

# Bluetooth-WIFI Communication Car

Author: Mehmetcan Güleşçi

Istanbul Bilgi University

*Computer Engineering*

May 27, 2017

## ABSTRACT

- Using of smartphones
- Development of smartphones in terms of low cost robots.
- About project

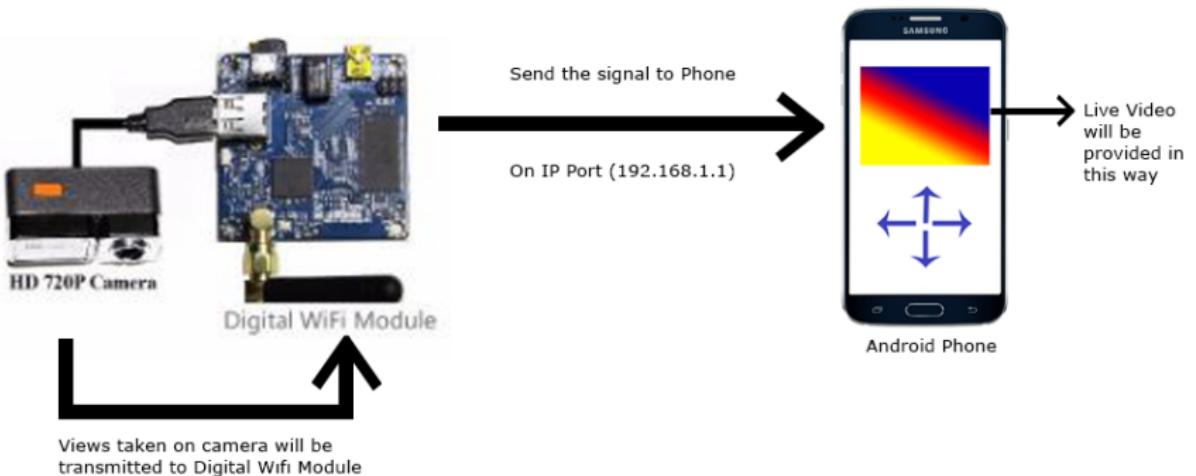
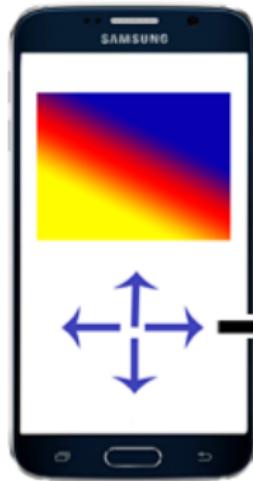


Figure: WIFI Communication



Android Phone

Thanks to signals, send  
from bluetooth



Press the any button, Motion will  
be provided



Figure: Bluetooth Communication

# INTRODUCTION

- A benefit of having a Bluetooth connection.
- Two parts of communication

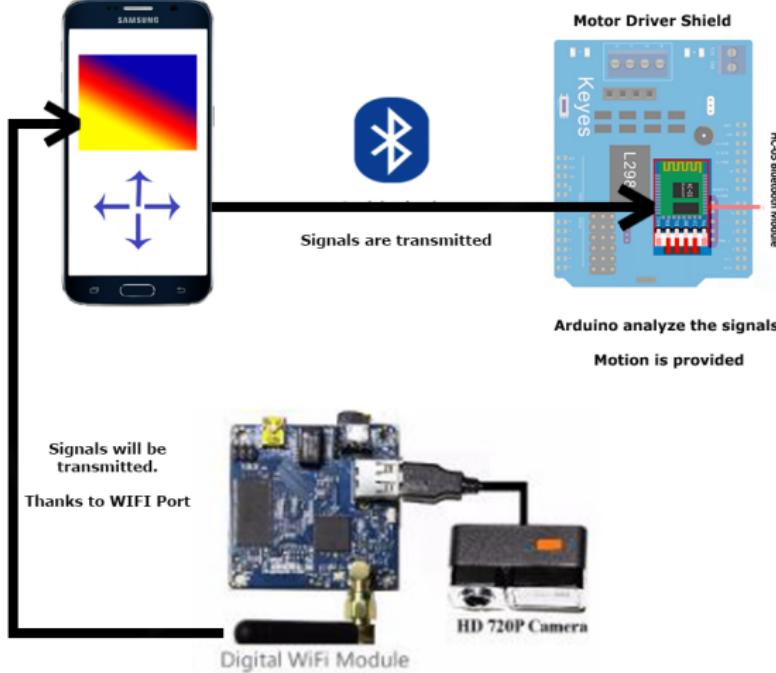


Figure: All Parts of Communications

## Communication Between Devices

From/To Android	From/To HD Camera	On WIFI Module	On - From/To Motor Shield (UNO R3)	On Bluetooth
Signal to Android from HD Cam (1)	Signal from HD Cam to Android (2)	●	●	
Signal from Android to Motor Shield (3)			Signal to Motor Shield from Android (4)	●

Table: Communication Tables

## Objectives

- 1 To enable inserting Arduino Code to give first motion to the car.
- 2 To enable Android buttons to control the Car as remoted.
- 3 To enable getting live view on the HD Cam by using WIFI Module and to display the robot eyes on Android Screen.

# Requirements

<i>Number</i>	<i>Needed</i>
1	UNO R3
1	L298P Motor Shield
1	Digital WIFI Module
1	HD Camera
1	HC05 Bluetooth Module
1	DC Motors

Table: Requirements

These are necessary parts which I will use for this robot. Now I want to talk about them.

