UNIVERSITY KASDI MERBAH OUARGLA

Faculty of New Technologies

Information and Communication

Department of Computer and Information Technology

Memory

For The Graduation Of

MASTER ACADEMIC

Domaine : Informatique

Filière : Informatique

Spécialité : Informatique industrielle

Presented By: - Saidi Mohamed Bachir - Boukhettala Aldja

Theme

Building an Autonomous Robot & Control it By Application Android.

Publicly Supported The: …/06/2015

In front of the jury:

Mr.	Mahdjoub	Med Bachir	University of Ouargla	**President**
Mr.	Bakkari	Fouad	University of Ouargla	**Examiner**
Mr.	Zega	Adel	University of Ouargla	**Protractor**

Acknowledgements

Acknowledgements First of all, we thank ALLAH our creator for giving our forces the will and courage to accomplish this modest work.

We extend our thanks to the Best teacher "adel zga" for his advice and headed from the beginning to the end of this work.

We also thank the gentlemen of the jury members for accepting and assess our project and for all their comments and criticisms.

Finally, we express our deep gratitude to our families who have supported us and every one given us some help to us to success our project .And all the teachers who have contributed to our learning.

Dedication

At the most beautiful creature that ALLAH created on earth,

In this source of tenderness, patience and

generosity,,,

To my mother!

To my father!

To all my brothers. My sisters. my whole family.

To all my friends and colleagues especially Rahim ,Tarek, Abd elbasset , Mohamed, khaled , wahid ,ahemd , & all my friends and to my special persons.

To all the students of the 2010/2015

for Mr. Adel Zga.

Mrs. Benzaoui Wafa .

For all, who gave me the strength

to continue .....

« ***Saidi Mohamed Bachir*** »

Dedication

At the most beautiful creature that God created on earth, In this source of tenderness, patience and generosity,,,

To my mother!

To my father!

To all my brothers. yacine and tarek

To my grand-parents and my whole family.

To all my friends and colleagues especially radia & sabrine .

To all the students of the 2011/2012

A Mr. haguigua abd elhafid.

To all those who, with a word, gave me the strength to continue ... ..

"BOUKHETTALA ALDJA "

Summary

chapter II: Conception for Arduino UNO

chapter III: Design And Software Implementation

Figure list

table list:

ABBREVIATION

LED: *light-emitting diode.*

IDII: *Interaction Design Institute Ivrea.*

MIT: *Massachusetts Institute of Technology.*

IDE: *Integrated Development Environment.*

PCB: *Printed Circuit Board.*

UART: *Universal Asynchronous Receiver Transmitter.*

OOP: *Object-Oriented Programming.*

TX: *Transmission Data.*

RX: *Reception Data.*

PWM: *Pulse-width modulation.*

Abstract

Abstract *:*Electronics become interactive surround us from every side, accompany you'll find wherever you go in your car at home in work or even in your small pocket, there is always a smart electronic tool interact with you anywhere you go, Arduino is a great open source project that aims to provide free software and Plate interactive development of open source used in the construction of electronic circuits intelligent and can interact with humans easily and is pleased and anyone can use them to work legitimately own without the need for prior knowledge with the knowledge of electronics, without studying the complexities of electronic circuits, and the painting is made up of the cycle of email with microcontroller accurate programmable through the arduino IDE program for the Arduino and which can be downloaded free of charge to all operating systems, in our project, and with merit Arduino we try to build machine "robot" ,it can avoiding obstacles and can be controlled by the application in Android system.

Key words: Electronics, Arduino, open source, arduino IDE, microcontroller, system android, obstacles.

ملخص *:*أصبحت الإلكترونيات التفاعلية تحيط بنا من كل جانب, ستجدها ترافقك أينما ذهبت في السيارة في بيتك في العمل أو حتى في جيبك الصغير, هناك دائما أداة الكترونية ذكيه تتفاعل معك في اى مكان تذهب إليه, اردوينو هو مشروع كبير مفتوح المصدر يهدف إلى توفير برمجيات مجانية و لوحه تطوير تفاعلية مفتوحة المصدر تستخدم في بناء دوائر الكترونية ذكية و تستطيع التفاعل مع البشر بسهولة و يسر و يمكن لأى فرد استخدامها لعمل مشروعة الخاصة دون الحاجة لمعرفه مسبقة بعلم الأإلكترونيات و دون دراسة تعقيدات الدوائر الأإلكترونية, و تتكون هذه اللوحة من دائرة الكترونية تحتوى على متحكم دقيق قابل للبرمجة عن طريق برنامج بيئة التطوير المتكاملة للاردوينو و الذي يمكن تحميله مجانا لجميع انظمه التشغيل, و بفضل اردوينو توصلنا في مشروعنا إلى ابتكار آلة متحركة تستطيع تجنب العقبات ويمكن التحكم فيها بواسطة نظام اندرو يد.

الكلمات الدالة *:* الإلكترونيات ,اردوينو , مفتوح المصدر, بيئة التطوير المتكاملة , متحكم دقيق , نظام اندرويد, العقبات

General introduction

Fusion of different sensors such as sound sensor, vision sensor, temperature sensor etc. allows the extraction of information which cannot be acquired by a single sensor. Different type of sensors work differently and they have their strengths and weaknesses. One sensor cannot provide all the necessary information. Sensor fusion combines the strength of different sensor to overcome the drawbacks of the other. For an autonomous mobile robot, sensor fusion is important to perceive its environment. Without knowing the surroundings it is not possible for it to navigate around.

An autonomous robot moves unsupervised. It obtains information of surrounding environments using its sensors and decides its course of action according to its programming without any external help. If the information provided is inaccurate or incomplete, it becomes hard for the robot to decide its next action. Sensor fusion allows a robot to perceive its surroundings like human beings. Human beings use their sense of vision, sound, smell, touch and taste to understand their surroundings. Information from one sensor alone is not enough to give accurate information. A food may look good but without the sense of smell and taste it cannot be determined whether the food is still edible or not. So our task is to fuse some sensors for the purpose of object detection and size measurement which is cost effective.

Robotics is a leading branch of engineering which demands knowledge of hardware, sensors, actuators and programming. The result is a system which can be made to do a lot of different things. However, to develop such a system is expensive and difficult. So, we have come up with a plan to build an autonomous mobile robot which is less expensive. A robot has three main different parts – preceptors, processors and actuators.

The perceptors are the sensors which provide information about the surrounding environment to the robot. There are many works had done for object detection. Those robots are efficient in the purpose of accuracy. But they are very costly. So we move in a direction where we can have a robot that is cost effective and good enough in its accuracy.

In our project we choose sensor ultrasonic and board arduino to building our robot and we present the history and development of components (arduino board) used in our project all this in chapter one and we description with definition all components (arduino uno , ultrasonic bluetooth ... etc)in chapter two ,also in chapter three we explained how we building our robot and it covering essentially two aspects :

First aspects : this aspect is building robot to avoiding obstacle with helping sensor ultrasonic and h-bridge l298n to control dc motors .

Second aspects: this aspect it have controling robot by application android to change direction and speed our robot .

Chapter I

Overview on Arduino & Robot

1-introduction

2-Now,the story in detail

3-Wath is arduino

4-Why arduino

5-Arduino development Environment

6-Conclision

1- Introduction

It was in the year 2005 that the first ever Arduino board was born in the classrooms of the Interactive Design Institute in Ivrea, Italy. Well, if you are not very familiar with the term an **Arduino is an Open Source microcontroller based development board** that has opened the doors of electronics to a number of designers and creative engineers.

It was in the Interactive Design Institute that a hardware thesis was contributed for a wiring design by a Colombian student named Hernando Barragan. The title of the thesis was "Arduino–La rivoluzione dell'open hardware" ("Arduino – The Revolution of Open Hardware"). Yes, it sounded a little different from the usual thesis but none would have imagined that it would carve a niche in the field of electronics.

A team of five developers worked on this thesis and when the new wiring platform was complete, they worked to make it much lighter, less expensive, and available to the open source community.[02]

2- About the Arduino

The new prototype board, the Arduino, created by Massimo Banzi and other founders, is a low cost microcontroller board that allows even a novice to do great things in electronics. An Arduino can be connected to all kind of lights, motors, sensors and other devices; easy-to-learn programming language can be used to program how the new creation behaves. Using the Arduino, you can build an interactive display or a mobile robot or anything that you can imagine.

You can purchase an Arduino board for just about US $30 or build your own board from scratch. Consequently, Arduino has become the most powerful open source hardware movement of its time.[02]

3- Now, the Story in Detail…

As mentioned earlier, it all started in Ivrea, Italy.

Let's have a look at how the name Arduino, that sounds quite strange for an electronic device, was chosen. This beautiful town of Ivrea, situated in Northern Italy, is quite famous for its underdog kings. In the year 1002 AD, King Arduin (you got it right!) ruled the country; two years later, he was dethroned by King Henry II of Germany. In memoir of this King Arduin, there is this 'Bar Di Re Arduino', a pub on the cobble stoned street in the town. Well, this place is where a new era in electronics had its roots! This bar was frequently visited by Massimo Banzi, one of the founders of Arduino, who taught at Ivrea. He was the one who gave the name Arduino to this low-cost microcontroller board in honor of the place!

