



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta strojní



Control System of Vehicle for Smart Factory Model with Principles of Industry 4.0.

Diplomová práce

Studijní program: N2301 – Mechanical Engineering
Studijní obor: 2302T010 – Machines and Equipment Design
Autor práce: **Anishkumar Nalanagula**
Vedoucí práce: Ing. Radek Votrubec, Ph.D.





TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechanical Engineering ■

Control System of Vehicle for Smart Factory Model with Principles of Industry 4.0.

Master thesis

Study programme: N2301 – Mechanical Engineering
Study branch: 2302T010 – Machines and Equipment Design
Author: **Anishkumar Nalanagula**
Supervisor: Ing. Radek Votrubec, Ph.D.



DIPLOMA THESIS ASSIGNMENT

(PROJECT, ART WORK, ART PERFORMANCE)

First name and surname: **Anishkumar Nalanagula**
Study program: **N2301 Mechanical Engineering**
Identification number: **S16000467**
Specialization: **Machines and Equipment Design**
Topic name: **Control System of Vehicle for Smart Factory Model with Principles of Industry 4.0.**
Assigning department: **Department of Manufacturing Systems and Automation**

R u l e s f o r e l a b o r a t i o n :

- 1) Meet Arduino development board, its accessories and its shields.
- 2) Assemble a vehicle for smart factory model controlled by Arduino which is able to transport small subjects.
- 3) Invent a system to control its position and equip it with appropriate sensors. The vehicle has to be able to communicate with other members of the factory via wifi.
- 4) Write an article with your supervisor about developed theme.

Scope of graphic works: **as required**
Scope of work report
(scope of dissertation): **50-60**
Form of dissertation elaboration: **printed/electronical**
Language of dissertation elaboration: **English**
List of specialized literature:

[1] **BEQUETTE, B. Process control: modeling, design, and simulation. Upper Saddle River, N.J.: Prentice Hall PTR, 2003. ISBN 0133536408.**

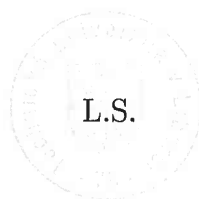
[2] **Arduino Learning: Getting Started with Arduino. In: Arduino [online]. 2014 [cit. 2015-01-09]. available from: <http://arduino.cc/en/Guide/HomePage>**

Tutor for dissertation: **Ing. Radek Votrubec, Ph.D.**
Department of Manufacturing Systems and
Automation

Date of dissertation assignment: **15 November 2017**

Date of dissertation submission: **15 May 2019**


prof. Dr. Ing. Petr Lenfeld
Dean



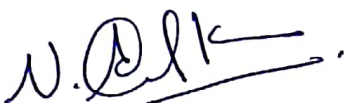

Ing. Petr Zelensky, Ph.D.
Head of Department

Liberec, dated: 15 November 2017

Declaration

I hereby certify that I have been informed the Act 121/2000, the Copyright Act of the Czech Republic, namely § 60 - Schoolwork, applies to my master thesis in full scope. I acknowledge that the Technical University of Liberec (TUL) does not infringe my copyrights by using my master thesis for TUL's internal purposes. I am aware of my obligation to inform TUL on having used or licensed to use my master thesis; in such a case TUL may require compensation of costs spent on creating the work at up to their actual amount. I have written my master thesis myself using literature listed therein and consulting it with my thesis supervisor and my tutor. Concurrently I confirm that the printed version of my master thesis is coincident with an electronic version, inserted into the IS STAG.

Date: 21/05/2018

Signature: 

Acknowledgments

It is with immense gratitude that I acknowledge the support of my University for providing me this great opportunity to develop in deepest manner my engineering skills while accomplishing this diploma thesis and help them to further develop the importance in machine construction design.

I am indebted firstly to thank especially **Ing. Radek Votrubec, Ph.D** for his professional guidance, encouragement and good advices all along. This thesis becomes much better because of him. Thanks to his supervision and support.

I would like to acknowledge **Ing. Petr Zelený, Ph.D** for his support in this work and helping me to develop the concept of it.

My Special thanks to **Ing. Maryna Garan** for helping me all the possible way with my thesis.

Finally, I would like to thank my staffs, family and friends for all their invaluable support.

Abstract

The control system of a smart factory involves highly sophisticated functions. This project aims at making an accurate control system for a smart factory vehicle. The smart factory vehicle is aimed at communicating with the other members of the smart factory using Wi-Fi communication technique. This targets to eliminate any human interference which was earlier present in the evolutions of the generations of the factory. Therefore it ensures there is no deviation from target which may arise due to human negligence, fatigue or performance efficiency of an individual. The smart factory vehicle will be navigating in a controlled atmosphere in a smart factory and the purpose of the vehicle is to carry small subjects from one place to another as per requirements of the factory. The vehicle will be equipped with various sensors aiming at automated guiding of the vehicle. There are certain desired positions where the vehicle will be expected to stop and this function of the vehicle can be achieved with the help of the colour sensor which recognizes colours. The colour sensor in this smart factory vehicle will be made to recognize colours namely red and blue. It is also equipped with a single IR sensor which will be used to stop at desired location. Hence with all the functions, it is possible to achieve a sophisticated control system for the smart factory vehicle.

Abstrakt

Řídicí systém inteligentní továrny zahrnuje vysoce sofistikované funkce. Cílem tohoto projektu je vytvořit přesný řídicí systém pro inteligentní tovární vozidlo. Inteligentní tovární vozidlo komunikuje s ostatními členy inteligentní továrny pomocí komunikační technologie wifi. Cílem je eliminovat veškeré lidské zásahy do výroby, které se dříve objevovaly v továrnách minulých generací. Proto tato koncepce zajišťuje, že nedojde k žádné odchylce od cílů, která by mohla vzniknout z důvodu nedbalosti člověka, únavy nebo výkonnosti jednotlivce. Inteligentní tovární automobil se bude pohybovat v prostředí inteligentní továrny a účelem vozu je převážet malé předměty z jednoho místa na druhé podle zadaných požadavků. Vozidlo bude vybaveno různými senzory zaměřenými na automatické řízení vozidla. Pomocí snímače barev v kombinaci s IR snímačem bude vozidlo umět zastavit na požadovaném místě. Snímač barev v této inteligentní továrně bude schopen rozpoznat tři barvy, jmenovitě červenou, modrou a zelenou. Je také vybaven jediným IR snímačem, který bude použit pro zastavení na požadovaném místě. Se všemi funkcemi vozidla a komunikací mezi různými členy inteligentní továrny je možné dosáhnout důmyslného řídicího systému pro inteligentní tovární vozidlo.

Contents

LIST OF TABLES	4
LIST OF FIGURES	5
1 - INTRODUCTION.....	6
2.1- CONCEPT OF THE SMART FACTORY VEHICLE	7
2 - WHAT IS INDUSTRY 4.0?.....	9
2.1 - ORIGINS OF INDUSTRY 4.0.....	9
2.1.1-Industry 1.0.....	10
2.1.2-Industry 2.0.....	10
2.1.3-Industry 3.0.....	11
2.1.4 -Industry 4.0.....	11
2.1.5 Smart Industry.....	13
2.1.6 Smart factory.....	13
3 - COMPONENTS OF VEHICLE.....	15
3.1 - ARDUINO H-BRIDGE MODULE L9110S	18
3.2 - 5 CHANNEL TRACKING SENSOR MODULE BOARD	20
3.3 - RGB LED DIODE.....	23
3.4 - GRAPHIC LCD DISPLAY ST7920	25
3.5 – ARDUINO COLOR DETECTOR	27
3.6 - WI-FI MODULE	30
3.7 - RFID CHIP+READER.....	32
3.8- SINGLE IR SENSOR.....	35
3.9 - ARDUINO MEGA 2560.....	38
3.9.1 Technical specifications	38
3.8.2 – Power.....	39
3.8.3 – Programming	40
3.8.4 - Automatic Reset	41
3.8.5 - USB overcurrent protection.....	41
4-FUNCTIONS OF THE SMART FACTORY VEHICLE.....	41
4.1- LINE FOLLOWING.....	43
4.1.1 Straight direction	45
TUL - Department of Manufacturing Systems and Automation	2

4.1.2 Right curve.....	45
4.1.3 Left curve	45
4.1.4 Sharp right curve.....	45
4.1.5 Sharp left curve	45
4.1.6 Demerging curves	46
4.1.7 Merging curves	46
4.2- DECISION MAKING ON LINE TRACKING	46
4.2.1 Automated path selection.....	47
4.2.2 - On command path selection	48
4.3- WI-FI COMMUNICATION BETWEEN ARDUINOS	49
4.3.1 - When the vehicle is powered on.....	49
4.3.2 - When the vehicle is tracking.....	50
4.4- COLOUR DETECTION.....	50
4.5- STOPPING AT DELIVERY POINT	52
4.6- RGB LED INDICATION.....	53
4.6 – RFID READING	53
5- ARDUINO CODE.....	54
6- CONCLUSION	61
7- REFERENCES	62

List of Tables

Table 1- Pins of H-module -----	19
Table 2 - Sample code for H-bridge-----	20
Table 3- Pins of 5 channel tracking sensor -----	21
Table 4- Sample code for Line tracking sensor -----	23
Table 5 - Sample code for RGB LED diode-----	25
Table 6 - Table for wiring of Graphic LCD -----	26
Table 7 - Sample code for Graphic LCD -----	26
Table 8- Sample code Arduino colour detector -----	29
Table 9- Sample code for Wi-Fi module (Transmitter) -----	31
Table 10 - Sample code for Wi-Fi module (Reciever) -----	32
Table 11 - Sample code for RFID chip+reader-----	34
Table 12- Sample code for Single IR-----	38
Table 13 - Technical specifications of Arduino Mega 2560-----	39
Table 14- Values of a colour sensor-----	51

List of figures

Figure 1- Scheme of the smart factory.....	6
Figure 2 – Smart factory vehicle.....	8
Figure 3 - Imaginary Assembly line of industry 4.0.....	9
Figure 4- Evolution of Industry 4.0	10
Figure 5- List of components of robot vehicle chassis	15
Figure 6 - First step of construction of vehicle chassis	16
Figure 7 - Second step of Construction of vehicle chassis.....	16
Figure 8 - Third step of construction of vehicle chassis	17
Figure 9- Fourth step of construction of vehicle chassis.....	17
Figure 10- Total assembly of vehicle chassis.....	18
Figure 11 - H-bridge module	18
Figure 12- 5Channel tracking sensor module.....	20
Figure 13- RGB LED diode.....	23
Figure 14 - Wiring diagram of RGB LED diode.....	24
Figure 15- Graphic LCD display	25
Figure 16- Arduino colour detector	27
Figure 17- Wiring diagram of Arduino colour detector.....	28
Figure 18 - Wi-Fi module.....	30
Figure 19 - Wiring diagram of Wi-Fi module	31
Figure 20- RFID Chip+reader	32
Figure 21- Wiring diagram of RFID chip+reader.....	33
Figure 22- Single IR sensor.....	35
Figure 23- Wiring diagram for single IR sensor.....	37
Figure 24 – Arduino Mega2560.....	40
Figure 25 – Track used for navigation	44
Figure 26 – Decision making- type I	47
Figure 27- Decision making – type II	48
Figure 28- Stopping at red colour	50
Figure 29- Stopping at blue colour	51
Figure 30- Stopping at delivery point	52

1 - Introduction

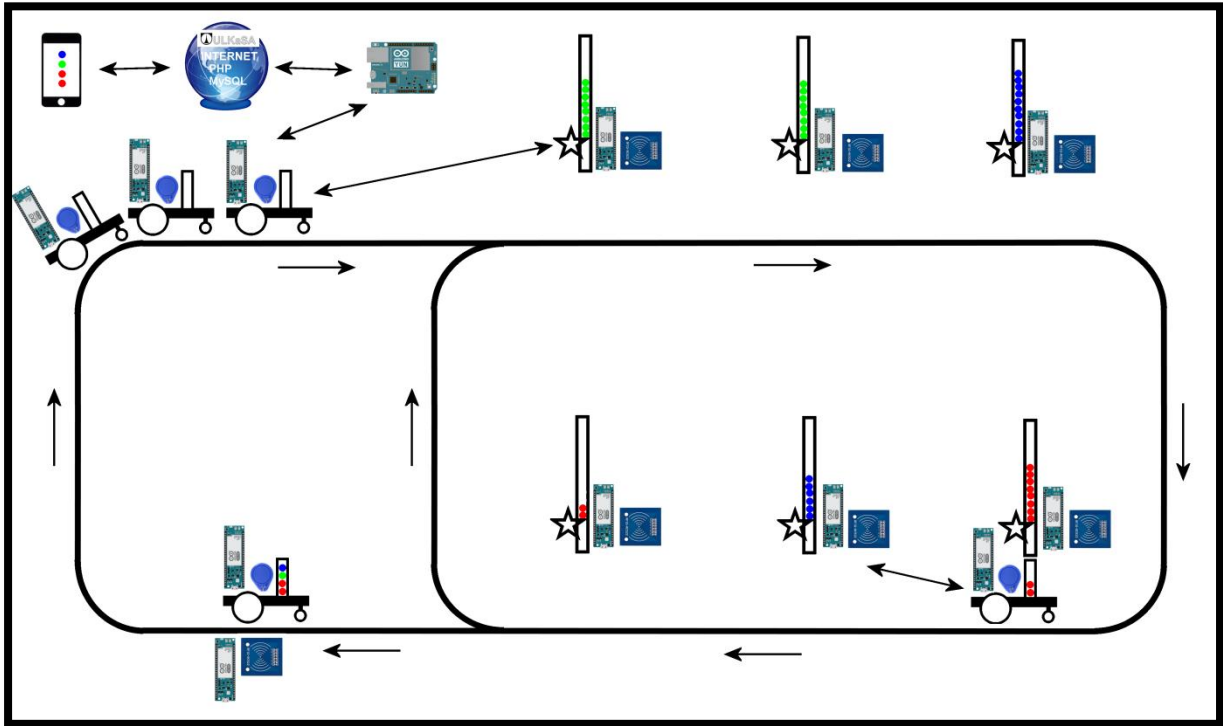


Figure 1- Scheme of the smart factory

The scheme of the smart factory is explained diagrammatically in the picture shown above. The smart factory is based on Industry 4.0 concept. This is an idea of automation which can be realized in large factories. This is an alternative to labor work. It can replace repetitive hard work and also reduces the chance of accidents as it eliminates any human interference in the work area. The work area is also customizable as per factory requirements. It can be different for different factories and this helps in easy implementation.

The idea of the automation can be extended or shortened based on the requirement modifications of the factory. The main function of this smart factory is transportation of goods from one place to another. There are six provisions in the smart factory where goods are delivered from the provision to the vehicle. There are five vehicles where they go around the track collecting goods from the provision.

Each Vehicle receives the information on the combination of goods to be collected from the provisions and starts the line following. This information is received by the vehicle through a Wi-Fi module on board the vehicle. The vehicle also receives the information on

the next available station. The process starts by the vehicle entering the inner loop. After receiving the combination of goods, it starts to travel in the loop and continues to travel on the inner loop until all the goods are on board. Once it is finished, the vehicle leaves the loop and travels to the desired position where the goods have to be delivered. To enable smooth operation of the smart factory, various modules have been used which are discussed in detail in the document.

A part of the smart factory has been taken for the thesis which includes the working of single smart factory vehicle on Wi-Fi. The information is sent on Wi-Fi from another arduino module which acts a transmitter.

This smart factory model is a solution for automation in the transportation of goods in the factory. This can be implemented in the future to enable uninterrupted work flow process. It has many advantages over the manual transportation which is making it more probable to be implemented in large factories where work load is high in the near future.

2.1- Concept of the smart factory vehicle

The smart factory vehicle has to perform several functions in the smart factory for the work flow to happen. The smart factory vehicle has to travel on the track provided in the smart factory. The track is split into two loops , inner loop and outer loop. The decision of inner loop or outer loop must be decided on Wi-Fi based on the work process. If the process of goods collection is finished, the vehicle has to exit the inner loop and join the outer loop and stop at the point where the goods are delivered.