# Arduino Uno R3

## What is Arduino?

This unit is a micro controller that makes what you want when you want it or controls physical inputs with various sensors and converts these inputs into the desired output.



Figure: Uno R3

## Arduino Uno R3

- Like your computer's processor.
- Like your brain.
- Easy to use
- Thanks to sensors and activators on Arduino, you can do whatever you want.
- This requires only a little electronic knowledge and a little programming knowledge.
- Writing a program with Arduino is simpler.

# Arduino Uno R3

## Input Outputs

- 1 Usb cable
- 2 External power supply (7-12V)
- 3 Digital pins 3,5,6,9,10,11 PWM
- 4 Rx (Receive data), Tx (Transmit data)
- 5 Reset Button
- 6 3.3V, 5V, GND

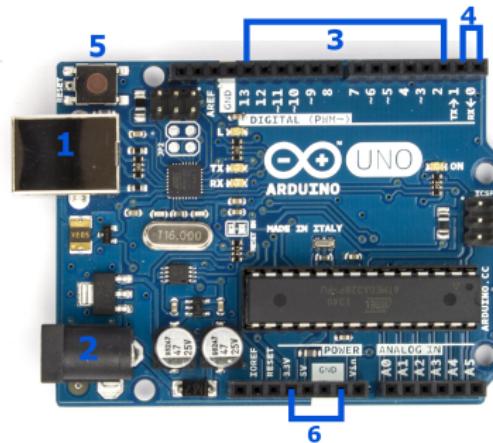


Figure: Arduino detailed review

# L298P Motor Shield

- Full-bridge motor driver
- Take Arduino's data.
- Directly plugged into the Arduino.
- Drive two separate 2A DC motors
- On board buzzer (D4), you can set the alarm ringtone.
- Bluetooth interface requires no wiring and you can plug directly.
- Six analog interfaces (A0, A1, A2, A3, A4, and A5).
- Seven digital interface that are not occupied (D2, D3, D5, D6, D7, D8, and D9).

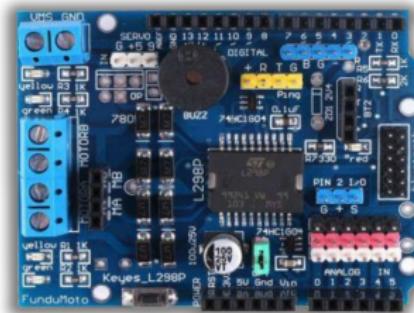


Figure: L298P Motor Shield

## L298P Motor Shield

<i>Function</i>	<i>Channel A Pin</i>	<i>Channel B Pin</i>
Direction	D12	D13
PWM	D10	D11

Table: Shield pin usage table

## Digital WIFI Module

- Thanks to this card, our Android application will be connected to Cam port by using IP which is in this module. (may be 192.168.1.1)
- The usb input on it allows us to connect our HD Cam.
- The cables from the WIFI Module are connected to the relevant parts of the L298P Motor Shield which is integrated on the Arduino. (Just to provide a power etc. GND-5V )
- The connection will be provided between android phone and camera on the WIFI Module's IP address.



Figure: Digital WIFI Module

# Digital WIFI Module

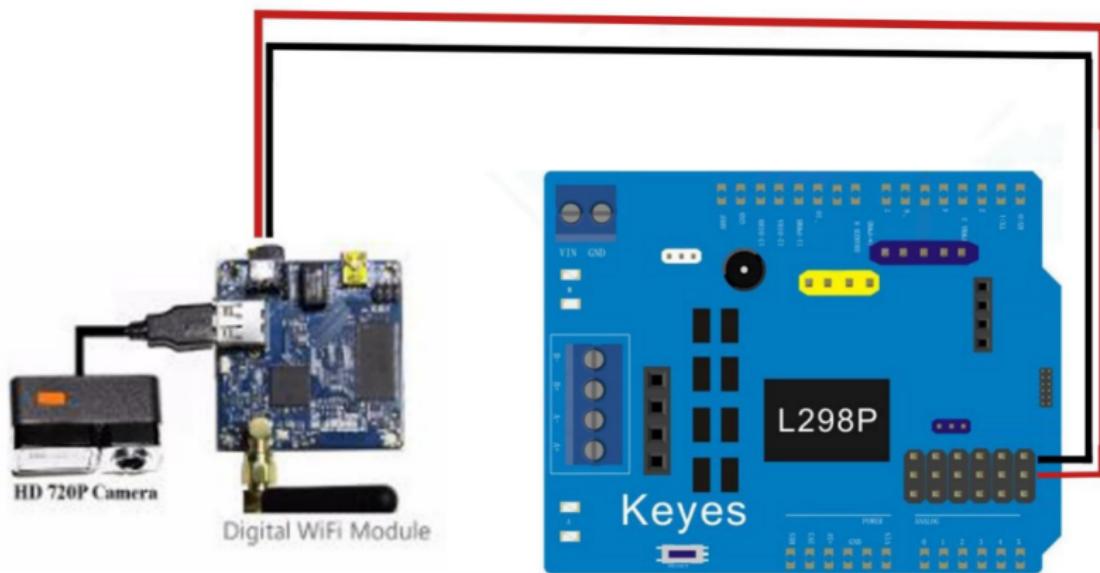


Figure: Digital WIFI Module and L298P Motor Shield

# HD Camera

## Using HD Cam

It will be possible to transfer live video to the phone with the HD camera when the necessary operation is done thanks to the usb connection which is connected to WIFI Module.



