AIP Conference Proceedings

Throughput, latency and cost comparisons of microcontroller-based implementations of wireless sensor network (WSN) in high jump sports

Cite as: AIP Conference Proceedings **1883**, 020010 (2017); https://doi.org/10.1063/1.5002028
Published Online: 14 September 2017

Afandi Ahmad, Muhammad Faris Roslan, and Abbes Amira

View Online Export Citation

ARTICLES YOU MAY BE INTERESTED IN

Real-time bus location monitoring using Arduino
AIP Conference Proceedings **1883**, 020016 (2017); https://doi.org/10.1063/1.5002034

The influence of dispersion slope to the higher-order dispersion coefficients for highly-nonlinear fibers
AIP Conference Proceedings **1883**, 020008 (2017); https://doi.org/10.1063/1.5002026

Connectivity, interoperability and manageability challenges in internet of things
AIP Conference Proceedings **1883**, 020004 (2017); https://doi.org/10.1063/1.5002022

© 2017 Author(s).

Throughput, Latency and Cost Comparisons of Microcontroller-based Implementations of Wireless Sensor Network (WSN) in High Jump Sports

Afandi Ahmad[1,2, a)], Muhammad Faris Roslan[2, b)] and Abbes Amira[3,c)]

[1]*Department of Computer Engineering, Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia, Johor, 86400, Malaysia*
[2]*Reconfigurable Computing for Analytics Acceleration (ReCAA) Research Laboratory, Microelectronic and Nanotechnology-Shamsuddin Research Centre (MiNT-SRC), Universiti Tun Hussein Onn Malaysia, Johor, 86400, Malaysia*
[3]*Department of Computer Science and Engineering, Qatar University, P. O. Box 2713, Doha, Qatar*

[a)]Corresponding author: afandia@uthm.edu.my
[b)]farisrag@gmail.com
[c)]abbes.amira@qu.edu.qa

Abstract. In high jump sports, approach take-off speed and force during the take-off are two (2) main important parts to gain maximum jump. To measure both parameters, wireless sensor network (WSN) that contains microcontroller and sensor are needed to describe the results of speed and force for jumpers. Most of the microcontroller exhibit transmission issues in terms of throughput, latency and cost. Thus, this study presents the comparison of wireless microcontrollers in terms of throughput, latency and cost, and the microcontroller that have best performances and cost will be implemented in high jump wearable device. In the experiments, three (3) parts have been integrated – input, process and output. Force (for ankle) and global positioning system (GPS) sensor (for body waist) acts as an input for data transmission. These data were then being processed by both microcontrollers, ESP8266 and Arduino Yun Mini to transmit the data from sensors to the server (host-PC) via message queuing telemetry transport (MQTT) protocol. The server acts as receiver and the results was calculated from the MQTT log files. At the end, results obtained have shown ESP8266 microcontroller had been chosen since it achieved high throughput, low latency and 11 times cheaper in term of prices compared to Arduino Yun Mini microcontroller.

INTRODUCTION

High jump sports is classified as complex cyclic-acyclic movements where the aim is to bring the jumper's centre of mass to a maximum height when crossing the bar [1]. To attain high-performance and low-injury, attention to the high jumpers' technique toward take-off is significant importance. To gain a good take-off technique, speed at approach stage [2]–[8] and force during take-off [2]–[5], [7], [9]–[14] are two (2) important parts that need to capture during the high jump. Thus, data capture like sensors need to record the run speed and force at ankle during the jump.

It is a common trend in sports to use wireless connection in the internet of things (IoT) era, most of the athletes nowadays appreciate wearable device that capture speed, calorie, heart-beat and steps to maintain their healthy life style and to ensure they fit to perform during competitions [15]. However, there is no real-time wearable device that capture speed and force of high jumpers. To develop a wearable device, the needs of sensor and microcontroller that have wireless capability function are important to ensure athletes use the device freely without any wire be deployed on their body.