As technology and cars become increasingly intertwined, the concept of smart manufacturing is jumping to the forefront of innovations in the auto industry. We are on the cusp of profound changes to the manner in which goods are manufactured, leading to fundamental modifications of the global competitive marketplace

There are two arduinos involved in this process. One arduino is on board the vehicle to receive information on the work flow. The other arduino is the transmitter of information whether the work is still pending or the vehicle can reach the delivery point.

The vehicle itself must be constructed sturdy with all the connections intact. The components have to be properly fastened and the wiring must be done according to the module under consideration. Loose contacts in wiring must be taken extra care as they may result in error in values. There are two Wi-Fi modules which are used for the communication process. One Wi-Fi module is on the vehicle which is the receiver. The other Wi-Fi module is on the vehicle which acts as the transmitter.

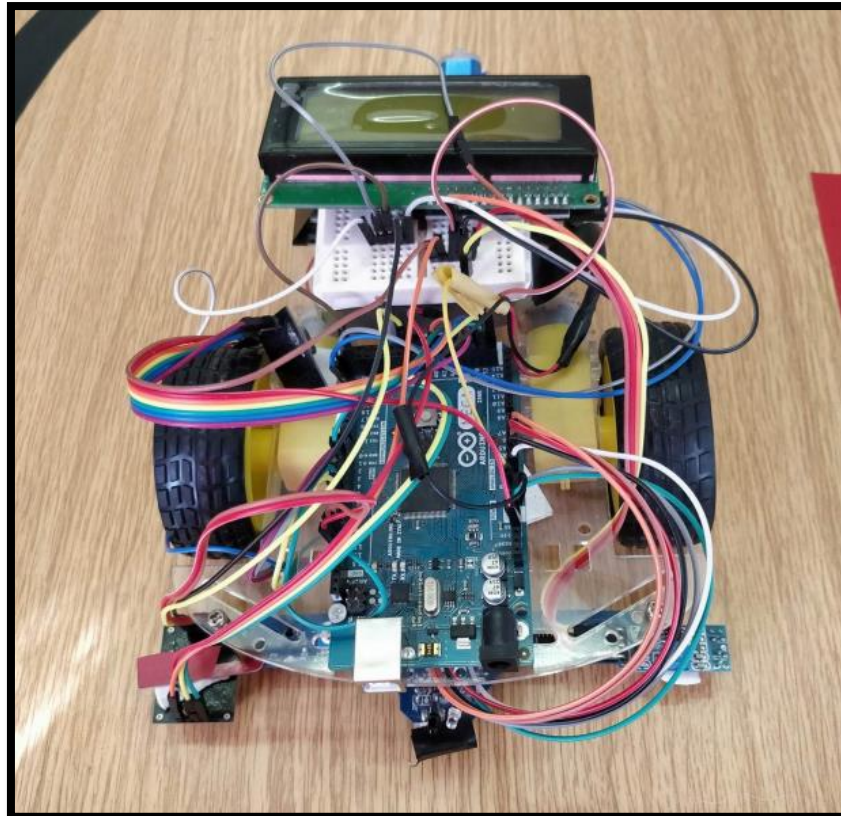


Figure 2 – Smart factory vehicle

The construction of the vehicle is shown step by step in the following pages. There are various components or modules used in the vehicle. The components such as tracking sensor, colour sensor, single IR Sensor should be mounted on the vehicle in a stable manner. The output values change according to the height at which they are placed from the ground level. There is an onboard power bank which supplies the input voltage required for the functioning of the modules. A bread board is used to enable wiring of all modules undisturbed.

2 - What is Industry 4.0?

Industry 4.0, The fourth industrial revolution is the digitization of manufacturing, influenced by technologies, like Data Analytics, Internet of Things; and convergence of Information Technology, robotics, data, Operational Technology and manufacturing processes to accomplish factories which are bonded, smart manufacturing, digital supply chain and self-optimizing systems in the cyber-physical space of the fourth industrial revolution which is information-driven.[1]

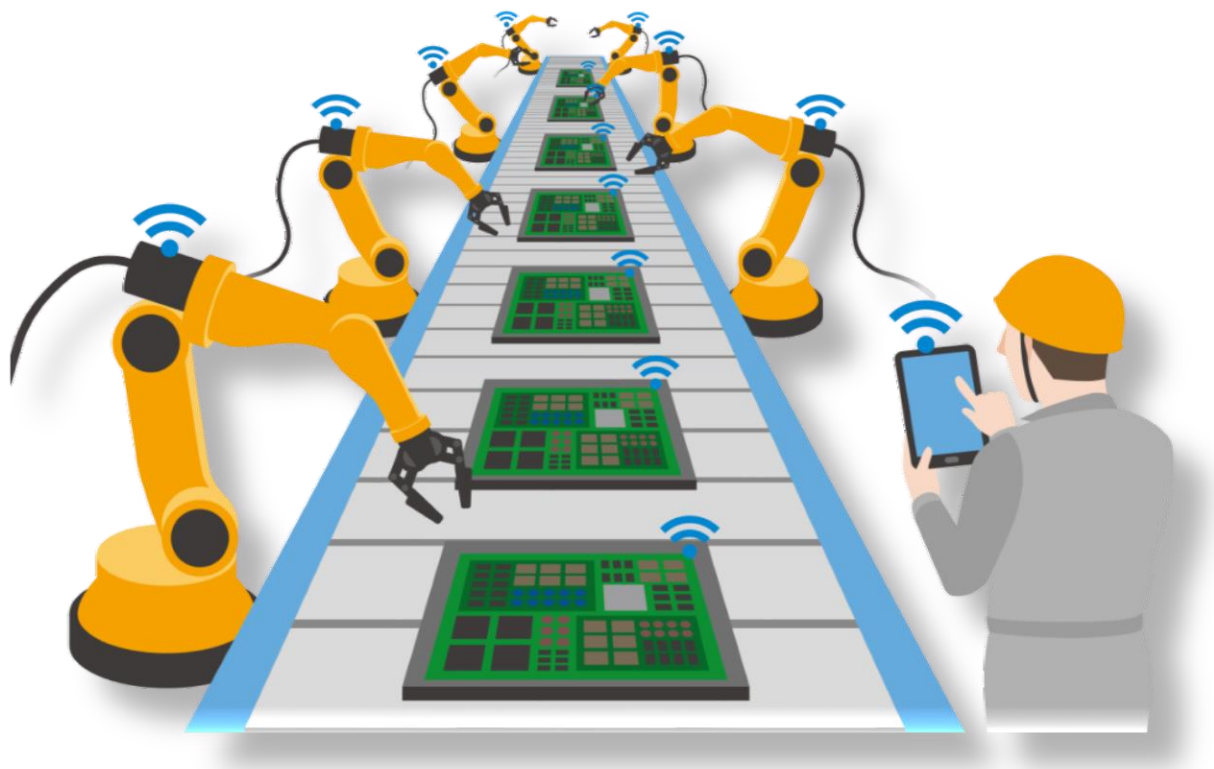


Figure 3 - Imaginary Assembly line of industry 4.0

2.1 - Origins of Industry 4.0

“Industrie 4.0- Smart manufacturing for the future” was launched by Germany at the Hanover Messe in 2011. The concept was to digitalize manufacturing which directly reduces the human interference in the process. This concept was launched by a group of Government officials, academics and industry leaders of Germany. The concept was far from reality and the officials who proposed the idea had no idea that it would become a widely used concept in the future. [2]

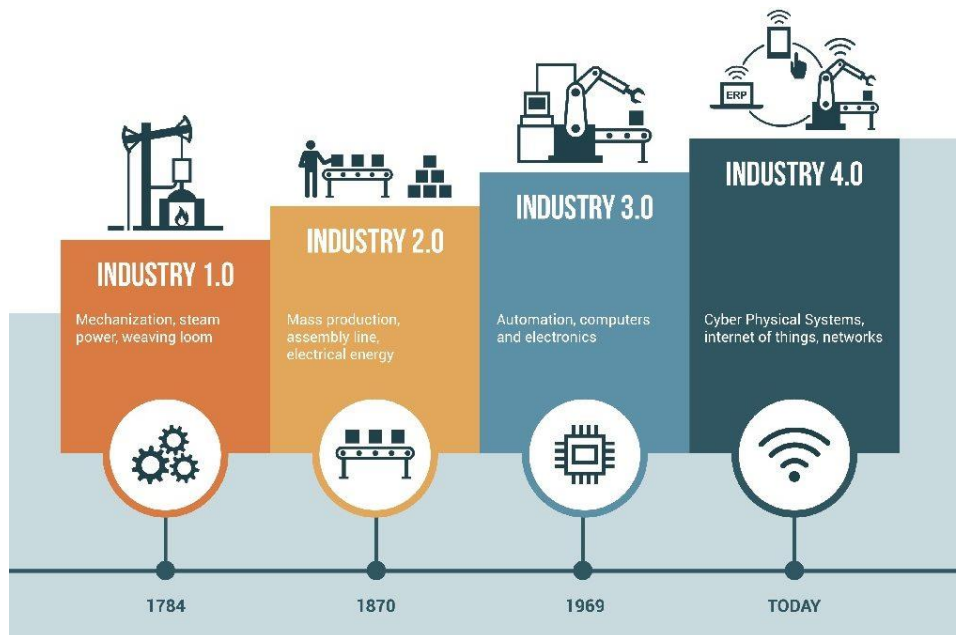


Figure 4- Evolution of Industry 4.0

2.1.1-Industry 1.0

1800s – Workers were aided by water or steam-powered machines. Owing to increased production facilities, business grew from individual cottage owners taking care of their own — and maybe their neighbour’s — needs to organizations with owners, managers and employees serving customers.[3]

2.1.2-Industry 2.0

The source of power was electricity at the start of 20th century. It was a good substitute for steam and water. It enabled the businesses to individualise power to machines. It led to the design of machines with individual power sources which helped portability of machines.

It was during this period, there were many management programs which concentrated on the increase of efficiency and effectiveness of manufacturing methods. Division of labor also was a concept that led to increase in productivity. Using assembly lines to gain high production became common. Finally, just-in-time and lean manufacturing principles qualified the manufacturing companies to improvise the output and quality.[3]

2.1.3-Industry 3.0

In the final few decades of the 19th century, the introduction of electronic devices, such as the transistor, integrated circuit (IC) chips, made it practically automate individual machines to supplement or replace operators more efficiently. This period also showed way for the development of software systems to improvise on the electronic hardware. Integrated systems like material requirements planning, were superseded by enterprise resources planning tools that enabled humans to plan, schedule and track product flows through the factory. Many manufacturers had to move component and assembly operations to low-expense countries due to pressure to reduce cost. This geographic dispersion resulted in the formalization of supply chain management.[3]

2.1.4 -Industry 4.0

In this century, Industry 4.0 unites the internet of things (IoT) and manufacturing techniques to guide intelligent actions by enabling systems to share information, analyze it and use it. It also incorporates cutting-edge technologies including additive manufacturing, robotics, artificial intelligence and other cognitive technologies, advanced materials, and augmented reality, according to the article “Industry 4.0 and Manufacturing Ecosystems” by Deloitte University Press.[3]

The programs which were developed during the later stages of the 20th century, such as manufacturing execution systems, shop floor control and product life cycle management, were concepts that far from implementation as it lacked the technology needed. Industry 4.0 can help achieve the full potential of these programs.[4]

INDUSTRIE 4.0 is the name given to the German strategic initiative to establish Germany as a lead market and provider of advanced manufacturing solutions. One of 10 “Future Projects” identified by the German government as part of its High-Tech Strategy 2020, the “INDUSTRIE 4.0” project represents a major opportunity for Germany to establish itself as an integrated industry lead market and provider. So-called because it will usher in a fourth industrial age, INDUSTRIE 4.0 is set to revolutionize manufacturing and production. INDUSTRIE 4.0 represents a paradigm shift from “centralized” to “decentralized” smart manufacturing and production.

“Smart production” becomes the norm in a world where intelligent ICT-based machines, systems and networks are capable of independently exchanging and responding to information to manage industrial production processes. This brave new world of decentralized, autonomous real-time production being pioneered in Germany has its basis in two things:

- Germany’s continued role as one of the world’s most competitive and innovative manufacturing industry sectors.
- Germany’s technological leadership in industrial production research and development.

Germany has the ideal conditions to become a global leader in innovative, internet-based production technology and service provision. Technological leadership and vision in the fields of manufacturing, automation and software-based embedded systems, as well as historically strong industrial networks, lay the cornerstone for the long-term success of the INDUSTRIE 4.0 project.

Today we stand on the cusp of a fourth industrial revolution. A revolution which promises to marry the worlds of production and network connectivity in an “Internet of Things” making “INDUSTRIE 4.0” a reality. The world as we know and experience it today has been shaped by three major technological revolutions. The first Industrial Revolution, beginning in the UK at the tail end of the 18th century and ending in the mid-19th century, represented a radical shift away from an agrarian economy to one defined by the introduction of mechanical production methods. The second period of radical transformation with the advent of industrial production and the birth of the factory at the start of the 20th century was no less precipitous; ushering in as it did an age of affordable consumer products for mass consumption.

In the late 1960s the use of electronics and IT in industrial processes opened the door to a new age of optimized and automated production. The Internet of Things and Services can now be added to the historical list of forces (mechanization, electricity, and information technology) powering industrial change. The changes being wrought by INDUSTRIE 4.0 will see it become the global language of production. INDUSTRIE 4.0 promises to increase manufacturing productivity levels by up to 50 percent – and halve the amount of resources

required. Germany's position as an embedded systems technology leader gives birth to enabling cyber-physical system (CPS) technologies which ingeniously marry the digital virtual world with the real world. Cyber-physical production systems (CPPS) made up of smart machines, logistics systems and production facilities allow peerless ICT-based integration for vertically integrated and networked manufacturing. Germany has set itself the goal of being an integrated industry lead market and provider by 2020.

2.1.5 Smart Industry

INDUSTRIE 4.0 opens the door to an age of "smart industry" in which people, devices, objects, and systems combine to form dynamic, self-organizing networks of production. Decentralized intelligence helps create intelligent object networking and independent process management, with the interaction of the real and virtual worlds representing a crucial new aspect of the manufacturing and production process. INDUSTRIE 4.0 represents a paradigm shift from "centralized" to "decentralized" production - made possible by technological advances which constitute a reversal of conventional production process logic. Simply put, this means that industrial production machinery no longer simply "processes" the product, but that the product communicates with the machinery to tell it exactly what to do.

Highly individualized, low-volume, real-time production becomes the norm as new industrial assistant systems allow new forms of machine-human production in a changing industry landscape. INDUSTRIE 4.0 connects innovative embedded system production technologies and smart production processes to pave the way to a new industrial age which will radically transform industry and production value chains and business models in tomorrow's smart factories.

2.1.6 Smart factory

The merging of the virtual and the physical worlds through cyber-physical systems and the resulting fusion of technical processes and business processes are leading the way to a new industrial age best defined by the "smart factory" concept. The deployment of cyber-physical systems in production systems gives birth to the "smart factory." Smart factory products, resources and processes are characterized by cyber-physical systems; providing significant real-time quality, time, resource, and cost advantages in comparison with classic

production systems. The smart factory is designed according to sustainable and service-oriented business practices. These insist upon adaptability, flexibility, self-adaptability and learning characteristics, fault tolerance, and risk management. High levels of automation come as standard in the smart factory - this being made possible by a flexible network of cyber-physical system-based production systems which, to a large extent, automatically oversee production processes.

Flexible production systems which are able to respond in almost real-time conditions allow in-house production processes to be radically optimized. Production advantages are not limited solely to one-off production conditions, but can also be optimized according to a global network of adaptive and self-organizing production units belonging to more than one operator. This represents a production revolution in terms of both innovation and cost and time savings and the creation of a "bottom-up" production value creation model whose networking capacity creates new and more market opportunities.

