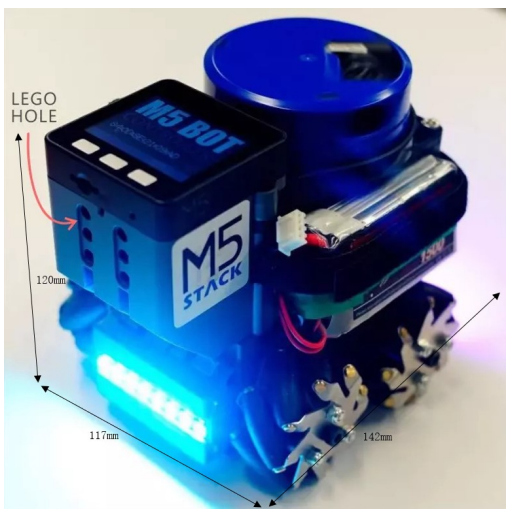
Application LidarBot



Description

LidarBot is a powerful development kit for Automated Guided Vehicles(AGVs). Comes with a 360 Lidar sensor, 4 Mecanum wheels, M5 Core, RGB Bars and a remote conroller with Joystick panel and more. With 4 Mecanum wheels, you can make it move to any direction, forward, backward, to left and to right. The Lipo Batteries enpower the Robot to run long-hours. You can display the map data, that obtained from he lidar sensor, on the screen or upload somewhere else thru Wi-Fi and program it into any format.

we have implemented Real-time communication via ESP-NOW between robot and remote, Mazing-runing , self-tracing and more. If you are interest in AGV development, We especially encourage you to modify the open source code we have offered on github and enhance it yourself.

Product Features

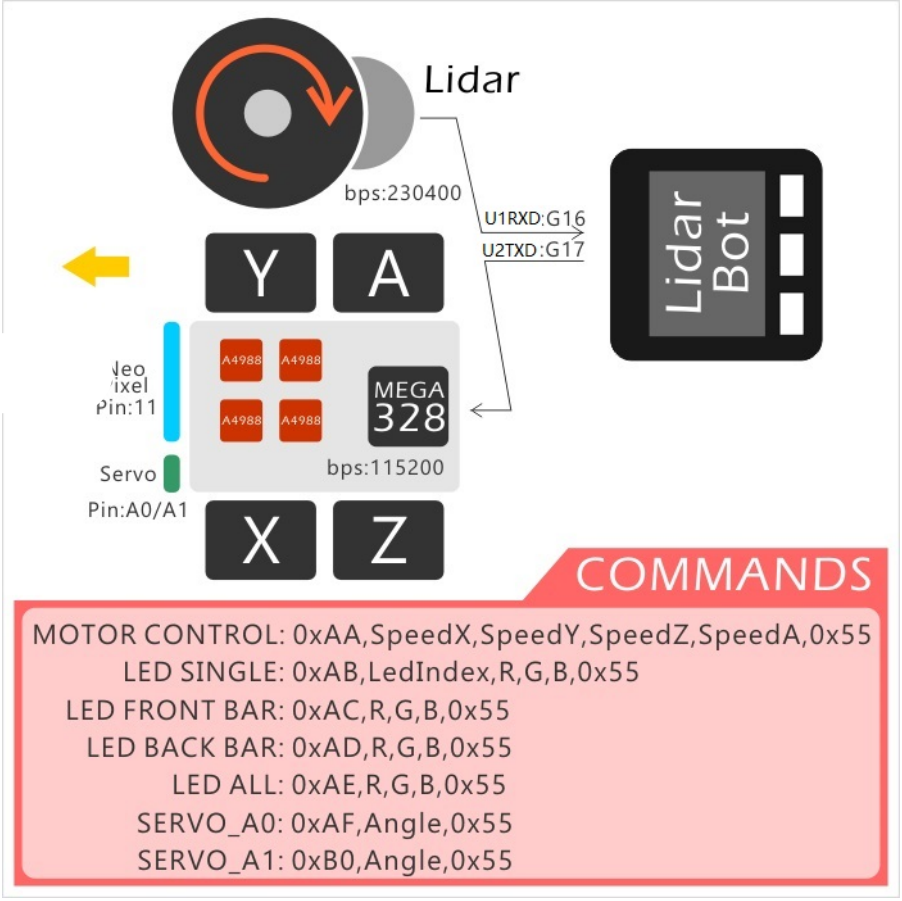
- Lidar: 8m @ 6Hz
- Programming Support
  - Arduino
  - UIFlow (Blockly)
  - Python
- ESP-NOW communication
- Mecanum wheels
- Compatible LEGO

