THE DEVELOPMENT A OF LOW-COST ROBOTIC SQUIRREL FOR THE STUDY OF SMALL ANIMAL BEHAVIOR

By

LAKSHMI DIVIJA SARUPYA BRAHMANDAM

Bachelor of Engineering in Electrical and Electronics Engineering

Osmania University

Hyderabad, Telangana, India.

2017

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
December, 2019

THE DEVELOPMENT A OF LOW-COST ROBOTIC

SQUIRREL FOR THE STUDY OF SMALL ANIMAL

BEHAVIOR

Thesis Approved:

Dr. Carl D Latino, Ph. D.

Committee Chair and Thesis Adviser

Dr. Rama Ramakumar, Ph. D.

Committee Member

Dr. Ekin Sabit, Ph. D.

Committee Member

ACKNOWLEDGEMENTS

I would like to thank Dr. Charles Abramson of the Psychology Department, who has had a long standing collaboration with Dr. Carl Latino, for creating the project idea that led to this thesis His sincere concern and suggestions made this work possible.

I would like to sincerely thank my advisor, Dr. Carl Latino for believing in me and being a remarkable support by helping me through this educational endeavor, without whom I would not have accomplished my goal.

I would like to thank Dr. Ekin Sabit and Dr. Rama Ramakumar for being my committee members, for their time and concern.

I would like to thank my father, Mr. Sai Prakash Brahmandam for his unrelenting support in my education and all points of my life. I would like to thank my brother Surendra Bhargava for sharing his technical insights and helping me with various aspects of my thesis. I would also like to thank my family and friends who believed in me and extended their support at all points in my life.

Lastly, I would like to thank Fernando Cavasos, Vice President, OSU Mercury Robotics, for his helpful suggestions and technical support.

Acknowledgements reflect the views of the author and are not endorsed by committee members or Oklahoma State University.

Name: LAKSHMI DIVIJA SARUPYA BRAHMANDAM

Date of Degree: DECEMBER, 2019

Title of Study: THE DEVELOPMENT A OF LOW-COST ROBOTIC SQUIRREL FOR THE STUDY OF SMALL ANIMAL BEHAVIOR

Major Field: ELECTRICAL AND COMPUTER ENGINEERING

Abstract:

This project focuses on building a low-cost remote controlled Robot Squirrel to support the study of small animal behavior. It can also be implemented as an educational tool for all the stages of the educational systems, from elementary to university level. It is a ground-based motion robot, which is driven at variable speeds utilizing slip steering (a method similar to the motion of a tank) which can be moved in any desired direction. It supports an on-board video system to record, and monitor various behavioral patterns of animals, primarily squirrels in this project even from a distance. It also comprises of an audio system which can record and playback sounds to the animals and a simple robot arm-like structure with two degree of freedom controlled by servos, which acts as a nut delivery system to the animals. An Android smart phone application was developed to control the motion and speed of the robot and other operational controls in the system, like record and playback control and also the movement of the robot arm. Arduino Uno, an ATmega328 based microcontroller board acts as the major control unit of the entire system. A Bluetooth module acts as an interface between the Arduino and the smart phone supporting the control and operation of the Robotic Squirrel.

TABLE OF CONTENTS

Chapter Page

I. INTRODUCTION 1

II. REVIEW OF LITERATURE 4

III. METHODOLOGY 9

3.1 INTRODUCTION 9

3.2 ARDUINO UNO 12

3.2.1 Power System 13

3.2.2 Memory Map 14

3.2.3 Pin Description 15

3.3 L298 MOTOR DRIVER SHIELD 17

3.3.1 Bidirectional Control of DC Motor using H-Bridge 19

3.3.2 Pulse Width Modulation Control (PWM) 20

3.4 HC-05 BLUETOOTH MODULE 22

3.4.1 Specifications 23

3.5 ISD1820 VOICE RECORDER MODULE 25

3.5.1 Specifications 26

3.5.2 Schematic 27

3.5.3 Pin Description 28

3.6 SERVO MOTORS 30

3.7 ArduCAM ESP8266 Wi-Fi BOARD AND 2MP MINI CAMERA 33

3.8 SYSTEM SETUP AND OPERATION 36

3.8.1 Power System 36

3.8.2 Schematic diagram and description 38

3.8.2.1 HC-05 Bluetooth and Arduino interface 39

3.8.2.2 ArduCAM ESP8266 39

3.8.3 Communication System 40

3.8.4 Development of an Android App to control the Squirrel Robot using MIT App Inventor 44

3.8.5 Arduino Code 47

Chapter Page

IV. CONCLUSION 51

V. FUTURE WORKS 52

REFERENCES 53

APPENDICES 55

LIST OF TABLES

Table Page

3.2 Overview of Arduino’s features 14
3.3.1 Direction of Motors A and B 20
3.5 ISD1820 Voice Recorder specifications 27
3.7 ArduCAM ESP8266 Pin Description 34
3.8.3a Division of 1-byte memory for control: motion and voice recorder 41
3.8.3b Division of 1-byte memory for control: servo and speed 42

LIST OF FIGURES

Figure Page

3.1a Remote-controlled Squirrel Robot sytem 9
3.1 b Block diagram of remote control Squirrel Robot 11
3.2 a Arduino Uno Overview 12
3.2b Arduino Memory Map 14
3.2 c Arduino Pin description 15
3.3 L298 Arduino Motor Driver Shield 18
3.3.1 L298 Dual Full H - Bridge 19
3.3.2 PWM Duty Cycle 21
3.4 HC – 05 Bluetooth Module 22
3.5 a ISD1820 Voice Recorder with 0.5W 8 Ohm Speaker 25
3.5b ISD1820 Schematic diagram 27
3.5 c ISD1820 Voice Recorder indicating the components 28
3.6 HiTec HS -322HD standard and 9g Micro servo motors 30
3.7 a ArduCAM ESP8266 Uno board and 2MP Mini Camera 33
3.7b 2MP Mini Camera mounted on ArduCAM ESP8266 board 34
3.8.1 Power distribution diagram of the Squirrel Robot 36
3.8.2 Schematic diagram of the Squirrel Robot system 38
3.8.4 Android App controls 44

LIST OF SYMBOLS

GSM	Global System for Mobile
GPRS	General Packet Radio Service
LED	Light Emitting Diode
RF	Radio Frequency
AVR	Advanced Virtual Chip
IDE	Integrated Development Environment
USB	Universal Serial Bus
SRAM	Static Random Access Memory
EEPROM	Electrically Erasable Programmable read Only Memory
PWM	Pulse Width Modulation
ROSC	Resistor Controlled Oscillator Input
EOM	End of Message
AGC	Automatic Gain Control
SCL	Serial Clock
SDA	Serial Data
CSN	Chip Select Not
MISO	Master In Slave Out
MOSI	Master Out Slave In
SCLK	Serial Clock

CHAPTER I

INTRODUCTION

Much research involves the study of animal behavior implementing unique and innovative technologies, many of which, employing robotic animals is increasing in today's world. Replicated models of animals or robots with similar animal behavior and responses are designed for research purposes. Animal models, for instance, the Sony Aibo robotic dog, built for establishing a real and apparent companionship with human beings, costs approximately $3000.00. There are other similar models, which also require expensive technology. The purpose of this study was to develop a low-cost robotic animal that can be used at all stages of educational system, from elementary through university levels.

The study of animal behavior is a keystone of experimental psychology. It seeks to shed light on the way animals interact with their environment and each other. The study of animal behavior supports the further understanding of human behavior, which is known as Comparative Psychology [18,19]. When it comes to dealing with the preservation of the species or how the behavior of animals aids them in their survival, animal behavioral research has been specifically pertinent.

Wireless communication technologies are being widely appreciated by consumers world-wide. There is a wide range of wireless technologies, such as: Bluetooth, GSM, GPRS, Wi-Fi, etc. Bluetooth technology integrated with smart phones has demonstrated various advantages and has completely changed the digital era from wired devices to wireless devices. A Bluetooth device is capable of connecting to several different Bluetooth modules through one link [2]. It provides connectivity to mobile phones, computers or any portable device with Bluetooth capability. It has the ability to deliver both data and voice transmissions [3].

This project is a collaborative effort between the Psychology Department and the Department of Electrical and Computer Engineering (ECE), at Oklahoma State University. We detail the use of a remote-controlled Arduino Robot, which was designed and developed to study the behavioral psychology of animals in the wild. The robot is operated and controlled remotely, using an application developed on an Android phone. The app has a means of controlling the robot to be moved in all the desired directions. It also has control to move the robot at different speeds as necessary, i.e., to approach the animals slowly and carefully without causing alarm or to move at higher speeds to keep pace with the subjects. An on-board video system is included to support live video streaming and observation of the animals' movements even from a distance so as to not alert them. It has an audio system which has the capability of recording and playing a sound back to the animal. There is a servo-controlled mechanism to deliver a nut or an acorn in the direction of the animals. This project is primarily based on off the shelf low-cost components, which cost less than $200.00 in parts.

Dr. Charles Abramson, of the Psychology Department, gained inspiration for this project from an international and interscholastic robotics competition created at Oklahoma State University called the Mercury Remote Robot Challenge. In this annual competition, the competitors are challenged to design and develop a remote-controlled robot capable of performing a mission, while guided from anywhere in the world. However, for this project, a worldwide communication system was not necessary and a short-range Bluetooth technology was employed instead. An attempt was made to package and enclose all the components in a model that loosely resembles the structure of a squirrel. This methodology could be applied for other animals as well. The resemblance was not a primary consideration, however, it is considered helpful to the user in increasing the chances of establishing an interaction between the real squirrel and the replica.

CHAPTER II

REVIEW OF LITERATURE

Implementation of robotic technologies is increasing in the field of Comparative Psychology. Initially, basic mock-up structures were used for the study purposes but there are many behavioral complexities that need to be addressed. Therefore, with the development of technology, rather advanced animal models were designed which are suitable to be able to blend in easily with the habitat of the targeting species to understand their lifestyle [13].

There have been many areas of research which demonstrate that animal behavior could be studied and influenced by the help of robotic technologies. The Collective and Individual Behavior of animals is the main question that would be addressed by the robotic models. There have been many research lines that focused on collective animal behavior, where species like cockroaches, chickens, etc., were studied to understand how they behave as a group. Once the robots are accepted into the habitat, two kinds of studies were performed. The first is the collective response of a group of animals to the influence created by the robots in their habitat. The second is the individual response towards the robots. It is believed that animals behave differently when in a group and when they are on their own. There are many responses that are determined by many factors such as: sound communication, color perception and pattern recognition.

Physical resemblance of robots may be another major factor in influencing the acceptance of the robot systems into the animal groups. But, there have also been studies contradicting this theory. According to a study that was conducted to control the social aggregation in cockroaches by the European project Leurre, robots called the InsBots were designed to resemble cockroaches only in their size and not in their appearance. Even with this, they were still accepted by the animal groups because of a special pheromone spread on the robots which is similar to the pheromones secreted by cockroaches. [14].

Similarly, in another study towards mixed societies of chickens and robots, a locomotive track-type mobile robot that does not resemble a chicken except in size was still accepted by the chicks. It is believed that from a natural learning process called filial imprinting, when the vocalizing robot is presented to the chicks shortly after hatching, they learn to believe that the robot is their mother [14].

However, there have been other non-conflicting theories that show that resemblance plays a key role in easier acceptance of the robots in the animal groups. According to the paper on introducing a 'Robofish' to investigate the collective behavior of fish, a robot was designed maintaining the morphological details of a three-spine stickleback. This Robofish is placed in a tank with the other stickleback fish and it is made to move around along with the other fish. Resemblance helps in the easy acceptance of the robots in the animal groups by not causing an alarm of introducing a foreign object into their living system [15].

Every animal possesses a different behavioral response even within the same species and the same environment. Robotic structures contribute immensely to the study of animal behavior which would otherwise require considerable animal training which can be a challenging task. These robots could be made to look, smell and even sound like animals and can be allocated to certain tasks similar to those performed by the animals to obtain a steady response from them [1]. These robotic models can be made simple or more advanced. However, they must be made with easy operation and implementation in mind. There are many design platforms, such as using a simple microcontroller, microprocessor based system, or by using advanced robotics which includes lengthy programming and coding. Many devices like Raspberry Pi, Arduino, Propeller, etc., can be used for robotic control. The Arduino is a relatively simple microcontroller system that is cost effective and easy to implement. Yet it has power to do a lot more than is needed. It is an ATmega328 microcontroller based board, which consists of hardware and flexible coding options, as it can be programmed in C, C++ and Java languages. The Arduino is readily available as it is an open source platform.

The functions of the robot should allow it to move and communicate with the targeted animals remotely with a human operator at a safe distance so as not to alarm the animals. This implementation is possible and is made easier by using an Arduino-based robot. There have been many lines of research on the Arduino-based robot system. There are many wireless remote controlled technologies that are available these days. The technologies most commonly used with the Arduino are Bluetooth, Radio Frequency (RF), etc., which support a reliable communication system. Many experiments and projects are conducted using RF and Bluetooth technologies with the Arduino. In RF, an electromagnetic field is generated when an alternating current is received by an antenna, which is used for wireless broadcasting, whereas a Bluetooth system uses much weaker radio signals. It is weaker by 1000 times, which once connected works without any interference from external Bluetooth devices. It not only uses less power, but it is also more resistant to interference. Bluetooth is more reliable and can perform multiple operations through one single device like from a smart phone. This Bluetooth technology could be implemented along with the Arduino robot system to achieve the required automation and the desired control system. A Bluetooth device that is compatible with the Arduino board is the HC-05 Bluetooth module. A serial communication is established between the two where this module serves as an interface between the Arduino and the smart phone.

The objective of building the robotic structures make the appearance and function as close to the real animal as possible. The different types of communication systems employed by the animals include verbal, non-verbal, visual communication etc. Verbal communication can be achieved by making the robot speak or make sounds similar to the animal. But introducing an automatic talking mechanism into the robot makes the design and implementation much more complicated. Instead, an audio recording and playback system would be less complex. This could lead to trust and acceptance of the robot into its environment, serving the purpose of establishing a communication. Also, it is necessary to monitor the movements and reactions of the animals in response to the influence of the robot. In order to achieve this, a video system could be introduced with a camera which would capture and save the movements of the targeted animal in response to the robot, which helps in a detailed observation and study.

Based on these researches, it seemed logical to design and construct a low-cost research tool for studying small animals in the wild.

THE SQUIRREL ROBOT SYSTEM:

The present project is a remote controlled ground motion-based robot system, designed to allow the study of squirrel behavior of in its environment. It has an audio system to record and playback the sounds to the animals, a video system which supports a live streaming video and saves it for future references. It has a two degree of freedom robot arm, controlled by servos to act as a nut delivery system.

The three major components that are required for this project are the robot, the smart phone and a laptop for visual feedback from the robot.

CHAPTER III

METHODOLOGY

3.1 INTRODUCTION

The Remote Controlled Arduino Robot is expected to provide assistance in studying animals in the wild by performing certain specific operations. Every animal study has different requirements based on the Ecosystem. This study focuses mainly on squirrels usually found in our neighborhood. The system requirements include operations like Video recording, Audio recording and playback, locomotion and a system to deliver an acorn or a nut in the direction of the animal. The robot moves a speed of 1.3 ft/sec.

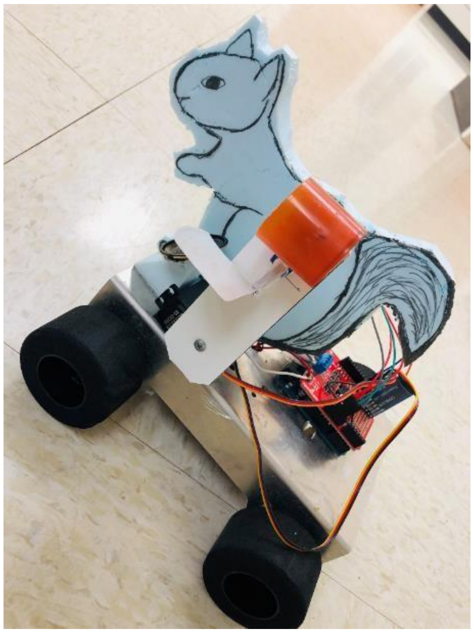


Fig:3.1a Remote-controlled Squirrel Robot system

This project fundamentally works on an Arduino Uno, an ATmega328P microcontroller based board, with a L298 Motor Shield mounted on top of the Arduino. This Motor Shield is used to run the four 12V DC motors which provide motion to the Robot. The entire system control is carried out by a Smart phone application, which sends packets of information via Bluetooth to the HC – 05 Bluetooth module serially connected to the Arduino where it is decoded to perform the necessary and required operations.

The hardware involved is as follows

1. Arduino Uno
2. Arduino L298 Motor Driver Shield
3. HC-05 Bluetooth module
4. Servo Motors
5. ArduCAM ESP8266 Uno Wi-Fi board
6. 2MP ArduCAM Camera Module
7. ISD 1820 Voice Recorder module
8. Four 12V DC Motors (Operating at 9.6V)
9. 1.2K, 0.6K Ohms resistors
10. 9.6V battery
11. Jumper cables

The block diagram of the entire system is given below which provides an overview of the project as a whole,

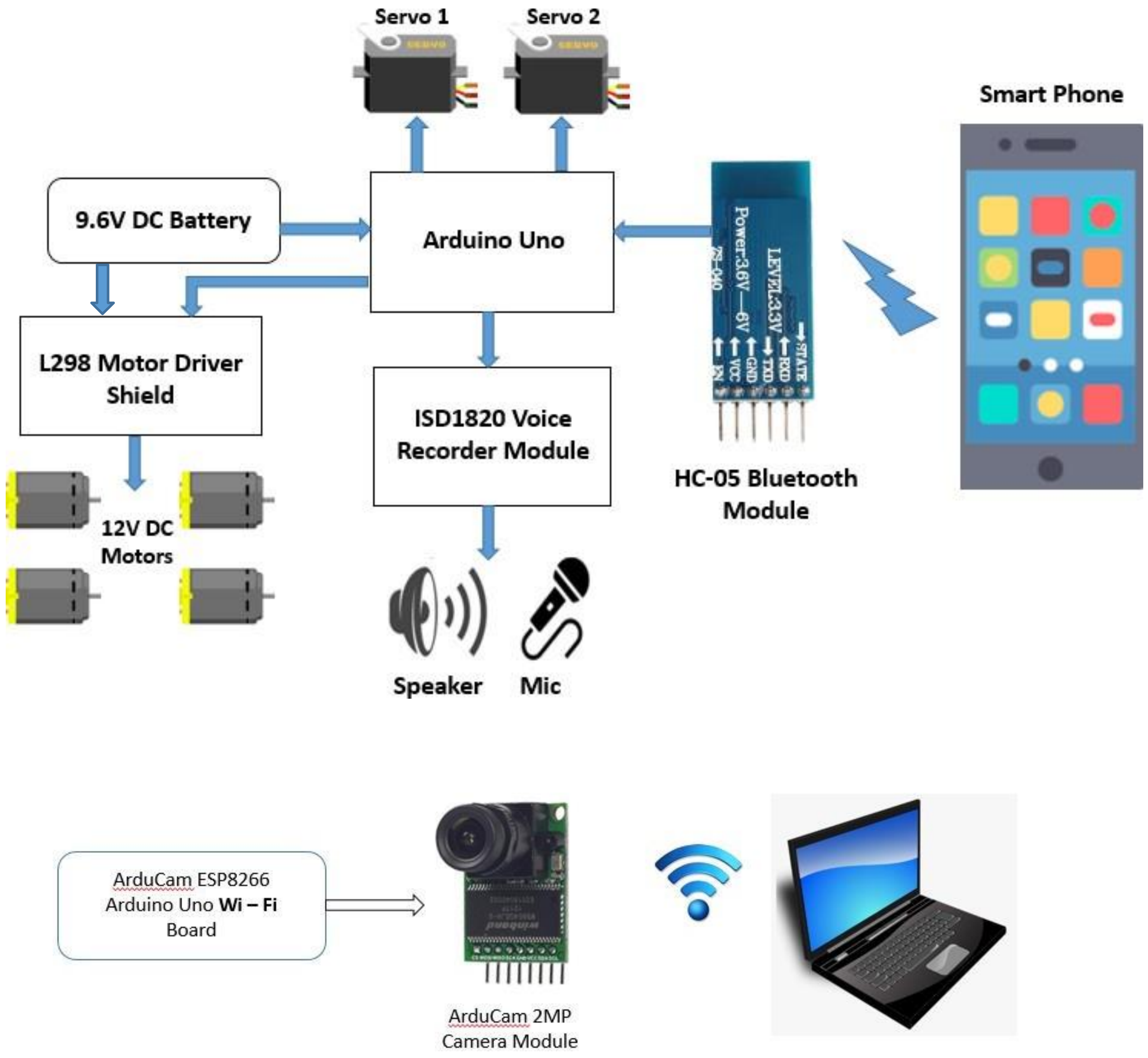


Fig: 3.1b – Block diagram of the Bluetooth Controlled System

3.2 ARDUINO UNO

Arduino Uno is a microcontroller platform, which is based on Advanced Virtual Chip(AVR) microcontroller ATmega328. To provide an affordable and adaptable means for professionals and students to control several devices in real-time, Interaction Design Institute Ivrea in 2003 designed the first ever project based on Arduino [4]. The software used for Arduino devices is the Arduino IDE (Integrated Development Environment) which is programmed using C, C++ languages. The Arduino Uno board could be directly connected to the computer using a USB port available on the board. Using this USB connection, any desired program can be uploaded to the Arduino board from the IDE software. IDE is compatible with operating systems like Windows, MAC and Linux [4]. An overview of all the features of an Arduino board is shown in Fig: 3.2a.

