

家庭监护机器人平台

产品简介

HCR 移动机器人套件是一套两轮驱动的两层机器人平台，当然你也可以根据需求调整所使用的部件。套装包含有两个电机，两只轮胎，一只万向轮，以及相关金属板及配套硬件。支撑结构包括伺服机和传感器，此结构上含有专用于 mini-itx 主板的接口。

这是 DFRobot 的第二代 HCR 平台，这次我们更换了新的电机和轮子，并更换了传感器的组成。现在 HCR 有 6 个超声波传感器感应 6 个方向，以及 DFRobot 定制的高质直流电机。此款电机采用光编码构造，具有静音，扭矩大的特点。可选用编码器每转 663 个脉冲，这使得轮子转动精度达到 0.54 度。其分辨率能达到常用 PID 速度控制器的要求。新 HCR 平台能够赋予用户全新的体验，用户能够自由发挥创想力，例如在平台上结合 Kinect 之类的部件。

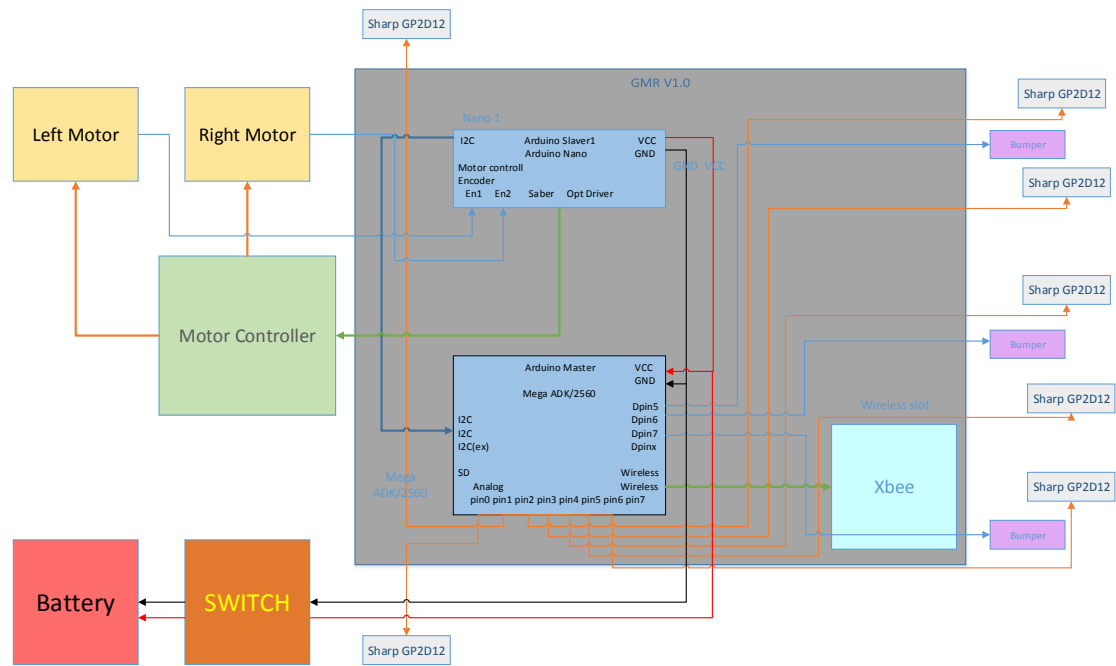
技术规格

- 橙色的铝涂层
- 相关板材和硬件
- 大小：30X30X35cm
- 12V 电机 减速比：51:1 负载转速 8000 转每分钟
- 3 层结构
- 轮子大小：直径 13cm

