COMPUTERIZE REMOTE CONTROL CAR

MUHAMMAD SHAFIQ BIN MOHD SHABERI

B. ENG. (HONS.) MECHATRONICS ENGINEERING
UNIVERSITI MALAYSIA PAHANG

MUHAMMAD SHAFIQ BIN MOHD SHABERI B. ENG. (HONS.) MECHATRONICS ENGINEERING 2016 UMP

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author’s Full Name : ________________________________

Identification Card No : ________________________________

Title : ________________________________

____

Academic Session : ________________________________

I declare that this thesis is classified as:

□	**CONFIDENTIAL**	(Contains confidential information under the Official Secret Act 1972)
□	**RESTRICTED**	(Contains restricted information as specified by the organization where research was done) *
□	**OPEN ACCESS**	I agree that my thesis to be published as online open access (Full text)

I acknowledge that Universiti Malaysia Pahang reserve the right as follows:

1. The Thesis is the Property of University Malaysia Pahang.
2. The Library of University Malaysia Pahang has the right to make copies for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

________________________________ ________________________________

(Author’s Signature) (Supervisor’s Signature)

____

Name of Supervisor

Date: ________________ Date: ________________

COMPUTERIZE REMOTE COONTROL CAR

MUHAMMAD SHAFIQ BIN MOHD SHABERI

Report submitted in partial fulfilment of the requirements
for the award of the degree of
Bachelor of Engineering in Manufacturing Engineering

Faculty of Manufacturing Engineering
UNIVERSITI MALAYSIA PAHANG

June 2016

SUPERVISOR'S DECLARATION

I herewith declare that I have checked this report and in my opinion, this report is sufficient in term of scope and quality for the award of the Bachelor of Mechatronic Engineering.

Signature:

Name of Supervisor: Dr. Ng. Liang Shing

Position: Senior Lecturer

Date:

STUDENT’S DECLARATION

I here with declare that the work in this report is my own except for quotations and summaries that are punctually acknowledged.

Signature:

Name of Student: Muhammad Shafiq Bin Mohd Shaberi

Student ID: FB11058

Date:

ACKNOWLEDGEMENT

Alhamdulillah, thanks to Allah SWT, because with His willing and for all blessing for giving me a good healthy and patience during completing this Final Year Project.

First of all, I would like to express the deepest appreciation to my supervisor Dr. Ng Liang Shin for her attention and guidance during the period to complete this task. Without him guidance and attention, this project would be not accomplishing.

I would like to thank University Malaysia Pahang for allowing me to study and complete this course in here. I would also to thanks to Faculty of Manufacturing Engineering for providing the facility for me to complete the entire project. Also not forget to all lecturer and staff of Faculty of Manufacturing Engineering, who gave me the possibility to complete this report.

Lastly, a thank you to my parents, family, friends and other for their cooperation, full support and encouragement from the beginning of this project until the end of the report completion. They all give me spirit to keep me going.

ABSTRACT

Computerized remote control car is the one project in the robotic field. This project is known as one of the robot platform that can be more develop into advance autonomous robot. The main feature of this project is it can be controlled wireless by using Android device. Android device nowadays have built in Bluetooth technology and one of the main our daily needs. This Bluetooth technology has been one of the most platforms in smartphone that was use in android system to develop an application to control all kind s of peripheral device like servo, DC gear motor, and others. RoboRemo software is the one of the software that can be use in developing Android app. The main objective of this project is to control the RC car and robot arm by using Android device through the Bluetooth communication. RoboRemo software will be installed on the Android device to create an application that can control the RC car and robot arm through Bluetooth. The implementation of the Android device for controlling RC car will solve a frequency problem that occurs while using RF transmitter. As the implantation of the Android device as controller of RC car can be apply by developing application using RoboRemo software.

ABSTRAK

Kereta kawalan jauh berkomputer adalah adalah salah satu projek yang terkenal didalam bidang robotik. Projek ini dikenali sebagai salah satu platform robot yang terbaik sebelum dipertingkatkan lagi kebolehannya untuk menjadi robot autonomi. Ciri-ciri utama projek ini adalah ia boleh dikawal tanpa wayar dengan menggunakan jenis perisian seperti peranti Android. Kebanyakan peranti Android pada masa kini telah dilengkapi teknologi Bluetooth dan menjadi salah satu peranti yang utama di dalam keperluan harian. Teknologi Bluetooth telah menjadi salah satu platform yang telah diiktiraf untuk digunakan dalam membina applikasi yang dapat mengawal perisian seperti DC gear motor, servo motor, dan lain-lain. Perisian tersebut adalah RoboRemo dan merupakan salah satu perisian yang boleh digunakan dalam membangunkan aplikasi Android. Objektif utama projek ini adalah untuk mengawal kereta RC dan lengan robot dengan menggunakan peranti Android melalui komunikasi Bluetooth. perisian RoboRemo akan dipasang pada peranti Android untuk mewujudkan satu aplikasi yang boleh mengawal kereta dan robot lengan RC melalui Bluetooth. Pelaksanaan peranti Android untuk mengawal kereta RC akan menyelesaikan masalah yang kerap berlaku semasa menggunakan pemancar Radio Frequency. Peranti Android boleh diimplimantasi sebagai alat kawalan kereta dengan menggunakan perisian RoboRemo.

TABLE OF CONTENTS

Page

CHAPTER 3 METHODOLOGY

LIST OF TABLES

LIST OF FIGURE

CHAPTER 1

INTRODUCTION

1.1.1 OVERVIEW

RC car is car powered by battery or gas that can be controlled from a certain distance by using a specialized device such as RF transmitter, Bluetooth, or remote. The meaning of “RC” is to mean both “remote controlled” and “radio controlled”. Remote controller is the vehicle that connected to their remote by wire while the radio controlled is the vehicle operate by using wave frequency at certain frequencies. It has a transmitter for transmitted data from remote control and the data will collect by the receiver in the car. RC car has 7 control movement functions; forward, left forward, right forward, backward, right backward, left backward, and stop.

Android is an operating system that is used in smartphone like Samsung, Lenovo, and others. This operating system is developed by Google based on Linux kernel source and it's designed to enable the user to touch, tap, swipe, and pinching to control the interface function of the phone.

1.1.2 PROJECT BACKGROUND

This project is to build a car that has a differential drive, multiple level of speed, control arm and gripper and also will be controlled by using a Bluetooth device that connected to the android phone. This car will use one DC motor and three servo motors. DC motor used to move the car forward and backward while the first servo motor is used to control the front tire to move left and right, second servo is to control the movement of the arm up and down and the last servo is to control the gripper. The command to control the movement of the car will send from the android device that connected to the car by Bluetooth.

This has been the new way to of controlling the RC car by using Bluetooth. Almost all devices nowadays has built in Bluetooth functionality such as smartphone, tablet, and laptop. The rapidly increasing number of smartphone user has made the Android operating become top of the chart of number user in the operating system. This new way that was controlled RC car using the Bluetooth system has a lot benefit because it's free and easily can be developed by using different kind of software.

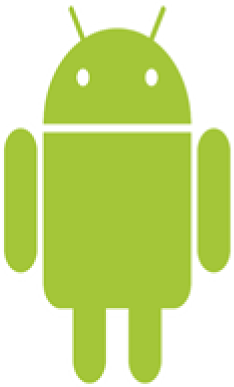
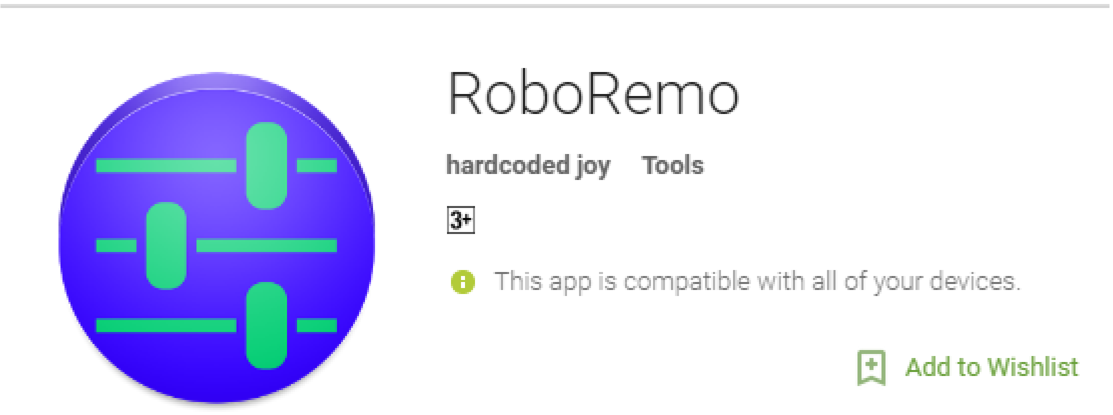


FIGURE 1.1: Android operating system.

Source: www.android.com



RoboRemo

hardcoded joy Tools

3+

This app is compatible with all of your devices.

FIGURE 1.2: RoboRemo software

Source: http://www.roboremo.com

1.1.3 PROBLEM STATEMENT

In recent years, the use of remote control car among school children and adult are increasing. Mostly the remote control car are used as a toy to fill their leisure time, but some people use it as a tool to investigate, tool for their work and to do an experiment. For children that use a remote control car as a toy sometimes cannot control the their car when they play more than one remote control car. This problem occurs because there is some frequency disturbance.

The controller, which is usually used is by using the laptop to control the robot wirelessly either through Wi-Fi or Bluetooth. However, using laptop has some major disadvantages which are laptop is not portable which makes the user inconvenient compares to using the low powered tablets or smart phones as Android devices is lighter comparable to a laptop.

Besides that, nowadays the famous tool to control the wireless communication is radio frequency control. It will use certain frequency to control the car. This type of control has a advantages in term of range, it will cover a further range than Bluetooth. However, this frequency, sometimes has disturbed when the other device uses same range frequency which make the communication to be faulty.

A stable RF control is not developed to be easy as it is prone to noise. The Wi-Fi module is also not suitable to implement to control an RC car because this module is not cheap. Therefore the Bluetooth module is more suitable to be implemented as a preliminary stage of controller for the trial stage before it developed to more complex and advance level of remote controller.

1.1.4 OBJECTIVE

The objective of this project is:

- To develop an system that can control the speed of RC car .
- To develop an Android app using RoboRemo software and control the movement of the car by using android device.
- To control the robot arm and gripper by using Android device.

1.2 PROJECT SCOPE

- Using Arduino to connect the remote control to Bluetooth module.
- Develop an android app using RoboRemo application software to control the movement of the car, robot arm and gripper.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

This chapter is about the review of the researches that related to the project. It summarizes and discussed the related knowledge about the Android app and Bluetooth project in order to learn the theory that will be applied in this project.

2.2 Visual Navigation of RC Vehicle using Wireless Video

Steven Northrup and Christopher Paros proposed to build an RC vehicle to track, predetermined object using wireless video feedback. This project involves image processing, control theory, circuit design, PC interfacing, and software design. The system required to control the radio control (RC) vehicle for its speed and direction using monocular video. PC will capture the signal from video monocular by using a video capture card and the image was process to handle RC controller to control the speed and direction of the vehicle. [2]

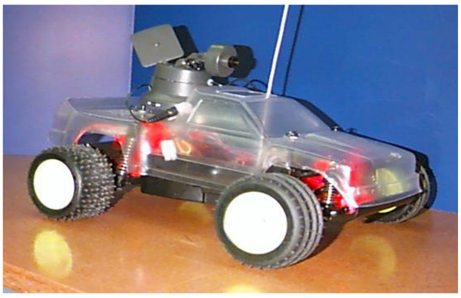


FIGURE 2.1 : RC vehicle with wireless camera on top.