Figure: HD Cam

# Bluetooth Module

- There are 4 pins on VCC, GND, Rx and Tx on the Bluetooth module.
- From these VCC and GND are used to feed the Uno module
- This project will be designed as sending data to Bluetooth module when certain button pressed from user. (figure)
- The Bluetooth module on Arduino receives the data and send to Arduino through the TX pin of Bluetooth module (RX pin of Arduino). (figure)

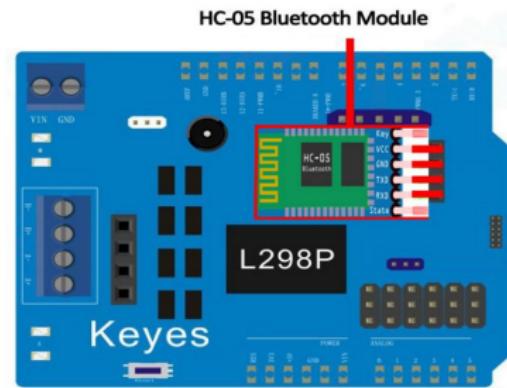


Figure: Bluetooth Module on L298P

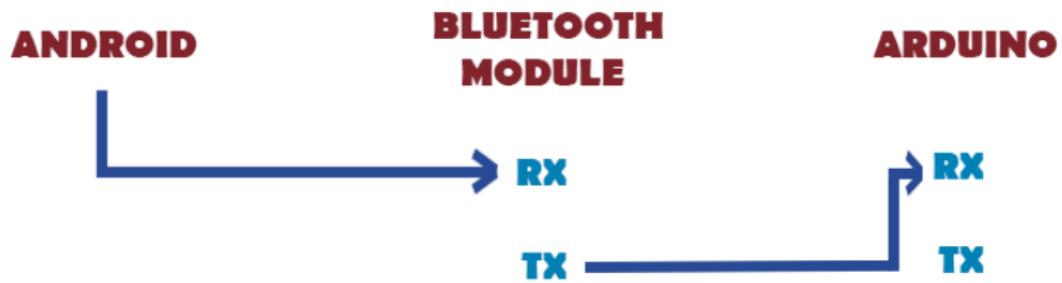


Figure: RX,TX Connection

# DC Motors

## Basic Features

- The most preferred type of motor
- Cheap, small and effective.
- Various size, shape and power

Now we will explain dc motors in terms of direction, speed, voltage and current.



Figure: DC Motor

# DC Motors

## DIRECTION

Direction of rotation of DC Motor  Direction of current

 direction  direction

## SPEED

DC Motor speed  Voltage  Load

## VOLTAGE

DC Motors 1,5V ↔ 48V

- Important because it determines the performance of the robots

# DC Motors

## CURRENT



### Specific Voltage

Dc Motor Current ✕ Load

! Shouldn't be overloaded exceed max current

### IN THIS CASE

- Dc Motor is short-circuited.
- Applied power turns into heat .
- This damages the dc motors.

✖ - Depends on  
∧ - And  
↔ - Between

# DC Motors

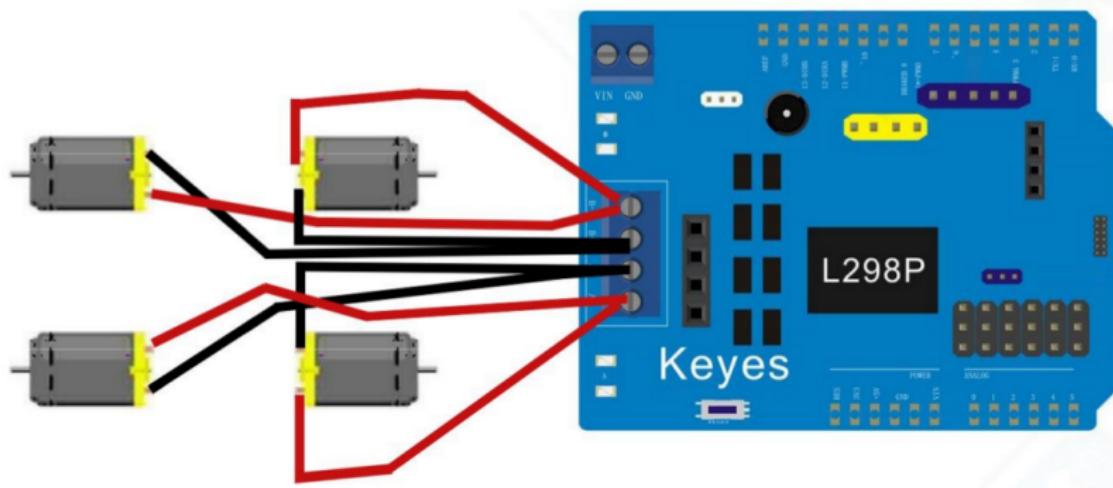


Figure: Connect L298P to DC Motors

## Car Building Process

On the hardware side, the first step is to mount the dc motors which are wired, to the chassis.