Smart factory production brings with it numerous advantages over conventional manufacture and production. These include: CPS-optimized production processes: smart factory "units" are able to determine and identify their field(s) of activity, configuration options and production conditions as well as communicate independently and wirelessly with other units. Optimized individual customer product manufacturing via intelligent compilation of ideal production system which factors account product properties, costs, logistics, security, reliability, time, and sustainability considerations; Resource efficient production; Tailored adjustments to the human workforce so that the machine adapts to the human work cycle. Smart factories, with their interfaces to smart mobility, smart logistics, and smart grids concepts, are an integral component of tomorrow's intelligent infrastructures.

3 - Components of vehicle

The following are the parts that comes into making an assembled robot vehicle chassis.










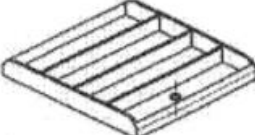
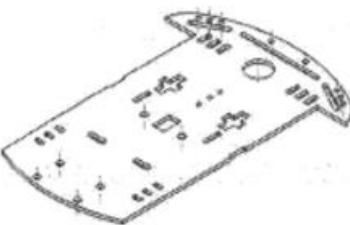
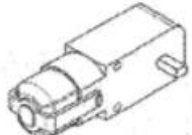
 M3 x 30 mm 4ks	 Sloupek M3 x 8 mm 4ks	 M3 x 8 mm 10ks
 Disk 2ks	 Matice M3 10ks	 Držák 4ks
 Tlačítko napájení	 Kolo 2ks	 Univerzální kolo 1ks
 Box na baterie	 Podvozek 1ks	 Motor 2ks

Figure 5- List of components of robot vehicle chassis

The first step in the construction of the robot vehicle is to place the holders in position. These holders serve the function of holding the motor with the chassis. There are a total of 4 holders given in the kit and these go into slots provided in the chassis as shown below. The chassis and the holders looks transparent after the stickers have

been removed. The sticker are provided on both sides and have to be peeled carefully so that they don't leave any marks. The chassis is provided with many slots to help assemble many other components such as battery case, screws for wheel etc.

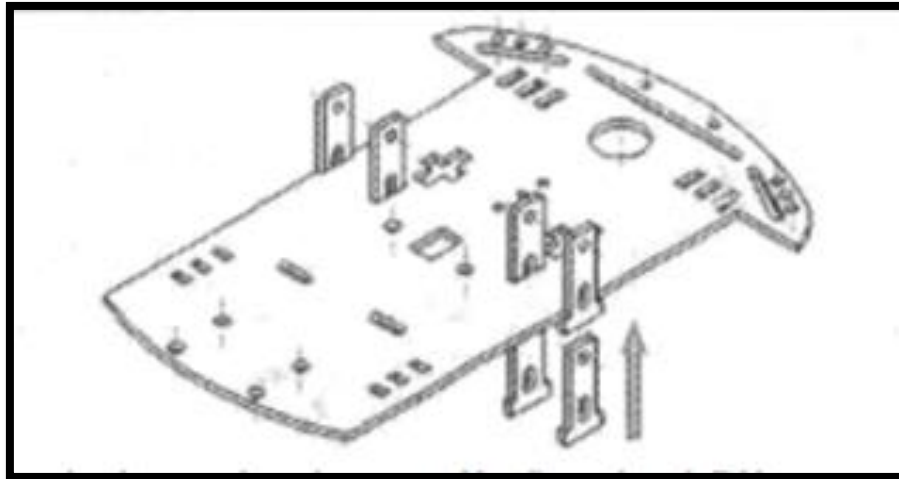


Figure 6 - First step of construction of vehicle chassis

The second step involves fixing of the motors to the chassis. The two tuning disks are placed in to their position as shown in the figure below. The motors are then placed in between the holders in their respective positions as shown below. The discs should not colloid with the chassis and should be able to rotate freely. This is ensured by placing in the plus shape gap provided in the chassis.

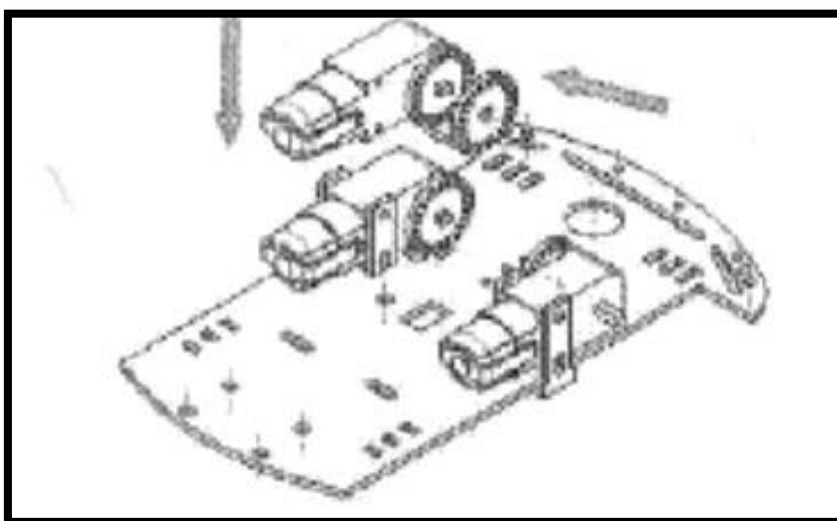


Figure 7 - Second step of Construction of vehicle chassis

The third step in the construction of the vehicle involves fastening of the motors with the chassis. There are four M3x30mm bolts which are provided along with the kit. These bolts are fastened through the holders into the motors.

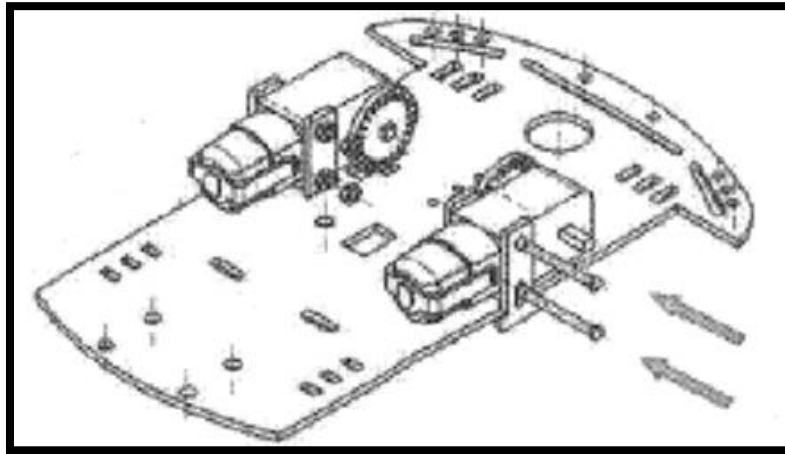


Figure 8 - Third step of construction of vehicle chassis

The next step in the construction of the robot car is to fix the caster wheel and the drive wheels in position. The castor wheel is fastened to the chassis with the help of tower bolts and M3x8mm bolts as shown in the figure below. The drive wheels are attached to the motor itself by slotted holes provided in the wheels

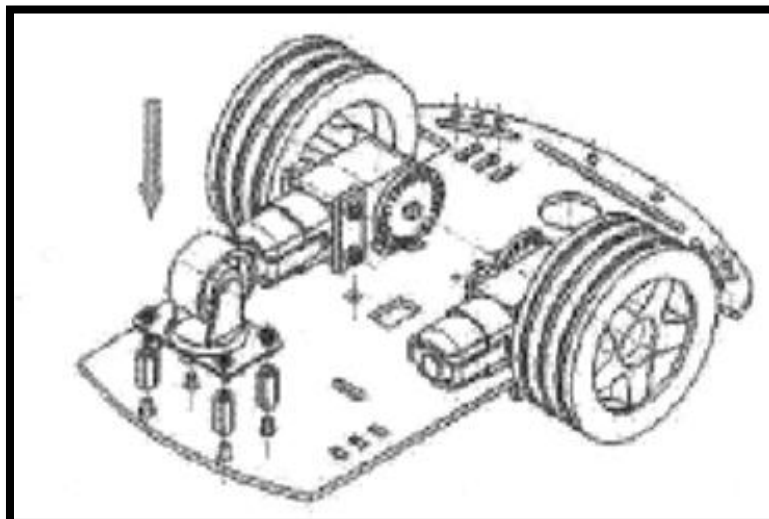


Figure 9- Fourth step of construction of vehicle chassis

The battery case is placed on top of the chassis and fixed with help of bolts M3x8mm and nuts provided. Thus the final assembly of the robot motor kit is shown below. This is used for further mounting of components such as sensor, LCD displays etc.

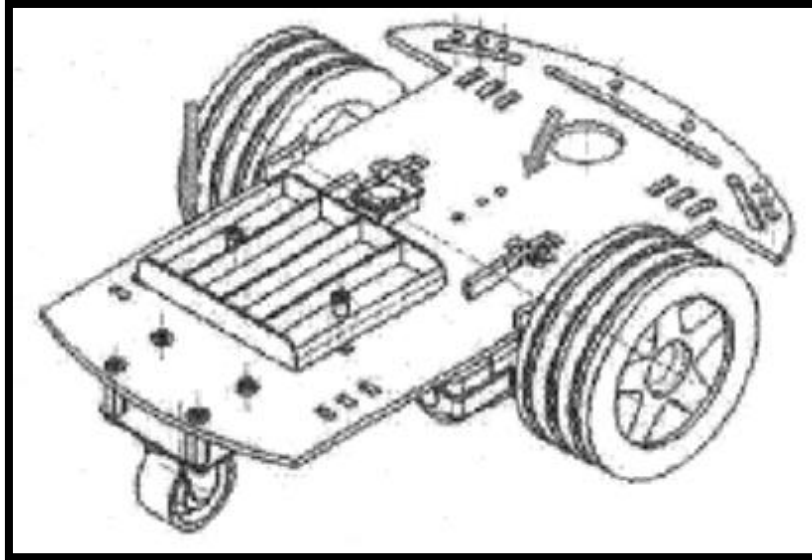


Figure 10- Total assembly of vehicle chassis

3.1 - Arduino H-bridge module L9110S

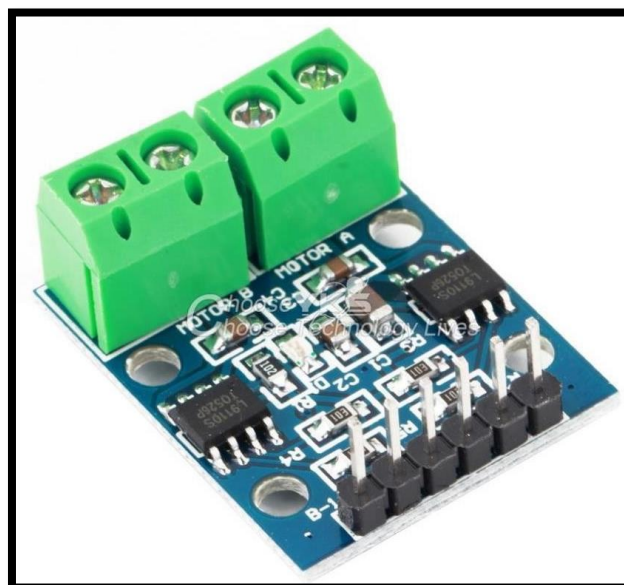


Figure 11 - H-bridge module

The connection of the H-bridge to the motors should be done as follows. There are 6 pins in total and the description of those pins is given below. [5]

B-IA	Motor B Input A (IA)
A-IA	Motor A Input A (IA)
VCC	Operating Voltage 2.5-12V
B-IB	Motor B Input B (IB)
A-IB	Motor A Input B (IB)
GND	Ground

Table 1- Pins of H-module

The pins B-IA, B-IB, A-IA, A-IB are connected to the arduino to different PWM pins whereas VCC and GND pins are connected to input power source and ground respectively.[6]

```
1  constint AIA = 9;
2  constint AIB = 5;
3  const int BIA = 10;
4  const int BIB = 6;
5  byte speed = 255;
6  void setup() {
7  pinMode(AIA, OUTPUT);
8  pinMode(AIB, OUTPUT);
9  pinMode(BIA, OUTPUT);
10 pinMode(BIB, OUTPUT);
11 }
12 void loop() {
13 forward();
14 delay(1000);
15 backward();
16 delay(1000);
```

```

17  }
18  void backward()
19  {
20  analogWrite(AIA, 0);
21  analogWrite(AIB, speed);
22  analogWrite(BIA, 0);
23  analogWrite(BIB, speed);
24  }
25  void forward()
26  {
27  analogWrite(AIA, speed);
28  analogWrite(AIB, 0);
29  analogWrite(BIA, speed);
30  analogWrite(BIB, 0);
31  }

```

Table 2 - Sample code for H-bridge

3.2 - 5 channel tracking sensor module board

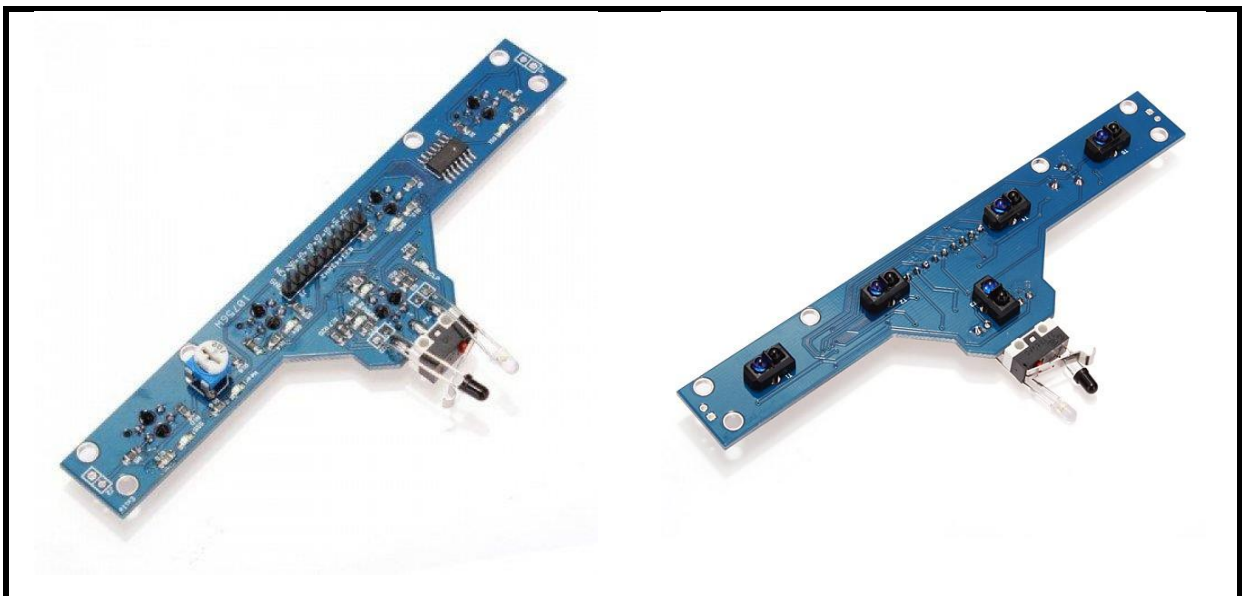


Figure 12- 5Channel tracking sensor module

PINS:

VCC + 5 Volts DC Power
SS1, SS2 – The two sensors on the left side
GND – Ground
CLP -- The Micro-switch on the front acting as physical bump sensor
Near -- The Forward-looking IR LED and Sensor on separate wires
SS4, SS5 – The two sensors on the right side
SS3 -- Center Sensor

Table 3- Pins of 5 channel tracking sensor

This 5 channel line following tracker has 5 IR sensors looking downwards. There are two sensors on the right and two sensors on the left and a sensor in the middle. These sensors change state when they are placed over a black surface. The infra-red sensor can differentiate between white and black lines as the infra red light is absorbed and not reflected by the black line. A suitable software program is used to allow the robot vehicle follow a track by using this line tracker sensor.[7]

