Design Flow of Wearable Internet of Things (IoT) Smart Workout Tracking System

Jose Toledo Monsalve, David Arnold, Won-Jae Yi and Jafar Saniie

Embedded Computing and Signal Processing (ECASP) Research Laboratory (http://ecasp.ece.iit.edu)
Department of Electrical and Computer Engineering
Illinois Institute of Technology, Chicago IL, U.S.A.

***Abstract*—In this paper, we present a wearable Internet of Things (IoT) based system that assists users' training sessions with precise workout records through real-time multiple sensor data analysis. By pairing an accelerometer and an ultrasonic sensor, our system can accurately detect user's workout sessions, while a central coordinator, Personal Communication Node (PCN), is used to analyze workout session recognitions and transmitting user movement detection data to the central server or the cloud. By utilizing customized web and Android software applications, users can keep track of their workout sessions remotely, along with viewing all collected sensor data. The system has been designed to be modular in order to allow new functionality to be added or replaced based on the needs of the user. For example, this system can be extended to detect complex workout movements and combine other vital sign sensor data for analyzing any sport activities. Also, this system has the potential to be reconfigured to different purposes such as a statistics and surveillance system for round-the-clock care of patients, people with disabilities and working in hazardous environment.**

I. Introduction

Athletes and people trying to stay in shape can benefit greatly from having accurate records of their workout sessions. With precise workout data, an athlete can assess which parts of their training are producing better results, can keep track of how close they are to achieving their goals, thereby keeping high levels of motivation and can plan future activities based on past performance. By having real-time data available for monitoring by coaches and physicians, the benefits can be increased further. For example, coaches can view the number of repetitions an athlete has performed and gauge their level of exertion. In addition, this information can be used to instruct the athlete to reduce or increase the pace of the workout. In the context of fitness competitions, such as the CrossFit Games [1], where athletes often compete to do the greatest number of repetitions of an exercise in a given time, it is also important to keep track of athlete's activities for scoring purposes.

By taking advantage of Wireless Body Sensor Networks (WBSN) and Internet of Things (IoT) architecture, both historical and real-time workout data can be obtained in an efficient and automated manner. WBSNs make use of light-weight, low power consuming wearable sensors to monitor a user's status, motion patterns or other physiological characteristics [2]. On the other hand, implementing an IoT architecture can increase a system's flexibility, usability and productivity by having devices connected through the Internet to computing resources running on the cloud [3].

In this paper, we introduce a design flow for workout tracking monitoring system using multiple wearable sensors based on the IoT architecture. To show the feasibility of this design flow, we have implemented a system that focuses on counting the number of sets and repetitions of the user's workout who performs two types of bodyweight exercises, squats and push-ups, which are the core of the most bodyweight training routines. Furthermore, the system is capable of providing real-time workout session histories to monitor the workout activity for multiple authorized users.

II. System Design

Our smart workout detection system is based on the concept introduced in [4] of a Personal Communication Node (PCN), which collects data from sensors, performs analysis and signal processing and forwards data to a central server through the Internet. The system consists of the following component: two sensor nodes on the user's thigh and chest, a PCN for collecting and processing data from sensor nodes, a central server for receiving, storing and analyzing data from the PCN. In addition, there are two different types of devices for controlling the system and viewing data: a PC client enables access to the system through a web application, and a mobile client, specifically an Android smartphone, that accesses the system through an Android software application. Figure 1 shows the overview of our smart workout detection system.

