Analysis of Distance Sensor in Lego Mindstorm

Wasana Leithe[1], Tor-Morten Grønli[1] and Muhammad Younas[2]

[1] Mobile Technology Lab, Department of Information Technology,
Kristiania University College, Norway
[2] School of Engineering, Computing and Mathematics,
Oxford Brookes University, Oxford, United Kingdom

Abstract. Internet of Things is a concept that many physical devices can connect and share information. IoT development in mobile apps aimed to control connected devices. This paper describes the form of an application-led project by building a smart application system using the Lego Mindstorm kit. It decides on and simulates scenarios for the IoT solutions and the design and develop a proof-of-concept mobile and IoT application with emphasis on the technical implementations, architectural considerations, and interoperability. It demonstrates through graphical programming environment the configuring, implement and evaluation of distance sensor technologies in a mobile application.

Keywords: Internet of Things, mobile computing, Lego Mindstorm

1 Introduction

At present, smart network sensors open new opportunities for designing control systems. Automotive and industrial automation systems, sensors are connected to field busses for distributed control. Mobile application systems such as robots and smart vehicles take spontaneously exploit sensor information provided by an instrumented environment is becoming increasingly powerful [1]. The IoT is a network that connects uniquely identifiable "things" to the Internet. The things have sensing/actuation and potential programmability capabilities. Through the exploitation of unique identification and sensing, information about the thing can be collected. It can change the state of the thing from anywhere, anytime, by anything. There is consensus that the IoT is one of the most important revolutions in technology in decades at present. It has attracted a lot of attention from both industry and academia. [2].

Moreover, the integration between the IoT with the cloud (IoT-Cloud or sensor-cloud) has received significant interest from academia and industry. Based on mobile user location tracking, the IoT cloud plays a role as a controller, which makes schedules for physical sensor networks on-demand [3]. The IoT has many benefits that should be considered. It is expected that by 2025, the IoT nodes will connect most of the objects, and many of them are essential in our day to day life. Many people in the world have a smart mobile at present. Most of the people will be interconnected with the internet and they will be online all the time in the future. The main purpose of the IoT is to uniquely

identify, signify and access things of our day to day life anytime and anywhere through the internet, and allow them to be controlled as far as possible [4].
This paper describes the form of an application-led project by building a smart application system using the Lego Mindstorm kit. The background is described in Section II. The detail of the design and implementation includes prototype architecture and three different scenarios with state machines describes in Section III. Section IV is a result of design and implementation. Section V is a discussion about the overall research. Plus, a conclusion in Section VI. Moreover, Appendix on the latest of the paper presents the mobile dashboard with the distance sensor information of all three scenarios.

2 Background

Lego Mindstorms kit is a simple robot with a programming environment for constructing an autonomous robot like block composition [5]. There are many applications of robots, which are useful in industrial, medical, and domestic environment [6]. It is very widely to use the Lego in academic scenarios for mobile robot platforms with sensors for vision and color recognition. Its use to teach programming languages, robotics, and embedded systems. Also, its use in teaching is a good way to motivate engineering students, which is fundamental to successful teaching [7], [8]. It makes a very convenient framework for course projects that integrate mobile application systems for color recognition, line-following, obstacle detection, and vehicle interaction, among the most common cause [9].

Even though the Lego Mindstorms kit was initially designed as a toy for children over 12 years of age, its use in university courses is increasing yearly. It gets the attention of people working in many areas. These are including artificial intelligence, embedded systems, control systems, robotics, and operating systems. The Lego Mindstorms kit is inexpensive, easily reconfigurable, reprogrammable, versatile, and robust. Therefore, it is well suited for use in teaching in education scenarios [10]. The Lego modularity makes the rapid prototyping of different robot configurations easier. This easiness presents itself as extra motivation for the persons who take their first steps in the world of mobile robotics [11].

It will be able to build the robot from the Lego Mindstorm kit. Then, download the Lego Mindstorms kit coding application that can run both mobile or tablet. Lego Mindstorms kit makes it possible to build embedded systems without any prerequisite knowledge of programming language. It can solve real problems with constraints such as sensors. This application allows the programming scenario to make programs. It runs programs that control the robot's behavior, for example, moving, picking up, throwing, and seeing. It can program the robot by using the Mindstorms application via block graphical programming. It easy to scratch programming environment using blocks, which will connect blocks and the program flow from top to bottom. The application connects Lego robots via Bluetooth, which has sensors to control the robot.

