Mobile Based Robotic Wireless Path Controller

Muna Dahir, Alyaa Obaid, Aisha Ali, Amna Mohammed,
Ali Abou-ElNour, and Mohammed Tarique

Dept. of Electrical Engineering,

Ajman University of Science and Technology

P.O. Box 2202, Fujairah, UAE

E-mail: m.tarique@ajman.ac.ae

Received: January 27, 2016 Accepted: June 19, 2016 Published: June 30, 2016

DOI: 10.5296/npa.v8i2.8947 URL: http://dx.doi.org/10.5296/npa.v8i2.8947

Abstract

For the past decade robotic systems have been successfully used to perform specific tasks with various degree of intelligence. Now-a-days, commercial robotics solutions are focusing more on personal services. Personal services robots are designed with necessary mobility and functional capabilities so that they can operate in a wide range of environment and can provide variety of services including health care, educational, domestic chore, entertainment, and rehabilitation. These personal robots perform various tasks and duties that are often unpredictable in nature. But, there are some technological challenges that need yet to be resolved. Integrating a robotic system with its components is usually a very complex task. Developing a modular robotic system that can perform well in various environments is also challenging. In addition, robots should be provided with necessary skills so that they can perform various tasks. Keeping all these challenges in mind we present a networked robotic system in this paper. The system is controlled by a smart phone. The path followed by the robot can be monitored via an LED panel. In order to control the operation an Android OS mobile application has been developed. Through this application a user can set a path for the robot to follow or can choose a path from a set of already defined paths. We tested our system in a controlled environment and the test results show that the robot can follow a path (either user defined or chosen) with a high accuracy.

Keywords: Wireless, Mobile, Path Controller, Bluetooth, BlueSMiRF, Microcontroller, Robot

1. Introduction

With the advent of wide spread use of wireless technology mobile systems are now being commonly used in industrial automation. Simple monotonous mechanical movements are being automated based on mobile systems. Numerous applications of industrial automation can be found in army [1-4], oil and gas industry [5-7], and automobile industry. A recent report [8] shows that the automobile industry is playing the leading role in employing robotics (see Fig.1). Now-a-days, robotics also finds applications in surveillance system. The World is under the threat of terrorism like never before. To counteract terrorism 24/7 automated surveillance systems have been initiated for public safety. The automated surveillance system has drawn considerable attentions in many applications where human surveillance systems are almost impossible to deploy and operate. Surveillance systems for monitoring gas and oil pipelines are examples of such applications. The gas and oil are distributed in the pipelines under extremely hot and high pressure conditions. In addition, safety and accuracy are major issues in gas and oil pipelines. Only small robots can fit inside the pipelines and can move around in order to monitor the status of the pipelines.

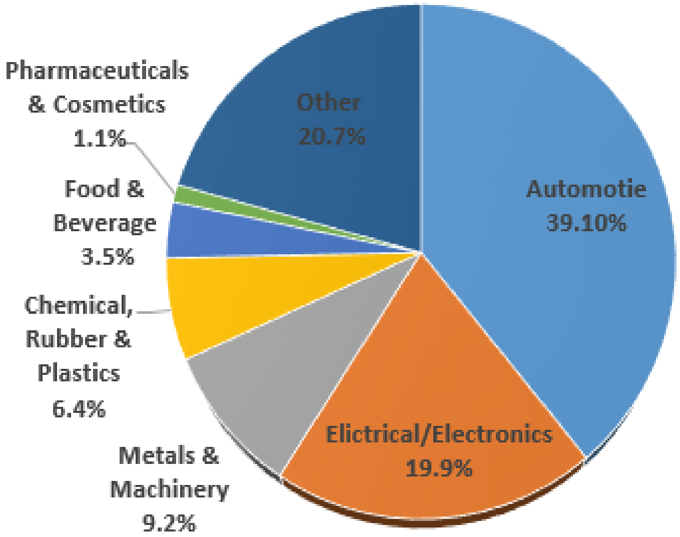


Fig. 1. Applications of Robotics in the Industry

In the automobile industry auto parts are delivered fast from the inventories to the manufacturing line where they are assembled to produce cars. Moving automobile parts might be considered dangerous at times. A high accuracy in the alignment of automobile parts is a must in automobile industry. The whole assembly line activities can be possibly done by laborers alone. But, it would be much more convenient in terms of accuracy, speed, and safety if the same can be done by robots. Using the modern wireless technology machinery can be operated and controlled from a safe distance by using a mobile device. An operator can control a machine to move object from one point to another point by sending commands to the machine by using a computer or a mobile device.

In this paper we present a mobile based path controller system based on a smart mobile phone. The proposed system can operate and control a robot, which is able (a) to follow a path defined by a user, and (b) to move objects from one place to another. Since the system is designed based on wireless technology one can operate the robot from a remote location. This robot would respond to commands set by a user. These commands would move the robot right, left, forward, and backwards. The user would be able to set a path and to command the robots to follow that particular path. The user can also save the path so that he/she would be able to use it rather than redefining the path again and again. The proposed system is user friendly. We have designed a Graphical User Interface (GUI) by using mobile app. The user would be able to control the robot through the application. In order to send the commands from the mobile app to the controlled device we use Bluetooth technology equipped with Arduino microcontroller. The rest of the paper is organized as follows. Section 2 presents some works that are directly related to our work. Section 3 presents the system model. We have presented the hardware and the software used in our work in section 4 and section 5 respectively. Section 6 concludes this paper.

2. Related Work

In this section we present some works that are directly related to our works. We compare our works with other related works. We also discuss the novelty of our work in this section.

The idea of networked robot originated from the concept of telerobots, which has been widely used in many applications including exploring undersea, outer space, and hazardous areas. One of the major limitations of the telerobots is their limited accessibility. Only trained and expert users can access the telerobots through a dedicated communication channel. But, telerobots have evolved with the advancements of internet and networking technology. The telerobots have become networked robots now. Several hundred networked robots have already been developed by the year 2012 [9]. Many researches and innovations have been done and are being done in this field. Even networked robots have been put online for public use [10]. Now-a-days, networked robots have expanded their scope beyond simple exploration. Instead, networked robots can now integrate human, agents, sensors, databases, and not the least cloud computing. More recent research activities on networked robots can be found in the website of IEEE Technical Committee on Network Robots [11]. Here, we mentioned a few of them that are directly related to our work.

An accelerometer-based human and robot interaction system has been proposed in [12]. The proposed system consists of an industrial robot using two low-cost and small 3-axis wireless accelerometers. These accelerometers are mounted on the human arms to capture gestures and postures. An Artificial Neural Network (ANN) algorithm has been used to recognize arm gestures and postures that are used as input to control the robot. The results presented in [12] show that the system is able to control an industrial robot in an intuitive way with a very high accuracy (92%) and with the very quick system response time (160 milliseconds).

A wireless controlling between a robot and an Android GUI application has been presented in [13]. The main purpose of the presented system is to make a surveillance robot that can be controlled via an Android application. The proposed system is divided into two parts namely the RF transmitter, and the RF receiver. The transmitter side includes an Android device, a programmable microcontroller, and a RF module transmitter. While the receiver side has two motors, another programmable microcontroller, RF Receiver, night vision wireless camera, and receiver unit. The actions of controlling the motors and the command of switching on and off the night vision camera have been carried out by the Android application. The captured videos are then transmitted to a TV through RF signal.

