

به نام نور



پروژه الکترونیک عملی

استاد راهنما: دکتر پاشاروش

دانشجو:

عرفان رادفر 99109603

ریحانه نیکوییان 99106747

محمدطه نوینی : 99106725

سال تحصیلی: 1402

فهرست

4	مشخصات فنی ماژول درایور موتور L298
5	بررسی پایه های ماژول درایور موتور L298
6	ماژول فرستنده گیرنده مادون قرمز IR FC51
6	ویژگی های سنسور FC51
7	بررسی پایه های سنسور FC51
10	مشخصات و ویژگی های ESP-01S
11	بررسی پایه های ماژول وای فای ESP-01
14	نحوه اتصال قطعات
16	کدها
19	شرح کد:
20	تنظیم ضرایب کنترلر:
22	عکس setup

شرح پروژه

پروژه پایانی

عنوان: کنترل از راه دور سرعت موتور DC

تعریف پروژه: در این پروژه با استفاده از برد آردوینو، یک سیستم کنترل از راه دور برای تنظیم سرعت موتور DC طراحی می‌شود. این سیستم به کاربر امکان می‌دهد تا در هر زمانی سرعت موتور DC را بر روی صفحه گوشی خود مشاهده کرده و در صورت نیاز، مقدار آن را تغییر دهد.

شرح پروژه: در این پروژه، با استفاده از ماژول L298 یک موتور DC راه‌اندازی خواهد شد و با استفاده از برد آردوینو در هر لحظه می‌توان سرعت و جهت چرخش موتور را تغییر داد. برای اندازه‌گیری سرعت چرخش موتور، از سنسور IR استفاده شود و سیگنال خروجی سنسور در اختیار برد آردوینو قرار می‌گیرد. همچنین برای برقراری ارتباط کاربر با سیستم و امکان تغییر سرعت یا جهت چرخش موتور و نمایش سرعت موتور برای کاربر در هر لحظه، از ماژول Wi-Fi استفاده خواهد شد. کاربر می‌تواند با اتصال گوشی خود به Wi-Fi سیستم، اطلاعات لازم را دریافت کند و دستورات خود را برای برد آردوینو ارسال کند تا بدین ترتیب موتور را کنترل کند. همچنین برای کنترل دقیق سرعت موتور، پیاده‌سازی کنترلر PI در آردوینو توصیه می‌شود. در هر لحظه اطلاعات دریافتی از سنسور IR به عنوان فیدبک به کنترلر ارسال می‌شود و خروجی کنترلی مناسب برای تنظیم سرعت موتور متناسب با ضرایب کنترلی تعیین می‌شود. برای بدست آوردن ضرایب کنترلی نیز می‌توان از روش‌های مختلفی همچون سعی و خطا، زیگلر-نیکولز و ... استفاده کرد.