Lego Mindstorms kit has four medium motors. It has two sensors that are distance sensor and color sensor. It can also touch, sound, ultrasound, and light.

3 Design and Development

Figure 1. presents the prototype architecture that application on smartphone or tablet connects hub via Bluetooth. Hub has two different sensors. The first sensor is a distance sensor that the robot can move at different distances. Another sensor is a color sensor that can distinguish colors. It will control the robot, for example, throwing a ball when a color is green

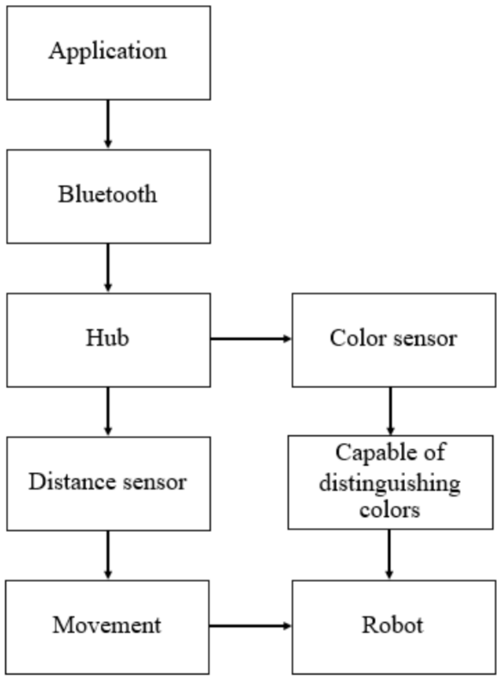

Figure 1: Prototype architecture

Figure 2 explains the detail of the Lego robot that its name Tricky. It has two different sensors, which color sensor and a distance sensor. It has three motors that are A, B, and C, which can behave as actuators. Motor A and B control movement. Motor C controls two arms. It is a two-wheeled robot that can move forward, backward, turn right and turn left. It can pick up the object, for example, a ball or something in scenario

A. Furthermore, it can play bowling in scenario B as the same components of Lego in this picture.

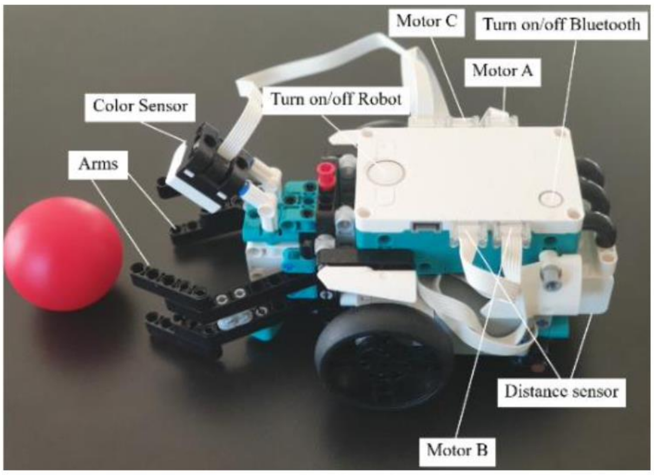

Figure 2 Components of bowling Lego robot

The ways are long 90 cm. in two different directions ways which are straight road and swing road. Each distance moves 10% and 100% of speed as the states Speed 10% and Speed 100%. That means it has four different cases in this scenario which, speed 10% at the straight road, speed 10% at the swing road, speed 100% at the straight road, and speed 100% at the swing road. Each case will try ten times. The result will calculate in percent of the robot that can lift and hold the object in the air.