A mobile surveillance vehicle controlling through an Android base mobile has been presented in [14]. The system consists of an Android phone, Bluetooth Module, Arduino microcontroller, and three controlled items: Vehicle Direction Actuators, Vehicle Light Actuators, and a Voice Control Unit. The devices used in the system are similar to the one used in our present work. The Android application is used to control the actuators direction, to toggle the light bit, or the voice bit. The work done in both [13] and [14] are considered similar in a way that both of them have used controlling technique through a mobile application. However, these both works have scope to be extended. The use of two microcontrollers in [13] has increased needed coding; where two programs are written for both microcontrollers, and the overall system cost is increased. This has been avoided in our present work and the work done in [14]. We use a single Arduino microcontroller for our system. However, both works (presented in [13] and [14]) are designed only to control the movement of devices in four directions. While in our work we designed the Android application in a more enhanced way. The users will have the ability to (a) control the robot in four directions, and (b) choose from a set of already programmed paths. Hence, our system is more advantageous in some applications like industrial applications to let the robot move in a path that can be defined by mathematical functions. The functions can be used to move the robot in any desirable path. Hence there is no need to keep controlling the direction constantly with four direction arrows on a graphical user interface.

3. System Model

In this section the system block diagram of the presented work is described, along with the description of the used items. The system block diagram of our proposed system is shown in Fig.2. The proposed system consists of a controlling device, wireless transmitter, wireless receiver, microcontroller, and controlled device (i.e., robot). In the proposed system a controlling device would send data wirelessly based on the user’s command. These data would be received by a microcontroller, which in turn would send controlling commands to the desired controlled object.

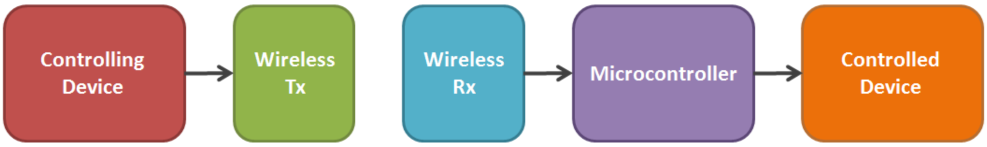


Fig. 2. The System Block Diagram

There are various controlling devices or user interface devices are available in the market. Among these computers and smart phones are widely used as user interface devices in the current generation systems. In our system we choose to use smart phone. Because a smart phone now can beat a computer in terms of the cost, portability, and even connectivity. For supporting wireless connectivity we consider four alternatives namely Wi-Fi, ZigBee, Bluetooth, and GSM. Each of these technologies is able to send the command messages from the user to the microcontroller. Each one of these technologies have their own unique properties and features. Among these technologies we use Bluetooth for implementing our system. Because Bluetooth operates in the 2.4 GHz ISM band. It support wireless connectivity among mobile phones, Personal Digital Assistants (PDAs), and other similar devices for exchanging data among them. Since Bluetooth system uses a radio link in the ISM band, there are several key advantages that its transport protocol can exploit. Bluetooth can effectively provide connectivity between a mobile device and other devices namely PDAs, Printers, LCD Projects, Mobile Phones, Wireless LAN devices, Notebooks, and desktop PCs.

We have used a microcontroller for linking the data obtained wirelessly from the user interface device and the controlled object. The controlled object would react depending on whatever data sent from the microcontroller. In the proposed system we choose Arduino as the microcontroller. Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs (i.e., light on a sensor, a finger on a button, or a Twitter message) and turn it into an output to activate a motor, turn on an LED, and publish data online.

In this work we also have designed a user interface. The user interface in the industrial design field of human–machine interaction is the space where interactions between humans and machines occur. The goal of this interaction is to allow effective operation and control of the machine from the human end, whilst the machine simultaneously feeds back information that aids the operator’s decision making process. In computing an interface is a shared boundary across which two separate components of computer system exchange information. The exchange can be among software, computer hardware, peripheral devices, humans, and combinations of these.

4. Hardware Implementation

In this section the hardware used to implement the system are explained in details, including the Android's Operating System, BlueSMiRF, Arduino, RGB 32x32 LED Panel, Arduino Robot, and Kobuki Base Robot. A comparison between the two robots has been made at the end of the section.

The hardware and software components used to implement the system include the Android's Operating System, BlueSMiRF, Arduino, and RGB 32 x 32 LED Panel, Arduino Robot, and Kobuki Base Robot as shown in Fig.3. In this work an Android operating system based smart phone has been used as the user interface device. The BlueSMiRF has been used as the wireless communication technology. Arduino Uno has been used as the microcontroller. An RGB 32x32 LED panel has been used to display the path. Arduino Robot and Kobuki Robot are the controlled devices in our designed system.

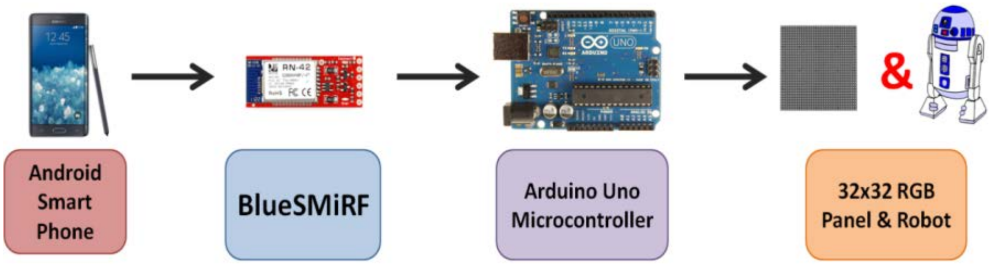


Fig. 3. Hardware System

We have used an Android Operating system based Smartphone as our controlling device in our proposed system. The Android system is based on the Linux Kernel and currently developed by Google with a user interface based on direct manipulation. The Android is designed primarily for touch screen smart devices such as smart phones and tablet computers with specialized user interfaces for televisions (Android TV), cars (Android Auto), and wrist watches (Android Wear). The operating system uses touch inputs that loosely correspond to real-world actions like swiping, tapping, pinching, and reverse pinching to manipulate on-screen objects and a virtual keyboard. Despite being primarily designed for touchscreen input it also has been used in game consoles, digital cameras, regular PCs, and other electronics.

Applications (abbreviated as Apps) extend the functionality of Android devices. They are developed primarily in Java programming language using the Android software development kit (SDK). The SDK includes a comprehensive set of development tools including a debugger, software libraries, a handset emulator based on QEMU, documentation, sample code, and tutorials. Other development tools are available including Native Development Kit and Google App Inventor. The Native Development Kit is used for applications or extensions in C or C++. The Google App Inventor is used in various cross platform mobile web applications frameworks. In this work we have used Google App Inventor devolvement tool in order to create an Android App for the users to control and set the path they would like to create on the LED Panel, or control the movement of both used robots.

We have used Bluetooth technology as a wireless communication between the smartphone and the Arduino. We used BlueSMiRF chip as our Bluetooth device. The BlueSMiRF, as shown in Fig.4, is the latest Bluetooth wireless serial cable replacement. It uses the RN-42 module, which has a bit less range than the RN-41 module. These modems work as a serial (RX/TX) pipe. Any serial data stream from 2400 bps to 115200 bps can be passed seamlessly from the smartphone to the target. The remote unit can be powered from 3.3 volt up to 6.0

volt for easy battery attachment. All signal pins on the remote unit are 3–6 volt tolerant. No level shifting is required. It cannot be attached directly to a serial port. An RS232 to TTL converter circuit would be needed for this purpose.