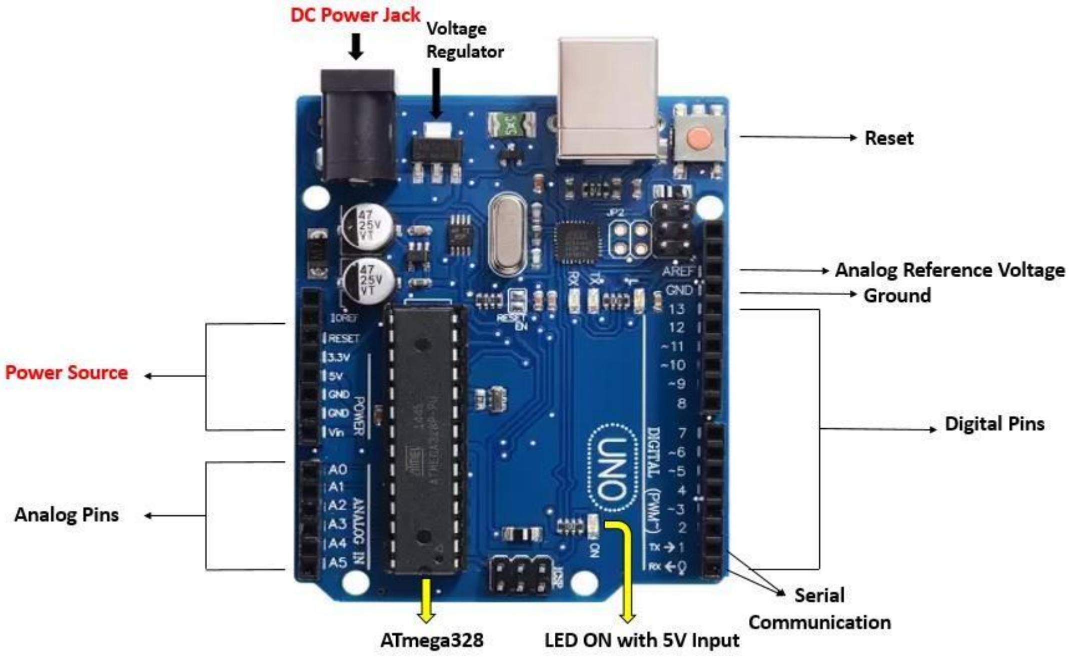


Fig: 3.2a – Arduino Uno Overview [37]

3.2.1 Power System

The Arduino can be powered from the computer, by connecting it to a USB port or by an external power supply such as an AC to DC adapter or a battery. It has an on board voltage regulation providing 3.3V and 5V [5].

3.2.2 Memory Map

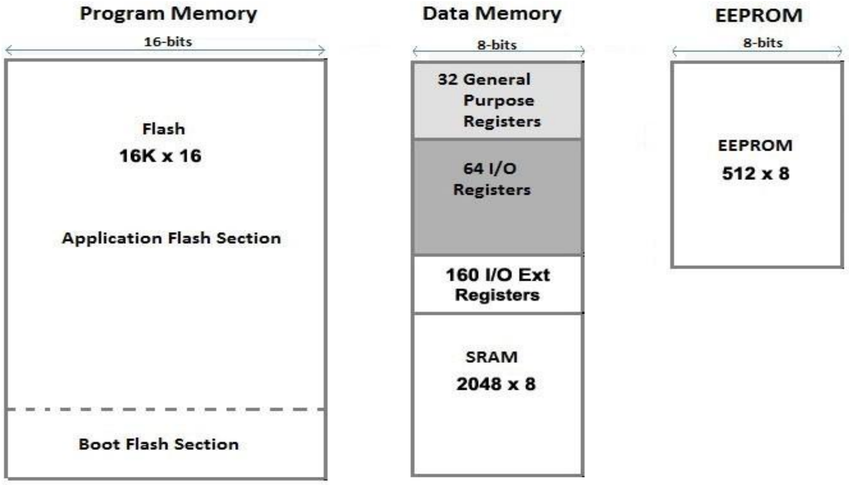


Fig: 3.2b – Arduino Uno Memory Map [36]

ATmega328 has 32KB (including the 0.5KB used for the bootloader). It also consists of a 2KB SRAM and 1KB EEPROM. Out of the 32KB Flash memory, 13KB is used to store the instructions in the form of code [4,5].

Table: 3.2 Overview of Arduino's features: [5]

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage(Limits)	7-12V
Input Voltage(Recommended)	6-20V
Analog Pins	6
Digital I/O Pins	14(of which 6 provide PWM output)
DC current for 3.3V Pin	50mA
DC current per I/O Pin	40mA

Flash Memory	32KB(ATmega328)
SRAM	2KB(ATmega328)
EEPROM	1KB(ATmega328)
Clock Speed	16MHz

3.2.3 Pin Description:

Arduino Uno has 14 I/O Digital pins and 6 Analog pins, which support external circuit board connections. A schematic pin diagram of the Arduino Uno is shown in Fig: 3.2b and explained below. Each of the 14 Digital pins could be used for input or output, with the help of digitalRead( ), digitalWrite( ) and pinMode( ) functions, while writing the Arduino program from the IDE software.

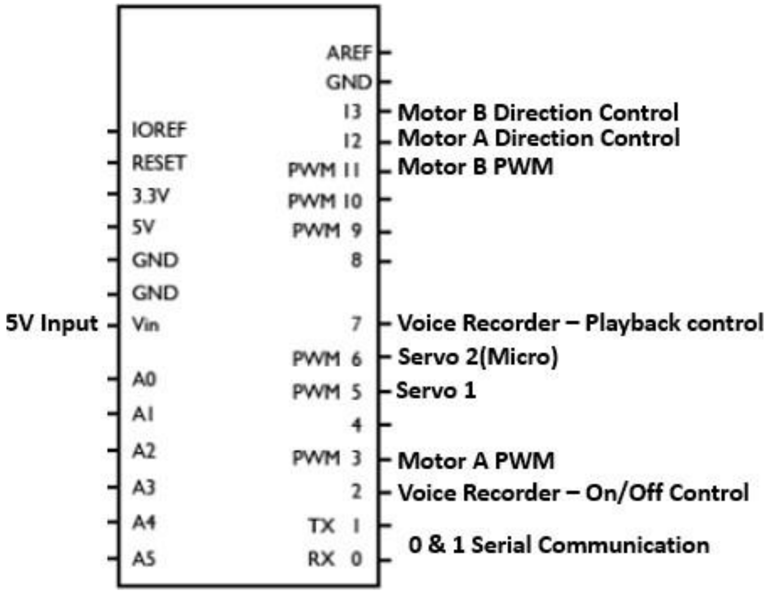


Fig: 3.2c - Arduino Uno Schematic Pin diagram [25]

These pins operate at 5 volts. The 6 Analog pins operate at voltages ranging from 0-5V, whereas to power the Arduino Uno, it requires only 5 volts, which can be achieved by connecting the board to a computer using the USB cable or by an external power source up to 12V which could be regulated in the Arduino board itself.

The pins that are

LED: Pin 13 is an inbuilt LED pin in the Arduino board. The pins on the board are turned ON and OFF by giving HIGH and LOW functions from the IDE code respectively, which means when the pin is HIGH, LED is ON and when the pin is LOW LED is OFF.

Vin: In order to supply voltage to the board through the power jack, this Vin pin can be used, which is different from the 5V regulated voltage pin. Voltages in the range of 7 to 20V are supported by this pin.

5V: The Arduino comes with an inbuilt voltage regulator which supplies about 5V of regulated output voltage.

If voltages supplied to the Arduino through 5V or 3.3V pins, exceed these limits, it could result in damaging the board.

GND: There are three ground pins are provided in the Arduino board.

Reset: This button on the board resets the program that is currently running on the Arduino.

PWM: Pins 3,5,6,9,10 and 11 provide Pulse Width Modulation (PWM). These pins are configured to provide 8-bit PWM output which could be obtained by using the analogWrite( ) function.

Serial Communication: Pins 0, 1 are used for serial communication. These pins are used to receive and transmit serial data respectively.

External Interrupts: By providing a changing value or LOW value, an interrupt is provided. This purpose is fulfilled by using pins 2 & 3.

The reason the Arduino was chosen is because,

1. It is an Open-source platform which means it is readily available for everybody to use.
2. It uses an easily understandable language.
3. External circuits like Shields for various purposes of the Arduino are available. For example, a Motor Shield, a Wi-Fi Shield etc. are available [6].

3.3 L298 MOTOR DRIVER SHIELD

The Motor Shield that is used in the proposed system is the Ardumoto Shield, which is a dual motor controller compatible with the Arduino board. The Motor Shield serves the purpose driving the motors. This device is capable of receiving two numbers each ranging from zero to 255, converting those into 0 to 100% Pule Width Modulated signals that drive the motors.

The Shield works on L298 Motor Driver placed at the center of the Motor Shield board, which conveniently drives two DC motors, capable of providing up to 2Amps of driving current to the motors, in the desired directions. Motors, in general draw large amounts of current. Therefore, the motors should not be directly connected to the Arduino board. Motor Driver Shield, such as L298 Ardumoto Shield converts the signals received from the Arduino to the power needed to drive the motors.

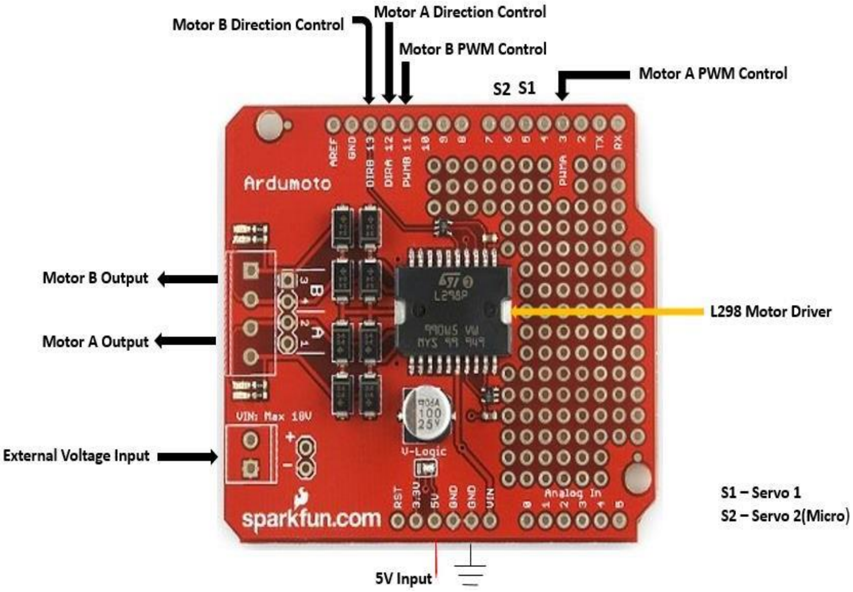


Fig: 3.3 L298 Arduino Motor Driver Shield [38]

The L298 is a dual – channel motor driver. It needs two power sources. The first is an external 9.6V input to provide power to drive the motors. The second is 5V is needed to power the Arduino. The Motor Shield is designed so that it can be conveniently mounted on top of the Arduino Uno board. In general, it can control up to two motors individually, comprising of a 2-wheeled system. Four DC motors drive as two sets of two wheels. The left two motors are designated as Motor A and the right two motors are designated as Motor B.

3.3.2 Bi-directional Control of DC motor using a H – Bridge:

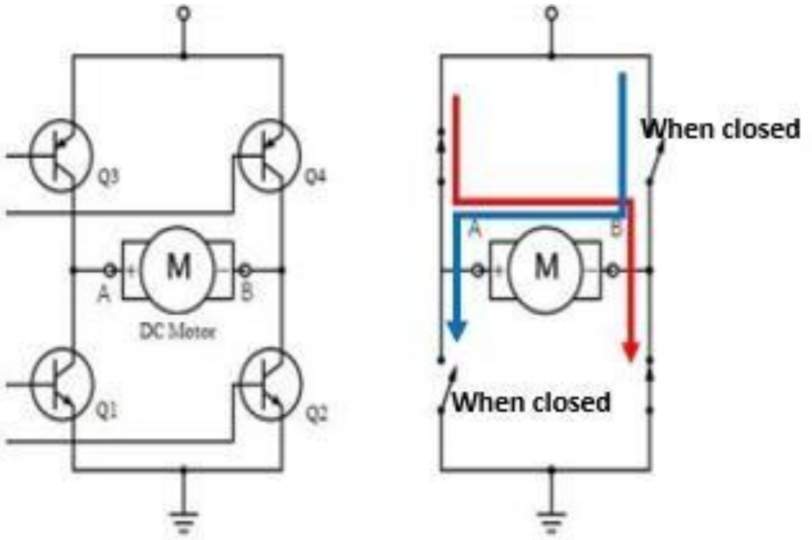


Fig: 3.3.1 L298 Dual Full H-Bridge [11]

DC motors rotate in two directions clockwise and anti-clockwise, depending on the direction of the current flow [8]. By changing the direction of current, the rotation of the motors is also changed. To achieve this, a H-bridge configuration of the motor driver is adopted. Four transistors, say, Q1, Q2, Q3 and Q4, are arranged to form an H-like structure, forming two sets each comprising of two transistors which determines the direction of rotation of the motors.

When transistors Q1, Q4 are switched on and Q2, Q3 are switched off, the DC motors rotate in clockwise direction, the robot moves forward and when transistors Q2, Q3 are switched on and Q1, Q4 are switched off, the motors rotate in anti-clockwise direction, the robot moves in the backward or in the reverse direction.

Table: 3.3.1 Direction of Motors A and B:

Motor A	Motor B	Robot motion
0	0	Rotate Left
0	1	Reverse
1	1	Rotate Right
1	0	Forward

0 = Counter-Clockwise direction, 1 = Clockwise direction.

The L298 Ardumoto Motor Driver Shield has two pins corresponding to the direction control of the two motors, Motor A and Motor B. Pin 12 controls the direction of Motor A and Pin 13 controls direction of Motor B. Logics 0 or LOW and 1 or HIGH are given to the pins through the Arduino IDE software turning the pins ON or OFF, which ultimately determines the direction of rotation of the motors. If Motor A is HIGH, which means that if transistors, Q1 and Q4 are ‘On’ while Q2 and Q3 are ‘Off’, then the left two motors rotate in clockwise direction. Then if Motor B is LOW, which means that if transistors, Q1 and Q4 are ‘Off’ while Q2 and Q3 are ‘On’, then the right two motors rotate in the counter-clockwise direction, moving the robot in the forward direction.

3.3.3 Pulse Width Modulation Control (PWM):

Another factor that determines the operation of DC motors is the speed control. The speed control of the DC motor is determined by the driving voltage. The change in the voltage changes the speed of the motors. [9]. The average voltage delivered to the load is controlled by turning the switch between supply and load ON and OFF at a fast pace [10]. PWM technique can control the speed of the motors and it also supports zero power losses [9,10].

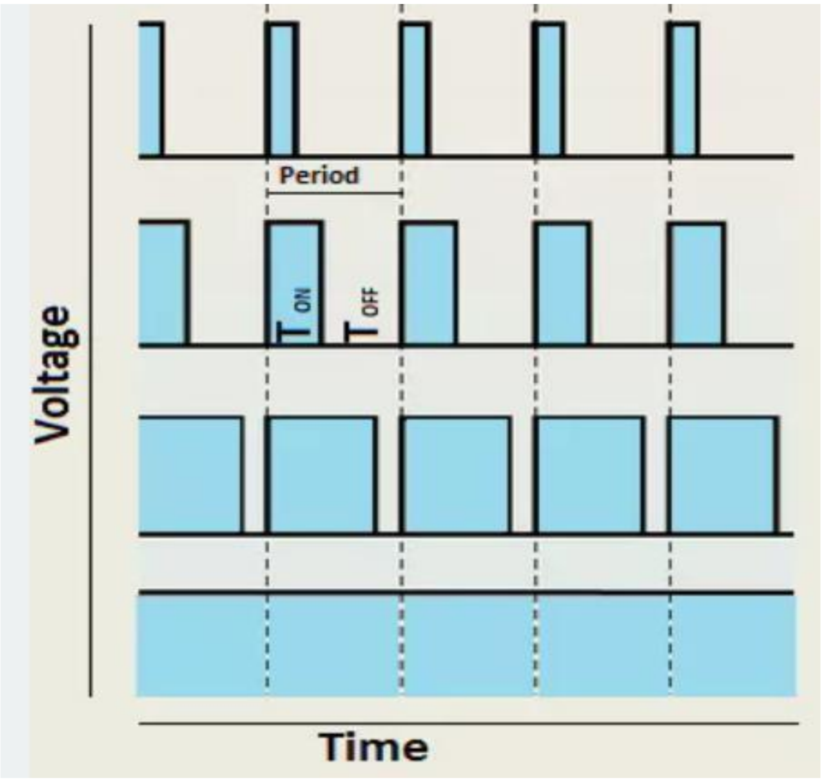


Fig: 3.3.2 PWM Duty Cycle [28]

Longer the switch is ‘On’, higher is the power supplied to the load, as compared to the time the switch is ‘Off’ [10]. This On and Off times are referred to as ‘Duty Cycle’ and it is expressed as a percentage. Higher duty cycle indicates that the switch if completely ‘On’.

$$\text{□□□□ □□□□□}\ (\%) = \frac{\text{□□ □□□□}}{\text{□□} + \text{□□□ □□□□}} * 100$$

$$= \frac{\text{□□ □□□□}}{\text{□□□□□ □□□□ □□□□□□}} * 100$$

Higher the Duty Cycle, higher is the speed of the motor [11]. In Arduino, 0% Duty Cycle

= 0 speed and 100% Duty Cycle = 255 speed, as the speed varies in the range of 0 – 255.

3.4 HC – 05 BLUETOOTH MODULE

The Bluetooth module used in this project is the HC – 05 Bluetooth module, which like any other Bluetooth device is a short range device used for data transmission and receiving. The frequency at which the device operates is 2.4GHz. The maximum range of a Bluetooth device is within the limits of 10 – 100 meters and the data transfer is done at a rate of 3Mbps. The process of data transfer between the Bluetooth module and the Arduino is called Serial Communication.

In a Serial Communication, a parameter called Baude rate determines the rate at which data is being transmitted. The set Baude rate for the communication between the Arduino and the Bluetooth module is 9600, which means that approximately about 9600 bits of data is transmitted per second [3].