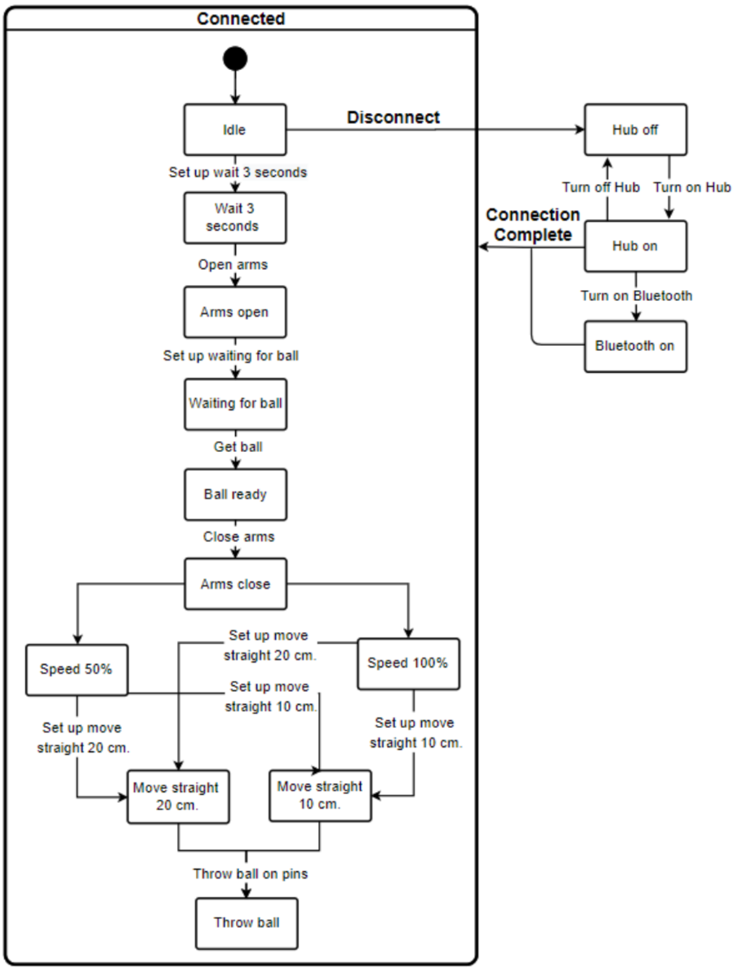

Figure 3 State machine diagram of robot playing bowling

Figure 3. presents the scenario that the robot can move and play bowling using the distance sensor. The component of the robot is the same in scenario A. It starts at the Idle state that is the be beginning of the state when the application connects to the robot as the robot will pick the object up. Also, the states, which Hub off, Hub on, Bluetooth on, and Wait 3 seconds are the same. The motor controls the robot's arm. The robot throws the ball on pins. It will stop moving forwards, reducing its movement, and then stop and exit the program. The robot will play bowling at two different distances between the robot and all pins, which are 10 cm. and 20 cm. Each distance moves 50% and 100% of speed. That means it has four various cases in this scenario, speed 50% at distance 10 cm., speed 50% at distance 20 cm., speed 100% at distance 10 cm., and speed 100% at distance 20 cm. Each case will try ten times. The result will calculate in percent of the robot that can knock down each pin.

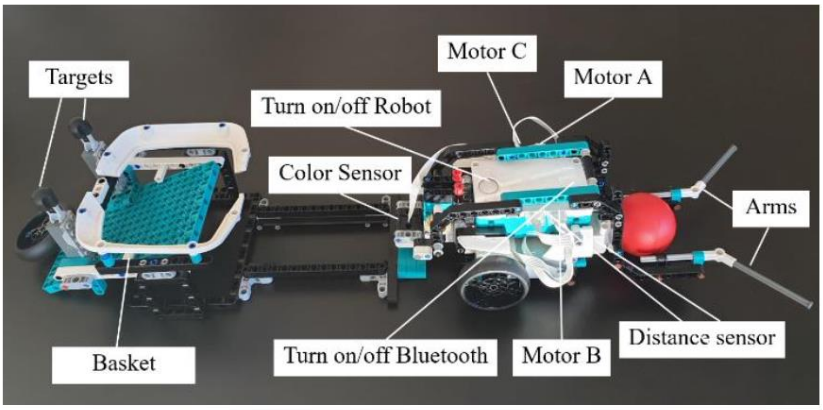

Figure 4 Components of basketball slam dunk Lego robot

Figure 4 explains the detail of the Lego robot that is almost the same figure 2. However, the distance sensor is in front of it, and the color sensor is in the back.