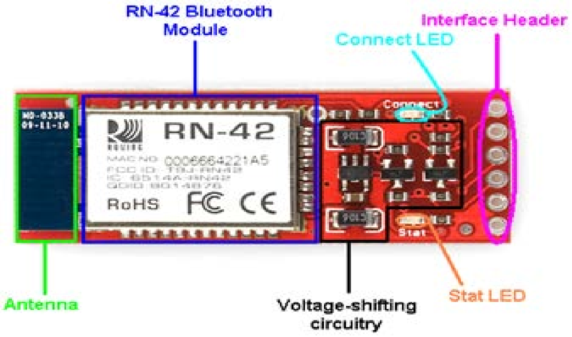


Fig. 4. BlueSMiRF Pin Diagram

We have used an Arduino Uno as a microcontroller for our system. Arduino is a family of single-board microcontrollers intended to make it easier to build interactive objects or environments. The hardware consists of an open-source hardware board designed around an 8-bit Atmel AVR microcontroller or a 32-bit Atmel ARM. Current models feature an USB interface together with six analog input pins and 14 digital I/O pins that can accommodate various extension boards.

The Arduino board (see Fig. 5) exposes most of the microcontroller's I/O pins for use by other circuits. The Diecimila, Duemilanove, and current Uno provide 14 digital I/O pins. Six of them can produce pulse-width modulated signals, and six analog inputs that can also be used as six digital I/O pins. These pins are on the top of the board via female 0.10 inch (2.5 mm) headers. Several plug-in application shields are also commercially available. The Arduino Nano, and Arduino-compatible Bare Bones Board, and Boarduino boards may provide male header pins on the underside of the board that can plug into solderless breadboard.

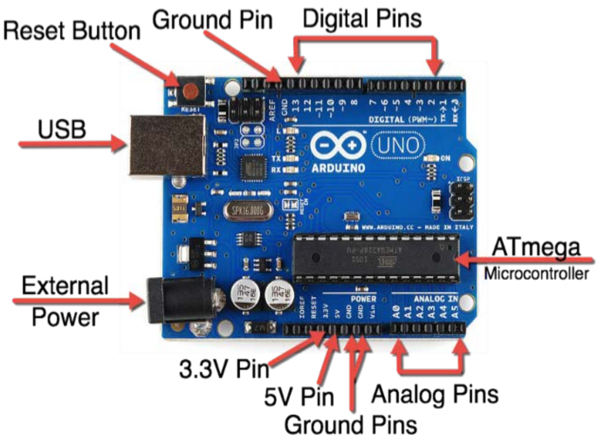


Fig. 5. Pin diagram of used Arduino Uno in this project

The first controlled device of our work was a 32x32 LED panel as shown in Fig.6. The LED panel contains 1024 RGB LEDs arranged in a matrix of dimension 32 x 32. Each RGB LED contains separate red, green, and blue LED chips assembled together in a single package. The display is subdivided horizontally into two halves. The top half consists of 32 columns and 16 rows. The bottom half also consists of 32 columns and 16 rows as shown in Fig.6. The display's columns and rows are driven by two different set of drivers. To illuminate an LED the drivers for both the column and the row for that LED must be turned on. To change the color of an LED, the red, green, and blue chips in each LED package are controlled individually and have their own column drivers.

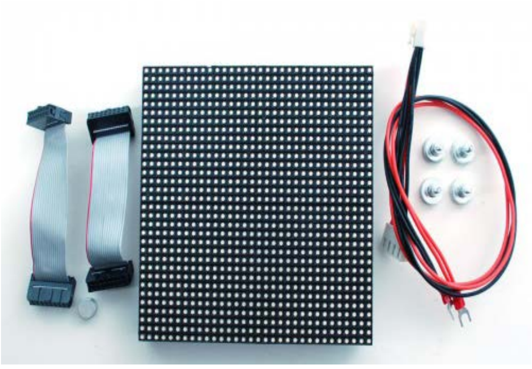


Fig. 6. The 32x32 RGB LED panel

In order to control any pixel (LED) of the panel, a hardware connection among the BlueSMiRF chip, Arduino, and the LED Panel is made as shown in Fig. 7. The connections between the previous devices were wired, whereas the smart phone was wirelessly connected to them through the Bluetooth technology.

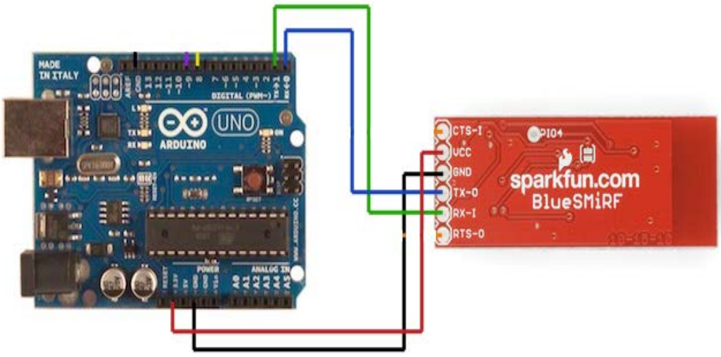


Fig. 7. The connection between BlueSMiRF and Arduino UNO

The second controlled device, was the Arduino Robot (Fig. 8), where the movement of the robot has been controlled. Similar to the LED Panel a user is able to either move the robot in an already defined paths, or directly control its movement in the four directions.

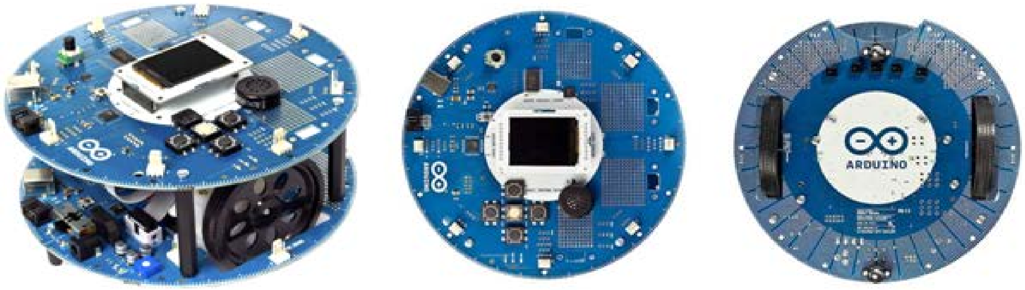


Fig. 8. Arduino Robot, side, top and bottom view of the robot (from left to right).

The Arduino Robotic is the first official Arduino on wheels. The robot has two processors mounted on two boards. The Motor Board controls the motors, and the Control Board reads sensors and decides how to operate. Each of the boards is Arduino board programmable using the Arduino IDE. Both Motor and Control boards of the robot are based on the ATmega32u4. The Robot has many of its pins mapped to on-board sensors and actuators. Programming the robot is similar to the process with the Arduino Leonardo. Both processors have built-in USB communication, eliminating the need for a secondary processor. This allows the Robot to be connected to a computer as a virtual (CDC) serial / COM port. We connected an Arduino UNO to the control board. Fig. 9 and Fig.10 show the connection between the Control Board and the Arduino UNO. In order to control the robot's movement we wrote two programs: one for the Arduino Control Board, and the other for the Arduino UNO.