Sample code:

```
1    #include <spi.h>
2    int L2=A0,L1=A1,M=A2,R1=A3,R2=A4;
3    int lMotFwd=4,rMotFwd=7,lMotRev=5,rMotRev=6;
4    byte pipeNo;
5    int gotByte = 0;
6    void setup()
7    {
8    pinMode(L2,INPUT);
9    pinMode(L1,INPUT);
10   pinMode(M,INPUT);
```

```

11  pinMode(R2,INPUT);
12  pinMode(R1,INPUT);
13  pinMode(lMotFwd,OUTPUT);
14  pinMode(rMotFwd,OUTPUT);
15  pinMode(lMotRev,OUTPUT);
16  pinMode(rMotRev,OUTPUT);
17  motorStop();
18  Serial.begin(9600);
19  }
20  void loop()
21  if((digitalRead(L2)==LOW)&&(digitalRead(L1)==HIGH)&&(digitalRead(M)==HI
GH)&&(digitalRead(R1)==HIGH)&&(digitalRead(R2)==LOW))
22  {
23  Serial.println("In decision");
24  radio.read( &gotByte, sizeof(gotByte) );
25  Serial.println(gotByte);
26  if(gotByte==1)
27  { turnLeft(160);
28  delay(100);}
29  else if(gotByte==2)
30  { turnRight(160);
31  delay(100);}
32  else{
33  motorStop();
34  }
35  gotByte = 0;}
36  else
37  if((digitalRead(L2)==HIGH)&&(digitalRead(L1)==HIGH)&&(digitalRead(M)==L
OW)&&(digitalRead(R1)==HIGH)&&(digitalRead(R2)==HIGH))
38  { moveForward();
39  }
else
if((digitalRead(L2)==HIGH)&&(digitalRead(L1)==LOW)&&(digitalRead(M)==HI

```

```

    GH)&&(digitalRead(R1)==HIGH)&&(digitalRead(R2)==HIGH))
40  {
41  turnLeft(160);}
42  else
    if((digitalRead(L2)==HIGH)&&(digitalRead(L1)==HIGH)&&(digitalRead(M)==HI
    GH)&&(digitalRead(R1)==LOW)&&(digitalRead(R2)==HIGH))
43  {
44  turnRight(160);}
45  else
    if((digitalRead(L2)==HIGH)&&(digitalRead(L1)==HIGH)&&(digitalRead(M)==L
    OW)&&(digitalRead(R1)==LOW)&&(digitalRead(R2)==LOW))
46  {turnRight(80):
47  }
48  else
    if((digitalRead(L2)==LOW)&&(digitalRead(L1)==LOW)&&(digitalRead(M)==LO
    W)&&(digitalRead(R1)==HIGH)&&(digitalRead(R2)==HIGH))
49  {
50  turnLeft(80);}
51  Else
52  {motorstop();
53  }}

```

Table 4- Sample code for Line tracking sensor

3.3 - RGB LED diode

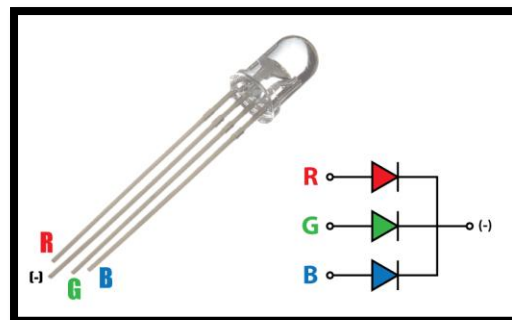


Figure 13- RGB LED diode

The RGB LED can emit different colors by mixing the 3 basic colors red, green and blue. So it actually consists of 3 separate LEDs red, green and blue packed in a single case. That's why it has 4 leads, one lead for each of the 3 colors and one common cathode or anode depending of the RGB LED type. In this tutorial I will be using a common cathode one.[8]

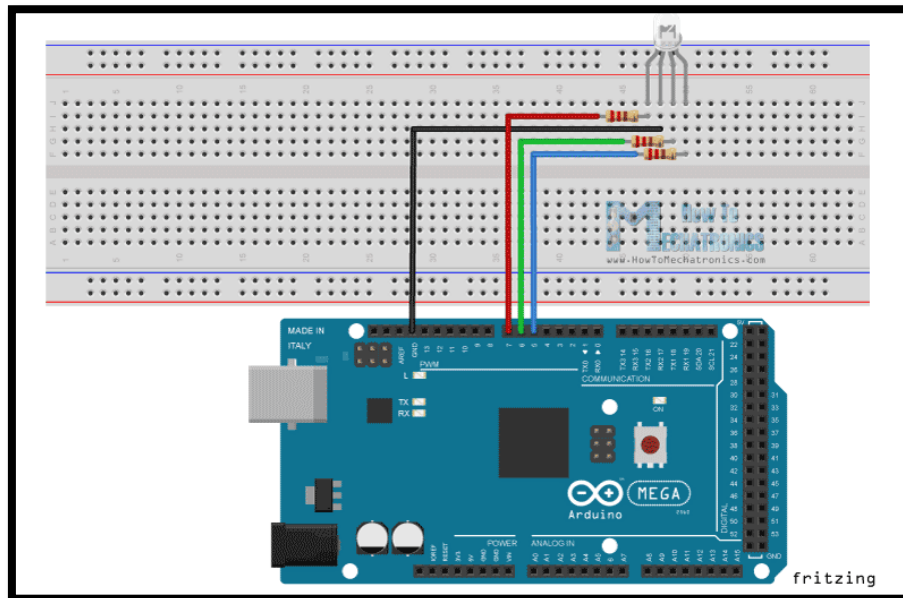


Figure 14 - Wiring diagram of RGB LED diode

```
1   intredPin= 7;
2   intgreenPin = 6;
3   intbluePin = 5;
4   void setup() {
5     pinMode(redPin, OUTPUT);
6     pinMode(greenPin, OUTPUT);
7     pinMode(bluePin, OUTPUT);
8   }
9   void loop() {
10    setColor(255, 0, 0); // Red Color
11    delay(1000);
12    setColor(0, 255, 0); // Green Color
```

```

13   delay(1000);
14   setColor(0, 0, 255); // Blue Color
15   delay(1000);
16   setColor(255, 255, 255); // White Color
17   delay(1000);
18   setColor(170, 0, 255); // Purple Color
19   delay(1000);
20   }
21   void setColor(intredValue, int
22   greenValue, intblueValue) {
23   analogWrite(redPin, redValue);
24   analogWrite(greenPin, greenValue);
25   analogWrite(bluePin, blueValue);
26   }

```

Table 5 - Sample code for RGB LED diode

3.4 - Graphic LCD display ST7920

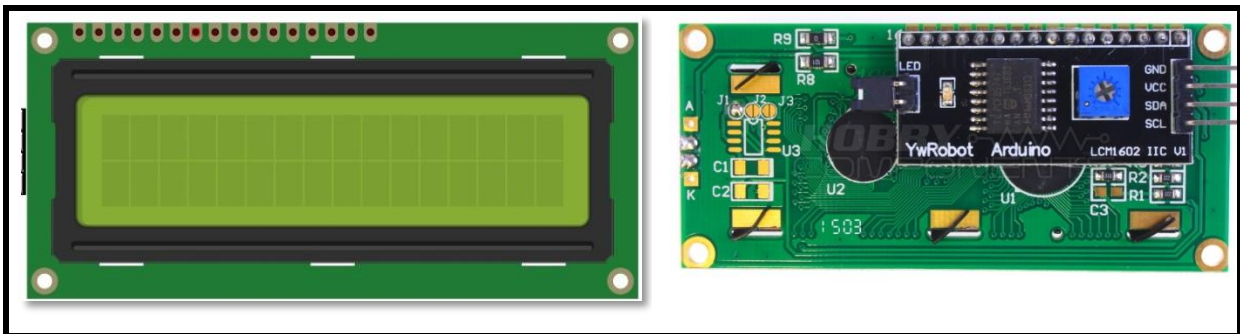


Figure 15- Graphic LCD display

This LCD display is used to display any required user data. The power source for the display is 5V. It is a 20 character, 2 rows display. It can be connected in two ways. One way is to connect using the 16 pin ridge or by using an I2C converter pins which uses just 4 pins GND,VCC,SDA and SCL. The contrast of the display can be adjusted by using a screw driver and adjusting the potentiometer provided at the back of the display. This controls the voltage supply to the display.[9]

Pin of the LCD display	Pin of the Arduino
GND	GND
VCC	5V
SDA	20
SLC	21

Table 6 - Table for wiring of Graphic LCD

```

1  # include <LiquidCrystal.h>
2  LiquidCrystallcd ( 12 , 11 , 5 , 4 , 3 , 2 );
3  void setup () {
4  lcd. begin ( 20 , 2 );
5  lcd. print ( " ARDUINO" );
6  lcd. setCursor ( 0 , 1 );
7  lcd. print ( "WELCOME!" );
8  delay ( 5000 );
9  }
10 void loop () {
11 lcd. setCursor ( 7 , 1 );
12 lcd. print ( millis () / 1000 );
13 }

```

Table 7 - Sample code for Graphic LCD

3.5 – Arduino Color Detector

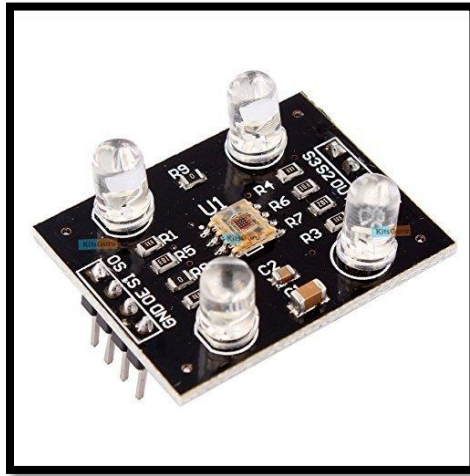


Figure 16- Arduino colour detector

This module is very interesting and can be used to detect colours when in close proximity. The module consists of four white colour LED's which are used to illuminate the subject under study and has a large number of photodiodes that filters colours. [10]

When light is reflected back to the module after illumination with white LEDs, and it can detect and generate the frequency of the RGB colour. For example, if placed over a green coloured object, the reflected light will have green component, which in turn generates the most current through the photodiodes with the green filter, which is then converted to the frequency. The higher the occurrence of a given colour, the lower its output frequency. A total of 7 connecting pins must be connected to successfully connect the Arduino board colour detector.

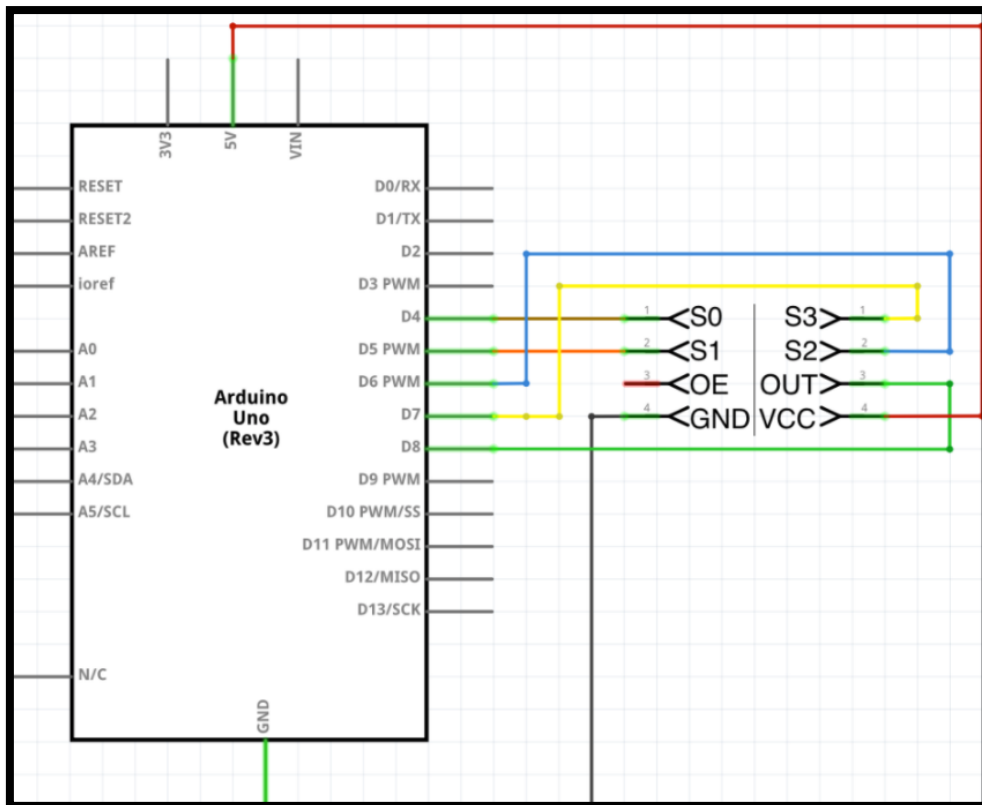


Figure 17- Wiring diagram of Arduino colour detector

```

1   #define pinS0
2   #define pinS1
3   #define pinS2
4   #define pinS3
5   #define pinOut
6   void setup() {
7     pinMode(pinS0, OUTPUT);
8     pinMode(pinS1, OUTPUT);
9     pinMode(pinS2, OUTPUT);
10    pinMode(pinS3, OUTPUT);
11    pinMode(pinOut, INPUT);
12    digitalWrite(pinS0,HIGH);
13    digitalWrite(pinS1,LOW);
14    Serial.begin(9600);
15  }

```

```

16 void loop() {
17   intfrequencyred, frequencygreen, frequencyblue;
18   digitalWrite(pinS2,LOW);
19   digitalWrite(pinS3,LOW);
20   delay(50);
21   frequencyred = pulseIn(pinOut, LOW);
22   digitalWrite(pinS2,HIGH);
23   digitalWrite(pinS3,HIGH);
24   delay(50);
25   frequencygreen = pulseIn(pinOut, LOW);
26   digitalWrite(pinS2,LOW);
27   digitalWrite(pinS3,HIGH);
28   delay(50);
29   frequency Blue = pulseIn(pinOut, LOW);
30   Serial.print("R: ");
31   Serial.print(frequencyRed);
32   Serial.print(" | G: ");
33   Serial.print(frequencygreen);
34   Serial.print(" | B: ");
35   Serial.print(frequencyblue);
36   if (frequencyred< 60) {
37     Serial.print(" | reddetection. ");}
38   if (frequencygreen< 60) {
39     Serial.print(" | greendetection. "); }
40   if (frequencyblue< 60) {
41     Serial.print(" | bluedetection. ");}
42   if (frequencyred< 60 &frequencygreen< 60 ) {
43     Serial.print(" | greendetection. ");}
44   Serial.println();
45   delay(900);
46   }

```

Table 8- Sample code Arduino colour detector

3.6 - Wi-Fi module

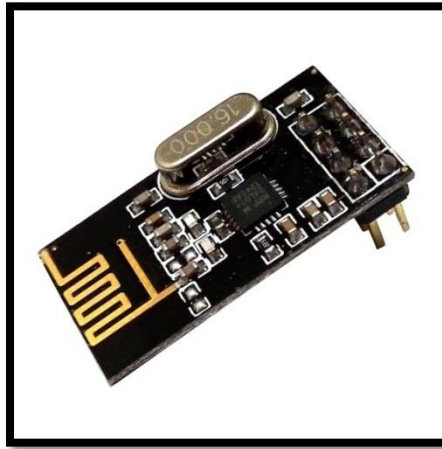


Figure 18 - Wi-Fi module

ArduinoWi-Fi module is a wireless module that allows communication between Arduinos when used in different places. The working frequency is 2.4 GHz and this frequency is generally used to connect computers or phones with wireless routers. The power requirement for wireless module is 3.3 volts. The transmission power can be set at four levels from MIN to MAX, but for HIGH and MAX it is suggested to use an external 3.3 V power supply because maximum current is no longer sufficient for these transmitting power, who can put a stabilizer on Arduino boards.[11]

This wireless module consumes hundreds of milliamperes for a short time when transmitting and receiving data, so it is recommended to connect a 10 microFarad capacitor between the 3.3V supply voltage and the ground, in addition to an external 3.3V supply.