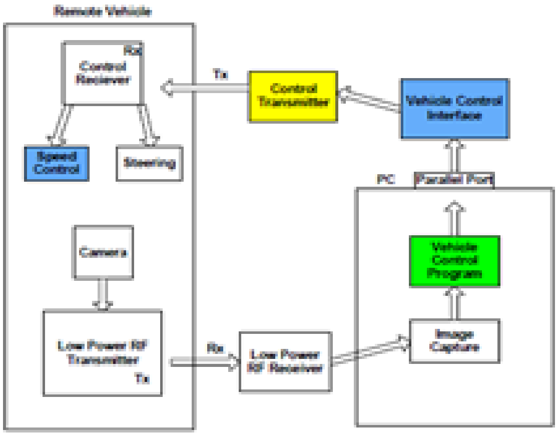


FIGURE 2.2: Block diagram

Figure 2.2 shows the block diagram of the system illustrate the component. The components that were purchased is not shaded but the yellow box are components that modified to fit the project application. The blue box are the hardware component design and built, and green box is software that programmed for the vehicle.

A video from the vehicle will send a signal via wireless transmitter to PC. The video will capture and processed by the vehicle control program. The image will process the color segmentation and size and center of mass algorithm. To determine the target distance and proper direction is depend on the size of mass algorithm by using monocular vision and center of mass algorithm. The parallel port is used to transmit commands to handle the speed and steering based on the data from size and center of mass algorithm. The joystick potentiometers will determine the analog signal and interpret it by the transmitter, then convert to RF- control of the vehicle.

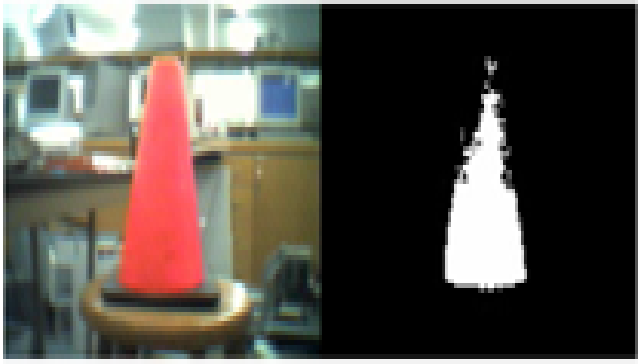


FIGURE 2.3: Image processing of he color segmentation.

When the control signal is received, it will cause the speed and steering servo to turn. The speed control circuit is designed to make it more sensitive level of control of lower speed. They also develop several GUI in order to calibrate the image, steering, and speed.

2.3 AndroRC: An Android Remote Control Car Unit for Search Missions

This research is presenting an inexpensive remote control car platform that can be controlled via a smartphone running on an Android application. This remote car was developed to avoid obstacles that were visible to the user. The equipment was use is ultrasonic sensor, a camera, a Bluetooth receiver, two 9v batteries, two Arduino microcontroller, and wifi transmitter. The movement of this car is controlled by android application to determine the direction (forward, backward, left and right). Figure 1 illustrates the block diagram of the system.

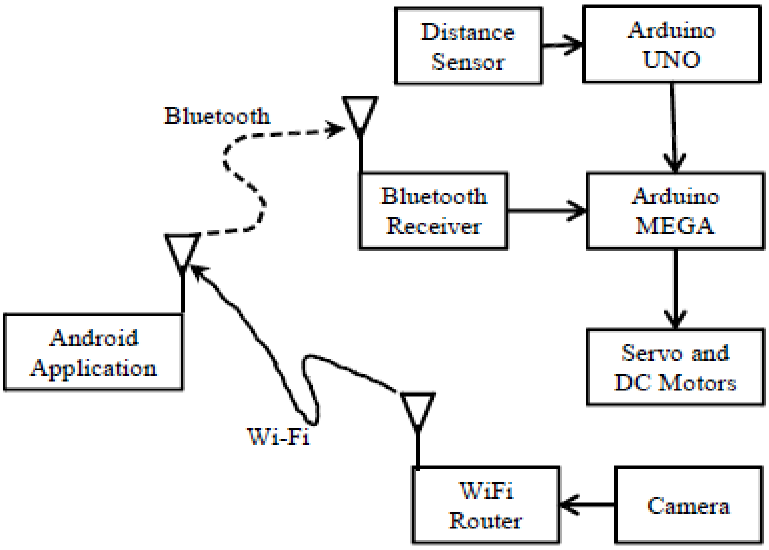


FIGURE 2.4: The block diagram of the AndroRC.

a) The RC Unit

The AndroRC in based on Tamiya 70112 Buggy Cay Chassis set that the speed can be adjusted for high and low input. The front of the car was modified to placed extra servo motor that will control the front wheel to left and right.

b) Microcontroller

They use Arduino Mega as the AndroRC microcontroller. This microcontroller will generate pulse to the DC motor and servo motor depend on the command from user to control the movement forward, backward, left and right.

Second microcontroller that they use is Arduino UNO. This Arduino is connected to a distance sensor.

c) Ultrasonic sensor

This sensor is mounted in the front of the AndroRC to detect the obstacle. When it detects an obstacle at a certain distance it will send pulses to the microcontroller to measure the distance of the obstacle.

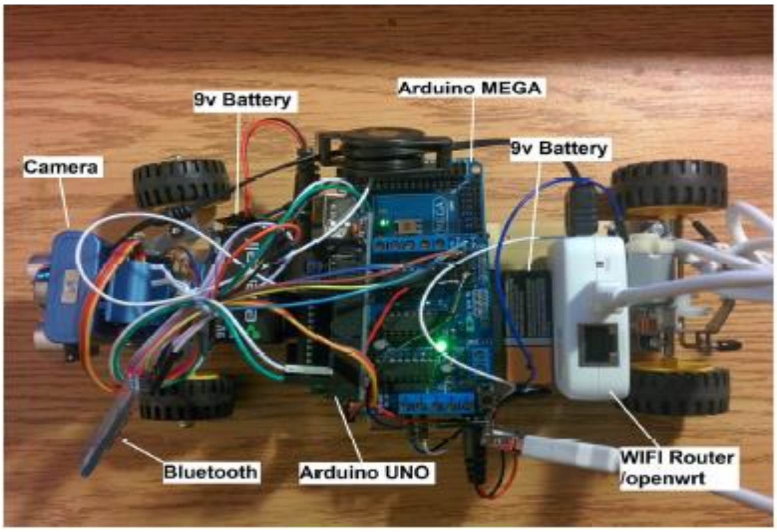


FIGURE 2.5: Top view of the AndroRC shows the camera, Bluetooth, 9-V batteries, Arduino boards, and WIFI router.

d) Bluetooth receiver.

This device is mounted on top of the MEGA board and communicateds through the 22.4 GHz ISM band. It will receive the directon command from the android device and send it to the microcontroller. When the AndroRC is powered up this bluetooth receiver(slave) will pair with the master(transmitter) on android device.

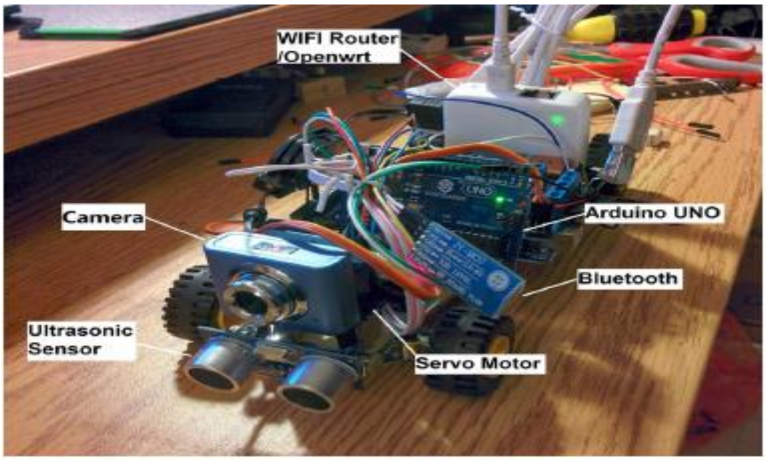


FIGURE 2.6 : Front view of the AndroRC show the camera, ultrasonic sensor, Bluetooth, servo motor, Arduino UNO board, and WIFI router.

e) The Application for android phone.

They developed an application that can be installed on Android smartphone to control the AndroRC. This application will call information of the orientation of the sensor to obtain the angle of the phone and a default coordinate. This application can translate the orientation data to propulsion commands which is forward, backward, left, and right.

2.4 Research On A Remote Control Vehicle

The purpose of this research was to efficiently control DC motor by using infrared remote control. The motor will control to accelerate, decelerate, reverse, and able to turn left and right. To avoid a crash, there are sensor in the RC vehicle. The sensor that will use is ultrasonic sensor and if it detects the obstacle, motor will stop immediately [4]. These sensors will make the car come to a complete stop when it is within 20 cm of crashing into obstacles. They choose 20cm because it will give enough time for the car to a complete stop. The car will no longer move until the user picks it up and places it somewhere beyond 20 cm away from any object. This crash detection not only prevents damage to the vehicle, but it also protects the safety of children playing with the car.

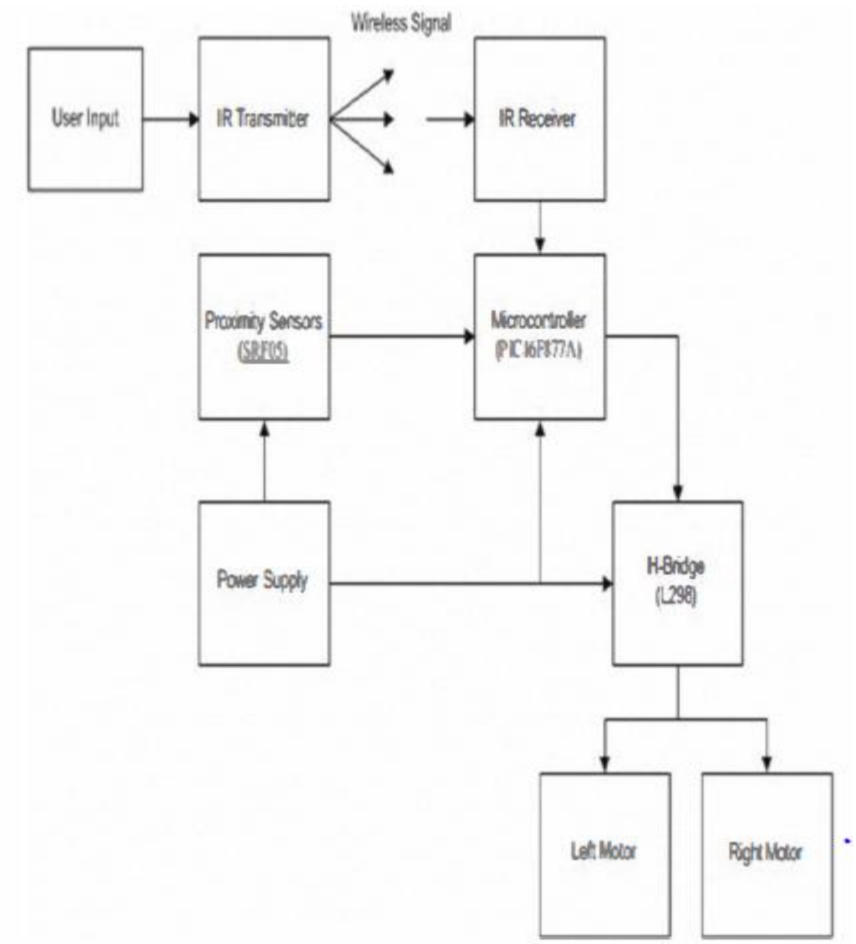


FIGURE 2.7: Block diagram of system flow.