توضیح قطعات و وسایل استفاده شده در پروژه

(a) درایور L298

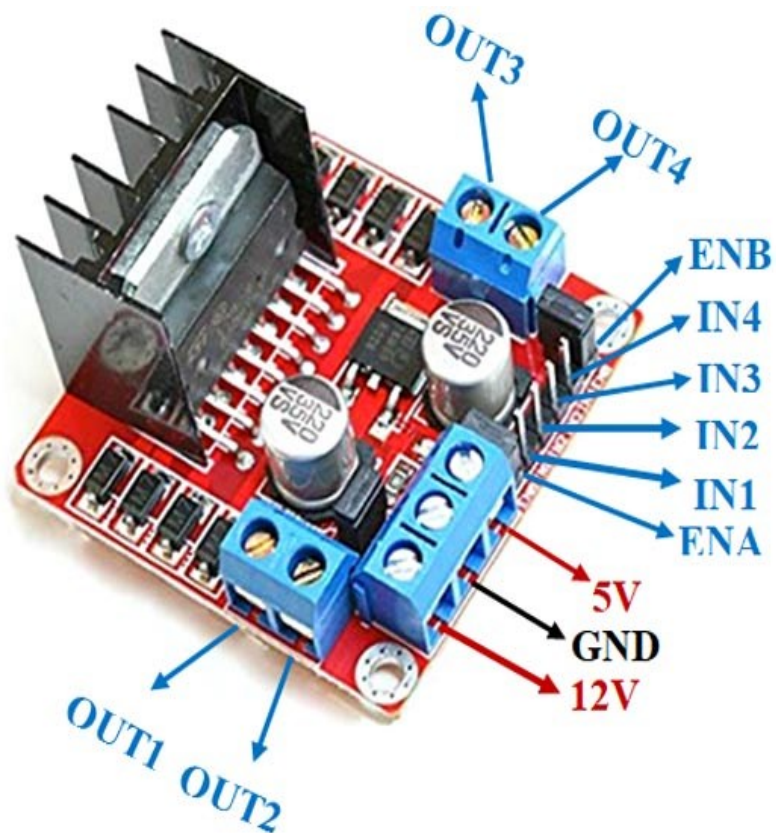
ماژول درایور موتور L298 یک ماژول درایور موتور است که برای کنترل حرکت موتورها در پروژه‌های الکترونیکی با استفاده از میکروکنترلرهایی مانند آردوینو Arduino و ESP ها استفاده می‌شود. ماژول درایور موتور L298 دارای دو کانال است که هر کدام از آنها به صورت مستقل قابل کنترل است و می‌تواند موتورهای DC و Stepper Motor با جریان حداکثر 2 آمپر را کنترل کند. ماژول L298 دارای ولتاژ تغذیه 5 تا 35 ولت است که به آن از طریق باتری یا منبع تغذیه خارجی متصل می‌شود. ماژول درایور موتور L298 دارای چهار ورودی است که دو ورودی آن برای هر کانال موتور به عنوان ورودی کنترل سرعت استفاده می‌شود. همچنین این ماژول دارای دو خروجی هستند که به دو موتور متصل می‌شوند. در این ماژول، جریان خروجی برای هر موتور به صورت مجزا قابل تنظیم است. با استفاده از ماژول درایور موتور L298 می‌توان توان و جهت حرکت موتورها را کنترل کرد و همچنین می‌توان آنها را به صورت دستی یا به صورت خودکار با استفاده از سنسورهای مختلف کنترل کرد. همچنین این ماژول قابلیت ارتباط با آردوینو و سایر میکروکنترلرها را دارد و برای پروژه‌هایی که نیاز به کنترل موتور دارند، یکی از بهترین گزینه‌هایی است که در دسترس است.

مشخصات فنی ماژول درایور موتور L298

- ولتاژ تغذیه: 5 تا 35 ولت
- جریان حداکثر خروجی: 2 آمپر برای هر کانال
- تعداد کانال‌ها: 2 کانال

- ورودی کنترل سرعت: 2 ورودی برای هر کانال
- قابلیت کنترل موتورهای DC و Stepper Motor
- دارای دو خروجی قابل تنظیم جریان خروجی برای هر موتور
- ابعاد کوچک و مناسب برای استفاده

بررسی پایه های مازول درایور موتور L298
 مازول درایور موتور L298 دارای 15 پایه است که به صورت زیر شماره گذاری شده اند:



(b)

ماژول فرستنده گیرنده مادون قرمز IR FC51

سنسور FC51 یک سنسور مادون قرمز (Infrared Sensor) قابل استفاده در پروژه‌های الکترونیکی می‌باشد. این سنسور قابلیت تشخیص اشیاء را با فاصله کمتر از ۷ سانتیمتر را داراست. سنسور FC51 دارای دو پین ولتاژ و یک پین خروجی می‌باشد. پین VCC به منبع تغذیه ۳ تا ۵ ولتی وصل می‌شود و پین GND به زمین متصل می‌شود. پین خروجی OUT به میکروکنترلر یا برد آردوینو وصل می‌شود تا بتوان اطلاعاتی که توسط سنسور جمع‌آوری شده است را پردازش کرد. عملکرد این سنسور بر اساس تغییرات فرکانس مادون قرمز است که به سنسور برمی‌گردد. این تغییرات فرکانس به دلیل انعکاس نور از سطح اشیاء در معرض مادون قرمز ایجاد می‌شود و سنسور قادر است این تغییرات را دریافت کند. سنسور FC51 به دلیل سادگی استفاده و ارزان بودن، در بسیاری از پروژه‌های الکترونیکی کاربرد دارد.