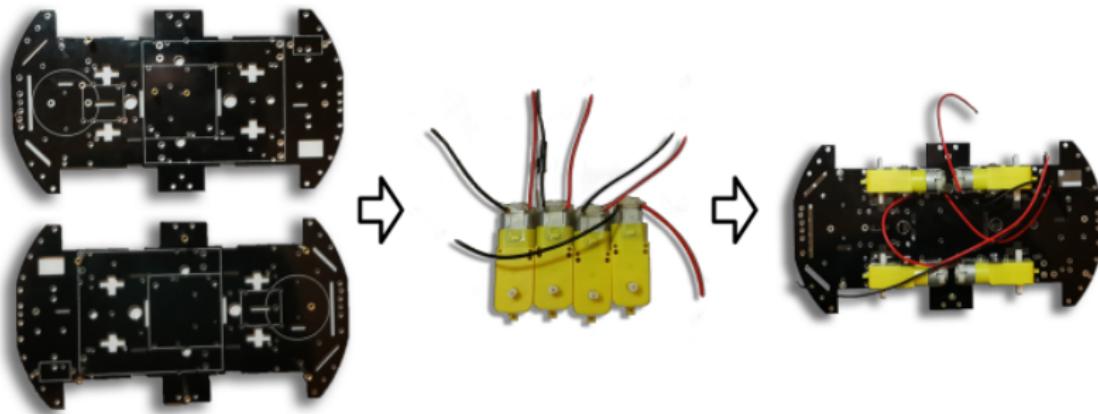


Figure: Building Process 1

## Car Building Process

The second step is to connect the dc motors to the motor driver shield. The third step is to mount the wifi shield on the chassis.

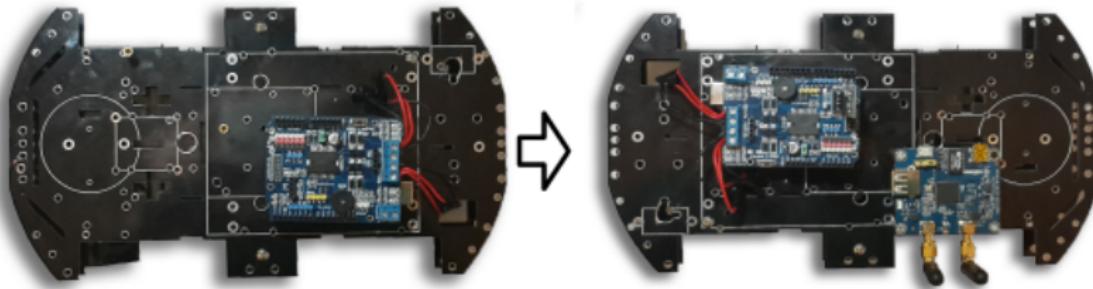


Figure: Building Process 2

## Car Building Process

The fourth step is to connect the HD Camera to the chassis. In the fifth step, the wheels are finally mounted and the car is ready.

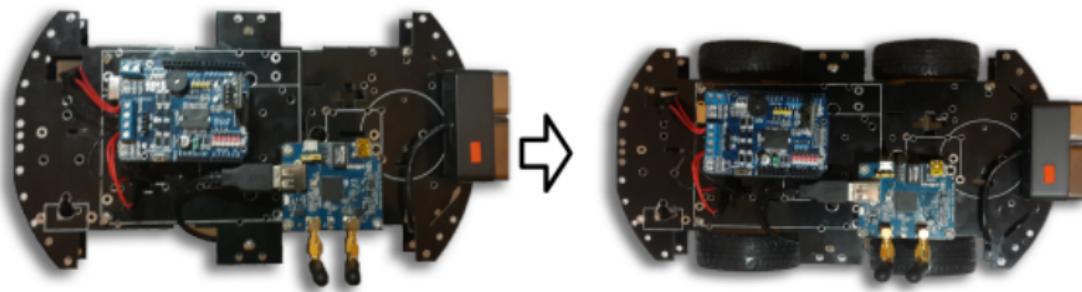
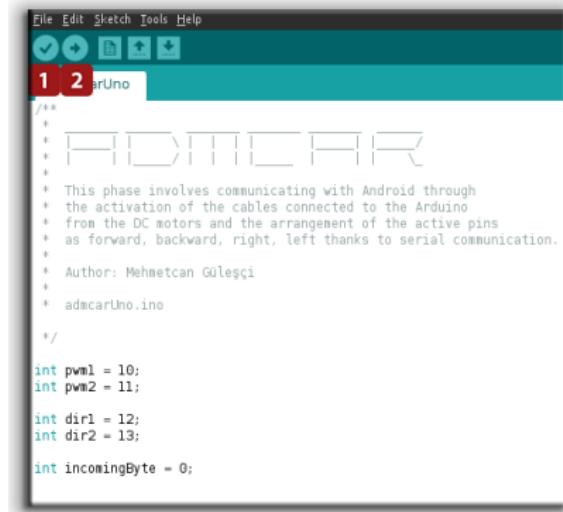


Figure: Building Process 3

- ★ Open-source Arduino Software (IDE)
- ★ It is easy to write code and upload it to the board.
- ★ The environment is written in Java.
- ★ This software can be used with any Arduino board.

