

DESIGN AND IMPLEMENTATION FOR IORT BASED REMOTE CONTROL ROBOT USING BLOCK-BASED PROGRAMMING

SeongYong Hong, Carson-Newman University, shong@cn.edu
YongHyun Hwang, Google Inc., freeaion@gmail.com

ABSTRACT

As the Internet of Things and robotics are emerging, it drives more attention to computer science education based on IoRT. The IoRT is a concept in which intelligent robots monitor the events around them, fuse their sensor data, exercise local and distributed intelligence for courses of actions, and manipulate and/or control objects. This paper describes the design and implementation of IoRT based remote control robot using block-based programming. In addition, a methodology for efficient IoRT project development is proposed using integrated Lego Mindstorms EV3 robot kit and block-based programming for an Android application. The prototype of the remote-control robot is based on the ultrasonic sensor and smart device. The prototype shows that the IoRT learner can enhance the capability of creativity, problem-solving, and efficiency in their thinking process.

Keywords: IoT (Internet of Things), Internet of Robotic Things (IoRT), IoRT Application, Robotics

INTRODUCTION

The Internet of Things (IoT) is one of the most important technologies in our daily life. It continues to pick up steam as more businesses realize the potential of connected devices to make them competitive. The definition of the Internet of Things states that IoT is “A dynamic global network infrastructure with self-configuring capabilities based on standard and interoperable communication protocols, where physical and virtual things have identities, physical attributes, and virtual personalities and use intelligent interfaces, and are seamlessly integrated into the information network” (Da Xu, et al. 2014). The IoT is a network of things that are connected to the internet, including IoT devices and IoT-enabled physical assets. They range from consumer devices to sensor-equipped connected technology. These items are essential drivers for customer-facing innovation, data-driven optimization, new applications, digital transformation, business models, and revenue streams across all sectors (Atzori, et al. 2010). Other IoT concept is evolving rapidly and influencing new developments in various application domains by using IoT technology, such as the Internet of Mobile Things (IoMT), Internet of Autonomous Things (IoAT), Internet of Nano Things (IoNT), and the Internet of Robotic Things (IoRT). The IoT influence represents new development and deployment challenges in different areas such as seamless platform integration, context-based cognitive network integration, new mobile sensor/actuator network paradigms, things identification and dynamic things discoverability and many others. The IoRT represents new convergence challenges and their need to be addressed, in one side the programmability and the communication of multiple heterogeneous mobile/autonomous/robotic things for cooperating, their coordination, configuration, exchange of information, security, safety and protection (Vermesan, et al. 2017). Especially, robotics and IoT technology is rapidly and widely developing, and so as computer science. One of the examples of rapid development is hybrid IT (Information Technology), which is also known as an embedded platform, with smaller size and designated purpose (Samie, 2016). People work in the office while talking with a co-worker through their smartphone or tablet, and they manage to their home inside through digital security robot cameras while working in the company. About in the decade, robotics is the intersection of science, engineering, and technology that produces machines, called robots, that substitute for human actions. Robots and IoT devices are similar in that they both rely on sensors to understand their environment, rapidly process large streams of data, and decide how to respond. The advent of new high-speed network and growing computer technology provides realistic opportunities for new IoT and robot controls and the realization of new methods of control theory. The IoT and robotics communities are coming together to create the Internet of Robotic Things. Therefore, IoRT technology has become fast growing those opportunities as learning computer information, science, and engineering majors. IoRT application development has taken on increasing importance as the fast and complexity of computer technology studies continues to grow. However, most computer technology learners or developers have difficulties for learning IoRT technology because it requires different fields of knowledge of the IoT and robotics technology. To address the the difficulites, this paper describes the design and implementation of IoRT based remote control robot using block-based programming for computer information or science major colleges. The prototype is implemented on the LEGO Mindstorms EV3

robotics kit and Android app application block programming. Furthermore, a case study with remote control robots in the computer science course project are shown.

RELATED WORKS

In recent years, a wide range of industrial IoT and robot applications have been developed and deployed. Therefore, the Internet of Things and robotics is emerging as a hot issue, which drives more attention to computer science education based on IoRT. IoRT technology is one of the elements that have strongly influenced computer science education in recent years, particularly since the IoT boom. The IoRT encompasses a huge variety of applications, ranging from smart home, smart city, smart restaurant, and smart manufacturing, etc. In this section, we research a remote-control robot technology based on the smart device environment and describe the IoRT concept, architecture, application, and technology. In addition, we review a few case studies of IoT application projects.

There are many developments that smart device controlled remote robots with network communication. Smartphone (or tablet) robots and applications are becoming increasingly prevalent in remote control robot research project. Accordingly, a growing interest in having smartphones interacting with peripheral devices such as motors, servos, and sensors led to the recent creation of a microcontroller hardware platform as an Arduino or Raspberry Pi. Arduino is an open-source platform used for constructing and programming of electronics for IoT. It can receive and send information to most devices, and even through the internet to command the specific electronic device. It uses IDE (Integrated Development Environment) software called Sketch to program the Arduino board (Badamasi, 2014). Kumar, et al. (2016) proposed IoT technology based on a smart alert system for garbage clearance by giving an alert signal to the municipal web server for instant cleaning of a dustbin with proper verification based on the level of garbage filling. According to this project, aided by the ultrasonic sensor which is interfaced with Arduino UNO to check the level of garbage filled in the dustbin and sends the alert to the municipal web server once if garbage is filled.

Brown & Hong (2019) design and suggested an automated parking system on a smart campus as an example of IoT research. In this research, they developed and experimented with a prototype system using the Arduino open-source platform and Android application programming. Shrugare, et al. (2018) described a case study of smartphone based remote control robots for domestic purposes using the HC-05 Bluetooth module and Raspberry pi 3. This research developed the Android application with MIT App Inventor designer for hardware control. Selvam's (2014) project describes and suggested developing a robotic vehicle using an android application for remote operation attached with a wireless camera for monitoring purposes. This project a new economical solution for robot control systems. The robots are controlled through wired networks and android applications programming. The robot along with the camera can wirelessly transmit real-time video with night vision capabilities. In addition, this project proposed using Bluetooth based architecture for Android communication with an articulated robot. The App Inventor for Android is an application originally provided by Google and now maintained by MIT. In the development of this app, the App Inventor provides a versatile opportunity to develop a customized application starts with establishing a Bluetooth connection by searching the available Bluetooth devices and make pair with them. It allows anyone familiar with computer programming to create software applications for the Android OS (Pokress and Veiga, 2013). It uses a graphical interface, very similar to Scratch or the Blockly block-based visual programming language, that allows users to drag-and-drop visual objects to create an application that can run on the Android system, which runs on many mobile devices (Pasternak, et al. 2017; Maloney, et al. 2010). Oros and Krichmar's (2013) surveyed and explained the new and exciting approach of smartphone-based robotics for research and education. They presented a promising trend in robotics, which leverages smartphone technology. According to this paper, smartphone robots are ideal for hobbyists, educators, students, and researchers. They also described different smartphone based robotic projects and demonstrated the relatively easy and inexpensive construction of an Android application based robotic platform.