配件列表

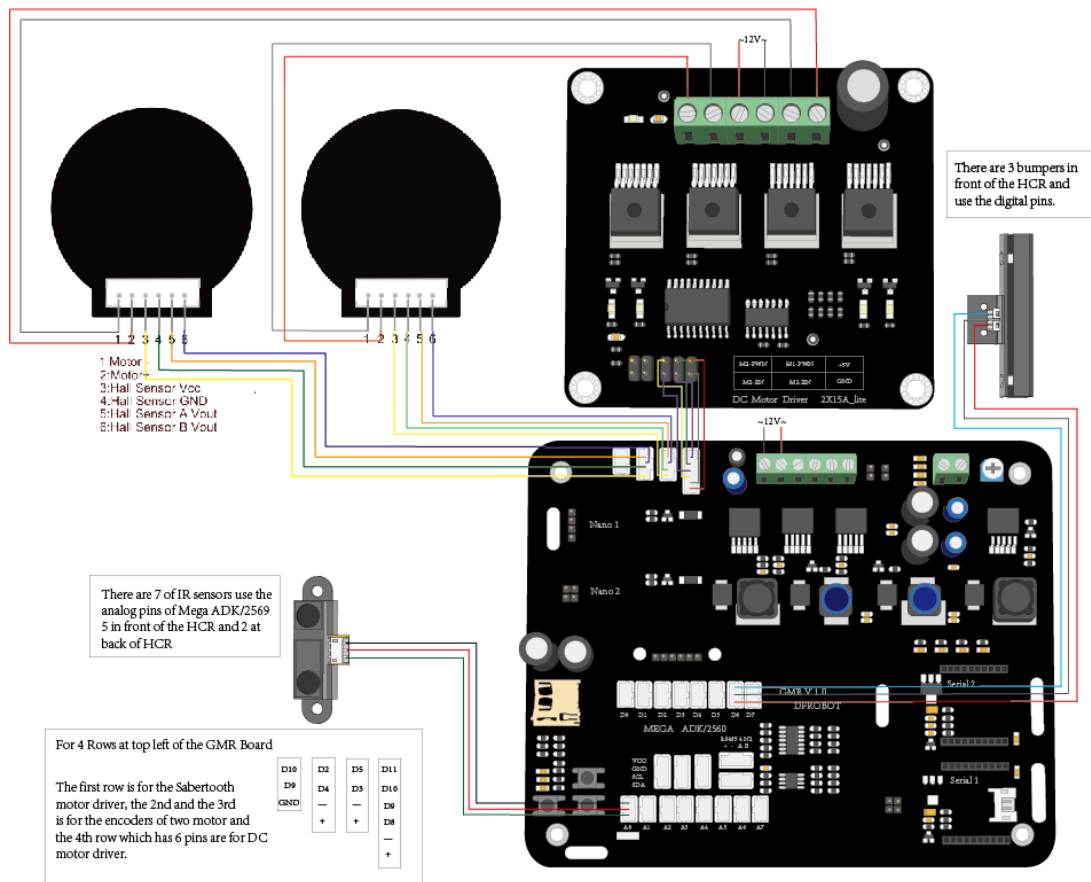
- 主控板：Arduino Mega ADK/2560
- 电机控制板：Arduino Nano
- 超声波驱动板：Arduino Nano
- 电机驱动：双路 15A 大功率电机驱动 - Lite/ Sabertooth 双路 12A 直流电机驱动
- 12V 电机 减速比：51:1 负载转速 8000 转每分钟
- 2 个（2 阶段的）编码器，使用 13 PPR
- 2 个轮子 和一个万向轮
- 3 个碰撞传感器
- 5 个红外测距传感器（Sharp GP2D12）
- 6 个超声波传感器