Before getting into how the Arduino was developed and used, let's know who the core members of the Arduino developer team are: Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, and David Mellis.[02]

Figure 01: Arduino developer team - David Cuartielles, Gianluca Martino, Tom Igoe, David Mellis, and Massimo Banzi. Photo Courtesy - Randi Klett/IEEE Spectrum

4- Arduino was an answer to how to teach students to create electronics fast…

It was in the year 2002 that Banzi, a software architect by profession, was recruited as an associate professor by IDII in order to promote novel ways of doing interactive design, in other words, physical computing. Though he had some good ideas, limited class time and shrinking budget didn't help him much. Like most of his colleagues, Banzi had to rely on the BASIC Stamp, a microcontroller developed by Parallax, a California based company. Engineers had been making use of this microcontroller for about a decade. The Stamp was coded using the BASIC programming language and looked like a tidy little circuit board packed with essentials of a power supply, memory, a microcontroller, and input/output ports to which hardware can be attached. However, the BASIC Stamp had two issues according to Banzi. One, it did not have sufficient computing power for some of the projects his students had conceptualized and two, it was pretty expensive. In fact, a board with its basic parts cost about US $100. Moreover, Banzi also required something that could run on Macintosh computers which were largely used by designers at IDII. *The new Arduino microcontroller that best suited their needs had signs of its roots at this point of time.*

Meanwhile a designer-friendly programming language called "Processing" had been developed by Banzi's colleague from MIT. Processing was quickly gaining popularity as it enabled even amateur programmers to create complex and beautiful data visualizations! It was an extremely easy-to-use Integrated Development Environment or (IDE). Banzi really liked this concept and wondered if he and his team could create similar software programs to code a microcontroller instead of graphics on a screen.[02]

5- Contribution of Hernando Barragan

One of Banzi's students, Hernando Barragan, took the first baby step in the direction towards creating software tools similar to Processing. He developed a new prototyping platform known as Wiring; it included both a user-friendly IDE as well as a ready-to-use circuit board. It turned out to be a promising project the success of which continues till date; however, Banzi was already having bigger dreams. He wished to make a platform that was even cheaper, simpler and easier to use.[02]

6- The First Prototype Board

Well, Banzi succeeded in creating the first prototype board in the year 2005; it was a simple design and at that time, it wasn't called Arduino. Of course, by now, you would know how he had coined the name later that year.[02]

7- Open Source Model – A Big Decision

Banzi and his collaborators strongly believed in open-source software. As the purpose was to develop a quick and easily accessible platform, they thought it would be better to open up the project to as many people as possible instead of keeping it closed. Another crucial factor that contributed to that big decision was that after operating for nearly five years, IDII had no more funds left and was in fact going to shut its doors. All the faculty members feared that their projects might not survive or would be embezzled. It was at this crucial point of time that Banzi decided to go ahead and make it open source![02]

8- How Banzi and team managed to create Arduino and make it available for public.

Pretty obviously, the open source model had always been used to fuel innovation for software and never hardware. If they had to make it work, they had to find a suitable licensing solution that could apply to the board. After a little investigation, Banzi and team looked at the whole thing from a different angle and decided to use a license from Creative Commons, a nonprofit group whose agreements were normally used for cultural works like writing and music. According to Banzi, hardware is a piece of culture that must be shared with other people!

Well, the next step was to make the board. The group decided to fix a specific, student-friendly price of $30 as their goal. Banzi felt that the Arduino should be affordable for all students. However, they also wanted to make it really quirky, something that would stand out and look cool as well. While other boards were green, they wanted to make theirs blue. While a few manufacturers saved on input and output pins, they added a lot to their board. Quite weirdly, they added a little map of Italy on the back of the Arduino board!

Gianluca Martino, one of the 'real' engineers on the team felt that the nontraditional and raw approach to circuit board design was pretty enlightening. He thought that the product created was a result of a new way of thinking about

electronics; not in an engineering way wherein you have to count electrodes, but using a DIY approach.

The product created by the team comprised of inexpensive parts that could be found easily if users wanted to create their own boards. However, an important decision was to ascertain that it would essentially be plug_and_play: something someone could just take out of a box, plug into a system and use it right away. On the other hand, boards such as the BASIC Stamp demanded the users to shell out a lot of other items that ultimately added to the total cost. However, for the Arduino, a user needs to just pull out a USB cable from the board and merely connect it to a computer to program the device.[02]

9- What is Arduino?

Arduino is a tool for making computers that can sense and control more of the physical world than your desktop computer. It's an open-source physical computing platform based on a simple microcontroller board, and a development environment for writing software for the board.

Arduino can be used to develop interactive objects, taking inputs from a variety of switches or sensors, and controlling a variety of lights, motors, and other physical outputs. Arduino projects can be stand-alone, or they can communicate with software running on your computer (e.g. Flash, Processing, MaxMSP.) The boards can be assembled by hand or purchased preassembled; the open-source IDE can be downloaded for free. The Arduino programming language is an implementation of Wiring, a similar physical computing platform, which is based on the Processing multimedia programming environment.[02]

10- Why Arduino?

There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package.

Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than $50.
- Cross-platform - The Arduino software runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller Systems are limited to Windows.
- Simple, clear programming environment - The Arduino programming environment is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with the look and feel of Arduino.
- Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
- Open source and extensible hardware - The Arduino is based on Atmel's ATMEGA8 and ATMEGA168 microcontrollers. The plans for the modules are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money. [02]

11- Compare board specs

- This simple table shows a quick comparison between the characteristics of all the Arduino boards.[03]

Name	Processor	Operating Voltage/Input Voltage	CPU Speed	Analog In/Out	Digital IO/PWM	EEPROM [KB]	SRAM [KB]	Flash [KB]	USB	UART
Uno	ATmega328	5 V/7-12 V	16MHz	6/0	14/6	1	2	32	Regular	1
Due	AT91SAM3X8E	3.3 V/7-12 V	84 MHz	12/2	54/12	-	96	512	2 Micro	4
Leonardo	ATmega32u4	5 V/7-12 V	16MHz	12/0	20/7	1	2.5	32	Micro	1
Mega 2560	ATmega2560	5 V/7-12 V	16MHz	16/0	54/15	4	8	256	Regular	4
Mega ADK	ATmega2560	5 V/7-12 V	16MHz	16/0	54/15	4	8	256	Regular	4
Micro	ATmega32u4	5 V/7-12 V	16MHz	12/0	20/7	1	2.5	32	Micro	1
Mini	ATmega328	5 V/7-9 V	16MHz	8/0	14/6	1	2	32	-	-
Nano	ATmega168 ATmega328	5 V/7-9 V	16MHz	8/0	14/6	0.512 1	1 2	16 32	Mini-B	1
Ethernet	ATmega328	5 V/7-12 V	16MHz	6/0	14/4	1	2	32	Regular	-
Esplora	ATmega32u4	5 V/7-12 V	16MHz	-	-	1	2.5	32	Micro	-
ArduinoBT	ATmega328	5 V/2.5-12 V	16MHz	6/0	14/6	1	2	32	-	1
Fio	ATmega328P	3.3 V/3.7-7 V	8MHz	8/0	14/6	1	2	32	Mini	1
Pro (168)	ATmega168	3.3 V/3.35-12 V	8MHz	6/0	14/6	0.512	1	16	-	1
Pro (328)	ATmega328	5 V/5-12 V	16MHz	6/0	14/6	1	2	32	-	1
Pro Mini	ATmega168	3.3 V/3.35-12 V 5 V/5-12 V	8MHz 16MHz	6/0	14/6	0.512	1	16	-	1
LilyPad	ATmega168V ATmega328V	2.7-5.5 V/2.7-5.5 V	8MHz	6/0	14/6	0.512	1	16	-	-
LilyPad USB	ATmega32u4	3.3 V/3.8-5V	8MHz	4/0	9/4	1	2.5	32	Micro	-
LilyPad Simple	ATmega328	2.7-5.5 V/2.7-5.5 V	8MHz	4/0	9/4	1	2	32	-	-
LilyPad SimpleSnap	ATmega328	2.7-5.5 V/2.7-5.5 V	8MHz	4/0	9/4	1	2	32	-	-
Yun	ATmega32u4	5 V	16MHz	12/0	20/7	1	2.5	32	Micro	1

12- Some of Products arduino" boards, shields ,accessories & kits"

12-1 Shields:

Shields are boards that can be plugged on top of the Arduino PCB extending its capabilities. The different shields follow the same philosophy as the original toolkit: they are easy to mount, and cheap to produce.[04]

12-2 Kits :

A kit that will teach you the basic of electronic and programming with Arduino.[05]

Figure 02: **The arduino kits.**

13- Arduino Development Environment

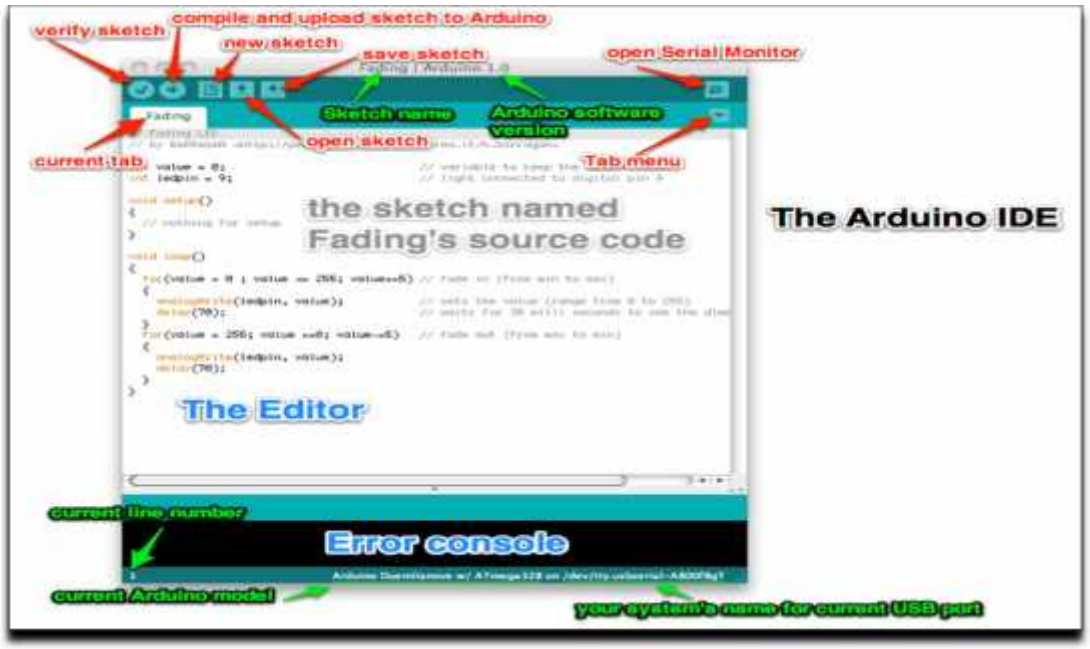
The Arduino development environment contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions, and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them.[06]

14- Arduino V 1.6.0

The Open-source Arduino software (IDE) make it easy to write code and upload it to the board . It runs on windows, Mac Os X , and Linux . The environment is written in java and based on Processing and other open-source software .This software can be used with any arduino board.[07]

15- Writing Sketches

Software written using Arduino are called **sketches**. These sketches are written in the text editor. Sketches are saved with the file extension .ino. It has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino environment including complete error messages and other information. The bottom righthand corner of the window displays the current board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.[04]



The Arduino IDE.