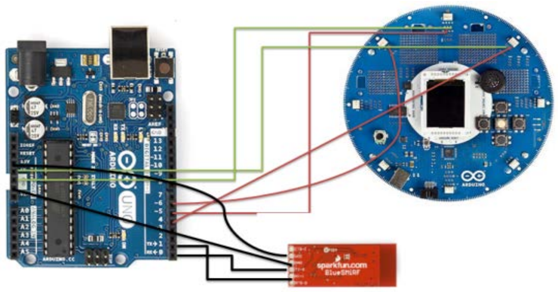


Fig.9 Arduino Robot Control Board and Arduino UNO

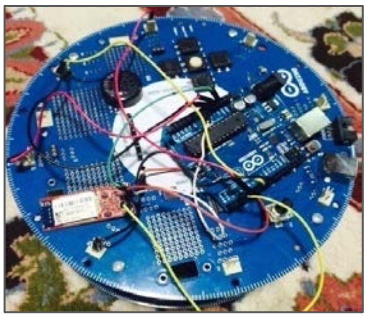


Fig.10 Real Connection

Fig.11 Kobuki Base Robot

Our third controlled device was Kobuki Base Robot (Fig.11). It is a low-cost state of art robot specially designed for education and research. With continuous operation in mind Kobuki robot provides power supplies for an external computer as well as additional sensors and actuators. It has a highly accurate odometer that enables precise navigation. In order to control the movement of the robot an Arduino UNO is connected to its serial pins.

Fig.12 and Fig.13 show the hardware connection of the robot with the Arduino UNO and the BlueSMiRF. The function of this circuit is to supply voltages to the robot's motors through the serial pins connection. The response and movement of each motor depend on the voltage sent to it. The threshold voltage of the Kobuki was 1.9 volt. If both motors were given 1.9 volt, both wheels would not move and the robot would stop. If the given voltage is higher than that of the threshold voltage, the wheel would move forward. If it's less than the threshold value it would move backward.

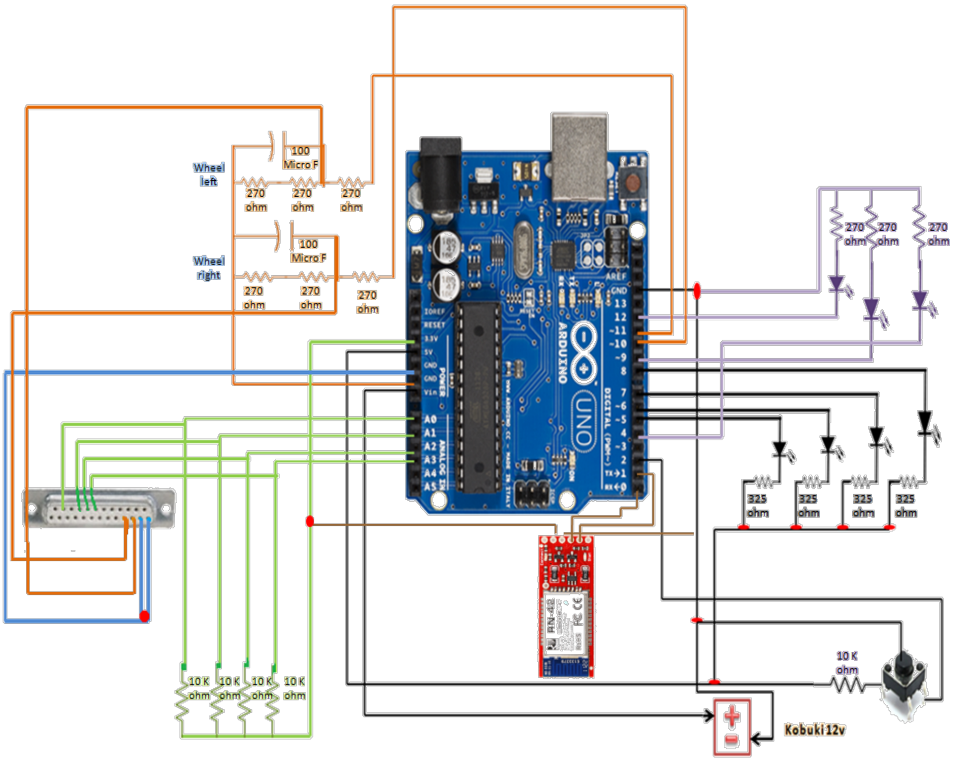


Fig. 12. Hardware connection of the Kobuki Robot.

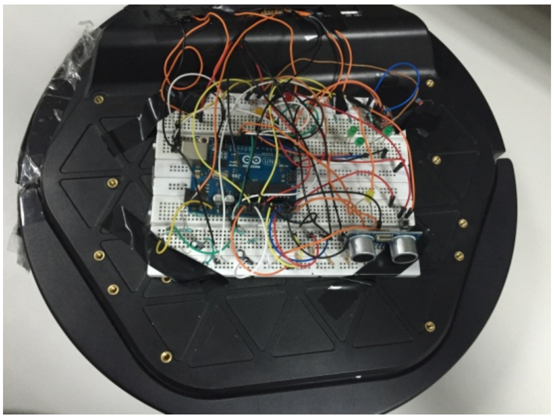


Fig. 13. Real Connection

A comparison between the two robots in terms of: size, speed, cost, reliability, rotation accuracy, easy of programming has been listed in Table1.

Table1: Comparison of Arduino Robot and Kobuki Robot

Topic	Arduino Robot	Kobuki Base Robot
Size	Small	Bigger
Speed	Speed can be changed be changed according to the corresponding voltage taken from the Arduino UNO, but in general it's faster than the Kobuki	Speed can be changed according to the corresponding voltage taken from the Arduino UNO, but in general it is slower than the Arduino Robot.
Sensors	5 Line following IR sensors.	3 bump sensors.
Reliability	Reliability may vary depending on the application used for the robot.	Reliability may vary depending on the application used for the robot.
Programming	Easy programming, and there is no need for any hardware connection in order to program it, since it comes with a built microcontrollers.	Not very easy to program, since an external hardware circuit must be connected to in order to be able to move the robot, and a lot of parameters must be taken into consideration while programming.
Cost	Less costly than the Kobuki (around 200$)	Higher cost (around 580$)

5. Software Implementation

In this section the Smartphone application, Arduino Robot, and the Kobuki Robot flow charts and software functions are presented. The final Smartphone's application design and its working mechanism are also explained in this section.

Fig.14 shows the flowchart of smartphone's application. The sent data, from the smart phone to the Arduino microcontroller via the BlueSMiRF device, are dependent on the command (button) that the users choose. If How to use button is chosen, another screen would be opened describing the How to use explanation. If the Contact Us is chosen, another screen that contains different contacting options between the user and the developers would be displayed. If Design Path, Saved Paths, or the Function Paths buttons are chosen, the smartphone would send data describing that chosen option to the Arduino microcontroller, which in turn would be displayed in each screen. This application is the same for all the controlled devices in this project, only one noticeable difference is in the LED Panel, where there is no Function Path option. The function path is only used in controlling the two robots.

The Application uses Graphical User Interface (GUI), which means that there is no need to have a go back button to be added in each screen. In this case the user would use the built in back button in the mobile itself. When user would just touch that button and the previous page would be displayed.