This project has four main features. The first is the user can control the speed of the car, either or not the car accelerates or decelerates, control direction to the left or right, and stop. Second features, is to prevent damage to the car by using crash detection sensor. They also want to improve the efficiency of the car and that was the third feature of this project. The last feature is this car can move forward and backward.

a) IR Transmitter/Receiver

The signal from the receiver is mapped to the micro controller by using reverse engineering. There are two output voltage in the receiver, one for right motor to accelerate or reverse and the other from left to turn left and right.

b) Microcontroller

The microcontroller will receive 2 input voltages from the receiver as signal and then send a proper duty cycle to H-Bridge after it built PWM driver.

c) Proximity Sensor

The proximity sensor is the sensor that will detect wall or another object to avoid vehicle from crashing. When it detects the vehicle about to crash it will send signals to microcontroller to reverse the output of PWM before sent to H-Bridge.

d) Drive System

They use H-bridge as drive system. This is because the H - bridge can have a higher voltage range and use two DC motor control variable speed and bi-directional.

e) Power Supply

There are three different voltage levels. First 9V for IR transmitter, second 12V and 5V for H-bridge and third 5V for Pic, ultrasonic sensor and op-amps.

2.5 A Bluetooth Toy Car Control Realization by Android Equipment

This paper is about research for control a remote toy car by an android mobile platform combine with Bluetooth technology. The mobile platform will send a command to the toy car by using Android Bluetooth system.

They're a few major part in this project, which are hardware part and mobile software part. On the hardware part, it uses the remote toy car equipped with a two stepper motor and motor driver. The toy car is installed with IR sensor at the front of the car. On the software part, it uses the Android 2.0 with four levels of development. The bottom level is core of Linux, the library and the Dalvik VM follow. The top level is application programs.

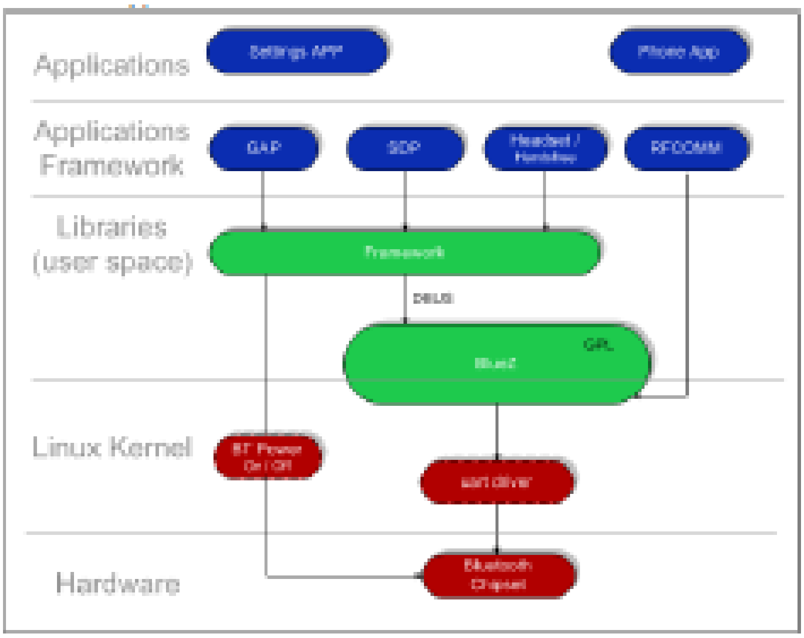


FIGURE 2.8: The Android Bluetooth Protocol

a) Bluetooth communication

Android Bluetooth protocol stack that can support GAP, SDP and RFCOMM standard and authorized by SIG (Bluetooth Special Interesting Group) is called BlueZ. This Bluetooth communication can receive data and instructions by using Android platform when the device is paired to the other Bluetooth devices.

The communication step between Android application and Bluetooth APIs is:

- Scan for other devices
- Query the local Bluetooth adapter for pair Bluetooth devices
- Establish an RFCOMM channel
- Connect to other devices
- Transfer receives data from other Bluetooth device
- Manage multiple connection

b) BlueZ

This system will provide support to the core Bluetooth layers and protocol. It is flexible and efficient to be implementation.

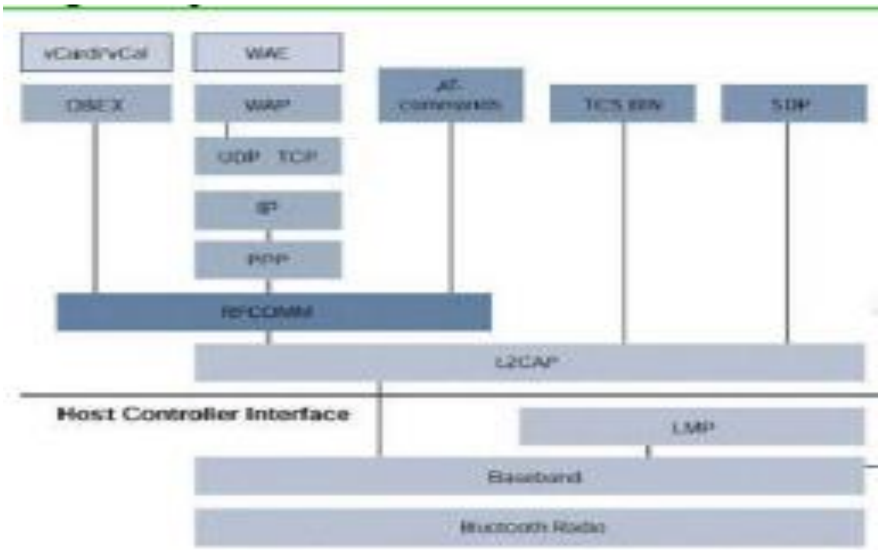


FIGURE 2.9: BlueZ structure

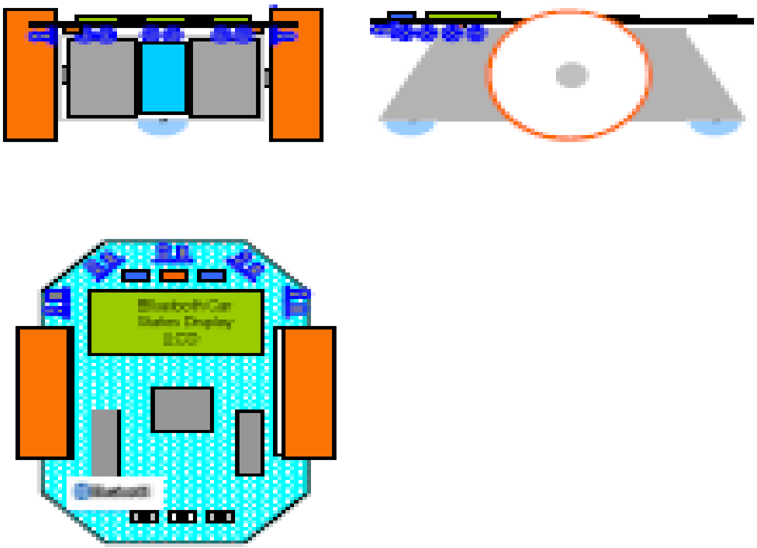
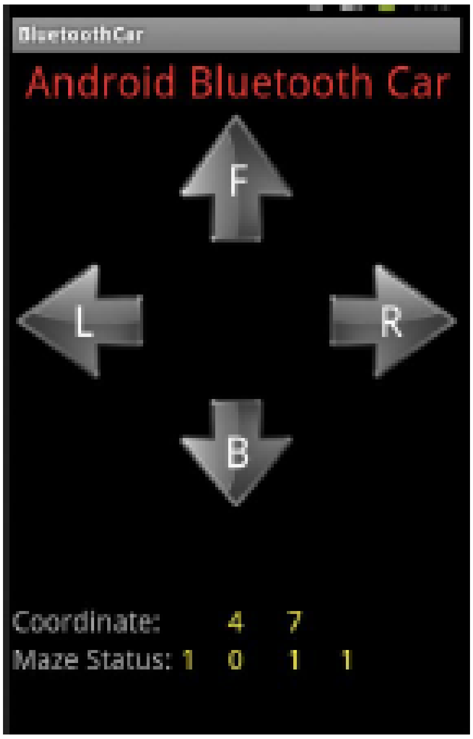


FIGURE 2.10: The Bluetooth remote controlled car.



Android Bluetooth Car

Coordinate: 4 7

Maze Status: 1 0 1 1

FIGURE 2.11: Software interface

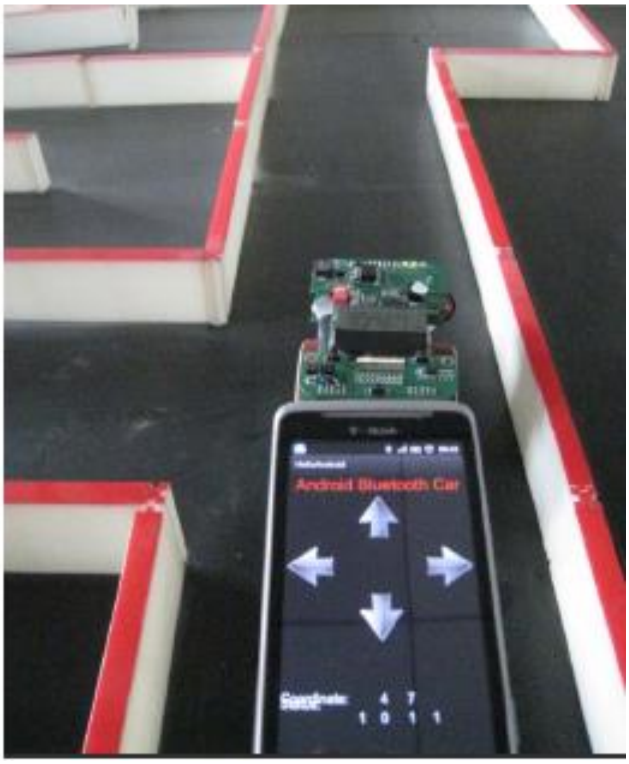


FIGURE 2.12: Experimenting in a maze

CHAPTER 3

METHODOLOGY

3.1 INTRODUCTION

This chapter is to discuss about the methodology that will use in to conduct the project. It is very important to know the suitable method to apply to this project to make it work properly. Based on the literature review, it can be concluded that many criteria need to be considered in computerized the remote control car.