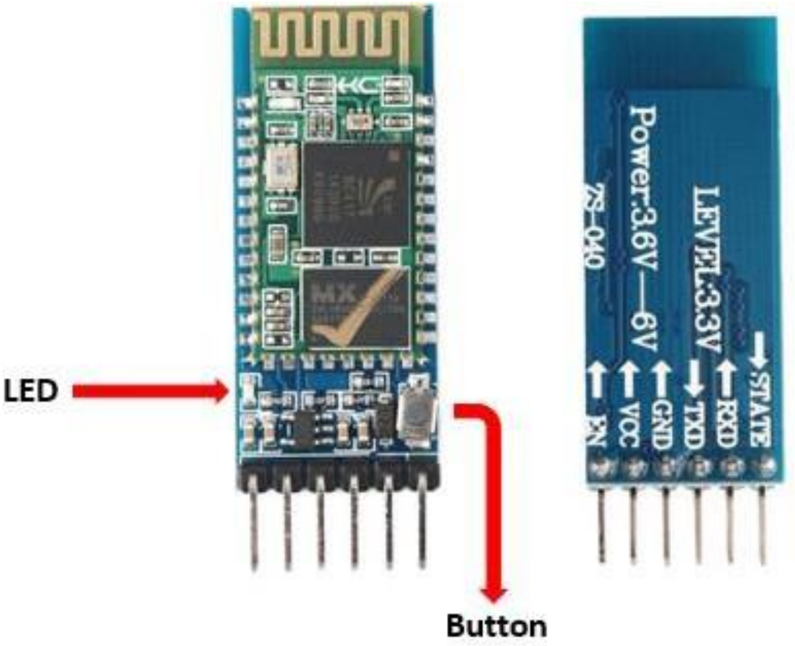


Fig: 3.4 HC-05 Bluetooth Module [39]

3.4.1 Specifications:

1. Operating Voltage: 3.6V – 6V.
2. Operating Current: 30mA.
3. The Bluetooth device can be operated at two different modes:
    i. Data Mode: During this mode, general serial communication such as sending and receiving data takes place.
    ii. AT Command Mode: During this mode, the default settings of the device can be changed.
4. HC – 05 Bluetooth Module is a 6 pin device.

a) **Enable Pin**: A switch between the two operating modes can be made by using the Enable/Key Pin. When the pin is set 'LOW', Data Mode is activated and when the pin is set 'HIGH', the AT Command Mode is activated.

b) **Tx (Transmitter) Pin**: This pin transmits the serial data to a device paired to the Bluetooth module.

c) **Rx (Receiver) Pin**: The serial data from another device is received by the Bluetooth module through this pin.

d) **Vcc**: The device is powered with a voltage within the range of 3.6V – 6V.

Note: When the device is connected and powered through the Arduino, it should be kept in mind that the pins on the Bluetooth module operate at a voltage of 3.3V. If the input is greater than the 3.3V limit, the HC – 05 Bluetooth module would get damaged.

e) **Ground**: The system is connected to the ground through this pin.

f) **State**: To check the proper functioning of the Bluetooth module, this pin is used and it is connected to the LED on the module.

g) **LED**: The status of the module is indicated by the LED. There are three stages,

   i. When the LED blinks continuously, it means that the device is waiting for a connection, in Data Mode.

   ii. When the LED blinks twice in 1 second, it means that a successful connection is established, in Data Mode.

   iii. When the LED blinks once in 2 seconds, it means that the Bluetooth module has entered Command Mode.

There is a Button on the device, which is used to control the Enable pin to switch between the Data Mode and the Command Mode.

3.5 ISD1820 VOICE RECORDER

The audio system in the current project is supported by ISD1820 device which is an Integrated Circuit chip voice recorder module is a single-chip single-message record and playback device. It supports easy recording and playback of messages. It can record for a duration of 8-20 seconds. The sample is 3.2K and a maximum of 20s for the recorder [20]. It provides zero-power message storage as the recordings are stored into an on-chip non-volatile memory. The recordings can be retained up to 100 years and approximately about 100,000 record/erase patterns can be achieved [21].

Fig: 3.5a ISD1820 Voice Recorder with a 0.5W 8 Ohm Speaker [29]

It is a 37 x 54 mm voice recording board. It has a record button and two types of playback-edge or level activated. Recording and playback functions could also be controlled by connecting the device to a microcontroller based board, such as an Arduino, Chipkit etc., [22].

3.5.1 Specifications [22]:

1. Power input, Vcc: 2.4V – 5V

   Note: Vcc should not exceed 3.3V, otherwise it may damage the device.

2. Ground, GND.
3. Record button.
4. Playback – edge or level activated.
5. Separate recording and playback pins are available, which can be connected and controlled by microcontroller based boards.
6. Automatic power-down mode.
7. 8-20 seconds recording duration.
8. By replacing a single resistor or by using a potentiometer, sample rate and the duration of recording can be changed.
9. An 8 Ohm, 0.5W Speaker.
10. An on-board microphone.

3.5.2 Schematic:

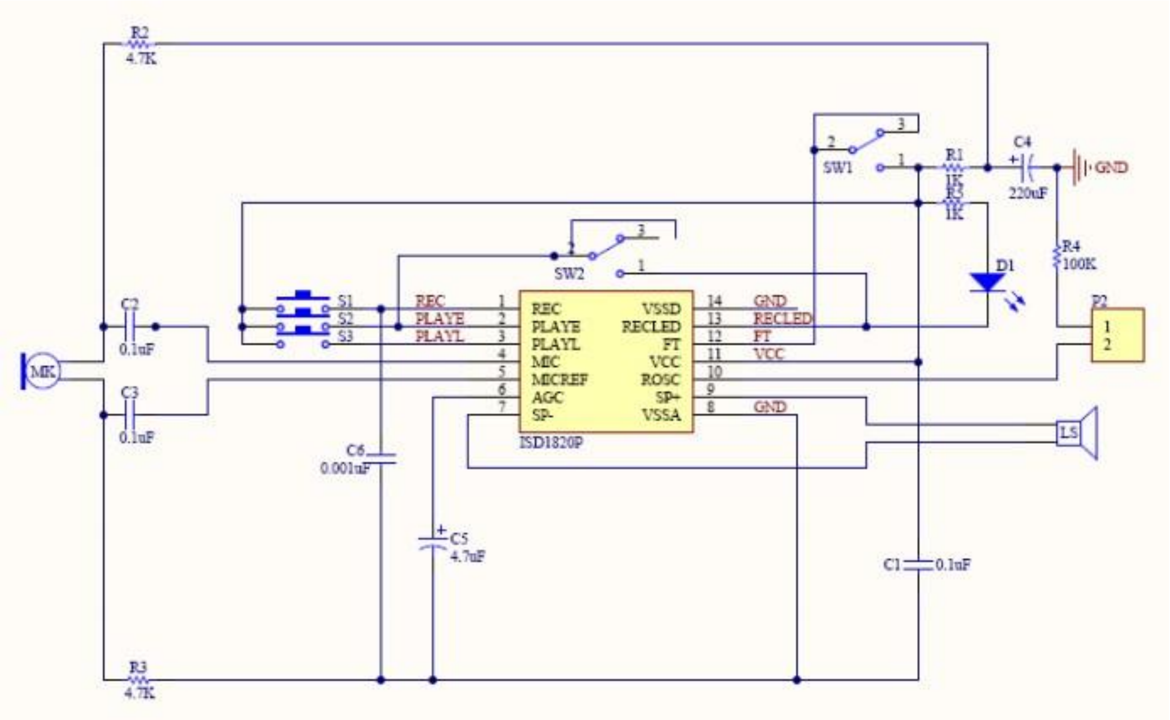


Fig: 3.5b ISD1820 Schematic diagram [30]

Table: 3.5 ISD1820 voice recorder specifications**:**

ROSC/R4	Duration	Sample Rate	Bandwidth
80K Ohm	8 secs	8.0KHz	3.4KHz
100K Ohm	10 secs	6.4KHz	2.6KHz
120K Ohm	12 secs	5.3KHz	2.3KHz
160K Ohm	16 secs	4.0KHz	1.7KHz
200K Ohm	20 secs	3.2KHz	1.3KHz

ROSC – Resistor Controlled Oscillator Input

The duration of recording is within the range of 8 – 20 seconds. But it is set at a default value of approximately 10 seconds as it is connected to a 100K Ohm resistor. The record duration can be increased to its maximum range by changing the sample rate and frequency by adding a suitable external resistor [23].

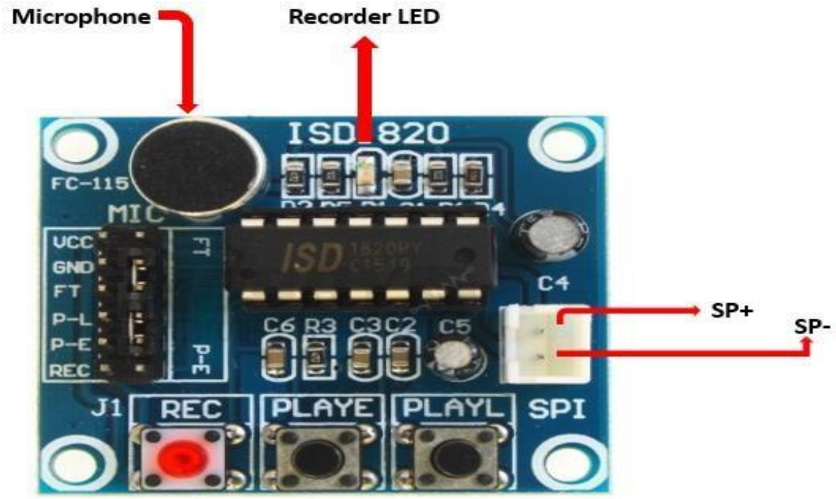


Fig: 3.5c ISD1820 Voice Recorder indicating the components [40]

3.5.3 Pin Description:

1. REC: The device starts to record when the REC button is pushed and when it is pushed, the Recorder LED turns on to indicate that the device is recording. When the recorder module is connected to a microcontroller device, the device starts recording when the REC is HIGH and continues to remain in that state until REC is LOW or until the completion of the maximum recorder duration, which is after 20 seconds.

REC outweighs both the playback signals. If REC signal is HIGH or the REC button is pushed, playback is immediately stopped and the device starts to record. If the REC pin remains in HIGH state, there exists an increase in the consumed current. An End-of-Message (EOM) indication is recorded which immediately eases a succeeding playback cycle.

2. PLAYE: It is Edge-activated playback pin. When the signal is HIGH it plays back the message recorded. Playback does not have any effect of the signal being LOW and continues to play the recording until the End-of-message indication is recorded internally.
3. PLAYL: It is Level-activated playback. As long as the signal is HIGH or the button is held pressed, the pin is activated and plays the message but as soon as the button is released it stops playback.

   Note: The device has an automatic power-down mode which makes it to go to standby mode when the REC, PLAYE or PLAYL signals are LOW.
4. Speaker Outputs: The SP+ and SP- are the speaker outputs available in the recorder module, which are internally connected between a 50K Ohm resistor.
5. Microphone: There is an on-chip preamplifier in the recorder module to which the microphone transmits its signals. The gain of the preamplifier is controlled by an Automatic Gain Control (AGC) circuit. If an external microphone is desired to be used, it is supposed to be AC coupled to the pin through a series capacitor. The low-frequency cut-off for the 1800 passband is regulated the series capacitor and a 10K Ohm resistance [23].

3.6 SERVO MOTORS

A robotic arm-like structure is built for the animal study using servo motors. Two 0 – 180-degree servo motors are used supporting two degrees of freedom: an elbow joint and a wrist. It is intended to design and operate the servos so that they form structure which will be able to deliver treats like a nut or an acorn to the squirrel. This would help to establish an acquaintance and easy acceptance of the robot into the squirrel’s habitat.

A standard rotation servo is opted for the movement of the elbow joint and a micro servo is opted. The micro servo is attached to an object where nuts or other treats are placed, which performs the wrist action of picking up, holding and dropping the nut in the targeted animal’s direction. These servos are powered directly from an Arduino board.

Servo motors, unlike the usual DC motors, can be placed at any desired angle or a position using a control signal. As long as the control signal remains unchanged, the motor retains its position. This type of control is proved to be very accommodating in order to control robotic arms and other unmanned structures [24].

Standard servo **Micro servo**

Fig: 3.6 HITec HS-322HD and 9g Micro Servo [31,32]

HiTec HD-322 HD standard servo motor:

Specifications:

1. Weight – 42.81g. It is uses a Hitec Karbonite gear-train.
2. Torque at 4.8V – 3.0 kg.cm.
3. Torque at 6.0V – 3.7 kg.cm.
4. Torque at 7.4V – 0.0 kg.cm.
5. Speed – 0.19/0.15 sec.
6. Bearing type – Nylon.
7. Motor type – 3 pole.
8. Dimensions – 39.88 x 19.81 x 36.32 mm.

Vcc = Red wire, 5V.

Gnd = Brown.

Control signal = Orange.

Micro servo motor:

Specifications [26]:

1. Dimensions – 23.2 x 12.5 x 22.0 mm.
2. Wire length – 20 cm.
3. Speed – 0.12 sec/60 degrees.
4. Weight – 9g.
5. Torque – 1.3 kg.cm.
6. Voltage – 4.8V – 6V.

7. Plastic gears.

Vcc = Red wire, 5V.

Gnd = Black

Control signal = Yellow.

The servo motors are controlled by sending a PWM signal through a control signal wire from the Arduino board. These signals are pulses of varying lengths. The desired angle correlates with the length of the pulses. The motors rotate within the range of 0 – 180 degrees. The minimum angle 0 degrees corresponds to 1 millisecond length of a pulse and the maximum angle 180 degreed corresponds to 2 millisecond length of a pulse. It means that the length of the pulses for the range of 0 – 180 degrees lies within the limits of 1 – 2 milliseconds [24].

3.7 ArduCAM ESP8266 ARDUINO Wi-Fi BOARD AND 2MP MINI CAMERA:

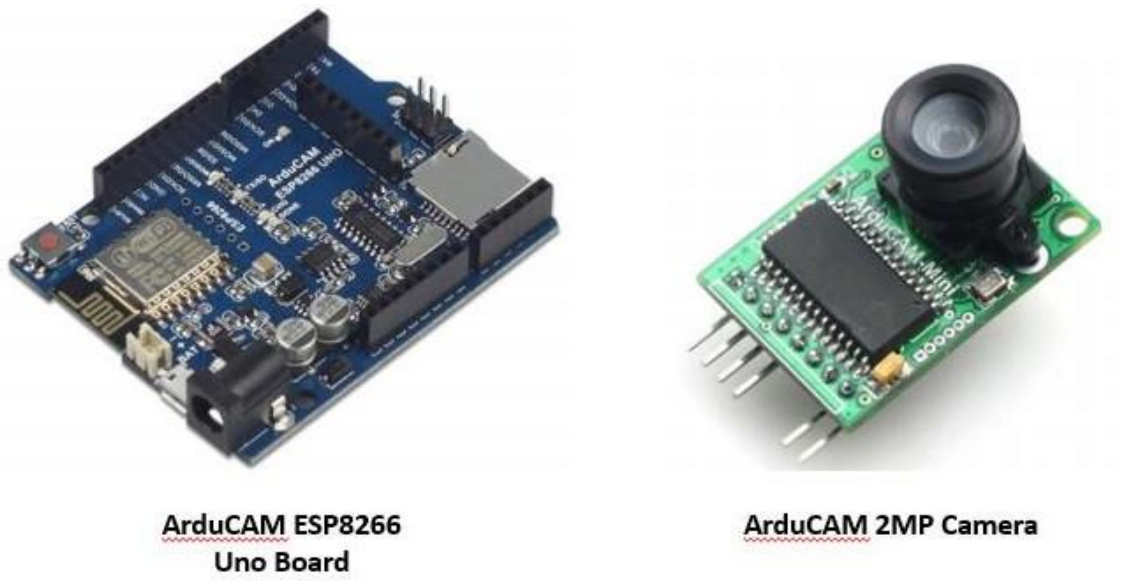


Fig: 3.7a ArduCAM ESP8266 Uno board and 2MP Mini Camera [33,34]

The video system is supported by a 2MP ArduCAM camera module which is mounted on a ArduCAM ESP8266 Arduino board. This Arduino board is similar to the standard ATmega 328 microcontroller based Arduino Uno but it has some additional features, such as:

1. It has an inbuilt Wi – Fi system.
2. It is compatible with both 2MP and 5MP ArduCAM mini camera modules.
3. It supports recharging Lithium battery power of 3.7/500 mA approximately.
4. An inbuilt SD card slot which can save the images and videos captured by the camera modules.

5. It is compatible with Arduino IDE.

There are a set of 8 pins on ArduCAM ESP8266 Uno board with respect to which the 2MP Mini camera module mounted [34].

Fig: 3.7b 2MP Mini Camera mounted on ArduCAM ESP8266 board [35]

Table: 3.7 ArduCAM ESP8266 Pin Description:

2MP Mini ArduCAM Camera	**ArduCAM ESP8266 Uno**
3.3V	3.3V
GND	GND
SCL	SCL
SDA	SDA
CSn	CSn

MISO	MISO
MOSI	MOSI
SCLK	SCLK

The ArduCAM ESP8266 UNO board libraries are intended to be installed to get the board activated.

3.8 SYSTEM SETUP AND OPERATION

The main control unit of the entire system is the ATmega328 microcontroller based Arduino board, which is powered by a 9.6V DC battery.

3.8.1 Power System

A 9.6V battery is used to power all the components in the system. The Motor driver shield is mounted on to the Arduino from where it is gets the power supply. Arduino has an inbuilt voltage regulator which supplies 3.3V and 5V.

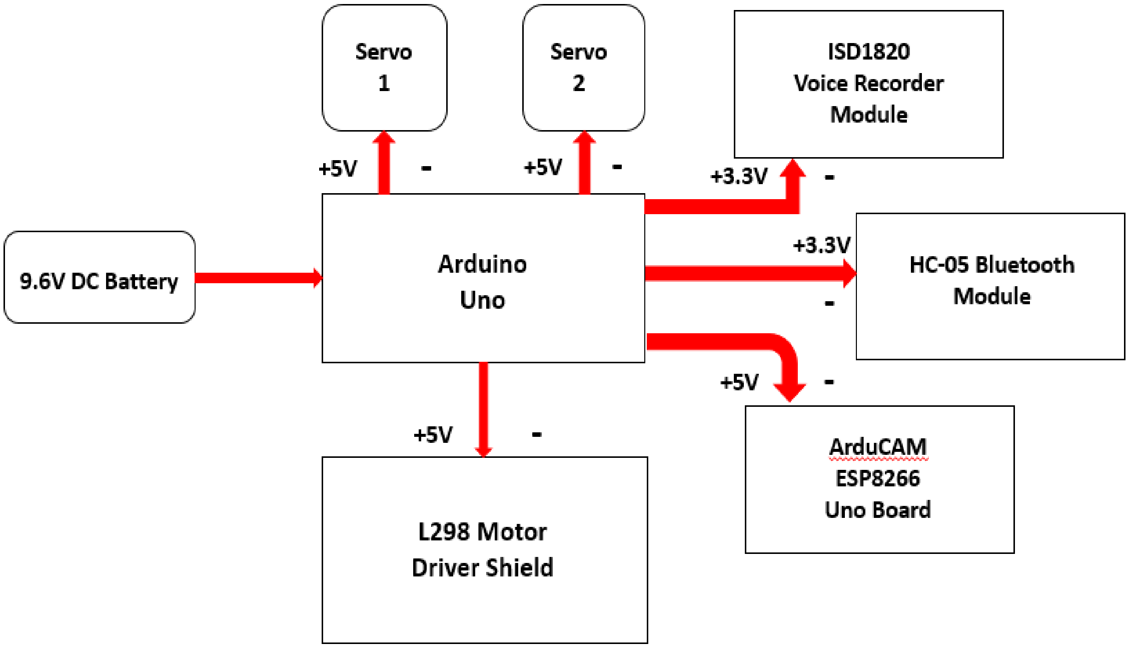


Fig: 3.8.1 Power distribution diagram of the Squirrel Robot

1. Bluetooth:

   Vcc = 3.3V.

   Ground.

2. ISD8266 voice recorder:

   Vcc = 3.3V.

   Ground.

3. ArduCAM ESP8266 Uno board:

   Vin = 5V.

   Ground.

4. Servo 1

   Vin = 5V.

   Ground.

5. Servo 2

   Vin = 5V.

   Ground.

6. Motor A (Left motors) and Motor B (Right motors)

   Vin = 9.6V

   Ground

The ArduCAM ESP8266 board works by itself without any influence of other components in the system, but it is powered through the 5V regulated pin of the Arduino.