# Arduino Programming Language

1. It allows us to check the Arduino code for errors.
2. It allows us to execute our Arduino code and upload to board.



The screenshot shows the Arduino IDE interface. The title bar says "admcarUno". The code editor contains the following sketch:

```
/*
 * This phase involves communicating with Android through
 * the activation of the cables connected to the Arduino
 * from the DC motors and the arrangement of the active pins
 * as forward, backward, right, left thanks to serial communication.
 *
 * Author: Mehmetcan Güleşçi
 *
 * admcarUno.ino
 */
int pwm1 = 10;
int pwm2 = 11;

int dir1 = 12;
int dir2 = 13;

int incomingByte = 0;
```

Figure: Arduino Usage

## Arduino Code

### Initialize variables

```
// Initialize variables
int pwm1 = 10;
int pwm2 = 11;

int dir1 = 12;
int dir2 = 13;

int incomingByte = 0;
```

```
void setup()
{
    pinMode(pwm1, OUTPUT);
    pinMode(pwm1, OUTPUT);
    pinMode(dir1, OUTPUT);
    pinMode(dir2, OUTPUT);

    digitalWrite(pwm1, LOW);
    digitalWrite(pwm2, LOW);
    digitalWrite(dir1, LOW);
    digitalWrite(dir2, LOW);

    Serial.begin(9600);
}
```

## Arduino Code

### Direction - Speed

```
void MotorKontrol(int mdir1, int mdir2, int pwmSpeed)
{
    digitalWrite(dir1, mdir1);
    digitalWrite(dir2, mdir2);
    analogWrite(pwm1, pwmSpeed);
    analogWrite(pwm2, pwmSpeed);
}
```

## Arduino Code

loop()

```
if (incomingByte == 10){
    MotorKontrol(HIGH, HIGH, 170); // Forward
}
else if (incomingByte == 20){
    MotorKontrol(LOW, LOW, 170); // Backward
}
else if (incomingByte == 30){
    MotorKontrol(HIGH, LOW, 170); // Left
}
else if (incomingByte == 40){
    MotorKontrol(LOW, HIGH, 170); // Right
}
else{ // Stop if another data comes
    MotorKontrol(LOW, LOW, 0);
}
```

**Step 1** Take our phone's bluetooth

**Step 2** If you have a bluetooth but it is not open, request to connect.

**Step 3** Take paired devices

**Step 4** Method that allows us to select desired devices to connect.

**Step 5** We get the mac address, the last 17 characters in the view.

**Step 6** We define an intent to start a new activity.

**Step 7** Start the activity.

## Android Code Flows

### MjpegStream.java

**Step 1** Mjpeg Input Stream which is not natively supported by Android is defined with 4 methods `getEndOfSequence`, `getStartOfSequence`, `parseContentLength` and `readFrame`.

**Step 2** URI - `http://192.168.1.1:8080/?action=stream`



**Step 3** Buffered Input Stream is created to read and skip many bytes at a time. So it prevents the cache memory from freezing.

**Step 4** Data Input Stream is created. It lets an application read primitive Java data types from an underlying input stream in a machine – independent way.

**Step 5** The Mjpeg Stream constructor is created to be called in the Car Activity class.

## Android Code Flows

### CarActivity.java

**Step 1** Sockets are created to send a data or connect to it.

**Step 2** Bluetooth sockets are defined for each forward, backward, left and right buttons to provide motion.

**Step 3** The surface create method created by the Surface.Holder Callback calls the mjpege stream constructor we created in the previous class.

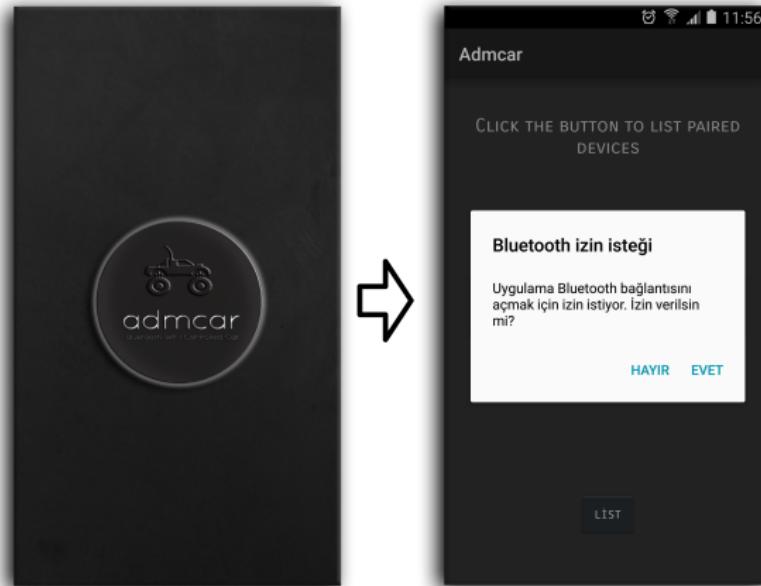
**Step 4** Through the collaboration of the Canvas and Bitmap classes defined in it, the full size image becomes active.

```
Bitmap bitmap; // This is used in getting a video stream
Canvas canvas;

canvas.drawBitmap (bitmap, null, new Rect(0, 0,
                                         canvas.getWidth(),
                                         canvas.getHeight()),
                  null) ;
```