16- Some Project of Arduino

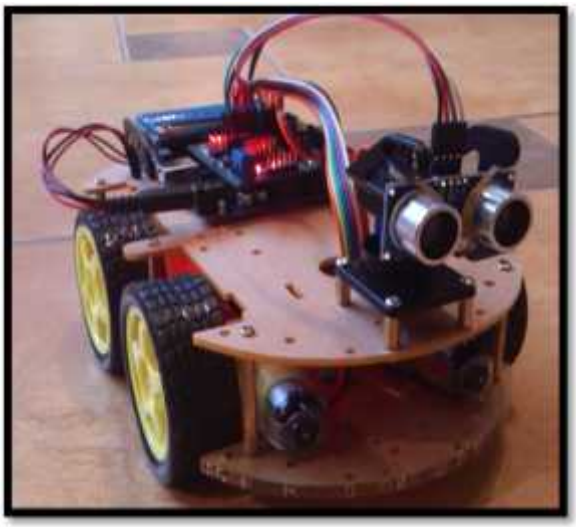


Figure 03: **Obstacle Avoidance Robot Car .**

Figure 04: infrared Robot from team arduino.

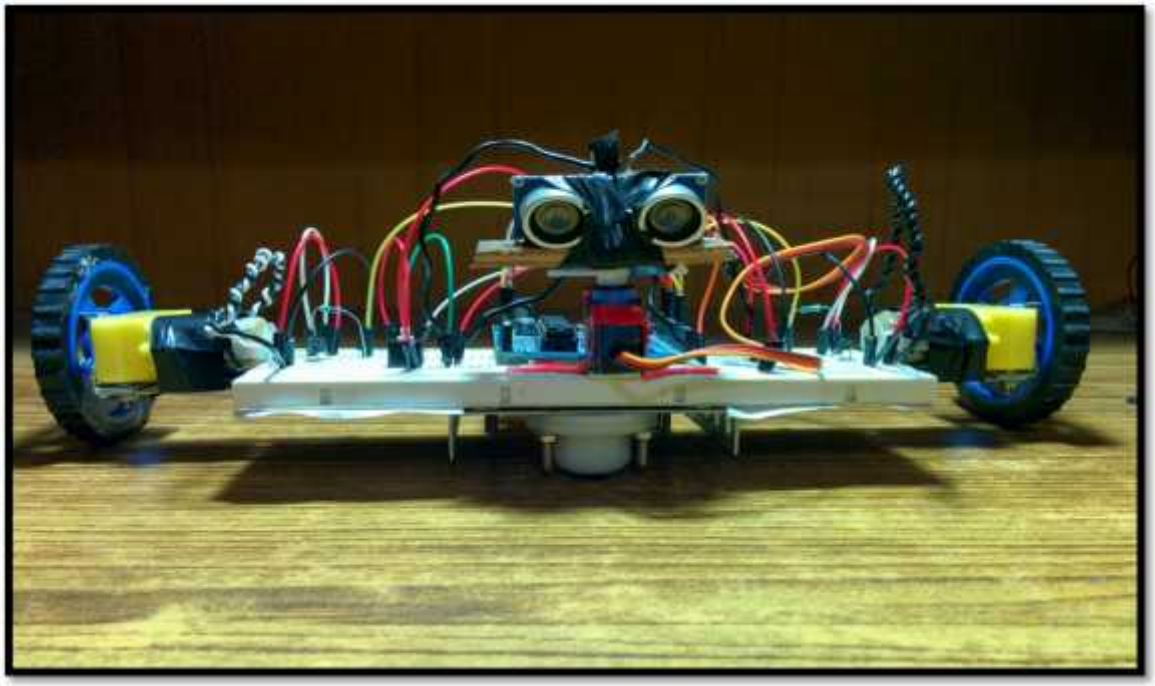


Figure 05: Obstacle Avoiding Robot: As Simple As It Can Be.

17- Problem Statement

«Build an autonomous robot which can avoiding obstacle and controlled by application system android ».

So, we have to make an autonomous robot capable of avoiding an obstacle consisting of detect obstacle and calculate distance after that make decision to avoiding it. Then the robot using a suitable gripping mechanism has to change direction by instructions programmed in microcontroller arduino uno and control speed of motors by module h-bridge L298N ,and the robot can be controlled by smartphone with application android to change direction. The robot consisting from two tasks :

- Detection the unknown obstacle by sensor ultrasonic and try to avoiding and change direction .
- The robot can be controlled by application system android. [08]

18-conclusion

finaly in this chapter we can say now we have some look about arduino because we talking about Story and History of Development of Arduino , as We discussed about definition of arduino and reason to choose it and about some products and project and in finally we put problem statement about robot autonomous and we talking about your project in second chapter .

Chapter II

Conception for Arduino UNO

1-introduction

2-definition of arduino uno

3-the arduino IDE

4-definition of extension

5-arduino language

6-Conclision

1- Introduction

An Arduino is an open-source microcontroller development board. In plain English, you can use the Arduino to read sensors and control things like motors and lights. This allows you to upload programs to this board which can then interact with things in the real world. With this, you can make devices which respond and react to the world at large.

In this section we will look for some definitions about Arduino and the appropriate hardware for the programmer a root to avoid obstacles.

2- Definition arduino uno:

The Arduino Uno is an open-source electronics prototyping platform based on flexible easy-to-use hardware and software. And its interested in creating interactive objects or environments.[09]

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino,moving forward. The Uno is the latest in a series of USB.[10]

2-1 Technical specification :

- Microcontroller ATmega328.
- Operating Voltage 5V.
- Input Voltage (recommended) 7-12V.
- Input Voltage (limits) 6-20V.
- Digital I/O Pins 14 (of which 6 provide PWM output).
- Analog Input Pins 6.
- DC Current per I/O Pin 40 mA.
- DC Current for 3.3V Pin 50 mA.
- Flash Memory 32 KB of which 0.5 KB used by boot loader.
- SRAM 2 KB.
- EEPROM 1 KB.
- Clock Speed 16 MHz.[10]

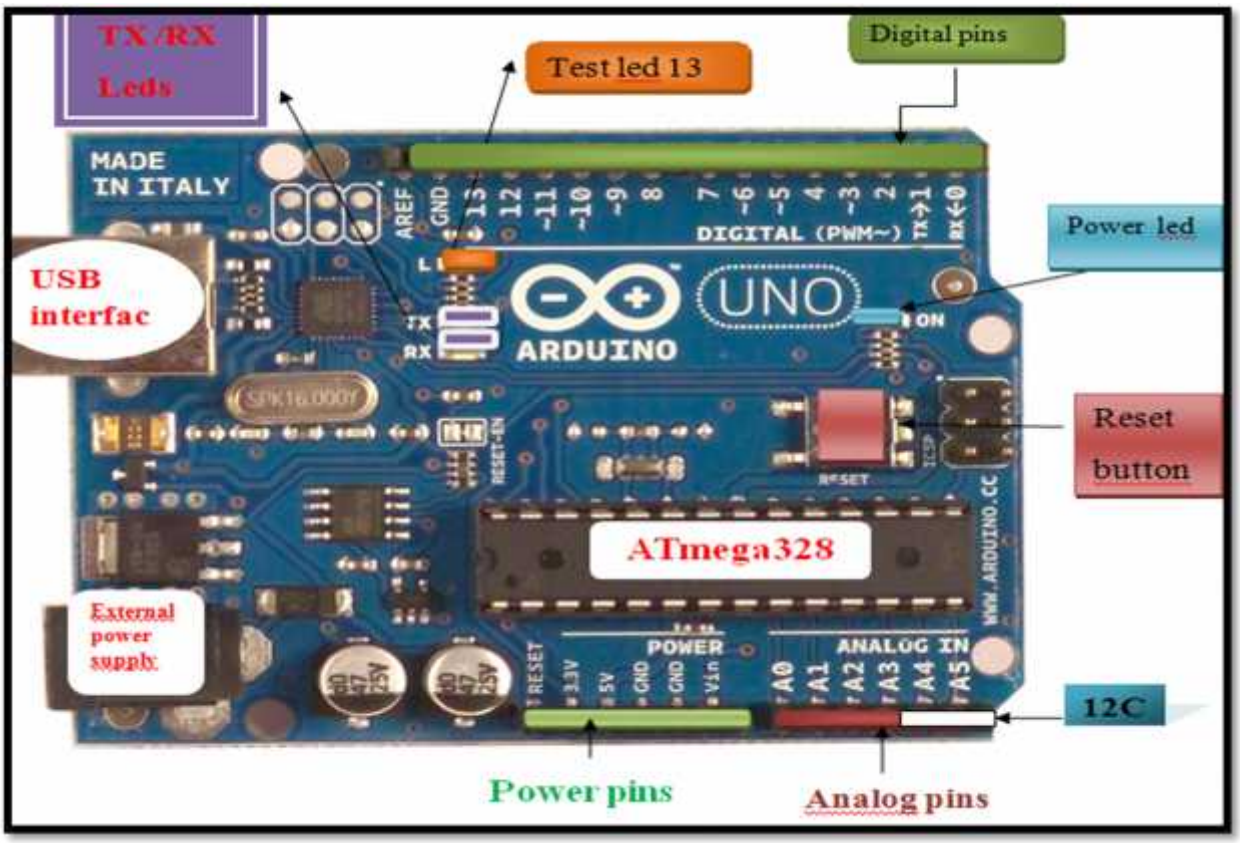


Figure 06: The board Arduino UNO

2-2 Memory:

The Atmega328 has 32 KB of flash memory for storing code (of which 0,5 KB is used for the boot loader); It has also 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).[11]

2-3 Communication:

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer.[11]

3- The arduino IDE:

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board.[12]

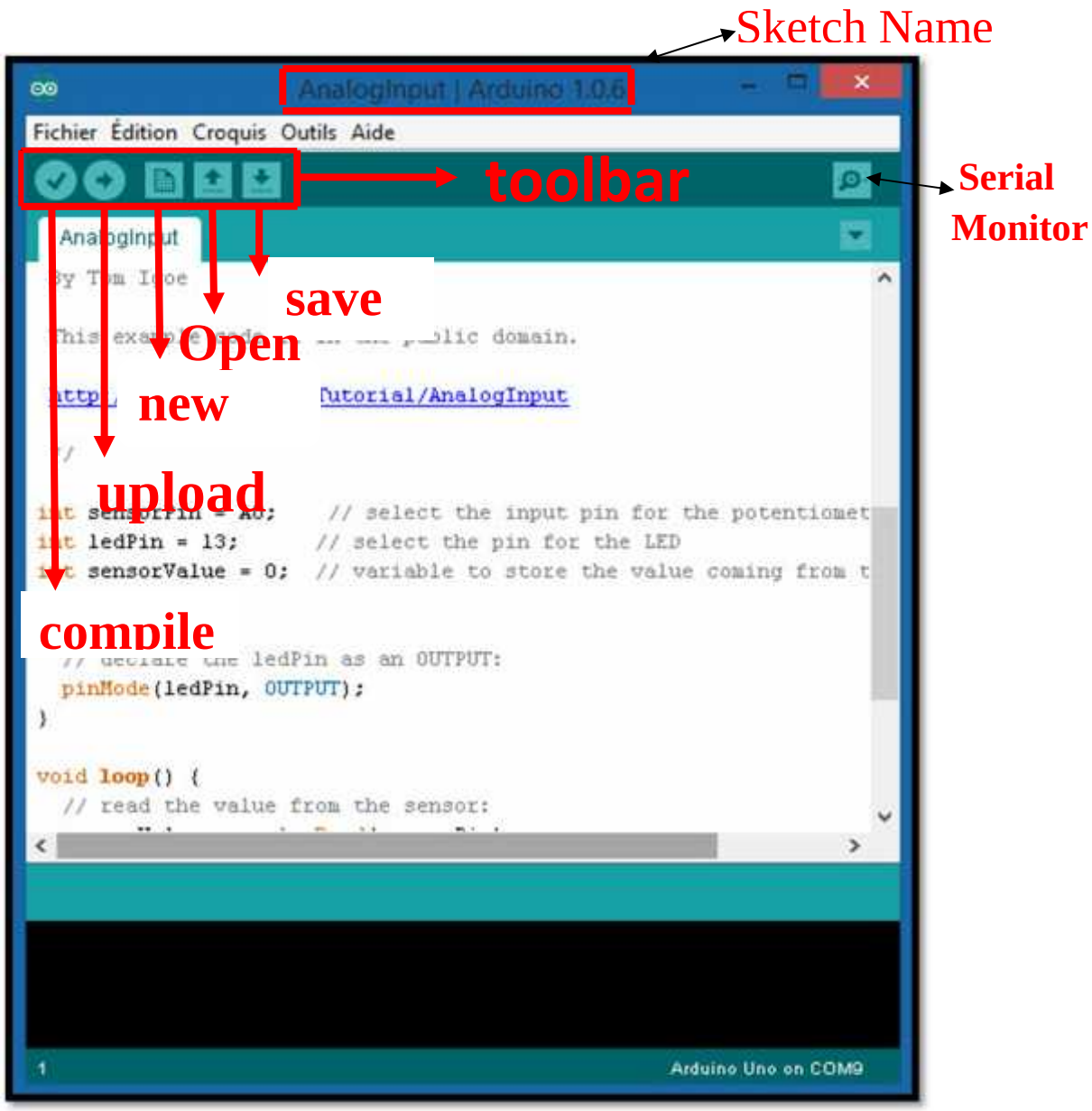


Figure 07: the arduino IDE

There are six icons from left to right that we have to know very well because we'll use them every time:

• **Compile** (check symbol): This provides code checking for errors .

• **Upload** (right-side arrow): This compiles and uploads our code to the Arduino board .

• **New** (small blank page): This creates a new blank sketch .

• **Open** (up arrow): This opens a list of all sketches already existing in our sketchbook .

• **Save** (down arrow): This saves our sketch in our sketchbook .

• **Serial Monitor** (small magnifying glass): This provides the serial monitoring .

4- Definition of the extension:

4-1 Ultrasonic:

4-1-1 Definition :

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit. The basic principle of work:

- ❖ Using IO trigger for at least 10us high level signal.
- ❖ The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.
- ❖ IF the signal back, through high level , time of high output IO duration is the time from sending ultrasonic to returning.
- ❖ Test distance = (high level time× velocity of sound (340M/S) / 2.[13]

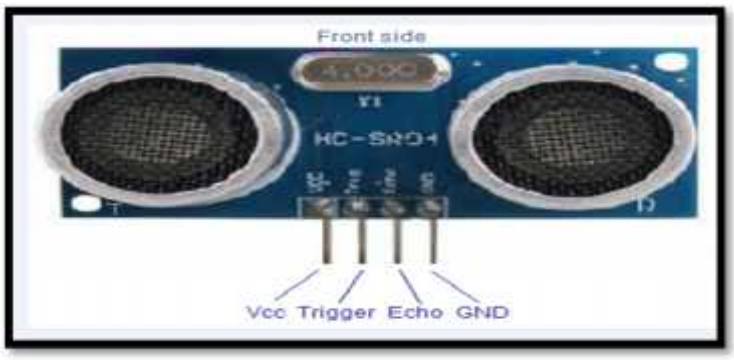


Figure 08: Ultrasonic ranging module HC - SR04

4-1-2 Table Module pin definitions :

Types	Pin Symbol	Pin Function Description
HC-SR04	VCC	5V power supply
	Trig	Trigger pin
	Echo	Receive pin
	GND	Power ground

4-2 Module HC-06 Serial Port Bluetooth:

4-2-1 Definition

Electronic brick of HC-06 serial port Bluetooth can be connected to hardware UART or analog UART on the control board. With Bluetooth communication, it can achieve wireless transmission which can be applied in various kinds of remote communication occasions.[14]

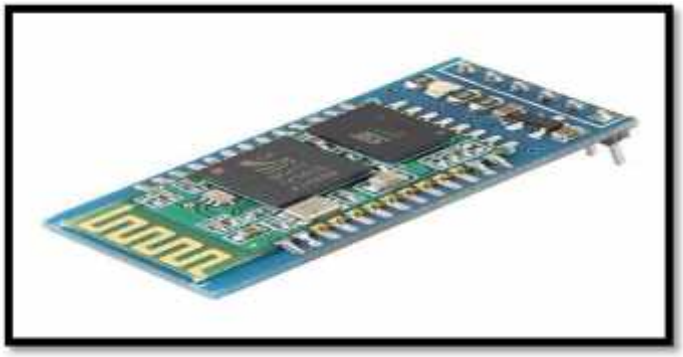


Figure 09: **Serial Port Bluetooth HC-06**

4-3L298N Dual H-Bridge:

4-3-1 Definition:

The Motor Shield is based on the L298, which is a dual full-bridge driver designed to drive inductive loads such as relays, solenoids, DC and stepping motors. It lets you drive two DC motors , controlling the speed and direction of each one independently.[15]

4-3-2 Technical Summary :

- Operating Voltage 4V to 35V.
- Motor controller L298N, Drives 2 DC motors or 1 stepper motor.
- Max current 2A per channel or 4A max.
- Free running stop and brake function.
- Chip: ST L298N.
- Logic power supply:5v.
- Max power:25w.
- Weight: 35g.
- Size:55mm x 60mm x 30mm.
- Storage temperature:-25 to +135.

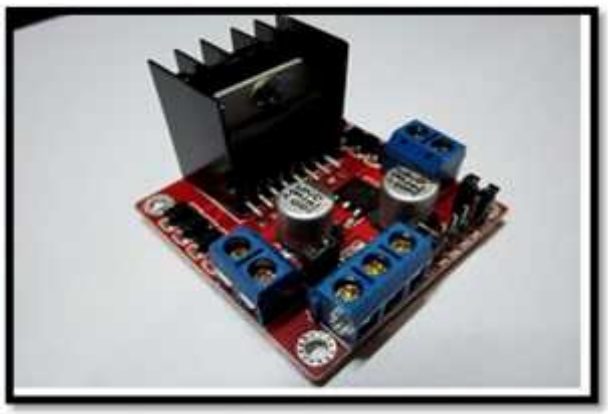


Figure 10 : L298N Programming Dual H-Bridge Stepper / DC Motor Driver Module

5- Arduino Language :

5-1 Arduino is programmed with C and C++:

C++ can be considered as a superset of C. It means C++ brings new concepts and elements to C. Basically, C++ can be defined as C with object-oriented implementation, which is a higher-level feature. This is a very nice feature that brings and provides new ways of design.

