AGV Technology Document

Updated at 22/5/2023

Contents

[**1. General solution** 5](#_Toc137112910)

[**1.1. Requirements** 5](#_Toc137112911)

[**1.2. Solutions** 5](#_Toc137112912)

[Solution 1: Line following using RFID + Beacon 5](#_Toc137112913)

[Solution 2: Line following using computer vision 5](#_Toc137112914)

[**2. Technologies** 5](#_Toc137112915)

[**2.1. Hardware** 5](#_Toc137112916)

[**2.1.1. General configurations** 5](#_Toc137112917)

[Solution 1: Line following using RFID + Beacon 5](#_Toc137112918)

[Solution 2: Line following using computer vision 5](#_Toc137112919)

[**2.1.2. RFID** 5](#_Toc137112920)

[**2.1.3. Bluetooth Low Energy** 5](#_Toc137112921)

[**2.1.4. Laser distance sensor** 5](#_Toc137112922)

[**2.1.5. Camera** 5](#_Toc137112923)

[**2.2. Software** 6](#_Toc137112924)

[**2.2.1. Description** 6](#_Toc137112925)

[**2.2.2. Environment setup** 6](#_Toc137112926)

[**AGV** 6](#_Toc137112927)

[**API server** 8](#_Toc137112928)

[**2.2.3. User interface** 8](#_Toc137112929)

[**AGV Integrated Display User Interface** 8](#_Toc137112930)

[**Dashboard User Interface** 8](#_Toc137112931)

[**2.2.4. Backend** 8](#_Toc137112932)

[**AGV Backend** 8](#_Toc137112933)

[**API Server** 8](#_Toc137112934)

[**3. Operations explanation** 9](#_Toc137112935)

[**3.1. Localizing** 9](#_Toc137112936)

[Solution 1: RFID + Beacon 9](#_Toc137112937)

[Solution 2: Line following using computer vision 9](#_Toc137112938)

[**3.2. Mapping** 9](#_Toc137112939)

[**Map structure** 9](#_Toc137112940)

[**Scanning map** 9](#_Toc137112941)

[Solution 2: Line following using computer vision 9](#_Toc137112942)

[**Synchronizing visual map with real time AGV position** 9](#_Toc137112943)

[Solution 1: Line following using RFID + Beacon 9](#_Toc137112944)

[Solution 2: Line following using computer vision 9](#_Toc137112945)

[**3.3. Guiding** 9](#_Toc137112946)

[**Navigation methods** 9](#_Toc137112947)

[**Navigation states** 10](#_Toc137112948)

[Solution 1: Line following using RFID + Beacon 10](#_Toc137112949)

[Solution 2: Line following using computer vision 10](#_Toc137112950)

[**3.4. Manual mode** 10](#_Toc137112951)

[Controlling direction 10](#_Toc137112952)

[Setting motor speed 10](#_Toc137112953)

[Stopping emergency 10](#_Toc137112954)

[Restarting AGV 10](#_Toc137112955)

[**3.5. Automated mode** 10](#_Toc137112956)

[**Idle mode** 10](#_Toc137112957)

[**Delivery mode ( Automated mode)** 10](#_Toc137112958)

[**3.6. Initial configurations** 11](#_Toc137112959)

[**Map initializing and setting** 11](#_Toc137112960)

[**Beacon scanner setting** 11](#_Toc137112961)

[**Laser sensor setting** 11](#_Toc137112962)

[**Executing board checking** 11](#_Toc137112963)

[**4. Integration** 11](#_Toc137112964)

[**5. Testing** 11](#_Toc137112965)

[**6. Performance reviews** 12](#_Toc137112966)

[**7. Upgrading** 12](#_Toc137112967)

[**7.1. Using computer vision to build 3D map** 12](#_Toc137112968)

[**7.2. Using Lidar to replace individual laser sensors** 12](#_Toc137112969)

[**8. Updating** 12](#_Toc137112970)

[**9. Troubleshooting** 12](#_Toc137112971)

[**9.1. Serial port opening** 12](#_Toc137112972)

[**9.2. Resetting executing board** 12](#_Toc137112973)

[**9.3. Port opening for sending messages to MQTT broker on AGV dashboard** 12](#_Toc137112974)