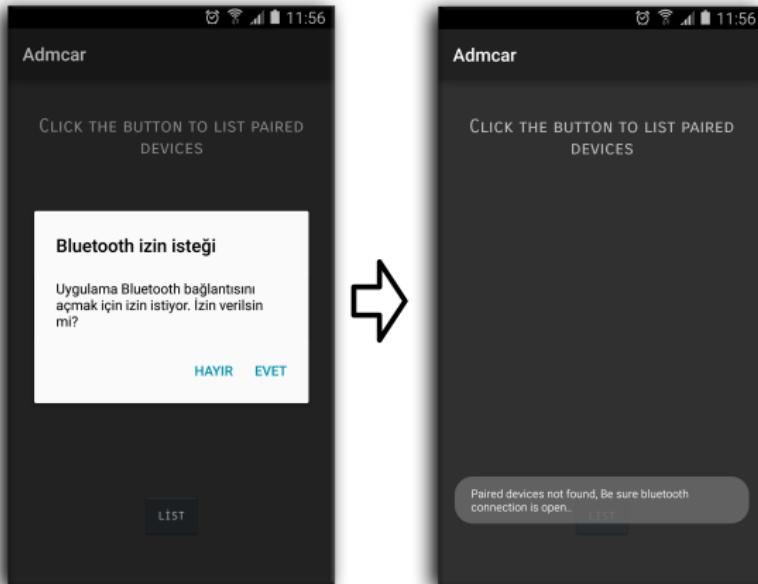
## User Interface

The application starts with the **intro picture**. Then the **second intent** comes out. This one requires connecting to the Bluetooth.



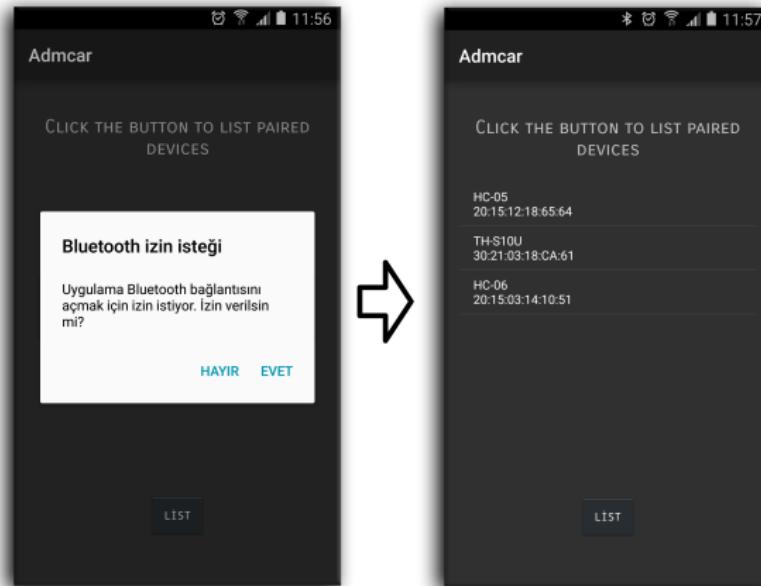
## User Interface

After the Bluetooth request has come out and when we click No option, the right window pops up and we get the warning of 'Paired devices not found. Be sure bluetooth connection is open.'



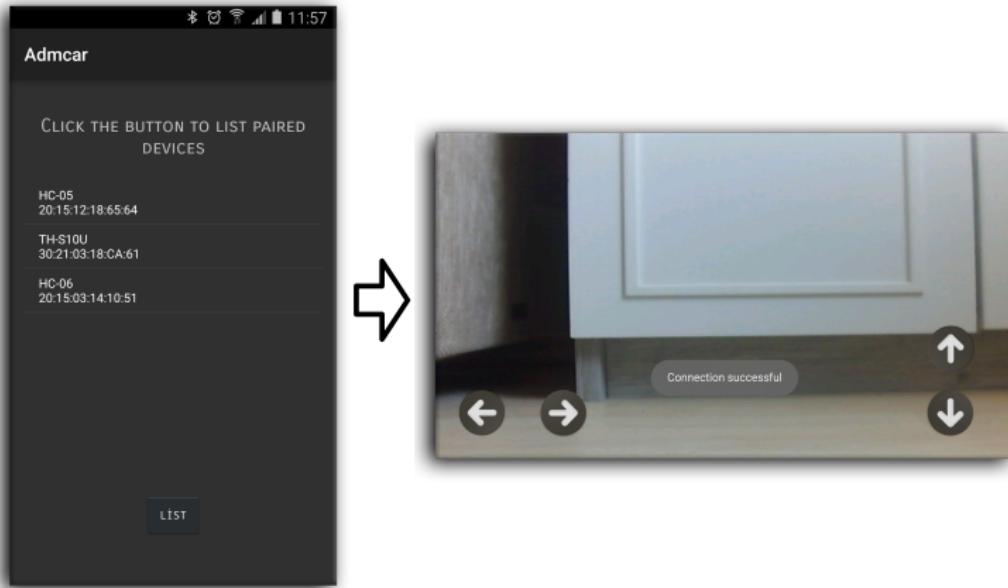
# User Interface

In the Bluetooth permission request, if you press Yes and then press the List button, we will see the list of paired devices.



## User Interface

As a final, when you click to bluetooth you are connected in, you will see the last intent to control the car. In this last intent, besides the view of the camera, it will be possible to control the car with the buttons.



# Class Diagrams

## Splash Activity

This class that provides image transition time with the help of threads.



Figure: Intro Screen Class Diagram

# Class Diagrams

## Bluetooth List

It is a class that provides Bluetooth access to the Bluetooth feature of the phone with the help of Bluetooth adapters and makes some kind of communication with the Arduino Bluetooth via the Sockets.