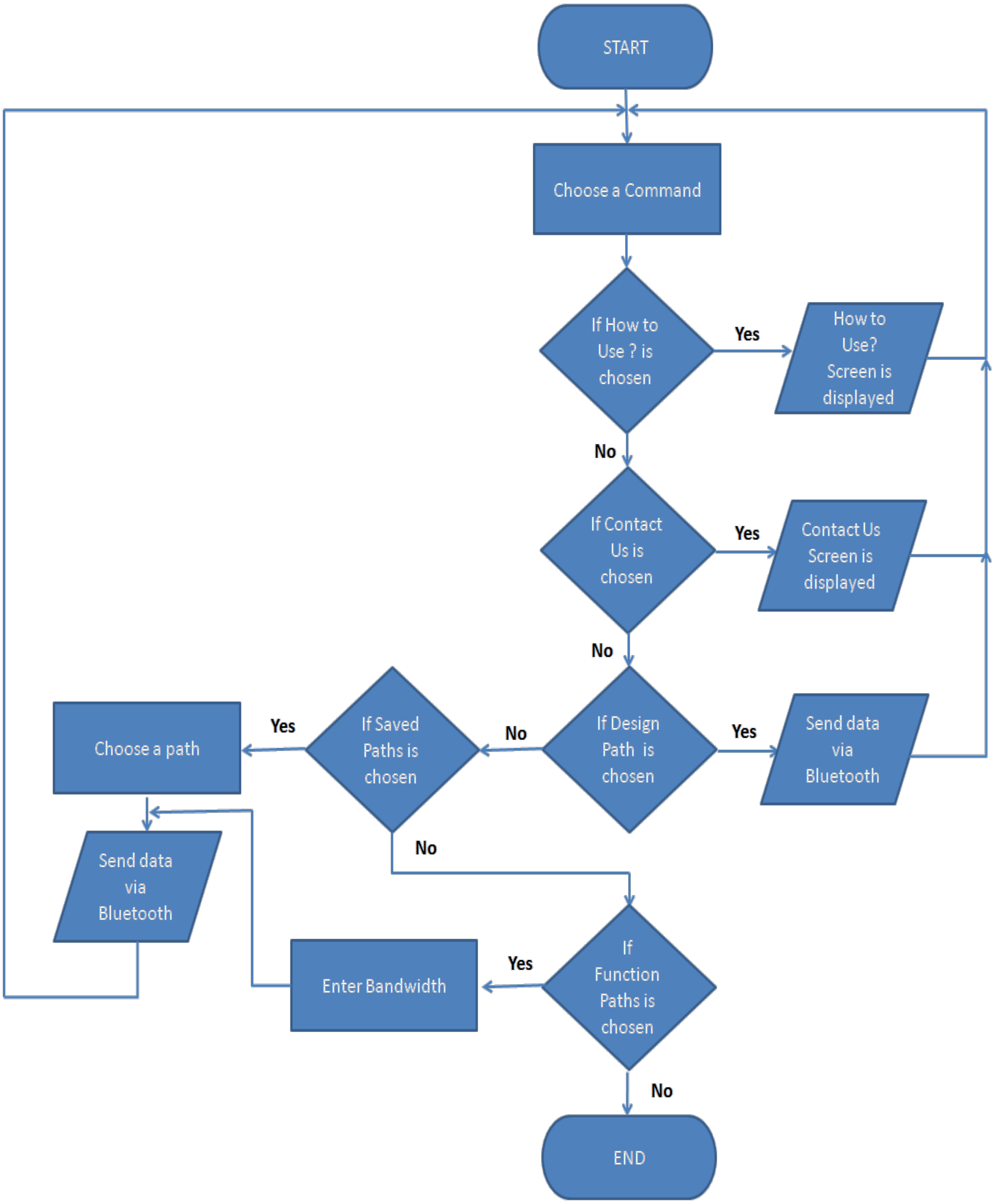


Fig.14 Smartphone Application Flow Chart

Fig.15 shows the software flow chart of the Arduino UNO for the LED Panel. The Arduino responds according to the data received from the Bluetooth device. The data received are programmed to be 13 different characters; each character would compute different command that would be sent to the LED panel. Table-2 shows the computed commands of each received character.

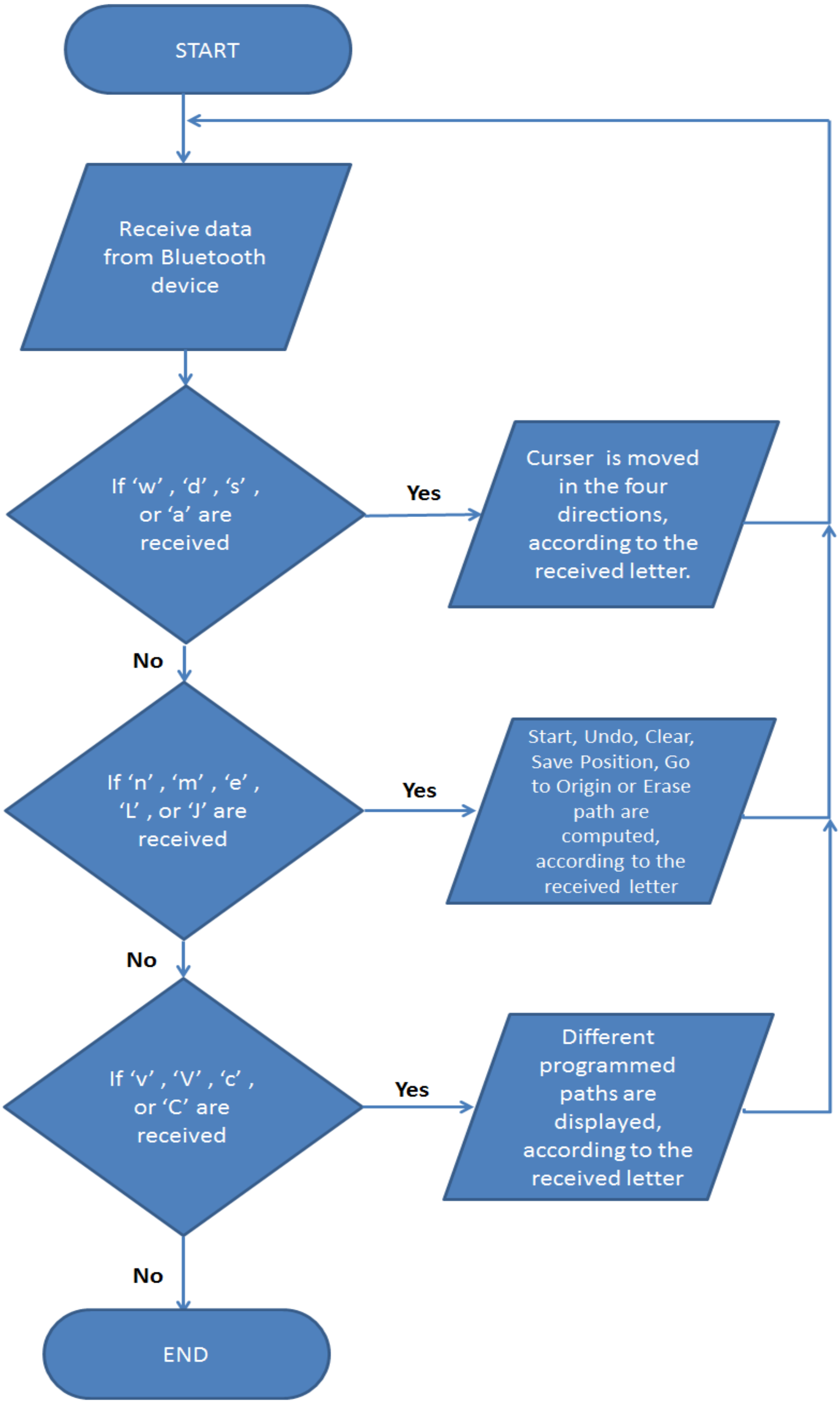

START
Receive data from Bluetooth device
If 'w' , 'd' , 's' , or 'a' are received
Yes
Curser is moved in the four directions, according to the received letter.
No
If 'n' , 'm' , 'e' , 'L' , or 'J' are received
Yes
Start, Undo, Clear, Save Position, Go to Origin or Erase path are computed, according to the received letter
No
If 'v' , 'V' , 'c' , or 'C' are received
Yes
Different programmed paths are displayed, according to the received letter
No
END