硬件构建



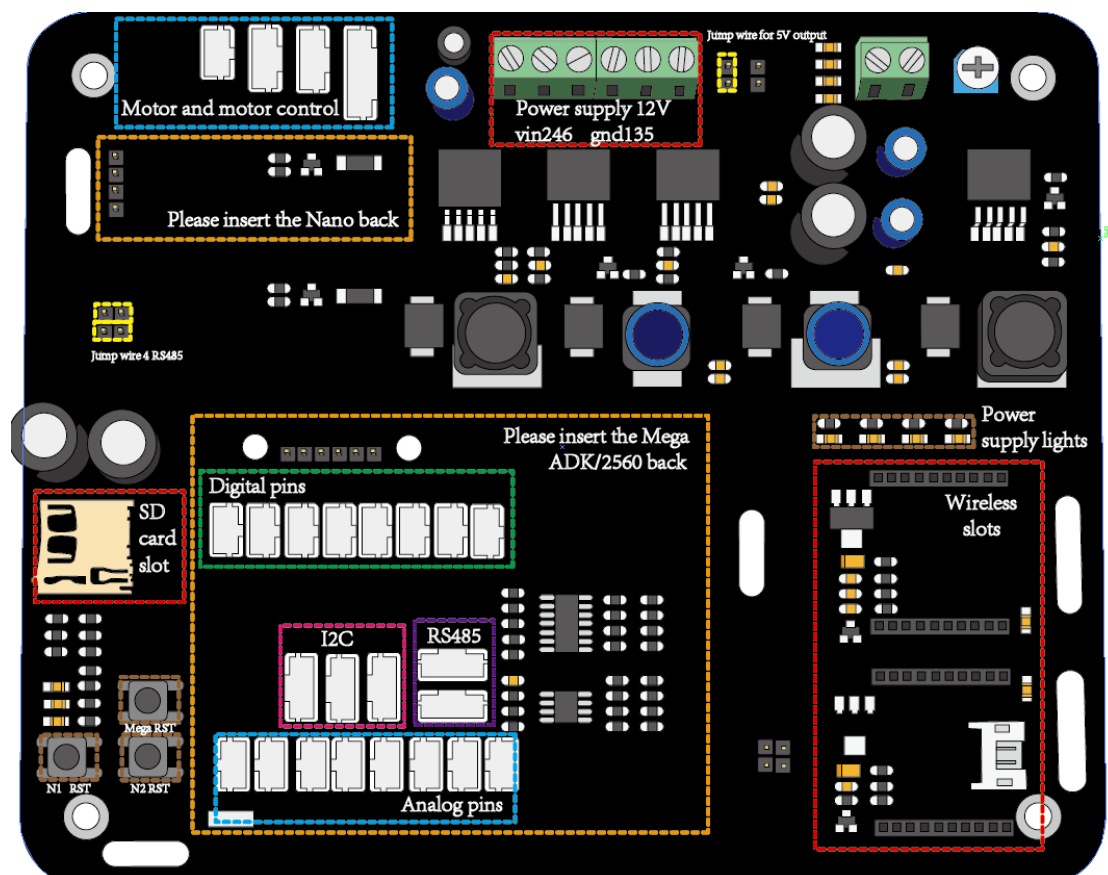
图片 1 硬件构建图

电路连接



图片 2 电路连接图

GMR Board



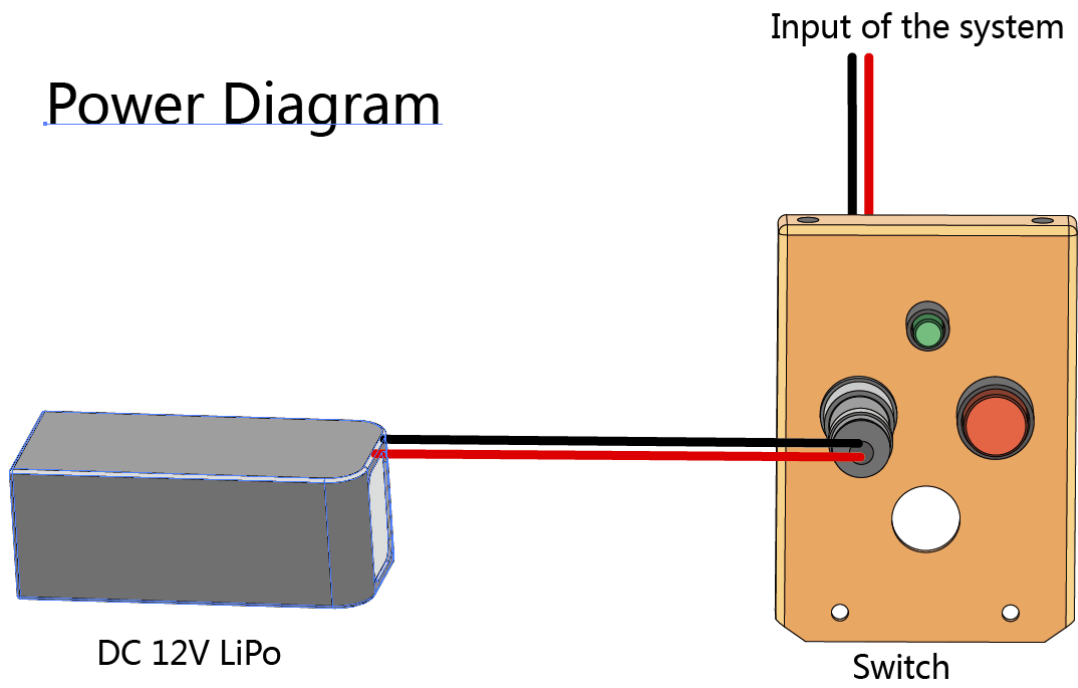
图片 3 the GMR board

注意：黄色部分需要连接跳线。

在这块集成版上需要插入 3 个微处理器(单片机), 2 块 Arduino Nano 和一块 Arduino Mega ADK。其中一块 Nano 作为电机控制以及数据采集, 另一块用来驱动超声波传感器。Arduino Mega ADK 作为主控板, 采集传输数据以及其他扩展模块, 另外还有 2 个无线通讯槽可使用 Xbee, 蓝牙以及其他通讯模块。