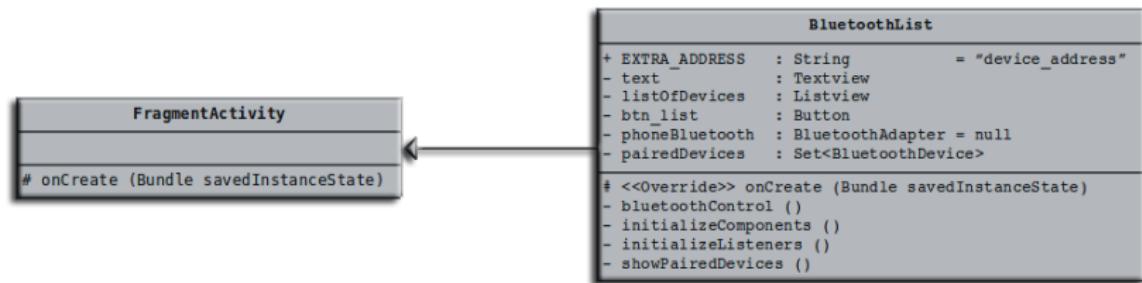


Figure: Bluetooth Class Diagram

# Class Diagrams

## Mjpeg Stream

This one contains classes that control the camera, including video stream. Basically, everything is managed from the MjpegStream, it's a view specialized to deal with Mjpeg video stream format, because Android doesn't support natively this format.

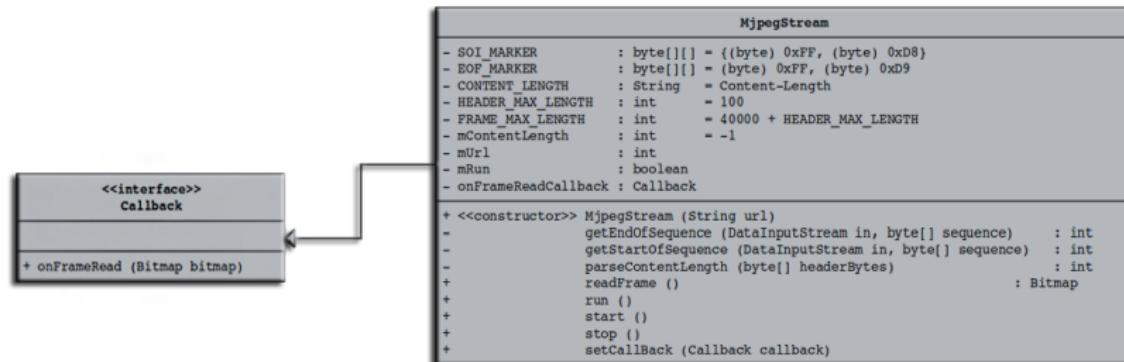


Figure: Camera Class Diagram

# Class Diagrams

## Car Activity

This class contains all classes in relationship with the car. The Car class is basically a controller for the car, it's this class that will control the direction via bluetooth sockets and monitor the live video stream.

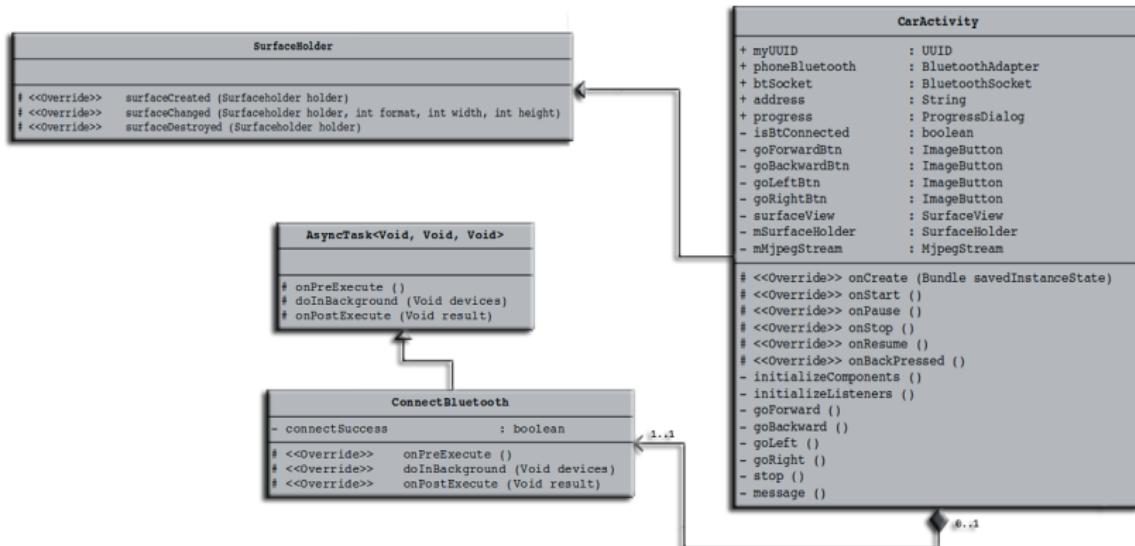


Figure: Car Activity Class Diagram

## Use Case Diagrams

To describe the system functionalities, we use the UML Use case diagrams. The Use Cases diagrams illustrated below show the actors that show also which kind of action they can use. This is the first version of the system, which can basically only move, the details about all the possible movements are described too.