Wireless sensor network (WSN) is a network of sensors that observe the data environment and send the data wirelessly to the system receiver [16]. To build WSN, it requires sensor and interfacing devices like microcontroller

Advances in Electrical and Electronic Engineering: From Theory to Applications
AIP Conf. Proc. 1883, 020010-1–020010-9; doi: 10.1063/1.5002028
Published by AIP Publishing. 978-0-7354-1563-8/$30.00

to attach together. High jumper total average run times from start to end is around six (6) seconds. Consequently, it needs WSN that has the ability to transmit data wirelessly at 10 data per seconds that is 100ms per data. It is because, high jump sports does not require to capture high speed data like others sports for instance soccer sports [17] that captured altitude of slope during slide. To solve the issues of data transmission, WSN devices in term of errors in latency and throughput need to be minimise and the design must be at affordable price. In this research, microcontroller ESP8266 and Arduino Yun Mini was used and connected to force and global positioning system (GPS) sensor that functions as a WSN, it is because both of the microcontroller contains built-in Wi-Fi and small in term of size. Besides, the proposed of this WSN device simply need to capture speed before take-off and force during take-off.

To overcome these concerns, this study discusses the transmission network issues of WSNs in terms of throughput and latency of the microcontroller. The analysis of throughput and latency of a WSN involves observing these parameters in varying setup mode. It is targeted that one (1) of the microcontroller will be selected in terms of best performances including latency and throughput of the reliability to determine networks quality of the system to be used in the high jump wearable devices. Other than that, the price also one (1) of the key selection of the microcontrollers.

The rest of the paper is organised as follows. The related work is summarised in the following section. Details of the method and analysis used to test the network of the system is addressed in the next section. Finally concluding remarks are given in final section.

DESIGN OF WIRELESS SENSOR NETWORK (WSN)

An overview of the throughput and latency issues of the WSN and the cost issues of the WSN devices are presented in the following sections.

Throughput and latency issues

Throughput is a measure of total units of information a system can process in a given amount of time [18]. Related measures of system output include, the speed with which some specific workload can be completed and response time, the amount of time between a single interactive user request and receipt of the response. Latency is the delay from input into a system to desired result. In data transmission, the medium itself examples like optical fiber, wireless, or any transmission medium introduces some delay, which differs from one medium to another [16]. The size of the packet transmit per time presents a delay, meanwhile a larger packet will take longer to receive and return than a small one. Similarly, when signals must be boosted by a repeater, this will introduce additional latency.

To reveal the importance of these experiments, a summary of issues by the previous researchers are tabulated in TABLE 1. In brief, it can be concluded that throughput and latency are one (1) of the important issues concerning the networks transmission and further research needs to be carried out, especially with the hardware implementation, the elements of embedded electronics like microcontrollers and sensors

TABLE 1. Summary of the related works

Refs.	Contributions
[18]	Performance analysis of an IP based protocol stack for WSNs
[16]	MX-MAC: A Multichannel based low latency asynchronous MAC protocol for Wireless Sensor Networks
[19]	Routing behaviour across WSN simulators: The AODV case study
[20]	Poster Abstract: Wake-Up Receivers for Energy Efficient and Low Latency Communication
[21]	Superframe Duration Allocation Schemes to Improve the Throughput of Cluster-Tree Wireless Sensor Networks
[22]	An approximate bandwidth allocation algorithm for trade-off between fairness and throughput in WSN

To overcome these issues, throughput and latency of WSN devices need to be measured to check the performance of successful transmission rate of the microcontroller. In these experiments, sensors will be used as an input to the microcontroller two (2) wireless built-in microcontrollers will be used as a performances comparison and the output of the wireless transmission will be measured in terms of throughput and latency.

Cost issues

The cost of WSN devices are one (1) of the concerning factor in the market today. Most of the sensors are cheap, however the WSN devices that connected or controlled the sensors are expensive. Some WSN devices have sensors, controller and wireless interface to send the data to the servers, this make the WSN devices bigger and costly. In learning WSN processes and protocols, a method might be found to implement the same functions without having to use expensive WSN device.

These days, most of the microcontrollers has implemented a built-in Wi-Fi and much cheaper compare to external wireless connection. Microcontrollers mass-produced by most productions normally have purposes that are not desired in particular system, whereas certain protocols are not necessary in low transmission network. The microcontrollers of ESP8266 and Arduino Yun Mini as illustrated in TABLE 2 can transmit low transmission data and are built in small size and it can turn into WSN devices that be able to attach to body of athletes, by not disturbing their movements.