开关系统

Power Diagram



图片 4 开关系统连接

测试

- 在 GMR 板上插入一块 nano 以及 mega ADK，参照图片 3 连接开关到 GMR 板电源输入端。
- 断开开关连接电源。
- 如图片 2 所示连接电机驱动以及电机，我们先测试电机控制。
- 使用软件构建中的电机控制代码烧入微处理器。
- 测试电机转速及方向，使其能按照自己的意愿。
- 按图示连接碰撞传感器和红外测距传感器，样例代码见软件构建部分。你可能需要使用另一块微处理器测试代码和传感器，比如 Arduino Uno,然后移植代码到 mega 板上。
- 电机控制的代码烧在 Nano 板上，碰撞传感器以及红外测距驱动部分代码烧在 Mega 板。Nano 板与 mega 板使用 I2C 进行通讯。
- 测试完全部代码之后，按照使用向导中的示范可以驱动 HCR。

使用向导

1. 参照组装教程组装 HCR
2. 移除 2, 3 两层, 假如事先没有装入电池。
3. 将电池固定在第一层 (12V) ,按图片 4 所示连接开关系统, 再安装上第二层。
4. 在 GMR 板上插入一块 nano 以及 mega ADK, 参照图片 3 连接开关到 GMR 板电源输入端。
5. 按图片 2 所示连接电路, 包括 2 个电机, 电机驱动, 3 个碰撞传感器和 7 个红外测距仪, 然后可以把电机驱动固定在 GMR 板上。
6. 根据你的需求, 在 Nano 和 Mega 板上烧入程序。样例代码在软件构架中给出。Nano 板与 mega 板使用 I2C 进行通讯。在 Nano 上的代码可以驱动轮子。
7. 安装第三层。

软件构架

在这个部分, HCR 的样例代码会给出, 使用 **Arduino** 编写, 包括电机

驱动，碰撞传感器和红外测距仪。

Arduino IDE 1.0.x

代码适用于 Arduino 1.0 以上版本。

所有硬件都适用于 **Adruino**

Arduino Nano 上的代码用于控制电机：

如果使用 **Sabertooth** 电机控制，可以使用 **<serve.h>** 库来完美驱动电机。

代码：

```
#include <Servo.h>

Servo Lmotor;
Servo Rmotor;

float Lspeed = 0; //from 0-180, 0 means max speed forwards, 180 means max speed backwards 90 means stop
float Rspeed = 0; //from 0-180,

void setup()
{
  Lmotor.attach( 9, 1000, 2000);
  Rmotor.attach( 10, 1000, 2000);
}

void loop()
{
  Lmotor.write(Lspeed);
  Rmotor.write(Rspeed);
  delay(500);
} // use this code to control the HCR move
```

如果使用 **DC Motor Driver 2x15A – Lite**

代码：

```
#define LF 0
#define RT 1

int E1 = 9; //M1 Speed Control
int E2 = 10; //M2 Speed Control
int M1 = 8; //M1 Direction Control
int M2 = 11; //M1 Direction Control

int a;
int b;
```



```

int counter=0;

void setup()
{
  int i;
  for(i=4;i<=7;i++)
  pinMode(i, OUTPUT);
  Serial.begin(57600);    //Set Baud Rate
  Serial.println("Run keyboard control");
  digitalWrite(E1,LOW);
  digitalWrite(E2,LOW);
}

void loop()
{
  Motor(2000,LF);
  Motor(1000,RT);
}

void Motor(int value,byte whichwheel)
{
  value = constrain(value,1000,2000);
  if(whichwheel == LF) {
    if(value>1500) {
      a=(value-1500)/1.961;
      analogWrite (E1,a);
      digitalWrite(M1,HIGH);
    }
    else {
      a=(1500-value)/2;
      analogWrite (E1,a);
      digitalWrite(M1,LOW);
    }
  }
  else if(whichwheel == RT){
    if(value>1500) {
      b=(value-1500)/1.961;
      analogWrite (E2,b);
      digitalWrite(M2,HIGH);
    }
    else {
      b=(1500-value)/2;
      analogWrite (E2,b);
      digitalWrite(M2,LOW);
    }
  }
}