We'll enter together into this concept a bit later in this book but basically, in object-oriented programs, you define structures called **classes** that are a kind of a model, and you create objects called **instances** of those classes, which have their own life at runtime and which respect and inherit the structure of the class from which they came.[16]

Object-oriented programming (**OOP**) provides four properties that are very useful and interesting:

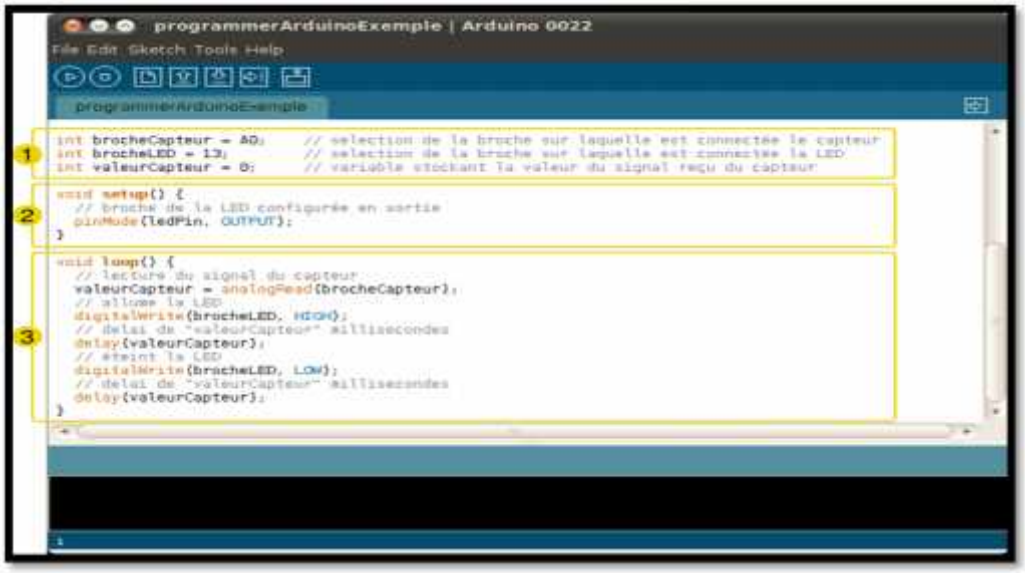
- ❖ Inheritance (classes can inherit attributes and behaviors from their parent classes).
- ❖ Data encapsulation (each instance retains its data and functions).
- ❖ Object identity (each instance is an individual).
- ❖ Polymorphism (each behavior can depend on the context).

In OOP, we define classes first and then we use specific functions called **constructors** to create instances of those classes. Imagine that a class is a map of a type of house, and the instances are all the houses built according to the map.

Almost all Arduino libraries are made using C++ in order to be easily reusable, Which is one of the most important qualities in programming.[16]

5-2 The structure of a program:

An Arduino program has three parts:



1. The declaration of the variables (optional).
2. Part initialization and configuration of inputs / outputs: the setup () function.
3. The main part that runs in a loop: the loop () function.[16]

6- Conclusion

The Arduino UNO is the most common cards. This is the first of its kind. However, there are other versions of Arduino boards more suited for certain projects.

In our project we use arduino uno and another extension (Ultrasonic, HC-06 Serial Port Bluetooth, L298N Dual H-Bridge),so now we have general idea about our project and we assemblate all this composent with programming it to make robot avoiding obstacle so that what we talking about in chapiter three.

Chapter III

Design And Software Implementation

1-introduction

2- what we need to building

Our project

3-first step

4- Schema of our robot Care

5- the Result of our Robot Care

6- Source code of our project

7-What is MIT App Inventor ?

8- Conclusion

1- Introduction

This chapter is the important chapter in our project because in beginning we was have simple idea about what we must to do in our project, but when we was working on it we see our project is the big more than what we imagine and now we make it in real with big obstacles because shortage source of material ,so in this chapter we will introduce our project (Robot avoiding obstacle) with option to control a Robot with application of system android.

2- what we need to building our project (Robot avoiding obstacle) ?

- ❖ Arduino UNO.
- ❖ Ultrasonic .
- ❖ H-Bridge L298N.
- ❖ Module Bluetooth.
- ❖ Jumper .
- ❖ Two DC Motors .
- ❖ Smartphone with System Android .

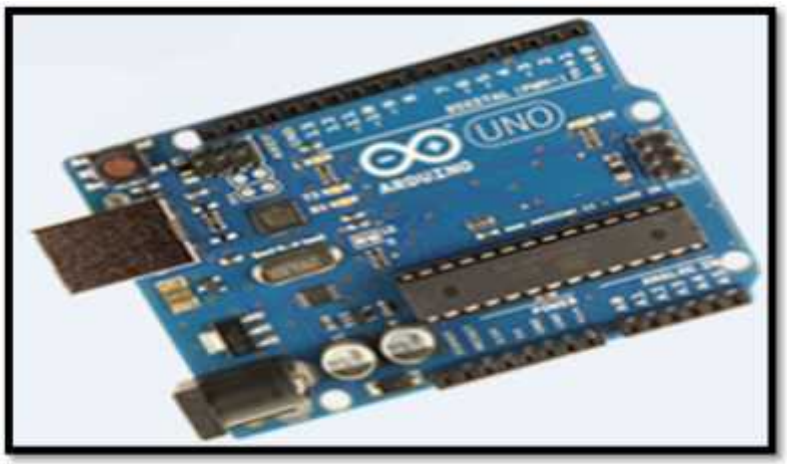


Figure 11 : Arduino Uno

Figure 12 : Ultrasonic ranging module HC - SR04

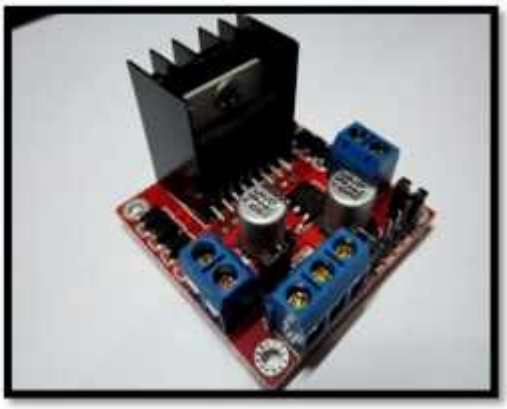


Figure 13 :H-Bridge L298N

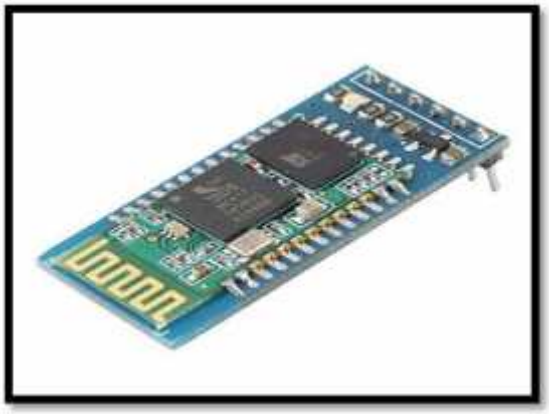


Figure 14: Module Bluetooth.

Figure 15 :jumper.

Figure 16 :Two DC Motors.

❖ **Application System Android To Controlled Robot .**

Figure 17: Application system android to controlled Robot

3- First step (testing ultrasonic)

- Source code for Ultrasonic to get & Print Distance.

- This is Result after uploading a source code and we Testing Ultrasonic.

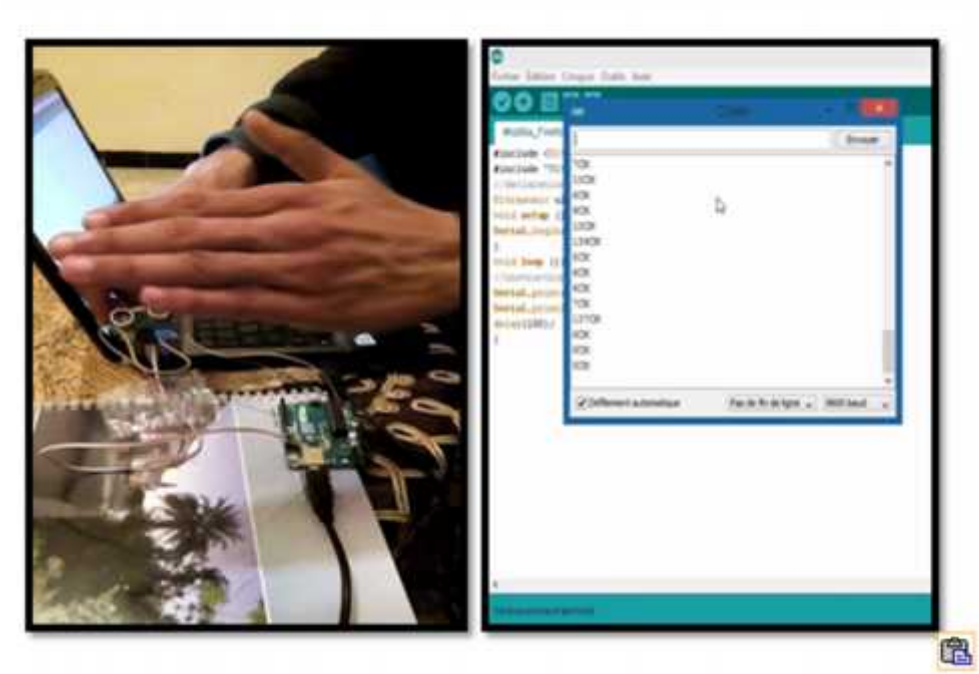


Figure 17:Result of Test Ultrasonic .