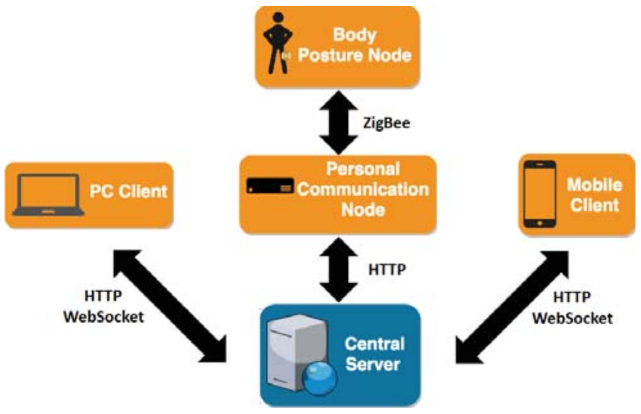


Figure 1. Overview of Smart Workout Detection System

In our implementation of the system, an accelerometer and an ultrasonic sensor are attached to the user's thigh. Other implementations could use other sensors such as gyroscopes or muscle activity sensors for more accurate measurements. Raw data from the sensors is interpreted by the PCN to infer the user's posture and movement. In our system, we use accelerometer data to classify the user's posture into 4 different positions as shown in Figure 2: standing, squats, plank high and plank low. Transitions between these positions are tracked to determine push-up and squat events. Events detected at this stage are forwarded over the Internet to the central server.

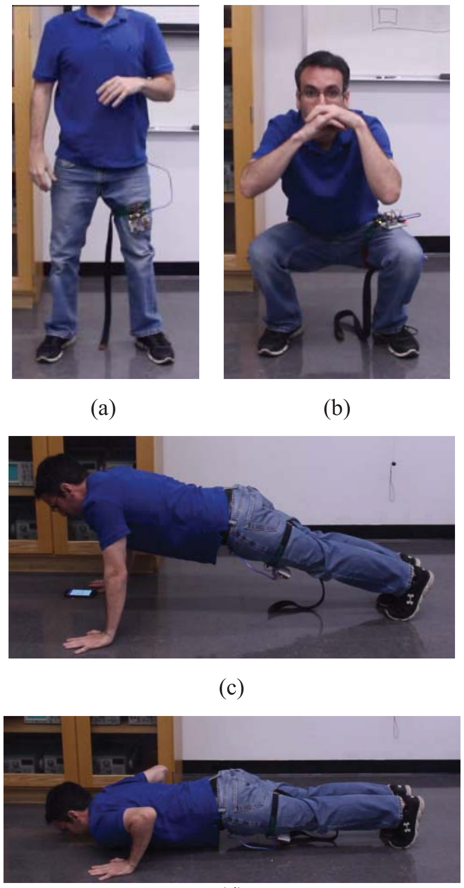


Figure 2. Different Workout Positions; (a) Standing, (b) Squat, (c) Plank High, (d) Plank Low

At the central server, data can be further analyzed to get insights into the user's workout performance and current workout progress. In our implementation, push-up and squat events are categorized into sets along with calculated resting times. Data from this stage can be stored on a database or pushed to either the PC clients or mobile clients in real-time that would perform the last stage of processing. In the last stage, data is displayed to the user, physicians or coaches in graphs and tables through a customized graphical user interface (GUI). In our system design, we present user data through an Android application on a smartphone, and a web application which can be accessed through any web browser.

III. System Configuration

The body posture node, as shown in Figure 3, comprises an accelerometer, ADXL345 low-power tri-axial accelerometer [5], to measure the tilt of the user's thigh, and an HC-SR04 ultrasonic sensor [6], to measure the distance of the user to the floor when in a plank position. Both sensor data are retrieved periodically by an Arduino UNO microcontroller and transmitted using an XBee S2C ZigBee transceiver to the Personal Communication Node (PCN).

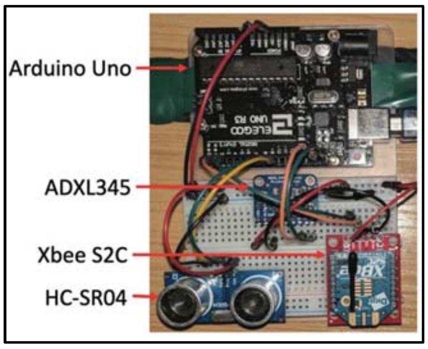


Figure 3. Body Posture Node Hardware

The Personal Communication Node (PCN), as shown in Figure 4, is implemented using a Raspberry Pi 3 B+ [7], which is a single-board computer whose small form factor and low power consumption make it easy to mobilize regardless of the user's workout location using an XBee S2C ZigBee transceiver connected via a USB port. Additionally, we utilize its built-in Wi-Fi adapter, Internet communication with the central server can be established easily.