```

```
    }  
  }  
}
```

第一件事使得轮子可以转起来 碰撞传感器代码:

BumperFunction.h

```
#include "Arduino.h"
```

```
int BumperR_pin;
```

```
int BumperL_pin;
```

```
int BumperC_pin;
```

```
byte BumperValue;
```

```
boolean blocked = false;
```

```
void OpenBumper(int,int,int);
```

```
void bumperRead();
```

```
/****** Details *****/
```

```
void OpenBumper(int LIO,int CIO,int RIO)
```

```
{
```

```
  BumperL_pin = LIO;
```

```
  BumperC_pin = CIO;
```

```
  BumperR_pin = RIO;
```

```
  pinMode(BumperL_pin,INPUT);
```

```
  pinMode(BumperC_pin,INPUT);
```

```
  pinMode(BumperR_pin,INPUT);
```

```
}
```

```
/******Bumper Sensor
```

```
Status******/
```

```
void bumperRead()
```

```
{
```

```
  BumperValue = 0x07;
```

```
  BumperValue=digitalRead(BumperL_pin)<<2;
```

```
  BumperValue|=digitalRead(BumperC_pin)<<1;
```

```
  BumperValue|=digitalRead(BumperR_pin);
```

```
  // Serial.println(BumperValue,BIN);
```

```
}
```

红外测距仪代码:

```
/****** IR sensor *****/
```

```
void IRreader()//detect distance on both sides
```

```
{
```

```
  static float IRdata[IrNumber] = {
```

```
    80,80,80,80,80
```

```
  };
```

```

for(int h=0;h<IrNumber-2;h++)
{
    float volts = analogRead(h + 1);
    _iR[h] = (6787 / (volts - 3)) - 4;
    if(_iR[h]<10) _iR[h] = 80;
    _iR[h] = min(_iR[h],80);
    _iR[h] = max(_iR[h],12);
}
for(int h=5;h<7;h++)
{
    float volts = analogRead(h-4);
    _iR[h] = (6787 / (volts - 3)) - 4;
    if(_iR[h]<10) _iR[h] = 80;
    _iR[h] = min(_iR[h],80);
    _iR[h] = max(_iR[h],12);
}
for(int h = 0 ; h < 5 ; h++)
{
    _iR[h] = smooth(_iR[h],0.9,IRdata[h]);
    IRdata[h] = _iR[h];
}
}
float smooth(float newdata, float filterVal, float smoothedVal)
{
    if (filterVal > 1)
        filterVal = .99;
    else if (filterVal <= 0)
        filterVal = 0;
    smoothedVal = (newdata * (1 - filterVal)) + (smoothedVal * filterVal);
    return smoothedVal;
}

```

附录

硬件技术规格

主控制器： Arduino Mega ADK/2560

技术规格

- 微处理器: ATmega2560
- 工作电压: 5V
- 输入电压 (推荐): 5-23V
- 数字口输入/输出: 54 (其中 14 个 PWM 输出)
- 模拟输入口: 16
- 输入输出直流电流: 40 mA
- 3.3V 端口电流: 50 mA
- 闪存 256 KB 其中 8 KB 被使用于存储代码
- 静态随机存储: 8 KB
- 可更改的只读存储器: 4 KB
- 时钟频率 :16 MHz

用于传输数据以及驱动碰撞传感器和红外测距仪

http://www.dfrobot.com/index.php?route=product/product&filter_name=MegaADK&product_id=520#.Ua7ly3SS2Uk

从控制器: **Arduino Nano**

技术规格

- 微处理器: Atmel Atmega328-20AU
- 工作电压: 5V
- 输入电压(推荐): 7-12V
- 输入电压 (限制): 6-20V
- 数字输入输出口: 14 (其中 6 个 PWM 输出)
- 模拟输入口: 8
- 端口直流电流: 40 mA
- F 闪存: 32 KB 其中 2KB 被使用于存储代码
- 静态随机存储: 2 KB
- 可更改的只读存储器: 1 KB
- 时钟频率: 16 MHz