Internet of Robotic Things

Robots and IoT devices are similar in that they both rely on sensors to understand their environment, rapidly process large streams of data, and decide how to respond. At this time, most IoT initiatives are focused on using connected devices with simple, onboard, passive sensors to manage, monitor and optimize systems and their processes. This alone will be hugely impactful; however, it is not too soon for forward-thinking companies to explore the more advanced and transformational aspects of ubiquitous connectivity to, and communication among, smart devices. IoT devices are generally designed to handle specific tasks. Robots need to react to unexpected conditions that their

developer may not have anticipated. AI and machine learning help robots deal with such situations (Roy, 2017). IoRT intelligent devices can monitor events, fuse sensor data from a variety of sources, use local and distributed “intelligence” to determine the best course of action, and then act to control or manipulate objects in the physical world, and in some cases while physically moving through that world. According to the International Federation of Robots and Loup Ventures’ new report, robotic spending will climb to \$13 billion in 2025. In addition, according to Statistics MRC, IoRT market was around \$4.37 billion in 2016 and is expected to touch \$28.03 billion by 2023 with a staggering CAGR growth of 30.4%. The compound annual growth rate for the IoRT market would be 29.7% until 2022 (Batth, et al. 2018). ABI Research defined the term “The Internet of Robotic Things in 2014” and outlined three main components that characterize IoRT devices:

- The device (robot) can be considered intelligent due to having embedded monitoring and sensing capabilities. At the same time, it can get sensor data from others which are fused together to drive the “acting” purpose of the device.
- A second intelligent aspect of the device is that it can leverage local and distributed “intelligence”. This means that it can analyze data from monitoring events and can also access analyzed data.
- These two components allow the device to autonomously determine which action of controlling or manipulating an object in the physical world it will perform. If it is designed to do so, the device can also move in the physical world. Additionally, it can also “notify”, or “alert” based on the analysis of a physical event to the actions.

Basically, IoRT is the next level of IoT in which robots utilize IoT data and technology to autonomously manipulate the physical world (Kara and Carlaw, 2014). Ray (2016) proposed an IoT based robotics architectural concept IoRT, as an advancement of current cloud networked robots. Internet of Robotic Things allows robots or robotic systems to connect, share, and disseminate the distributed computation resources, business activities, context information, and environmental data with each other, and to access novel knowledge and specialized skills not learned by themselves, all under a hood of the sophisticated architectural framework. As a result, they described the Internet of Robotic Things novel concept, technologies, and challenges to provide a better understanding of the architectural assimilation of IoRT and identify important research directions on this term. Simoens, et al. (2016) surveyed and reviewed that advance the abilities of both the current Internet of Things and the current robotic systems, thus enabling the creation of new, potentially disruptive services. Razafimandimby, et al. (2016) implemented an IoT-based and an ANN control scheme to maintain global connectivity among multiple IoRT robots. In the research, the proposed approaches tried to capture the trade-off between network coverage and communication quality expressed as RSSI level. Also, Afanasyev, et al. (2019) represented a comprehensive overview of the concepts and challenges in the IoRT and proposes the IoRT architecture. This paper explained some issues, there are since the IoRT system works on the connection between the connected devices when the connection is broken, a failure occurs that can disrupt the work and make it impossible for the robots to achieve their goals. Thus, securing communications is another open problem where network protection provides IoRT system security. IoT and Robotics technology has acquired huge changes in different aspects of human society. One of the examples, IoMRT (Internet of Medical Robotics Things) is an innovative concept and very much useful in the medical field. The IoMRT can be used to collect the patient's health data utilizing the sensors and devices connected to the internet-based health monitoring systems through online networks. IoMRT is processed by connecting and communicating the machine to machine (M to M) of medical devices equipped with Wi-Fi or Li-Fi. The received data from the IoMRT devices are stored in the cloud server database, which is linked with cloud platforms and is then analyzed (Guntur, et al. 2019).

EDUCATIONAL ROBOT PLATFORM FOR IORT

Robots are widely used in manufacturing, assembly and packing, transport, earth and space exploration, surgery, weaponry, laboratory research, and mass production of consumer and industrial goods (Lungarella, et al.2003). Robotics technology influences every aspect of our daily life. Robotics has a great potential to positively transform lives and work practices. It also raises efficiency and safety levels, which results in enhanced levels of service. Even more, robotics is set to become the driving technology underpinning a whole new generation of autonomous devices and cognitive artifacts that, through their learning capabilities, interact seamlessly with the world around them, and hence, provide the missing link between the digital and physical world. Robotics is already the key driver of competitiveness and flexibility in large scale manufacturing industries. In this IoT industries robotics already

underpins employment. Increasingly robotics is becoming more relevant for smaller manufacturing IoT industries which are central to the world's manufacturing and employment capacity (Grieco, et al.2014).Therefore, a programmable robotic platform is the best way to create motivation in IoRT learners and students to learn how to program robots and to become familiar with the full power of computer programming and coding. In this section of the paper, we review and explain a useful educational robot platform kit to computer information and science student in the college for IoRT project. Figure 1 shows and explains the educational robot hardware platform as a programmable robot kit.