In this project, we are going to create a new remote control by using Android software that is RoboRemo application and Arduino IDE which is used for programming the Arduino Mega.

3.2 MATERIAL PREPARATION

Material preparation is the first step of the experiment. The material that will be used in this project is:

1. Microcontroller
2. DC gear motor
3. 3 × Servo motor
4. Bluetooth module

5. 1 × breadboards to place the Bluetooth module and the motor driver to make it easier for handling and wiring.
6. Wires and alligator clip to make the connection.
7. Dual H-Bridge motor driver-L293D.

3.3 HARDWARE

3.3.1 BLUETOOTH MODULE

The communication between Android and the RC car is by using the Bluetooth module which is HT Bluetooth Module HC-05. Bluetooth system is easy to use compared to other type of communication because it has no interference when operating. This module communication is designed for transparent wireless serial connection setup. It uses CSR Bluecore 04-External single chip Bluetooth system with CMOS technology. The serial communication will transfer data through the RX/TX pin using wavelength UHF radio waves in the ISM band from 2.4 to 2.485 GHz. This module also can support the baud rate 9600,19200,38400,57600,115200,230400 and 460800. In a proper connection network the party must operate at the same baud rate.

To establish the connection between this module and party, the pairing protocol has to be done with the pairing key of "1234". After that the connection can be initiated. Other than that, the Arduino microcontroller also has the RX/TX pin that can conduct serial communication. This will make the Bluetooth module easy to interface with the Arduino microcontroller.

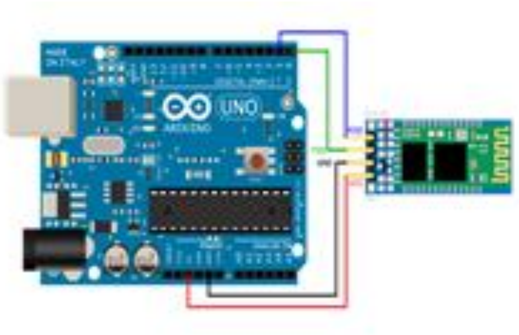
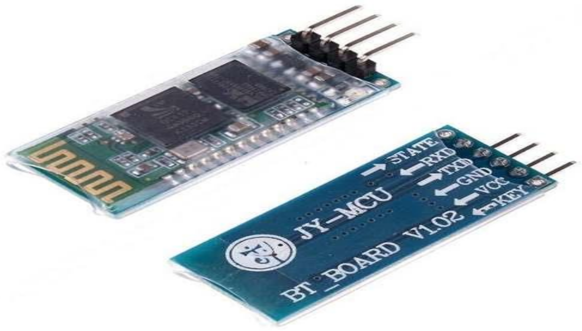


FIGURE 3.1: Bluetooth module connection to the Arduino board

Source: http://www.instructables.com

Table 3.1: Bluetooth module, HC-05 specifications

 STATE RXD TXD GND VCC KEY JY-MCU BT_BOARD V1.02
Features
• Typical -80dBm sensitivity • Up to +4dBm RF transmit power • UART interface with programmable baud rate • With integrated antenna • With edge connector

Key specifications

- Bluetooth protocol: Bluetooth Specification v2.0+EDR
- Frequency: 2.4GHz ISM band
- Modulation: GFSK(Gaussian Frequency Shift Keying)
- Emission power: ≤4dBm, Class 2
- Sensitivity: ≤-84dBm at 0.1% BER
- Speed: Asynchronous: 2.1Mbps(Max) / 160 kbps, Synchronous: 1Mbps/1Mbps
- Security: Authentication and encryption
- Profiles: Bluetooth serial port
- Power supply: +3.3VDC 50mA
- Working temperature: -20 ~ +75Centigrade
- Dimension: 26.9mm x 13mm x 2.2 mm

3.3.2 MICROCONTROLLER

In this project, the original controller of the RC car is replaced with the new microcontroller which is Arduino UNO. This microcontroller is based on Atmega328P and easy to use and it also has already built compiler which is Arduino IDE. It has 14 digital input/output pin includes PWM pin, 6 analog pins, USB connection and a reset button.

This Arduino microcontroller also is an open source project which means there a lot of example and reference can be obtained easily. The most important is this Arduino UNO microcontroller can establish the serial communication between the Bluetooth module and Android device.

Table 3.2: Arduino UNO specifications

Key specifications	
• Microcontroller	Atmega328P
• Operating Voltage	5V
• Input Voltage (recommended)	7-12V
• Input Voltage (limits)	6-20V
• Digital I/O Pins	14 (of which 6 provide PWM output)
• Analog Input Pins	6
• DC Current per I/O Pin	20 mA
• DC Current for 3.3V Pin	50 mA
• Flash Memory	256 KB of which 8 KB used by bootloader
• SRAM	2 KB
• EEPROM	1 KB
• Clock Speed	16 MHz

3.3.3 RC PLATFORM

The RC platform consists of an RC truck, which has a tough body frame to carry the new controller with the gripper attach. This RC truck also has a DC gear motor that has enough speed and torque and also have servo motor to control the direction of the car. This component is suitable and can be easily interfaced Arduino controller.

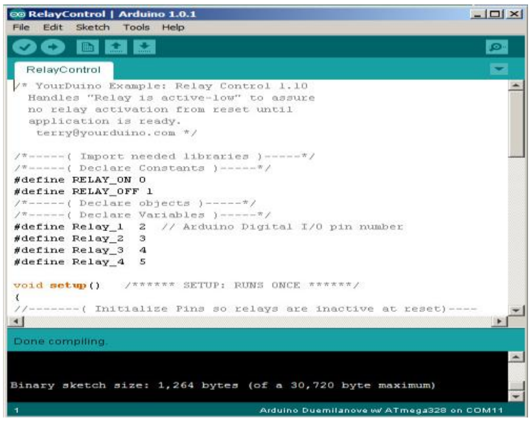
3.4 SOFTWARE

There are two software that was used in completing this project, which is Arduino IDE and RoboRemo. This software is easy to use to program the Arduino microcontroller and develop Android app respectively.

3.4.1 ARDUINO IDE

This software is open source software which is easy to write code and upload it to any Arduino board. The environment is written in Java and based on processing and other open source software. It can be easily downloaded from http://www.arduino.cc/en/Main/Software without paying anything.

On this project, this software is used to program the Arduino UNO board. This Arduino board is used to control all the components of the RC car which is DC gear motor and 3 servo motor. It also will read the input command that will send from the Android device and convert it to the output command to the specific component.



```
/* YourDuino Example: Relay Control 1.10
  Handles "Relay is active-low" to assure
  no relay activation from reset until
  application is ready.
   terry@yourduino.com */

/*-----( Import needed libraries )-----*/
/*-----( Declare Constants )-----*/
#define RELAY_ON 0
#define RELAY_OFF 1
/*-----( Declare objects )-----*/
/*-----( Declare Variables )-----*/
#define Relay_1  2  // Arduino Digital I/O pin number
#define Relay_2  3
#define Relay_3  4
#define Relay_4  5

void setup()   /****** SETUP: RUNS ONCE ******/
{
//-------( Initialize Pins so relays are inactive at reset)----
```

```
Done compiling.

Binary sketch size: 1,264 bytes (of a 30,720 byte maximum)
```

FIGURE 3.2: Arduino IDE.

3.4.2 ROBOREMO APP

RoboRemo software was originally developed by hardcoded joy and published on Google Play store and the latest update was on May 25, 2016. This software has 582k in size and required Android 2.2 and above. This software is a usable customizable remote control application intended mainly for electronics project.

Beside that, this software also can connect over Bluetooth (RFCOMM), Internet or WiFi (TCP, UDP), and USB (CDC-ACM, FTDI, CP210X). To connect over Bluetooth, a remote device must contain a Bluetooth serial adapter such as HC-05, HC-06, BlueSMIRF, etc. and microcontroller programmed to interpret command from RoboRemo. Furthermore, welcomed from this software are text strings, ending with a command ending, which if

LF character ‘\n’ by default. But this command can change depending on the user.

It is not recommended to set command ending in an empty string, because receiving

Commands from microcontroller to the app will not work properly. (App “will think” that the command ends after each received character). However the commands from app to microcontroller will be sent as expected (a button configured to send “ABC” will send “ABC”).

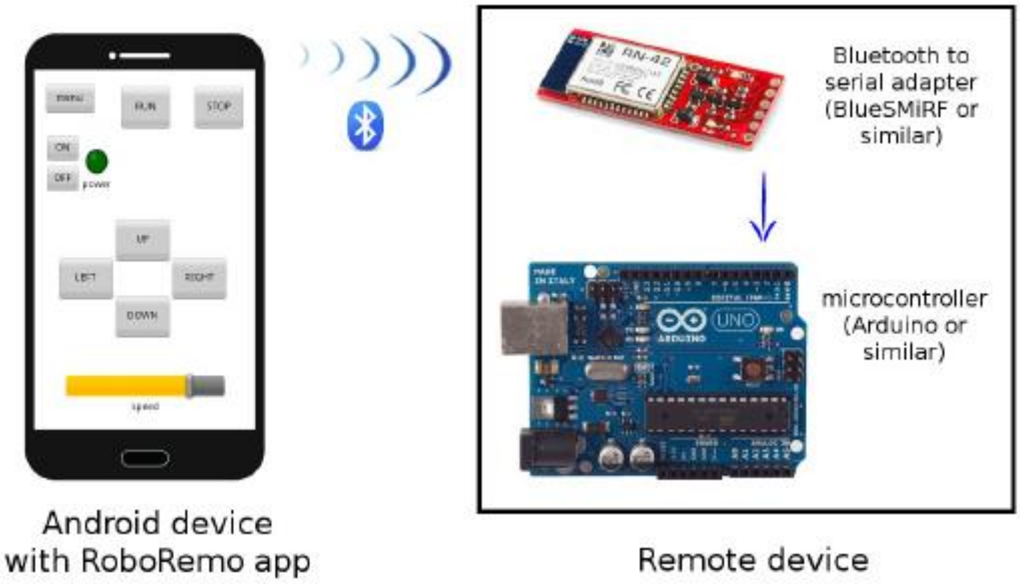
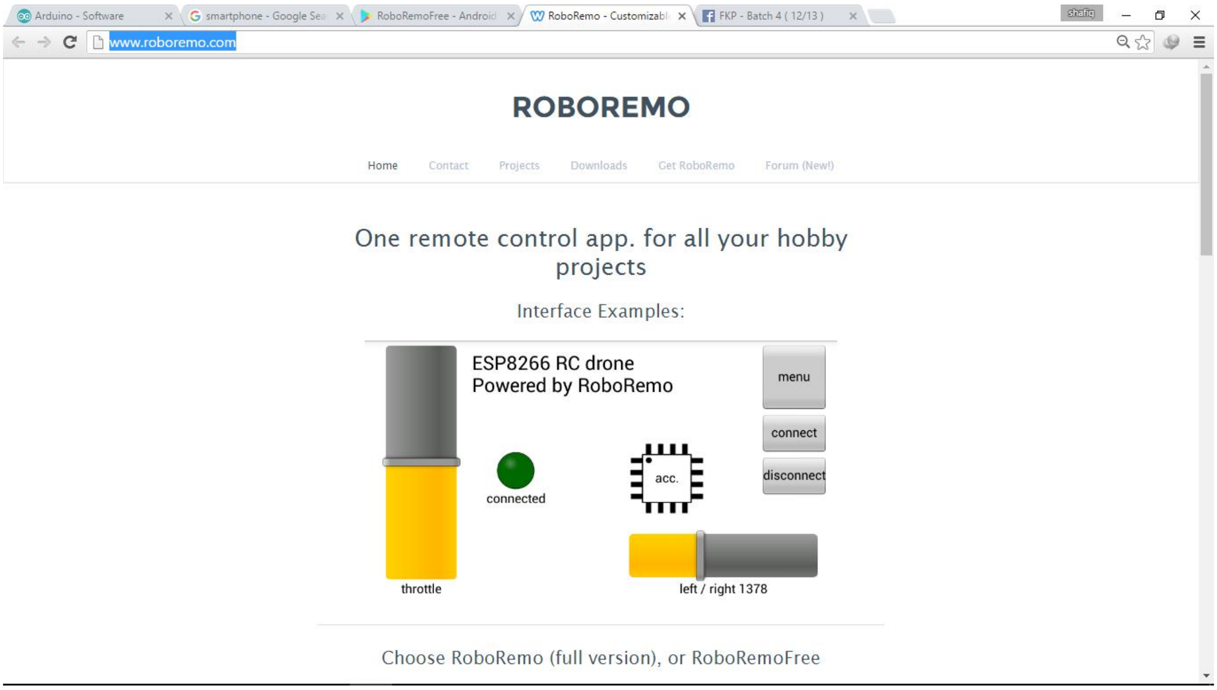