4- Schema of our robot Care

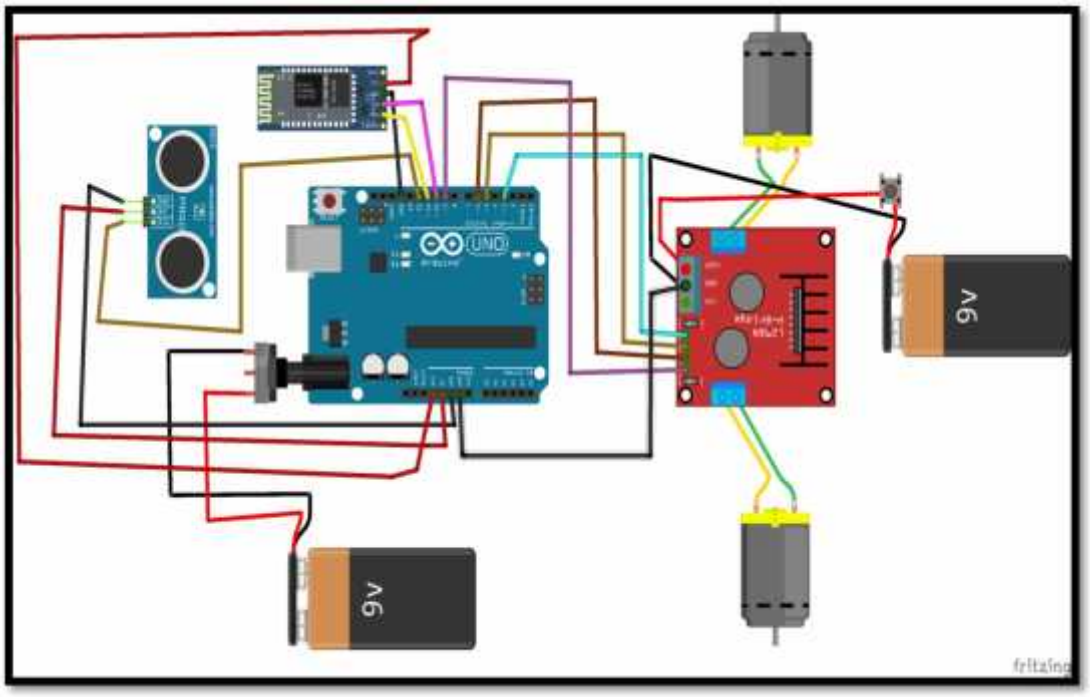


Figure 18 :Schema of robot

- We wired all components with jumper and put two power (9V) first one for h-bridge L298N And second for arduino to power ultrasonic and bluetooth .

5- The Result of our Robot Care

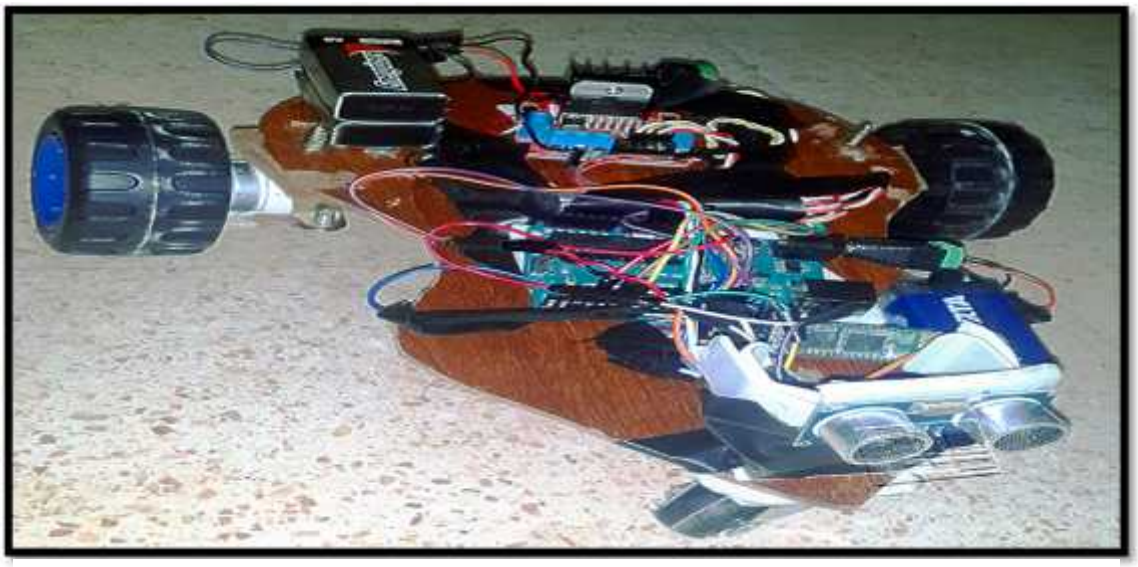


Figure 9 :The Robot care

6- Source code of our project

- **This is first part of code** :

Figure 10 :source code 1

1- Include library to our source code .

2- Declaration and define pin of ultrasonic and Bluetooth & other variable.

- **This source code contains two fonction :**
  Fonction scan : for Ultrasonic ranging module HC - SR04.

  Fonction Movebackward : for direction robot and controlling speed of motors.

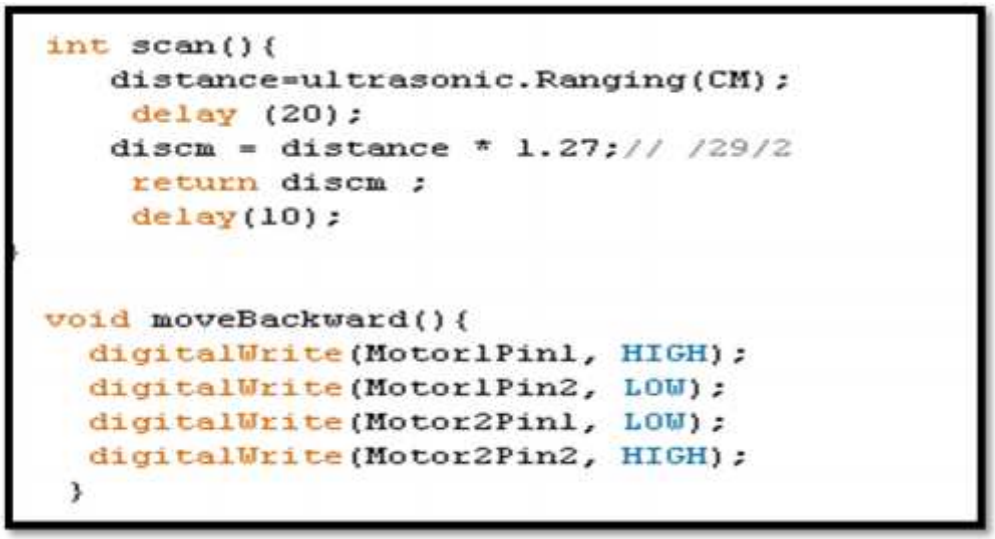


Figure 11 :source code 2

About instruction « digitalwrite (motor1Pin1,HIGH) ; » .

- **Syntax :** DigitalWrite(pin, value) ;
- **Parameters:** Pin: the pin number.
- **Value:** HIGH or LOW.

- **This part of source code contains one of principal part in langage arduino c :**

```
//initialization all components and all pins
void setup() {
  BT.begin(9600);
  Serial.begin(9600);
  pinMode(3, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(9, OUTPUT);
  }
```

Figure 12 :source code 3

Responsibility of ‘’**Void setup**’’ is initialization pin to put it output or input and set speed of communication bit per second by instruction « serial.begin(9600) ; ».

- **This part of source code contains the main of program**

```
void loop() {
 blu();
 if(cond == true){
  //check if arduino receive data
 if (readdata.length() > 0) {
   Serial.println(readdata);//print data after Reception
   //if arduino Receive word 'forward' from app android to do fonction forward.
  if(readdata == "forward")
 {
   moveForward();
   delay(100);
 }
```

Figure 13 : source code 4

Void loop() : is the main program and this function to allow program to change and respond, and this loop is function (repeat).

SECOND PART OF OUR PROJECT

In the second part of our project we discusses about **« Control robot with system android by bluetooth »**,We choose **''Android''** and we use bluetooth because its most popular system According to statistics of applications android it used more than other application As it shown in the following vote :

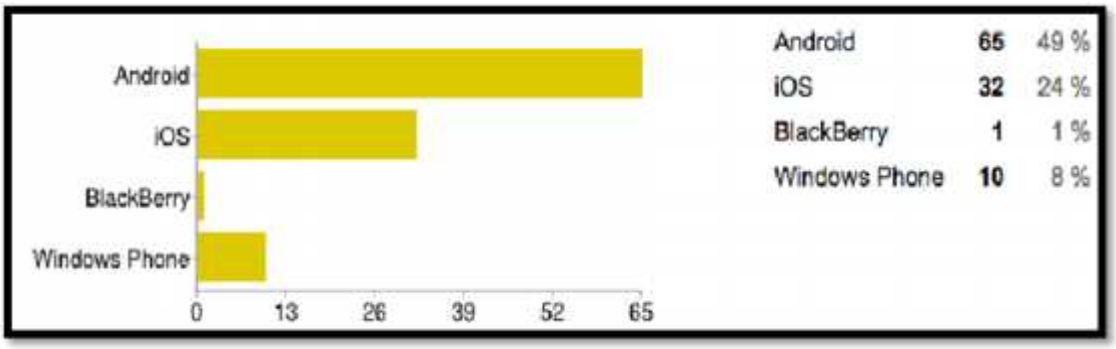


Figure 14 ;poll of use android application and other system By mathieu[18]

26-01-2015

7- What is MIT App Inventor ?