Hardware Robot Platform

Pioneer 3DX research robots are the world's most popular intelligent mobile robots for education and research. Their versatility, reliability, and durability have made them the preferred platform for advanced intelligent robotics. Pioneers are pre-assembled, customizable, upgradeable, and rugged enough to last through years of laboratory and classroom use (Martínez, et al. 2014). Pioneer 3DX is a small lightweight two-wheel two-motor differential drive robot ideal for indoor laboratory or classroom use. The robot comes complete with front SONAR, one battery, wheel encoders, a microcontroller with ARCOS firmware, and the Pioneer SDK advanced mobile robotics software development package (Figure 1. (a)). The iRobot Corporation is an American technology company that designs and builds consumer robots. It was founded in 1990 by three members of MIT's Artificial Intelligence Lab who designed robots for space exploration and military defense. The iRobot Create is based on the same grounds as the robotic vacuum cleaner Roomba, so it has similar hardware equipment (Dekan, et al, 2013). iRobot Create is based on differential drive chassis with two driving wheels and two support wheels. It has many sensors that allow it to perceive its surroundings. There is one distance sensor designed to detect walls, further distance sensors to avoid cliffs and bump sensors hidden in the front bumper. On the bumper, there is also an omnidirectional IR receiver that receives information from the docking station, virtual walls, and remote control. There are also sensors of internal parameters such as battery status, current consumption, and status of buttons (Dekan, et al, 2012). In addition, it is a four-wheel differential drive chassis with two driving wheels that contain incremental encoder and sensor of lifting, one supporting swing wheel with a sensor of lifting and one supporting fixed wheel. Communication and control of the iRobot Create chassis can be conducted by multiple interfaces (Buksh, et al, 2014). The iRobot company support their products with hardware and code samples that are very simple and makes the programmer in a vivid manner familiar with the basic principles of programming and offer many pre-made functions usable in other projects. Designed for educators, developers, and high-school and college-age students, iRobot Create 2 comes with a programmable robot with a rechargeable battery, a home base, and communication cable (Figure 1. (b)). The Boe-Bot is a robot that can be used in a variety of ways including combining Microsoft Robotics Developer Studio software with the Boe-Bot to control the robot's movements. The BOE-Bot stands for the Board of Education robot. It is the trade name of a robot kit that is used in middle, high school, and college robotics classes. Boe-Bot mobile robot kit consists of a main circuit board and breadboard, a plug-in BASIC Stamp microcontroller, an aluminum chassis and two servo motors to drive the wheels (Balogh, 2008). The robot can be programmed to follow a line, solve a maze, follow light, or communicate with another robot (Ghorbel, et al. 2013). It can be adjusted to walk on six legs, sense objects, or pick up things. Boe-bots are available in various configurations and can be controlled using multiple programming languages (Figure 1. (c)).

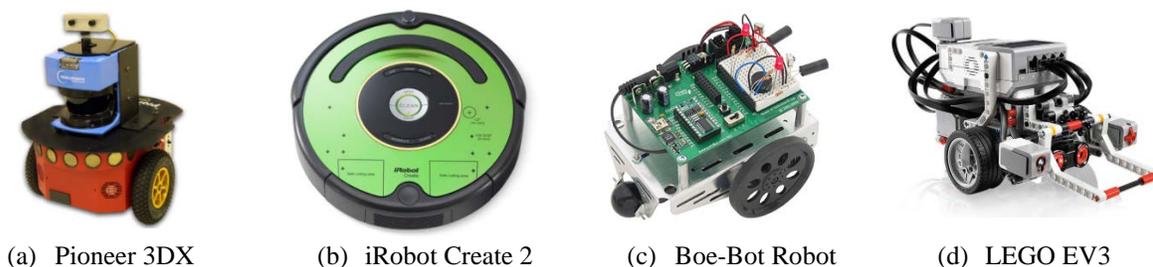


Figure 1. Educational Robot Hardware Platform

The most famous robotic kit is the LEGO Mindstorms NXT. Lego Mindstorms is a kit containing sensors, motors, and a computing module. It allows children and adults to use Lego bricks to build and program autonomous robots. Some components from the Lego Company have been part of some of the robots for a long time, though the real

passion for robotics followed the arrival of its own control system and its own sensors. The NXT is the brain of a Mindstorms robot. It is an intelligent, built-in computer-controlled (ARM 7 processor) LEGO brick that lets a Mindstorms robot come alive and perform different operations. The hardware architecture of the ARM processor and its features are available at the LEGO NXT website (Ranganathan, et al. 2008). LEGO Mindstorms EV3 is the latest generation of LEGO Mindstorms, released in 2013. In general, LEGO Mindstorms EV3 is more powerful than the LEGO Mindstorms NXT. The main difference between the NXT system (the Programmable brick, motors, and sensors) and the EV3 system is the brick itself. The EV3 brick features a more powerful processor, 4 motor ports, a micro SD card slot, a USB host port, and a Linux operating system. In addition, it has a slightly larger screen, more buttons, and it is possible to control the EV3 brick using both iOS and Android devices, as opposed to Android only for the NXT (Vokorokos, et al. 2015). The EV3 comes with an instructional manual to build 17 different robotic designs, each one with its own function and purpose. Once all the motors and sensors are in place, you can sync the robot to the EV3 mobile programming app to control it or adjust its parameters (Figure 1. (d)).

Educational robotics is pedagogically grounded in a constructivist approach to learning because it inherently incorporates traditional, individualized problem-solving with social activities. Complete educational robot starter kits can be divided into two classes. It is a versatile assembly robot such as Lego Mindstorms Ev3 kits designed around basic building blocks. Another style is a non-versatile robot such as industrial robots, household robots, robotic aircraft, and humanoid robots. Versatility is valued because it allows the possibility of morphological changes to the robot and whether it allows the possibility of expanding the hardware. Therefore, in this paper, we selected and used the Lego Mindstorms EV3 robot kit for our research prototype design and implementation.

Block-based Programming

Block-based programming can be easier to learn than traditional text-based programming languages, like Python, C++, or Java. However, block coding is far more limited than text languages, which requires more specific instructions from the programmer to complete actions. Block-based programming is often used to help computer science people learn how to create basic programs and understand programming. Nowadays, block-based programming helps learners to be introduced to the practice of programming and the field of computer science more broadly. Block-based programming refers to programming language and IDE that separates executable actions into modular portions called blocks. These blocks are generally represented with icons that can be clicked and dragged to reorder them. Editable fields, like drop-down menus, allow users to provide further input. This graphical representation of the code can demonstrate the process to new users who may be unfamiliar with programming (Hermans, et al. 2016). Figure 2 illustrates the educational block-based programming case for robot control.

