Open source hardware and software platform for robotics and artificial intelligence applications

Liang S Ng, K O Tan, Clement T H Lai, S K Ng, Mohammed A H Ali, Musa Mailah, Wan Azhar Yussof, Zamzuri Hamedon, Zulkifli Yussof

Faculty of Manufacturing Engineering, Universiti Malaysia Pahang, 26600 Pekan, Pahang, Malaysia.

E-mail: `nlshing.ump@gmail.com`

Abstract.   Recent developments in open source hardware and software platforms (Android, Arduino, Linux, OpenCV etc.) have enabled rapid development of previously expensive and sophisticated system within a lower budget and flatter learning curves for developers. Using these platform, we designed and developed a Java-based 3D robotic simulation system, with graph database, which is integrated in online and offline modes with an Android-Arduino based rubbish picking remote control car. The combination of the open source hardware and software system created a flexible and expandable platform for further developments in the future, both in the software and hardware areas, in particular in combination with graph database for artificial intelligence, as well as more sophisticated hardware, such as legged or humanoid robots.

1. Introduction

Recent developments in open source hardware and software platforms (Android, Arduino, Linux, OpenCV etc.) have enabled rapid development of previously expensive and sophisticated system within a lower budget and flatter learning curves for developers. Using these platform, we designed and developed a Java-based 3D robotic simulation system, with graph database, which is integrated in online and offline modes with an Android-Arduino based rubbish picking remote control car. The combination of the open source hardware and software system created a flexible and expandable platform for further developments in the future, both in the software and hardware areas, in particular in combination with graph database for artificial intelligence, as well as more sophisticated hardware, such as legged or humanoid robots.

This paper outlines the applications of open source hardware and software modules, as listed below, in order to create a low cost, integrated platform (comprising both hardware and software) for research as well as teaching and learning in robotics and artificial intelligence:

(i) jMonkey and jmeSim [1]: java based 3D simulation environment, for simulation, testing and integration with hardware modules. (see figure 1)

(ii) Arduino [2]: interface between Android app (human operator control) and remote control (RC) gripping robot (modified from RC car). (see figures 2 and 3)

(iii) Android MIT app inventor [3]: human operator control program. (see figure 12)