menu
RUN
STOP
ON
OFF
power
UP
LEFT
RIGHT
DOWN
speed
Bluetooth to serial adapter (BlueSMiRF or similar)
RN-42
microcontroller (Arduino or similar)
UNO
ARDUINO
Android device with RoboRemo app
Remote device

FIGURE 3.3: Android device with RoboRemo app interface to Remote device

Source : http://www.roboremo.com/



One remote control app. for all your hobby projects

Interface Examples:

ESP8266 RC drone
Powered by RoboRemo

Choose RoboRemo (full version), or RoboRemoFree

FIGURE 3.4: RoboRemo Home page

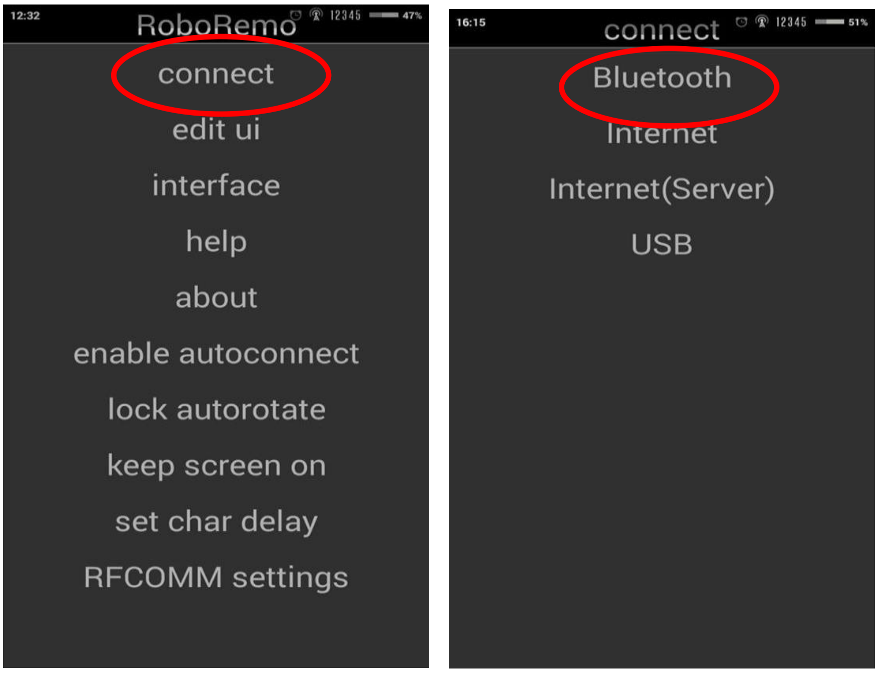
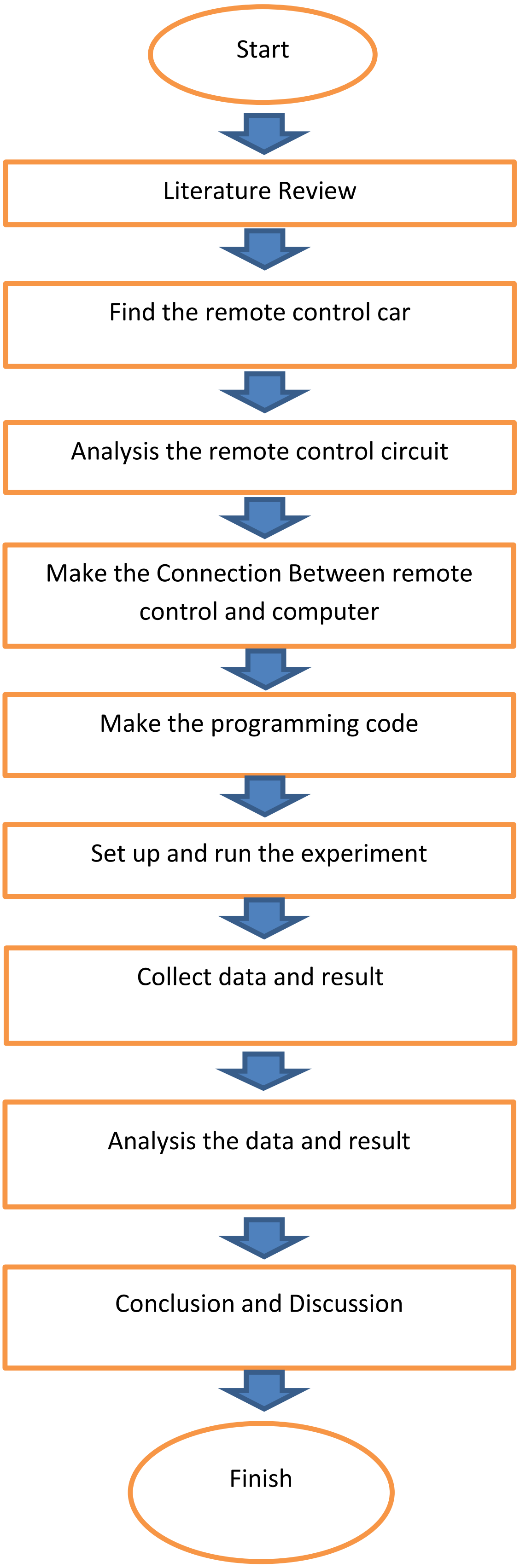


FIGURE 3.5 : Example how to make Bluetooth connection.

3.5 FLOW CHART


Start
Literature Review
Find the remote control car
Analysis the remote control circuit
Make the Connection Between remote control and computer
Make the programming code
Set up and run the experiment
Collect data and result
Analysis the data and result
Conclusion and Discussion
Finish

CHAPTER 4

RESULT AND DICUSSION

4.1 INTRODUCTION

This chapter is about the result and the outcome of the project. This includes the developed android app, the communication between Bluetooth and Arduino, and control the movement of the car and robot arm.

4.2 ANDROID APP

The app is developed using RoboRemo software to utilize the Bluetooth device with the Android smartphone to control the RC car either from the input of touch button or movement of the slider.

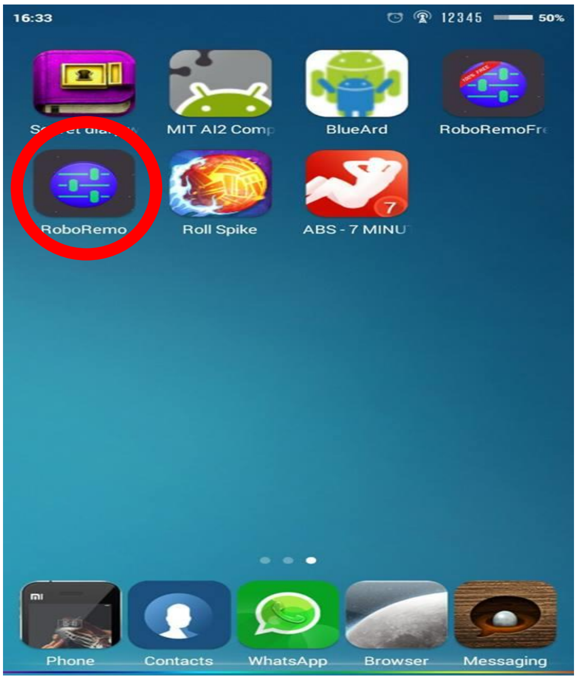


FIGURE 4.1 : Initiate the RoboRemo app.

After Bluetooth device on Android smartphone is turned on, initiate the RoboRemo app just to touch the icon on the smartphone screen as shown in Figure 4.1 and it will show the interface as shown in Figure 4.2.

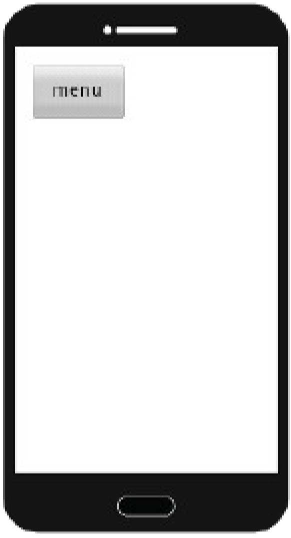


FIGURE 4.2: RoboRemo interface.

To connect the app to the Bluetooth module on the RC car it must be paired. After that touch the menu button and select the Bluetooth connection as shown in Figure 4.3. If the Android Bluetooth device is not turned on, it will show a request to switch on the Android Bluetooth as shown in Figure 4.5. Select the address of the Bluetooth module of the RC car.

On the RoboRemo app, it will show the interface to control the RC car with robot arm as shown in Figure 4.4. It has touch button to control the movement of the RC car and slider to control the movement of the robot arm.

On the touch button, the Android device will read the input from the touch button to send the specific command to the RC car. The touch button will control the direction of the car to move forward, backward, left, and right. It also can control the speed of the RC car which is set to 0, 100, 150, 200, and 255 as shown in Figure 4.4. When the button release the Android device will not send any command to the RC car.

While, for the slider function is to control the movement of the robot arm. It will control the robot arm to lift or drop the gripper and grip and release the gripper. The slider is set for maximum and minimum to limit the movement of the robot arm to prevent from interrupt command.

