OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

INF FAKULTÄT FÜR INFORMATIK

Scientific Individual Project

On

Development of a control algorithm for an Automated Guided Vehicle (AGV)

By

"Vasu dev Mukku"

Under the Supervision of

Dr.-Ing. Tobias Reggelin

M.Sc. Sebastian

Abstract

This project is designed to build a prototype of Automated Guided Vehicle (AGV) in smart industries to serve the conveyor systems inside the production plant. The conveyor systems use Bluetooth communication for accessing the AGV. It can select the path based on the command received from the Central Master where all conveyor systems are connected. It can able choose the path, follow the path and able to navigate to the start point. It uses IR communication to decode the commands from Central Master. So, the vehicle can translate the Bluetooth commands to IR data using another controller mounted on top of it. The AGV has Arduino Leonardo controller with integrated IR line following sensors and IR receiver, to translate Bluetooth commands to IR data Arduino UNO is used on top of vehicle. Central Master has push button switches and Arduino UNO and Bluetooth Master Module HC-05.

Keywords: AGV, Arduino UNO, Bluetooth module HC-05, IR

Contents

List of Figures

1. Introduction

In fast growing Automation technology the world is moving towards automation. For instance, from toys to surgical operations everything is automated. Now the world is moving towards smart industries. There are so many advanced technologies evolving to build smart industries. Especially, in production industry companies are moving towards integrating new technologies in production plants.

In Production plant raw materials, intermediate and final products need to transport from one place to another place inside the plant. Pallets are used for transport products from one conveyor system to another conveyor system and so on. For transporting pallets from storage location to conveyor system require an intelligent vehicle to serve all conveyor systems in smart industry.

Therefore, the central idea behind this project is to build a smart vehicle which can be accessed by all conveyor systems. The vehicle is able to communicate with the conveyor systems and should provide the required service. This intelligent vehicle can able to communicate with the conveyor system and can able to detect obstacles in the path. It has to serve the Conveyor system to get the empty pallets from storage location.

Furthermore, this intelligent vehicle uses different types of communication protocols and sensors to provide an automated service. This vehicle can be named as Automated Guided Vehicle (AGV). In this project it uses IR sensors for line following, obstacle detection and communication. Bluetooth communication is used from conveyor system to AGV.

2. Conceptual Model

Before implement of the original product better to have a conceptual design to drive the implementation towards success. Here is the conceptual design of AGV and path prototype.

2.1 Conceptual Design of AGV:

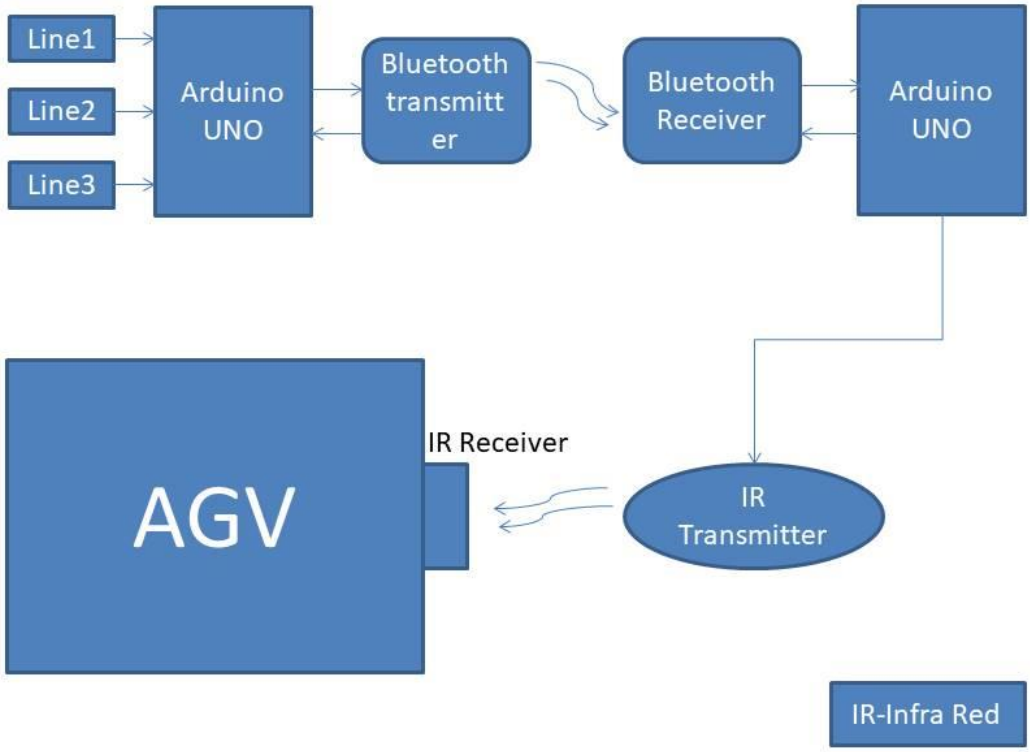

Fig.2.1 Conceptual design of AGV

The AGV has to communicate with the Conveyor systems (line1,line2,line3) which are connected to Master controller(Arduino UNO)and Bluetooth transmitter module. There is Bluetooth receiver module and Slave controller on top of AGV to receive the commands from Master controller. The received Bluetooth commands are translate into IR data to communicate with the AGV.

2.2 Conceptual Design of path prototype:

Currently, the conveyor systems are able to communicate with the AGV. This is the conceptual design for the AGV to find path to serve particular conveyor system and to return to start point.