(iv) OpenCV [4]: image processing used on mobile gripping robot. (see figure 4)
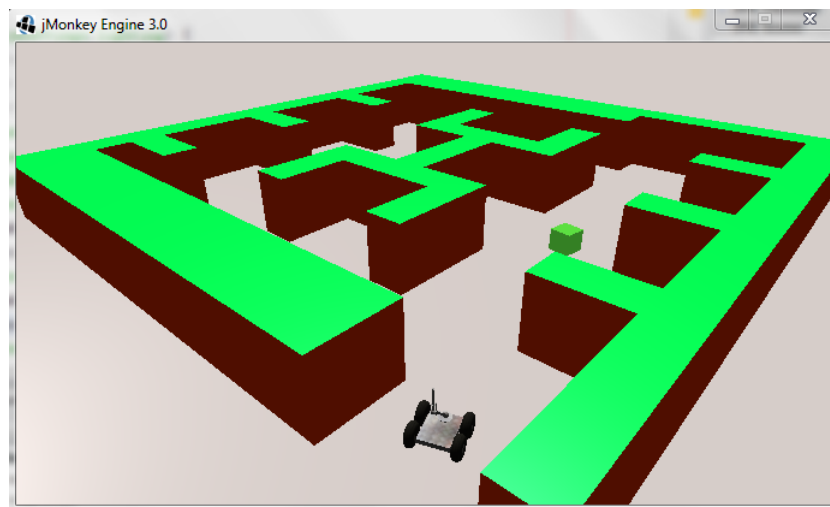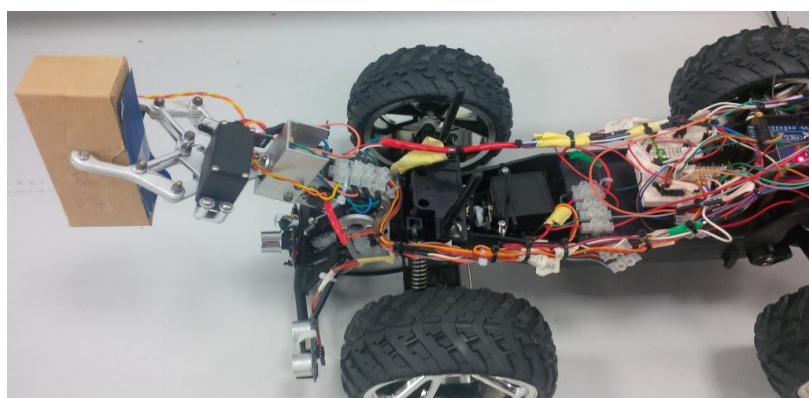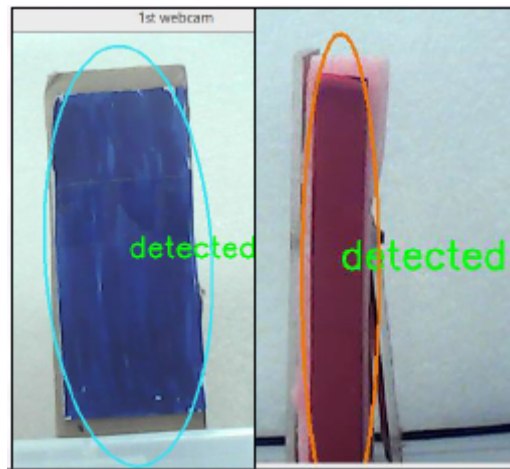
Figure 1. Indoor Environment (Bright) in JmonkeyEngine.
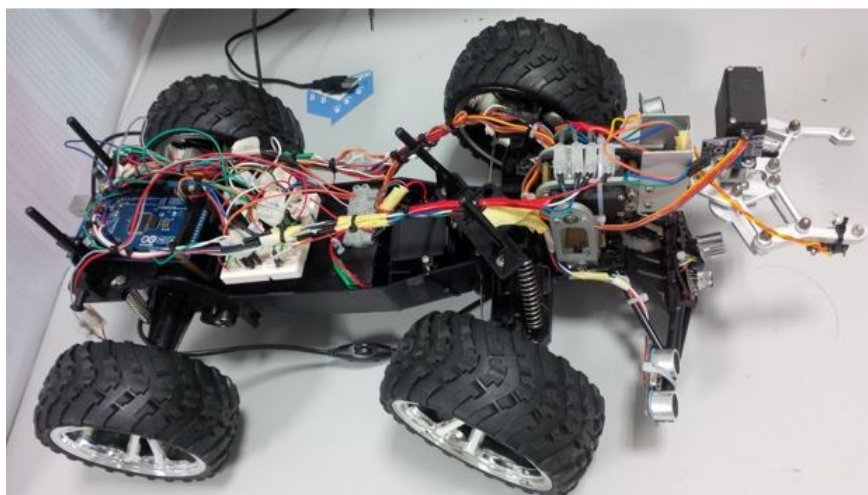
Figure 2. The mobile gripping robot.

Figure 3. Robot successfully gripped object.

Figure 4. The object detected using OpenCV. The left object was blue colour box and the right object was a red colour box.

2. System Modules

2.1. Virtual Environment and Robot

Open source simulation software such as webot, JmeSim, USARSim and Gazebo is surveyed base on it feature and capabilities. JmeSim is then to be chosen as the virtual robot development software in this project.

JMonkey Engine Software Development Kit (SDK) contains all the things we need to get started easily and efficiently in designing a virtual robot and environment. JMonkey Engine come with build in libraries, different type of plug-in, sample code, Javadoc, and plenty of virtual robot development utilities.

JmeSim is a robot simulator that build inside JmonkeyEngine. JmeSim is built based on the game engine, the open source JMonkey Engine. Since the JmeSim and JMonkey Engine are purely java based, JmeSim is architecture neutral; it has been tested on Windows as well as the Linux operating system. JMonkey Engine is capable of delivering good graphical performance at real time frame rates.

The physic simulation inside JmeSim is performed by jBullet where jBullet is a Java port of the Bullet Physics library. It makes work easier to test and analysis the physic simulation on virtual robot. Moreover, it provides fabulous stability, speed and accuracy which better than other commercially licensed physics engines.