TABLE 2. General WSN microcontrollers comparison [23], [24]

Specification	ESP8266	Arduino Yun Mini
Controller	Xtensa single core 32-bit L106	Atheros AR 9331
Wi-Fi protocol	802.11 b/g/n 2.4 GHz	802.11 b/g/n/e/i 2.4 GHz
Clock speed	80Mhz	400Mhz
SRAM	160kB	64MB
GPIO	17	12
Operating voltage	3.3V	5V
Power consumption	170mA	170mA
Size	48 × 25 × 5mm	71.12 × 22.86 × 5mm

DESIGN & IMPLEMENTATION

An overview of the proposed WSN device that measured throughput and latency is demonstrated in Fig. 1**,** ESP8266 and Arduino Yun Mini microcontroller are selected for several reasons. The implementation of this microcontroller in real-time monitoring system is specifically to do tasks that any others microcontroller unable to do efficiently, such as low-power consumption, small area and Wi-Fi integrated. The Wi-Fi integrated in the microcontroller are mainly to transmit or receive data from the server wirelessly, thus it can be a wireless sensor node. An overview of the method is discussed in the following subsection and results obtained for both at the system and respondent levels are highlighted. It covers:

1. input used to test the transmission rate;
2. the process of the microcontroller to transmit the data; and
3. the output data received from microcontroller.

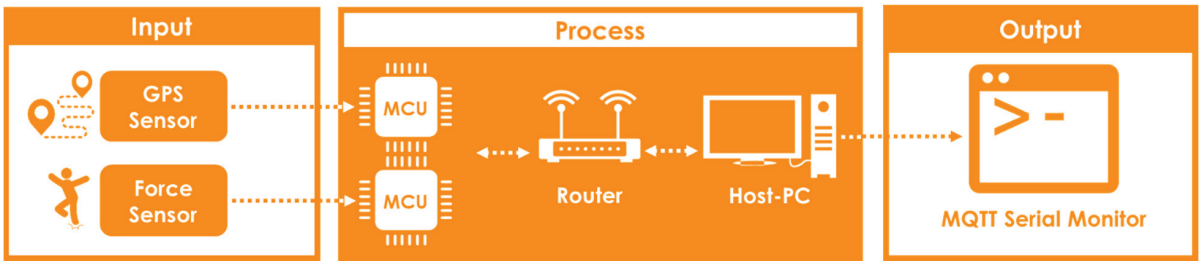


FIGURE 1. An overview of the method used to measured throughput and latency of the WSN device

Input

Two (2) sensors, consist of force and global positioning system (GPS) sensors act as an input to the device and all two (2) sensors were tested. The GPS sensor was giving an output of speed and the force sensor output is forces. This

is common sensors to track athlete in track and field sports like running, high jump and long jump. Each sensor was connected to one (1) of the microcontroller because, both of the sensors are used mostly in different location which is force sensor is at the ankle of the leg and GPS sensor at the waist of the body as illustrated in Fig. 2. Both of these sensors only acts as an input data to test the throughput and latency of the microcontroller.

FIGURE 2. Sensor inputs used by microcontroller

Each input sensors were tested using millis function to measure the throughput and latency. The millis function acts as time interval between data transmission. The selected time interval value are 1ms, 10ms, 100ms, 500ms and 1000ms. Each sensor produces an output of 3bit of data per transmission. This is because, speed and force produces three (3) arrays, one (1) array contain one (1) bit of data. TABLE 3 shows the input data to test throughput that had been converted from the input data interval in second using Equation (1).

$$Input\ Data\ (bit/s) = \frac{3bit}{Input\ data\ interval\ (ms)} \tag{1}$$

TABLE 3: Data setup value for throughput and latency

	Data input setup				
Input data interval (ms)	1	10	100	500	1000
Input Data (bit/s)	3000	300	30	6	3