MSRDS (Microsoft Robotics Developer Studio) is a Windows-based environment for robot control and simulation. Also, MSRDS provides a platform for developing applications involved with robotics. This basically helps to easily program the necessary procedures to run robot hardware, without thinking about the complexity and hassles of the hardware system – which, in turn, makes life easier for every designer who has an interest in developing robots. This software is an Integrated Development Environment (IDE) developed by Microsoft for the development of robotics projects. It is based partly on CCR (Concurrency and Coordination Runtime), a .NET framework for managing concurrent parallel tasks. It includes features like a visual programming tool, Microsoft Visual Programming Language for creating and debugging robot applications, web-based and windows-based interfaces, 3D simulation, easy access to a robot's sensors and actuators (Yusof, et al. 2012). Microsoft VPL (Visual Programming Language) is service-oriented, uses workflow-based visual programming, and has strong support for parallel computing (Figure 2. (a)). The MSRDS software is providing off-the-shelf services for working with simulated and real hardware as Mobile Robots, Aldebaran, CoroWare, Fishertechnik, iRobot, Kuka, Lego, Parallax, RoboRealm, Segway, and more than 50 (Cepeda, et al. 2010).

The EV3 brick is programmed with new programming software. Although the software is different from the NXT programming software, existing NXT users will recognize many of the features of the EV3 software. We can still program the robot's actions using drag and drop programming blocks, as shown in Figure 2, (b). The Lego brick is the part that controls all the parts of the robot. This part includes slots for USB, SD card and charger and eight ports to connect the engines and sensors. In addition, a small-size screen and the buttons allow controlling it when not connected to a computer (Kunduracioğlu, 2018). The software provided by Lego has quite an easy and user-friendly interface. Though it is easy to use, it can be used in many areas for various purposes. If the number of parts is sufficient,

a robot that can serve multi-functions can be developed. In addition, programs were developed to examine the working principles of all sensors in detail (Chaudhary, et al. 2016).

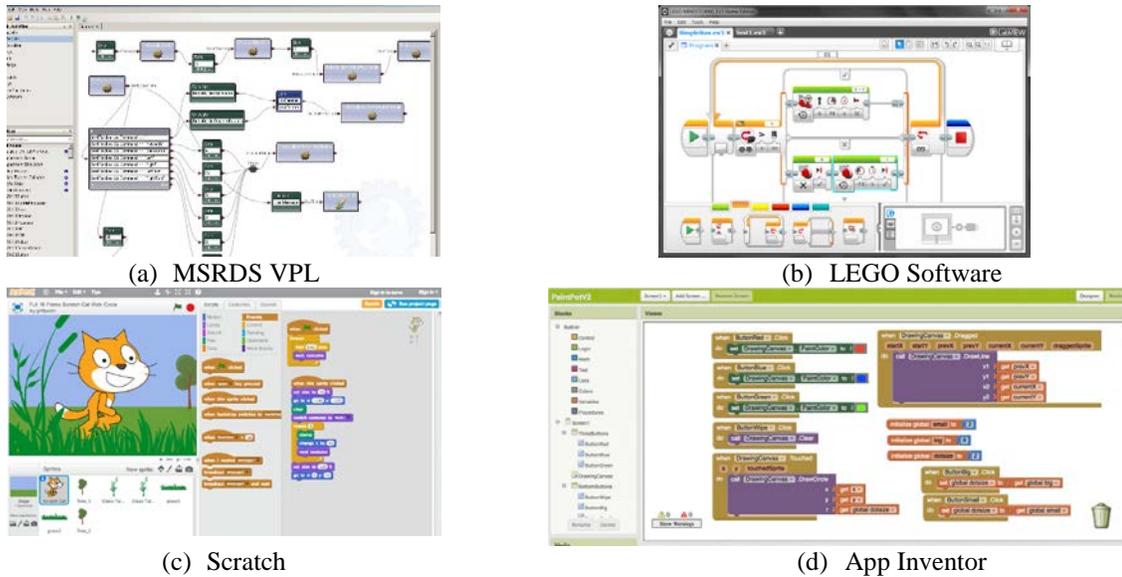


Figure 2. Educational Block-based Programming IDE

Scratch is a free programming language developed by MIT that makes it easy to create interactive stories, animations, games, music, and art, and share your creations on the web. Scratch can run from within a modern web browser or downloaded as an app. It is a block-based programming language and an online community where children can program and share interactive media such as stories, games, and animation with people from all over the world. Scratch is a block-based visual programming language and website targeted primarily at children or computer person of no major. Users of the Scratch site can create online projects using a block-like interface. The goal of Scratch was to create a programming environment with sufficient scaffolds for novices to start to program with little or no formal instruction while also being able to support sophisticated programs. Scratch allows students to learn and use all the essential elements of coding and computer science (Yünkül, et al. 2017). From creating variables to building functions, students snap drag-and-drop blocks of code together to create programs for animation, digital storytelling, art, math, etc., In addition, scratch is a block-based imperative, event-driven, dynamically typed, and interpreted programming language. As shown in Figure 2 (c), the Scratch screen is divided into three sections: the stage on the right side (where you see the results of your code in action), the workspace in the center (where you put the code together), and the blocks palette on the left (where you find all the code blocks). Students code the actions of multiple sprites (the different characters) or screen elements and can also add sounds, images, and textual elements to build almost anything (Maloney, et al. 2010).