Protocol for CarBottomBoard

Protocol Format: Data Header (command type) + Data Packet + Data Tail

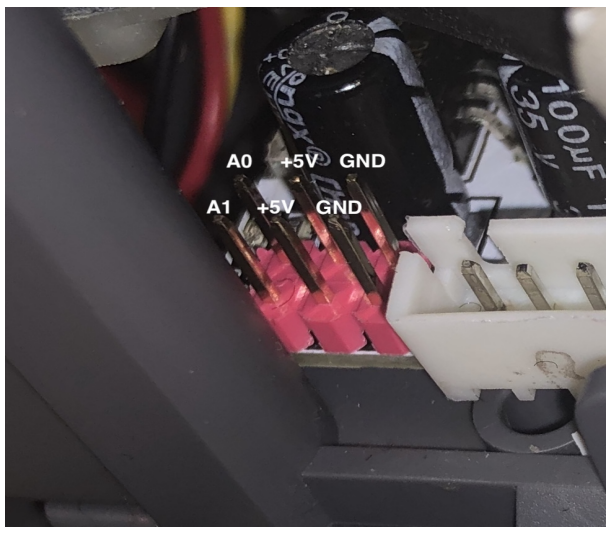
Control Target	Protocol Format	Example	Function
Wheels	0xAA,SpeedX(-7 ~ 7),SpeedY,SpeedZ,SpeedA,0x55	0xAA, 5, 5, 5, 5, 0x55(Go ahead, speed: 5)	ControlWheel(5, 5, 5)
One RGB	0xAB,LedIndex,R(0 ~ 254),G,B,0x55	0xAB, 3, 20, 50, 100, 0x55(3th RGB displays specific color)	setLedColor(3, 20, 50, 100)

Front RGB Bar	0xAC,R(0 ~ 254),G,B,0x55	0xAC, 20, 50, 100, 0x55(Front LED Bar displays specific color)	setFrontLedBar(20, 50, 100)
Back RGB Bar	0xAD,R(0 ~ 254),G,B,0x55	0xAD, 20, 50, 100, 0x55(Back LED Bar displays specific color)	setBackLedBar(20, 50, 100)
All RGB	0xAE,R(0 ~ 254),G,B,0x55	0xAE, 20, 50, 100, 0x55(All LED display specific color)	setLedAll(20, 50, 100)
ServoMotor0	0xAF,Angle(0 ~ 180),0x55	0xAF, 100, 0x55(Servo 0 turns angle 100 degree)	setServo0Angle(100)
ServoMotor1	0xB0,Angle(0 ~ 180),0x55	0xB0, 100, 0x55(Servo 1 turns angle 100 degree)	setServo1Angle(100)



PARAMETER

- The size of LidarBot: 142mm x 117mm x 120mm
- Communication Parameter
  - M5Core <-> Lidar (U1RXD(GPIO16) <-> Lidar sensor) Serial Configuration: "230400bps, 8, n, 1"(8 bits data, no parity, 1 stop bit)
  - M5Core <-> Bottom Board (U2TXD(GPIO17) <-> Bottom Board) Serial Configuration: "115200bps, 8, n, 1"(8 bits data, no parity, 1 stop bit)
- PinMap
  - ServoMotor0 <-> A0(MEGA328)
  - ServoMotor1 <-> A1(MEGA328)
  - RGB LED <-> 11(MEGA328)



Include

- 1x LidarBot
- 1x Remote Control Handle
- 2x Battery(1300mAh @ 11.1V)
- 1x Power Charger
- 1x Type-C USB Cable

Applications

- Indoor Navigation
- Autonomous walking maze
- Route plan
- Autopilot

Weight and Size

- Package size:208mm x 208mm x 167mm
- Package weight:2140g

EasyLoader

1.EasyLoader is a simple and fast program burner. Every product page in EasyLoader provides a product-related case program. It can be burned to the master through simple steps, and a series of function verification can be performed.(**Currently EasyLoader is only available for Windows OS**)

2.After downloading the software, double-click to run the application, connect the M5 device to he computer via the data cable, select the port parameters, and click **"Burn"** to start burning.

3.The CP210X (USB driver) needs to be installed before the EasyLoader is burned. Click here to view the driver installation tutorial

Example

To get complete code, please click here。

Tree for Example Directory

├─LidarBot_CarMain_V1.1 - Main program of LidarBot

├─LidarBot_RemoteController_V1.0 - Program of RemoteController V1.0

└─LidarBot_RemoteController_V1.2 - Program of RemoteController V1.2(higher precision)

Program analysis:

1. Main program of LidarBot:

```arduino
/* Main program */
void loop()
{
  espnow.BotConnectUpdate();// ESPNOW reconnect
  lidarcar.MapDisplay();// display map
  esp_now_send(espnow.peer_addr, lidarcar.mapdata, 180);// ESPNOW sends map data
}
```

- **Single function resolution:**
  - Usage of reading radar data

```
lidarcar.Init();
GetData();//save radar data to array distance[]
```

- Usage of line following

```arduino
#include "rprtrack.h"
Rprtrack rprtrack;

SensorStatus();// save line following data to array sensorValue[]
CalTrackDev();// handle array sensorValue[], get car offset and save it
```

- Usage of ESP_NOW

Please refer to https://github.com/m5stack/M5-espnow

Program of RemoteController

```arduino
/* Main program */
void loop()
{
  espnow.RemoteConnectUpdate();// ESPNOW reconnect
  keyboard.GetValue();// read data of joystick
  // ESPNOW sends joystick data to car
  esp_now_send(espnow.peer_addr, keyboard.keyData, 3);
  MapDisplay();// display map
  accessport.AnalzyCommand();// send map data to PC software
}
```

- **Single function resolution:**

  - Usage of JOYSTICK

```arduino
#include "keyboard.h"
KeyBoard keyboard;

keyboard.Init();
// get joystick data and save to adX, adY
GetValue();
```

- Usage of communication with PC software

```arduino
#include "accessport.h"
AccessPort accessport;

accessport.AnalzyCommand();// send map data to PC software
```