The throughput was tested by measuring the transmission of input and output data in 60s interval to get average of 1s using Equation (**2**). Then, the percentage of error and successful using the following Equation (3) and (4). While, the latency was calculated using Equation (5) to get error of time interval between the transmission of one (1) data by substituting the output interval from the server and the input interval that has been setup from the microcontroller.

$$Throughput\ (bit/s) = \frac{Output\ Data\ (bit)}{60s} \tag{2}$$

$$Percentage\ of\ Error\ (\%) = \frac{Input\ Data\ (bit) - Output\ Data(bit)}{Input\ Data\ (bit)} \times 100\% \tag{3}$$

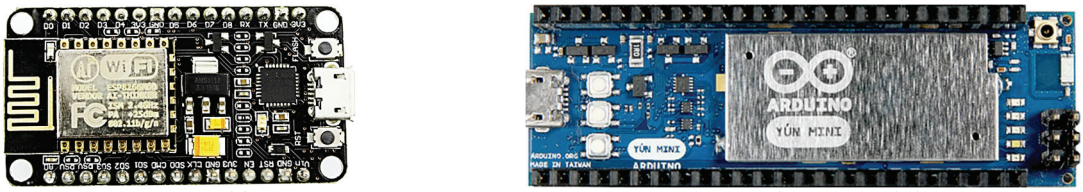
$$Percentage\ of\ Successful\ (\%) = \frac{Output\ Data\ (bit)}{Input\ Data\ (bit)} \times 100\% \tag{4}$$

$$Latency\ (ms) = Output\ Interval\ (ms) - \ Input\ Interval\ (ms) \tag{5}$$

Process

In this research, two (2) microcontroller were used that is ESP8266 and Arduino Yun Mini as demonstrated in Fig. 3. These microcontrollers process GPS and force input sensors that produce speed and force of the athlete that was programme by Arduino IDE software. The data from the sensors then been transmitted wirelessly to the internet router

using built-in Wi-Fi in both microcontroller boards. The used of Wi-Fi in microcontrollers acts as sensor node that transmit data to the router. The microcontrollers search any router that subscribe to it via internet protocol (IP) address using message queuing telemetry transport (MQTT) software and the Arduino serial monitor display the connectivity of microcontrollers toward the server as shown in Fig. 4.



(a) ESP8266 (b) Arduino Yun Mini

FIGURE 3. Microcontrollers used in these experiments.

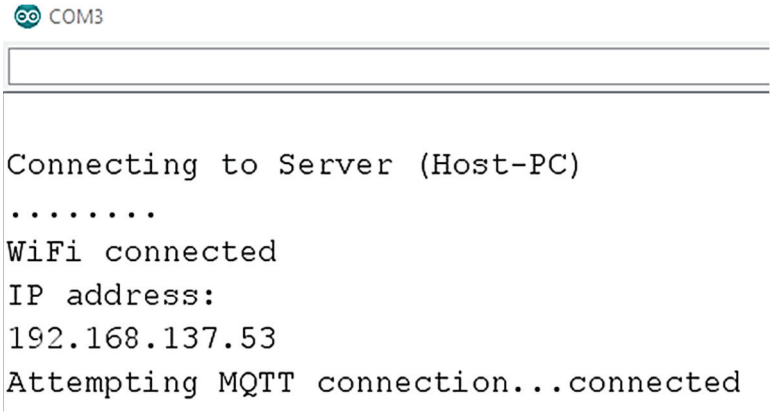


FIGURE 4. An overview of Arduino serial monitor.

Output

The server that acts as host-PC subscribe the sensor node and accept the data from it by using MQTT software. The data from the sensors been display through MQTT serial monitor or command windows as shown in Fig. 5. The received data will be save in the MQTT log file, the latency and throughput were measured and analysed from the log files.