MIT App Inventor is a web application integrated development environment originally provided by Google company and now maintained by MIT. App Inventor is a visual, drag-and-drop tool for building mobile apps on the Android platform (Wolber, 2011). We design the user interface (the visual appearance) of an app using a web-based GUI (Graphical User Interface) builder, then we specify the app's behavior by piecing together "blocks" as if we were working on a puzzle. App Inventor blocks-based coding programs inspire intellectual and creative empowerment. Figure 2 (d) shows an example of the App Inventor blocks-based coding programs GUI environment. One of the characteristics of working with MIT App Inventor is that developers can see their creations while they are building them. This allows users to develop their apps incrementally and encourages them to test as they build. Whenever the user drags a new component to the designer or creates new functionality through blocks, those new artifacts are automatically and readily available in the connected device or emulator. Therefore, App Inventor is used by students, teachers, developers, hobbyists, and entrepreneurs to develop apps for collaboration, productivity, personal use, recreation, learning, social good, and community activism (Gray, et al. 2012). Particularly, we can program this robot that combines a LEGO Mindstorms NXT or EV3 and an Android phone. Our programs will be Android applications that control the robot using the Bluetooth capabilities of the phone. We will be able to use the sensors that are on the

NXT or EV3 (ultrasonic, touch, and light) as well as the phone's features (accelerometer, orientation sensor, speech recognition, speech synthesis, playing sounds). Papadakis & Orfanakis (2016) presented a combined approach to utilizing educational robotics and visual programming through Lego Mindstorms and App Inventor for teaching and understanding of basic programming structures. According to this paper, students through their involvement seem to have understood the basic concepts of programming and technology as they engaged in practical work in an interdisciplinary authentic environment. Roy (2012) explained an experience report about using AIA (App Inventor for Android) in summer camps. They provided a detailed description of designing our camps with AIA including the process of selecting and setting up an Android device and instructional materials that students developed and made available to everyone. Also, they evaluated our camps through surveys to determine the effects on the students. According to research, they found that there was mostly a slight increase in the favorable disposition towards computing. In this paper, we decided to design and programming the MIT App Inventor for our research prototype IoRT application.

DESIGN AND IMPLEMENTATION OF IORT

The paper implements a remote-control robot with educational LEGO Mindstorms EV3 robot kit using App Inventor block-based programming. App Inventor consists of the Designer and the Blocks Editor. Figure 3 shows App Inventor based IoRT environment for the design and implementation for Android devices using LEGO Mindstorms EV3.

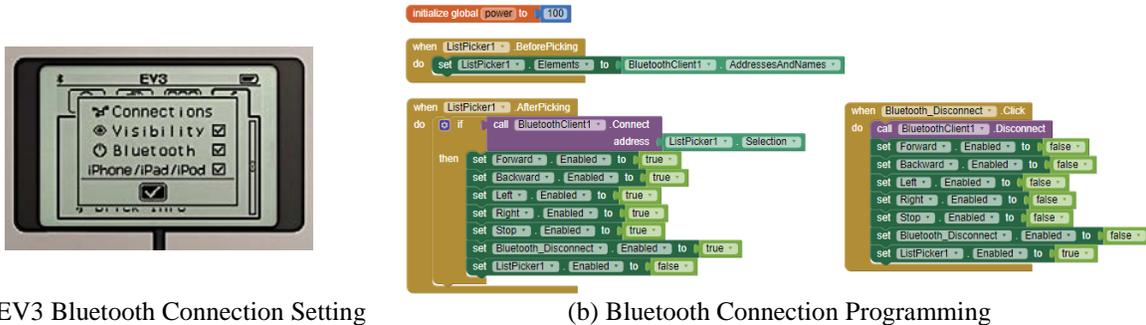


Figure 3. App Inventor based IoRT Develop Environment

We can remote control robots by smart devices, and store robot's sensor data to cloud storage. The sensor data is analyzed for the decision making of intelligent robot control. The data are also used in the predictive analytics of some robots. For example, if there are some obstacles in front of the robot, the robot must automatically stop and send some messages to the user. After the design and implementing IoRT project-based App Inventor application, we can install APK (Android Application Package) files to smart devices. The APK file is the file format to distribute and install application software onto Google's Android operating system. Using APK files enables you to download and use apps before their official release. Once we download the APK file to our computer, we can share it by sending it as an attachment to an email or putting it in an accessible place on the web. When our smartphone downloads this APK file, it will appear on the phone with an Android icon. When we click the icon, the app will download it to the smartphone. We can then install it just like any app you download from the Android Market.

Wireless projects using Bluetooth or Wi-Fi technology is very appealing to IoRT learner, which makes them engaged better. Bluetooth is a wireless communication standard that allows electronic devices to connect and interact with each other. The App Inventor designer supports the *BluetoothClient* component that should be used for communication. To make the Bluetooth connection, we need the unique Bluetooth address of the LEGO EV3 robot. Bluetooth addresses

consist of eight two-digit hexadecimal numbers (a way of representing binary values) separated by colons. Figure 4 (a) shows the LEGO EV3 Bluetooth settings for communication with smart devices. The app will automatically target the EV3 Brick we just connected if this EV3 Brick is the only one connected to the smart device. Figure 4 (b) shows the block-based programming part for connecting EV3 robot. The property tells which *BluetoothClient* component to use for communication with the robot. We need to explicitly add a *BluetoothClient* component to your project. If you have one robot, you should have one *BluetoothClient* component. These components provide control of LEGO Mindstorms EV3 robots using Bluetooth. In *ListPickerConnect.BeforePicking* event, we must set its element to *BluetoothClient1.AddressAndNames*. After that we can connect (*BluetoothClient1.Connect*) our paired EV3 in *ListPickerConnect.AfterPicking* event. If connect successfully, *Listpicker* will be set to unable, but the other button will be set to enabled for the user to operate.



(a) EV3 Bluetooth Connection Setting

(b) Bluetooth Connection Programming

Figure 4. EV3 Bluetooth Connection

The prototype of a sensor robot

For the robot hardware design, a totally personal method should be applied based on one's needs. The robot in Figure 5, which is also found in the basic set guideline, was developed for the use of all the sensors and motors together with the parts in the Lego EV3 Basic Set. It does not have a specific shape, and it should be customized totally in line with the purpose. Therefore, before the designing process, it is necessary to determine for what purpose the robot will be used.

The EV3 Ultrasonic Sensor works on the principle of SONAR and RADAR system which is used to determine the distance to an object. An ultrasonic sensor generates the high-frequency sound (ultrasound) waves. When this ultrasound hits the object, it reflects as echo which is sensed by the receiver as shown in Figure5 (a). By measuring the time required for the echo to reach the receiver, we can calculate the distance. This is the basic working principle of the Ultrasonic module to measure distance. The digital EV3 Ultrasonic Sensor generates sound waves and reads their echoes to detect and measure the distance from objects. It can also send single sound waves to work as sonar or listen for a sound wave that triggers the start of a program. Students could design a traffic-monitoring system and measure distances between vehicles, for instance. There is an opportunity to discover how technology is used in everyday items such as automatic doors, cars, and manufacturing systems. The ultrasonic sensor measures the distance to an object up to a maximum of 255cm (or 100 inches) away. It does this by sending out high-frequency sound waves that bounce off any object in range and measuring how long it takes the sound to return to the sensor. In the software, you can select whether the distance is given in centimeters or inches. The ultrasonic sensor also has a "listen-only" mode that can detect whether another robot is using an ultrasonic sensor nearby. In this mode, the sensor listens for signals but does not send them.