The App Inventor project was led by Hal Abelson, the Class of 1922 Professor of Computer Science and Engineering, who spent a sabbatical year at Google as a visiting professor. Instead of having to write traditional computer code, users of App Inventor can create programs by snapping together virtual, color-coded instruction “blocks.” For instance, to add a button to an application, the user would drag the button block into App Inventor’s workspace window and determine the button’s visual properties by selecting from pull-down menus. Then, to determine what the button will do, the user would snap a block that defines a function — like emitting a noise, or making a phone call, or changing the screen’s background color — into the button block. [17]

7-1 web site of MIT APP INVENTOR

Figure 15 :web site of MIT APP INVENTOR

When we click to Orange Button « create ! » we in to software Mit App Inventor in browser and it ask you to sign in or create new email in gmail to going to next step in software .

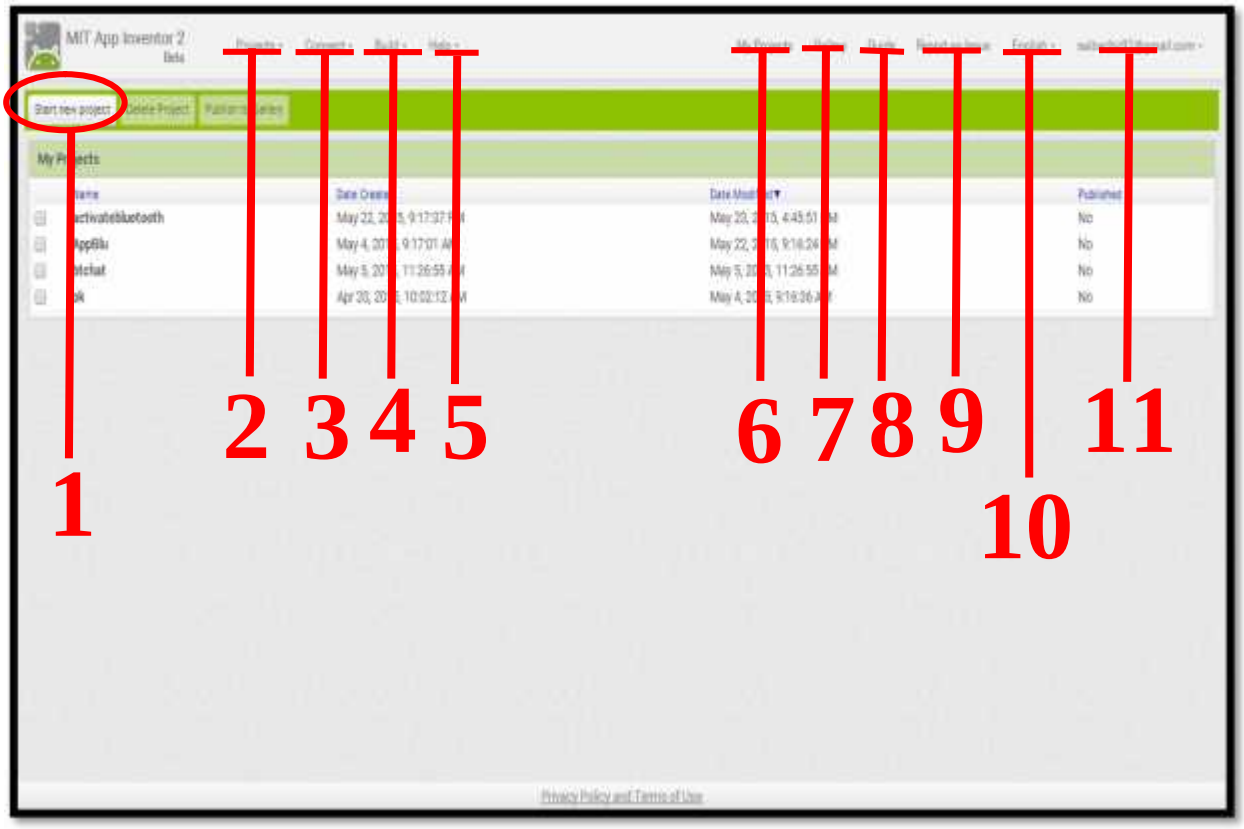


Figure 16 :Interface for App inventor

.

1- To start new project .
2- To import , save , save as , start new project … etc .
3- To connect to your device phone or to enable emulator …etc .
4- Building your application as apk to try it on your phone.
5- Get help from site like how it work … etc .
6- To see your project .
7- To see other application .
8- Guide .
9- 'Rapport an Issue' put if you want rapport in group google .
10- Select your prefer language .
11- Your adresse email .

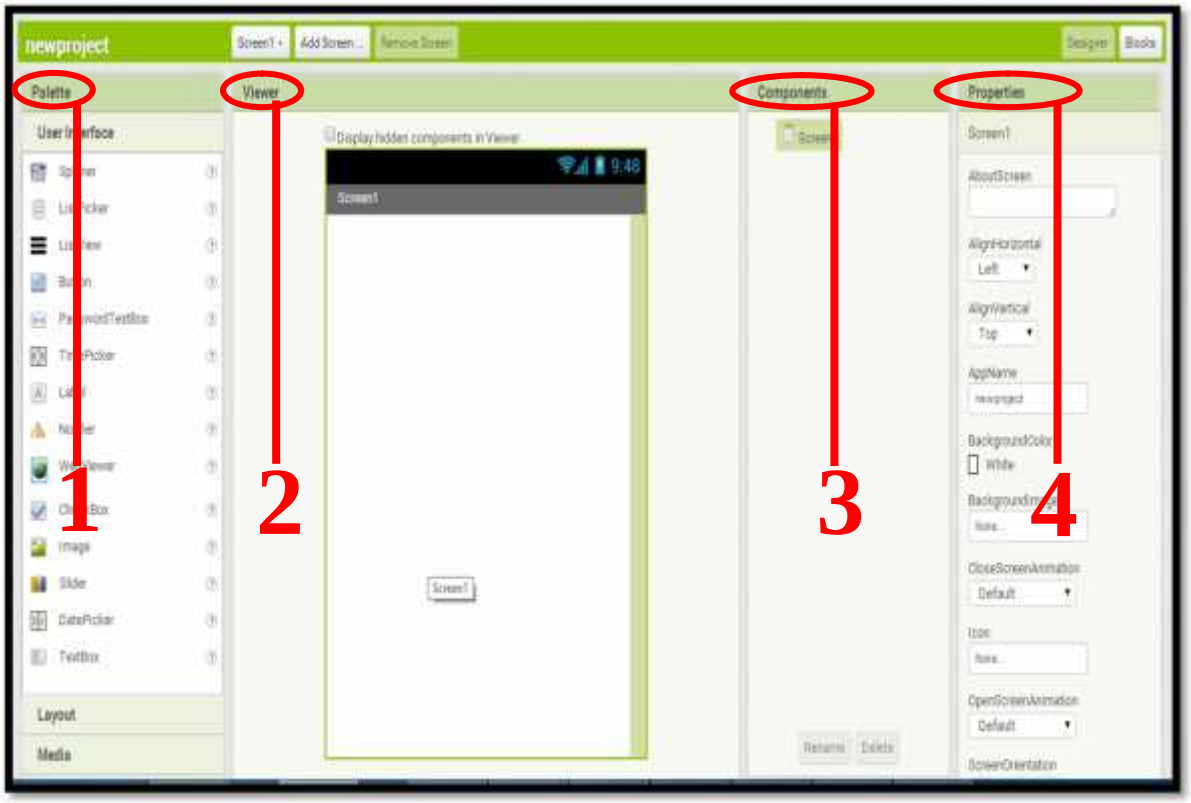


Figure 17 :User interface (designer)

1- Choose components .

2- Arrange components .

3- Componenets list heirarchical view .

4- Change components setting .

And this first step to start application and create design like you want in middel screan after creating design we going to next step Block to programme application and we find all components, button, label, listpicker … etc .

8- Description design of application

In the first we put label to show to us condition of bluetooth if connect or not , in second we put button to connect with robot by bluetooth , after that in middle we put five buttons to control direction of our robot (forward,backward,right,left ,and stop) , in bottom of application we put too other label to print situation of robot when we click to some button of direction .