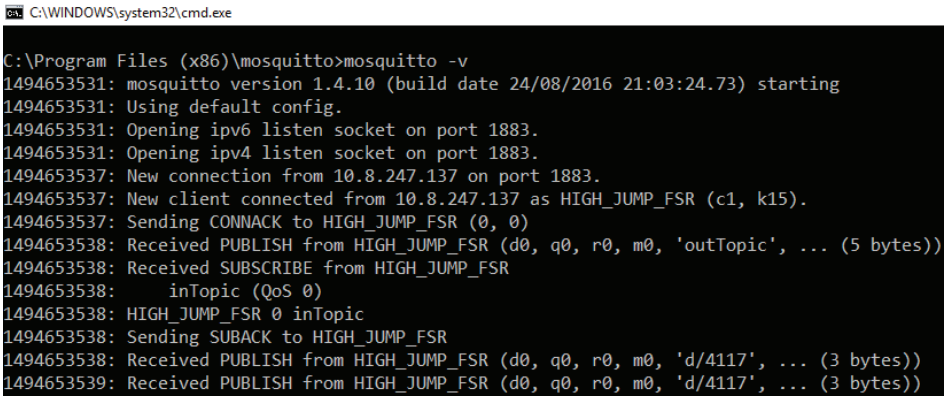


FIGURE 5. An overview of MQTT serial monitor.

RESULT & ANALYSIS

Two (2) simple wireless sensor network (WSN) deployment were constructed, tested and compared by using the microcontroller ESP8266 and Arduino Yun Mini. The input data are produced by force sensor that used three (3) bit to send data and the same applied to GPS sensor. In this section, results obtained for both microcontrollers are highlighted. It covers experimentation of throughput and latency.

Throughput Experimentation

The throughput was tested on fixed time that is at total of 60 seconds and on varied setup throughput of data (bit) to test the performance of both microcontrollers at a baud rate of 115,200bps. The environments were tested in open space settings that is at the track and field stadium. Five (5) test input data setup were chosen for both microcontrollers that are 3bit, 6bit, 30bit, 300bit and 3000bit data. A test data was transmitted and the number of successfully received data was counted in order to get the amount of received data.

Table 4 summarises the output data transmitted and percentage of errors while Fig. 6 shows the percentage of the successful transmitted data.

TABLE 4. Results of output data transmitted and percentage of errors.

Input data (bit)	Output Data (bit)		Error (%)	
	ESP8266	Arduino Yun Mini	ESP8266	Arduino Yun Mini
3	3.00	3.00	0.00	0.00
6	6.00	5.95	0.00	0.83
30	29.95	29.40	0.17	2.00
300	96.90	89.80	67.70	70.07
3000	99.40	90.20	96.69	96.99

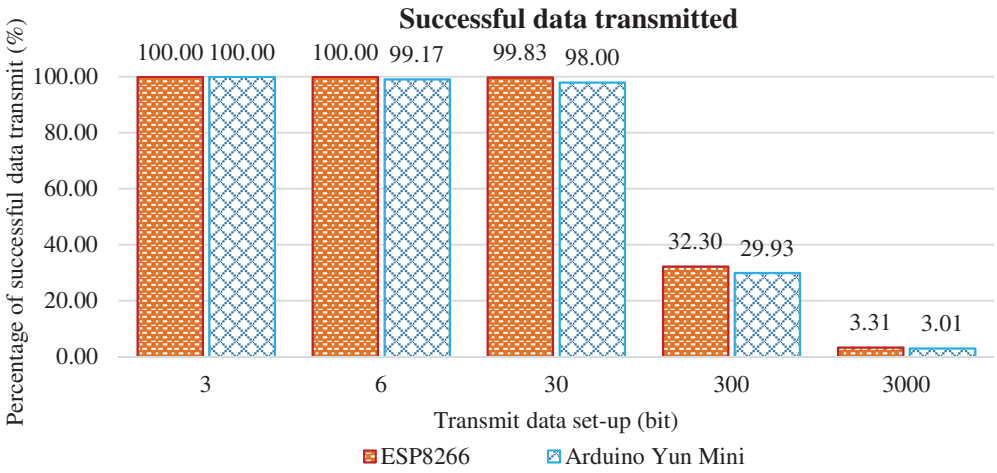


FIGURE 6. Percentage of the successfully transmit data.