The Wi-Fi module consists of 8 pins and the connection to arduino is very specific. The wiring of the Wi-Fi module to the arduino is shown diagrammatically below. The Wi-Fi module must not be connected to a 5v supply as this may result in the malfunction of the module and hence must be taken extra care when connections are given.

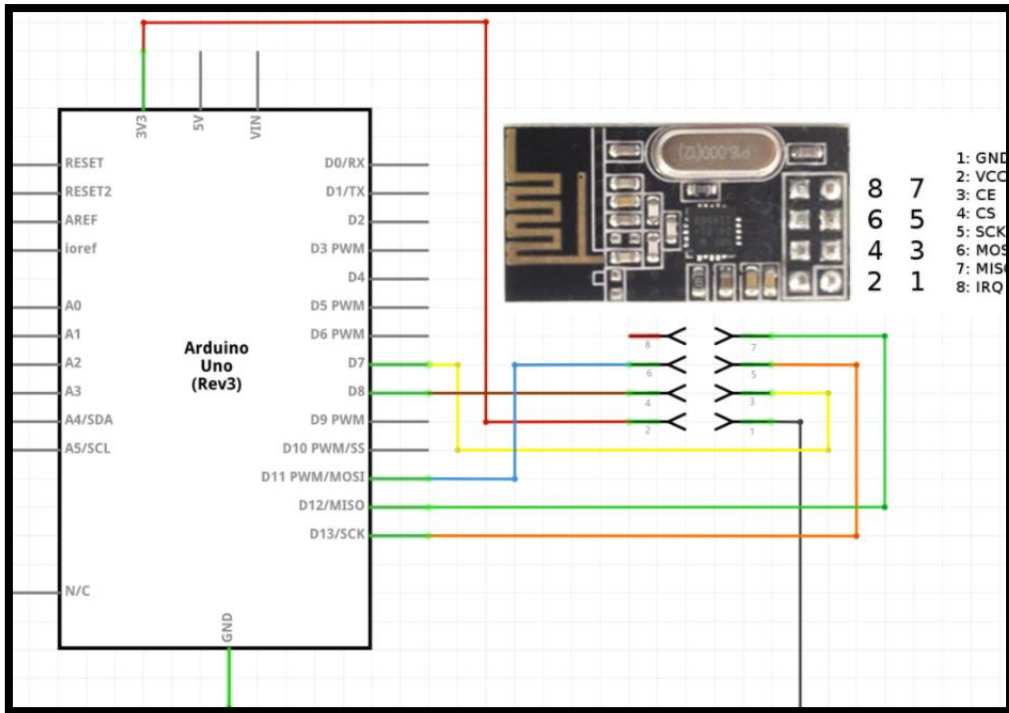


Figure 19 - Wiring diagram of Wi-Fi module

```

1  #include <SPI.h>
2  #include <nRF24L01.h>
3  #include <RF24.h>
4  RF24 radio(7, 8);
5  const byte address[6] = "00001";
6  void setup() {
7    radio.begin();
8    radio.openWritingPipe(address);
9    radio.setPALevel(RF24_PA_MIN);
10   radio.stopListening();
11   }
12   void loop() {
13     const char text[] = "Hello World";
14     radio.write(&text, sizeof(text));
15     delay(1000);
16   }

```

Table 9- Sample code for Wi-Fi module (Transmitter)

```

1  #include <SPI.h>
2  #include <nRF24L01.h>
3  #include <RF24.h>
4  RF24 radio(7, 8);
5  const byte address[6] = "00001";
6  void setup() {
7  Serial.begin(9600);
   radio.begin();
8  radio.openReadingPipe(0, address);
9  radio.setPALevel(RF24_PA_MIN);
10 radio.startListening();
11 }
12 void loop() {
13   if (radio.available()) {
14   char text[32] = "";
15   radio.read(&text, sizeof(text));
16   Serial.println(text);}
17 }

```

Table 10 - Sample code for Wi-Fi module (Receiver)

3.7 - RFID chip+reader



Figure 20- RFID Chip+reader

The RFID readers are divided according to the frequencies used. This module RC522 operates at a frequency of 13.56 MHz. To connect RFID readers with an Arduino board, it is necessary to give 7 wire connections. We connect SDA with pin 10, SCK with pin 13, MOSI with pin 11, MISO with pin 12, GND with ground, RST with pin 9 and 3.3V with 3V3 pin on Arduino. The reader module transmits information with the Arduino using SPI protocol. Hence, it is necessary to use these pins on the Arduino board, only two pins can be changed at the beginning or before the program and they are SDA and RST.

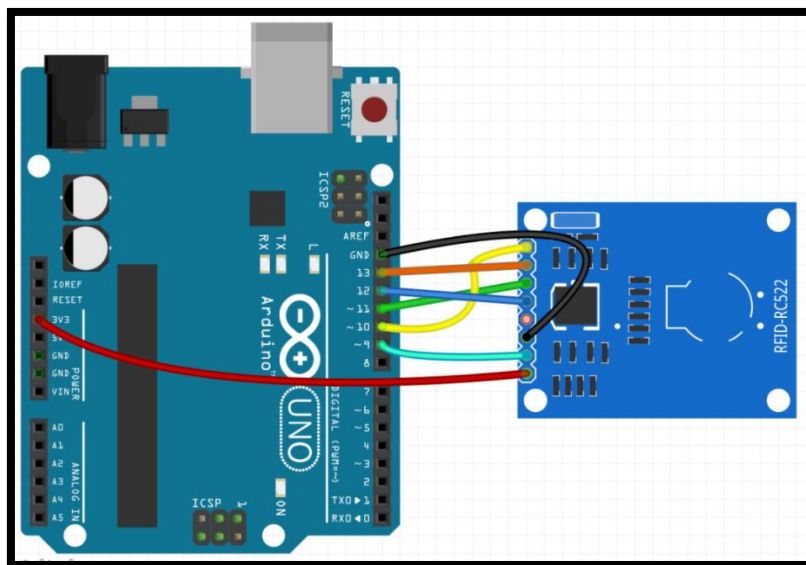


Figure 21- Wiring diagram of RFID chip+reader

```

1  #include <SoftwareSerial.h>
2  #define enablePin 9
3  #define rxPin 10
4  #define txPin 11
5  #define BUFSIZE 11
6  #define RFID_START 0x0A
7  #define RFID_STOP 0x0D
8  SoftwareSerial rfidSerial = SoftwareSerial(rxPin, txPin);
9  void setup()
10 {
11  pinMode(enablePin, OUTPUT);
12  pinMode(rxPin, INPUT);

```

```

13  digitalWrite(enablePin, HIGH);
14  Serial.begin(9600);
15  while (!Serial);
16  Serial.println("\n\nParallax RFID Card Reader");
17  rfidSerial.begin(2400);
18  Serial.flush();
19  }
20  void loop()
21  {
22  digitalWrite(enablePin, LOW);
23  char rfidData[BUFSIZE];
24  char offset = 0;
25  rfidData[0] = 0;
26  {
27    if (rfidSerial.available() > 0)
28    {
29      rfidData[offset] = rfidSerial.read();
30      if (rfidData[offset] == RFID_START)
31        offset = -1;
32    }
33    else if (rfidData[offset] == RFID_STOP)
34    {
35      rfidData[offset] = 0;
36      break;
37    }
38    offset++;
39    if (offset >= BUFSIZE) offset = 0;
40  }
41  }
42  Serial.println(rfidData);
43  Serial.flush();

```

Table 11 - Sample code for RFID chip+reader

3.8- Single IR sensor

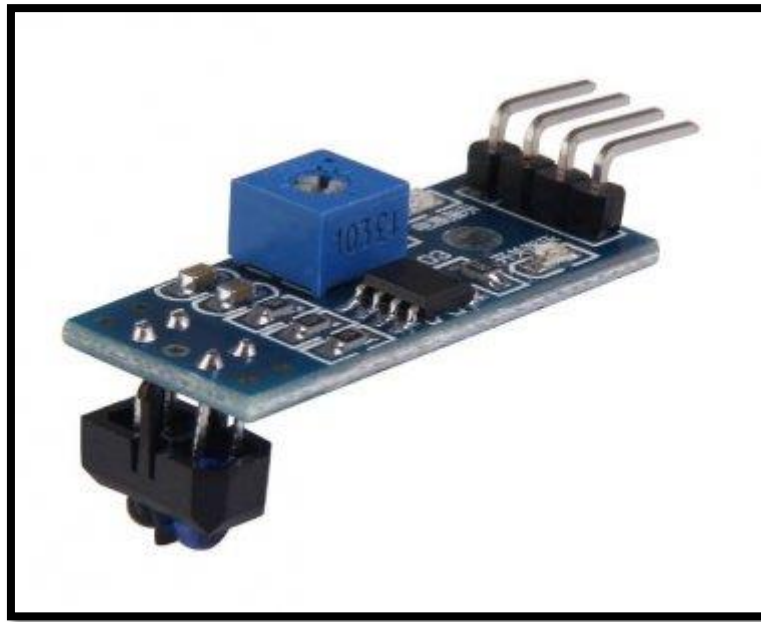


Figure 22- Single IR sensor

The infrared optical sensor has a wide variety of uses. Its main function is to detect the body's proximity to the sensor, whether it's a robot obstruction, paper in the printer or wheel speed measurement. This infrared optical sensor works, as the name suggests, with a pair of components, namely an infrared diode and an optical receiver, in this case a phototransistor.

Looking closer to the sensor on the module, we can see that these two components are separated by a plastic partition, without which the sensor would permanently detect the approach. Its whole function lies in the fact that the infrared diode sends out a light pulse that reflects from an obstacle and is captured by an optical detector. Depending on the amount of incident light, the transistor will be adequately opened and, thanks to the operational amplifier on the module, we can measure the analog voltage within the range of 0-5 V. It is important, however, to emphasize that this optical sensor has a maximum detectable distance of about 20 cm. The most accurate is about 5 cm, with the distribution of the output voltage not corresponding to the linear distance, for example half the voltage corresponds to about 3 cm.

This module with an infrared optical sensor has four interconnecting pins. The connection with Arduino is simple and is shown in the picture below. Just connect the Vcc to

the supply voltage in the range of 3.3 - 5 V, Gnd to the ground and we have the pins A0 and D0. These are used to get information from the sensor.

Pin A0 provides an analogue voltage value, so it is connected to the analogue input A0 on Arduin. Pin D0 then provides a digital value where the switching threshold is selected by the trimmer located on the module. In the case of a distant obstacle, the digital output D0 is equal to the supply voltage and falls on Gnd when detected. This output is advantageous when we do not need to measure the distance dynamically, but we only need to know the "closed / disabled" information.

The sample code for the connection and testing of the infrared optical sensor contains the first initialization of the data pins and the variable for reading the voltage value. In the setup subroutine, we first set up a 9600 baud serial communication line, set both pins as input and also connect the digital input D0 to pin 2 as an input for interrupt.

This interruption has the form: the interrupt pin number (for us 0 or 1 for pin 3), the name of the subroutine containing the commands to be executed, and the last type of the detection type. This detection has four different settings: LOW for logical zero detection (Gnd), CHANGE for detecting changes from one level to another, RISING for leading edge detection (log0 to log1), and FALLING for log-to-log detection. Due to the behavior of the module when the proximity detection is set to log0,

The loop loop contains commands that first upload the corresponding voltage value in the range 0-1023, where 0 corresponds to Gnd. This is followed by the "prerus" subroutine, which prints a motion detection message on the serial line on the serial line.

Wiring diagram:

The wiring diagram for the single IR sensor is shown below. This wiring diagram shows the connection to a arduino UNO which can be used as a reference for giving connections

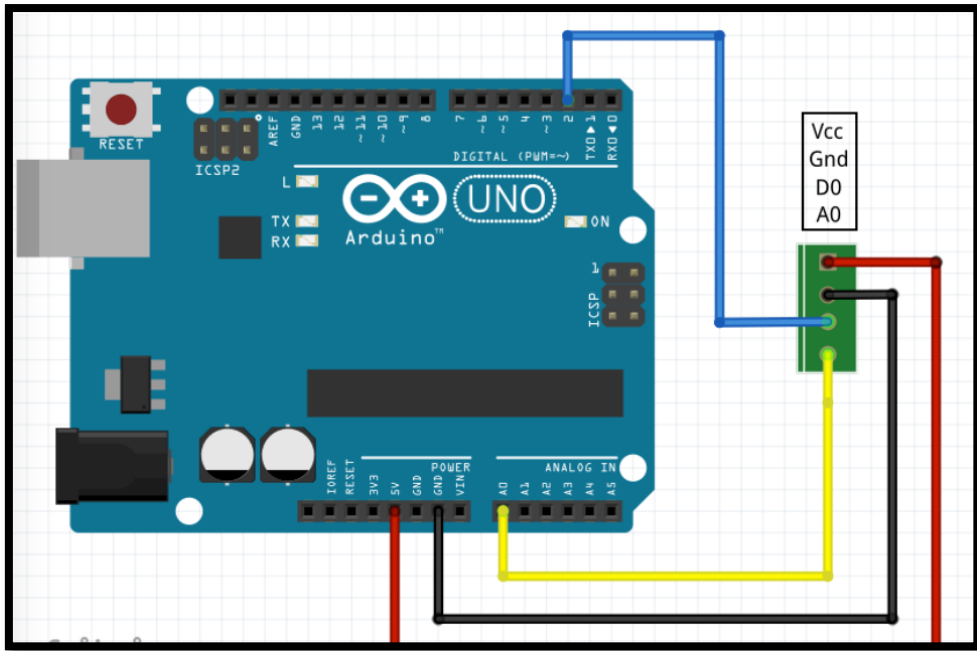


Figure 23- Wiring diagram for single IR sensor

Sample code:

```

1      #include <SPI.h>
2      int singleIR;
3      Int lMotFwd=4,rMotFwd=7,lMotRev=5,rMotRev=6;
4      void setup()
5      {
6          pinMode(singleIR,INPUT);
7          pinMode(lMotFwd,OUTPUT);
8          pinMode(rMotFwd,OUTPUT);
9          pinMode(lMotRev,OUTPUT);
10         pinMode(rMotRev,OUTPUT);
11         motorStop();
12         Serial.begin(9600);
13     }
14     void loop()
15     {
16         if(singleIR == LOW){
17             motorStop();

```

18	}
19	Else{
20	Followline();
21	}}

Table 12- Sample code for Single IR

3.9 - Arduino Mega 2560

Arduino Mega 2560 is a microcontroller board. There a total of 54 digital input/output pins and 15 of them can be used as PWM outputs, 16 of them are analog inputs, 4 hardware serial ports, a 16 Mhz crystal oscillator, reset button, an ICSP header, a power jack and a USB connection. It has all that is required to support the microcontroller by connecting it to a computer using an USB cable or battery power source or by an AC to DC adapter.[12]

The Arduino Mega 2560, the successor to the Arduino Mega, is a microcontroller board based on a ATmega2560 AVR microcontroller. It has 70 digital input/output pins (of which 14 can be used as PWM outputs and 16 can be used as analog inputs), a 16 MHz resonator, a USB connection, a power jack, an in-circuit system programming (ICSP) header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

The Mega 2560 differs from the preceding Mega in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter. This auxiliary microcontroller has its own USB bootloader, which allows advanced users to reprogram it.