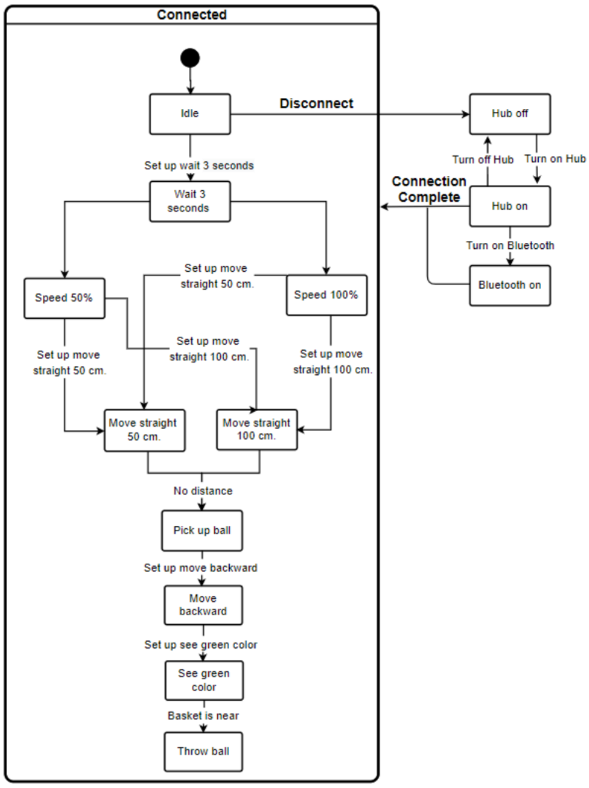

Figure 5 State machine diagram of the robot in basketball slam dunk

It has three motors as same figure 2 and the motor C controls two lang arms, which can score a slam dunk in the basketball. When it can throw the ball in the correct position, the two targets will come out. Figure 5. presents the scenario that the robot can move and play basketball using the distance sensor. It starts at the Idle state that is the be beginning of the state when the application connects to the robot as the robot will pick up the object.

4 Results

Figure 6 explains that the robot could pick up the object at speeding 10% better than 100%. It was better at straight road than swing road. Furthermore, it was the best when the robot moved slowly at the straight road. Therefore, speeding 10% at the straight road is the best case in this scenario.

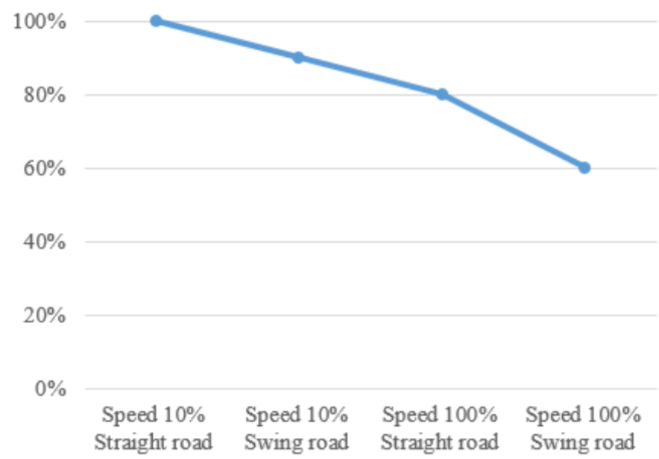

Figure 6 Percent of the robot could pick up the object

Figure 7 explains that the robot played bowling and could knock down the pins at speeding 100% better than 50%. It was better when the robot was near to the pin 10 cm. than 20 cm. Furthermore, it was best when the robot moved fast and near the pins. Therefore, speeding 100% in the distance of 10 cm. is the best case in this scenario.