ویژگی های سنسور FC51

- نوع سنسور: مادون قرمز (Infrared Sensor)
- ولتاژ کاری: 3 تا 5 ولت
- جریان کاری: حداکثر 20 میلی آمپر
- فاصله تشخیص اشیاء: حداکثر 7 سانتیمتر
- خروجی: سیگنال دیجیتال با مقادیر HIGH و LOW
- زاویه تشخیص: ۳۵ درجه
- زمان پاسخگویی: کمتر از ۲۵ میلی ثانیه
- دمای کاری: ۲۰ تا ۵۰ درجه سلسیوس
- سادگی و قابلیت استفاده در پروژه‌های الکترونیکی
- ارزان بودن و قیمت مناسب
- عملکرد پایدار و دقیق در شرایط مختلف
- قابلیت استفاده در پروژه‌هایی که نیاز به تشخیص اشیاء با فاصله کم دارند
- قابلیت اتصال به بردهای مختلف میکروکنترلر و بردهای آردوینو

بررسی پایه های سنسور FC51

- پایه VCC: ولتاژ تغذیه 5 ولت برای سنسور
- پایه GND: مربوط به زمین سنسور
- پایه S: خروجی دیجیتال سنسور که متصل به پین دیجیتال میکروکنترلر می شود
- پایه +: خروجی آنالوگ مثبت سنسور که به پین آنالوگ میکروکنترلر متصل می شود
- پایه -: خروجی آنالوگ منفی سنسور که به پین آنالوگ میکروکنترلر متصل می شود

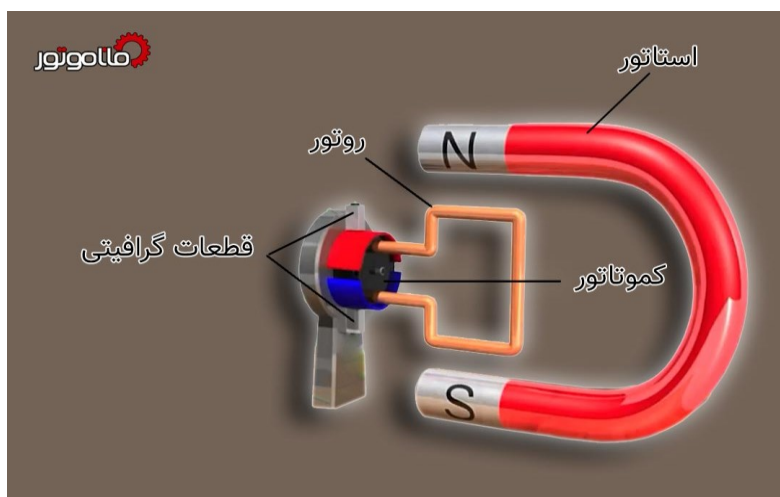


(c)

موتور dc 5 ولت

موتور dc وسیله ای است که انرژی الکتریکی را به کمک یک آهنربا به انرژی مکانیکی تبدیل میکند. به علت اینکه از جریان برق مستقیم یا DC 24 ولت استفاده میکند به آن DC میگویند. یکی از معایبی که موتورهای DC دارند این است که به دلیل اینکه ممکن است در طول زمان کموتاتور آن ها فرسود شود نیاز به تعمیر و نگهداری بیشتری دارند.

برای فهمیدن اینکه موتور DC چگونه کار میکند، یک مدار الکتریکی ساده را در نظر بگیرید که در آن انرژی الکتریکی توسط یک باتری تامین میشود. وقتی جریان الکتریکی در مدار جاری میشود، یک میدان مغناطیسی در اطراف خود ایجاد میکند و باعث میشود آرمیچر به صورت پیوسته به چرخش درآید.





(d)

ماژول esp8266 wifi

ماژول ESP-01S یک ماژول وای فای کوچک با چیپست ESP8266 است که به طور خاص برای کاربردهای اینترنت اشیا (IoT) طراحی شده است. این ماژول دارای آنتن داخلی است و با ولتاژ 3.3 ولت کار می‌کند. ESP-01S دارای پین‌های ورودی/خروجی عمومی (GPIO) برای اتصال به سنسورها و دستگاه‌های دیگر است. همچنین دارای پین‌هایی برای ارتباط سریالی با کامپیوتر و یا میکروکنترلرهای دیگر است. ماژول ESP-01S از طریق دستورات AT قابل برنامه‌ریزی است و می‌تواند با استفاده از میکروکنترلرهای مختلفی مانند Arduino و Raspberry Pi کار کند. همچنین می‌توان از زبان برنامه‌نویسی میکروپایتون برای برنامه‌نویسی ESP-01S استفاده کرد.