In addition, JmeSim also offer a bundle of sensors which give data typically needed by an autonomous mobile robot. The JmeSim build in sensor are thermal camera, depth camera, sonar sensor, and laser range finder.

The environments are designed in Blender software [5]. Blender is a free and open source 3D creation suite. It supports the entirety of the 3D pipeline—modeling, rigging, animation, simulation, rendering, compositing and motion tracking, even video editing and game creation. In this project, Blender software was used to design the model for the difference environments.

Blender was used in this project because it is able to export the file into Ogre3D format which are support by JmonkeyEngine. Blender exports the model into .mesh and .material files. Both files were converted into .j3o format inside JmonkeyEngine and load into the scene. JmonkeyEngine support the model file with format .j3o. Therefore, Blender was one of the design software that can export the design model into JmokeyEngine.

2.2. Hardware Modules

2.2.1. Controller Development   Arduino IDE is an open source software which is written specially to program Arduino board. It easily can be download from http://www.arduino.cc/en/Main/Software
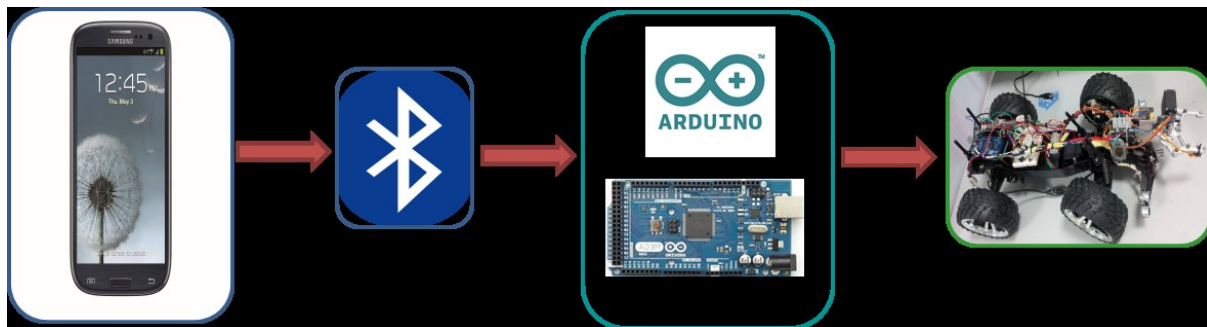
Figure 5. Flow of the whole controlling system.

without the needs to pay anything. It is written based on Java programming language and it is available for all kinds of environment which include Linux, Windows, and Mac OSX. Other than that, Arduino software is also hosted by Github which all different kinds of code which has been developed by different developers can be found in Github.This project Arduino IDE is used to program the Arduino ATmega 2560 board which is used to control all the actuators of the mobile gripping robot which includes the a DC geared motor and 4 servos motor. It also trigger and read the input from the ultrasonic sensor which used for measuring the distance. In order to communicate with the Android device, in the IDE the serial communication of the Arduino board will be initialized for receiving command from the Android device. The flow of the whole system is shown in the Figure 2.

2.2.2. Sensors and Interface   The ultrasonic sensor used for this project is HC- SR04, which is easily available on any electronic store [6]. The purpose of it is to detect the distance between the robot and any object or obstacles in front of the robot. The working principle of this ultrasonic sensor is the same as the sonar echo used by bats. The sensor itself will generate ultrasonic sonar echo which is 40 kHz and is beyond the range of hearing of human ear. The sonar echo is reflected back to the sensor when the sonar echo hit on any object of obstacles. The reflected echo is detected by the echo sensor. The time for a pulse of sonar echo generated from the sensor to travel to the object and reflected is used to calculate the distance between the sensor and the object.

The communication between the android device and the mobile robot itself is established by using the Bluetooth module which is HT Bluetooth Module HC-05 [7]. Using the Bluetooth module the wireless communication between the robot and the android device easily can be established compare to other type of communication. For example, RF control which is prone to interference when it is operating in the same frequency with other network. This Bluetooth module is easy to establish a proper communication with other devices. Other than that, this module is easy to interface with micro-controller, Arduino. This is because Arduino also contains the TX/ RX pin which enable serial communication with the Bluetooth module.