3.8.2 Schematic diagram and description

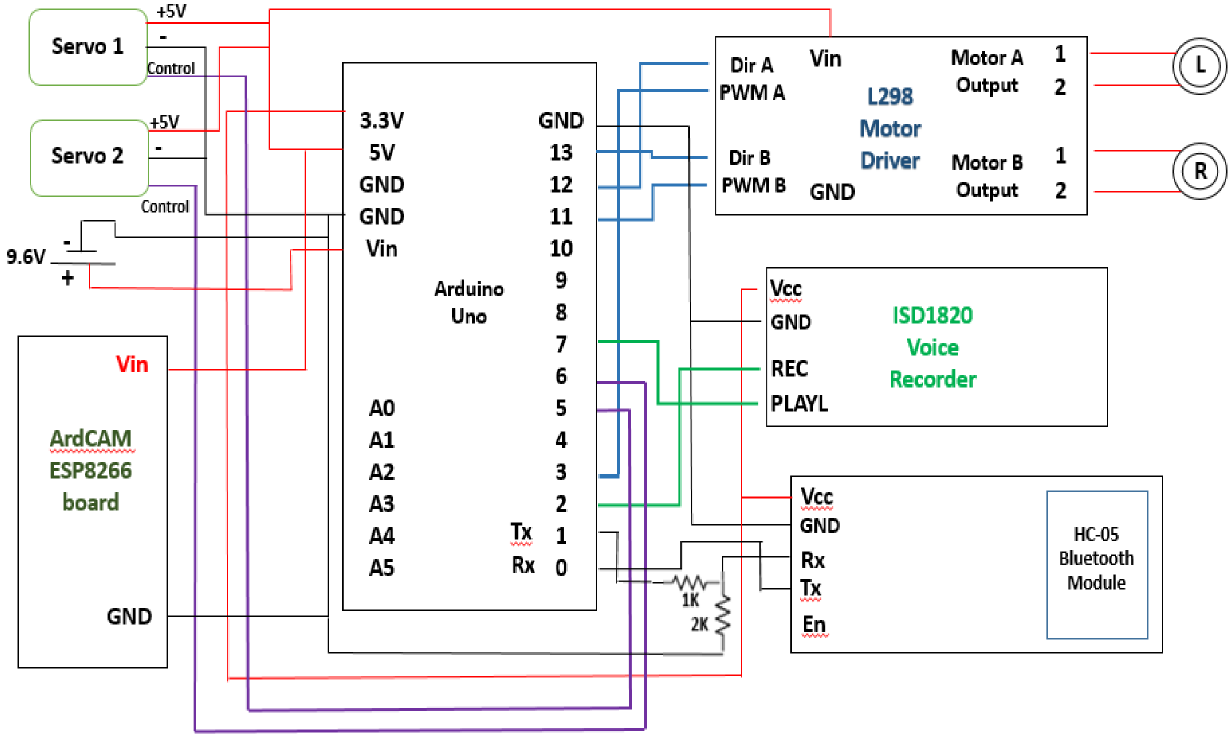


Fig: 3.8.2 Schematic diagram of the Squirrel Robot

1. The motor shield drives the left and the right DC motors. The direction of the motors is determined by the I/O pins 12 and 13, and the speed or the rotation control are determined by the I/O pins 3 and 11 for Motor A and Motor B respectively.
2. In ISD8266 voice recorder and playback module, the recorder is connected to I/O pin 2 and the playback is connected to pin 7.
3. Servos are controlled by passing a Pulse Width Modulation signal (PWM) through them. Servo 1 or the standard servo is connected to PWM pin 5 and the micro or Servo 2 motor is connected to PWM pin 6.

All these components are controlled by an Android application. A Bluetooth module connected to the serial communication ports, 0 & 1, also called Rx and Tx pins acts as an interface between the smart phone and the Arduino board.

3.8.2.1 HC – 05 Bluetooth Module and the Arduino interface:

The Vcc pin is connected to the 3.3V pin of the Arduino.

The Tx pin of the Bluetooth module is connected to the Rx pin of the Arduino.

The transmitter pin, Tx of the Arduino and the receiver pin, Rx of the Bluetooth module cannot be connected directly. This is because, the output voltage of the Tx pin of the Arduino(5V) is higher than the input voltage of the Rx pin of the Bluetooth module (3.3V). If these two pins are connected directly, then it will cause damage to the Bluetooth device. Therefore, in order to overcome this problem, a voltage divider circuit with a 1K and 2K Ohm resistors is introduced between the Tx of the Arduino and the Rx of the Bluetooth device, which ensures a safe operating voltage.

3.8.2.2 ArduCAM ESP8266:

The ArduCAM ESP8266 2MP Mini Camera module works independent of the rest of the robot system but serves the purpose of a having a live streaming video showing the behavioral patterns of squirrels in response to the influence of the remote-controlled squirrel robot system.

The ArduCAM ESP8266 board has an inbuilt Wi-Fi feature, that helps with the live video streaming to a computer system or a laptop.

3.8.3 Communication System

The motion and the operation of the system is controlled by an Android application. This application has various controls in it which supports the operation of the components in the robotic model. The communication between the Robot and the smart phone is achieved with the help of an HC-05 Bluetooth module. This Bluetooth module acts as a medium through which data transmission takes place.

A unique and an effective single-byte, single–command communication system has been devised. This system allows the conveyance of multiple commands and operations required to control the Squirrel Robot system, within the limit of one byte.

The data that is transferred is in the simple form of 1-byte memory, which is a range of 0 to 255 decimal numbers. The communication between the smart phone and the robot is through an 8-bit message, which represents a set of 256 values. In the App Inventor software, a tool called the Bluetooth Client is attached, which provides the required connectivity to the smart phone which thereby supports a link with the HC-05 Bluetooth module. The application is developed in such a way that, each block corresponding to a particular operation of the control in the app, sends 1-byte number from the range of 0 – 255. There are 12 different operations in the robotic model, which are

- Forward
- Reverse
- Rotate left
- Rotate right
- Stop

- Speed control of the motors
- Start recording
- Stop recording
- Start playback
- Stop playback
- Servo 1 movement
- Servo 2 movement

1 byte of memory is shared amongst all the 12 operations. Each operation is given a number or a range of numbers between 0 and 255. The first 9 decimal numbers are pertained to the following set of operations,

Table: 3.8.3a Division of 1-byte memory for control – motion and voice recorder

Range of 1-byte memory	**Corresponding Operation**
1 (0000 0001)	FORWARD
2 (0000 0010)	REVERSE
3 (0000 0011)	ROTATE LEFT
4 (0000 0100)	ROTATE RIGHT
5 (0000 0101)	STOP
6 (0000 0110)	START RECORDING
7 (0000 0111)	STOP RECORDING
8 (0000 1000)	START PLAYBACK
9 (0000 1001)	STOP PLAYBACK

This means that when a desired button or a control is pressed in the app, it sends the corresponding number to the Bluetooth module. This HC-05 Bluetooth module receives and transmits this number to the Arduino which activates the appropriate pin. For instance, numbers 1 – 5 pertain to the direction of the motors. When a direction control button or a control is pressed, the data or the number corresponding to the control is received by the Bluetooth module and it is transmitted to the Arduino. This data transmission results in activating the relevant pin on the Arduino, which causes the robot to stop or move in the appropriate direction. For, the number ‘2 ’, corresponding to the binary number, 0000 0010 is given, then the robot moves in the reverse direction.

Similarly, when a range of 6 – 9 decimal data is transmitted, it activates the ISD1820 voice recorder to perform the suitable actions.

Table: 3.8.3b Division of 1-byte memory for control – servo and speed control

Range of 1-byte memory	**Corresponding Operation**
10 – 200	SPEED CONTROL OF MOTORS
201 – 229	SERVO 1 CONTROL
230 – 255	SERVO 2 CONTROL

The speed of the motors is determined by a slider control in the app. Any data number within the range of 10 – 200, corresponds to the speed control of the motors. These numbers are mapped as 0 – 255 in the Arduino such that it determines the minimum and maximum speed of the motors, 0 being the minimum and the motors do not run and 255 being the maximum and the motors run at full speed. For instance, if the number 105 is provided then it would be approximately 50% duty cycle.

Now, the data between 201 and 255 is divided into two sets of values to operate two servomotors. 201 – 229, mapped as 0 – 180 degrees in the Arduino which is about 6-degree resolution, corresponds to the motion of standard servo or Servo 1, which is at the elbow of the robot-arm and 230 – 255, mapped as 0 – 90 degrees in the Arduino, which is about 2.7 degrees' resolution, corresponds to the motion of micro or servo 2 which is at the wrist of the robot arm. For example, if the number 215 is given, the Servo1 is positioned at 90 degrees. Similarly, when the number 242 is given, the Servo2 is positioned at 45 degrees.

3.8.4 Development of an Android App to control the Squirrel Robot system using MIT App Inventor:

MIT App Inventor is a visual programming web application developer which helps to build fully functioning apps for smartphones. It supports a user-friendly and understandable blocks and designing system allowing everyone to work on it.

The Android app developed using the MIT App Inventor may look as follows,

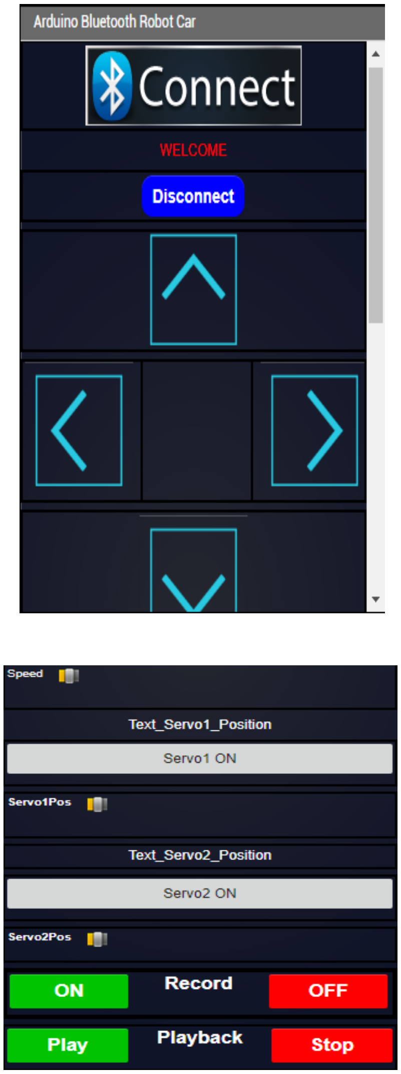
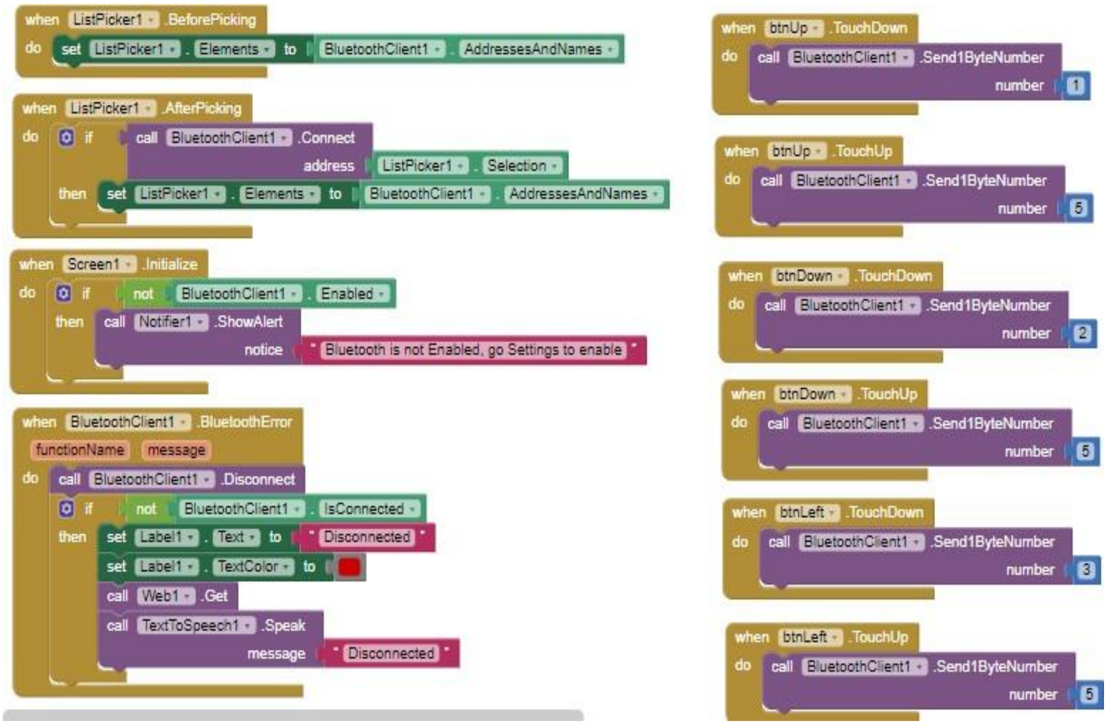
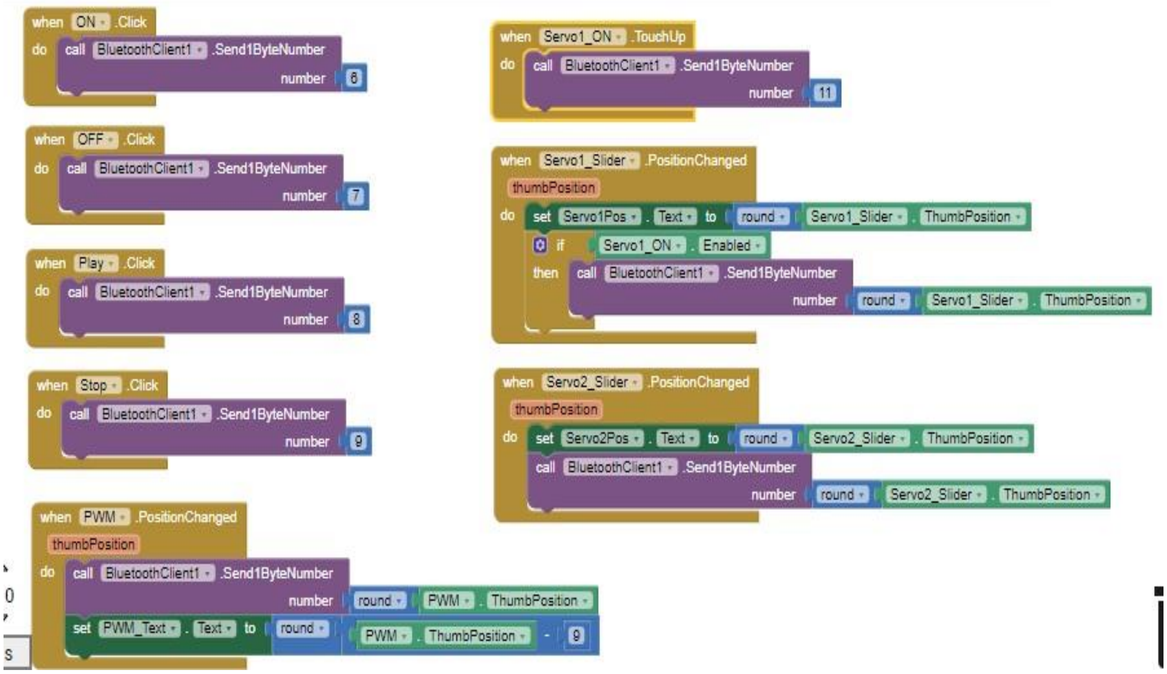


Fig: 3.8.4 Android App controls

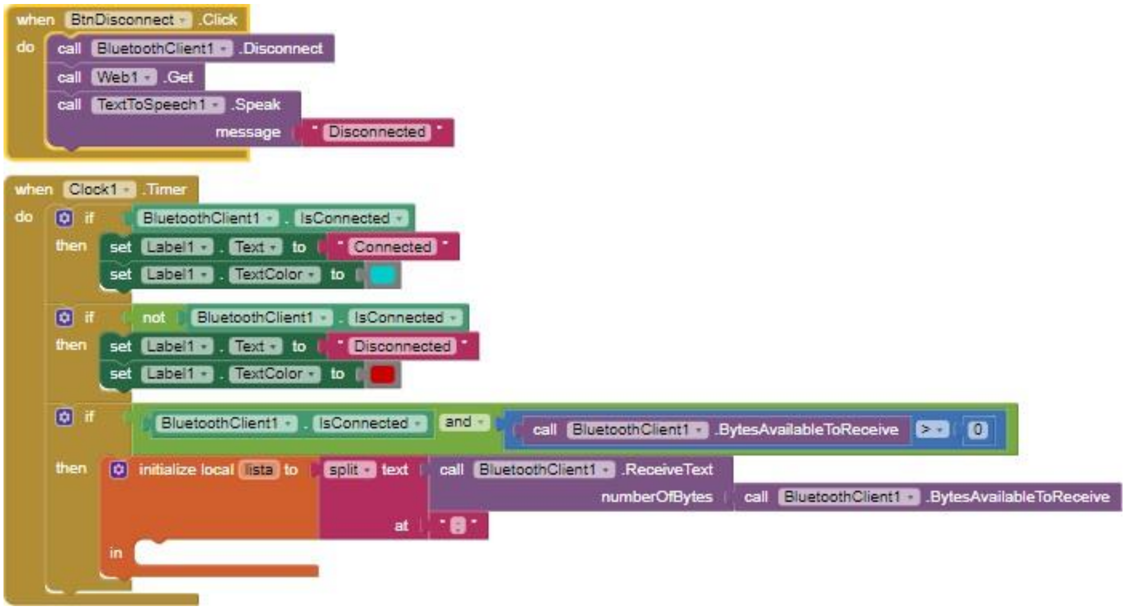
There are two parts of the MIT App Inventor, the visual graphic section which allows the user to design the appearance of the App in a desired manner and then there is a functional part that describes the function performed by each of the graphics. Each graphic has a corresponding block that defines its functionality.



```
when ListPicker1 .BeforePicking
do  set ListPicker1 . Elements to BluetoothClient1 . AddressesAndNames

when ListPicker1 .AfterPicking
do  if    call BluetoothClient1 .Connect
                               address  ListPicker1 . Selection
    then  set ListPicker1 . Elements to BluetoothClient1 . AddressesAndNames

when Screen1 .Initialize
do  if    not BluetoothClient1 . Enabled
    then  call Notifier1 .ShowAlert
                         notice  " Bluetooth is not Enabled, go Settings to enable "

when BluetoothClient1 .BluetoothError
 functionName  message
do  call BluetoothClient1 .Disconnect
    if    not BluetoothClient1 . IsConnected
    then  set Label1 . Text to " Disconnected "
          set Label1 . TextColor to
          call Web1 .Get
          call TextToSpeech1 .Speak
                            message  " Disconnected "

when btnUp .TouchDown
do  call BluetoothClient1 .Send1ByteNumber
                          number  1

when btnUp .TouchUp
do  call BluetoothClient1 .Send1ByteNumber
                          number  5

when btnDown .TouchDown
do  call BluetoothClient1 .Send1ByteNumber
                          number  2

when btnDown .TouchUp
do  call BluetoothClient1 .Send1ByteNumber
                          number  5

when btnLeft .TouchDown
do  call BluetoothClient1 .Send1ByteNumber
                          number  3

when btnLeft .TouchUp
do  call BluetoothClient1 .Send1ByteNumber
                          number  5
```



```
when ON.Click
do  call BluetoothClient1.Send1ByteNumber
         number  6

when OFF.Click
do  call BluetoothClient1.Send1ByteNumber
         number  7

when Play.Click
do  call BluetoothClient1.Send1ByteNumber
         number  8

when Stop.Click
do  call BluetoothClient1.Send1ByteNumber
         number  9

when PWM.PositionChanged
  thumbPosition
do  call BluetoothClient1.Send1ByteNumber
         number  round  PWM.ThumbPosition
    set PWM_Text.Text to  round  PWM.ThumbPosition - 9

when Servo1_ON.TouchUp
do  call BluetoothClient1.Send1ByteNumber
         number  11

when Servo1_Slider.PositionChanged
  thumbPosition
do  set Servo1Pos.Text to  round  Servo1_Slider.ThumbPosition
    if    Servo1_ON.Enabled
    then  call BluetoothClient1.Send1ByteNumber
               number  round  Servo1_Slider.ThumbPosition

when Servo2_Slider.PositionChanged
  thumbPosition
do  set Servo2Pos.Text to  round  Servo2_Slider.ThumbPosition
    call BluetoothClient1.Send1ByteNumber
         number  round  Servo2_Slider.ThumbPosition
```



```
when BtnDisconnect.Click
do  call BluetoothClient1.Disconnect
    call Web1.Get
    call TextToSpeech1.Speak
         message  " Disconnected "

when Clock1.Timer
do  if    BluetoothClient1.IsConnected
    then  set Label1.Text to  " Connected "
          set Label1.TextColor to  🟦
    if    not  BluetoothClient1.IsConnected
    then  set Label1.Text to  " Disconnected "
          set Label1.TextColor to  🟥
    if    BluetoothClient1.IsConnected  and  call BluetoothClient1.BytesAvailableToReceive  >  0
    then  initialize local lista to  split text  call BluetoothClient1.ReceiveText
                                                       numberOfBytes  call BluetoothClient1.BytesAvailableToReceive
                                          at  " : "
          in
```

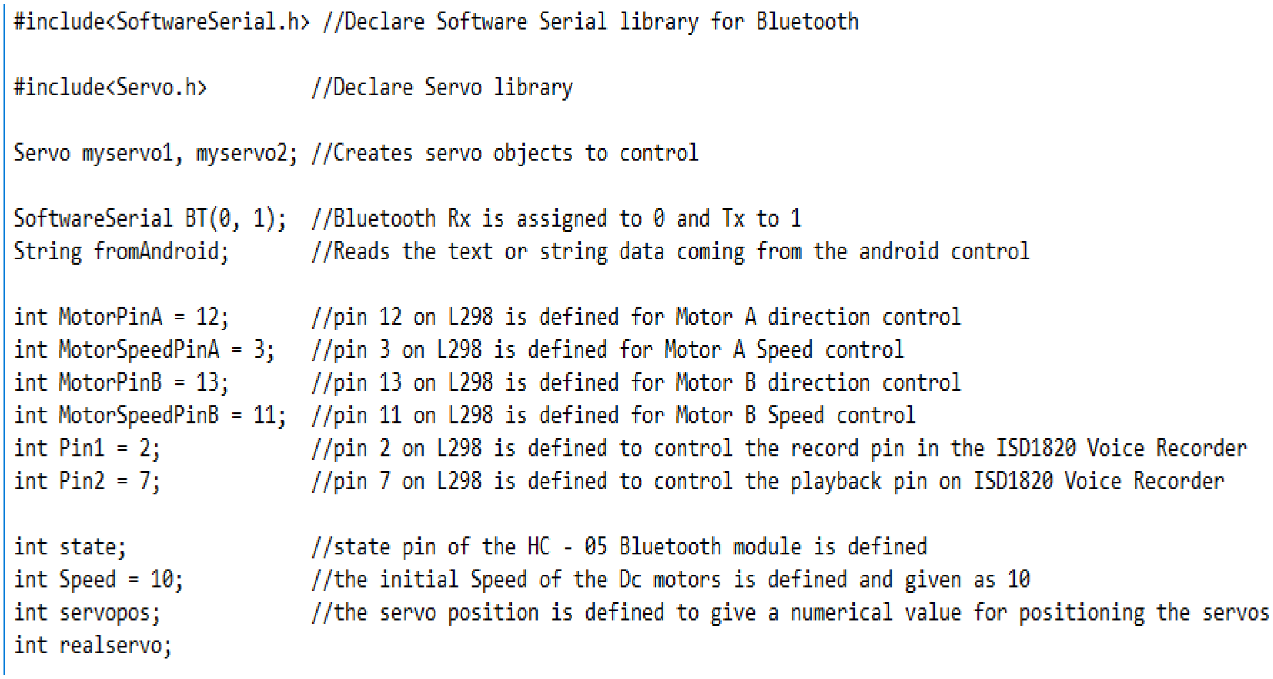
Every block here corresponds to every control in the Android app.