مشخصات و ویژگی‌های ESP-01S

ماژول وای فای ESP-01S به دلیل اندازه کوچک و هزینه مناسب خود، به عنوان یکی از بهترین گزینه‌های برای پروژه‌های اینترنت اشیا شناخته می‌شود. مزیت‌های استفاده از این ماژول عبارتند از:

- دارای ولتاژ کاری 2.56 الی 3.6 ولت
- ESP-01S دارای حافظه فلش 8 مگابایت است.
- دارای آنتن PCB قرار گرفته روی برد ماژول
- قابلیت پروگرام و اجرای برنامه‌های آردوینو
- دارای دو پایه GPIO جهت استفاده در برنامه‌ها
- قابلیت تعریف و قرارگیری در مد های کم مصرف
- دارای فرامین AT کامند جهت اجرای دستورات کاربر
- دارای پروتکل ارتباطی سریال UART جهت ارتباط با میکروکنترلرها و رایانه‌ها
- قابلیت قرارگیری در مد STATION جهت اتصال به مودم‌ها و هات اسپات تلفن همراه
- دارای وب سرور داخلی جهت قرارگیری در مد سرور و اجرای درخواست های کلاینت

○ قابلیت قرارگیری در مد نقطه دسترسی (Access Point) جهت اتصال سایر ماژول ها و یا

تلفن های همراه

بررسی پایه های ماژول وای فای ESP-01

○ VCC: ولتاژ تغذیه (3.3 ولت)

○ GND: زمین

○ GPIO0: پین ورودی/خروجی عمومی

○ GPIO2: پین ورودی/خروجی عمومی

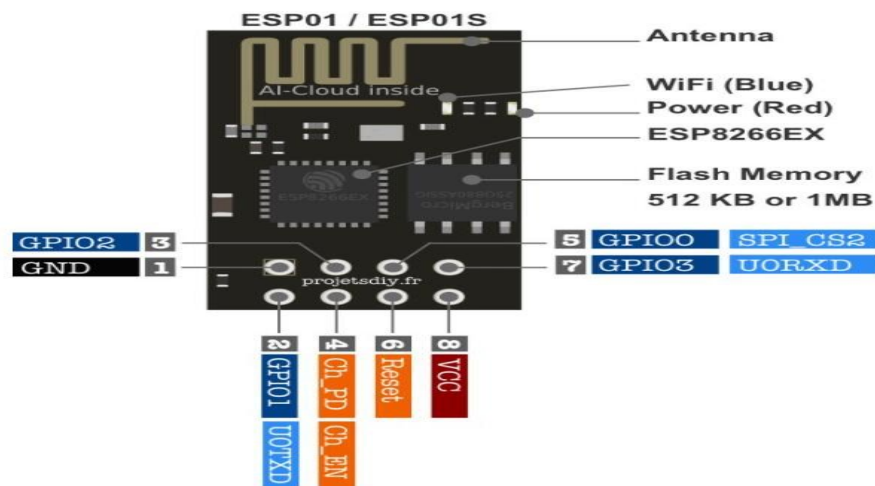
○ CH_PD: پین خاموشی/روشنی میکروکنترلر. برای فعال کردن میکروکنترلر باید به حالت