Figure 4. Personal Communication Node using ZigBee-enabled Raspberry Pi

The central server is implemented using a Ruby on Rails application which is hosted on Amazon AWS and provides RESTful API, a web interface, access to SQLite database and ability to interface through WebSocket using ActionCable framework [8]. The web interface using the WebSocket keeps the data in real-time, and provides charting tools using Google Charts JavaScript library to provide insights about recorded workout sessions.

The mobile client in our system design is an Android smartphone where the customized application is responsible of starting and ending recording workout sessions. It connects to the central server using RESTful API and retrieves data in JSON format.

Table I shows customized messages of RESTful API implemented by the central server, where this is used throughout the system for the PCN, the mobile client and the PC client. POST request URLs are implemented to start and register detected workout events on the central server. GET request URLs are implemented to retrieve workout data stored in the central server, and a PUT request URL is implemented end recording workout sessions.

TABLE I. RESTFUL APIS FOR COMMUNICATING WITH CENTRAL SERVER

Message	Method	URL	Result
1	POST	/workouts	Starts new Workout
2	POST	/squat	Add new squat repetition to current workout
3	POST	/pushup	Add new push-up repetition to current workout
4	GET	/workouts	Get a list of workouts
5	GET	/workouts/{id}	Get details of a workout
6	PUT	/workouts/{id}	Ends a workout

IV. WORKOUT DETECTION ALGORITHM

The body posture node sends tilt and distance readings periodically to the PCN, in the form of a position vector $p = (x,y,z,d)$, where x,y and z are the acceleration values read on each axis of the accelerometer and the distance value d read by the ultrasonic sensor. The PCN analyzes the position vector in a similar way to the one introduced in [9] to detect push-up and squat events performed by the user by first classifying the user's position by comparing the position vector to the thresholds as shown in Table II. This information is used by the algorithm shown in Figure 5, which saves the current position and the last position to analyze user movements. This algorithm can detect transitions from the squat position to standing position (recorded as a squat event), and transitions from the plank low position to the plank high position (recorded as a push-up event).

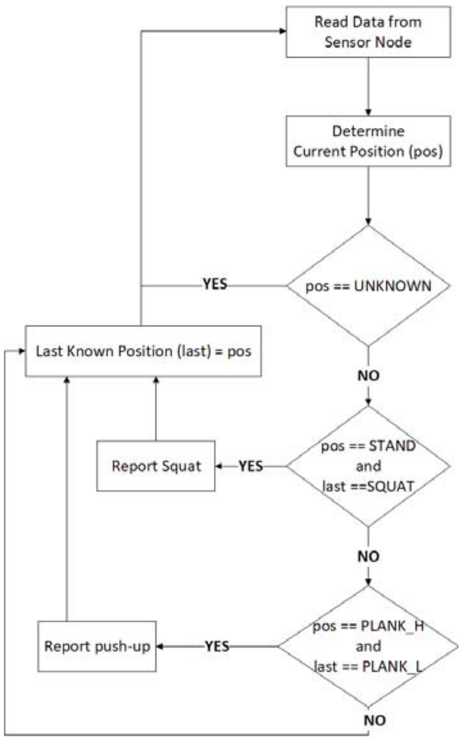

Read Data from Sensor Node
Determine Current Position (pos)
pos == UNKNOWN
YES
NO
Last Known Position (last) = pos
pos == STAND and last ==SQUAT
Report Squat
YES
NO
pos == PLANK_H and last == PLANK_L
Report push-up
YES
NO

Figure 5. Workout Detection Flowchart

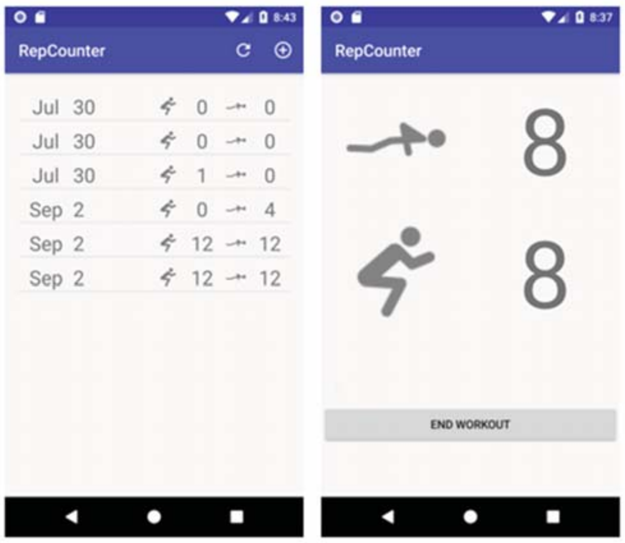
TABLE II. THRESHOLDS FOR WORKOUT POSTURE ASSESSMENT