3.9.1 Technical specifications

Microcontroller	ATmega2560
Operating Voltage	5V

Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	A total 54 and 14 of them provide PWM output
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Table 13 - Technical specifications of Arduino Mega 2560

3.8.2 – Power

The source of power for Arduino Mega 2560 is via the USB connection or using an external power supply. The selection of power source is done automatically. External power source will be from either battery or an AC to DC adapter. A 2.1mm centre positive plug is used to connect the adapter into the board's power jack. Leads from the battery is given to Gnd and Vin pin headers of the power connector. It is possible to operate the board with an external power supply of 6 to 20 volts. If the supply is less than 7V, there may be a drop of power supply from 5V pin and the board may become unstable. In other case of a supply more than 12V, there may be over-heating of the voltage regulator and may damage the arduino board. The range of volts that is recommended is between 7 and 12 volts.[13] The power pins are as follows:

- Vin –The power source for the arduino can be given using this pin by combining it with Gnd or when the board is using an external power source, power can be provided to other components via this Vin pin.

- 5V – This pin can be used to give a regulated power supply to other components or can be used to power the micro controller itself
- 3.3V –It also functions similar to 5V pin but the regulated power supply here is 3.3V.. Maximum current draw is 50 mA. •
- GND - Ground pins.

3.8.3 – Programming

The Arduino Mega2560 can be programmed with the Arduino software which is available as an open source. The Arduino Mega with the Atmega2560 is preburned with a bootloader which allows you to upload new code to it without the use of an external hardware programmer

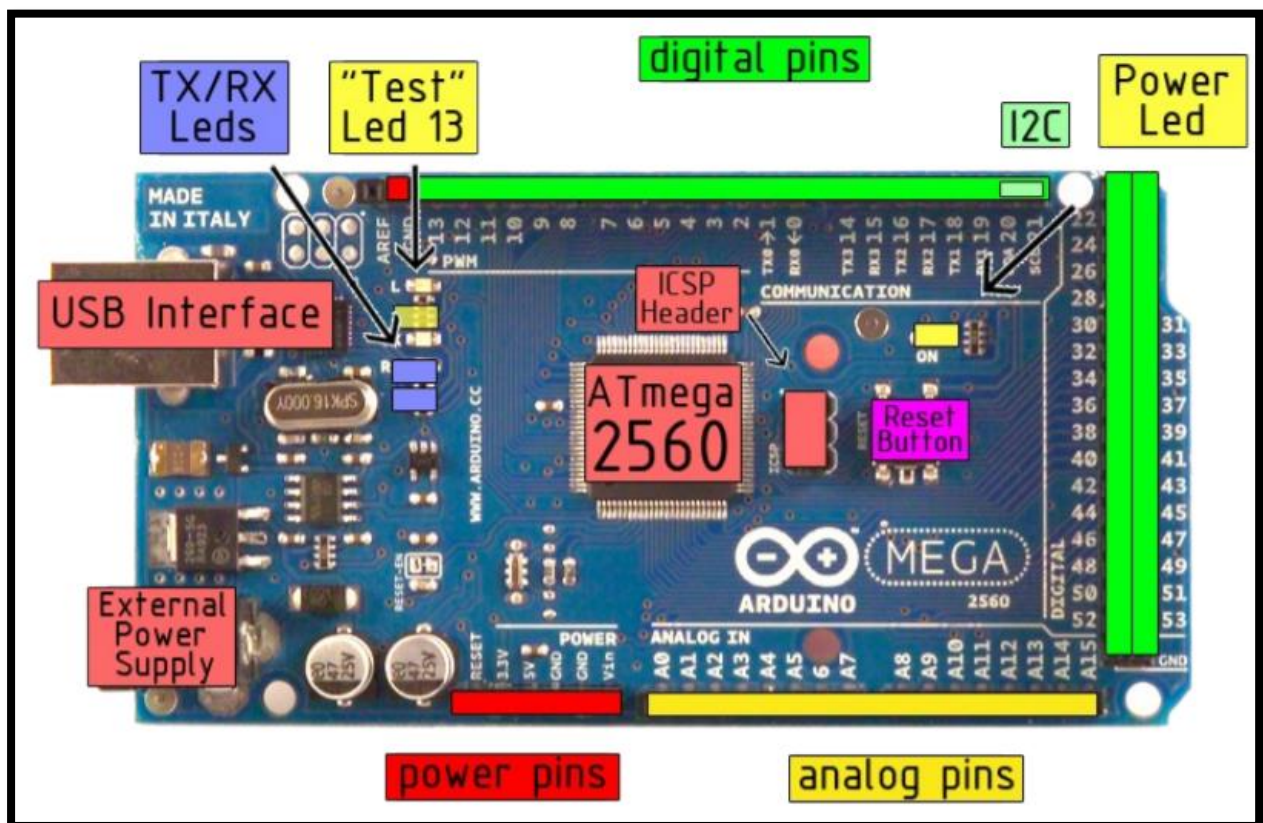


Figure 24 – Arduino Mega2560

3.8.4 - Automatic Reset

The Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer rather than requiring a physical press of the reset button before an upload. In the ATmega8U2, one of the flow control lines of the hardware is connected to the reset line of the ATmega2560 using a 100 nano farad capacitor. The reset line drops long enough to reset the chip when this line is asserted.[14]

3.8.5 - USB overcurrent protection

The Arduino Mega is provide with a resettable polyfuse to protect the USB ports of computer from shorts and overcurrent. The fuse provides an extra layer of protection although most computers provide their own internal protection,. If a current of more than 500 mA is provided to the USB port, the fuse will automatically break the connection until the short or overload is removed.

4-Functions of the Smart Factory vehicle

The Smart Factory vehicle is designed and programmed to facilitate easier and quicker transportation of goods in a typical factory environment. The environment required for the Smart Factory vehicle to work efficiently has been explained and illustrated in section 4. The vehicle is programmed to follow the track in a pre-determined direction.

A Smart Factory vehicle is basically a robot designed to follow a ‘line’ or path already predetermined by the user. This line or path may be as simple as a physical white line on the floor. In order to detect these specific markers or ‘lines’, various sensing schemes can be employed. These schemes may vary from simple low cost line sensing circuit to expensive vision systems. The choice of these schemes would be dependent upon the sensing accuracy and flexibility required. From the industrial point of view, line following robot has been implemented in semi to fully autonomous plants. In this environment, these robots functions as materials carrier to collect products from one manufacturing point to another where rail, conveyor and gantry solutions are not possible. Apart from line following capabilities, these robots should also have the capability to navigate junctions and decide on which junction to turn and which junction to ignore.

This would require the robot to have 90 degree turn and also junction counting capabilities. To add on to the complexity of the problem, sensor positioning also plays a role in optimizing the robots performance for the tasks mentioned earlier. The vehicle moves on a specified path to collect the components from specified locations. The vehicle is a self-operating robot that detects and follows a line drawn on the floor. The path to be taken is indicated by a white line on a black surface. The control system used must sense the line and man oeuvre the robot to stay on course while constantly correcting the wrong moves using feedback mechanism, thus forming a simple yet effective closed- loop system.

The path or the track that the vehicle is supposed to traverse is set up in a way that there are two loop: inner and outer loop that merges and demerges at two different locations. All the colour patches or delivery stations are placed inside the inner loop while a black patch is placed along the outer loop. This project has multiple delivery stations placed next to a colour patch. These delivery stations contain goods that are supposed to be transferred to the vehicle when the vehicle halts in front of the delivery station. The vehicle halts when it encounters the desired colour. The working of the delivery station function is beyond the scope of this project and this project comprises of the functions of the vehicle only.

Apart from the line following capabilities, the vehicle is employed with other functions such as Wi-Fi communication, colour detection and halting at desired location. Another stationary arduino is used in this project that host the transmitter Wi-Fi module while the receiver Wi-Fi module is mounted in the vehicle. The transmitter mounted arduino is used by the user to communicate to the vehicle on several occasions. The transmitter Wi-Fi module communicates a colour pattern to the vehicle at the beginning of the process. The vehicle receives this pattern and starts traversing the inner loop of the black line while constantly searching for the first colour on the colour pattern, once the vehicle finds the colour, it halts at that location and waits for the delivery station to transfer the goods to the vehicle. The vehicle then starts searching for the second colour in the pattern and continues to do the same for all the other remaining colours in the pattern.

While the vehicle is searching for the next colour in the pattern, the vehicle comes across the demerging junction of the inner and the outer loop multiple times, for this situation the vehicle is programmed to choose the inner loop while it still has remaining colours to traverse according to the colour pattern. The vehicle is programmed to choose the outer loop when it has finished collecting from all the colours/delivery station and has no more colours

left in the colour pattern. This function can also be overridden by the Wi-Fi module and the vehicle can be made to stop at the demerging junction and await a command from the transmitter Wi-Fi module on which direction to take or when the transmitter sends the command in prior, the vehicle takes an appropriate direction without halting at the junction.

Along the course of the inner loop, multiple colour patches are placed next to the appropriate delivery station. The colour sensor is programmed in such a way that when it encounters a colour that is next in order in the colour pattern, it halts in front of the colour patch until the delivery station transfers the goods to the vehicle. Once the vehicle collects the goods, the vehicle starts searching for the next colour in the colour pattern and follows the same procedure.

Once the vehicle is done collecting from all the colours/delivery stations, the vehicle always chooses the outer loop. The outer loop has a black patch along its course and the vehicle has a single IR sensor apart from the five IR sensors used for line following. When the single IR sensor encounters this black patch along the outer loop, the vehicle is programmed to stop and that marks the completion of the process.

4.1- Line following

This vehicle is capable of following a line without the help of any external source. The vehicle uses two motors to control front wheels and the single rear wheel which is free. It has 5-infrared sensors on the bottom for detection of black tracking tape. When the middle sensor detects the black colour, this sensor output is given to the comparator. The output of comparator compares this sensor output with a reference voltage and gives an output. The output of comparator will be low when it receives an input from the sensor. We follow a simple logic to implement this project.

As we know that black colour is capable of absorbing the radiation and white colour or a bright colour reflects the radiation back. Here we use 5 pairs of IR Tx and Rx .The robot uses these IR sensors to sense the line and the arrangement is made such that the sensors face the ground. The output from the sensors is an analog signal which depends on the amount of light reflected back and this analog signal is given to the comparator to produce 0s and 1s.

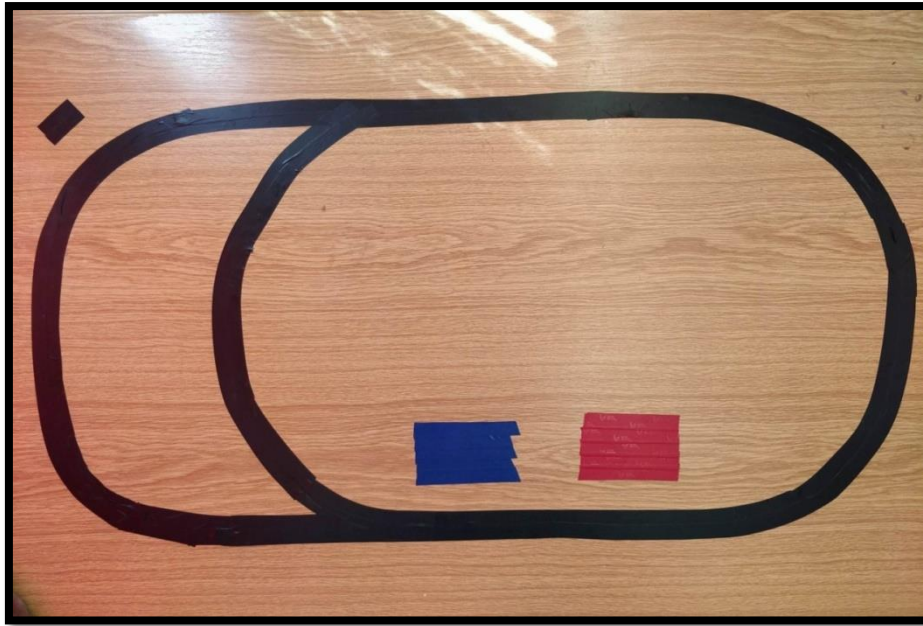


Figure 25 – Track used for navigation

Internally the arduino is programmed to control the rotation of the wheels using these digital outputs. The rotation of these wheels depends up on the response from the comparator. Let us assume that when a sensor is on the black line it reads 0 and when it is on the bright surface it reads 1. Here we can get 7 different cases, they are: 1. Straight direction 2. Right curve 3. Left curve 4. Sharp Right curve 5. Sharp Left curve 6. Demerging curves 7. Merging curves

4.1.1 Straight direction

We can expect our vehicle to move in straight direction when the middle sensor's response is low and the remaining sensors' response is high. i.e., according to our arrangement the middle sensor will always be on the line and as the line is black in colour it will not reflect the emitted radiation back and the response of the sensor will be low and the response of the remaining sensors will be high as they will be on the bright surface.

4.1.2 Right curve

When a right curve is found on the line the responses will change i.e. the response of the first sensor which is to the right will become low as that sensor will be facing the black line and the remaining sensors' response will be high. When this data is achieved the control of the wheels is changed i.e. the right wheel is held and the left wheel is made to move freely until the response from the middle sensor becomes low. Then the same process repeats again.

4.1.3 Left curve

When a left curve is found on the line the response, then response of the first sensor to the left will be changed from high to low as the sensor will now face the black or the dark surface. Then the control of the wheel changes i.e. by holding the left wheel and allowing the right wheel to move freely until the middle sensor changes its response for high to low.

4.1.4 Sharp right curve

When a sharp right curve, like a 90 degree right turn, is found on the line the responses will change i.e. the response of all the right sensors will become low as those sensors will be facing the black line and the remaining sensors' response will be high. When this data is achieved the control of the wheels is changed i.e. the right wheel is held and the left wheel is made to move freely until the response from the middle sensor becomes low. Then the same process repeats again.

4.1.5 Sharp left curve

When a left curve, like a 90 degree left turn, is found on the line the response, then response of all the left sensors will be changed from high to low as the sensors will now face the black or the dark surface. Then the control of the wheel changes i.e. by holding the left

wheel and allowing the right wheel to move freely until the middle sensor changes its response for high to low.

4.1.6 Demerging curves

In this project, the track contains a single set of demerging curves as shown and explained in section 4. The demerging curves result in two closed loops merging at a different location in the track. The vehicle continues to traverse the inner loop until it completes the collection of all the colour components according to the pattern given by the user through Wi-Fi signals before the start of the line following process. The vehicle then moves to the outer loop as directed by the Wi-Fi signals given by the user. This will be explained further in section 5.2.

When the vehicle traverses a curve that splits into two, the response of the left most and the right most sensors will become low whereas the other sensors will remain high. At this juncture, the vehicle expects or waits for a command from the user through Wi-Fi. Based on the command the vehicle is programmed to turn left or right to traverse the outer or inner loop respectively.

4.1.7 Merging curves

When the inner loop and the outer loop merges, the response of the middle three sensors will become low whereas the other sensors will remain high. At this juncture, the vehicle is programmed to carefully assess the track slowly and take a gradual right turn until the middle sensor remains low and the other sensors' values are recorded high.

4.2- Decision Making on line tracking

The prototype of the track is designed to have two loops: inner and outer loop. The vehicle is expected to loop inside the inner loop while it is collecting the goods from the delivery station as instructed by the user. Once the vehicle has completed the task of collecting, the vehicle quits the inner loop and enters the outer loop at the demerging point. The outer loop has a black patch on its side. The single IR, on contact with this black patch, stops the motors of the vehicle.