2.2.3. Mobile Application Development MIT App Inventor software was originally developed by Google but now it is fully hosted by Massachusetts Institute of Technology (MIT). This software has been developed since 2010 under Google's Mark Friedman and MIT Professor Hal Abelson co-led the development of App Inventor while Hal was on sabbatical at Google.This software is based on the block programming where all the codes of program are represented in the terms of simplify blocks. Therefore, it is easy to be use and simplify a lot of work in developing an Android app. It is also free and easily available at the web site with free cloud base which linked with Google mail to store all the data that has been developed by the user.It is easy to be use as an Android smart-phones user just have to install the "MIT Ai2 Companion" app to linked the Android device with the MIT App Inventor software. An emulator of Android device is available too for the user without personal Android device.
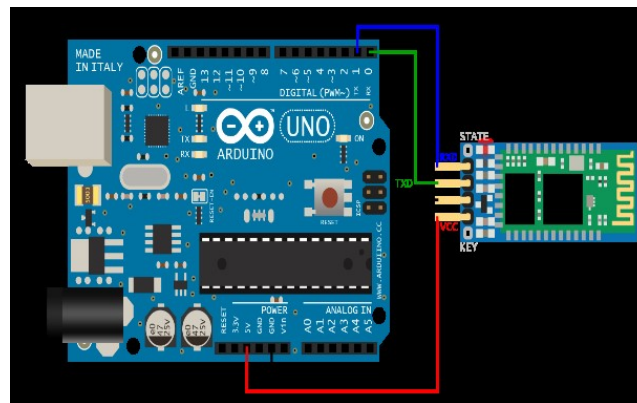
Figure 6. Connection of Arduino and Bluetooth module.

2.3. Software for Image Processing
The Visual Studio C++ compiler is used to compile OpenCV module and Arduino module. OpenCV is used for camera vision and image progressing and Arduino module is used to control the robot movement.

Image processing plays an important role in robotics applications [8]. There are still many unsolved problems in a simple set up such as controlling a robot to manipulate objects [9]. In this project, we hope to create a simpler robotics integration platform, for the purposes of research, teaching and learning, as an alternative to existing solutions such as the Robot Operating System [10].

3. Results and Discussions
3.1. Virtual Robot Simulator with JmonkeyEngine
A cylinder mesh was used to design the tree model in blender while the circle plane was used to design the leave for the tree. Than the material and geometry was added to the tree model. After that, the tree model was exported into 2 files; .material and .mesh. Those 2 files are then saved in the texture and was converted to .j3o file. The .j3o file is then saved in the model folder so that it can be loaded inside the robot simulation.

A cube plane was scaled to desired size and subdivided into scale 15. The shape of the environment is then designed by selecting the small cubes accordingly and extrudes it by 5mm. The material and geometry was added to the environment model. After that, the environment model was exported into 2 files; .material and .mesh. Those 2 files are then saved in the texture and was converted to .j3o file. The .j3o file is then saved in the model folder so that it can be loaded inside the robot simulation.

A cube plane is scaled and subdivided into scale 10. A step field is then designed by selecting the small cubes accordingly and extrude it to 0.5mm height. The uneven ground for outdoor environment was indicated by the step filed inside the robot simulation. The material and geometry was added to the environment model. After that, the environment model was exported into 2 files; .material and .mesh. Those 2 files are then saved in the texture and was converted to .j3o file. The .j3o file is then saved in the model folder so that it can be loaded inside the robot simulation.