Position	X (m/s^2)	Y (m/s^2)	Z (m/s^2)	D (cm)
Standing	$-3.6 < X < 3.6$	$8.0 < Y < 12.0$	$-3.6 < Z < 3.6$	$0 < D$
Squat	$-6.0 < X < 6.0$	$-3.6 < Y < 3.6$	$7.0 < Z < 12.0$	$0 < D$
Plank High	$-3.6 < X < 3.6$	$-1.0 < Y < 6.3$	$-12.0 < Z < -6.2$	$20 < D$
Plank Low	$-3.6 < X < 3.6$	$-1.0 < Y < 6.3$	$-12.0 < Z < -6.2$	$0 < D < 16$

Exercise events registered by the PCN are reported to the central server, where they are grouped into sets for each type of exercise. For each event, the server compares its type with that of the last event received, if they are different a new set is created. The duration of each set is calculated using the timestamp of the first and last repetition. The resting time for the workout session is calculated by subtracting the total duration of all sets from the duration of the workout.

V. EXPERIMENTAL RESULTS

Our system is capable of displaying historical data of previous workouts, as well as real-time information of the current workout. We utilize customized messages of RESTful APIs implemented on the central server to communicate with the PCN and the Android application. On the Android software application, shown in Figure 6, data from previous workouts are presented along with the current workout information (number of push-ups and squats).



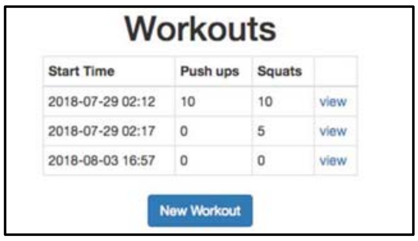
Jul 30	0	0
Jul 30	0	0
Jul 30	1	0
Sep 2	0	4
Sep 2	12	12
Sep 2	12	12

8

8

Figure 6. Workout Android Smartphone Application

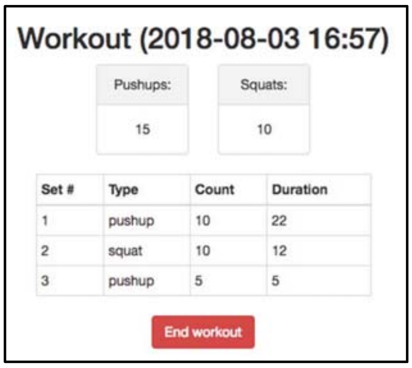
For each finalized workout, a summary which includes information about sets performed, a timeline for the workout and rest time is displayed as shown in Figure 7. The web interface also shows real-time information about the current workout sessions as shown in Figure 8. Lastly, the web interface application displays historical data of previous workouts as shown in Figure 9.



Workouts

Start Time	Push ups	Squats	
2018-07-29 02:12	10	10	view
2018-07-29 02:17	0	5	view
2018-08-03 16:57	0	0	view