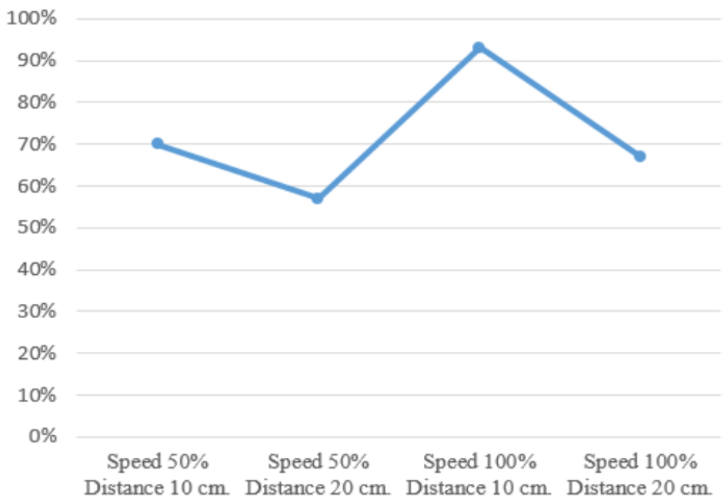

Figure 7 Percent of the robot could knock down the pins

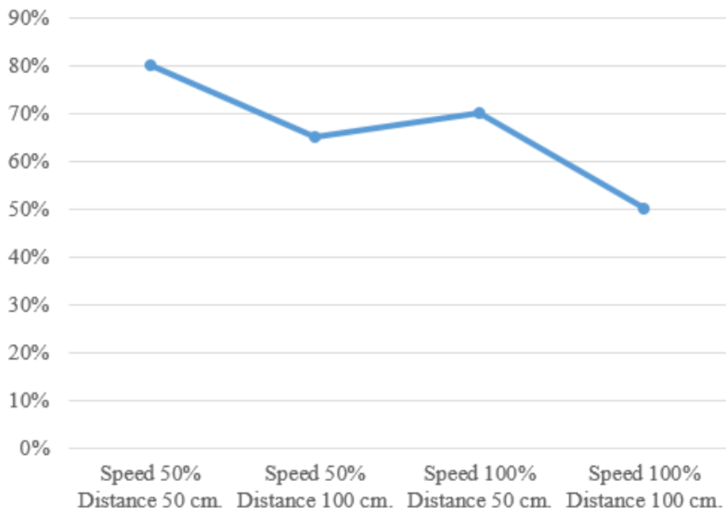

Figure 8 Percent of the robot got to score a slam dunk

Figure 8 explains that the robot got to score a slam dunk in basketball at speeding of 50% and distance from the ball 50 cm. is the best score. In contrast, 100% speeding and distance from the ball 100 cm. is the worst score.

5 Discussion

WE designed the IoT application scenarios, build the Lego Mindstorm kit system, and implement a block-based visual programming language. As the result of scenario A. It programmed the robot to move, pick up, hold the objects in the air at different speeds, and directions way. Speeding affected how the robot picked up the object. The distance sensor is used to control the robot's arms motor when picking up. Moving slower could pick up better. Swing road made it moved in the wrong direction. Both speeding and swing road can affect it. Sometimes it moved very fast as maximum speed. It moved not straight and stopped the incorrect position as it should be. Therefore, it could not pick up the object. It is always a good idea to reduce the speed for motors that require precision. If it moves too fast, it cannot pick up the objects effectively. In the real world, it is the same when driving vehicles. When driving is so fast at swing the road, maybe they can move out of the road and out of control.

In scenario B, the robot played bowling and could knock down the pins. The distance sensor in front of it controlled itself to moved forward. The distance between it and the pins was near 10 cm. and a maximum speed of 100% was the best result. It could knock down the pins very well. In contradiction, the result was not so good when speeding slower and far from the pins. Sometimes it could not knock down any pins because it moved slowly at 50% of speed. The ball moved not straight, stopped in the incorrect position as it should be. Therefore, the ball was far away from the pins. Slowing speed was not knock down all the pins occasionally because it did not have enough energy. There is a particular floor or slide that is the same in the real world when playing bowling. It is easier to knock down the pins when throwing the ball very fast with energy and close them.

In scenario C, the robot got to score a slam dunk. The color sensor in front of it controlled itself to score a slam dunk when the robot could see green color on the floor in front of the basket. The best score for this experiment was speed 50% and moved forward 50 cm. as backward. The robot moved both forward and backward slowly in the correct direction way. It could pick up the ball and come back to the basket correct position at the green color in front of the basket. Therefore, it could get more score than another case. In opposite, when speeding 100% and moved forward 100 cm. was the worst case. The robot moved too fast and could not come back to the correct position. Sometimes it moved not straight and stopped in the wrong position. Also, it was not enough energy to make the two targets jumped out from the basket. Therefore, it could not score or only one score a slam dunk. It is the same when persons throw a ball in a basket or something a target. They should stand at the appropriate position and throw a ball with suitable energy. Thus, they will get a score. Speed and distance that control by distance sensor is significant in this experiment.

Moreover, WE think about the real-world in a self-driving smart car. It can operate its own by measuring the distance of various objects beside roadsides and with other vehicles running on roads [12]. However, there are many difficulties in a self-driving field because of the dynamic environment and the fast and complex movement. Different tasks are needed in a self-driving field such as vehicle distance measurement, vehicle detection, and obstacle detection [13]. Sensor controls cars, which can drive on asphalt, sand, or another. Plus, the robots in all three scenarios in this paper moved on the tree floor. It is smoother than asphalt on the real road. Additionally, the robot's wheels are smooth plastics. Therefore, many variables make robots move in different as to need.

6 Conclusion

In this paper, WE presented an analysis of a distance sensor in the Lego Mindstorm kit. In all three scenarios, the distance sensor controlled the robot's movement. The distance and speeding affect how the robots picked up the object, played bowling, and scored a slam dunk in basketball. Moving slower could pick up the object better. Swing road made the robot moved in the wrong direction. However, when the robot played bowling, it could knock down the pins better when closed to them. In the scenario of scored a slam dunk in basketball, it lowered speeding and closed the basket to help it get more score. That means both distance and speeding could affect the robot's movement.