روشن شود

○ RST: پین ریست

○ UART TXD: پین فرستادن سریالی

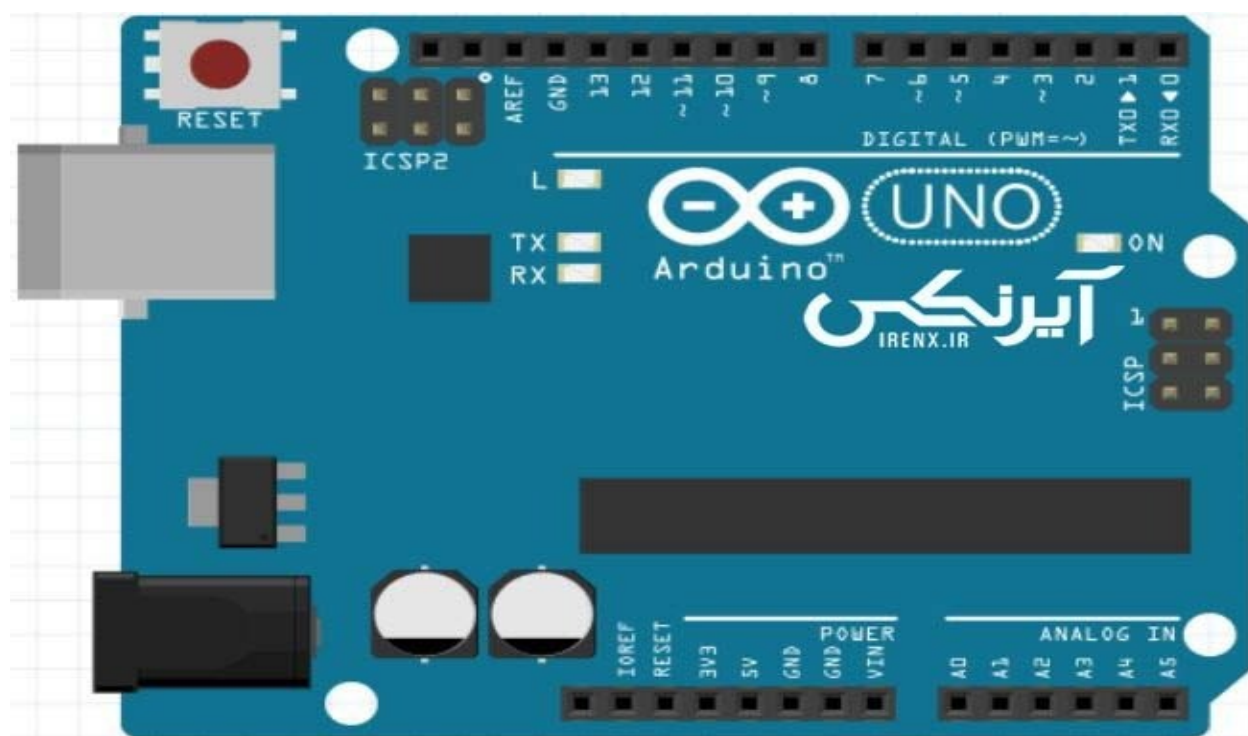
○ UART RXD: پین دریافت سریالی



اردوینو uno

Arduino Uno یک برد متن باز مبتنی بر میکروکنترلر Microchip ATmega328P است و توسط Arduino.cc ساخته شده است. این برد مجهز به مجموعه ای از پین های ورودی / خروجی دیجیتال و آنالوگ است که میتواند به راحتی با شیلد های آردوینو Uno ارتباط برقرار کند. برد Arduino R3 دارای 14 پین ورودی / خروجی دیجیتال است که 6 پین از این 14 پین قابلیت PWM را دارند. این برد با استفاده از نرم افزار Arduino IDE از طریق کابل USB نوع B قابل برنامه ریزی است. برد آردوینو Uno میتواند آن را از طریق کابل USB یا باتری 9 ولتی خارجی تغذیه کرد. این برد ولتاژ بین 7 تا 20 را میتواند به عنوان ورودی استفاده کند زیرا دارای یک رگولاتور داخلی برای تبدیل ولتاژ است.

- Microchip ATmega328P : میکروکنترلر
 - ولت 5 : ولتاژ عملکرد
 - 20 ولت 7 : ولتاژ ورودی
 - (تولید کنند PWM پین میتوانند خروجی 6) 14 : دیجیتال I/O پین های
 - 6 : پین های ورودی آنالوگ
 - میلی آمپر 20 : برای هر پین دیجیتال DC جریان
 - میلی آمپر 50 : برای هر پین 3.3 ولت DC جریان
 - کیلوبایت از هر 0.5 کیلوبایت توسط بوت لودر استفاده میشود 32 : (فلش) Flash حافظه
 - مگا هرتز 16 : کلاک Clock سرعت
 - میلی متر 68.6 : طول
 - میلی متر 53.4 : عرض
 - گرم 25 : وزن