Figure 7. List of Workout Histories through the Web Interface

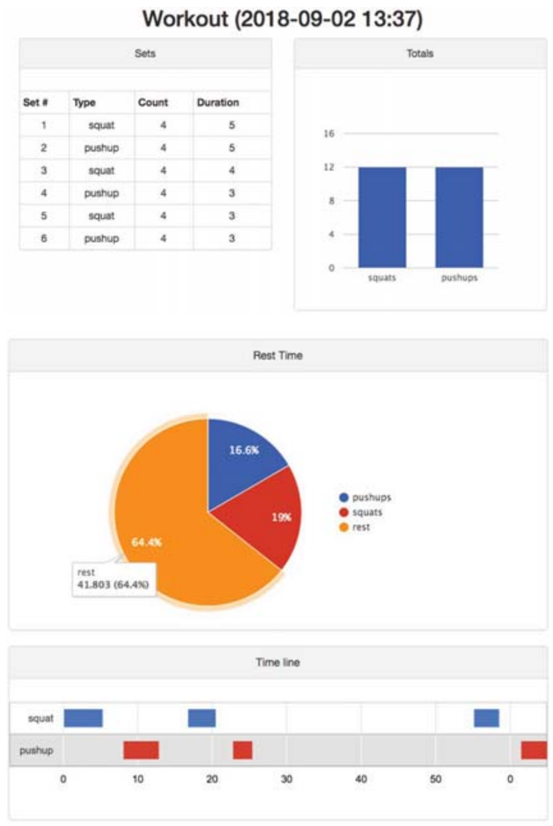


Workout (2018-08-03 16:57)

Pushups:	Squats:
15	10

Set #	Type	Count	Duration
1	pushup	10	22
2	squat	10	12
3	pushup	5	5

Figure 8. Real-time Workout Tracking through the Web Interface



Workout (2018-09-02 13:37)

Sets

Set #	Type	Count	Duration
1	squat	4	5
2	pushup	4	5
3	squat	4	4
4	pushup	4	3
5	squat	4	3
6	pushup	4	3

Figure 9. Completed Workout Session Summary through Web Interface

VI. Conclusion

In this paper, we have presented a system design for a smart workout tracking system which leverages WSBN technology and IoT architecture to provide accurate and automated tracking of workout activities. The system allows real-time tracking of workout sessions by utilizing multiple sensor nodes, which enables coaches and physicians to monitor athletes or patients during their activity. This is valuable in different scenarios such as athlete's training camps, patients going through rehabilitation programs, fitness competitions or regular user's trying to keep track of their fitness levels. Furthermore, an algorithm for detecting exercise events was implemented successfully, however, different implementations of the same architecture could use an increased number of sensors along with different analysis techniques, possibly based on machine learning and signal processing, to detect other types of movements. In addition to providing real-time workout progress, this system has the capability of managing long-term data storage to provide both the user and medical professionals with detailed information. Furthermore, our system is not limited to the devices presented in this paper, but other devices can be added to fit the user's needs.

References

[1] CrossFit, Inc., "About the Games | CrossFit Games," CrossFit, 2019. [Online]. Available: https://games.crossfit.com/about-the-games. [Accessed 26 Feb. 2019].

[2] M. Chen, S. Gonzalez, A. Vasilakos, H. Cao and V. Leung, "Body Are Networks: A Survey," *Mobile Networks & Applications,* vol. 16, no. 2, pp. 171-193, 2010.

[3] M. Weyrich and C. Ebert, "Reference Architectures for the Internet of Things," *IEEE Software,* vol. 33, no. 1, pp. 112-116, 2016.

[4] W.-J. Yi, S. Niu, T. Gonnot and J. Saniie, "System Architecture of Intelligent Personal Communication Node for Body Sensor Network," *2013 IEEE International Instrumentation and Measurement Technology Conference,* pp. 1717-1720, 2013.

[5] Analog Devices, "3-Axis Digital Accelerometer ADXL345," 2015. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/ADXL345.pdf. [Accessed 26 Feb. 2019].

[6] ELEC Freaks, "Ultrasonic Ranging Module HC-SR04," 2016. [Online]. Available: http://mouser.com/ds/2/813/HCSR04-1022824.pdf. [Accessed 26 Feb. 2019].

[7] Raspberry Pi Foundation, "Raspberry Pi 3 Model B+," 2018. [Online]. Available: https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-Model-Bplus-Product-Brief.pdf. [Accessed 26 Feb. 2019].

[8] D. Heinemeier Hannson, "Ruby on Rails | A web-application framework that includes everything needed to create database-backed web applications according to the Model-View-Controller (MVC) pattern.," 2019. [Online]. Available: https://rubyonrails.org/. [Accessed 16 Mar. 2019].

[9] W.-J. Yi, O. Sarkar, S. Mathavan and J. Saniie, "Wearable Sensor Data Fusion for Remote Health Assessment and Fall Detection," *IEEE International Conference on Electro/Information Technology,* pp. 303-307, 2014.