Additionally, in these three scenarios, blocking path encoding or the next function controls the robots about moving, action, and doing something. It identified the problem as step by step for the robots to reach its destination. It is the same as real-world programming that programmers must change their algorithms to control the robot's behavior and situations. It plans to write the coding of what happens before, present, and future. Sometimes it should use a condition, for example, if-else, switch-case. It is essential to design sequence, speed, distance that the robots can move slower or faster. WE have learned the thought process behind creating a program, programming functions, and how WE cloud to control the robots' actions or events. It used block coding programming in the mobile application. There are many different possibilities to design, build on scenarios, and code using the Lego Mindstorm kit. Therefore, this paper described and helped to understand and approaches by which mobile computing, sensors, and Internet of things.

In the future research targets mobile computing, sensors, and Internet of things. It would analyze more different types of roads. It is important where robots drive. Additionally, it can explain cases scenario with more specific detail. The result may be different.

7 References

1. T. Brade, J. Kaiser and S. Zug, "Expressing validity estimates in smart sensor applications," 26th International Conference on Architecture of Computing Systems, Prague, Czech Republic, 2013, pp. 1-8, 2013.
2. Xing Liu and O. Baiocchi, "A comparison of the definitions for smart sensors, smart objects and Things in IoT," 2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, 2016, pp. 1-4, 2016.
3. S. D. Bachpalle and M. R. Shinde, "Integration of Sensors for Location Tracking using Internet of Things," 2018 International Conference on Information , Communication, Engineering and Technology (ICICET), Pune, pp. 1-4, 2018.
4. T. Islam, S. C. Mukhopadhyay and N. K. Suryadevara, "Smart Sensors and Internet of Things: A Postgraduate Paper," in IEEE Sensors Journal, vol. 17, no. 3, pp. 577-584, 1 Feb.1, 2017.
5. H. Tominaga, Y. Onishi, T. Hayashi and T. Yamasaki, "LEGO Robot Programming Exercise Support for Problem Solving Learning with Game Strategy Planning Tools," 2007 First IEEE International Workshop on Digital Game and Intelligent Toy Enhanced Learning (DIGITEL'07), Jhongli City, pp. 81-88, 2007.
6. Pham Trung, Nitin Afzulpurkar and Dhananjay Bodhale, "Development of vision service in Robotics Studio for road signs recognition and control of LEGO MINDSTORMS ROBOT," 2008 IEEE International Conference on Robotics and Biomimetics, Bangkok, pp. 1176-1181, 2009.
7. M. Pinto, A. P. Moreira and A. Matos, "Localization of Mobile Robots Using an Extended Kalman Filter in a LEGO NXT," in IEEE Transactions on Education, vol. 55, no. 1, pp. 135-144, Feb. 2012.
8. K. Padayachee, P. M. Gouws and M. Lemmer, "Evaluating the Effectiveness of Lego Robots in Engaged Scholarship," 2015 Annual Global Online Conference on Information and Computer Technology (GOCICT), Louisville, KY, pp. 16-20, 2015.
9. Ariza, D. V., Palacio, A. M., Aragón, I. P., Pulido, C. M., Logreira, E. A., & McKinley, J. R. (2017). Application of color sensor programming with LEGO-Mindstorms NXT 2.0 to recreate a simplistic plague detection scenario. Scientia et Technica, 22(3), 268–272, 2017.
10. S. S. Prieto, T. A. Navarro, M. G. Plaza and O. R. Polo, "A Monoball Robot Based on LEGO Mindstorms [Focus on Education]," in IEEE Control Systems Magazine, vol. 32, no. 2, pp. 71-83, April 2012.
11. J. Goncales, J. Lima, P. Malheiros and P. Costa, "Realistic simulation of a Lego Mindstorms NXT based robot," 2009 IEEE Control Applications, (CCA) & Intelligent Control, (ISIC), St. Petersburg, pp. 1242-1247, 2009.
12. M. I. Mamun, A. Rahman, M. A. Khaleque, M. F. Mridha and M. A. Hamid, "Healthcare Monitoring System Inside Self-driving Smart Car in 5G Cellular Network," 2019 IEEE 17th International Conference on Industrial Informatics (INDIN), Helsinki, Finland, pp. 1515-1520, 2019.
13. A. Zaarane, I. Slimani, A. Hamdoun and I. Atouf, "Vehicle to vehicle distance measurement for self-driving systems," 2019 6th International Conference on Control, Decision and Information Technologies (CoDIT), Paris, France, pp. 1587-1591, 2019.