نحوه اتصال قطعات

برای کنترل موتور به درایو نیاز داریم و درایو را نیز به آردوینو متصل می کنیم. ورودی های موتور را به دو پرت output درایو متصل می کنیم که ولتاژ را از درایو به موتور منتقل می کند. برای تامین انرژی چرخش موتور، پرت vs درایو به سر مثبت باتری 9 ولت و ground درایو به سر منفی باتری متصل می شود. پرت ENB به پرت 6 آردوینو متصل می شود (ورودی PWM) که سرعت ورودی دلخواه ما را به درایو منتقل می کند و درایو متناسب با سرعت داده شده ، ولتاژ را به موتور منتقل می کند تا موتور با سرعت خواسته شده ما بچرخد. In3 و In4 به پرت 11 و 12 آردوینو متصل می کنیم که ترکیب Low , High این دو ورودی خاموش یا روشن بودن موتور و جهت چرخش را مشخص می کنند.

Input3	Input4	Spinning direction
Low	Low	Off
High	Low	Forward
Low	High	Backward
High	High	Off

ماژول IR وظیفه تشخیص مقدار سرعت چرخش موتور را به عهده دارد. پرت vcc این ماژول را به 5 ولت آردوینو و ground ماژول را به ground آردوینو متصل می کنیم. خروجی ماژول نیز به پرت دو آردوینو متصل می کنیم تا از سنسور خروجی بگیریم و بعد در کد این خروجی به سرعت موتور تبدیل می کنیم. سنسور IR به تشخیص رنگ سفید حساسیت بیشتری دارد بنابراین یکی از پره های موتور را سفید کردیم تا به ازای هر دور چرخش سنسور یک بار مانع را تشخیص می دهد و به ازای موقعیت های دیگر عدم حضور مانع است. در نتیجه هر یک دور سنسور یک بار فعال می شود. سنسور IR داخل جعبه طوری قرار دادیم که سنسور تشخیص دهنده روبروی پره ها قرار داشته باشد و نویزهای محیطی و مانع های بیرونی روی تشخیص ما تاثیر نگذارد.

برای اینکه سرعت را بتوانیم از راه دور با گوشی تنظیم کنیم به ماژول wifi نیاز داریم. نحوه اتصال این ماژول به آردوینو به این صورت است که پرت ch_Pd و پرت Vcc را باید به ولتاژ 3.3 آردوینو متصل کنیم. ground ماژول به ground آردوینو متصل می کنیم و پرت Tx و ایفای به پرت Rx آردوینو و پرت Rx و ایفای به پرت Tx آردوینو متصل می شود. از آنجا که همه ورودی های و ایفای نباید ولتاژ از 3.3 بیشتر شود ورودی Rx و ایفای را با استفاده از سه مقاومت موازی می کنیم طوری که خروجی اش 3.3 ولت شود.

البته اتصال این پرت ها بعد از آپلود کد وایفای روی برد است. بعد از آپلود کد ها usb را از کامپیوتر جدا می کنیم این اتصالات را برقرار می کنیم دوباره usb را به کامپیوتر متصل می کنیم.

```

1  // #define REMOTEXY__DEBUGLOG
2  // RemoteXY select connection mode and include library
3  #define REMOTEXY_MODE__ESP8266_HARDSERIAL_POINT
4  #include <RemoteXY.h>
5
6
7  // RemoteXY connection settings
8  #define REMOTEXY_SERIAL Serial
9  #define REMOTEXY_SERIAL_SPEED 115200
10 #define REMOTEXY_WIFI_SSID "Esp3"
11 #define REMOTEXY_WIFI_PASSWORD "abcd12345"
12 #define REMOTEXY_SERVER_PORT 6377
13 ///////////////////////////////////////////////////////////////////
14 // RemoteXY include library //
15 ///////////////////////////////////////////////////////////////////
16
17 // you can enable debug logging to Serial at 115200
18 #define REMOTEXY__DEBUGLOG
19 #define IR 2
20 #define Motor_voltage 6
21 float time=0;
22 int det1,det2;
23 float velocity,t=0,vel=0;
24 float sum=0;
25 int time0;
26 int i;
27 float u,ui=0,up;
28 float Kp,Ki;
29 int start_signal=0;
30