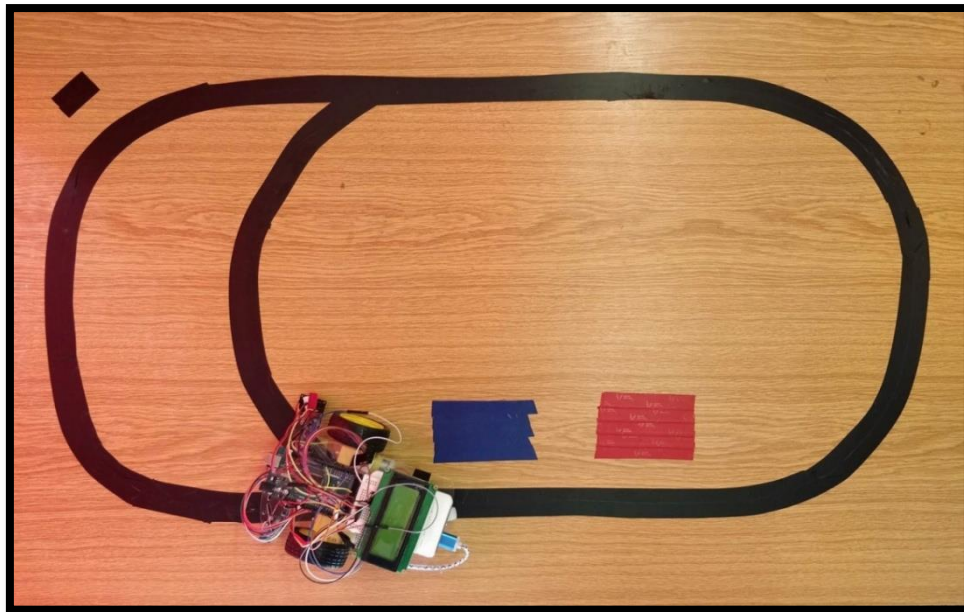


Figure 26 – Decision making- type I

The vehicle identifies the demerging point when the left most and the right most sensors become low while the other sensors remain high. At this point, the vehicle can be made to turn in two direction to follow either of the two loops. When the right motor is stopped and the left motor moves, the vehicle takes a right turn until the middle sensor becomes low and follows the inner loop. Whereas, when the left motor is stopped and the right motor moves, the vehicle takes a left turn and follows the outer loop.

In this project, we use Wi-Fi signals to communicate to the vehicle. When the vehicle is keyed in with a pre-determined colour pattern to collect the goods from the respective colour(delivery station), the vehicle has to traverse the inner loop multiple times to complete the task. While traversing, the vehicle comes in contact with the demerging curves multiple times. At each junction, the vehicle is programmed to behave in 4 different ways, 2 of which are directed by the user and the other 2 are a automated way of decision making based on the task completed.

4.2.1 Automated path selection

In this mode, the vehicle automatically chooses the direction or loop based on the pending work load.

1) In a situation in which the vehicle still has more colours to visit and collect from, the vehicle is programmed to automatically choose the inner loop at every demerging junction. The vehicle continues to traverse the inner loop until it visits and collects from all the colours.

2) When the vehicle reaches the demerging point after collecting from all the colours, the vehicle automatically chooses the outer loop and continues to traverse the outer loop.

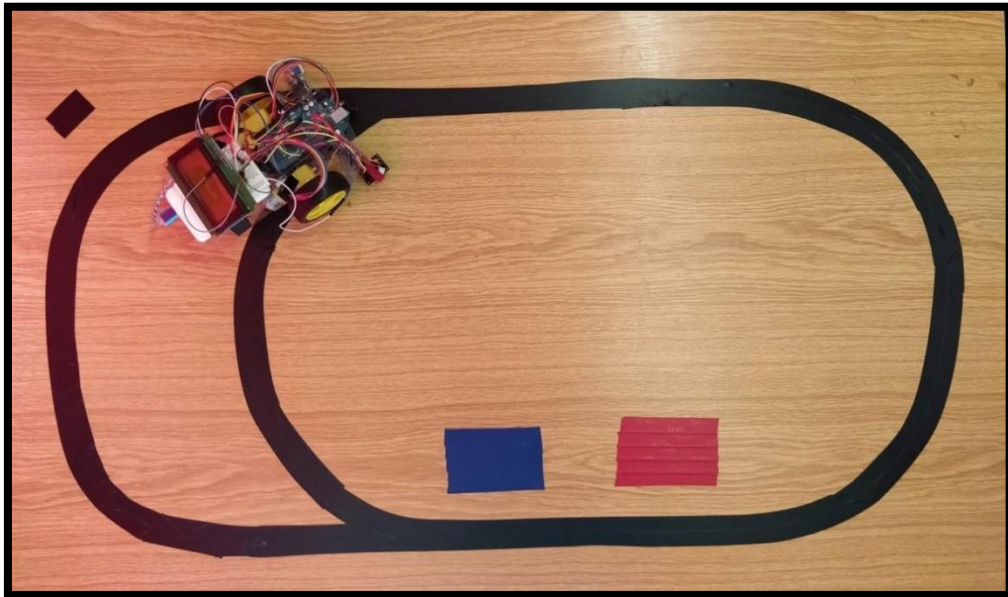


Figure 27- Decision making – type II

4.2.2 - On command path selection

In this mode, the vehicle awaits a command from the user through Wi-Fi on which direction or loop to traverse.

3) Before the vehicle reaches the demerging point, the user can send a command through Wi-Fi to the vehicle to change or continue the direction of movement. The vehicle as per the user's command either continues to traverse the inner loop while collecting loads from the colour points or changes its direction and traverses the outer loop.

4) At every junction, the vehicle can also be programmed to stop and await a user command. Once the user communicates the direction to the vehicle, the vehicle takes a turn to traverse the inner or outer loop accordingly.

4.3- Wi-Fi communication between arduinos

A pair of Wi-Fi modules have been employed in this project. One Wi-Fi module works as a transmitter while the other acts as a receiver. The receiver is connected to the arduino on the vehicle whereas the transmitter is connected to another arduino that is stationary and used by the user to send commands to the vehicle. Wi-Fi communication is very critical in this project because the Wi-Fi commands contribute to vital decision making process. At the beginning of the process, the transmitter Wi-Fi module will send the colour pattern to the vehicle. The vehicle captures this pattern and traverses the loops according to this pattern. While the vehicle is traversing the loops, at the junction of demerging curves, the vehicle needs to decide on which direction to take. The user can send commands to the vehicle through the transmitter on which direction to take.

The transmitter and the receiver Wi-Fi modules constantly communicate with each other until the vehicle is powered off. At the beginning of the process, the receiver awaits to receive the colour pattern. The transmitter send a colour pattern to the receiver. Once the colour pattern has been communicated to the receiver, the Wi-Fi modules go into the loop function, in which the receiver is constantly listening and the transmitter is constantly writing to the receiver. This helps in the decision making process that is explained in section 5.2.

In this project, all the modules are interlinked to each other through arduino. While the line follower module tracks the given line, a situation might arise in which the decision of the user is necessary. In that case, the line follower modules switches the control to the Wi-Fi module to receive a command from the user. Once the receiver Wi-Fi module on the vehicle receives a command, the arduino interprets the command and switches the control back to line follower module. The receiver might receive the following commands from the transmitter:

4.3.1 - When the vehicle is powered on

Upon on powering both the arduino that mounts the receiver and the arduino that mounts the transmitter, the vehicle (receiver) does not start tracking the path instantly. The vehicle awaits a colour pattern from the transmitter. The user immediately sends the pattern to the receiver through the transmitter. Upon receiving the pattern, the vehicle starts tracking the path according to the colour pattern received.

4.3.2 - When the vehicle is tracking

While the vehicle is tracking, the user has to send commands to the receiver on which direction to take. This command can be communicated well in prior so the vehicle will automatically make an appropriate turn when it encounters a demerging curve. If the user does not communicate the command in advance, the vehicle will stop at the demerging junction and will await a command from the user. Once the user sends a command, the vehicle will take an according direction.

4.4- Colour Detection

In this project we are going to interface colour sensor which can detect any number of colours with right programming. The colour sensor contains RGB (Red Green Blue) arrays. The square boxes inside the eye on sensor are arrays of RGB matrix. Each of these boxes contain three sensors, one is for sensing RED light intensity, one is for sensing GREEN light intensity and the last one for sensing BLUE light intensity.

Each of sensor arrays in these three arrays are selected separately depending on requirement. Hence it is known as programmable sensor. The module can be featured to sense a particular colour and to leave the others. It contains filters for that selection purpose. There is a fourth mode that is no filter mode. With no filter mode the sensor detects white light.

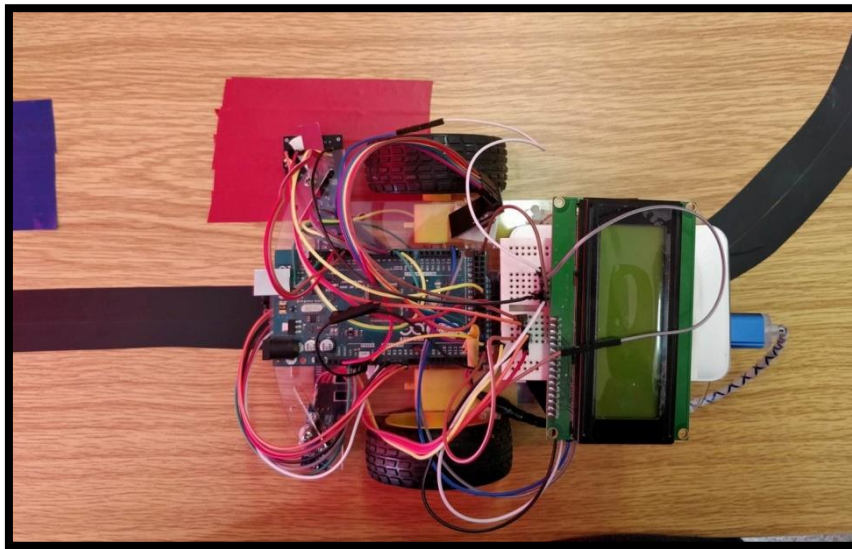


Figure 28- Stopping at red colour

The color which needs to be sensed by the colour sensor is selected by two pins S2 and S3 of the colour sensor. With these two pins logic control we can tell sensor which color light intensity is to be measured. Say we need to sense the RED color intensity we need to set both pins to LOW. Once that is done the sensor detects the intensity and sends the value to the control system inside the module.

S2	S3	Photodiode Type
Low	Low	Red
Low	High	Blue
High	Low	Clear (no filter)
High	High	Green

Table 14- Values of a colour sensor

The transmitter Wi-Fi module communicates a colour pattern before the start of the process, the receiver (vehicle) receives this pattern and traverses the path accordingly. For instance, if the transmitter sends a pattern <Red, Blue, Red>, the vehicle follows the black line while constantly searching for the colour RED, once the colour sensor senses the RED colour, the vehicle is programmed to stop at that location for 15 seconds until the delivery station fixed at that location, working independently from this project, transfers the goods to the vehicle. The vehicle then traverses the line searching for BLUE colour and stop at blue for 15 seconds and then finally searches for RED again and follows the same procedure. The vehicle chooses the outer loop at the demerging point after collecting the pattern

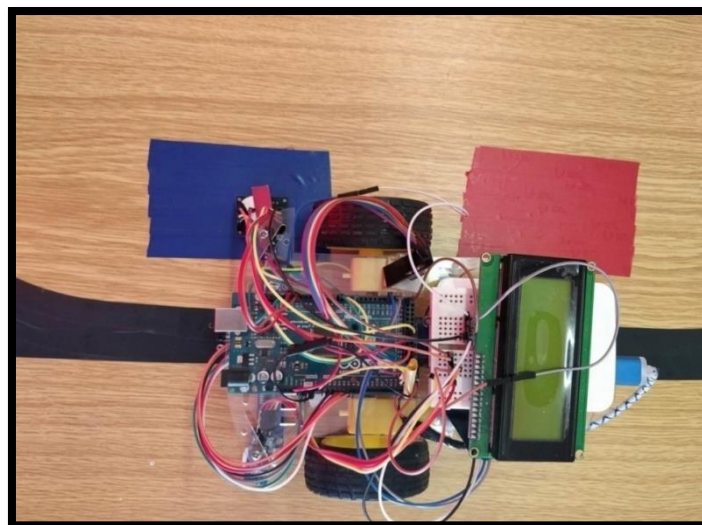


Figure 29- Stopping at blue colour

4.5- Stopping at Delivery point

Apart from the 5 IR sensors used for following the line, another single IR sensor is used in this project to stop the vehicle. This IR sensor is a pair of Transmitter (Tx) and Receiver(Rx). The transmitter emits an IR wave and the wave reflects back. The reflected wave is captured by the receiver. The output from the sensor is an analog signal which depends on the amount of light reflected back and this analog signal is given to the comparator to produce 0s and 1s. Black colour absorbs the wave while white colour reflects it back. So when the IR sensor is placed on a black coloured surface, the output of the sensor is 0 but when the IR sensor is placed on a white surface, the output of the sensor is 1.

Once the Vehicle is finished traversing all the colours and has collected all the goods according to the colour patten shared by the Wi-Fi transmitter, the vehicle is programmed to choose the outer loop at the demerging point. A black patch is placed along the course of the outer loop and the vehicle is programmed to stop once the single IR sensor encounters this black patch and the output becomes 0 from 1. When the arduino receives 0 signal from the single IR sensor, the arduino is programmed to stop the motors by sending LOW signals to the motors. When the vehicle stops at the black patch along the outer loop, that signifies the completion of the process.

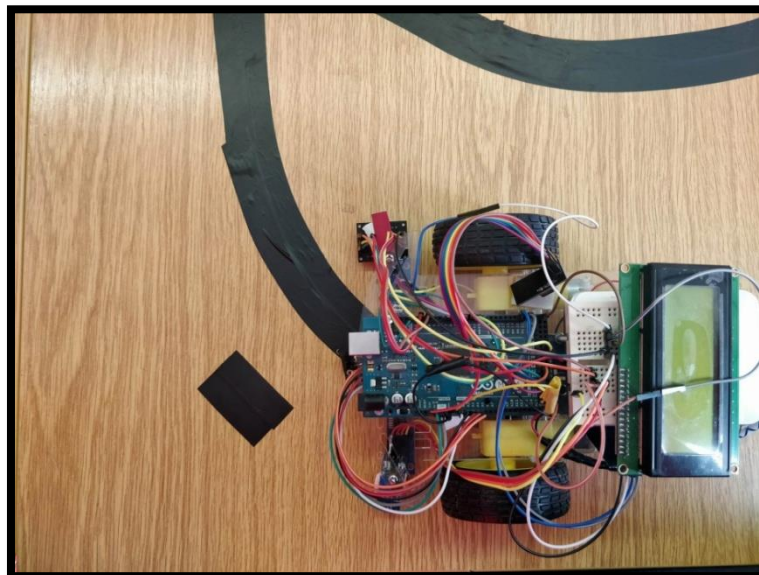


Figure 30- Stopping at delivery point

4.6- RGB led indication

The RGB LED is used to indicate the colour of the station at which the smart factory vehicle is waiting. It can be explained in other words as the indication for the colour which is detected by the colour sensor on the side of the track . This indication is just an information for the user to know what the vehicle is currently doing in the smart factory. It is the status of the vehicle in the smart factor which is just a visual indication .

4.6 – RFID reading

RFID belongs to a group of technologies referred to as Automatic Identification and Data Capture (AIDC). AIDC methods automatically identify objects, collect data about them, and enter those data directly into computer systems with little or no human intervention.

RFID methods utilize radio waves to accomplish this. At a simple level, RFID systems consist of three components: an RFID tag or smart label, an RFID reader, and an antenna. RFID tags contain an integrated circuit and an antenna, which are used to transmit data to the RFID reader (also called an interrogator). The reader then converts the radio waves to a more usable form of data. Information collected from the tags is then transferred through a communications interface to a host computer system, where the data can be stored in a database and analyzed at a later time.