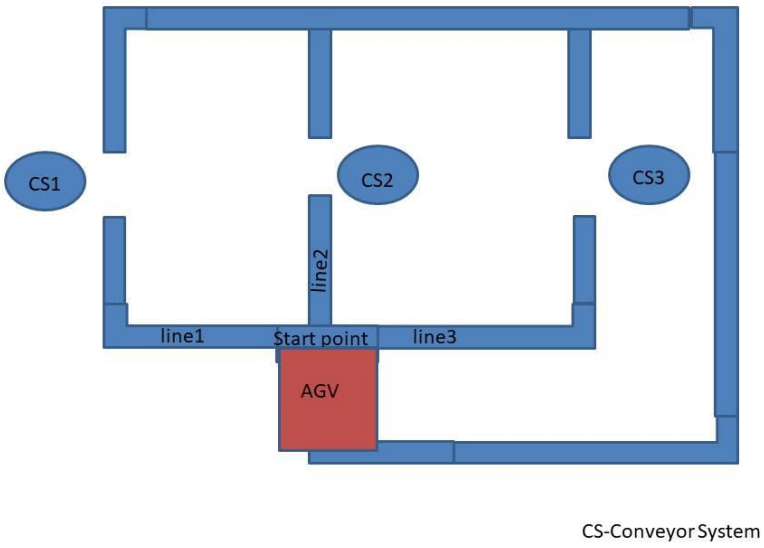

Fig.2.2 Conceptual Design for Path prototype

Based on the command received from the Master controller the AGV will follow the line accordingly.

3. Simulation

The Proteus Design Suite is a software tool from Labcenter electronics for electronic design automation. This tool can be used to design schematics, Microcontroller simulation, create electronic design for Printed Circuit Boards (PCB) and 3D verification.

3.1 Proteus Design Suite 8.6:

This is a nice software tool to perform microcontroller simulation. In this tool for simulating Arduino[2] and Bluetooth[3] need to add their libraries and all components required are available in Proteus library.

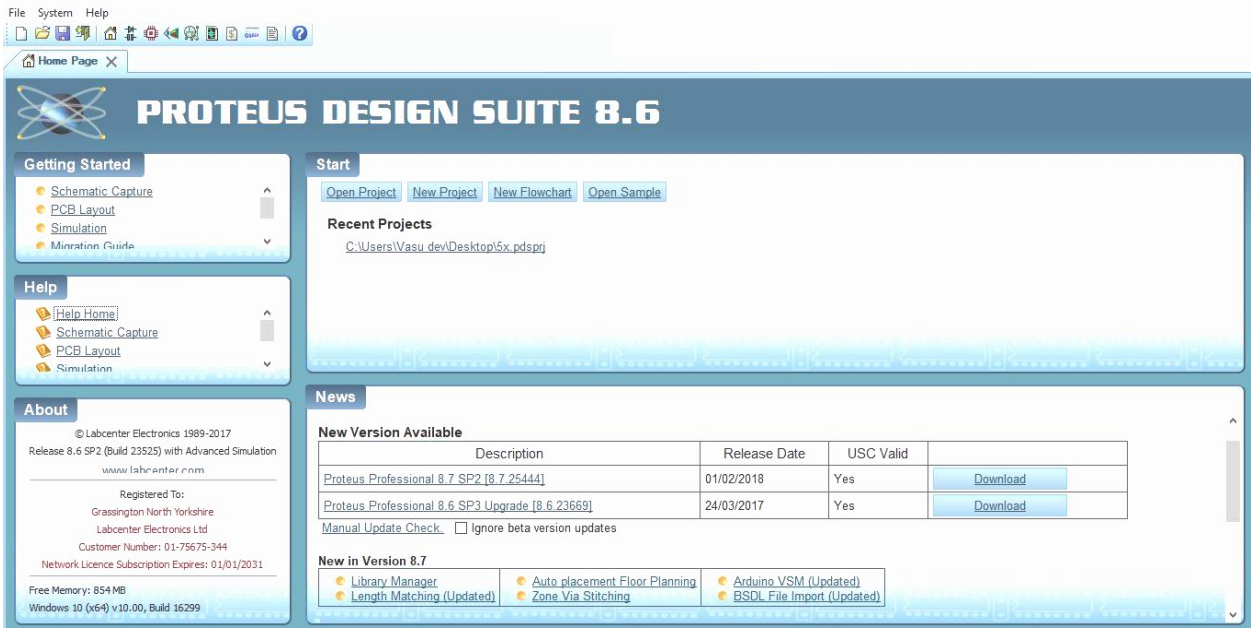

Fig.3.1 Proteus Design Suite 8.6[1]

3.2 Design of Bluetooth Transmitter in Proteus:

In this section Bluetooth is used for transmitter. It requires Arduino UNO, push buttons, Bluetooth module, pull-up resistors.

Procedure:

- Add Arduino UNO sketch from Arduino library.
- Add Bluetooth from Bluetooth library.
- Set Bluetooth port number as COM1.
- Add push buttons and pull-up resistors from component library.
- Add Vcc and ground from terminal mode.
- Write the transmitter code in Arduino IDE with UNO as board.
- Compile code and create HEX file.

- Connect the circuit as shown in figure.
- Upload HEX file on UNO.
- Now the transmitter is able to send data.