# **1. General solution**

## **1.1. Requirements**

## **1.2. Solutions**

### Solution 1: Line following using RFID + Beacon

### Solution 2: Line following using computer vision

# **2. Technologies**

## **2.1. Hardware**

### **2.1.1. General configurations**

### Solution 1: Line following using RFID + Beacon

### Solution 2: Line following using computer vision

### **2.1.2. RFID**

Definition

Working principle

### **2.1.3. Bluetooth Low Energy**

Definition

Working principle

### **2.1.4. Laser distance sensor**

Definition

Working principle

### **2.1.5. Camera**

Definition

Working principle

**2.1.6. Battery changer station**

Definition

Working principle

## **2.2. Software**

### **2.2.1. Description**

### **2.2.2. Environment setup**

### **AGV**

**Individual installation**

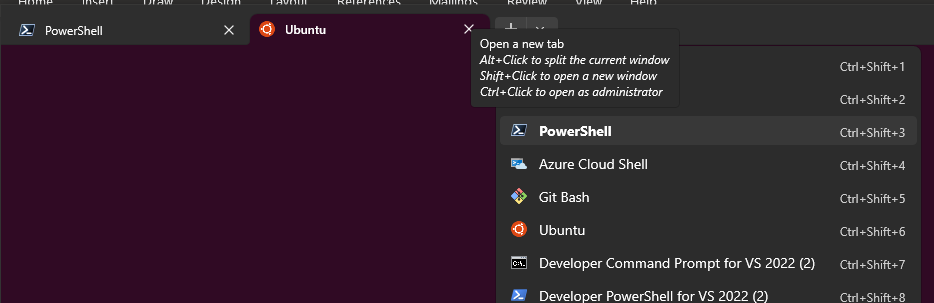
**Raspberry Pi OS installation**

See:https://www.raspberrypi.com/documentation/computers/getting-started.html#using-raspberry-pi-imager

**Mosquitto MQTT installation**

On linux: sudo apt install mosquitto mosquitto-clients

On windows: Using Powershell and run Ubuntu/Linux distro:



Then, run command: sudo apt install mosquitto mosquitto-clients

**Firewall configuration**

On linux: Run these commands:

sudo ufw allow ssh

sudo ufw allow 1883

sudo ufw allow 9001

sudo ufw enable

If ufw already on, use: sudo ufw reload

On windows: From Powershell, run Ubuntu/Linux distro:

If sudo ufw enable not working, see solution: <https://superuser.com/questions/1775776/enabling-ufw-failed-with-ubuntu-from-wsl2>

**Running main control console application**

On Linux & Windows WSL2: Cloning project repo from https://github.com/kirichain/AGV

- Then run these commands:

cd ~/AGV

sudo git reset --hard

sudo git pull

cd ~/AGV/Raspberry/Console/Console/bin/Release/net7.0/publish/linux-arm64

sudo chmod 777 ./Console

./Dashboard

- After chmod 777 is executed, future execution times need only ./Dashboard command to get the console works

### **Dashboard**

**Nginx installation**

On Linux & Windows WSL2: Run these commands

sudo apt install nginx

### **API server**

**Firewall configuration**

**Nginx installation**

**Running installation script**

### **2.2.3. User interface**

### **AGV Integrated Display User Interface**

Login screen

Battery indicator

Delivery request creating menu

Delivery destination choosing menu

Confirmation screen

### **Dashboard User Interface**

Control panel

Main data display panel

System panel

Working principle

Functions/Methods

### **2.2.4. Backend**

### **AGV Backend**

Working principle

Functions/Methods

### **API Server**

API Endpoints

# **3. Operations explanation**

## **3.1. Localizing**

### Solution 1: RFID + Beacon

### Solution 2: Line following using computer vision

## **3.2. Mapping**

### **Map structure**

Dimension

Layer

Cell

### **Scanning map**

### Solution 2: Line following using computer vision

Integrated Camera > Captured Image > Python Script > OpenCV > Binary threshold > Detect Edges > Draw Contours > Confirm if drawn contour is coincident in lane area > Get the biggest contour > Find contour moment > Check moment coordinate > Density of black > Finally confirm this is a lane > Detect direction of lane (horizontal/vertical) > Send data [direction (vertical/horizontal), tilt angle, moment coordinate]

### **Synchronizing visual map with real time AGV position**

### Solution 1: Line following using RFID + Beacon

### Solution 2: Line following using computer vision

## **3.3. Guiding**

### **Navigation methods**

Move (Forward, backward, turn left, turn right)

Stop

Play warning sound

Avoid obstacle

### **Navigation states**

In track

Out of track

Lost

### Solution 1: Line following using RFID + Beacon

### Solution 2: Line following using computer vision

## **3.4. Manual mode**

### Controlling direction

### Setting motor speed

### Stopping emergency

### Restarting AGV

## **3.5. Automated mode**

### **Idle mode**

Waiting for delivery request coming

### **Delivery mode ( Automated mode)**

Creating delivery request

Processing delivery request

Planning path

Identify other AGVs which are in delivering mode

Calculating departure timestamp

Starting transporting

Returning home

## **3.6. Initial configurations**

### **Map initializing and setting**

Creating map

Setting map dimension

Identifying cell value

### **Beacon scanner setting**

Scanning available around beacons

Reading beacon profile data

Assigning beacon name and coordinate to map

### **Laser sensor setting**

Measuring distance

Setting avoiding distance

### **Executing board checking**

# **4. Integration**

**4.1. MES Mobile App**

# **5. Testing**

# **6. Performance reviews**

# **7. Upgrading**

## **7.1. Using computer vision to build 3D map**

## **7.2. Using Lidar to replace individual laser sensors**

# **8. Updating**

# **9. Troubleshooting**

## **9.1. Serial port opening**

## **9.2. Resetting executing board**

## **9.3. Port opening for sending messages to MQTT broker on AGV dashboard**

9.4.