Each of the vehicle in the smart factory has a separate tag which identifies the vehicle in the smart factory. There are many vehicles in the smart factory which are in operation. Each station in the smart factory has a RFID reader and when the vehicle reaches the station, it will detect the tag on the vehicle and identifies the vehicle separately by differentiating it from the others based on the tag. Thus it is very essential module used in the smart factory

5- Arduino code

Code for the Smart Factory vehicle

```
#include <SPI.h>
#include <LiquidCrystal.h>
#include "nRF24L01.h"
#include "RF24.h"
RF24 radio(40,53);
byte address[][6] = {"1Node","2Node","3Node","4Node","5Node","6Node"};
int L2=A0;
int L1=A1;
int M=A2;
int R1=A3;
int R2=A4;
int lMotFwd=4;
int rMotFwd=7;
int lMotRev=5;
int rMotRev=6;
#define pins0 3
#define pins1 2
#define pins2 12
#define pins3 11
#define pinOut 34
int singleIR = 28;
int i=0;
int pattern[5];
//LCD
const int rs = 14, en = 13, d4 = 18, d5 = 17, d6 = 16, d7 = 15;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
//LED
int redPin = 10;
int bluePin = 9;
int greenPin = 8;
int RED, GREEN, BLUE;
void followLine();
void motorForward();

void motorStop();
void findRed();
void findBlue();
void ledSetColor(int,int,int);
void calculateRGB();
```

```

void setup()
{
  pinMode(L2, INPUT); // Left most sensor
  pinMode(L1, INPUT); // Left sensor
  pinMode(M, INPUT); // Middle sensor
  pinMode(R2, INPUT); // Right sensor
  pinMode(R1, INPUT); // Right most sensor
  pinMode(singleIR, INPUT); // Single IR sensor
  pinMode(lMotFwd, OUTPUT); // Left forward motor
  pinMode(rMotFwd, OUTPUT); // Right forward motor
  pinMode(lMotRev, OUTPUT); // Left reverse motor
  pinMode(rMotRev, OUTPUT); // Right reverse motor
  digitalWrite(lMotFwd, LOW);
  digitalWrite(rMotFwd, LOW);
  digitalWrite(lMotRev, LOW);
  digitalWrite(rMotRev, LOW);
  lcd.begin(16, 2);
  //LED
  pinMode(redPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  //wifi
  radio.begin();
  radio.setAutoAck(1);
  radio.setRetries(0, 15);
  radio.enableAckPayload();
  radio.setPayloadSize(32);
  radio.openReadingPipe(1, address[0]);
  radio.setChannel(0x60);
  radio.setPALevel(RF24_PA_MAX);
  radio.setDataRate(RF24_250KBPS);
  radio.powerUp();
  radio.startListening();
  Serial.begin(9600);
}

void loop()
{
  radio.read(&pattern, sizeof(pattern));
  i=0;
  while( i < sizeof(pattern) )
  {
    if( pattern[i] == 1)

```

```

    {
        calculateRGB();
        findRed();
    }
    else if( pattern[i] == 2)
    {
        calculateRGB();
        findBlue();
    }
        followLine();
    }
    if(singleIR == LOW)
    {
        motorStop();
    }
    followLine();
}
void followLine()
{
    if( ((digitalRead(L2)==HIGH) && (digitalRead(L1)==LOW) &&
        (digitalRead(M)==HIGH) && (digitalRead(R1)==LOW) && (digitalRead(R2)==HIGH) )
        || ((digitalRead(L2)==LOW) && (digitalRead(L1)==HIGH) &&
        (digitalRead(M)==HIGH) && (digitalRead(R1)==HIGH) && (digitalRead(R2)==LOW) )
        || ((digitalRead(L2)==LOW) && (digitalRead(L1)==HIGH) &&
        (digitalRead(M)==HIGH) && (digitalRead(R1)==LOW) && (digitalRead(R2)==HIGH) )
        || ((digitalRead(L2)==HIGH) && (digitalRead(L1)==LOW) &&
        (digitalRead(M)==HIGH) && (digitalRead(R1)==HIGH) && (digitalRead(R2)==LOW) ) )
    {
        int gotByte = 0;
        radio.read( &gotByte, sizeof(gotByte) );
        Serial.println(gotByte);


---


        if(i<sizeof(pattern))
        {
            gotByte = 2;
        }
        if(gotByte==1)
        {
            Serial.println("Im in gotbyte 1");
            digitalWrite(lMotFwd,160);
            digitalWrite(rMotFwd,LOW);
            digitalWrite(lMotRev,LOW);
            digitalWrite(rMotRev,LOW);
            delay(100);
        }
    }
}

```

```

        else if(gotByte==2)
    {
        Serial.println("Im in gotbyte 2");
        digitalWrite(lMotFwd,LOW);
        digitalWrite(rMotFwd,160);
        digitalWrite(lMotRev,LOW);
        digitalWrite(rMotRev,LOW);
        delay(100);
    }
    else
    {
        motorStop();
    }
    gotByte = 0;
}
//Move forward
else if(((digitalRead(L2)==HIGH) && (digitalRead(L1)==HIGH) &&
(digitalRead(M)==LOW) && (digitalRead(R1)==HIGH) && (digitalRead(R2)==HIGH) )
|| ((digitalRead(L2)==HIGH) && (digitalRead(L1)==LOW) &&
(digitalRead(M)==LOW) && (digitalRead(R1)==LOW) && (digitalRead(R2)==HIGH) ))
{
    motorForward();
}
//Slight left
else if(((digitalRead(L2)==HIGH) && (digitalRead(L1)==LOW) &&
(digitalRead(M)==LOW) && (digitalRead(R1)==HIGH) && (digitalRead(R2)==HIGH) )
|| ((digitalRead(L2)==HIGH) && (digitalRead(L1)==LOW) &&
(digitalRead(M)==HIGH) && (digitalRead(R1)==HIGH) && (digitalRead(R2)==HIGH) ))
{
    digitalWrite(lMotFwd,160);
    digitalWrite(rMotFwd,LOW);
    digitalWrite(lMotRev,LOW);
    digitalWrite(rMotRev,LOW);
}
//Slight right
else if(((digitalRead(L2)==HIGH) && (digitalRead(L1)==HIGH) &&
(digitalRead(M)==LOW) && (digitalRead(R1)==LOW) && (digitalRead(R2)==HIGH) )
|| ((digitalRead(L2)==HIGH) && (digitalRead(L1)==HIGH) &&
(digitalRead(M)==HIGH) && (digitalRead(R1)==LOW) && (digitalRead(R2)==HIGH) ))
{
    digitalWrite(lMotFwd,LOW);
    digitalWrite(rMotFwd,160);
    digitalWrite(lMotRev,LOW);
    digitalWrite(rMotRev,LOW);
}
}

```

```

//Sharp right
else if(((digitalRead(L2)==HIGH) && (digitalRead(L1)==HIGH) &&
(digitalRead(M)==LOW) && (digitalRead(R1)==LOW) && (digitalRead(R2)==LOW) )
||((digitalRead(L2)==HIGH) && (digitalRead(L1)==HIGH) &&
(digitalRead(M)==HIGH) && (digitalRead(R1)==LOW) && (digitalRead(R2)==LOW) )
||((digitalRead(L2)==HIGH) && (digitalRead(L1)==LOW) &&
(digitalRead(M)==HIGH) && (digitalRead(R1)==LOW) && (digitalRead(R2)==LOW) )
||((digitalRead(L2)==HIGH) && (digitalRead(L1)==HIGH) &&
(digitalRead(M)==HIGH) && (digitalRead(R1)==HIGH) && (digitalRead(R2)==LOW) )
||((digitalRead(L2)==HIGH) && (digitalRead(L1)==LOW) &&
(digitalRead(M)==LOW) && (digitalRead(R1)==LOW) && (digitalRead(R2)==LOW) ) )
{
    analogWrite(lMotFwd,LOW);
    digitalWrite(rMotFwd,80);
    digitalWrite(lMotRev,LOW);
    digitalWrite(rMotRev,LOW);
}
//Sharp left
else if(((digitalRead(L2)==LOW) && (digitalRead(L1)==LOW) &&
(digitalRead(M)==HIGH) && (digitalRead(R1)==HIGH) && (digitalRead(R2)==HIGH) )
||((digitalRead(L2)==LOW) && (digitalRead(L1)==LOW) &&
(digitalRead(M)==LOW) && (digitalRead(R1)==HIGH) && (digitalRead(R2)==HIGH) )
||((digitalRead(L2)==LOW) && (digitalRead(L1)==LOW) &&
(digitalRead(M)==LOW) && (digitalRead(R1)==LOW) && (digitalRead(R2)==HIGH) )
||((digitalRead(L2)==LOW) && (digitalRead(L1)==LOW) &&
(digitalRead(M)==HIGH) && (digitalRead(R1)==LOW) && (digitalRead(R2)==HIGH) )
||((digitalRead(L2)==LOW) && (digitalRead(L1)==HIGH) &&
(digitalRead(M)==HIGH) && (digitalRead(R1)==HIGH) && (digitalRead(R2)==HIGH) ) )
{
    digitalWrite(lMotFwd,80);
    digitalWrite(rMotFwd,LOW);
    digitalWrite(lMotRev,LOW);
    digitalWrite(rMotRev,LOW);
}
else
{
    motorStop();
}
delay(1);
}
void findRed()
{
    if(RED<60 && RED>30 && GREEN<190 && GREEN>120 && BLUE<155 && BLUE>100)
    {
        Serial.println("RED ");
        motorStop();
    }
}

```



```

    i++;
    lcd.setCursor(0, 0);
    lcd.print("On Red");
    ledSetcolor(255,0,0);
    delay(5000);
    while (RED<60 && RED>30 && GREEN<210 && GREEN>100 && BLUE<190 && BLUE>100)
    {
        motorForward();
        calculateRGB();
    }
}
}
void findBlue()
{
    if (RED<220 && RED>180 && GREEN<200 && GREEN>155 && BLUE<145 && BLUE>110)
    {
        Serial.println("BLUE ");
        motorStop();
        i++;
        lcd.setCursor(0, 0);
        lcd.print("On Blue");
        ledSetcolor(0,0,255);
        delay(5000);
        while (RED<220 && RED>180 && GREEN<220 && GREEN>155 && BLUE<145 && BLUE>110)
        {
            motorForward();
            calculateRGB();
        }
    }
}
void ledSetcolor(int red, int green, int blue)
{
    #ifdef COMMON_ANODE
        red = 255 - red;
        green = 255 - green;
        blue = 255 - blue;
    #endif
    analogWrite(redPin, red);
    analogWrite(greenPin, green);
    analogWrite(bluePin, blue);
}
void motorStop()
{
    digitalWrite(lMotFwd, LOW);
    digitalWrite(rMotFwd, LOW);
    digitalWrite(lMotRev, LOW);
    digitalWrite(rMotRev, LOW);
}
}

```

```
void motorForward()  
{  
    digitalWrite(lMotFwd,HIGH);  
    digitalWrite(rMotFwd,HIGH);  
    digitalWrite(lMotRev,LOW);  
    digitalWrite(rMotRev,LOW);  
}
```

6- Conclusion

The control system for the vehicle of smart factory was achieved with the use of various modules in connection with the arduino mega2560 as the brain of the vehicle. The smart factory vehicle was made to communicate with the other members of the smart factory with the help of Wi-Fi module. This was a prototype of the real idea of a smart factory vehicle to check its functions. The control system of the vehicle tracks the vehicle of its position without any deviation. The control system is very accurate in its function of maintaining the navigation on the track specified for it. It has been equipped with several sensors such as tracking sensor, colour sensor, single position sensor. All these sensors function to make the smart factory vehicle navigate in the smart factory with a proper control system. The case where many such smart vehicles are used, each vehicle can be uniquely identified with the help of onboard RFID tag. This smart factory vehicle can be made to transport small objects without human interference and the whole process happens automated. The vehicle is flexible to navigate according to changes in the requirement of the factory. The vehicle can be calibrated to changes in conditions in the factory such as colour and track thickness. This advantage makes the smart factory vehicle have diverse applications in different factories.

Scope of the thesis:

The smart factory vehicle can be further aimed at having extended functions which may be specific as per requirements of a factory.

- As advanced navigation techniques are being invented, we can take one step further by implementing it to transport people between shop floors of a factory
- It can be used as a mobility solution in real life where it can replace trams. It also benefits from avoiding extra costs such as laying the track.
- Artificial intelligence can be used to carry out operations of the smart factory which can supplement Wi-Fi communication.

7- References

- [1] i-SCOOP. (2018). Industry 4.0: the fourth industrial revolution - guide to Industrie 4.0. [online] Available at: <https://www.i-scoop.eu/industry-4-0/> [Accessed 6 Jan. 2018].
- [2] Issuu. (2018). Industry 4.0: an overview from the perspective of a German-headquartered firm (1.st Part). [online] Available at: https://issuu.com/cie-comunicacao/docs/art_12-16 [Accessed 11 Jan. 2018].
- [3] Apics.org. (2018). Industry 1.0 to 4.0: the Evolution of Smart Factories | APICS Magazine. [online] Available at: <http://www.apics.org/apics-for-individuals/apics-magazine-home/magazine-detail-page/2017/09/20/industry-1.0-to-4.0-the-evolution-of-smart-factories> [Accessed 9 Jan. 2018].
- [4] Geissbauer, D., Vedso, J. and Schrauf, S. (2018). Industry 4.0: Building the digital enterprise. [online] Pwc.com. Available at: <https://www.pwc.com/gx/en/industries/industries-4.0/landing-page/industry-4.0-building-your-digital-enterprise-april-2016.pdf> [Accessed 22 Jan. 2018].
- [5] Bajdi.com. (2018). L9110 H-bridge module | Bajdi.com. [online] Available at: <http://www.bajdi.com/l9110-h-bridge-module/> [Accessed 10 Feb. 2018].
- [6] Bananarobotics.com. (2018). How to use the HG7881 (L9110) Dual Channel Motor Driver Module. [online] Available at: [https://www.bananarobotics.com/shop/How-to-use-the-HG7881-\(L9110\)-Dual-Channel-Motor-Driver-Module](https://www.bananarobotics.com/shop/How-to-use-the-HG7881-(L9110)-Dual-Channel-Motor-Driver-Module) [Accessed 11 Feb. 2018].
- [7] Makerlab-electronics.com. (2018). 5 Channel Tracking Sensor Module – Infrared Tracking Sensor Philippines | Makerlab Electronics. [online] Available at: <https://www.makerlab-electronics.com/product/5-channel-tracking-sensor-module-infrared-tracking-sensor/> [Accessed 2 Apr. 2018].
- [8] HowToMechatronics. (2018). How to use a RGB LED with Arduino | Tutorial. [online] Available at: <https://howtomechatronics.com/tutorials/arduino/how-to-use-a-rgb-led-with-arduino/> [Accessed 5 Mar. 2018].
- [9] Navody.arduino-shop.cz. (2018). LCD Displej | Arduinonávody. [online] Available at: <http://navody.arduino-shop.cz/zaciname-s-arduinem/lcd-displej.html> [Accessed 27 Feb. 2018].
- [10] Navody.arduino-shop.cz. (2018). Arduinodetektorbarvy TCS230 | Arduinonávody. [online] Available at: <http://navody.arduino-shop.cz/navody-k-produktum/arduino-detektor-barvy.html> [Accessed 10 Feb. 2018].

- [11]Navody.arduino-shop.cz. (2018). ArduinoWi-Fimodul NRF24L01 | Arduinonávody. [online] Available at: <http://navody.arduino-shop.cz/navody-k-produktum/arduino-Wi-Fi-modul-nrf24l01.html> [Accessed 18 Apr. 2018].
- [12]Mantech.co.za. (2018). Arduino mega2560. [online] Available at: <http://www.mantech.co.za/datasheets/products/A000047.pdf> [Accessed 16 Mar. 2018].
- [13]Create.arduino.cc. (2018). Arduino Create. [online] Available at: <https://create.arduino.cc/> [Accessed 4 Apr. 2018].
- [14]Robotshop.com. (2018). Arduino mega 2560 datasheet. [online] Available at: <https://www.robotshop.com/media/files/pdf/arduinomega2560datasheet.pdf> [Accessed 6 Apr. 2018].