Robot ultrasonic sensors use a concept of thresholds to determine "Close distance" and "Far distance" as show in Figure 5 (b). The threshold is used to determine at which point the robot should be performing a different behavior. Thresholds are values that set a cutoff in a range of values so that even if there are many possibilities, the value eventually falls above the threshold, or below the threshold. A threshold is the cross-over point where the program thinks the robot is either close or far away from an object.

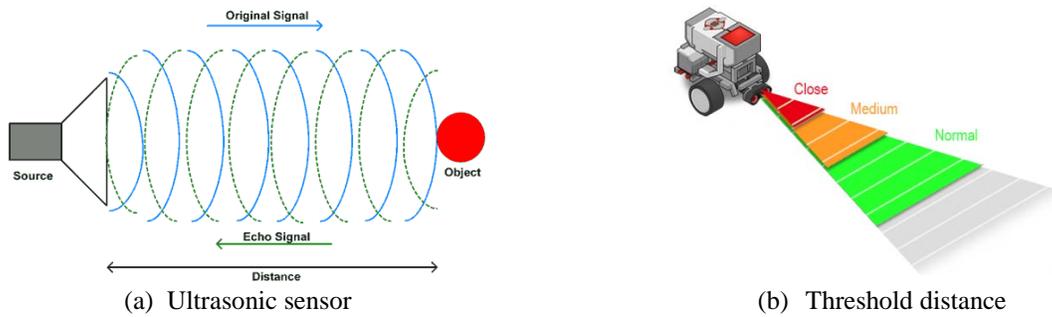


Figure 5. EV3 ultrasonic sensor

EXAMPLE OF PROJECT BASED ON IORT

In this section, we explain the experiment of the IoRT based remote control robot using block-based programming presented in this paper. The IoRT project can be integrated into lesson plans and activities in the classroom at all levels. This paper presents an example of IoRT project in teaching to the course named "Special Topic - Robotics Programming" to sophomore, junior, and senior computer science students at the computer science department. The main goal was to motivate students to understand and apply IoRT concept and methodology to improve their implementation ability of IoRT project. Obstacle detection is one of the key problems in areas such as driving assistance and mobile robot navigation by using distance measurement sensors. Therefore, Table 1 explains IoRT project method using the ultrasonic sensor obstacles auto-detection for auto control of robot speed and visualization. It can be combining obstacle avoidance, path planning, or self-speed control for intelligent robot.

Based on the results of our study, we propose a five-step prototype development IoRT based on remote control robot using App Inventor visual programming. The benefit of IoRT education is to help student learn very fast and effectively understand Internet of Things and robotics together. First, to define IoRT project goal concept, we discussed creative thinking and ideas based on IoT and robotics. For example, when your robot driving, some obstacles appear in front of your robot.

EV3 robot detects different distance and displays different colors. If your robot approach too close to obstacles, it could be displayed red color and stop. In the second step, we define and explain the algorithm for a remote-control robot motor power and sensor data. Designing and representing control algorithms is challenging in robotics, where the remote-control robot performance depends on interactions between robots and with their environment. Therefore, we need consideration of robot hardware platforms and sensors. In the third step, we design consider the App program functions as a connection robot method, TTS (Text to Speech) service, object detection, and measuring distance, etc. Also, we need to make functions like automatically change button color and motor speed. In the fourth step, we design a user interface for a remote-control robot with sensors. Users can remotely control the robot with buttons or touchpad. In addition, users can access or read sensor data of the robot. It helps users control the robot because sensor data could be analyzed for decision making. In the last step, we develop the App Inventor block-based program based on defined algorithm in step two.

Table 1. An example of IoRT Project

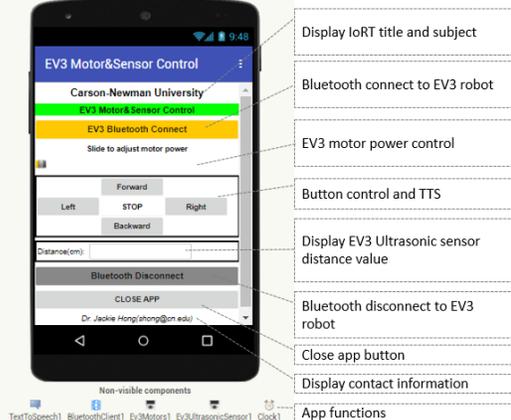
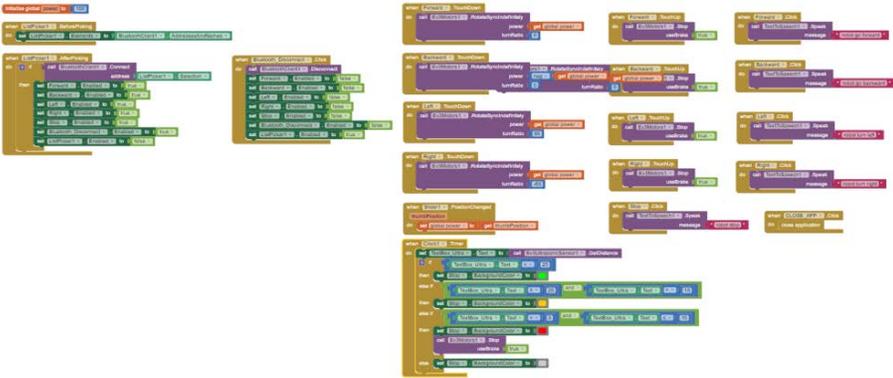
Project Subject	Remote Control Smart Robot
STEP 1. Project Goal	<ul style="list-style-type: none"> • Understanding IoRT (IoT + Robotics) concept • Remote control of robot wheel power and direction • Detection obstacles by an ultrasonic sensor • Measuring distance from the robot to obstacles
STEP 2. Algorithm	<p>If distance > 45 <i>display green color and 100% robot speed</i> Else if distance > 15 and distance < 45 <i>display yellow color and reduce 50% robot speed</i> Else if distance < 15 <i>display red color and reduce 100% robot speed</i></p>
STEP 3. Functions	<ul style="list-style-type: none"> • Connection Bluetooth to EV3 robot • TTS (Text to Speech) Service • Object detection and measuring distance by an ultrasonic sensor • Automatically change button color and motor speed
STEP 4. App Design	 <p>The screenshot shows the app interface on a smartphone. Callouts point to the following elements:</p> <ul style="list-style-type: none"> Display IoRT title and subject Bluetooth connect to EV3 robot EV3 motor power control Button control and TTS Display EV3 Ultrasonic sensor distance value Bluetooth disconnect to EV3 robot Close app button Display contact information App functions
STEP 5. Block-based Programming	 <p>The screenshot shows the block-based programming code for the app, organized into several sections:</p> <ul style="list-style-type: none"> EV3 Motor Control: Contains blocks for setting motor power and direction. EV3 Ultrasonic Sensor: Contains blocks for measuring distance and triggering actions based on distance thresholds. Bluetooth: Contains blocks for connecting to and disconnecting from the EV3 robot. Text to Speech (TTS): Contains blocks for speaking text. UI Control: Contains blocks for changing button colors and other UI elements.