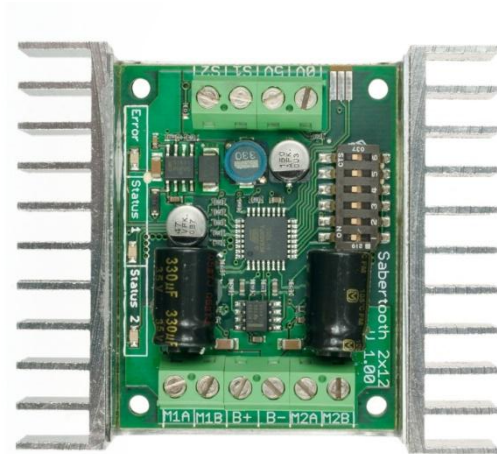
用于驱动带编码器电机以及驱动超声波传感器。

http://www.dfrobot.com/index.php?route=product/product&filter_name=Nano&product_id=67#.Ua7nVHSS2Uk

电机驱动:

DC Motor Driver 2x15A – Lite 或者 Sabertooth dual 12A motor driver

Sabertooth 双路 12A 直流电机驱动

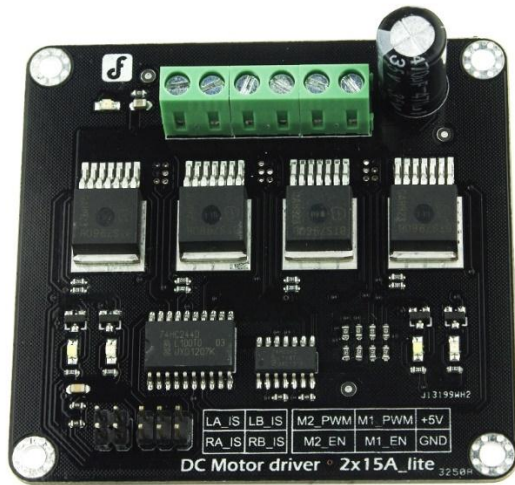


技术规格

- 至多 18V 输入: 每个通道 12A 连续, 15A 峰值电流
- 24V 输入: 每个通道 12A 连续, 25A 峰值电流
- 同步可更新驱动
- 超声波频率选择
- 热能和过量电流保护
- 锂电池保护模式
- 输入模式: 模拟, R/C, 简化版串行, 封装化串行
- 尺寸: 2.3x 3x.7" (59x75x17mm)

http://www.dfrobot.com/index.php?route=product/product&path=51_105&product_id=304#.UZxogXSS2Uk

双路 15A 大功率电机驱动 - Lite



技术规格

- 驱动部分输入电压：4.8-35V
- 最大输出电流：单路 15A@13.8V
- 单路峰值电流：20A@13.8V
- PWM 最高输出频率：25KHZ
- 驱动输出接口：5.08mm 间距接线柱
- 控制接口：4 路数字信号（其中 2 路 PWM 信号）
- 驱动形式：双路大功率 H 桥驱动
- 其它特性：电流隔离功能，保护单片机
 - 双路电流检测诊断功能
 - 短路，过热，过压保护
- 尺寸：73x68x14mm

http://www.dfrobot.com/index.php?route=product/product&path=51_105&product_id=796#.Ua6vN3SS2Uk

电机:

12V 直流减速电机 146rpm 带编码器

技术规格

- 供电要求: DC 12V
- 空载电流: 130mA
- 空载转速: 8000rpm
- 输出转速: 146rpm
- 减速比: 51:1 (行星减速齿轮)
- 额定转矩: 1.0N·m
- 接口模式: JST PHR-6, P=2.0-6P
- 编码器脉冲数: 13PPR
- 外形尺寸: 123×36×36mm
- 安装孔规格: 4-M3
- 安装孔节径: $\Phi 28$

http://www.dfrobot.com/index.php?route=product/product&filter_name=DC_motor&product_id=777#.UZ2MsHSS2Uk

Xbee/Wifi/Bluetooth:

与主控器 Arduino Mega ADK/2560 连接, 提供控制遥控 HCR 的方法

红外测距仪:

SHARP GP2D120X 红外距离传感器



技术规格

- 信号类型: 模拟输出
- 探测距离: 4-30cm
- 工作电压: 4.5~5.5V
- 标准电流: 33mA
- 接口类型: PH2.0-3P
- 最大尺寸: 40x20x13.5 mm

[http://www.dfrobot.com/index.php?route=product/product&filter_name=Sharp GP2D12&product_id=328#.Ua7oRXSS2Uk](http://www.dfrobot.com/index.php?route=product/product&filter_name=Sharp+GP2D12&product_id=328#.Ua7oRXSS2Uk)

碰撞传感器:

使用数字口.

Last two kinds of sensors help the HCR to avoid the obstacle.