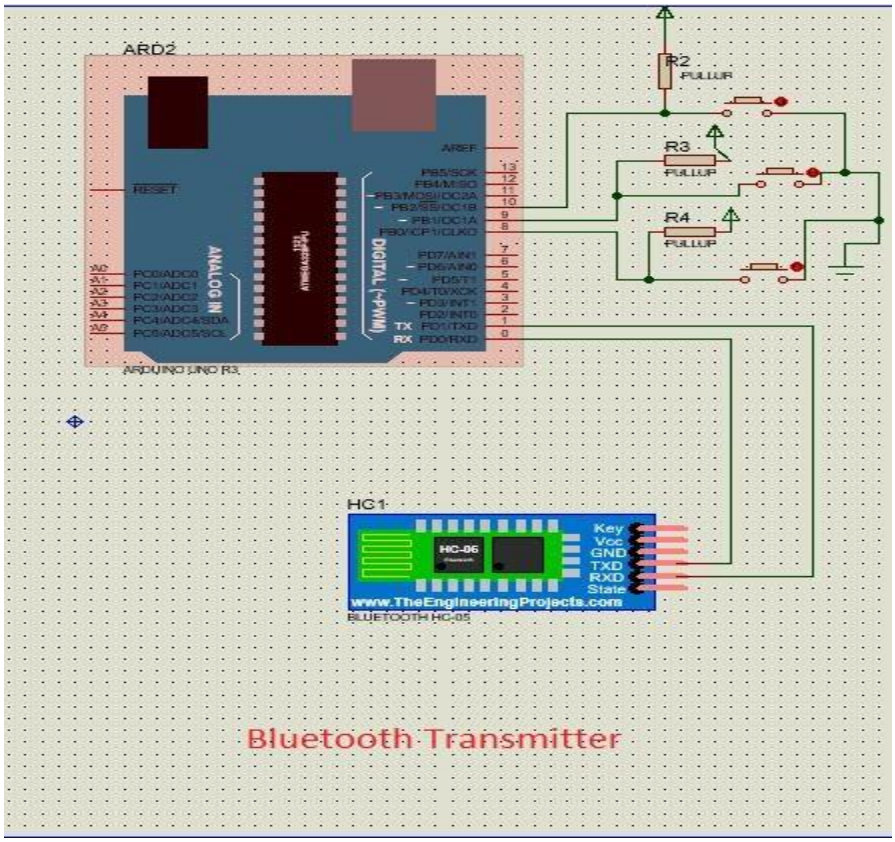

Fig.3.2 Proteus design of Bluetooth transmitter

3.3 Design of Bluetooth Receiver in Proteus:

In this section Bluetooth is used as receiver. It requires Arduino UNO, Bluetooth module, LEDs, resistors, IR LED.

Procedure:

- Follow same procedure for adding components as described in Transmitter section.
- Set Bluetooth port number as COM2.
- Write receiver code in Arduino IDE as Arduino UNO as board.
- Compile and create HEX file.
- Connect the circuit as shown in figure.
- Upload HEX file on Receiver UNO board.
- Now pair both Bluetooth modules using VSPE(Virtual Serial Port Emulator)[4].
- Now both Transmitter and Receiver are ready to communicate.

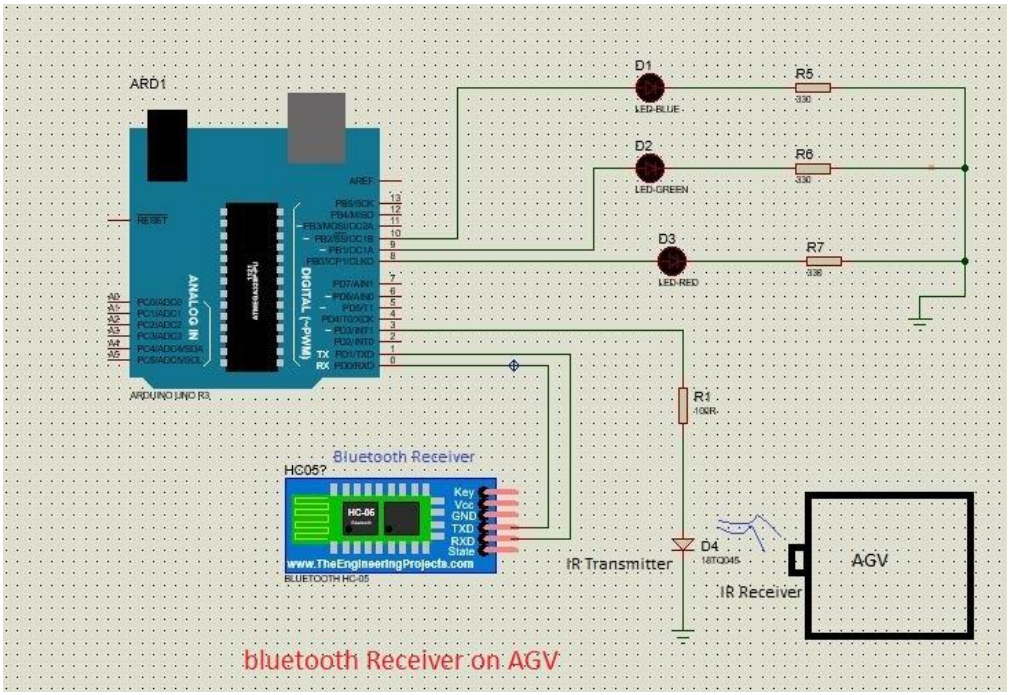

Fig.3.3 Proteus design of Bluetooth receiver

4. Implementation

After successful verification of simulation, now time to implementation. Before that here is the small description about software, hardware tools and electronic components[9] used.

4.1 Software and Hardware Tools:

Arduino IDE:

Arduino 1.8.4 IDE [5] is the open-source software used to write, compile, verify and upload the code on any Arduino board. It has all inbuilt libraries and can add the required libraries to the sketch.