Fig.15 Smartphone Application Flow Chart

Table 2: Received Data in the microcontroller and their computed commands

Received Data	Computed Commands
'n'	Begin controlling the path.
'w'	The controlled pixel on the LED panel would take a step up (front).
'd'	The controlled pixel on the LED panel would take a step to the right.
's'	The controlled pixel on the LED panel would take a step down (back).
'a'	The controlled pixel on the LED panel would take a step to the left.
'm'	Undo movements.
'e'	Clear the LED panel screen.
'L'	Save the current position of the pixel.
'J'	Return controlled pixel to the origin point (0,0).
'C'	The controlled pixel would move in a path similar to a sawtooth function.
'c'	The controlled pixel would move in a programmed random shape.
'V'	The controlled pixel would move in a path similar to a straight line function.
'v'	The controlled pixel would move in a path similar to a stair function.

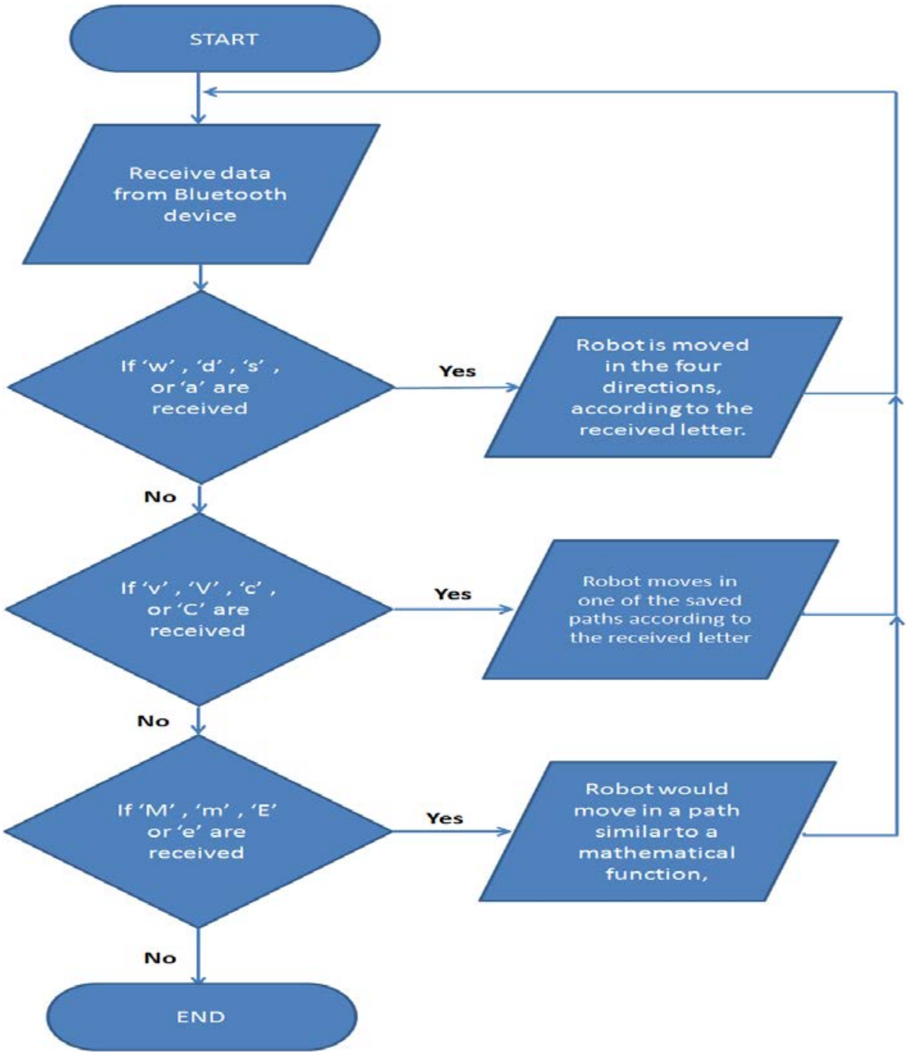

START
Receive data from Bluetooth device
If 'w' , 'd' , 's' , or 'a' are received
Yes
Robot is moved in the four directions, according to the received letter.
No
If 'v' , 'V' , 'c' , or 'C' are received
Yes
Robot moves in one of the saved paths according to the received letter
No
If 'M' , 'm' , 'E' or 'e' are received
Yes
Robot would move in a path similar to a mathematical function,
No
END

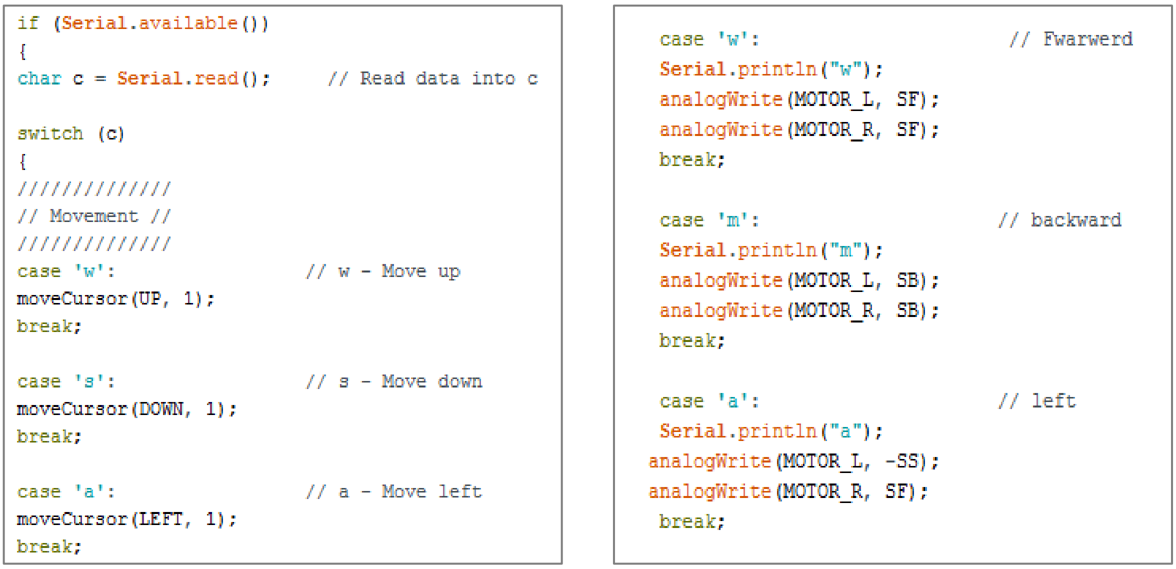
Fig.16 Robots Flow Chart

Fig.16 shows the software flow chart used in both robots. It is similar to the one used in the LED Panel. The only difference is that rather than controlling the pixel's movement, the two motors are being controlled. Table 3 shows the computed commands in both robots.

Table3: Received Data in the microcontroller & their computed commands

Received Data	Computed Commands
'n'	Begin controlling the path.
'w'	The robot moves forward.
'd'	The robot moves to the right.
's'	The robot moves backward.
'a'	The robot move to the left.
'L'	Robot stops.
'C'	The robot would move in the first saved path.
'c'	The robot would move in the second saved path.
'V'	The robot would move in the third saved path.
'v'	The robot would move in the fourth saved path.
'M'	The robot would move in the first function path.
'm'	The robot would move in the second function path.
'E'	The robot would move in the third function path.
'e'	The robot would move in the fourth function path.