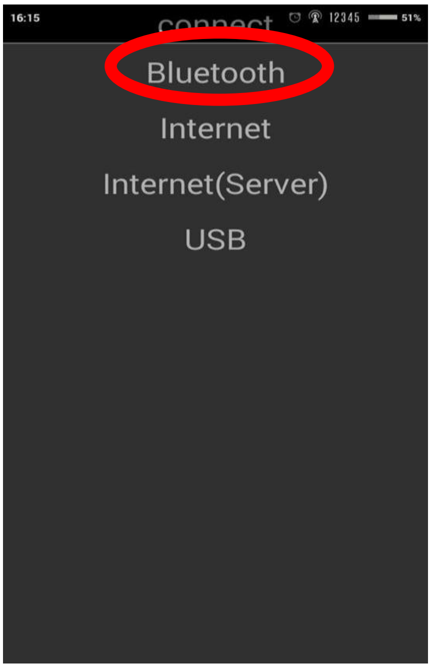


FIGURE 4.3 : Connection selection.

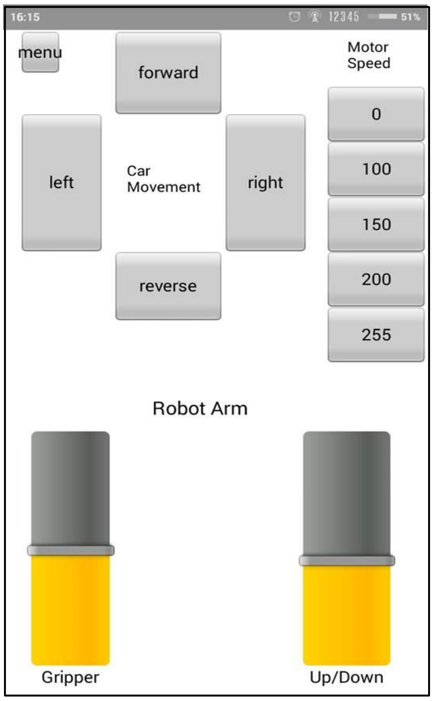
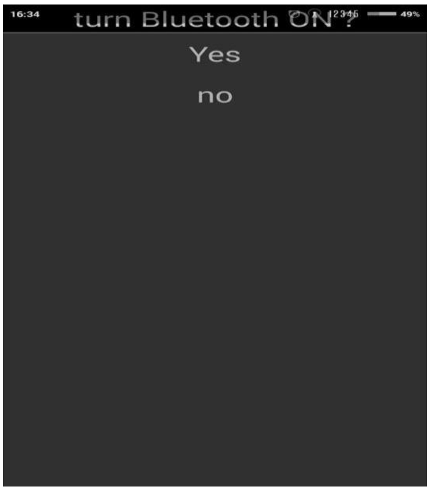


FIGURE 4.4: RoboRemo Interface to control RC car.

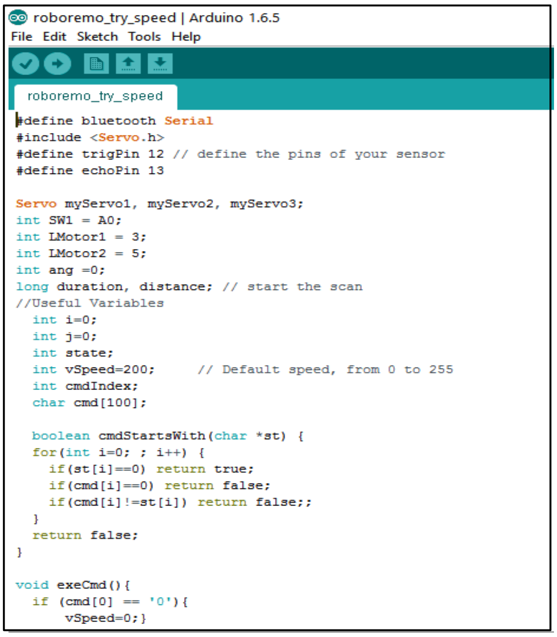


turn Bluetooth ON ?

FIGURE 4.5 : Turn on the Blutooth module of the Android device.

4.3 CONTROLLER OF RC CAR.

This controller is used Arduino is coded to send command to control the RC car as mentioned in the previous chapter. This Arduino will interact with the Bluetooth module to send and receive commands to control the RC car. The Bluetooth module (slave) has a specific address and it will pair to the Android Bluetooth device (master) to control the RC car and robot arm.


roboremo_try_speed | Arduino 1.6.5
File Edit Sketch Tools Help
roboremo_try_speed

```
#define bluetooth Serial
#include <Servo.h>
#define trigPin 12 // define the pins of your sensor
#define echoPin 13

Servo myServo1, myServo2, myServo3;
int SW1 = A0;
int LMotor1 = 3;
int LMotor2 = 5;
int ang =0;
long duration, distance; // start the scan
//Useful Variables
  int i=0;
  int j=0;
  int state;
  int vSpeed=200;     // Default speed, from 0 to 255
  int cmdIndex;
  char cmd[100];

  boolean cmdStartsWith(char *st) {
  for(int i=0; ; i++) {
    if(st[i]==0) return true;
    if(cmd[i]==0) return false;
    if(cmd[i]!=st[i]) return false;;
  }
  return false;
}

void exeCmd(){
  if (cmd[0] == '0'){
      vSpeed=0;}
```

FIGURE 4.6 : Sample coding of the RC car

This project demonstrates that the RC car can be controlled wirelessly by using wireless communication which is Bluetooth. A simple controller can be built by using the Bluetooth system that was more cheap, simple and the most important is it can prevent disturbance from the other frequency.

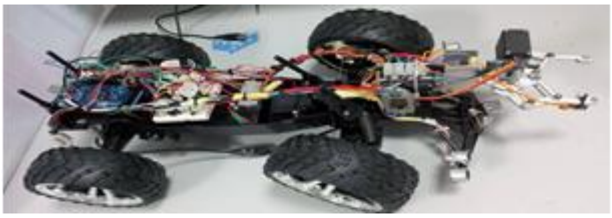


FIGURE 4.7 : RC car turning right

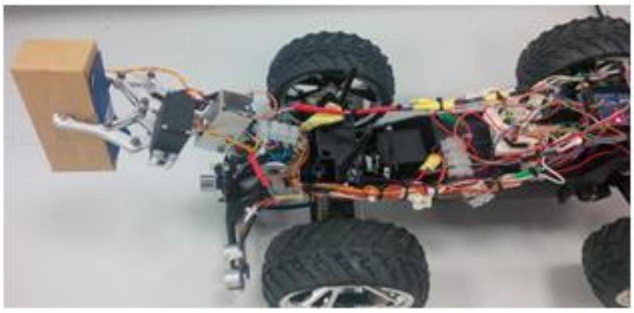


FIGURE 4.8: The gripper lift an object

For this project, this Bluetooth system has a good response time from the input which is an Android device to the output from the RC car. The RC car able to respond less than 1 second when the command is sent to the Bluetooth device. With this result, this system is suitable to be implemented in controlling the RC device like car, boat, or robot.

This system also has been tested with two different Android application software which is RoboRemo app and MIT inventor 2 apps. For RoboRemo application software, it has a good response time and can be controlled further away up to 50 meters from the controller which is Android device. This software also easy to use because the interface is user friendly. The user can manage how to control the RC car either to use touch button or slider. They just set the command that was set in Arduino coding for each button or slider to control the RC car.

While for the MIT inventor application software, it has good respond time same with RoboRemo but it can not control the RC car when the RC car is further away from the Android device. This software also difficult to be controlled when the system of the robot arm is added to the car movement system. The robot arm uses two servo motors, one for lift and drop the gripper and the other one is to grip and ungrip. While the car direction uses one servo motor to control the car to the left or right.

The Android app developed by RoboRemo software is easy to use because it is user friendly. This software is fully customizable. The user can build their own interfaces (add items, move them, resize them). The available item of this software is touch button, slider, led, level indicator, accelerometer sensor and text field. This item is enough to use as input and indicator for the Bluetooth system to control RC cars. In this project, just few items were used like touch button and slider. It shows the positive result when the RC car and the robot arm can respond smoothly to the input command.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 CONCLUSION

In conclusion, this project objectives are achieved and it is shown that Bluetooth system can be applied to control an RC car by using proper Android application and simple Android device. The system had worked properly according to its function and it's user friendly. Even though, there are multiple project has been using the same system, but different function, it still follows the objectives to control the RC car by using different kind of feature available on the Android device that can be used to control RC car and Android app and it can be properly developed by using free software which is RoboRemo software.

5.2 PROBLEM ENCOUNTERED

During testing, this MIT inventor app software can control just one servo motor, when the second servo is added the system is not running properly. There will interrupt occur that will cause the system failure and lost connection between Bluetooth module and Android device. It must reconnect again to control the RC car and the problem occurs again. When this system is changed to use RoboRemo software, this problem is solved. It can control multiple servo motor smoothly and no interrupt occurs with the connection.

Beside that, this RC car also tested by adding the system that can avoid obstacles. This system will use the ultrasonic sensor to detect certain range distance of the obstacle and when it reaches a minimum distance it will send command to avoid the obstacle. When this system is tested, this system is not running properly. It just overwrite the main system to control the movement of RC car and robot arm and the main system cannot be run. After troubleshooting this problem, this still remains unsolved, so this system is not used in this project.

5.3 RECOMMENDATION

Based on the system that was developed in this project, it can be used for another project with quite similar function. In addition, this RC car is already equipped with a gripper, Arduino controller, and Bluetooth module which can be implemented as a gripping mobile to pick up an object.

In this project, there are several improvements for future work that could be implemented to increase the performance and function of the system from my point of view which are:

1. The system could be integrated with other sensor like ultrasonic sensor, sonar sensor, or a laser range finder that can be used to avoid an obstacle. By adding this system the car can be safe to use because this system will detect the whole environment and provide accurate measurement of obstacle's distance.

2. This RC car will have better function if the system of image processing for detecting the environment and the object are developed. The user can control the RC car through the screen that will display the image processing from the camera on the RC car.

REFFERENCES

[1] http://en.wikipedia.org/wiki/Radio-controlled_car

[2] Steven Northrup and Christopher Paros,” Work in Progress - Capstone Experience – Visual Navigation of an RC Vehicle using Wireless Video Feedback to a PC” Department of Electrical Engineering, Western New England College, Springfield 2005.

