants and variables
Establish server
Listen for clients requests
Print sensor values in serial monitor
False
True
URI request recieved
Handle request

Figure 5. Flowchart of the server Microcontroller

4.4. The Unity 3D

Unity is a cross-platform game engine. A paddle object is imported to Unity and the script component is attached to it. Unity uses the C# programming language. In the code script, the unity listens to the serial port where the microcontroller is sending raw sensor values. The complementary filter is applied to those and pitch and roll values are calculated and the object is rotated based on those. [7]

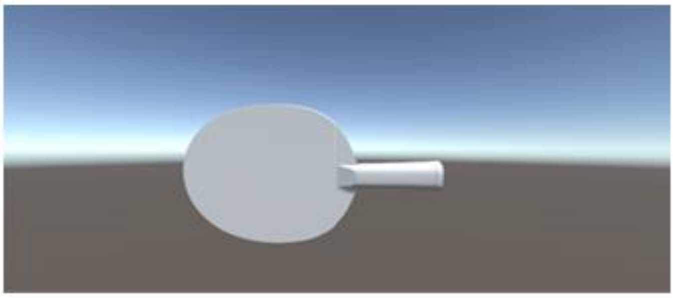

Figure 6. Paddle object in Unity 3D

5. Results Analysis

We have carefully discussed all the embedded components that have been utilized for the analysis

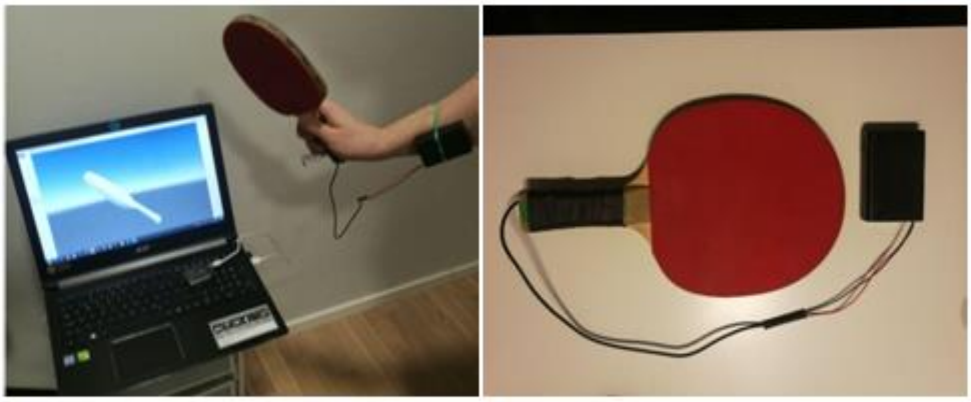

Figure 7. Working Porotype

of the inertial measurement of the ping-pong paddle. The working prototype is shown in Figure 7 and the current orientation movement is only based on gyroscope values and it drifts over time. We were able to transmit the data seamlessly from the gyroscopes to the server for further analysis. Sensor values were transferred smoothly only after the server microcontroller was changed from D1 mini to Node MCU. It has been noticed that with the MPU-9250 nine-axis (Gyro + Accelerometer + Compass) sensor, yaw-value could be calculated with the digital compass in it and the object would rotate around all three axes.

6. Conclusion and Future Direction

This work acts as proof of concept for wireless orientation transformation with MPU-6050. The earlier idea was to develop the Smart Ping-Pong Paddle for the use of table tennis players to track game performance. The orientation and movement of the paddle are successfully transmitted wirelessly on the 3D modeling Unity platform. However, we are still improving our work. In the current situation, the yaw (z-axis rotation) has not been working fine due to the complementary filter. There are significant potential and possibilities to add to the Smart Ping Pong Paddle simulation as with Unity, a whole game could be built around the wireless controller to combine sports and gaming.

Acknowledgments

The authors would like to thank the Department of Future Technologies, the University of Turku for financial support.