There are 3 core objects within the virtual robot; chassis, wheel, and neck. Each of the core object is programmed part by part than at the end attached together to build the virtual robot. A camera sensor is attached at the neck of the virtual robot and the scene is displayed at the small window frame during the simulation. Moreover, a torch light programming is attached on the chassis of the virtual robot. With the torch light, the virtual robot is able to operate in dark environment.
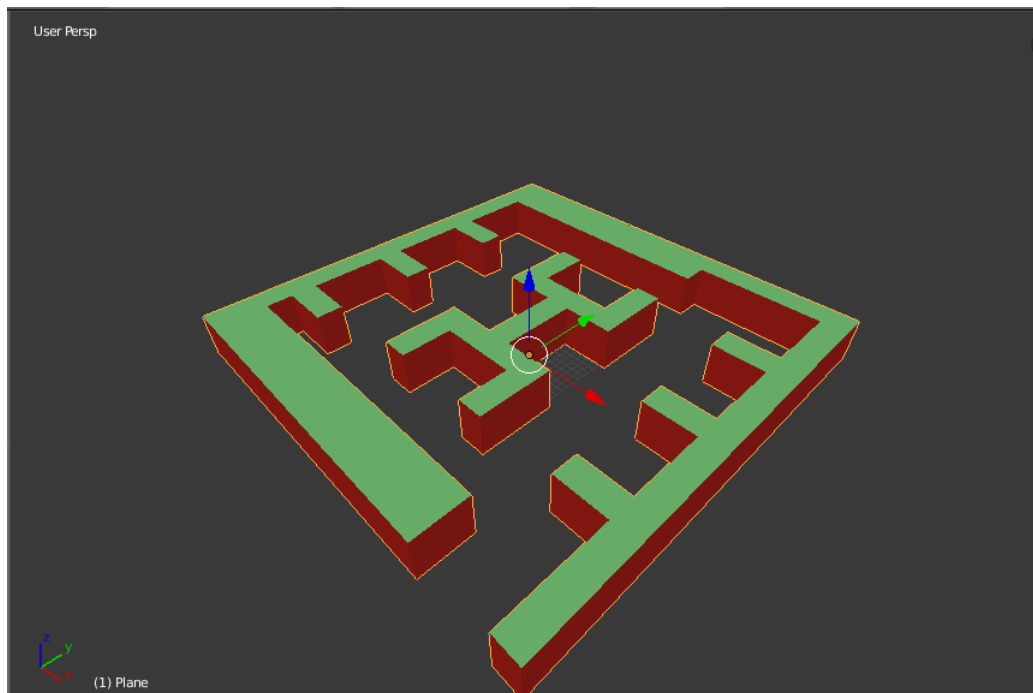
Figure 7. Indoor environment in Blender.
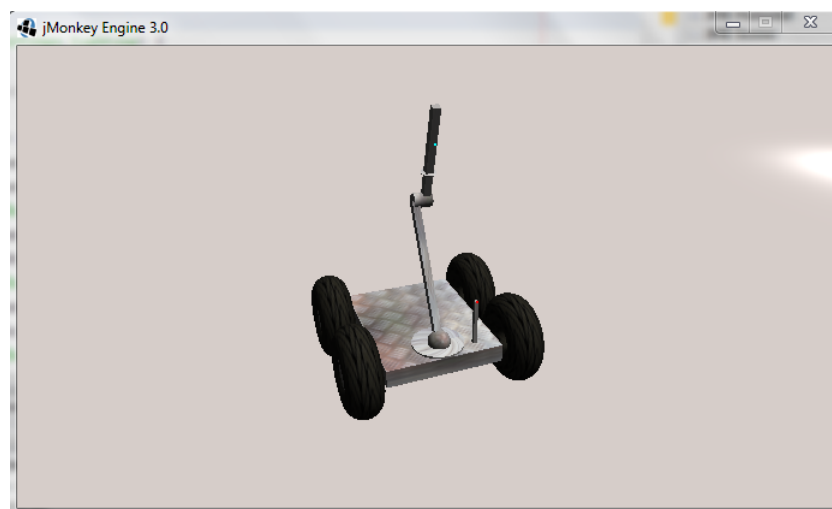
Figure 8. Virtual robot in JmonkeyEngine.
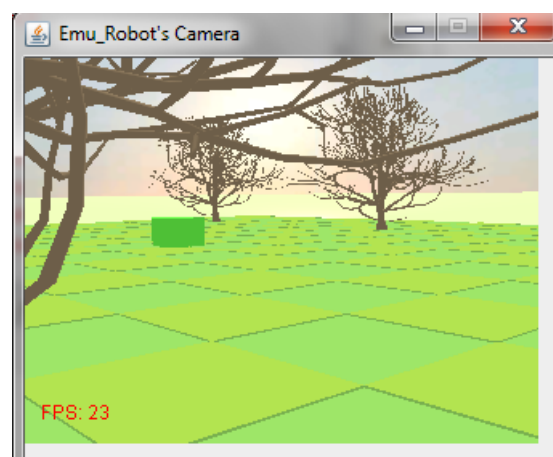
3.2. Mobile App Controller