[3] Chomtip Pompanomchai, Burin Wiranurak, Woraphol Weerawut and Linjong Kawrungruang, ” Operation of a Radio-Controlled Car Using Face Images” 2010 2nd International Conference on Mechanical and Electronics Engineering (ICMEE 2010)

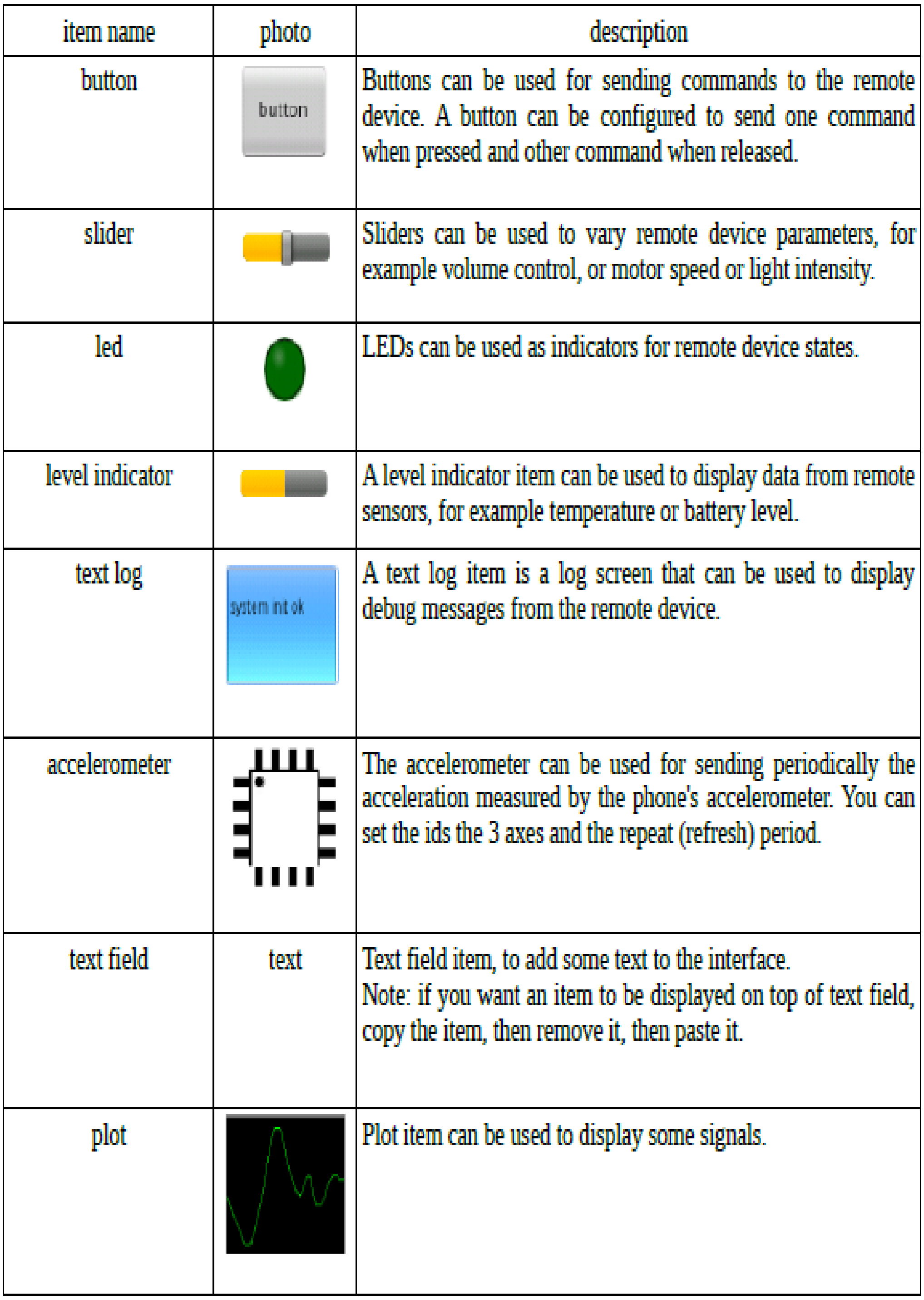
[4] Liu Huan, Wang Dahu, Zhang Tong, Huang Kerning; “RESEARCH ON A REMOTE CONTROL VEHICLE”, 2010 2nd international Conference on Education Technology and Computer (ICETC).

[5] Heikki Rissanen, Jukka Mahonen, Keijo Haataja,Markus Johansson, JamoMielikainen, Pekka ToivanenDepartment of Computer Science University of Kuopio, (2009 IEEE)

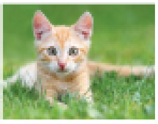
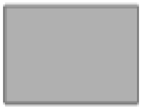
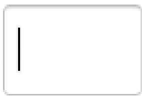
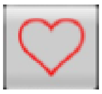
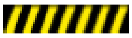
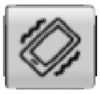
[6] A. González-Mendoza, J. L. Pérez-Benítez, J. A. Pérez-Benítez, J.H. Espina-Hernández. M. Sc. in Electronics, Esime Zacatenco IPN México City, México (2015IEEE)

APPENDIX A

Available interface item

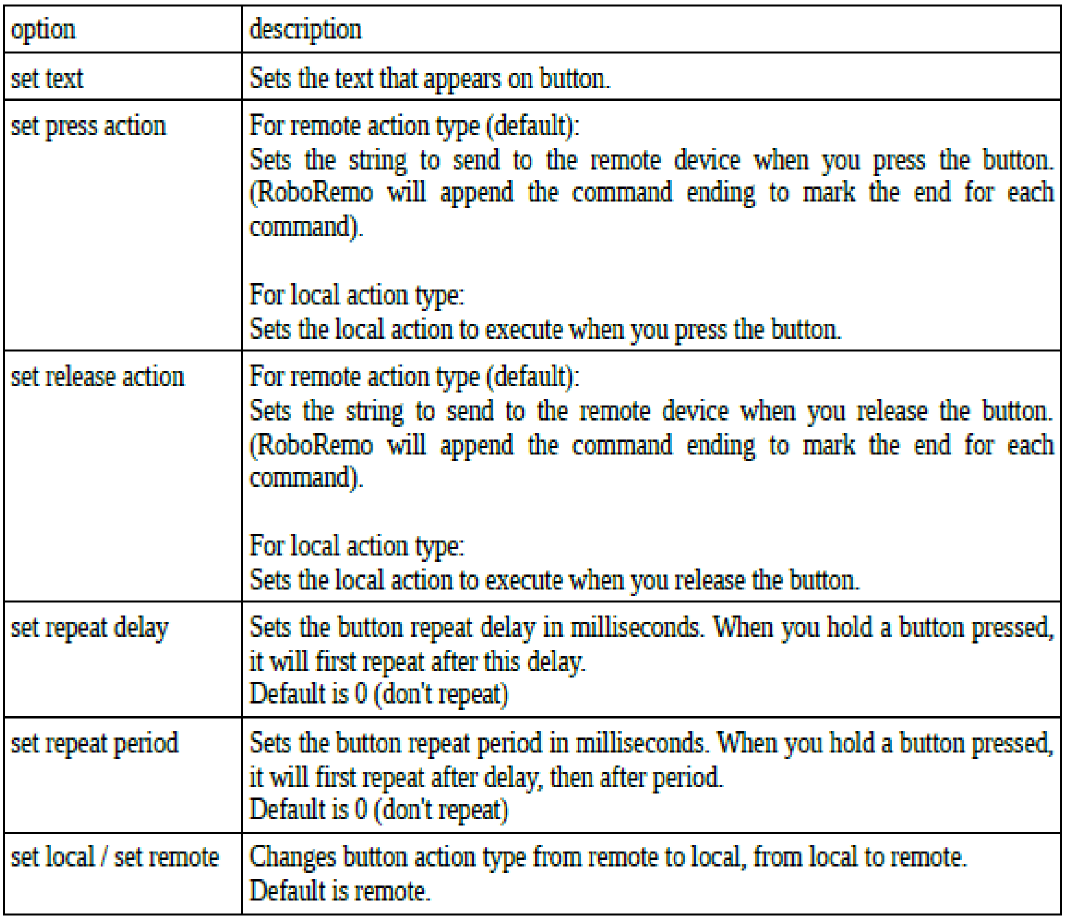


item name	photo	description
button	button	Buttons can be used for sending commands to the remote device. A button can be configured to send one command when pressed and other command when released.
slider		Sliders can be used to vary remote device parameters, for example volume control, or motor speed or light intensity.
led		LEDs can be used as indicators for remote device states.
level indicator		A level indicator item can be used to display data from remote sensors, for example temperature or battery level.
text log	system init ok	A text log item is a log screen that can be used to display debug messages from the remote device.
accelerometer		The accelerometer can be used for sending periodically the acceleration measured by the phone's accelerometer. You can set the ids the 3 axes and the repeat (refresh) period.
text field	text	Text field item, to add some text to the interface. Note: if you want an item to be displayed on top of text field, copy the item, then remove it, then paste it.
plot		Plot item can be used to display some signals.

image*		Image item can be used to display an image that is sent from the microcontroller. *available only in RoboRemo Full version
touchpad		Touchpad item can be used to send position and motion of your finger.
kbd connector		Kbd connector can be used to send commands using the keyboard.
heartbeat sender		heartbeat sender can be used to periodically send some string (like a heartbeat signal). The remote device can watch for this signal to make sure it is still connected.
touch stopper		If you put a reset button for example, surround it with touch stoppers, so it doesn't get pressed when you touch the screen near to it. When you exit ui editor, touch stoppers become invisible, so they don't annoy you.
vibrator*		Vibrator item gives the remote device access to the phone's vibrator (you can send a command from microcontroller to make your phone vibrate). *available only in RoboRemo Full version.

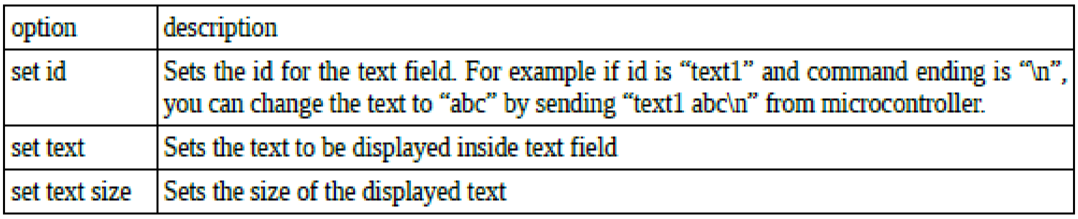
APPENDIX B
EDIT OPTIONS

1) BUTTON EDIT OPTIONS



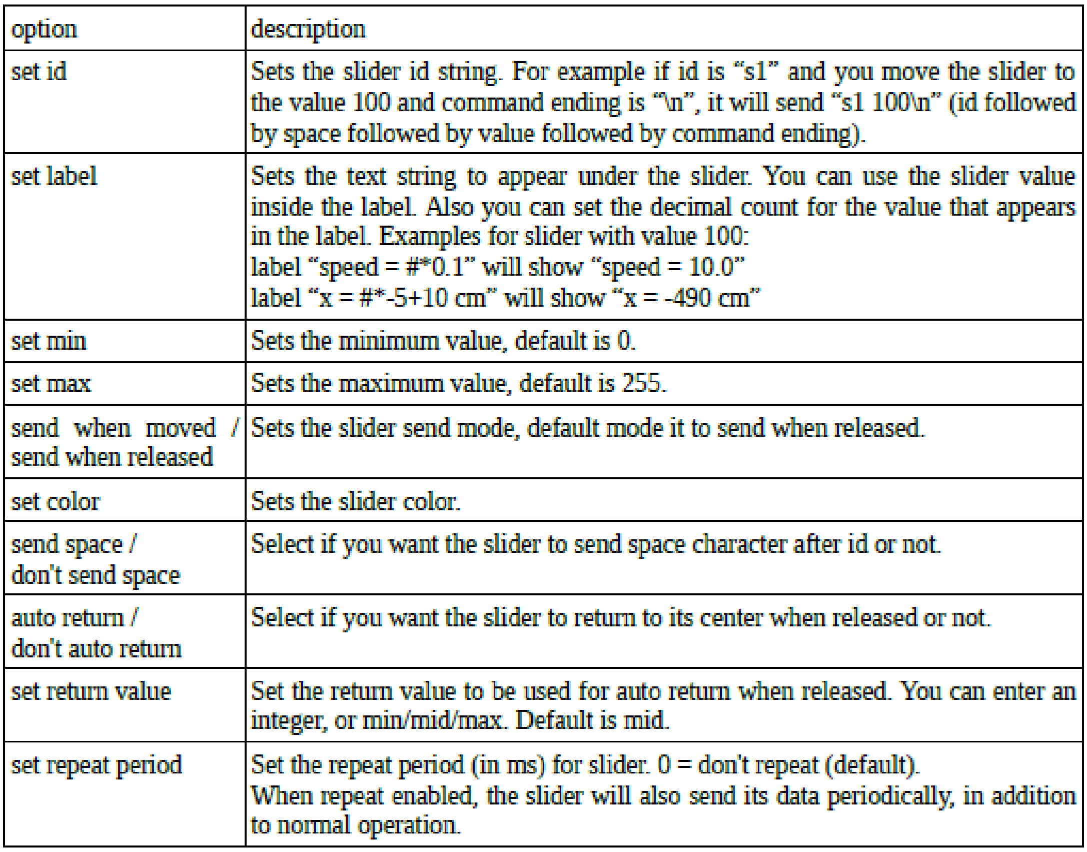
option	description
set text	Sets the text that appears on button.
set press action	For remote action type (default): Sets the string to send to the remote device when you press the button. (RoboRemo will append the command ending to mark the end for each command). For local action type: Sets the local action to execute when you press the button.
set release action	For remote action type (default): Sets the string to send to the remote device when you release the button. (RoboRemo will append the command ending to mark the end for each command). For local action type: Sets the local action to execute when you release the button.
set repeat delay	Sets the button repeat delay in milliseconds. When you hold a button pressed, it will first repeat after this delay. Default is 0 (don't repeat)
set repeat period	Sets the button repeat period in milliseconds. When you hold a button pressed, it will first repeat after delay, then after period. Default is 0 (don't repeat)
set local / set remote	Changes button action type from remote to local, from local to remote. Default is remote.