References

[1] Ping Yang, Zhengtao Zhang, Huawei Wang and De Xu, "Design and motion control of a ping pong robot," *2010 8th World Congress on Intelligent Control and Automation*, Jinan, 2010, pp. 102-107, doi: 10.1109/WCICA.2010.5553878

[2] T. Yamashita and T. Kobayashi, "Smart ping pong racket by ultrathin piezoelectric strain sensor array," *2018 Symposium on Design, Test, Integration & Packaging of MEMS and MOEMS (DTIP)*, Roma, 2018, pp. 1-3, doi: 10.1109/DTIP.2018.8394237.

[3] Rajeev Kanth, Jukka-Pekka Skön, Kari Lehtomäki, Paavo Nevalainen, Petra Virjonen, Mikko-Jussi Laakso, Jukka Heikkonen, ”Image Analysis and Development of Graphical User Interface for pole vault action”, Journal of Image and Graphics, Vol 6, No. 2, pp: 109-116, December 2018, doi: 10.18179/joig.6.2.109-116

[4] Olli Koskenranta, “Manipulating 3D objects with Gaze and Hand Gestures”, Published as bachelor of engineering thesis at Information Technology and Telecommunication department, Oulu University of Applied Sciences Source: https://www.theseus.fi/bitstream/handle/10024/46652/Koskenranta_Olli.pdf?sequence=1&isAllowed=y

[5] A. Yudhana, J Rahmawan and C U P Negara, “Flex Sensors and MPU6050 sensors responses on smart flove for sign language translation”, IOP conference series on Materials and Engineering 403 (1):012032, dio: 10.1088/1757-899X/403/1/012032.

[6] A technical report on “MPU-6000 and MPU-6050 Register Map and Descriptions Revision 4.2”, published online by InvenSense Inc., document number RM-MPU-6000A-00, Revision 4.2, Release date 08/19/2013. https://www.invensense.com/wp-content/uploads/2015/02/MPU-6000-Register-Map1.pdf, last accessed on 20 January 2020

[7] An open source free 3d model using unity 3D, https://free3d.com/3d-model/tabletennispaddle-shakehandstyle-v1--10646.html

AUTHORS

Dr. Rajeev Kanth received a Doctor of Science (D.Sc.) in Information and Communication Technology from the University of Turku, Finland, in 2013 and a docent title in 2019. He is currently working as a Senior Lecturer at the Savonia University of Applied Sciences, Finland, where he is focusing on teaching and research on the Industrial Internet of Things (IIoT). Previously, he has worked at the Indian Space Research Organization (ISRO), Ahmedabad India, Royal Institute of the Technology (KTH), Stockholm, Sweden and the University of Turku (UTU), Finland, where he has been a Researcher, Post-doctoral Researcher, and the Senior Researcher respectively. His current research interests include image and video analysis, the Internet of Things, Big Data Analytics, and Artificial Intelligence. He has published more than 45 scientific articles in peer-reviewed conference proceedings and refereed journals in the field of computer science and communication technology. He has been a member of the IEEE communication society, IEEE cloud computing community, IEEE Earth Observation Community, and the green ICT community.

Tuomas Korpi was born in Finland, and currently, he is pursuing a bachelor's degree in the Internet of Things (IoT) from Savonia University of Applied Sciences, Finland. He has a keen interest in developing innovative and novel applications of the Internet of things using Arduino and Raspberry pi platforms with suitable sensors.

Jukka Heikkonen has been a professor of computer science at the University of Turku, Finland, since 2009. His current research as the head of the Algorithms and Computational Intelligent (ACI) research group at the University of Turku is related to intelligent and learning systems, especially including machine learning, probabilistic, and information-theoretical modeling issues applied in wide varying application domains. He has worked at top-level research laboratories and the Center of Excellence in Finland and international organizations (European Commission, Japan) and has led many international and national research projects. He has authored more than 150 scientific articles.