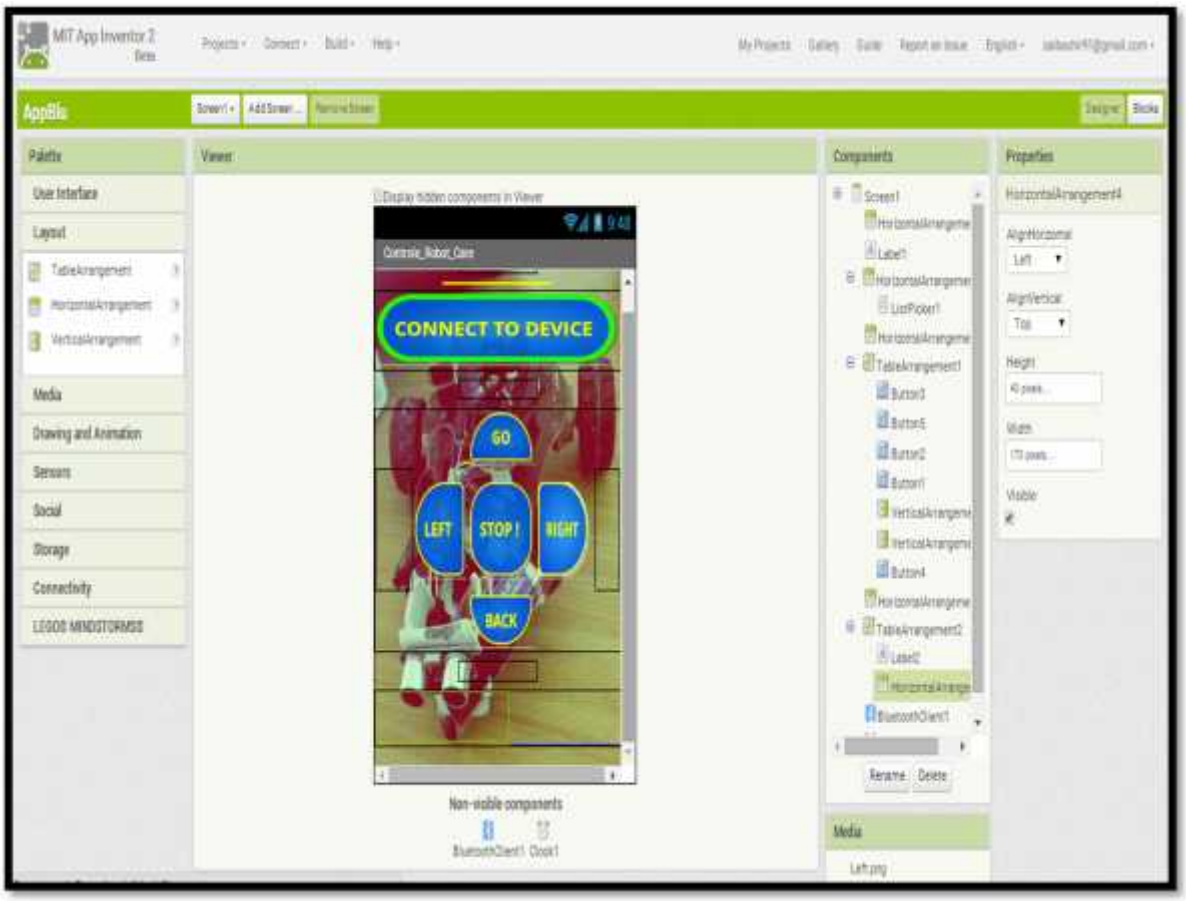


Figure 18 :Application Designe

.

8-1 description of block (code source)

- **First block in application**

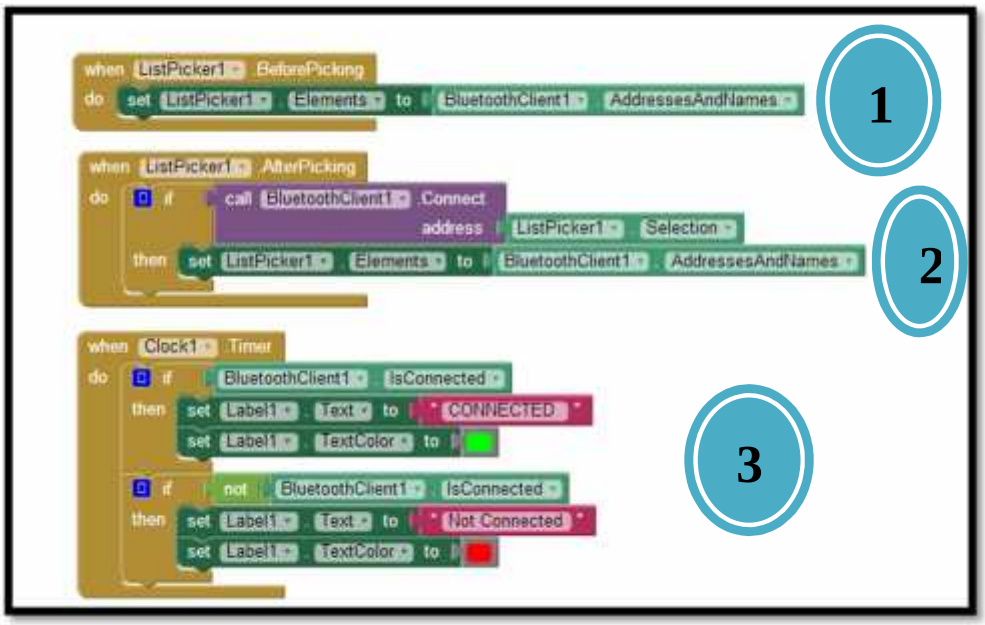


Figure 30 : application block 1

1- Listpicker 1 before Packing show to us place of list Device .

2- List picker 1 after picking it show to us list of devices and let us to selection.

3- Clock 1 it's timer it means every 1 seconds check if bluetooth on and write connect or not connect .

- **Second Block in application**

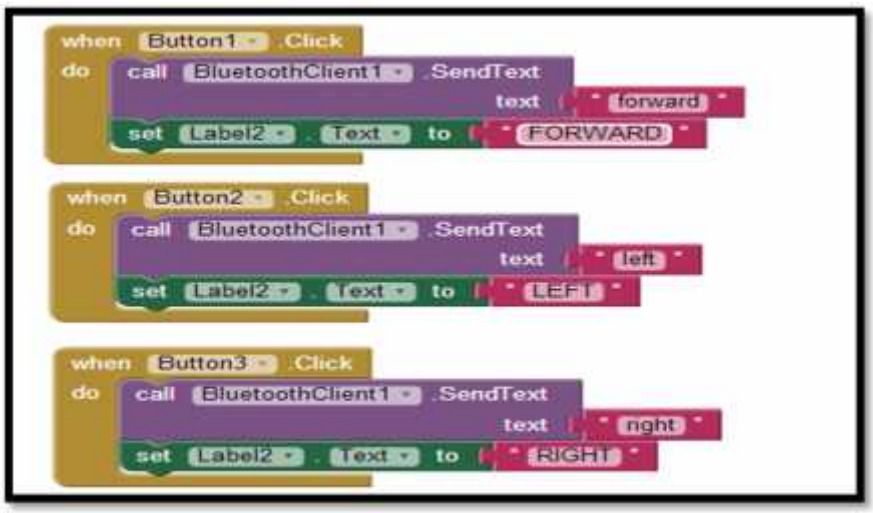


Figure 19 :application block 2

This block special for five buttons and every button with one direction « forward, backward , left, right , stop » ,for example when we click on button 1 call bluetooth to send text to arduino for controlling direction of robot.

9- Conclusion

Finely In this chapter, we discussed the implementation of our project consists of several sub-parts, and we presented our obstacle avoidance robot project. And also we discussed how to building application android to controle our robot with bluetooth :

- First part we explained all the components that we used .
- In the second we explained how we making our robot (first part of project).
- And in third we discussesd about second aim of project (why, what, and how)to building our application and finely th result of our robot .

General Conclusion

We can say at the conclusion of the project , we have an independent robot can avoid obstacles with the option to control by the application in the Android system yet and We have achieved our aims in our project and discussed that in three seasons with getting the perfect result as follows :

In Chapter one we discussed history of arduino, who created arduino and why arduino like we talking about some project of arduino so we can say we have general look about board arduino .

In chapter two by virtue of conception of some components like "board arduino uno ultrasonic module H-bridge L298N , Module bluetooth "and after all that description we can say now we have become know many information about microcontroller arduino and how to building autonomous robot because we know all what we need of components .

In the last section of our project we present our autonomous robot and how we able to building it ,we discussed how we wired it by schema of all component we used we also show how we programmed microcontroller arduino uno to avoiding obstacles and we discussed second aim , is control robot with application android and using module bluetooth to communication between robot and smartphone (system android) with transmission and receiving information to control direction and movement (speed) of robot ,we also talking about application arduino and, why we chosen system android and how to building application with Mit app inventor, it's so recent software to building applications android, and after all this we can say we got success in our project

Eventually we want to develop this project in future to do many important mission :

- robot going to some place or station you choose it without control .

- software programed in arduino for getting us path where car going .

- robot to detect mines and showing to us a safety path ... etc.

And we hope if someone of students and decide to choose our project and he can starts where we stopped to develop project and get more useful .

Reference

[01]. - **fayaz shaldib MD wali ullah bhuiyn,Md kamrul hasan,hasan mahmud** international journal of computer applications [Revue]. - [s.l.] : IJCA journal, 2013. - 9 : Vol. 66.

[02]. - 15 02 2015. - http://www.circuitstoday.com/story-and-history-of-development-of-arduino 26/03/2014 john.

[03]. - 17 02 2015. - http://arduino.cc/en/Products.Compare.

[04]. - 20 09 2015. - http://arduino.cc/en/Main/ArduinoShields .

[05]. - 21 02 2015. - http://store.arduino.cc/category/2.

[06]. - 28 02 2015. - http://www.arduino.cc/en/Guide/Environment..

[07]. - 05 03 2015. - http://arduino.cc/en/main/software .

[08]. - 07 03 2015. - http://www.robotix.in/tutorials/categ/eventtutes/stepup.

[09]. - 08 03 2015. - http://www.arduino.cc/en/pmwiki.php?n=Main/HomePage. .

[10]. - pdf. - 10 03 2015. - http://docs-asia.electrocomponents.com/webdocs/0e8b/0900766b80e8ba21.pdf.

[11]. - 23 03 2015. - http://arduino.cc/en/Main/ArduinoBoardUno.

[12]. - pdf. - 25 04 2015. - http://www.lakos.fs.uni-lj.si/images/Predmeti/MK/2014/C%20Programming%20for%20Arduino.pdf.

[13]. - pdf. - 25 04 2015. - http://www.elecfreaks.com/store/download/product/Sensor/HC-SR04/HC-SR04_Ultrasonic_Module_User_Guide.pdf.

[14]. - pdf. - 27 04 2015. - ftp://imall.iteadstudio.com/Electronic_Brick/IM120710006/DS_IM120710006.pdf.

[15]. - pdf. - 02 05 2015. - http://www.elecdesignworks.com/index.php/product-details-77?tmpl=component&format=pdf&iframe=true&width=800&height=550.

[16]. - pdf. - 11 05 2015. - http://www.dsc.ufcg.edu.br/~rangel/Beginning%20C%20For%20Arduino%20(2012).pdf.

[17]. - 20 05 2015. - http://newsoffice.mit.edu/2010/android-abelson-0819 .

[18]. - 21 05 2015. - http://rennes24.info/2015/01/.