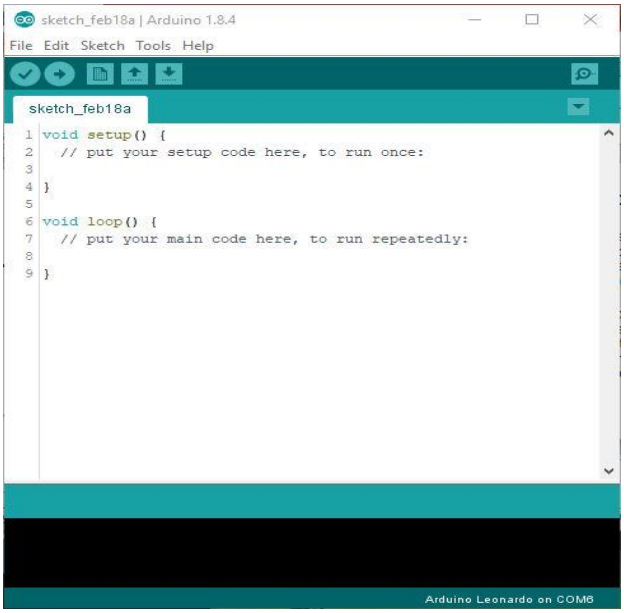

Fig.4.1.1 Arduino 1.8.4 IDE

Arduino UNO:

This can be programmed with the Arduino IDE using USB cable. This board comes with ATmega328 processor. In this project two UNO boards are used for transmitter and receiver section.

Fig.4.1.2 Arduino UNO[6]

Bluetooth module (HC-05):

This is an easy to use Serial Port Protocol (SPP) module, can be used for transparent wireless serial communication. It is simple to interface with the controller and pairing with other Bluetooth modules. In this project two HC-05[7] modules are used as transmitter and receiver. The transmitter module is acts as master and receiver acts as slave.HC-05 has pairing mode and command mode.

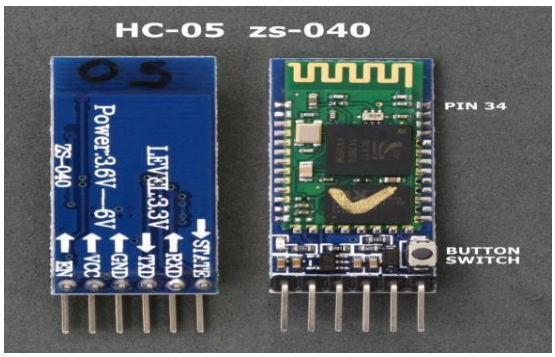

Fig.4.1.3 Bluetooth Module (HC-05)

By default, HC-05 is in pairing mode, set it in command mode by following procedure

- Label the HC-05 modules as Master and slave
- Upload an empty sketch on Arduino UNO.
- Unplug the power source from UNO and connect the HC-05 module to the board.
- Connections of HC-05 and UNO shown in figure below.

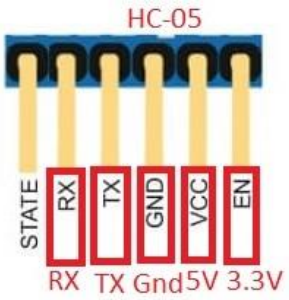

Fig.4.1.4 HC-05 to UNO connections

- Now power on the UNO while press the button switch on HC-05.
- Now HC-05 enter in to AT command mode
- Open the serial monitor on Arduino IDE and set Baud rate as 38400.
- Type the AT commands to configure as Master and slave.

Configuring HC-05 as Slave:

- AT
- AT+ROLE? (initially HC-05 acts as slave)
- +ROLE=0 as slave
- +ROLE=1 as master
- AT_ADDR? (to find address of HC-05)
- AT+ADDR=98d3:31:fb4487

Configuring HC-05 as Master:

- AT
- AT+ROLE?
- +ROLE=0 as slave
- +ROLE=1 as master
- AT+ROLE=1 (to set HC-05 as master)
- AT+CMODE? (to check connection mode)
- CMODE:0-Connect to one
- CMODE:1-connect to any
- AT+CMODE=0 (to connect master to one slave)
- AT+BIND=98d3,31,fd4487 (connected to slave)

Now both HC-05 acts as master and slave respectively. Connect Tx pin of HC-05 to Rx pin of UNO and Rx pin of HC-05 to Tx pin of UNO for establishing communication.

IR Transmitting LED:

IR(Infrared) LED can be used for transmitting IR data. It operates in 950nm of 38 KHz operating frequency. As the receiver on AGV is of same operating frequency. IR communication [8] is simple and can be seen in TV remote controller and can also be used for object sensing.

Fig. 4.1.5 IR transmitting LED[12]

Pushbutton switches:

These small Single Pole Single Throw switches are momentary buttons can be used send state to Arduino as input signal. It can also be used as reset button.

Fig. 4.1.6 pushbutton[13]

LEDs:

These are used for indicating current status of AGV.

Fig.4.1.7 LED[14]

Connecting wires and Resistors:

These are used for connecting modules to boards, jumpers and components. Resistors are used as potential divider, pull-up and pull-down.

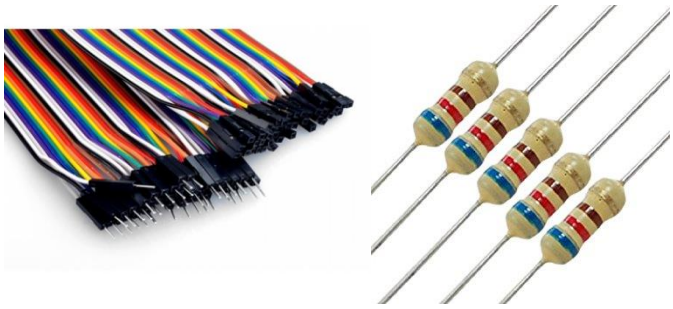

Fig.4.1.8 connecting wires[15] and resistors[16]

Batteries:

To power up an Arduino UNO one 600mAh 9V battery is used.