3.8.5 Arduino Code:

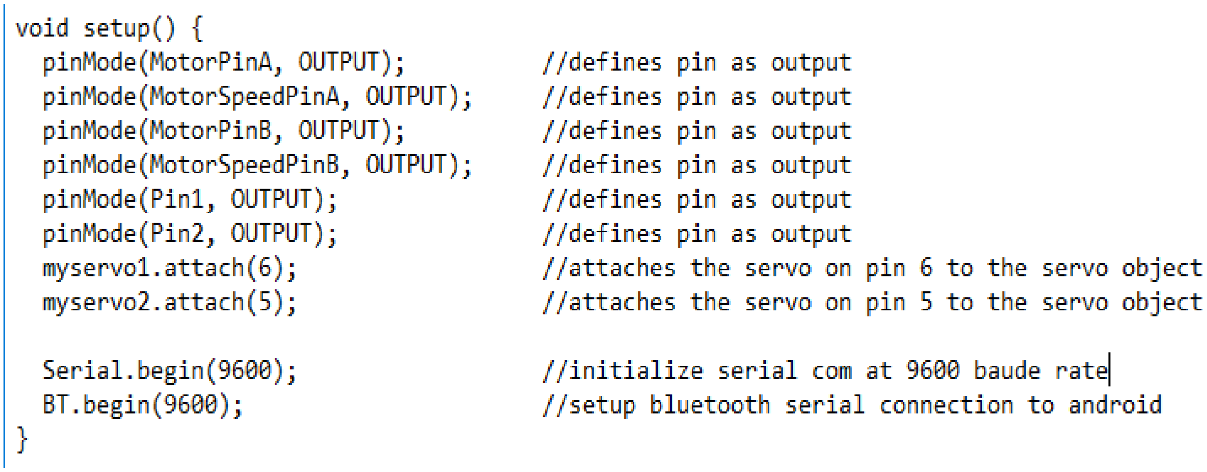
The Arduino receives 1-byte of information form the Bluetooth module and translates that into a function that drives the motors, servos and operates the recording system. The program for the robot system is written in an Arduino IDE software. It has inbuilt libraries, which supports the connection and operation of components such as: Bluetooth communication and servo motor control.

The appropriate program for the system control is written, it is compiled to check for any possible error and remedy them. After successful compilation, the program is uploaded to the Arduino Uno board using a USB cable.

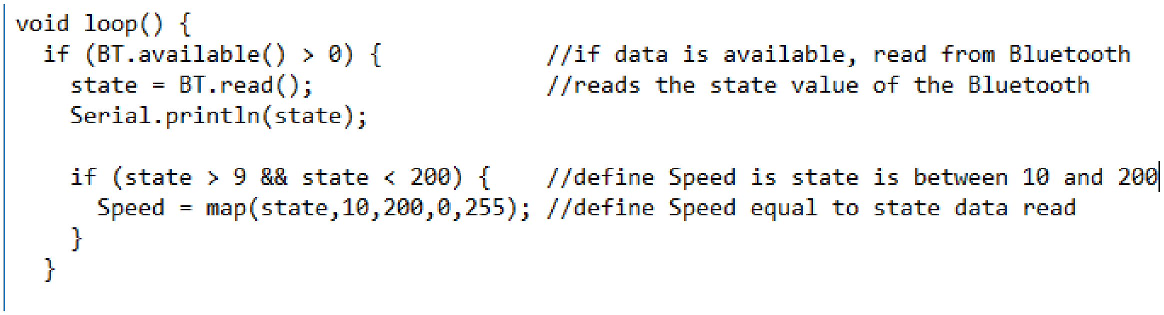
The following is the program for the remote control squirrel robot system:



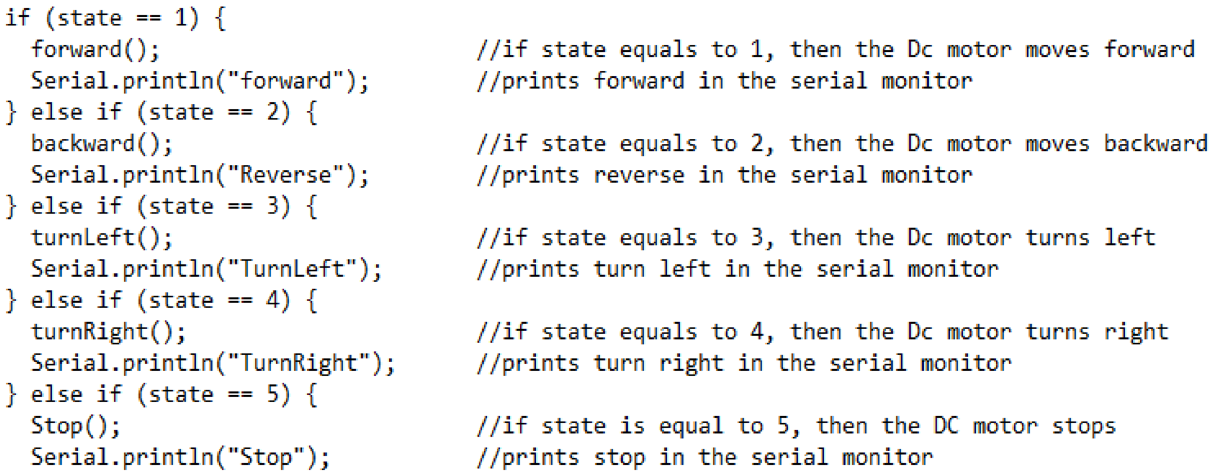
```
#include<SoftwareSerial.h> //Declare Software Serial library for Bluetooth

#include<Servo.h>          //Declare Servo library

Servo myservo1, myservo2; //Creates servo objects to control

SoftwareSerial BT(0, 1);  //Bluetooth Rx is assigned to 0 and Tx to 1
String fromAndroid;       //Reads the text or string data coming from the android control

int MotorPinA = 12;       //pin 12 on L298 is defined for Motor A direction control
int MotorSpeedPinA = 3;   //pin 3 on L298 is defined for Motor A Speed control
int MotorPinB = 13;       //pin 13 on L298 is defined for Motor B direction control
int MotorSpeedPinB = 11;  //pin 11 on L298 is defined for Motor B Speed control
int Pin1 = 2;             //pin 2 on L298 is defined to control the record pin in the ISD1820 Voice Recorder
int Pin2 = 7;             //pin 7 on L298 is defined to control the playback pin on ISD1820 Voice Recorder

int state;                //state pin of the HC - 05 Bluetooth module is defined
int Speed = 10;           //the initial Speed of the Dc motors is defined and given as 10
int servopos;             //the servo position is defined to give a numerical value for positioning the servos
int realservo;
```



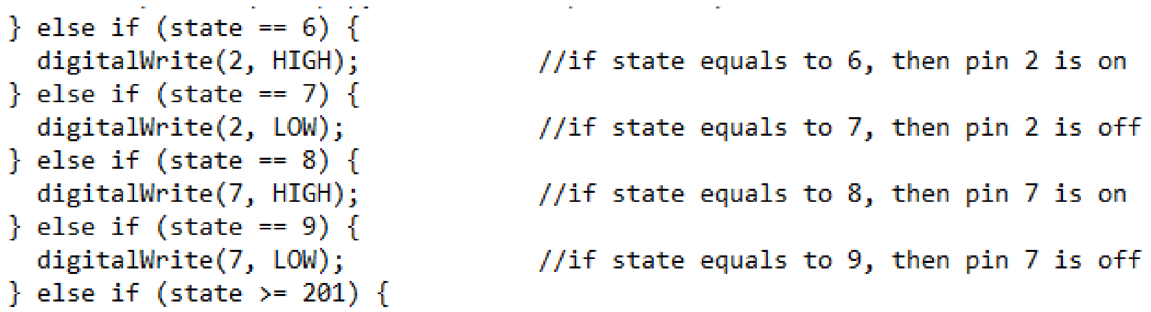
```
void setup() {
  pinMode(MotorPinA, OUTPUT);          //defines pin as output
  pinMode(MotorSpeedPinA, OUTPUT);     //defines pin as output
  pinMode(MotorPinB, OUTPUT);          //defines pin as output
  pinMode(MotorSpeedPinB, OUTPUT);     //defines pin as output
  pinMode(Pin1, OUTPUT);               //defines pin as output
  pinMode(Pin2, OUTPUT);               //defines pin as output
  myservo1.attach(6);                  //attaches the servo on pin 6 to the servo object
  myservo2.attach(5);                  //attaches the servo on pin 5 to the servo object

  Serial.begin(9600);                  //initialize serial com at 9600 baude rate
  BT.begin(9600);                      //setup bluetooth serial connection to android
}
```



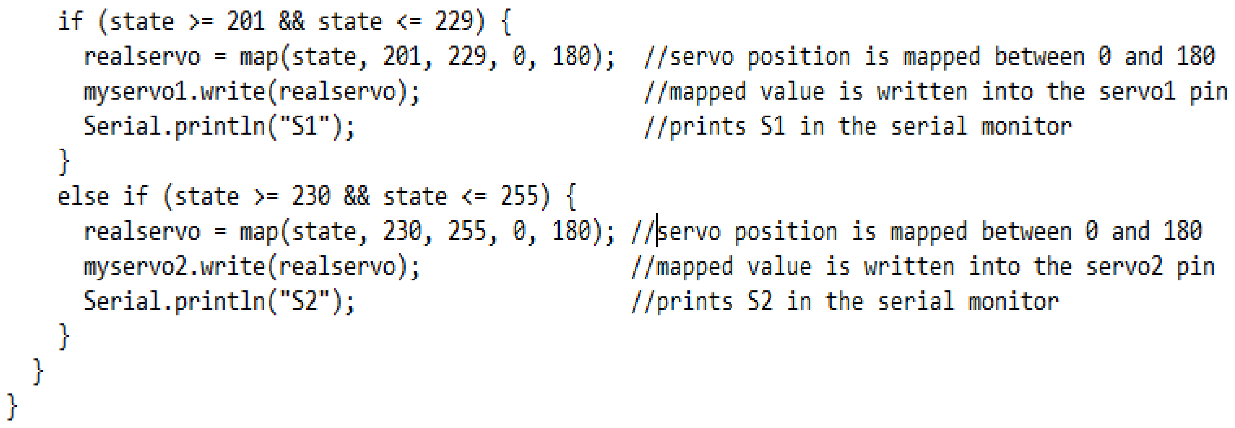
```
void loop() {
  if (BT.available() > 0) {            //if data is available, read from Bluetooth
    state = BT.read();                 //reads the state value of the Bluetooth
    Serial.println(state);

    if (state > 9 && state < 200) {    //define Speed is state is between 10 and 200
      Speed = map(state,10,200,0,255); //define Speed equal to state data read
    }
  }
```



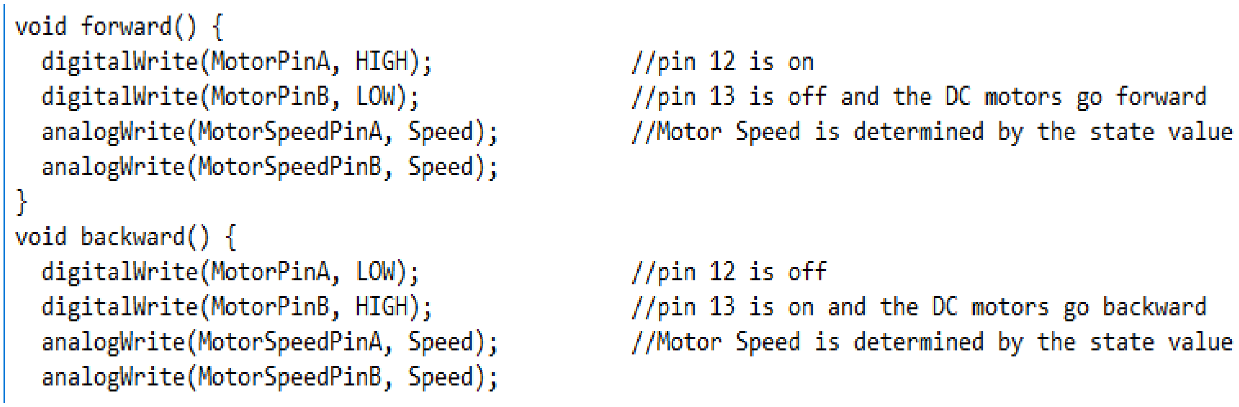
```
if (state == 1) {
  forward();                        //if state equals to 1, then the Dc motor moves forward
  Serial.println("forward");        //prints forward in the serial monitor
} else if (state == 2) {
  backward();                       //if state equals to 2, then the Dc motor moves backward
  Serial.println("Reverse");        //prints reverse in the serial monitor
} else if (state == 3) {
  turnLeft();                       //if state equals to 3, then the Dc motor turns left
  Serial.println("TurnLeft");       //prints turn left in the serial monitor
} else if (state == 4) {
  turnRight();                      //if state equals to 4, then the Dc motor turns right
  Serial.println("TurnRight");      //prints turn right in the serial monitor
} else if (state == 5) {
  Stop();                           //if state is equal to 5, then the DC motor stops
  Serial.println("Stop");           //prints stop in the serial monitor
```



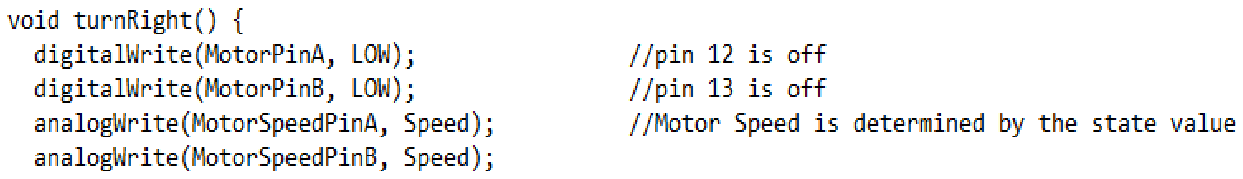
```
} else if (state == 6) {
  digitalWrite(2, HIGH);          //if state equals to 6, then pin 2 is on
} else if (state == 7) {
  digitalWrite(2, LOW);           //if state equals to 7, then pin 2 is off
} else if (state == 8) {
  digitalWrite(7, HIGH);          //if state equals to 8, then pin 7 is on
} else if (state == 9) {
  digitalWrite(7, LOW);           //if state equals to 9, then pin 7 is off
} else if (state >= 201) {
```



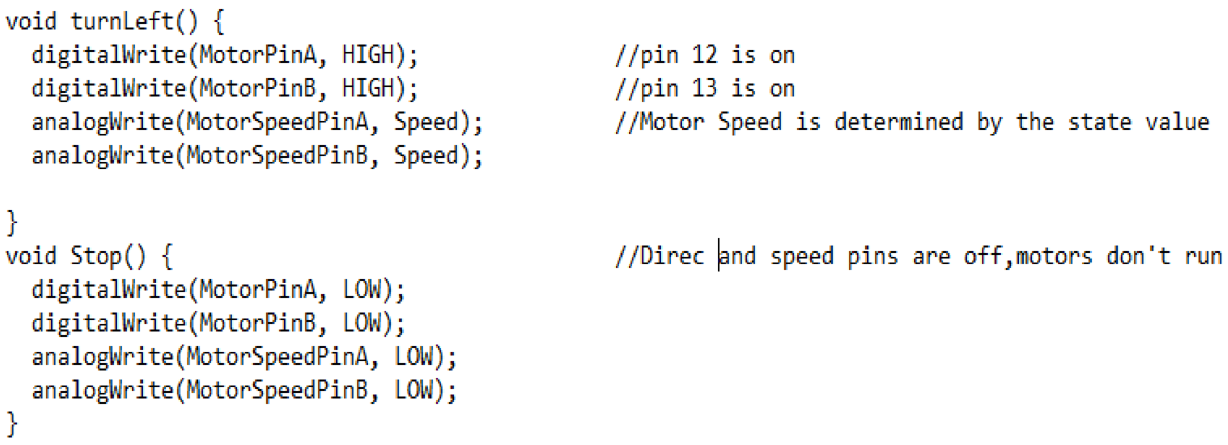
```
    if (state >= 201 && state <= 229) {
      realservo = map(state, 201, 229, 0, 180);  //servo position is mapped between 0 and 180
      myservo1.write(realservo);                 //mapped value is written into the servo1 pin
      Serial.println("S1");                      //prints S1 in the serial monitor
    }
    else if (state >= 230 && state <= 255) {
      realservo = map(state, 230, 255, 0, 180); //servo position is mapped between 0 and 180
      myservo2.write(realservo);                //mapped value is written into the servo2 pin
      Serial.println("S2");                     //prints S2 in the serial monitor
    }
  }
}
```



```
void forward() {
  digitalWrite(MotorPinA, HIGH);              //pin 12 is on
  digitalWrite(MotorPinB, LOW);               //pin 13 is off and the DC motors go forward
  analogWrite(MotorSpeedPinA, Speed);         //Motor Speed is determined by the state value
  analogWrite(MotorSpeedPinB, Speed);
}
void backward() {
  digitalWrite(MotorPinA, LOW);               //pin 12 is off
  digitalWrite(MotorPinB, HIGH);              //pin 13 is on and the DC motors go backward
  analogWrite(MotorSpeedPinA, Speed);         //Motor Speed is determined by the state value
  analogWrite(MotorSpeedPinB, Speed);
```



```
void turnRight() {
  digitalWrite(MotorPinA, LOW);                //pin 12 is off
  digitalWrite(MotorPinB, LOW);                //pin 13 is off
  analogWrite(MotorSpeedPinA, Speed);          //Motor Speed is determined by the state value
  analogWrite(MotorSpeedPinB, Speed);
```



```
void turnLeft() {
  digitalWrite(MotorPinA, HIGH);              //pin 12 is on
  digitalWrite(MotorPinB, HIGH);              //pin 13 is on
  analogWrite(MotorSpeedPinA, Speed);         //Motor Speed is determined by the state value
  analogWrite(MotorSpeedPinB, Speed);

}
void Stop() {                                 //Direc and speed pins are off,motors don't run
  digitalWrite(MotorPinA, LOW);
  digitalWrite(MotorPinB, LOW);
  analogWrite(MotorSpeedPinA, LOW);
  analogWrite(MotorSpeedPinB, LOW);
}
```

CHAPTER IV

CONCLUSION

The objective of this project was to build a low-cost remote controlled robot that would allow the study of squirrels in the wild. Dr. Charles Abramson suggested this project idea. It was an inspiration from the Mercury Remote Robot Challenge, which was created by Dr. Carl Latino, at the Oklahoma State University in 2010. Dr. Abramson believed that this idea could be useful in the creation of research and educational tools.