It's important to note, that these characters (the received letters) are nothing but a case structure in the Arduino program (see Fig. 17), which we've defined to compute different commands.



```
if (Serial.available())
{
char c = Serial.read();      // Read data into c

switch (c)
{
//////////////
// Movement //
//////////////
case 'w':                   // w - Move up
moveCursor(UP, 1);
break;

case 's':                   // s - Move down
moveCursor(DOWN, 1);
break;

case 'a':                   // a - Move left
moveCursor(LEFT, 1);
break;
```

```
 case 'w':                           // Fwarwerd
 Serial.println("w");
 analogWrite(MOTOR_L, SF);
 analogWrite(MOTOR_R, SF);
 break;

 case 'm':                          // backward
 Serial.println("m");
 analogWrite(MOTOR_L, SB);
 analogWrite(MOTOR_R, SB);
 break;

 case 'a':                          // left
 Serial.println("a");
analogWrite(MOTOR_L, -SS);
analogWrite(MOTOR_R, SF);
 break;
```

Fig.17 Part of the Arduino code for LED & two robots (Left to Right)

In order to design the Android Smart Phone App, App Inventor, an open-source web application originally provided by Google and is maintained by the Massachusetts Institute of Technology (MIT), has been used.

App Inventor allows programmers to create software applications for Android Operating System (OS) smart phones. It uses a graphical interface that is very similar to Scratch and StarLogo TNG user interface. It allows users to drag-and-drop visual objects to create an application that can run on Android devices. We've used App Inventor 2 in designing our mobile Application, which has an additional and easier features than the App Inventor Classic.

The first phase of application design goes through App Inventor Design page. Design is accessible through the web page and all the ingredients for the app are available on the left side of the window. The ingredients include elements like a screen for the app, buttons for tapping, text boxes, images, labels, animations and many more. The right side of the design allows users to view the screen and components added to the screen. Additionally, the properties section of the window allows users to modify the properties of components (see Fig. 18). The Block screen is used to set any components on the Design, basically the Block is where programmers would control the function of each element on the Design screen (Fig. 19).

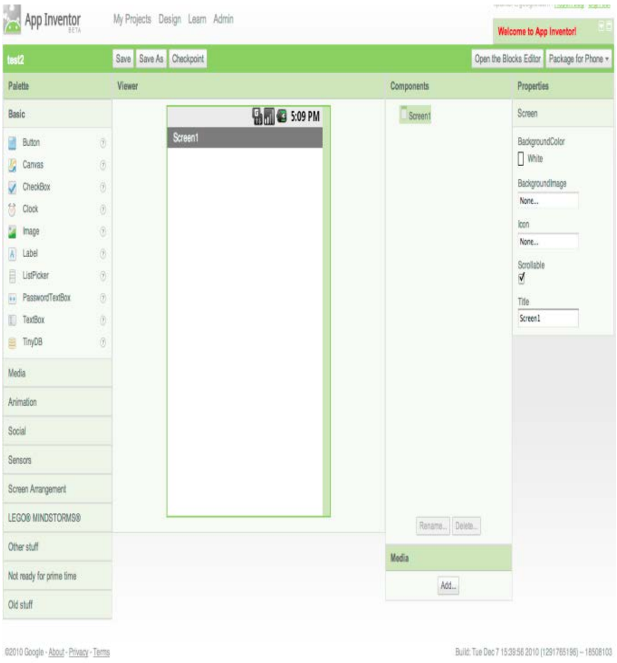


Fig. 18. App Inventor 2 Designer

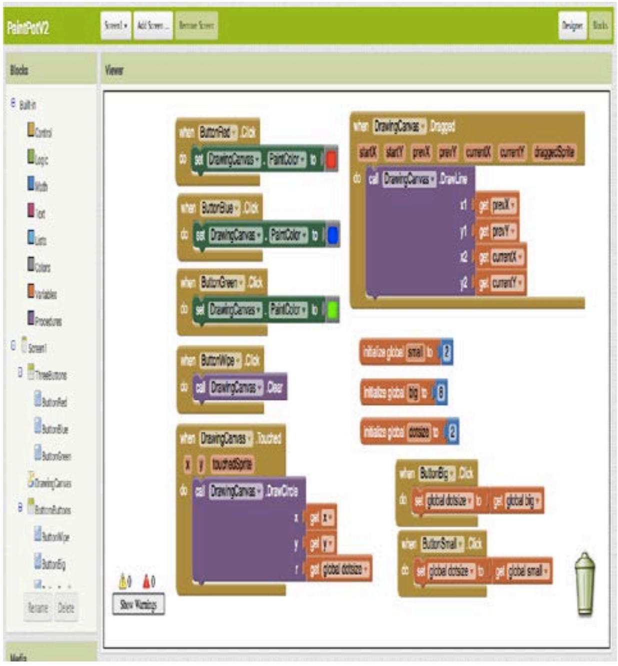


Fig. 19. App Inventor2 Block

To control the movement of either the robot motors or the LED pixel of the panel, we have design an Android Application to pass the users commands to the controlled devices. We’ve named the Application E-Messenger. The Panel’s E-Messenger application enables user to choose between drawing any shape manually, or choose from a built in paths to display on the LED Panel as shown in Fig.20. These paths are the paths that the user would want to set for the object to move in.

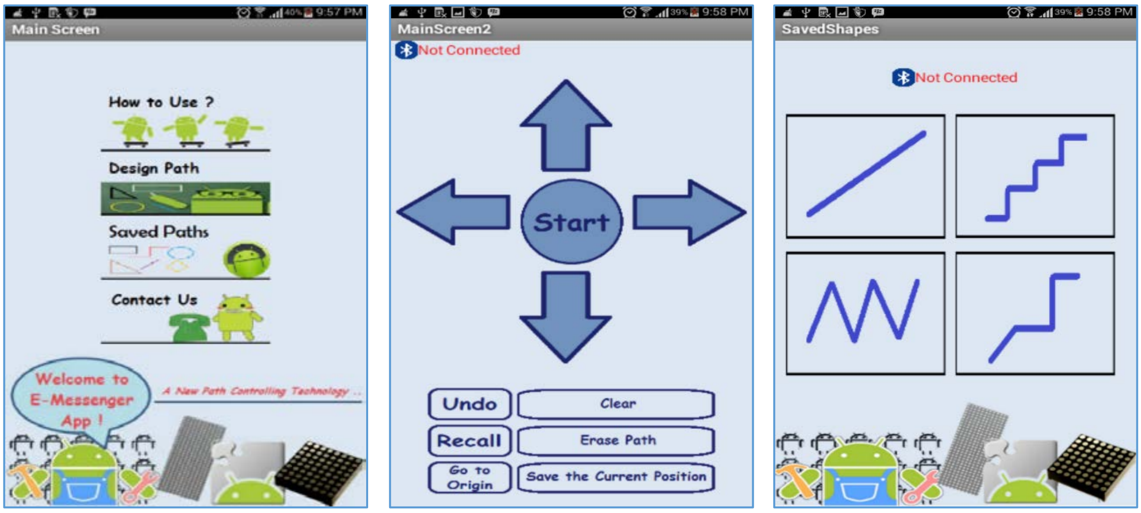


Fig. 20. Main, Design Path and Saved Paths of the LED Panel App (Left to Right)

When the user clicks any button on the App, the App would send the character indicating that button to the Bluetooth module that in turns forwards the character to the Arduino microcontroller used. The Arduino microcontroller then receives that character, and matches it with the set conditions to send a command to the LED Panel to display. Fig. 21 and Fig. 22 show a saved path and a set path by user respectively.