Fig.4.1.9 Batteries[17]

4WD MiniQ Arduino Robot V2.0:

This Robot comes with Arduino Leonardo controller and has different features Line following, obstacle avoidance, remote control. It has all sensors integrated on single chip. The complete information about this robot is available in this reference[10].

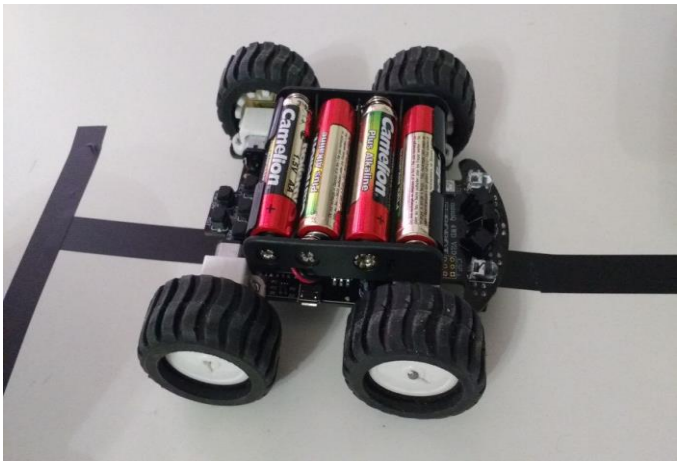

Fig.4.1.10 4WD MiniQ Arduino Robot V2.0

4.2 Implementation of Transmitter section:

This section contains Master controller, push button switches, 9V battery and Master Bluetooth module located at conveyor systems**.** Connect the circuit as shown in Fig.3.2.

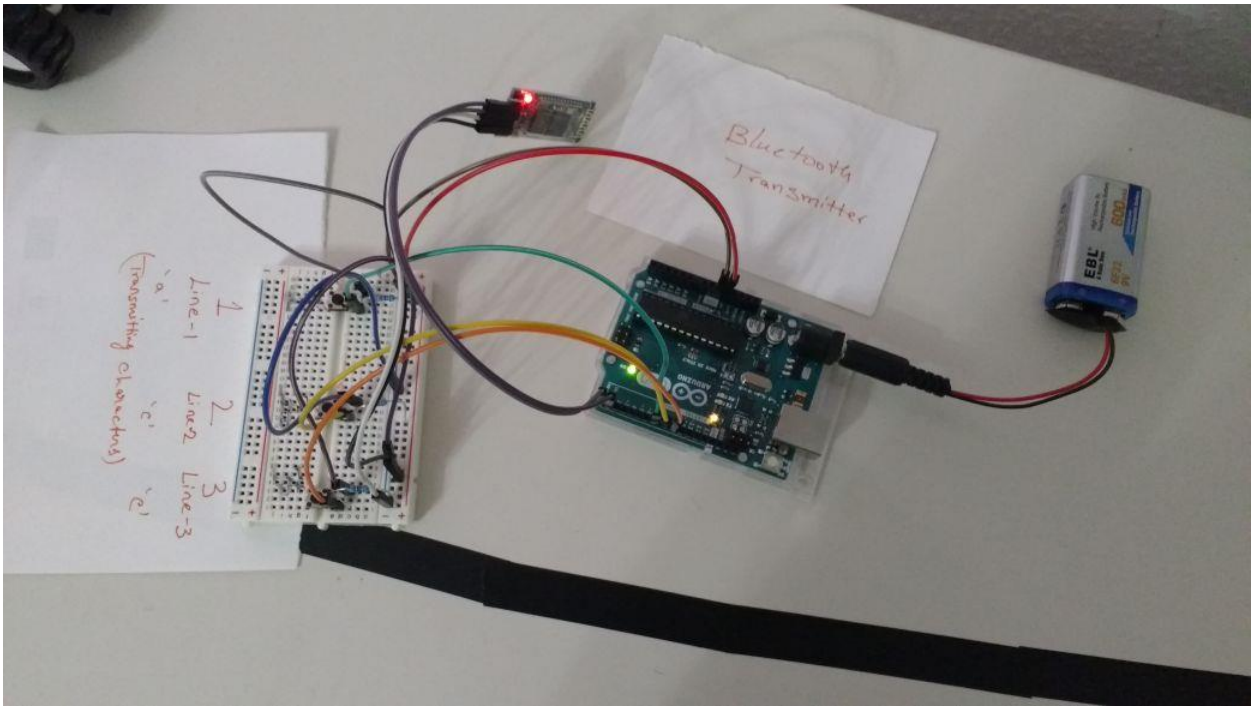

Fig.4.2.1 Transmitter section

Conveyor system can send the commands to AGV by pressing corresponding push button. Master Bluetooth module send the command based on button pressed.