The app is developed using MIT App Inventor Software which utilize the Android device Bluetooth module to control either from the input of touch button or the using the Accelerometer by tilting the Android device.In the app it contains two mode which is the "Button Mode" and "Accelerometer Mode" which is shown in Figure 5 and Figure 6 respectively. In 'Accelerometer Mode" ,the Android device is needed to be tilted and press on the "ON" button to send the input to the mobile gripping robot. The movement of the robot will be synchronous with the tilting of the Android device each time the "ON" button is pressed.While, in the "Button Mode" the Android device will read the input from the touch screen to send specific input to the mobile gripping robot. In this mode the control of the gripper is included
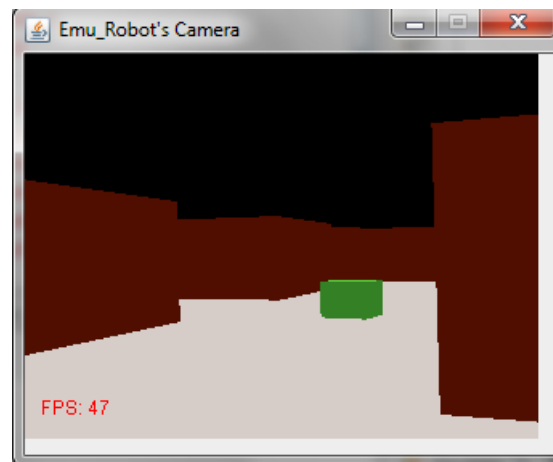
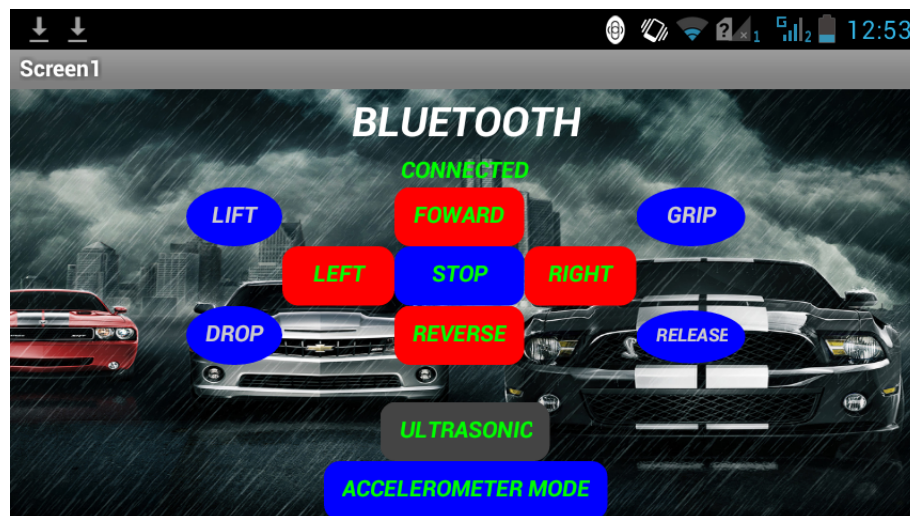**Figure 9.** Camera display (Indoor Bright).
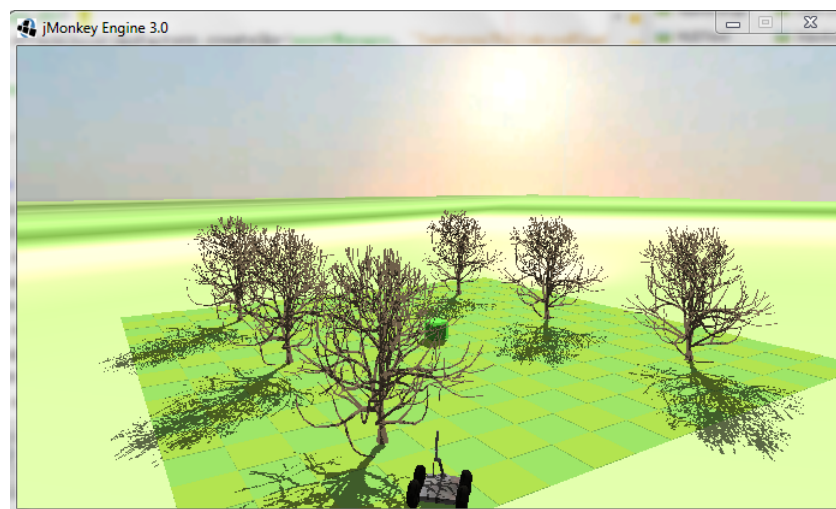
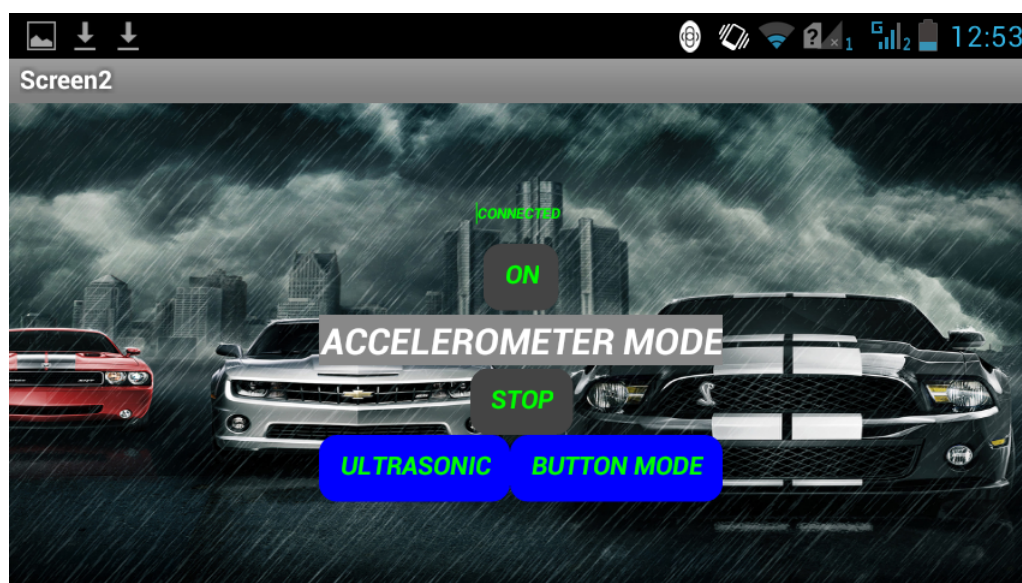**Figure 10.** Environment in JmonkeyEngine.