The results for throughput was measured with different input data transmission setup, both microcontroller has slightly different in terms of error. Both cannot transmit data at 3000bit accurately and gain maximum of 96.99% of error from Arduino Yun Mini. While, at 3bit and 30bit the data transmission starts to stabilise and the minimum error of 0.17% were counted from ESP8266. The percentage of successful was both 100% at input data of 3bit and 6bit by ESP8266, yet Arduino Yun Mini happen to have an error at 6bit that is 0.83%. Hence, by decreasing the input data will decrease the error of the throughput. From the result, it can be assumed that starting from 30bit to 3bit the percentage of error are approach to 0% when tested to both microcontrollers. Hence, the ESP8266 microcontroller occurred to be less error of throughput compared to Arduino Yun Mini.

Latency Experimentation

The latency was tested by measuring the error of interval between the input and output of one (1) data and on varied input time interval (ms) to test the performance of both microcontrollers at a baud rate of 115,200bps. The environments were tested in open space settings that is at the track and field stadium. Five (5) input time interval were chosen for both microcontrollers that are 1ms, 10ms, 100ms, 500ms and 1000ms. A test data was transmitted and the time interval between data transmit and receive are measured in order to get the amount of time interval delay per data. TABLE 5 summarises the output time interval while Fig. 7 shows the graph of the time delay error.

TABLE 5. Results of output time interval per data.

Input Time Interval (ms)	Output Time Interval (ms)	
	ESP8266	Arduino Yun Mini
1	40.99	43.02
10	39.48	42.84
100	100.01	114.4
500	500	513.86
1,000	1,000	1,009.9

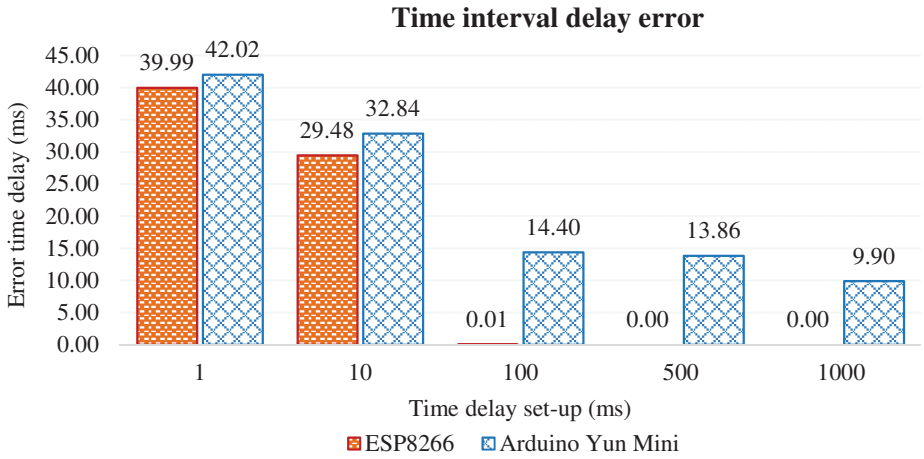


FIGURE 7. Transmit data time delay.

The results for latency was measured with different input time interval per data, both microcontroller has slightly different in terms of error. Both cannot transmit data at time interval of 1ms accurately and the time interval delay error exceed by maximum of 42.02ms from Arduino Yun Mini. However, there is no error occurred for 500ms and 1000ms time interval of ESP8266 and the minimum error of 0.01ms were counted at 100ms time interval. While, Arduino Yun Mini stills have great gap that is around 14.40ms to 9.90ms time interval delay error at 100ms, 500ms and 1000ms. Thus, only ESP8266 successful transmitted data with no error at 500ms and 1000ms, thus far Arduino Yun Mini happen to have an error at every input setup time interval. Hence, by increasing the time interval input setup will decrease the error or the latency of the time interval. From the result, it can be assumed that starting from 100ms to 1000ms the error of time interval delay are approach to 0ms when tested to ESP8266. Therefore, the ESP8266 microcontroller occurred to be less error of time delay compared to Arduino Yun Mini.

CONCLUSION

In this paper, an investigation of the performances of two (2) microcontrollers ESP8266 and Arduino Yun Mini in term of throughput and latency had been analysed. The best performance between two (2) microcontrollers have been selected and implemented in the WSN device of high jump sports that measured speed and force. The throughput results show the best input setup data that need to transmit per seconds is starting from 30bit and below. While, latency results show the best setup time interval that need to transmit per data is starting from 100ms and above. This is due to the lower the setup data that need to transmit, the higher the percentage of successful throughput data of a microcontroller. While, the higher the time interval setup data, the less latency occurs to the microcontroller. It is because, it gave more time and less data to transmit per cycle. Therefore, it proved that ESP8266 gave better performance in terms of latency and throughput. Other than that, ESP8266 has smaller size that can fit in any WSN devices and it is also 11 times cheaper compare to Arduino Yun Mini.