Algorithm for Transmitter:

The transmitter section works based on following algorithm

- when button pressed, the status of button is given to Master controller.
- Master controller continuously reads input from buttons.
- It process the input and send the commands to Bluetooth module.
- Bluetooth module sends the characters according to input.
- Complete Transmitter code is available in this reference[11]

```
void setup() {
  Serial.begin(9600);// Default communication rate of the Bluetooth module
  //setting lines as inputs to Arduino UNO
  pinMode(line1, INPUT);
  pinMode(line2, INPUT);
  pinMode(line3, INPUT);

}
void loop() {
 // Reading the buttons status
 value1= digitalRead(line1);
 value2= digitalRead(line2);
 value3= digitalRead(line3);
//based on status of button performs operation
 if (value1 == LOW) {
  delay(350);//response delay of transmission
   Serial.write('a'); //when line1 is active then UNO transmits character 'a' through HC-05 module
 }
 if (value2==LOW){
  delay(350);//response delay of transmission
   Serial.write('c'); //when line2 is active then UNO transmits character 'a' through HC-05 module
 }
 if (value3==LOW){
  delay(350);//response delay of transmission
   Serial.write('e');//when line3 is active then UNO transmits character 'a' through HC-05 module
 }
}

```

Fig.4.2.2 Transmitter Algorithm

4.3 Implementation of Receiver section:

This section is mounted on top AGV, it contains Arduino UNO, Slave Bluetooth module, status LEDs, IR transmitting LED,9V battery and 100 ohm resistor. Connect the circuit as shown in Fig.3.3. The importance of this section is receiving the Bluetooth data from transmitter section and translate to IR packets to communicate with AGV. It uses IRremote library[11] for IR communication.

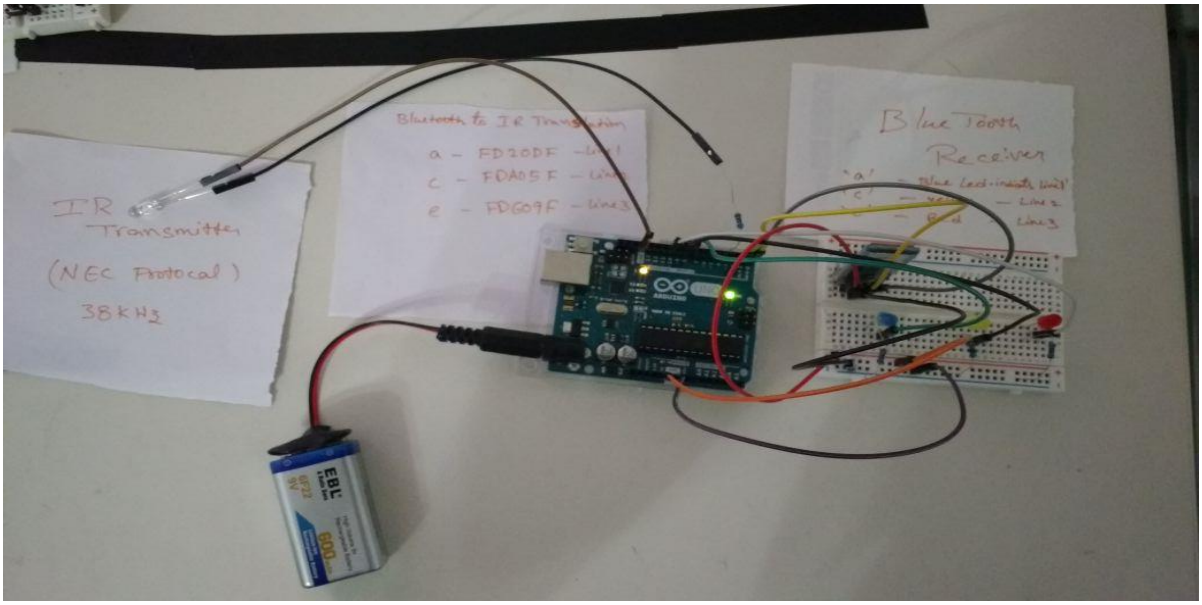

Fig.4.3.1 Receiver section

Algorithm for Receiver:

The Receiver section works based on following algorithm

- Slave Bluetooth module receives the commands from the Master module.
- Process the received command and translates it into' IR Data' based on NEC remote Protocol.
- It uses IRremote library[11] to translate the data from bluetooth module.
- NEC protocol[8] is used because the IR receiver on AGV works with that protocol.
- NEC is 32 bit data format(Ex:0x00fd20df).
- IR receiver on AGV can understand this format.
- For each Bluetooth command there is unique IR data packet is transmitted by using IR transmitting LED.
- Status LEDs provide the line status of AGV.
- Complete Receiver code is available in this reference[11].

```
void loop() {

  //checking for serial data availability
if(Serial.available()>0)
   {
      char data= Serial.read(); // reading the data from the bluetooth module
      switch(data)
      {
        //when character 'a' recevied then it translates the character into IR packet based on NEC protocol
        case 'a':
               digitalWrite(line1_status, HIGH);
               Serial.println("line1 is active");
               delay(50);
               irsend.sendNEC(0x00fd20df, 32);// sends IR packet '0x00fd20df'
               delay(500);
               digitalWrite(line1_status,LOW);
               break;
      //when character 'a' recevied then it translates the character into IR packet based on NEC protocol
       case 'c':
               digitalWrite(line2_status, HIGH);
               Serial.println("line2 is active");
               delay(50);
               irsend.sendNEC(0x00fda05f, 32);//sends IR packet '0x00fda05f'
               delay(500);
               digitalWrite(line2_status,LOW);
               break;
```

Fig.4.3.2 Receiver Algorithm

4.4 Implementation of AGV:

Now AGV can able to receive the commands from Master controller. The AGV has to perform line following, obstacle detection and serve the conveyor systems based on the commands received. The conveyor systems can communicate with AGV to perform their tasks, so the AGV has to respond conveyor systems by serving the desired task. The experimental setup is as shown in Fig.4.4.4. Lines are indicated by status LEDs. Line 1 blue, line 2 yellow, line 3 red.

Algorithm for AGV:

In this program three algorithms are implemented to serve the purpose, waiting for command from Central Master, Decoding the command and selecting the line, line follow and obstacle avoidance.

Waiting for command from Central Master:

- AGV always in reception mode
- once command received, it will serve the task
- and again in reception mode.