**Figure 11.** Camera display (Outdoor).

Figure 12. Mobile application: Button Mode.

Figure 13. Mobile application: Accelerometer Mode.

comparing to the accelerometer mode which only control the movement of the robot . The button for lift or drop the gripper and grip and release the gripper is available. While, in this mode also included the detection of the ultrasonic sensor which is used avoiding obstacles. With the ultrasonic turned on , the mobile gripping robot will automatically abort the input order to move forward and turn either way by depending on the reading from the sensors.

The robot is control system is developed using Arduino and ultrasonic sensor with Bluetooth module is added on to the mobile gripping robot. The addition of Bluetooth module with specific MAC address is used in the programming of the Android app so that the app will connected to mobile gripping robot automatically without needing the user to search or scan for the available Bluetooth connection. While, the ultrasonic sensor is used for detecting the obstacles which come across by the robot so that it can avoid the obstacles. The flow of the robot control is shown in Figure 7.
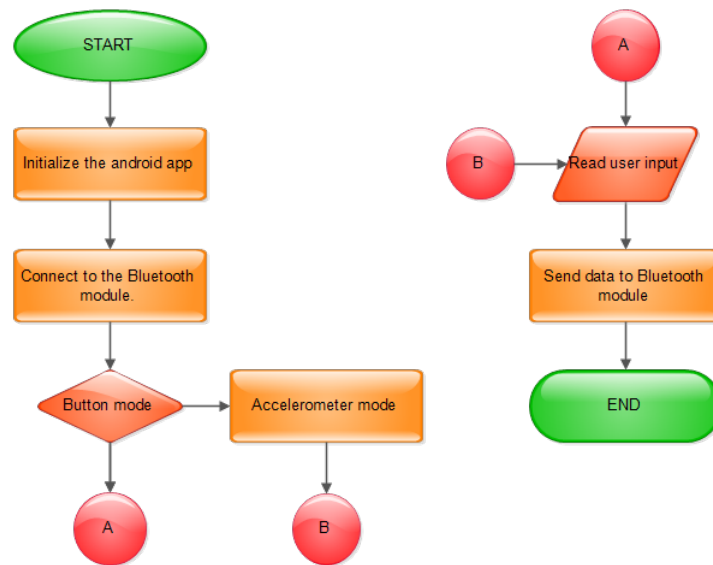
Figure 14. Flow of the robot control.