The robot was intended to possess capabilities, such as: remote control, the ability to record and playback sounds, stream live video and have an operational robot arm that acts as a nut delivery system. In addition to this, the major objective was to design a low-cost robotic model for less than $200. The robot that was constructed, met all the desired specifications.

Bluetooth technology was chosen over the Mercury communication system, because of its lower cost and simpler complexity. A longer-range communication system could also be employed in future robots if needed. In this project, a unique yet simple and effective means of operating the robot was devised. This method required only a single-byte to convey all the information necessary to control the Squirrel Robot system (for details please refer section 3.8.3).

CHAPTER V

FUTURE WORKS

The robot could be made to have more capabilities such as: a more complex robotic arm and structure. A longer range communication system like the Mercury robot could be employed. Since this project did not have any time critical functions to perform, a single function operation system was adequate for our needs. If additional functionality is required, a more sophisticated communication system could be developed to permit multiple functions to perform simultaneously. This idea was designed to work with Android phones but it can be implemented with iOS phones as well. This project was designed to study squirrels, but this idea could be modified to be implemented for other animals, such as: aquatic animals.

REFERENCES

[1] Sachit Butail, Nicole Abaid, Simone Macri and Maurizio Porfiri, “Fish Robot Interactions: Robot Fish in Animal Behavioral Studies,”

[2] R. Piyare, M. Tazil, “Bluetooth based home automation system using cell phone control,” article in IEEE Transactions on Consumer Electronics, June 2011.

[3] Anisha Cotta, Naik Trupti Devidas, Varda Kalidas Naik Ekoskar, “Wireless Communication Using HC – 05 Bluetooth Module Interfaced with Arduino,” IJSETR, Vol. 5, Issue 4, April 2016.

[4] “L298 Dual Full-Bridge Driver/datasheet,” January 2000.

[5] “Arduino Uno datasheet,” Farnell.

[6] Nelson Rai, Deepak Rasaily, Tashi Rapden Wangchuk, Manoj Gurung, Rit Kr. Khawas, “Bluetooth Remote Controlled Car using Arduino,” IJETT, Vol. 33, No. 8, March 2016.

[7] “Ardumoto Shield Hookup Guide,”Sparkfun

[8] Vibhor Gupta, “Working Analysis of the H-Bridge Motor Driver Circuit Designed for Wheeled Mobile Robots,” IEEE, 2010.

[9] Khan Masoom Raza, Mohd. Kamil, Pushpendra Kumar, “Speed Control of DC Motor by using PWM,” IJARCCE, Vol. 5, Issue 4, April 2016.

[10] Pratik J Patel, Hardeep J Patel, Apeksha D Unadkat, Chintan U Patel, Sanjay Bhanderi, “PWM Based Speed Control for a DC Motor,” Vol.6, Issue 4, April 2017, ISSN: 2278 – 7798.

[11] Tolga Ozer, Sinan, Yuksel Oguz, “H-Bridge DC Motor Driver Design and Implementation with using dsPIC30f4011,” IJIRSET, Vol.6, Special Issue 10, May 2017.

[12] “Components101 HC-05 Bluetooth Module”, HC-05 module

[13] Elisa Donati, Martin Worm, Stefano Mintchev, Marleen vander Wiel, Giovanni Benelli, Gerhard von der Emde and Cesare Stefanini, “Investigation of Collective Behavior and Electrocommunication in the Weakly Electric Fish, Mormyrus rume, through a biomimetic Robotic Dummy Fish,” Bioinspiration & Biomimetics, 30 November 2016.

[14] Alexey Gribovskiy, Jose Halloy, Jean-Louis Deneubourg, Hannes Bleuler and Francesco Mondada, “Towards Mixed Societies of Chickens and Robots,” IEEE/RSJ International Conference on Intelligent Robots and Systems, October 2010.

[15] Jolyon j. Faria, John R. G. Dyer, Romain O. Clement, Iain D. Couzin, Natalie Holt, Ashley J. W. Ward, Dean Waters, Jens Krause, “A novel method for investigating the Collective Behavior of fish: introducing ‘Robofish’,” June 2010.

[16] Vijayalakshmi S, Archana M, “Robotic Car Arduino with Bluetooth Controller,” IJISE, 2019.

[17] Ayan Maity, Avijit Paul, Priyanka Goswami, Ankan Bhattacharya, “Android Application Based Bluetooth Controlled Robotic Car,” IJIIS, 2017.

[18] Charles I. Abramson, “Let Us Bring Comparitive Psychology Back,” International Journal of Comapritive Psychology, Vol. 31, October 2015.

[19] Charles I. Abramson, “A Crisis in Comparitive Psychology: Where have all the Undergraduates gone?,” Frontiers in Psychology, opinion Article, October 2015.

[20] ISD1820, datasheet

[21] “ISD1820 Vice Recording and Playback Module,” EIModule

[22] “ISD1820 – Voice Recorder,” Rajguru Electronics

[23] “ISD1820 Voice Recorder Module User Guide”, Rev 1.0, Oct 2012.

[24] Future Electronics, “Servo Motors Control & Arduino,” Future Electronics Egypt Ltd. (Arduino Egypt)

[25] “HS – 322 HD Servo,” Electricwingman.com

[26] “9g Micro servo,” Robotshop.com

[27] “Introduction to Arduino and Physical Computing,” Arduino to GO.

[28] “Arduino-projecthub-Arduino-PWM-tutorial”

[29] “robotrus/ISD1820”

[30] “Hackstore EImodule/ISD1820”

[31] “Ammaimhobbies.com”

[32] “Perintang.com/servomotor”

[33] “ArduCAM-M-2MP Camera Shield-2MP SPI Camera User Guide”

[34] “ArduCAM ESP8266 Uno board User guide”

[35] “UCTRONICS/ArduCAM 2MP Mini Camera”

[36] “Arduino as ISP and Arduino Bootloaders – Arduino Memory Map”

[37] “Arduino Uno-Google image”

[38] “Ardumoto shield,” For projects

[39] “HC-05 Bluetooth Module,” Finger Point Engg

[40] “Voice Recorder Module,” Allelectronics.com

APPENDICES

A. Getting started with MIT App Inventor:

The following are the instructions to be followed to install and implement the MIT App Inventor, found on Google.

For this project, an Android Application was developed. However, there is another app called Thunkable, which is compatible with both Android and ios operating systems. It uses the software similar to the App Inventor

1. Install the App Inventor Setup Software Package

(Uninstall any previous App Inventor Software that were installed in the computer)

   i. Download the installer.
   ii. Locate the file MIT_Appinventor_Tools_2.3.0 (80MB approximately) in your Downloads file.
   iii. Open the file.
   iv. Click through the steps of the installer.
   v. You may be asked if you want to allow a program from an unknown publisher to make changes to the Computer, **Click yes.**

2. Locating the setup software.
   i. In most cases, the App Inventor should be able to locate the Setup Software on its own. But if it asks for the location of the software, the path to enter is **C:\Program Files\Appinventor\commands_for_Appinventor.** If you are using a 64 – bit machine, you should search for Program Files(x86) rather than Program Files.
   ii. If you did not install the software as an administrator, if it was installed in your local directory rather than in **C:\Program Files,** you will need to search for the file to find the correct name.

Note: If there is something wrong, visit the **Troubleshooting Page** or **Check the App Inventor User Forum for help.**

3. Continue with setup – choose the connection instruction you were following:
   - Select Setup Emulator

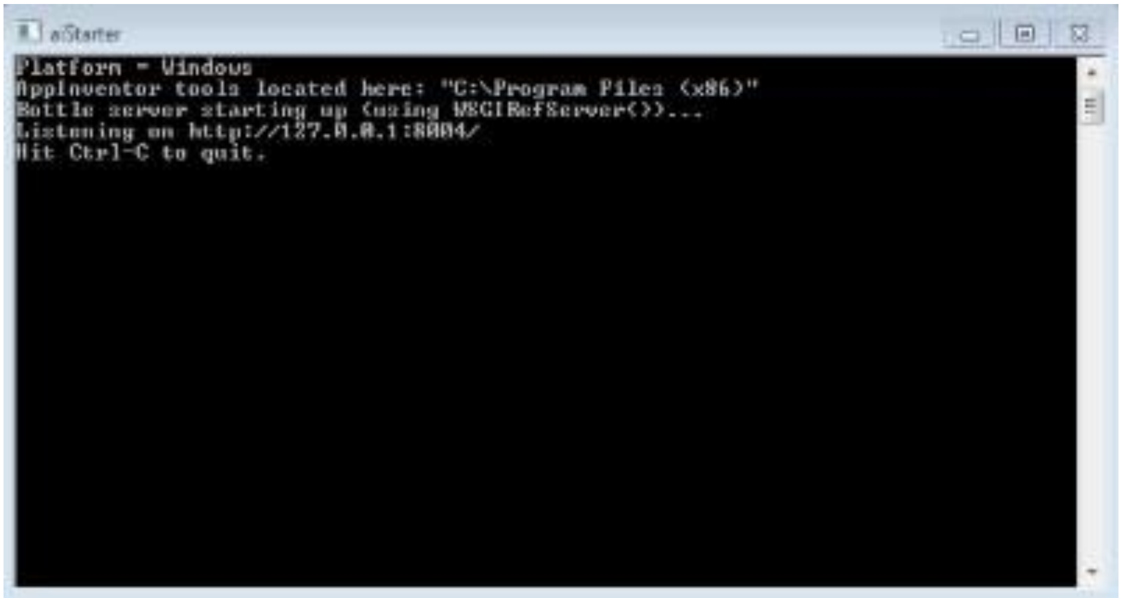
Setting up the Emulator

1. Install the App Inventor Setup Software for the respective Operating system (Windows, Mac, GNU/Linux)
2. Launch aistarter (Windows and GNU/Linux)

Note: On a Mac, aistarter will start automatically when you log in to your account and it will run invisibly in the background.

a. On Windows, there will be a shortcut to aistarter from your Desktop, from the Start Menu, from All programs and from Startup Folder. If you want to use the Emulator with the App Inventor, you will need to automatically launch the aistarter on your Computer when you log in. You can start the aistarter by clicking the icon on your Desktop or using the entry in your Start Menu.

b. To launch aistarter on Windows, double click on the icon shown above. You will know that you have successfully launched aistarter when you see a window like the following:



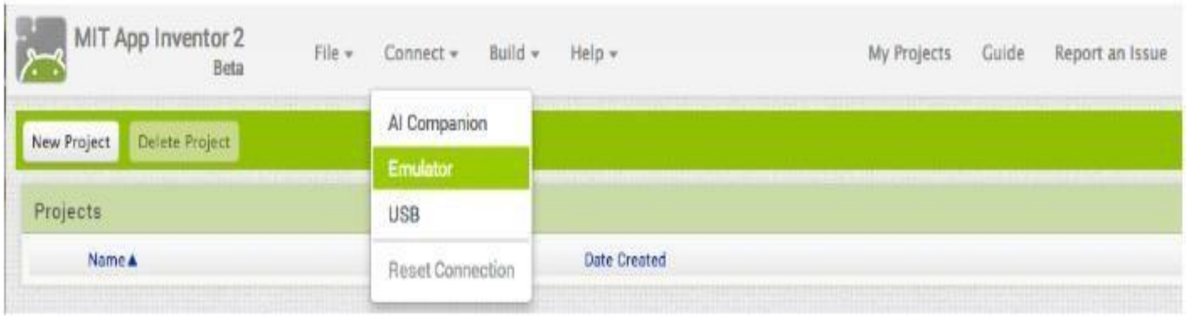
```
Platform = Windows
AppInventor tools located here: "C:\Program Files (x86)"
Bottle server starting up (using WSGIRefServer())...
Listening on http://127.0.0.1:8004/
Hit Ctrl-C to quit.
```

On GNU/Linux, aistarter will be in the folder:

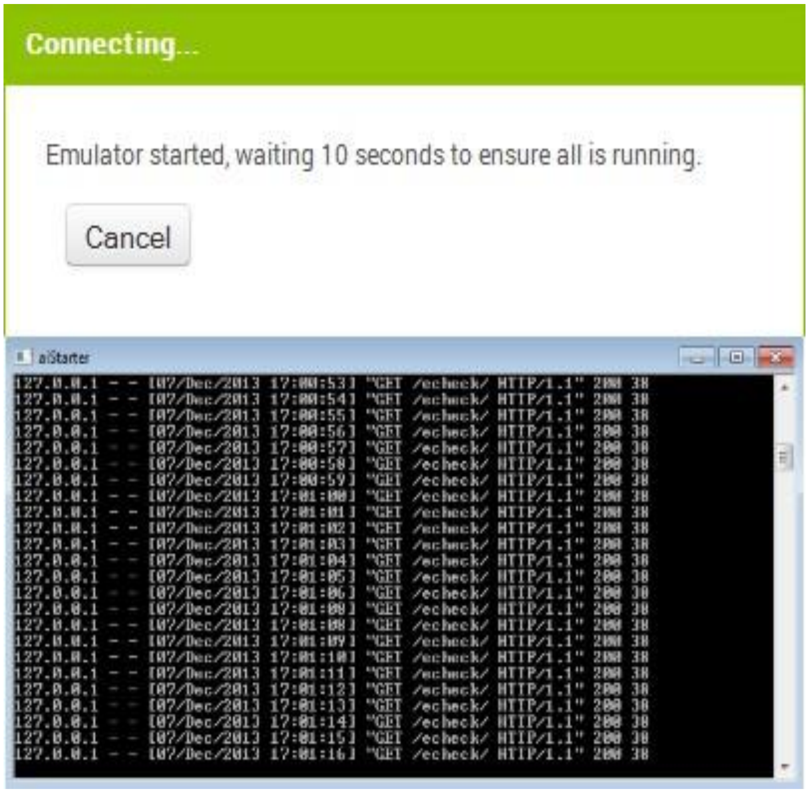
\usr\google\appinventor\commands_for_Appinventor and you will need to launch it manually. You can launch it from the command line with

\usr\google\appinventor\commands_for_appinventor\aistarter&

c. Open an App Inventor Project and connect it to the Emulator
   a) First, go to the App Inventor and open a project or create a new one by clicking on Project in the Menu then select the option New Project and give it a name.
   b) Then, form App Inventor's menu, on the App Inventor cloud – based software at ai2.appinventor.mit.edu, go to the Connect menu and click the Emulator option



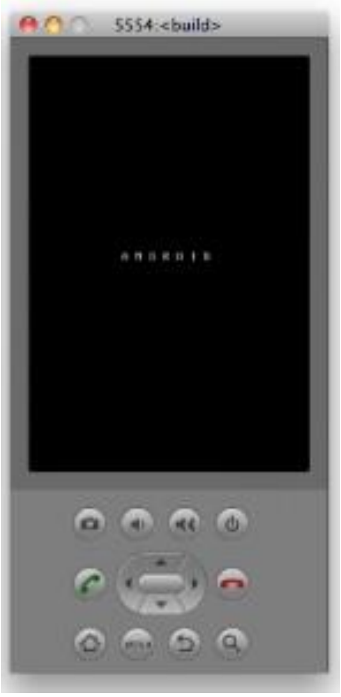
c) You’ll get a notice saying the Emulator is connecting. Starting the Emulator can take a couple of minutes. You may see update screens like the following as the Emulator starts up:


Connecting...

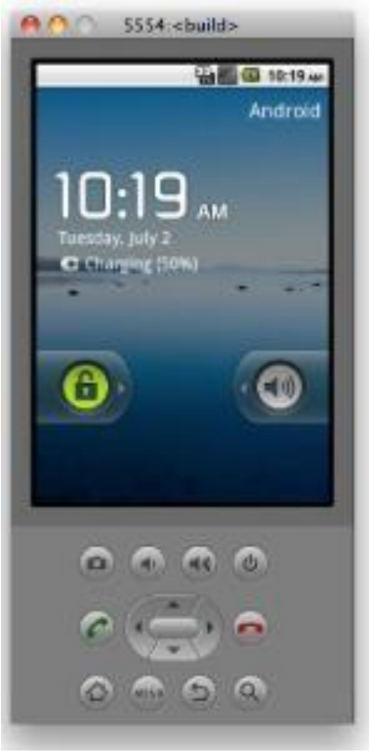
Emulator started, waiting 10 seconds to ensure all is running.

```
127.0.0.1 - - [07/Dec/2013 17:00:53] "GET /echeck/ HTTP/1.1" 200 38
127.0.0.1 - - [07/Dec/2013 17:00:54] "GET /echeck/ HTTP/1.1" 200 38
127.0.0.1 - - [07/Dec/2013 17:00:55] "GET /echeck/ HTTP/1.1" 200 38
127.0.0.1 - - [07/Dec/2013 17:00:56] "GET /echeck/ HTTP/1.1" 200 38
127.0.0.1 - - [07/Dec/2013 17:00:57] "GET /echeck/ HTTP/1.1" 200 38
127.0.0.1 - - [07/Dec/2013 17:00:58] "GET /echeck/ HTTP/1.1" 200 38
127.0.0.1 - - [07/Dec/2013 17:00:59] "GET /echeck/ HTTP/1.1" 200 38
127.0.0.1 - - [07/Dec/2013 17:01:00] "GET /echeck/ HTTP/1.1" 200 38
127.0.0.1 - - [07/Dec/2013 17:01:01] "GET /echeck/ HTTP/1.1" 200 38
127.0.0.1 - - [07/Dec/2013 17:01:02] "GET /echeck/ HTTP/1.1" 200 38
127.0.0.1 - - [07/Dec/2013 17:01:03] "GET /echeck/ HTTP/1.1" 200 38
127.0.0.1 - - [07/Dec/2013 17:01:04] "GET /echeck/ HTTP/1.1" 200 38
127.0.0.1 - - [07/Dec/2013 17:01:05] "GET /echeck/ HTTP/1.1" 200 38
127.0.0.1 - - [07/Dec/2013 17:01:06] "GET /echeck/ HTTP/1.1" 200 38
127.0.0.1 - - [07/Dec/2013 17:01:08] "GET /echeck/ HTTP/1.1" 200 38
127.0.0.1 - - [07/Dec/2013 17:01:08] "GET /echeck/ HTTP/1.1" 200 38
127.0.0.1 - - [07/Dec/2013 17:01:09] "GET /echeck/ HTTP/1.1" 200 38
127.0.0.1 - - [07/Dec/2013 17:01:10] "GET /echeck/ HTTP/1.1" 200 38
127.0.0.1 - - [07/Dec/2013 17:01:11] "GET /echeck/ HTTP/1.1" 200 38
127.0.0.1 - - [07/Dec/2013 17:01:12] "GET /echeck/ HTTP/1.1" 200 38
127.0.0.1 - - [07/Dec/2013 17:01:13] "GET /echeck/ HTTP/1.1" 200 38
127.0.0.1 - - [07/Dec/2013 17:01:14] "GET /echeck/ HTTP/1.1" 200 38
127.0.0.1 - - [07/Dec/2013 17:01:15] "GET /echeck/ HTTP/1.1" 200 38
127.0.0.1 - - [07/Dec/2013 17:01:16] "GET /echeck/ HTTP/1.1" 200 38
```

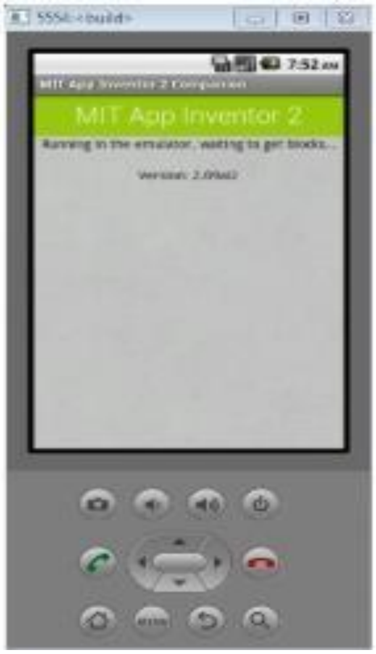
d) The Emulator will initially appear with an empty black screen.



e) Wait until the Emulator is ready with a colored screen background.