Decoding the command and selecting the line:

- It uses IRremote library[11] to decode the data received
- if it receives “0x00fd20df” then line 1 is selected.
- if it receives “0x00fda05f “ then line 2 is selected
- if it receives “0x00fd609f “ then line 3 is selected

```
void loop() {
  //Receives IR data from transmitter and decode the data received
  if (irrecv.decode(&results)) {
    Serial.println(results.value, HEX);
     irrecv.resume();
  }
  if (results.value == 0x00fd20df) { //select line 1
 turn_left();
   line_follow(1);//line following
    results.value = 0x00000000;
  }
  if (results.value == 0x00fda05f) { //select line 2
   go_forward();
   line_follow(1);
    results.value = 0x00000000;
  }
  if (results.value == 0x00fd609f) { //select line 3
    turn_right();
    line_follow(1);
    results.value = 0x00000000;
  }
}
```

Fig.4.4.1 Algorithm for selecting line

Line following and obstacle avoidance:

- it has five line sensing detectors, detects black and white lines
- when middle sensor on black line it will follow the black line
- adjacent sensors will make the AGV to follow without deviation.
- if there is an obstacle it has to move away from line and circumnavigate the obstacle.
- and it will follow the line again until it reaches destination.
- complete code is available in this reference[11].

```
//Line following function
void line_follow(int line)// black return value is less than 700, white is greater than 800
{
  int stopped=0;
   while(line==1 && stopped!=1)
   {
     Read_Value();// reading line follow sensor values
 if((irSensorAnalog[0]<650 || irSensorAnalog[1] <650) && (irSensorAnalog[3]>800 && irSensorAnalog[4]>800))//Black line on the left
  {
    // Serial.println("right");
    Motor(Forward,0,Forward,80);//turn right
    delay(10);
  }
  else  if((irSensorAnalog[3]<650 || irSensorAnalog[4] <650) && (irSensorAnalog[0]>800 && irSensorAnalog[1]>800))//The right black line
  {
    //Serial.println("left");
    Motor(Forward,80,Forward,0);//Turn left
    delay(10);
  }
  else if(irSensorAnalog[2]<700)//Middle black line
  {
    //Serial.println("forward");
    Motor(Forward,30,Forward,30);//carried out
    delay(20);
  }
```

Fig.4.4.2 Algorithm for line follow

```
void loop() {
 int ll=digitalRead(RECV_PIN);// reads obstacle detecor sensor
 if (ll==0)// obstacle detected
 {
  Obstacle_Avoidance();//avoid obstacle
  }
  else
  {
   line_follow();// follow the line
  }
}
```

```
void Obstacle_Avoidance()
{
  stop_a();
  turn_right();
  stop_a();
  go_forward();
  turn_left();
  stop_a();
  go_forward();
  stop_a();
  turn_left();
  go_forward();
  turn_right();
  }
```