TEXT FIELD EDIT OPTIONS



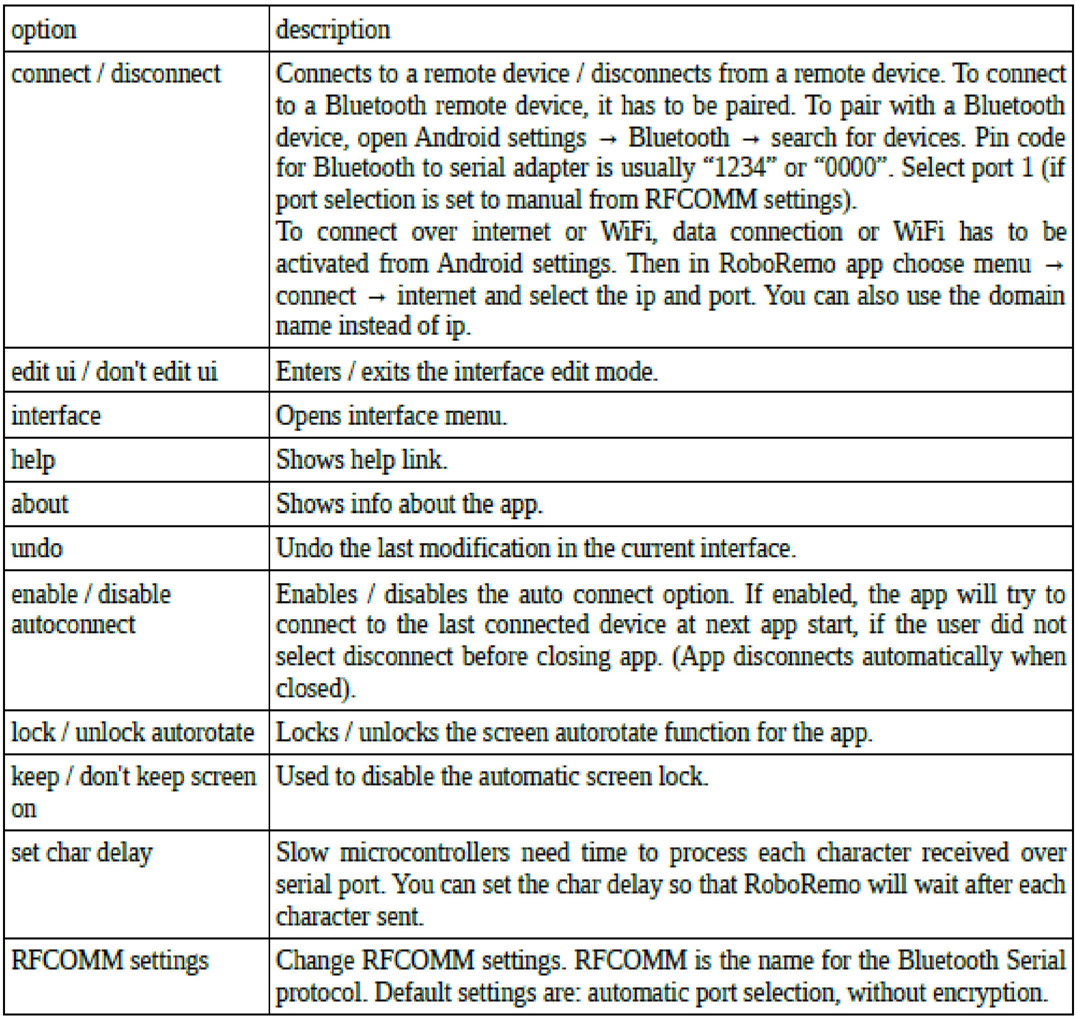
option	description
set id	Sets the id for the text field. For example if id is “text1” and command ending is “\n”, you can change the text to “abc” by sending “text1 abc\n” from microcontroller.
set text	Sets the text to be displayed inside text field
set text size	Sets the size of the displayed text

SLIDER EDIT OPTIONS



option	description
set id	Sets the slider id string. For example if id is “s1” and you move the slider to the value 100 and command ending is “\n”, it will send “s1 100\n” (id followed by space followed by value followed by command ending).
set label	Sets the text string to appear under the slider. You can use the slider value inside the label. Also you can set the decimal count for the value that appears in the label. Examples for slider with value 100: label “speed = #*0.1” will show “speed = 10.0” label “x = #*-5+10 cm” will show “x = -490 cm”
set min	Sets the minimum value, default is 0.
set max	Sets the maximum value, default is 255.
send when moved / send when released	Sets the slider send mode, default mode it to send when released.
set color	Sets the slider color.
send space / don't send space	Select if you want the slider to send space character after id or not.
auto return / don't auto return	Select if you want the slider to return to its center when released or not.
set return value	Set the return value to be used for auto return when released. You can enter an integer, or min/mid/max. Default is mid.
set repeat period	Set the repeat period (in ms) for slider. 0 = don't repeat (default). When repeat enabled, the slider will also send its data periodically, in addition to normal operation.

MENU OPTIONS



option	description
connect / disconnect	Connects to a remote device / disconnects from a remote device. To connect to a Bluetooth remote device, it has to be paired. To pair with a Bluetooth device, open Android settings → Bluetooth → search for devices. Pin code for Bluetooth to serial adapter is usually "1234" or "0000". Select port 1 (if port selection is set to manual from RFCOMM settings). To connect over internet or WiFi, data connection or WiFi has to be activated from Android settings. Then in RoboRemo app choose menu → connect → internet and select the ip and port. You can also use the domain name instead of ip.
edit ui / don't edit ui	Enters / exits the interface edit mode.
interface	Opens interface menu.
help	Shows help link.
about	Shows info about the app.
undo	Undo the last modification in the current interface.
enable / disable autoconnect	Enables / disables the auto connect option. If enabled, the app will try to connect to the last connected device at next app start, if the user did not select disconnect before closing app. (App disconnects automatically when closed).
lock / unlock autorotate	Locks / unlocks the screen autorotate function for the app.
keep / don't keep screen on	Used to disable the automatic screen lock.
set char delay	Slow microcontrollers need time to process each character received over serial port. You can set the char delay so that RoboRemo will wait after each character sent.
RFCOMM settings	Change RFCOMM settings. RFCOMM is the name for the Bluetooth Serial protocol. Default settings are: automatic port selection, without encryption.

APPENDIX C

SAMPLE CODING

```
#define bluetooth Serial
#include <Servo.h>
#define trigPin 12 // define the pins of your sensor
#define echoPin 13

Servo myServo1, myServo2, myServo3;
int SW1 = A0;
int LMotor1 = 3;
int LMotor2 = 5;
int ang =0;
long duration, distance; // start the scan
//Useful Variables
  int i=0;
  int j=0;
  int state;
  int vSpeed=200;     // Default speed, from 0 to 255
  int cmdIndex;
  char cmd[100];

  boolean cmdStartsWith(char *st) {
  for(int i=0; ; i++) {
    if(st[i]==0) return true;
    if(cmd[i]==0) return false;
    if(cmd[i]!=st[i]) return false;;
  }
  return false;
}

void exeCmd(){
  if (cmd[0] == '0'){
      vSpeed=0;}
    else if (cmd[0] == '5'){
      vSpeed=255;}
    else if (cmd[0] == '1'){
      vSpeed=100;}
    else if (cmd[0] == '2'){
      vSpeed=150;}
    else if (cmd[0] == '3'){
      vSpeed=200;}
    else if (cmd[0] == '4'){
      vSpeed=255;}

    if( cmdStartsWith("servo2 ") ) {
    int val = atoi(cmd+7);
    myServo2.writeMicroseconds(val);

  }

 if( cmdStartsWith("servo3 ") ) { // example: if cmd is "servo1 1500"
    int val = atoi(cmd+7);  // val will be 1500
                            // cmd+7, because value comes after "servo1 " which is 7
characters
    myServo3.writeMicroseconds(val);
  }

 if (cmd [0] == 'f')  //forward
  {
    analogWrite(LMotor1,0);
    analogWrite(LMotor2,vSpeed);
  }

   if (cmd [0] == 's') //motor stop
  {
   analogWrite(LMotor1,0);
   analogWrite(LMotor2,0);
  }

  if (cmd [0] == 'b')// reverse
  {
    analogWrite(LMotor1,vSpeed);
    digitalWrite(LMotor2,0);
  }

  if (cmd [0] == 'r') //right
  {
        myServo1.writeMicroseconds(2000);
  }

 if (cmd [0] == 'c') //center
 {
        myServo1.writeMicroseconds(1500);
 }

  if (cmd [0] == 'l') //left
  {

        myServo1.writeMicroseconds(1000);}
```

```
}
void setup() {
  // Set pins as outputs:
 delay(500); // wait for bluetooth module to start
 bluetooth.begin(9600); // Bluetooth default baud is 115200
 myServo1.attach(6, 1000, 2000);
 myServo2.attach(7, 1000, 2000);
 myServo3.attach(8, 1000, 2000);
 int ang =0;
 pinMode(SW1, INPUT);
  // Initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

void loop() {
  if(bluetooth.available()) {

  char c = (char)bluetooth.read();
  if(c=='\n') {
   cmd[cmdIndex] = 0;
   exeCmd();  // execute the command
   cmdIndex = 0; // reset the cmdIndex
  } else {
   cmd[cmdIndex] = c;
   if(cmdIndex<99) cmdIndex++;
  }
 }
}
```