3.3. Image Processing

Firstly, the users need to identify the colours and the shape of the targeted object (rubbish). The object tracking image processing by adjust the colour RBG. Besides that, the changing of light intensity; histogram HUE setting. Furthermore, the technique of morphology method to erode and dilate the unwanted small particles or distortion pixels to ensure the image was fully thresholded. Furthermore, the technique of canny edges detector and work together with the ultrasonic sensor to find and to avoid the obstacles and walls. Next, the robot movement will run synchronously with the image processing.

The robot may automatically or be manually moved to seek for the targeted object. Besides that the users will be able to watch and monitor the whole process of the robot moment with a live streaming webcam from user's computer or through local host system. The users can record the video or capturing the image and upload to the database for further investigation. The users also can set the robot to be autopilot mode or manual pilot mode.

In addition, the robot has installed a robot arm and gripper to grip and pick up the object. Since, this project is only a prototyping robot, the gripper is not strong enough to grip the object's weight heavier than 5kg. But the concept of the robot is to show that, the users will be able to use the robot for victims rescue during disaster, for picking radioactive materials, or to test in mitigation strategies environment and be used in manufacturing assembly.

After the targeted object was detected via the camera, the image processing will filter the image threshold method and compared with SURF methods to make the result of accuracy become more accurate and precise. When the targeted object is confirmed, the robot will automatically move to the pinpoint of the object and pick up the object by using the gripper which installed in front of the robot.

4. Conclusions

We demonstrated that with simple Android and Arduino devices and addition of an Android app, it can be used to control a mobile gripping robot. There have been other projects on controlling mobile robot using Android device but the main purposes of this project is to show how open source hardware (Android, Arduino) and software (jMonkey, jmeSim) platforms can be integrated to create a framework for teaching and learning as well as research.

The robot simulation was successfully run in JmonkeyEngine. Different environment was programmed with different scene inside simulation. The robot simulation was able to access the real

time video from external webcam for navigation and object detection.

The robot was able to detect the targeted object successfully and can identifying the position of the real robot and the visual robot. Users can monitor the robot via live streaming using a PHP server.

Acknowledgments

References

[1] Haber A, McGill M and Sammut C 2012 Jmesim: An open source, multi platform robotics simulator *Proceedings of Australasian Conference on Robotics and Automation (December 2012)*
[2] Lim H, Park J, Lee D and Kim H J 2012 *Robotics & Automation Magazine, IEEE* **19** 33–45
[3] Pokress S C and Veiga J J D 2013 *arXiv preprint arXiv:1310.2830*
[4] Puttemans S, Howse J, Hua Q and Sinha U 2015
[5] Blain J M 2014 *The Complete Guide to Blender Graphics: Computer Modeling and Animation* (CRC Press)
[6] Galeriu C, Edwards S and Esper G 2014 *The Physics Teacher* **52** 157–159
[7] Subhan F, Hasbullah H, Rozyyev A and Bakhsh S T 2012 Analysis of bluetooth signal parameters for indoor positioning systems *Computer & Information Science (ICCIS), 2012 International Conference on* vol 2 (IEEE) pp 784–789
[8] Katz D, Venkatraman A, Kazemi M, Bagnell J A and Stentz A 2014 *Autonomous Robots* **37** 369–382
[9] Košecká J, Marchand E and Corke P 2015 *The International Journal of Robotics Research* **34** 399–401
[10] Al-Khammasi S, Qayym A and Sobh T M 2015