ACKNOWLEDGMENTS

The authors would like to thanks to Postgraduate Research Grant Scheme (GPPS) UTHM for funding this research work and Institut Sukan Negara (ISN) for their support.

REFERENCES

1. F. Langer and A. Langerova. Kinesiology. **40**, 107–113 (2008).
2. M. P. Grieg and M. R. Yeadon, J. Appl. Biomech.**16**, 367–378 (2000).
3. J. Isolehto, M. Virmavirta, H. Kyrolainen, and P. Komi. New Stud. Athl. **22**, 17–27 (2007).
4. M. Čoh. Serbian J. Sport. Sci. **4**, 127–135 (2010).

5. C. Wilson, M. A. King, and M. Yeadon. J. Biomech. **44**, 2207–2212 (2011).
6. J.-P. Goldmann, B. Braunstein, K. Heinrich, M. Sanno, B. Stäudle, W. Ritzdorf, G.-P. Brüggemann, and K. Albracht. ISBS-Conf. Proc. Arch. **33**, 3 (2016).
7. R. M. Alexander. Philos. Trans. Biol. Sci. **329**, 3–10 (1990).
8. B. Dragos, N. Mircea, B. Alexandru, and M. Ilie. Ovidius Univ. Ann. **16**, 7 (2016).
9. B. Van Gheluwe, P. Roosen, and K. Desloovere. J. Appl. Biomech. **19**,13–27 (2003).
10. I. Blažević, L. Antekolović, and M. Mejovšek. Kinesiology. **38**, 63–71 (2006).
11. C. Wilson, M. M. R. Yeadon, and M. A. King. J. Biomech. **40**, 3155–3161 (2007).
12. J. C. C. C. Tan and M. R. Yeadon. J. Sports Sci. **23**, 775–80 (2005).
13. M. A. King, C. Wilson, and M. R. Yeadon. J. Appl. Biomech. **22**, 264–274 (2006).
14. G. M. Dousoky, M. Shoyama, and T. Ninomiya. *13th European Conference Power Electronics and Applications.* 2009. pp. 1–8.
15. M. Maheedhar, A. Gaurav, V. Jilla, V. N. Tiwari, and R. Narayanan. *38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. 2016. pp. 6290–6293.
16. M. M. Kamal, S. S. Moni, and M. S. Alam. *29th International Conference on Electrical and Computer Engineering (ICECE)*. 2016. pp. 439–442.
17. K. W. Nixon, Xiang Chen, and Yiran Chen. *21st Asia and South Pacific Design Automation Conference (ASP-DAC)*. 2016. pp. 563–568.
18. S. Thombre, R. Ul Islam, K. Andersson, and M. S. Hossain. *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 2016. pp. 360–365.
19. I. Minakov, R. Passerone, A. Rizzardi, and S. Sicari. *IEEE World Conference on Factory Communication Systems (WFCS)*. 2016. pp. 1–8.
20. F. A. Aoudia, M. Magno, M. Gautier, O. Berder, and L. Benini. *15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. 2016. pp. 1–2.
21. E. Leão, C. Montez, R. Moraes, P. Portugal, and F. Vasques. Sensors. **17**, 249, (2017).
22. Y. Cheng, S. Xiao, J. Liu, F. Guo, R. Qin, B. Li, and X. Yuan. Wirel. Networks. 1–13 (2017).
23. "ESP8266EX Overview | Espressif Systems." [Online]. Available: https://espressif.com /en/products/hardware/esp8266ex/overview. [Accessed: 17-Mar-2017].
24. "Arduino - Arduino Board Yun Mini." [Online]. Available: https: //www. arduino.cc /en/ Main/ Arduino Board Yun Mini. [Accessed: 17-Mar-2017].