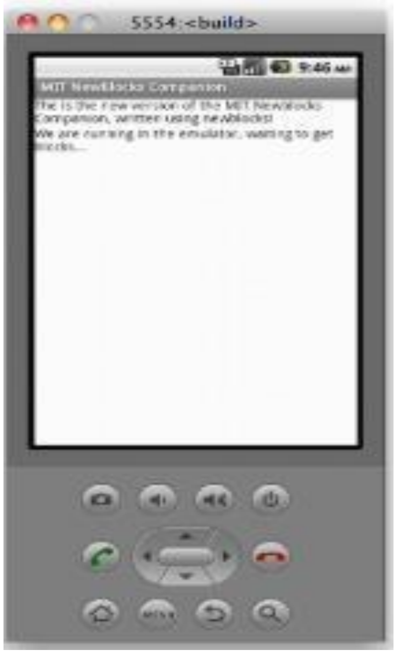


f) Even after the background appears, you should wait until the emulated phone has finished preparing its SD card: there will be a notice at the top of the screen while the card is being prepared. When connected, the Emulator will launch and show the app you have open in the App Inventor. Setup is now complete.



Running in the emulator, waiting to get blocks...

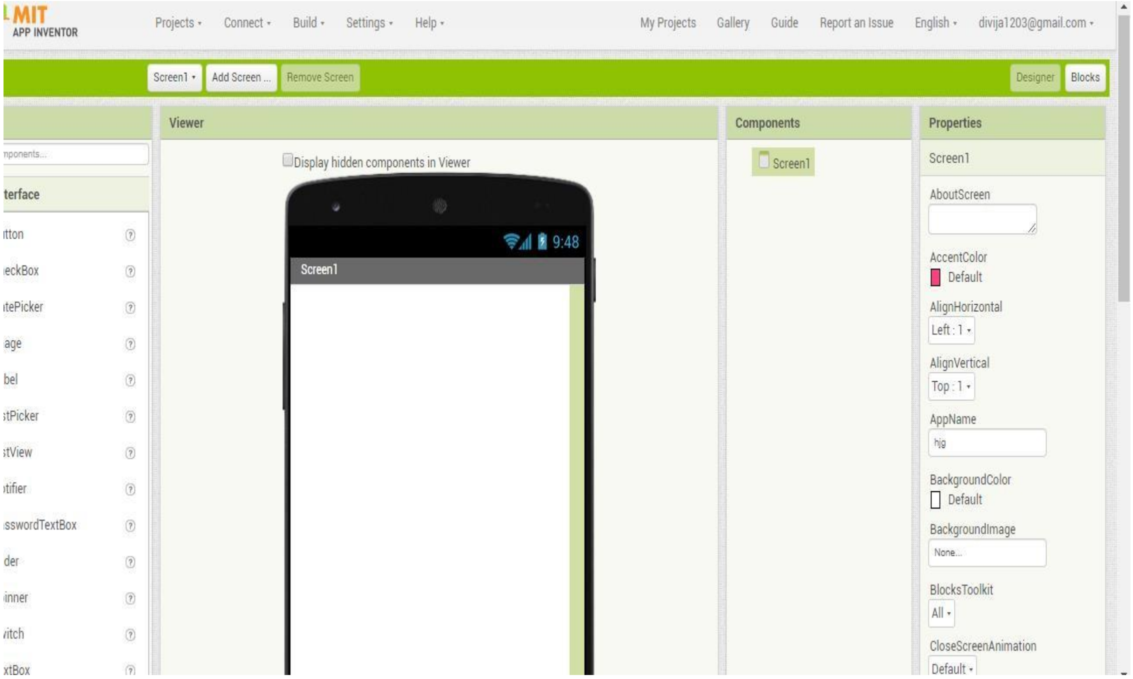
Version: 2.09ai2



The is the new version of the MIT NewBlocks Companion, written using newblocks!
We are running in the emulator, waiting to get blocks...

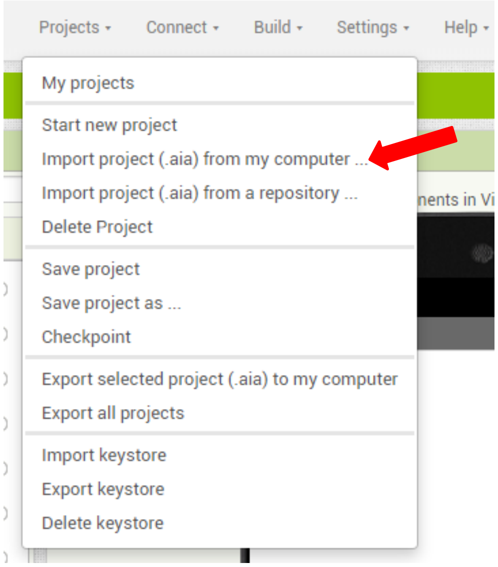
Using the App Inventor:

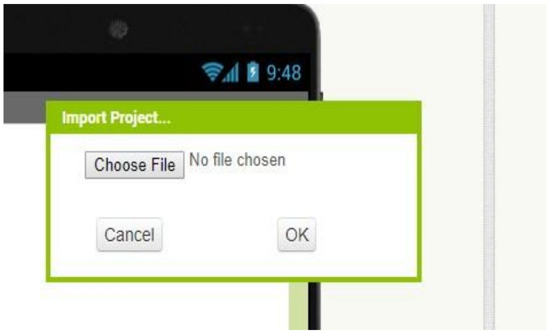
1. The new project screen looks like the following:



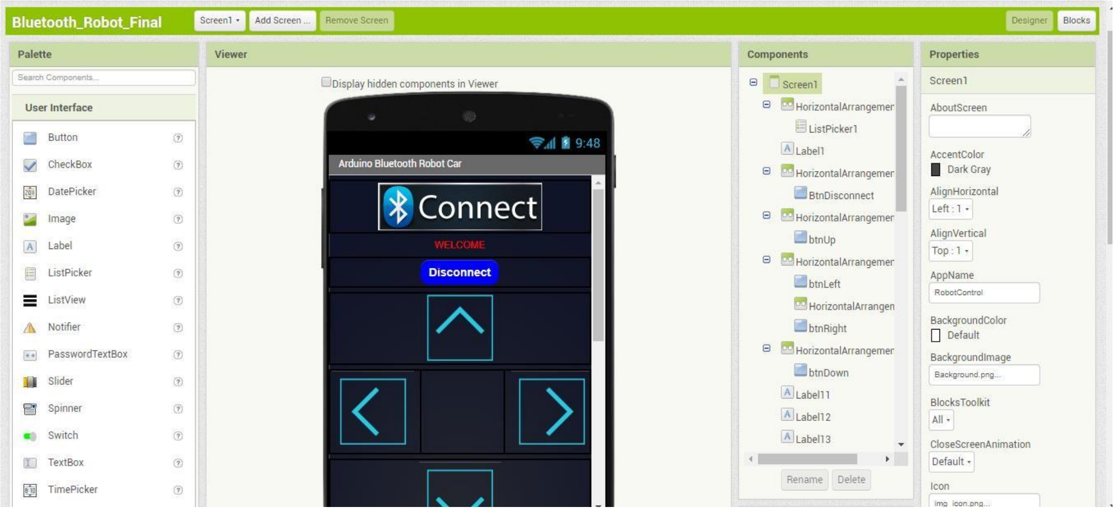
Screen1
AboutScreen
AccentColor Default
AlignHorizontal Left : 1 ▾
AlignVertical Top : 1 ▾
AppName hjg
BackgroundColor Default
BackgroundImage None...
BlocksToolkit All ▾
CloseScreenAnimation Default ▾

2. Download the .aia file with the controls related to this project.
3. Import the download aia file from the Computer.





4. Then a screen with the Bluetooth Robot App appears on the screen which looks like the following:



AboutScreen

AccentColor
Dark Gray

AlignHorizontal
Left : 1

AlignVertical
Top : 1

AppName
RobotControl

BackgroundColor
□ Default

BackgroundImage
Background.png...

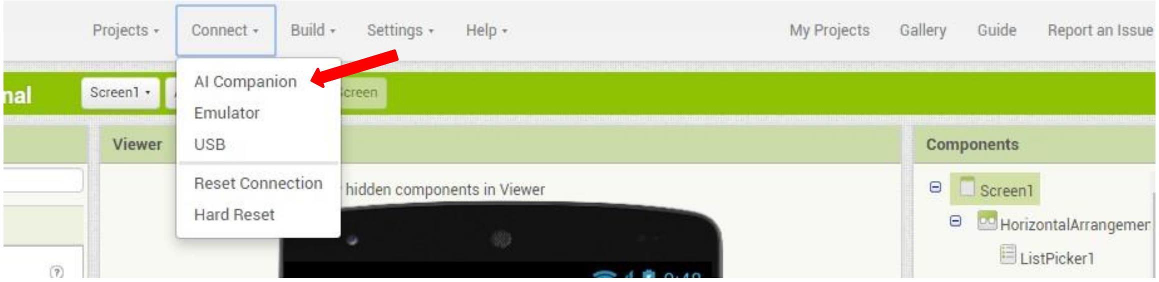
BlocksToolkit
All

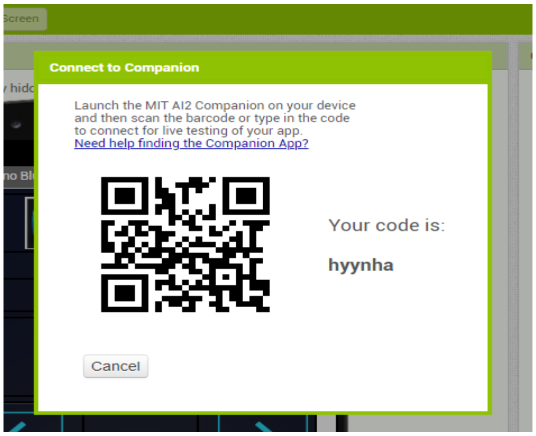
CloseScreenAnimation
Default

Icon
img_icon.png...

Connect the MIT App Inventor on the Computer with a Smart Phone:

1. Download the MIT AI2 Companion app from Google Play Store on your Android phone.
2. Click the Connect option on the App Inventor menu on the Computer screen and select AI Companion option. It displays a QR code.





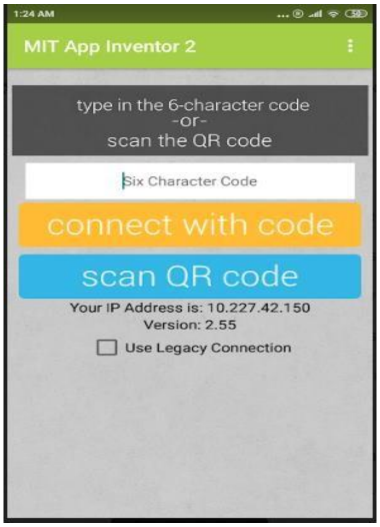
Connect to Companion

Launch the MIT AI2 Companion on your device and then scan the barcode or type in the code to connect for live testing of your app.
Need help finding the Companion App?

Your code is:

hyynha

3. Open the downloaded MIT AI2 Companion App on the Android phone. It opens as follows:

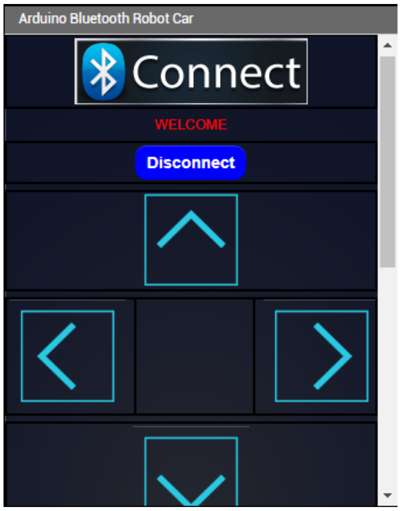


type in the 6-character code
-or-
scan the QR code

Your IP Address is: 10.227.42.150
Version: 2.55

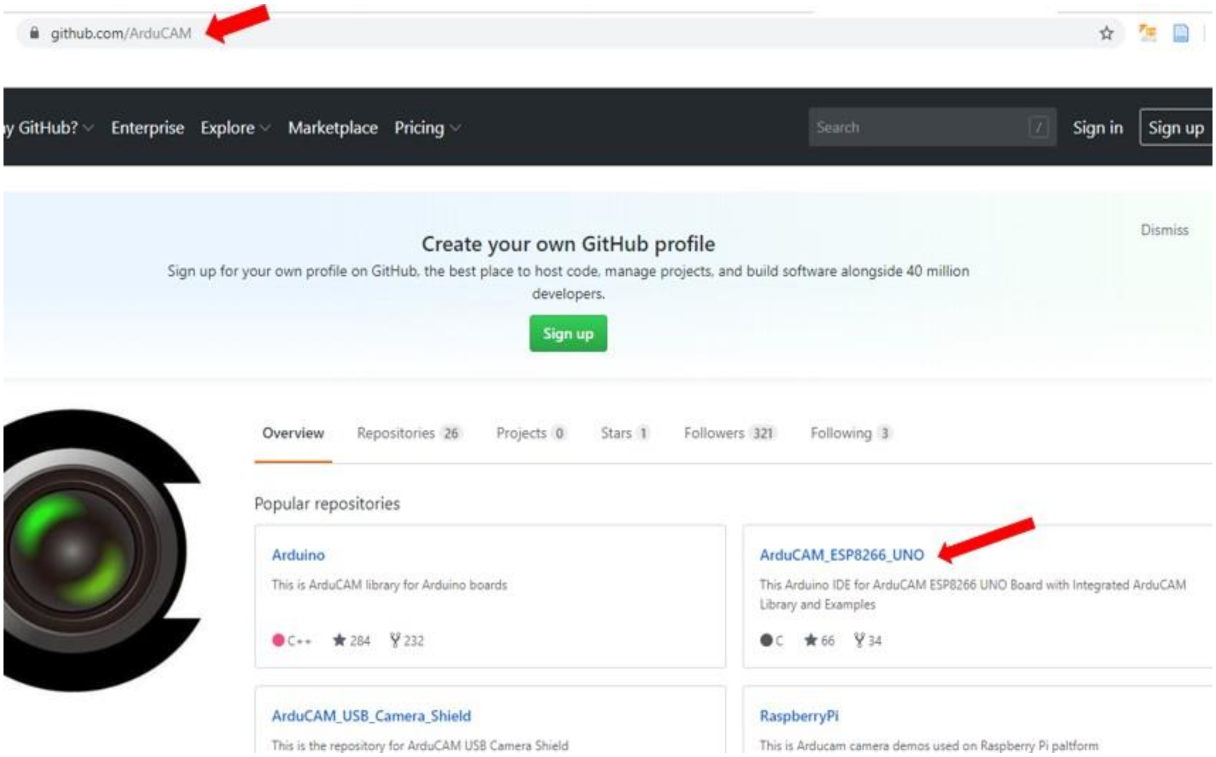
□ Use Legacy Connection

Enter or scan the QR code on the computer to the Smart phone. It automatically displays the controls from the computer on the phone and it looks as follows:



B. Configuration of ArduCAM ESP8266 Uno Board and 2MP Mini Camera

1. Go to https://github.com/ArduCAM.
2. Select the ArduCAM_ESP8266_UNO



Create your own GitHub profile

Sign up for your own profile on GitHub, the best place to host code, manage projects, and build software alongside 40 million developers.

Popular repositories

Arduino

This is ArduCAM library for Arduino boards

ArduCAM_ESP8266_UNO

This Arduino IDE for ArduCAM ESP8266 UNO Board with Integrated ArduCAM Library and Examples

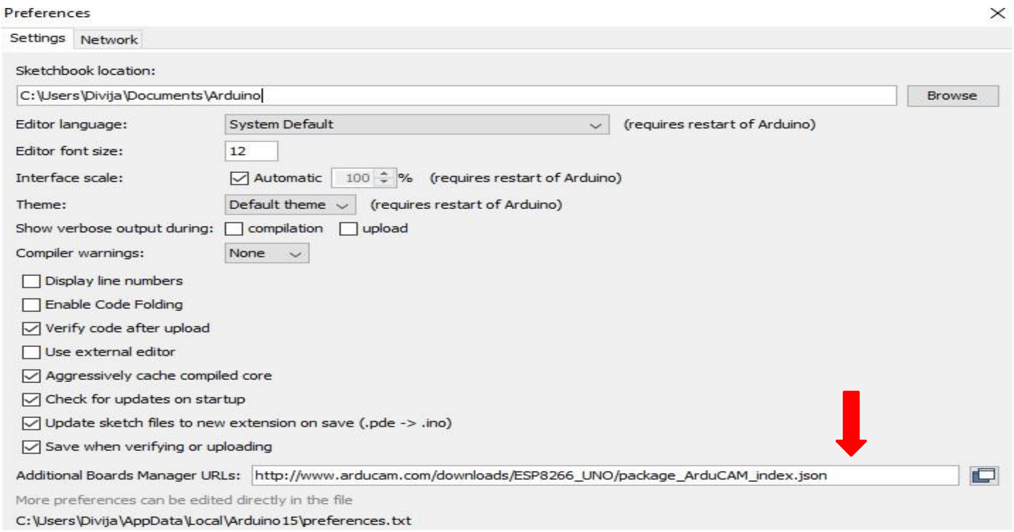
ArduCAM_USB_Camera_Shield

This is the repository for ArduCAM USB Camera Shield

RaspberryPi

This is Arducam camera demos used on Raspberry Pi paltform

3. **http://www.arducam.com/downloads/ESP8266_UNO/package_ArduCAM_index.json** - Scroll down and copy this link.
4. Open Arduino IDE
5. Go to File – Preferences – Copy the above link in the Additional Boards Manager URLs.



Sketchbook location: C:\Users\Divija\Documents\Arduino

Editor language: System Default (requires restart of Arduino)

Editor font size: 12

Interface scale: ☑ Automatic 100 % (requires restart of Arduino)

Theme: Default theme (requires restart of Arduino)

Show verbose output during: □ compilation □ upload

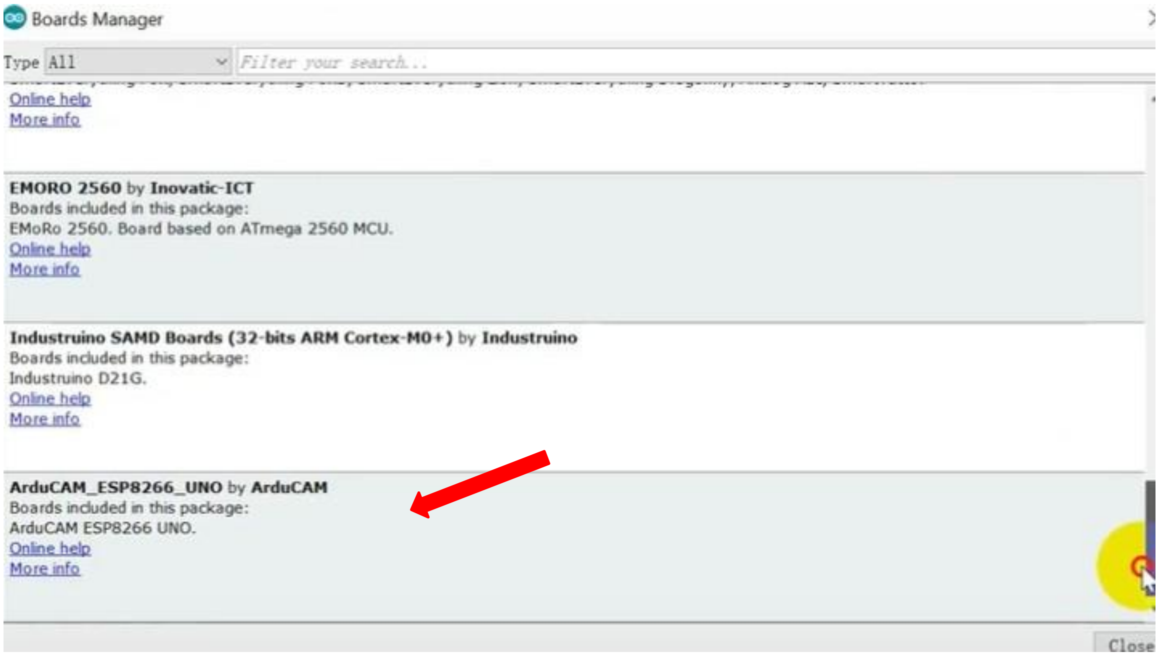
Compiler warnings: None

□ Display line numbers
□ Enable Code Folding
☑ Verify code after upload
□ Use external editor
☑ Aggressively cache compiled core
☑ Check for updates on startup
☑ Update sketch files to new extension on save (.pde -> .ino)
☑ Save when verifying or uploading

Additional Boards Manager URLs: http://www.arducam.com/downloads/ESP8266_UNO/package_ArduCAM_index.json

More preferences can be edited directly in the file
C:\Users\Divija\AppData\Local\Arduino15\preferences.txt

6. Open Tools – Board – Board Manager – Install ArduCAM_ESP8266_UNO by ArduCAM Package.



Boards Manager

Type All Filter your search...