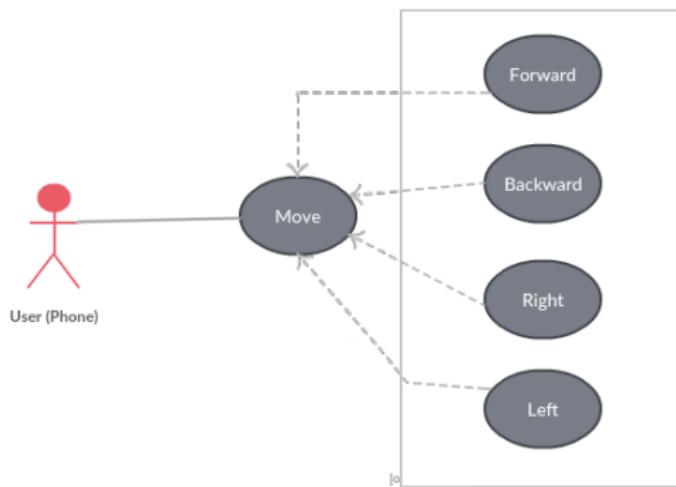


Figure: Robot UML V1

The second version is a major release of the system because we will reach our goal that is to have a video stream on the phone and control the car almost in real time with good performances.

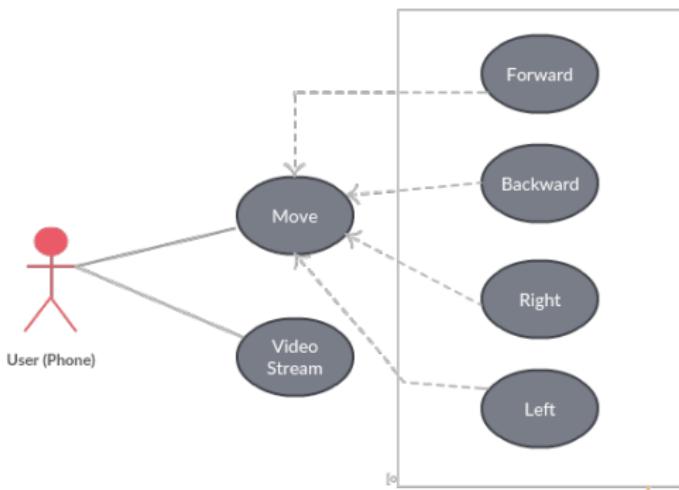


Figure: Robot UML V2

### Who can use this application ?

The users that will be play with the system could be child or adults, they just have to know how to use a smartphone, launch an application and push some buttons. There is no really limitation about the age while they know how to use the phone.

## Strengths

- The project works, we are able to do what we wanted.
- The source code for both Android and Arduino is commented, well explained and documented with external documents.
- All the source code and the documentation are available for future usages for other people and free. (Git)
- Good and speed video stream.
- With the video streaming width designed to fill the user's phone screen, the user will be able to control the car comfortably.
- It connects to Bluetooth without difficulty and never breaks unless we are away from a certain distance.

## Weaknesses

- Sometimes WiFi can shut down automatically. In this case, the video is freezing because WiFi is off.
- Really short range with video stream. ( 8 meters)

## Suggested improvements (Future of system)

- Use external antenna for greater range.
- Use only one source of power, so use the big battery used by the motor for both motor and Arduino using the regulator.
- Study and fix bugs on the applications.
- Improve the Android application to be able to get the video stream and the car control after lost them (out of range) automatically.
- Add sounds to warn the user on some events such as "out of range", "video lost", etc.
- Improve the application with new modules such as distance detection using sensor.
- Improve Android application with a different way to control the car using sensors instead of buttons and propose the choice to the user between both.
- Some sensors of the Arduino can be added to bring joy to the car.  
(honk-buzzer, distance-ultrasonic, headlights-lamp sensor etc.).

# Demo View

# TIME TABLE AND WORK SCHEDULE

Task Name	Start	Finish	2016			2017											
			Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
<b>Time and Working Schedule</b>																	
Decide Process	10/18/16	10/15/16															
Research about the working mechanism	10/15/16	10/22/16															
Learning of Materials	10/27/16	11/09/16															
Research what I use	11/13/16	11/27/16															
Decide what I use	12/01/16	12/05/16															
Research about communication between Android & Arduino via WiFi	12/07/16	12/30/16															
Process of Providing Materials	01/15/17	01/20/17															
Building the Body	01/21/17	01/28/17															
Integrating whole Parts of Body	02/03/17	02/04/17															
Software Parts (Communication Android & WiFi)	02/04/17	02/09/17															
Software Parts (Communication Android & Arduino Codes)	02/11/17	02/26/17															
Combining all functions	02/28/17	03/04/17															
Testing all parts and system (Margin of error and editions))	03/06/17	05/23/17															
Last controls	05/23/17	05/25/17															
Submitting and Presentation Process	05/27/17	06/15/17															

Figure: Time Table and Work Schedule

## REFERENCES

-  [Nov2016] - <http://electrotech.tv/arduino-ya-giris-arduino-nedir-ne-degildir-1bolum/>
-  [Jan2017] - [http://www.robotiksistem.com/dc\\_motor\\_ozellikleri.html](http://www.robotiksistem.com/dc_motor_ozellikleri.html)
-  [Feb2017] - <http://android.serverbox.ch/?p=1039>
-  [Feb2017] - <http://stackoverflow.com/questions/3205191/android-and-mjpeg>
-  [Mar2017] - <http://forum.arduino.cc/>
-  [Mar2017] - <https://sites.google.com/site/androidhowto/how-to-1/display-a-web-page>
-  [April2017] - <http://www.bluecove.org/bluecove/apidocs/javax/bluetooth/UUID.html>
-  [April2017] - <http://stackoverflow.com/questions/4032391/android-bluetooth-where-can-i-get>

# The End