Figure 6 (a) shows the App interface, if the obstacle is enough distance (estimate over 45 cm) to the robot, it displays a green color in the center button. But, if the obstacle is the near distance (estimate over 15~45 cm), it displays an orange color in the center button and automatically reduces robot speed as shown in figure 6 (b). If the obstacle is very close (estimate less than 15 cm) to the robot, it displays a red color in the center button and then automatically stops as shown in figure 6 (c).

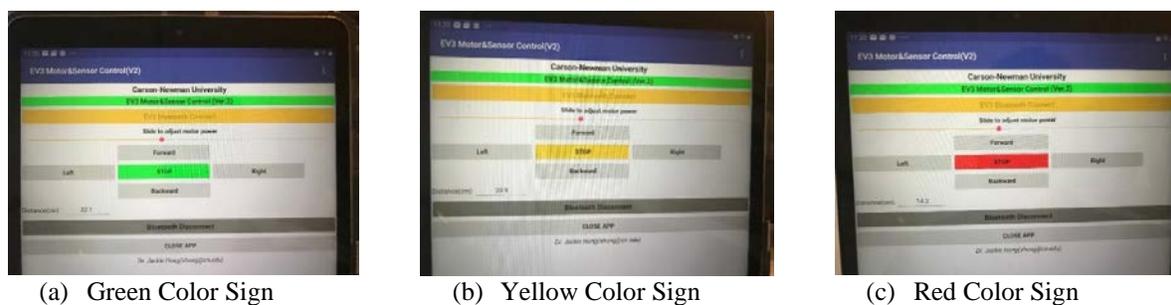


Figure 6. Result of IoRT App Project

CONCLUSIONS

The IoRT (Internet of Robotic Things) is an emerging paradigm that brings together autonomous robotic systems with the IoT vision of connected sensors and smart objects pervasively embedded in everyday environments. This merger can enable novel applications in almost every sector where cooperation between robots and IoT technology can be imagined. This paper explained the design and implementation of IoRT based remote control robot using block-based programming. It also proposes a method of efficiently developing IoRT projects using integrated Lego Mindstorms EV3 robot kit and App Inventor block-based programming for an Android application. The prototype of the remote-control robot is based on the Ultrasonic Sensor and smart device. For the validation and case study, teaching method case studies are suggested for an integrated IoRT project. The next step for our research is to investigate learner's IoRT learning satisfaction result and find some improvement for the IoRT development methodology. Furthermore, our research plans to apply IoRT data analysis based on cloud storage.

ACKNOWLEDGMENTS

We gratefully thank the Carson-Newman Mathematics & Computer Science Department Professor of Henry Suters for research and experimental equipment support.

REFERENCES

- Afanasyev, I., Mazzara, M., Chakraborty, S., Zhuchkov, N., Maksatbek, A., Yesildirek, A., & Distefano, S. (2019, October). Towards the internet of robotic things: Analysis, architecture, components and challenges. In 2019 12th International Conference on Developments in eSystems Engineering (DeSE) (pp. 3-8). IEEE.
- Atzori, L., Iera, A., & Morabito, G. (2010). The internet of things: A survey. *Computer networks*, 54(15), 2787-2805.
- Badamasi, Y. A. (2014, September). The working principle of an Arduino. In 2014 11th international conference on electronics, computer and computation (ICECCO) (pp. 1-4). IEEE.
- Batth, R. S., Nayyar, A., & Nagpal, A. (2018, August). Internet of robotic things: Driving intelligent robotics of future-concept, architecture, applications and technologies. In 2018 4th International Conference on Computing Sciences (ICCS) (pp. 151-160). IEEE.
- Balogh, R. (2008, March). Basic activities with the Boe-Bot mobile robot. In Proceedings of conference DidInfo.
- Brown, D., & Hong, S. (2019). A DESIGN OF AN AUTOMATED PARKING SYSTEM ON A SMART CAMPUS. *Issues in Information Systems*, 20(2).
- Buksh, R., Routh, S., Mitra, P., Banik, S., Mallik, A., & Gupta, S. D. (2014). Implementation of MATLAB based object detection technique on Arduino Board and iROBOT CREATE. *International Journal of Scientific and Research Publications*, 4(1), 1-5.