Fig.4.4.3 Obstacle detection and avoidance

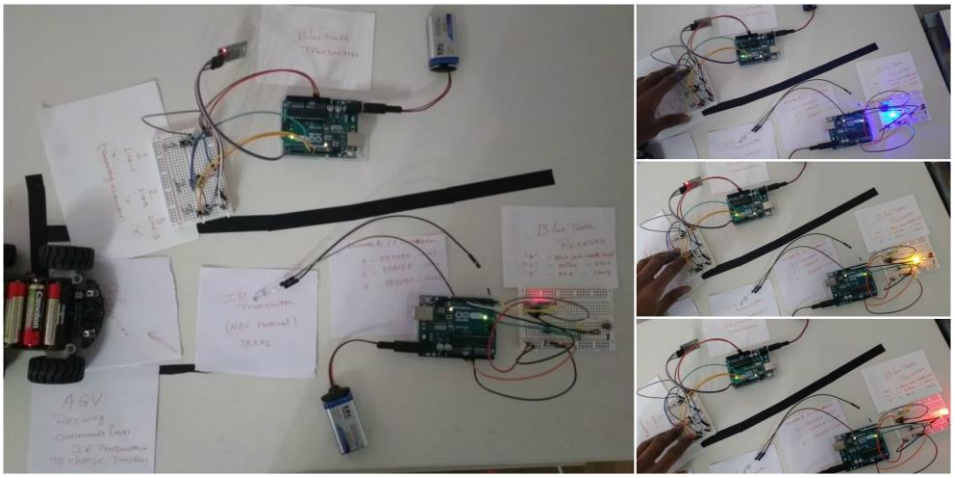

Fig.4.4.4 Complete setup of AGV

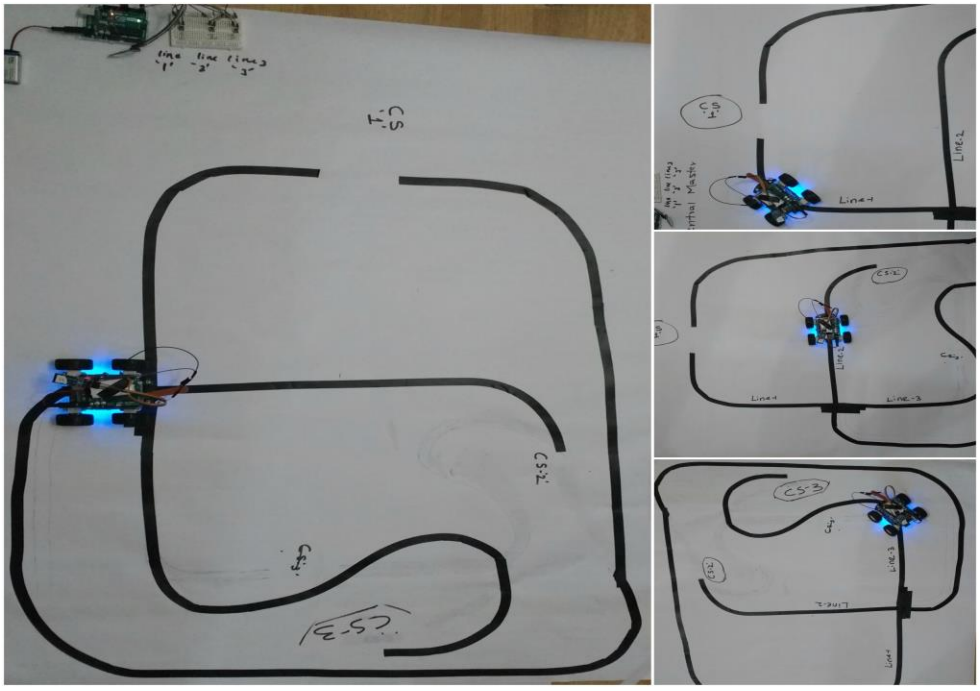

Fig.4.4.5 Complete path and start point navigation

The AGV comes with 4 dc motors the speed of the motors are defined in the AGV_ Final_Receiver program[11] to follow the line.

5. Design Problems and Limitations

5.1 Design Problems:

As the 4WD MiniQ Robot comes with integrated single chip controller, there is no possibility to extend the functionality by integrating other modules. This is one of the bottlenecks of this project. As it is not high end device the precession of the sensors are not good. There is problem with obstacle detection, as IR communication and obstacle detector uses single sensor. For obstacle detection IR has low range, it is restricted to very small distances.

There are two different approaches to find the start point navigation. One approach is to take the longest path as shown in the Fig.4.4.5. Second approach is to navigate from the same path shown in Fig.5.1.The second approach is failed to navigate the start point.

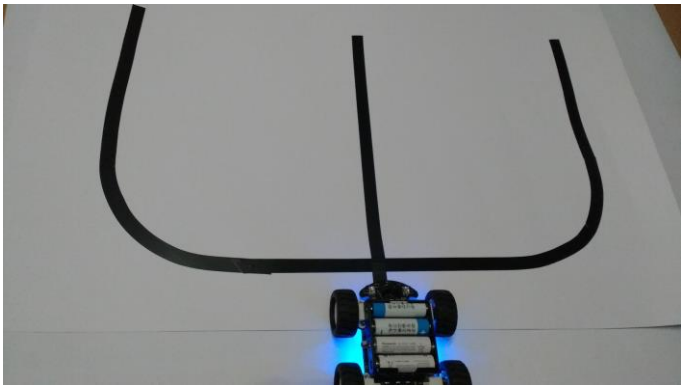

Fig.5.1 Start point navigation from same path

5.2 Limitations:

Power supply is the bottleneck for every robotic vehicle. The AGV has four 1.5v batteries to power up. As, the motors take more current batteries are drain quick.

6. Conclusion

In conclusion, the AGV can able to receive the commands from the Central Master. It follows the line according to the received command. It serves the conveyor system and can able to navigate to the start point. The translation of Bluetooth data to IR data is achieved. The conveyor systems can able to access the AGV with Bluetooth transmitter.

7. Future work

The AGV has wide scope of improvement in obstacle avoidance by using Ultrasonic sensors. These sensors can able to calculate the exact distance and can be more precise. It can be improve the exact tracking of line after detecting obstacle.

PID control algorithm improves the line tracking very precisely. It can be achieved by using the PID controller in the circuit. This can improve the speed as well. By using high level line following sensor more accurate line tracking is possible.

Furthermore, Artificial intelligence is taking over the Automation industry, by applying machine learning techniques and swarm intelligence can take the AGV to other level.

References:

[1] **Proteus Design Suite:** https : //proteus.soft112.com/

[2]**ArduinoLibrary:**https://www.theengineeringprojects.com/2015/12/arduino-library-proteus-simulation.html

[3]**BluetoothLibrary:**https://www.theengineeringprojects.com/2016/03/bluetooth-library-for-proteus.html

[4] **VSPE:** http://www.eterlogic.com/Products.VSPE.html

[5]**Arduino IDE:** https://www.arduino.cc/en/Main/Software

[6**]Arduino UNO:** https://store.arduino.cc/arduino-uno-rev3

[7]**Bluetooth Module:** http://www.martyncurrey.com/arduino-with-hc-05-bluetooth-module-at-mode/

[8]**IR Communication:** https://learn.sparkfun.com/tutorials/ir-communication

[9]**Charles Platt.** Encyclopedia of electronic Components Volume 1.Published by O'Reilly Media, in 2012.

[10]**4WD_MiniQ_Robot:**

https://www.dfrobot.com/wiki/index.php/4WD_MiniQ_Complete_Kit_(SKU:ROB0050)

[11]**GitHub Link:**https://github.com/vasu1992/Automatic-Guided-Vehicle

[12]**IR LED:**https://cdn.sparkfun.com//assets/parts/2/9/1/3/09349-1.jpg

[13]**Pushbutton:**https://cdn.sparkfun.com//assets/parts/9/0/00097-03-L.jpg

[14]**LED:**http://thenscaler.com/wp-content/uploads/2015/02/3-leds.gif

[15]**Connecting wires:**

http://www.bitstoc.com/image/cache/data/Products/wiremalefemaleb-500x500.jpg

[16]**Resistors:**https://images-na.ssl-images-amazon.com/images/I/41abov0cVhL._SX342_.jpg

[17]**Battery:**https://images-na.ssl-images-amazon.com/images/I/61mDwpU4taL._SL1000_.jpg