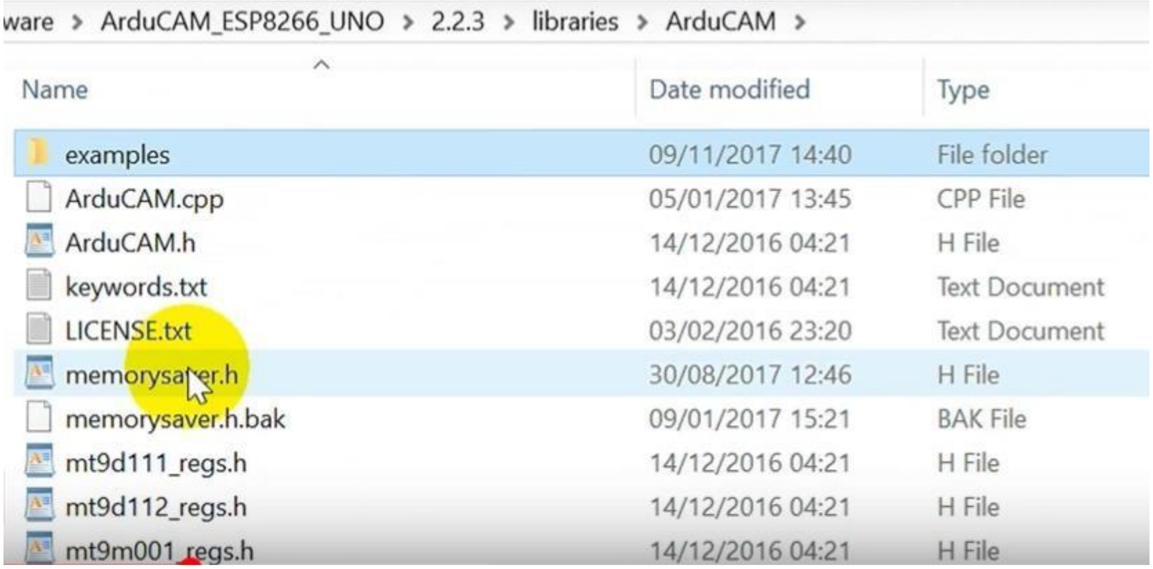
Online help
More info

EMORO 2560 by **Inovatic-ICT**
Boards included in this package:
EMoRo 2560. Board based on ATmega 2560 MCU.
Online help
More info

Industruino SAMD Boards (32-bits ARM Cortex-M0+) by **Industruino**
Boards included in this package:
Industruino D21G.
Online help
More info

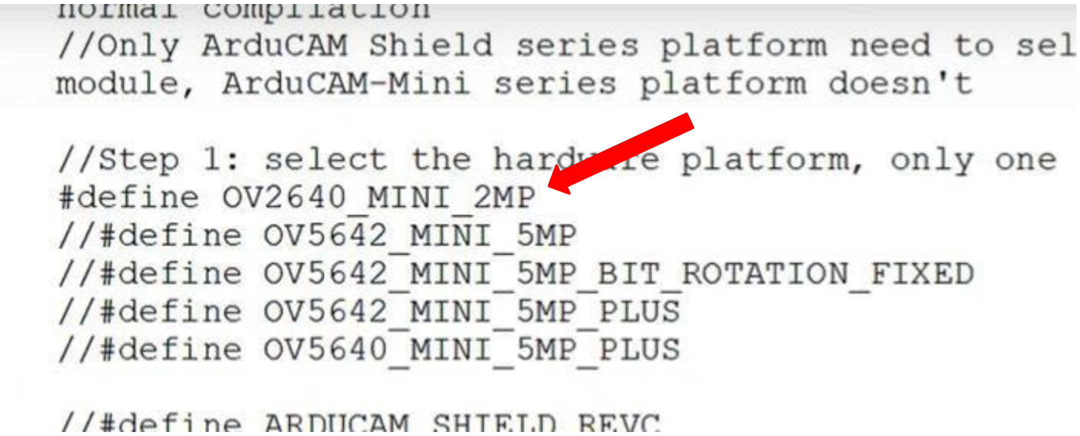
ArduCAM_ESP8266_UNO by **ArduCAM**
Boards included in this package:
ArduCAM ESP8266 UNO.
Online help
More info

7. Configure the 2MP Camera.
   a) The ***memorysaver.h*** file enable the 2MP camera and disable the others.
      i. The ***memorysaver.h*** can be located at C:\Users\Your Computer Name\AppData\Local\Arduino15\Packages\ArduCAM_ESP8266_UNO\ hardware\ArduCAM_ESP8266_UNO\2.2.4\libraries\ArduCAM.



ware > ArduCAM_ESP8266_UNO > 2.2.3 > libraries > ArduCAM >

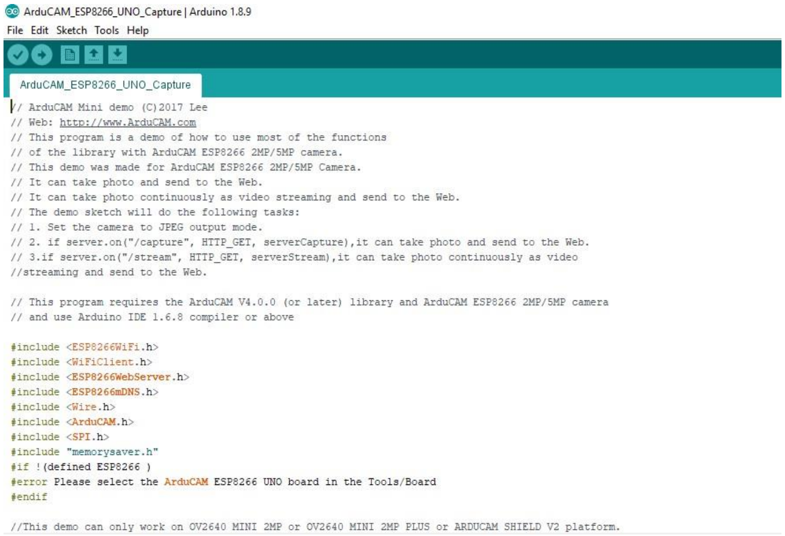
Name	Date modified	Type
examples	09/11/2017 14:40	File folder
ArduCAM.cpp	05/01/2017 13:45	CPP File
ArduCAM.h	14/12/2016 04:21	H File
keywords.txt	14/12/2016 04:21	Text Document
LICENSE.txt	03/02/2016 23:20	Text Document
memorysaver.h	30/08/2017 12:46	H File
memorysaver.h.bak	09/01/2017 15:21	BAK File
mt9d111_regs.h	14/12/2016 04:21	H File
mt9d112_regs.h	14/12/2016 04:21	H File
mt9m001_regs.h	14/12/2016 04:21	H File



```
normal compilation
//Only ArduCAM Shield series platform need to sel
module, ArduCAM-Mini series platform doesn't

//Step 1: select the hardware platform, only one
#define OV2640_MINI_2MP
//#define OV5642_MINI_5MP
//#define OV5642_MINI_5MP_BIT_ROTATION_FIXED
//#define OV5642_MINI_5MP_PLUS
//#define OV5640_MINI_5MP_PLUS

//#define ARDUCAM_SHIELD_REVC
```

8. Using Arduino IDE
   a) After installation, open Tools – Board – Select ArduCAM ESP8266
   b) Ports – Select the corresponding port to which the ArduCAM ESP8266 board is connected.
   c) Open File – Examples – ArduCAM – ESP8266 – Select ArduCAM_ESP8266_UNO_Capture.

9. Compile and Upload then code to the ESP8266 UNO board connected to the Computer.



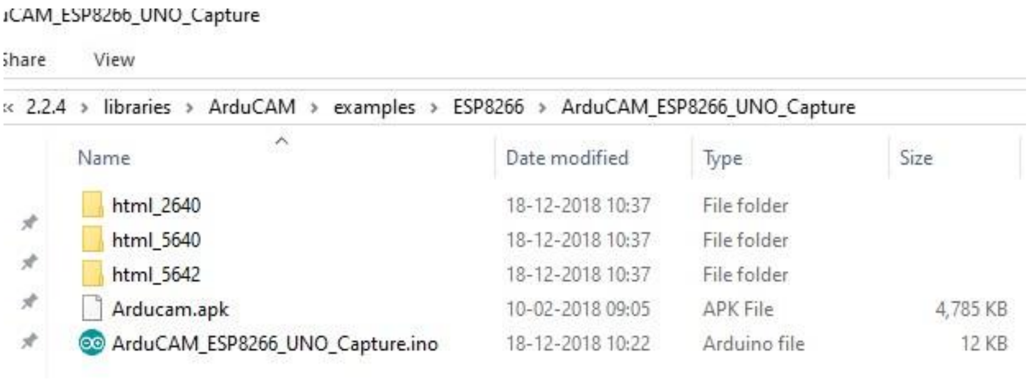
ArduCAM_ESP8266_UNO_Capture | Arduino 1.8.9
File Edit Sketch Tools Help
ArduCAM_ESP8266_UNO_Capture

```
// ArduCAM Mini demo (C)2017 Lee
// Web: http://www.ArduCAM.com
// This program is a demo of how to use most of the functions
// of the library with ArduCAM ESP8266 2MP/5MP camera.
// This demo was made for ArduCAM ESP8266 2MP/5MP Camera.
// It can take photo and send to the Web.
// It can take photo continuously as video streaming and send to the Web.
// The demo sketch will do the following tasks:
// 1. Set the camera to JPEG output mode.
// 2. if server.on("/capture", HTTP_GET, serverCapture),it can take photo and send to the Web.
// 3.if server.on("/stream", HTTP_GET, serverStream),it can take photo continuously as video
//streaming and send to the Web.

// This program requires the ArduCAM V4.0.0 (or later) library and ArduCAM ESP8266 2MP/5MP camera
// and use Arduino IDE 1.6.8 compiler or above

#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
#include <ESP8266mDNS.h>
#include <Wire.h>
#include <ArduCAM.h>
#include <SPI.h>
#include "memorysaver.h"
#if !(defined ESP8266 )
#error Please select the ArduCAM ESP8266 UNO board in the Tools/Board
#endif

//This demo can only work on OV2640 MINI 2MP or OV2640 MINI 2MP PLUS or ARDUCAM SHIELD V2 platform.
```

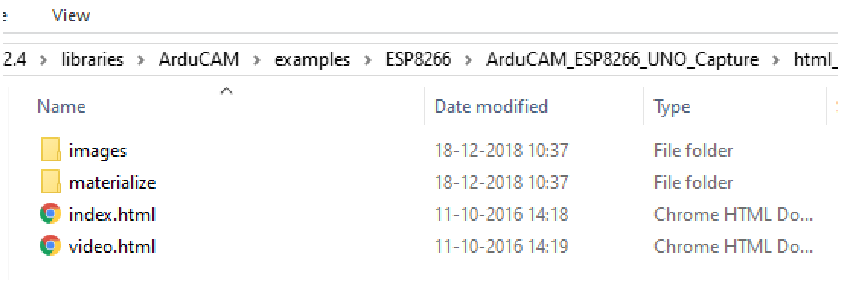
10. After the code is uploaded and the device is turned on, then open the Wi-Fi settings on a laptop or a computer to find the ESP8266 connection available. Connect to the network.
11. Finally, open index.html or video.html, input the IP address from the serial monitor then take pictures or videos respectively.

a) Once the ArduCAM libraries are installed and the Arduino board and tools are selected to be ArduCAM ESP8266 Uno board, then go to Sketch – Sketch Folder and select html_2640.

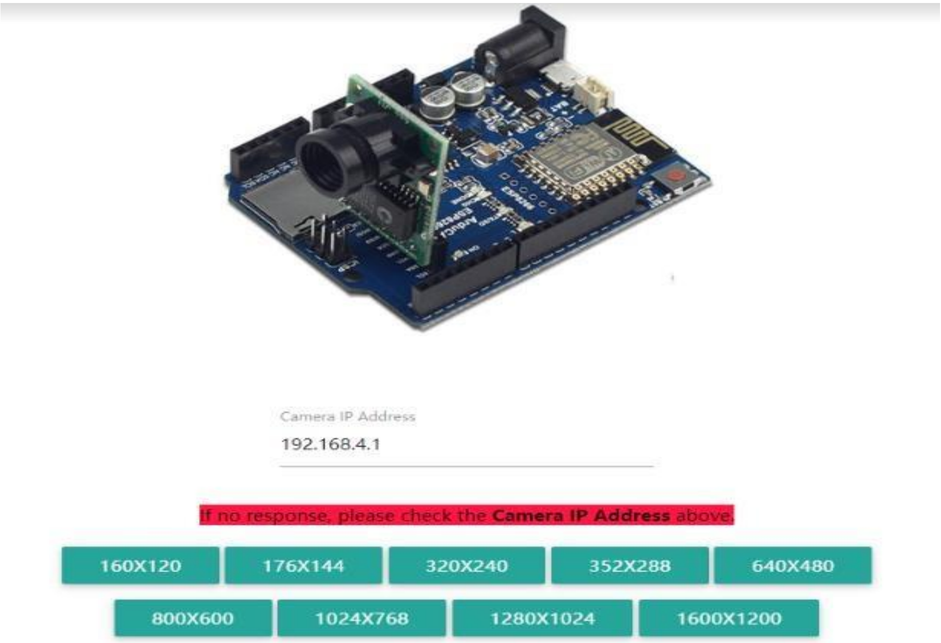
b) Select index.html to capture a picture or video.html to record a video using the ArduCAM ESP8266 2MP Mini camera.


ıCAM_ESP8266_UNO_Capture
Share
View
2.2.4 › libraries › ArduCAM › examples › ESP8266 › ArduCAM_ESP8266_UNO_Capture

Name	Date modified	Type	Size
html_2640	18-12-2018 10:37	File folder	
html_5640	18-12-2018 10:37	File folder	
html_5642	18-12-2018 10:37	File folder	
Arducam.apk	10-02-2018 09:05	APK File	4,785 KB
ArduCAM_ESP8266_UNO_Capture.ino	18-12-2018 10:22	Arduino file	12 KB


View
2.4 › libraries › ArduCAM › examples › ESP8266 › ArduCAM_ESP8266_UNO_Capture › html_

Name	Date modified	Type
images	18-12-2018 10:37	File folder
materialize	18-12-2018 10:37	File folder
index.html	11-10-2016 14:18	Chrome HTML Do...
video.html	11-10-2016 14:19	Chrome HTML Do...



Camera IP Address

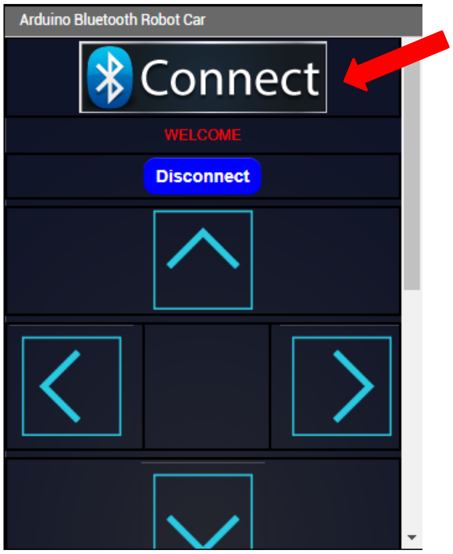
192.168.4.1

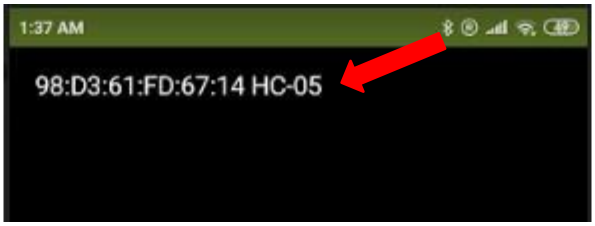
If no response, please check the **Camera IP Address** above.

c) The resolution of the image and the video can be changed according to the system requirements.

C. User Manual to control the Arduino Robot:

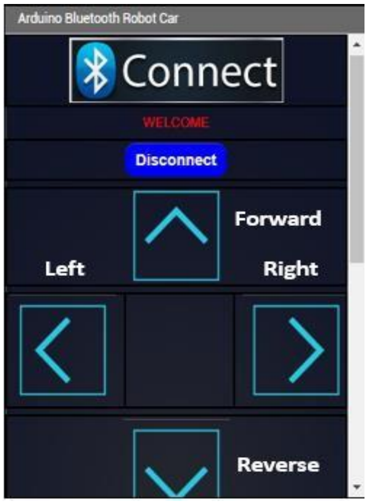
1. After switching on the Arduino Robot, turn on the Bluetooth on your phone and pair the device with the HC – 05 Bluetooth module.
2. Now, on the App, click on the Connect button and connect to the paired, HC – 05 Bluetooth device.





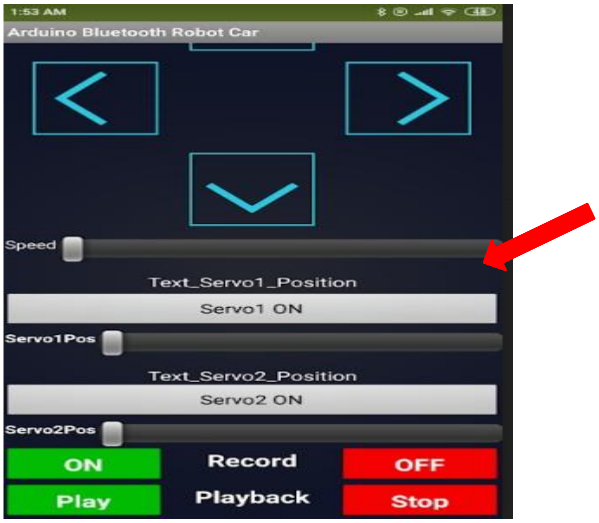
98:D3:61:FD:67:14 HC-05

3. Once the device is connected to the HC – 05 Bluetooth module, the App is ready to control the Arduino Robot.



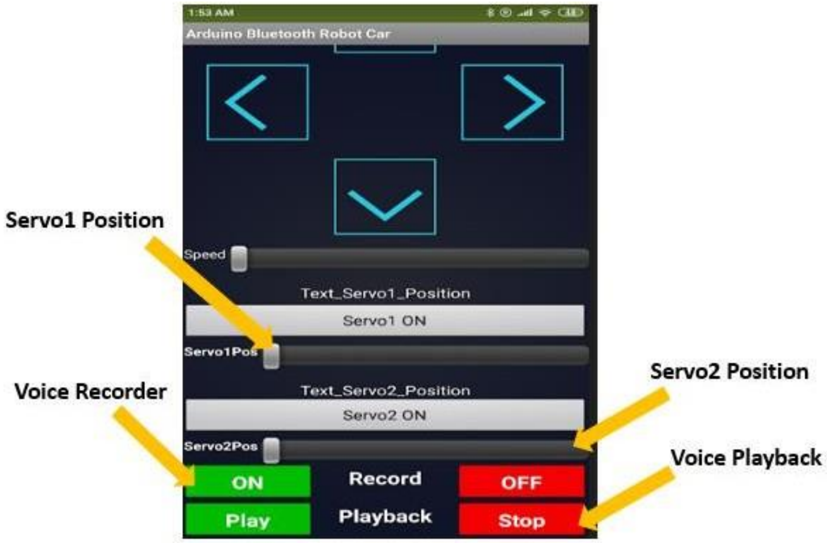
A. By clicking on the upward arrow the Robot moves forward.
B. It moves in the Reverse direction when the downward arrow button is clicked on.
C. Similarly, it moves left and right when the left and the right arrows are clicked on respectively.

There is a slider that says Speed, which when moved the Robot can be operated at different speeds.



<u>Note</u>: When the buttons are clicked, the Arduino Robot moves in the respective directions and continue to move in those directions as long as the buttons are kept pressed.

4. Scroll down to find the controls and sliders for the servos and the Voice recorder.



Servo1 Position

Voice Recorder

Servo2 Position

Voice Playback

VITA

Lakshmi Divija Sarupya Brahmandam

Candidate for the Degree of

Master of Science

Thesis: THE DEVELOPMENT A OF LOW-COST ROBOTIC SQUIRREL FOR THE STUDY OF SMALL ANIMAL BEHAVIOR

Major Field: Electrical and Computer Engineering

Biographical:

Education:

Completed Master of Science at Oklahoma State University, Stillwater, Oklahoma in December, 2019.

Completed Bachelor of Engineering in Electrical and Electronics Engineering degree at Osmania University, Hyderabad, Telangana, India in May 2017.

Experience: Research Assistant at Oklahoma State University, June 2018 – Dec. 2018.
Teaching Assistant at Oklahoma State University, Jan 2019 – Dec. 2019