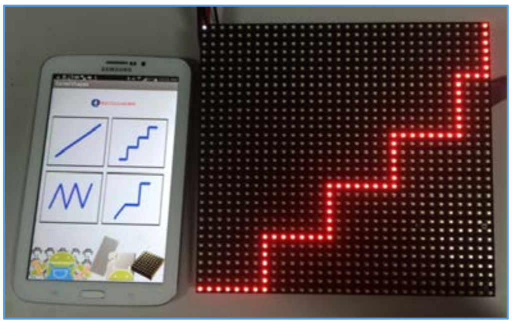


Fig. 21. Saved Path

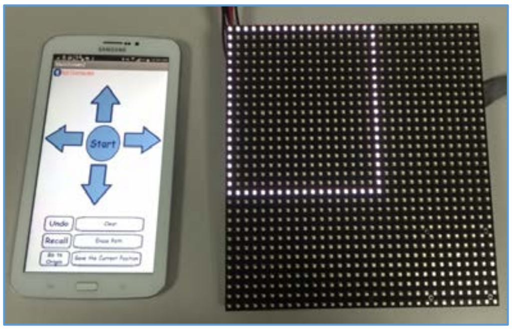


Fig. 22. Set Path by User

Similarly, another application with same functions has been designed to control the other two robots. The application name is also E-Messenger. Both Apps works the same, except that a page called “Function Path” has been added. Through this page users would be able to move the robots in a path similar to a specific function. The only thing needed by user is to set the bandwidth, or the distance that the robot must move in, and then the robot would just follow that path. Fig.23 shows the Function path page. Fig. 24 shows how a Kubuki robot is being moved along as saved square path.

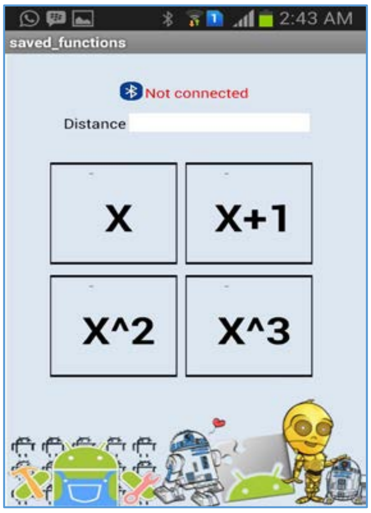


Fig. 23. Function Path

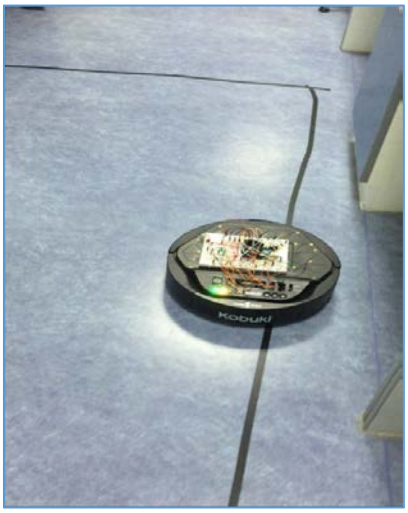


Fig. 24. Kuboki robot moving in a saved squire path

6. Conclusion

In this paper we have presented wireless path controller system for moving Robots. The system uses an interface hardware, which is a smart phone device. We have created an Android Application for users to choose from various built in paths or set their own paths to move the controlled object. The selected commands were wirelessly sent via a Bluetooth device called BlueSMiRF. The BlueSMiRF consequently sends a certain character to the Arduino microcontroller depending on the selected command on the App. We tested our proposed system to control three devices namely the LED Panel, Arduino robot, and Kobuki Base robot. We presented the E-Messenger application where user would be able to (a) design his/her own path by directly controlling the movement of the robots, (b) choose from an already saved paths, (c) choose from a set of mathematical functions and inputting the distance that the user would like the robot to move in. So far the function of the robots is limited. In future the robots can be made smart, and can be enhanced.

7. References

[1] George M. Pierce II,” Robotics: Military Applications for Special Operation Forces”, Research Report available at http://www.au.af.mil/au/awc/awcgate/acsc/00-142.pdf

[2] Gerald Miles, “Military robots of the present and the future”, Technology, Vol. 9, No.1, 2011, pp. 125-137.

[3] Voth, D., “A new generation of military robots”, IEEE Intelligent System, Vol. 19, No. 4, August 2004, pp. 2-3. http://dx.doi.org/10.1109/MIS.2004.30

[4] Raj Madhavan, “Robots in Military and Aerospace Technologies”, IEEE Robotics and Automation Magazine, Vol. 7, No. 2, June 2010, pp. 6-6. http://dx.doi.org/10.1109/MRA.2010.936954

[5] Shukla A. and Karki H. "A review of robotics in onshore oil-gas industry", Proceedings of IEEE International Conference on Mechatonics and Automation, Takamatsu, August 2013, pp. 1153-1160. http://dx.doi.org/10.1109/ICMA.2013.6618077

[6] S. Stavinoha, H. Chen, M. Walker, B. Zhang, T. Fuhlbrigge, "Challenges of robotics and automation in offshore oil&gas industry", Proceedings of the 4th IEEE Annual International Conference on Cyber Technology in Automation, Control and Intelligent Systems, Hong Kong, June 2014, pp. 557-562. http://dx.doi.org/10.1109/CYBER.2014.6917524

[7] Anisi, D.A.; Gunnar, J.; Lillehagen, T.; Skourup, C., "Robot automation in oil and gas facilities: Indoor and onsite demonstrations", Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, October 2010, pp. 4729 – 4734. http://dx.doi.org/10.1109/IROS.2010.5649281

[8] Arthur Visser, Industrial Automation and Robotics: What is Next? Available at: http://www.connectorsupplier.com/industrial-automation-and-robotics-whats-next/

[9] Dezhen Song, Networked Robots: from Telerobots to Cloud Robotics. Available at http://faculty.cs.tamu.edu/dzsong/pdfs/044NetworkedRobots.pdf

[10] Ken Goldberg and Roland Siegwart, "An introduction to Online Robots", MIT press 2002.

[11] IEEE Robotics and Automation Society. Available at: http://www.ieee-ras.org/

[12] P. Neto, J. Pires, and A. Morcira, "Accelerometer-based control of an industrial robotic arm", Proceedings of the 18th IEEE International Symposium on Robot and Human Interactive Communication, Toyama, September-October, 2009, pp. 1192 – 1197. http://dx.doi.org/10.1109/ROMAN.2009.5326285

[13] M. Selvam, "Smart Phone Based Robotic Control for Surveillance Application", International Journal of Research in Engineering and Technology, Vol. 03, Issue 02, 2013, pp 229-232. http://dx.doi.org/10.15680/ijircce.2015.0301020

[14] Syed M Taha Saquib, Sarmad Hameed, Syed M Usman Ali, Raza Jafri, and Imran Amin, "Wireless Control of Miniaturized Mobile Vehicle for Indoor Surveillance", IOP Conf. Series: Materials Science and Engineering, Vol. 51 (2013). Pp. 1-7. http://dx.doi.org/10.1088/1757-899X/51/1/012025

Copyright Disclaimer