- Chaudhary, V., Agrawal, V., & Sureka, A. (2016). An experimental study on the learning outcome of teaching elementary level children using Lego mindstorms EV3 robotics education kit. arXiv preprint arXiv:1610.09610.
- Cepeda, J. S., Chaimowicz, L., & Soto, R. (2010, October). Exploring Microsoft Robotics Studio as a mechanism for service-oriented robotics. In 2010 Latin American Robotics Symposium and Intelligent Robotics Meeting (pp. 7-12). IEEE.
- Da Xu, L., He, W., & Li, S. (2014). Internet of things in industries: A survey. *IEEE Transactions on industrial informatics*, 10(4), 2233-2243.
- Dekan, M., Chovanec, L., Babinec, A., & Vitko, A. (2012). New modules for the iRobot Create platform. *Procedia Engineering*, 48, 65-72.
- Dekan, M., Duchoň, F., Jurišica, L., Vitko, A., & Babinec, A. (2013). iRobot create used in education. *Journal of Mechanics Engineering and Automation*, 3(4), 197-202.
- Ghorbel, M., Amouri, L., & Hie, C. A. (2013). Path Planning and Obstacle Avoidance for Boe Bot Mobile Robot. *Journal of Electronic Systems Volume*, 3(1), 11.
- Guntur, S. R., Gorrepati, R. R., & Dirisala, V. R. (2019). Robotics in Healthcare: An Internet of Medical Robotic Things (IoMRT) Perspective. In *Machine Learning in Bio-Signal Analysis and Diagnostic Imaging* (pp. 293-318). Academic Press.
- Grieco, L. A., Rizzo, A., Colucci, S., Sicari, S., Piro, G., Di Paola, D., & Boggia, G. (2014). IoT-aided robotics applications: Technological implications, target domains and open issues. *Computer Communications*, 54, 32-47.
- Gray, J., Abelson, H., Wolber, D., & Friend, M. (2012, March). Teaching CS principles with app inventor. In *Proceedings of the 50th Annual Southeast Regional Conference* (pp. 405-406).
- Hermans, F., Stolee, K. T., & Hoepelman, D. (2016, September). Smells in block-based programming languages. In *2016 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)* (pp. 68-72). IEEE.
- Kara D and Carlaw S. (2014). The Internet of Robotic Things. Technical Report, ABI Research
- Kunduracioğlu, İ. (2018). Examining the interface of lego mindstorms ev3 robot programming. *Journal of Educational Technology and Online Learning*, 1(1), 28.
- Kumar, N. S., Vuayalakshmi, B., Prarthana, R. J., & Shankar, A. (2016, November). IOT based smart garbage alert system using Arduino UNO. In *2016 IEEE Region 10 Conference (TENCON)* (pp. 1028-1034). IEEE.
- Lungarella, M., Metta, G., Pfeifer, R., & Sandini, G. (2003). Developmental robotics: a survey. *Connection science*, 15(4), 151-190.
- Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The scratch programming language and environment. *ACM Transactions on Computing Education (TOCE)*, 10(4), 1-15.
- Martínez, S. I., Rocha, J. A. C., Menchaca, J. L., Berrones, M. G. T., Obando, J. G., Cobos, J. P., & Rocha, E. C. (2014). An Autonomous Navigation Methodology for a Pioneer 3DX Robot. *Computer Technology and Application*, 5(2).
- Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The scratch programming language and environment. *ACM Transactions on Computing Education (TOCE)*, 10(4), 1-15.

- Oros, N., & Krichmar, J. L. (2013). Smartphone based robotics: Powerful, flexible and inexpensive robots for hobbyists, educators, students and researchers. *IEEE Robotics & Automation Magazine*, 1, 3.
- Pasternak, E., Fenichel, R., & Marshall, A. N. (2017, October). Tips for creating a block language with blockly. In *2017 IEEE Blocks and Beyond Workshop (B&B)* (pp. 21-24). IEEE.
- Papadakis, S., & Orfanakis, V. (2016, November). The combined use of Lego Mindstorms NXT and App Inventor for teaching novice programmers. In *International Conference EduRobotics 2016* (pp. 193-204). Springer, Cham.
- Pokress, S. C., & Veiga, J. J. D. (2013). MIT App Inventor: Enabling personal mobile computing. *arXiv preprint arXiv:1310.2830*.
- Ray, P. P. (2016). Internet of robotic things: Concept, technologies, and challenges. *IEEE Access*, 4, 9489-9500.
- Razafimandimby, C., Loscri, V., & Vegni, A. M. (2016, April). A neural network and IoT based scheme for performance assessment in internet of robotic things. In *2016 IEEE first international conference on internet-of-things design and implementation (IoTDI)* (pp. 241-246). IEEE.
- Ranganathan, P., Schultz, R., & Mardani, M. (2008, October). Use of LEGO NXT Mindstorms brick in engineering education. In *Proceedings of the 2008 ASEE North Midwest Sectional Conference* (pp. 17-19).
- Roy Chowdhury, A. (2017). IoT and Robotics: a synergy. *PeerJ Preprints*, 5, e2760v1.
- Roy, K. (2012, February). App inventor for android: report from a summer camp. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education* (pp. 283-288).
- Samie, F., Bauer, L., & Henkel, J. (2016, October). IoT technologies for embedded computing: A survey. In *2016 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ ISSS)* (pp. 1-10). IEEE.
- Simoens, P., Dragone, M., & Saffiotti, A. (2018). The Internet of Robotic Things: A review of concept, added value and applications. *International Journal of Advanced Robotic Systems*, 15(1), 1729881418759424.
- Selvam, M. (2014). Smart phone based robotic control for surveillance applications. *International Journal of research in Engineering and Technology*, 3(3), 229-232.
- Shrungare, P. S., Bokde, A. A., Kashte, N. U., & Raut, S. S. (2018). Smart Phone Based Robot for Domestic purpose using Bluetooth. *Sensors*, 115200(230400), 460800.
- Vermesan, O., Bröring, A., Tragos, E., Serrano, M., Bacciu, D., Chessa, S., ... & Simoens, P. (2017). Internet of robotic things: converging sensing/actuating, hypoconnectivity, artificial intelligence and IoT Platforms.
- Vokorokos, L., Mihalov, J., & Chovancová, E. (2015). Potential of LEGO® EV3 mobile robots. *Acta Electrotechnica et Informatica*, 15(2), 31-34.
- Wolber, D. (2011, March). App inventor and real-world motivation. In *Proceedings of the 42nd ACM technical symposium on Computer science education* (pp. 601-606).
- Yünkül, E., Durak, G., Çankaya, S., & Abidin, Z. (2017). The effects of scratch software on students' computational thinking skills. *Necatibey Eğitim Fakültesi Elektronik Fen ve Matematik Eğitimi Dergisi (EFMED)*, 11 (2), 502-517.
- Yusof, Y., Hassan, M., Saroni, N. M., & Azizan, W. C. W. (2012). Development of an Educational Virtual Mobile Robot Simulation. In *Proc. World Congr. Eng. 2012-WCE 2012 (Vol. 2)*.