```



```

33 // RemoteXY configurate
34 #pragma pack(push, 1)
35 uint8_t RemoteXY_CONF[] = // 36 bytes
36 { 255,1,0,32,0,29,0,16,54,1,4,128,6,75,53,5,2,26,67,5,
37   21,22,20,5,2,26,21,67,5,21,47,20,5,2,26,11 };
38
39 // this structure defines all the variables and events of your control interface
40 struct {
41     | // input variables
42     | int8_t slider_1; // =0..100 slider position
43     |
44     | // output variables
45     | char text_1[21]; // string UTF8 end zero
46     | char text_2[21]; // string UTF8 end zero
47     |
48     | // other variable
49     | uint8_t connect_flag; // =1 if wire connected, else =0
50 } RemoteXY;
51 #pragma pack(pop)
52
53 void setup() {
54     RemoteXY_Init ();
55     //Serial.begin(9600);
56     pinMode(IR, INPUT);
57     pinMode(11,OUTPUT);
58     pinMode(12,OUTPUT);
59     pinMode(6,OUTPUT);
60 }
61
62 void loop() {
63     RemoteXY_Handler ();
64     digitalWrite(11,HIGH);digitalWrite(12,LOW);
65     time0=millis();

```

```

67 velocity=1800;
68 det2= 1;
69 sum=0;
70 i=1;
71 Kp=0.3;
72 Ki=0.0007;
73 ui=0;
74 while(1){
75     RemoteXY_Handler ();
76     velocity=RemoteXY.slider_1*4600/100;
77     if(start_signal){ui+=(velocity-vel)*Ki*t/1000;}
78     up=Kp*(velocity-vel);
79     u =int( up + ui );
80     if(u>255){u=255;}
81     analogWrite(6,u);
82     det1=det2;
83     det2 = digitalRead(IR);
84     if(det1-det2 == 1){
85         start_signal=1;
86         t=millis()-time;
87         time=millis();
88         vel=60/t*1000;
89     }
90     i++;
91     sum+=vel;
92     dtostrf(sum/i/4500,0,1,RemoteXY.text_1);
93     dtostrf(RemoteXY.slider_1,0,1,RemoteXY.text_1);
94     //if(i>800){Serial.print(ui);Serial.print(' ');Serial.print(up);Serial.print(' ');Serial.println(sum/i);i=1;sum=0;}
95 }
96 }
97
98

```

شرح کد:

بخش اول کد اتصال و ایفای و مشخصات و ایفای شامل رمز و اسم و اطلاعاتی از این دست هست. سپس تعدادی متغیر که برای بررسی سرعت مورد نیاز هست معرفی می شود. کاراکتر `test1` , `test2` برای نمایش سرعت واقعی موتور و سرعت ورودی دلخواه ماست .

در `setup` پین های ورودی و خروجی که قبل تر توضیح دادیم معرفی می شوند. در `loop` وضعیت پین های 12 و 11 که تعیین کننده جهت چرخش موتور هستند مشخص می شود. سپس تایم اولیه ثبت می شود ضرایب کنترلر را مشخص می کنیم. در `while` حلقه کنترلی دائم اجرا می شود و بر اسکیل شده عدد اسلایدر یک که مشخص کننده سرعت دلخواه ماست کنترلر را اجرا می کنیم (انتگرال می گیریم و در بخش تناسبی ضرب می کنیم) و این سرعت را به موتور اعمال می کنیم (ارسال داده به پرت 6)

بخش بعدی برای تشخیص سرعت با استفاده از سنسور IR است به این شکل که وقتی وضعیت تغییر می کند یعنی از عدم حضور مانع به حضور مانع می رسیم وارد شرط می شود و مقدار زمان یک دور چرخش می شود زمان حال منهای آخرین باری که حضور مانع را دیده و بعد سرعت با توجه به این اختلاف تایم در لحظه محاسبه می شود و با سرعت های قبلی جمع می شود تا میانگین سرعت واقعی فعلی موتور روی نمایشگر چاپ شود. وقتی کاربر عدد سرعت دلخواه را عوض می کند، کد از ابتدای لوپ دوباره ران می شود و انتگرال گیری و غیره و محاسبه سرعت میانگین از ابتدا انجام می شود.

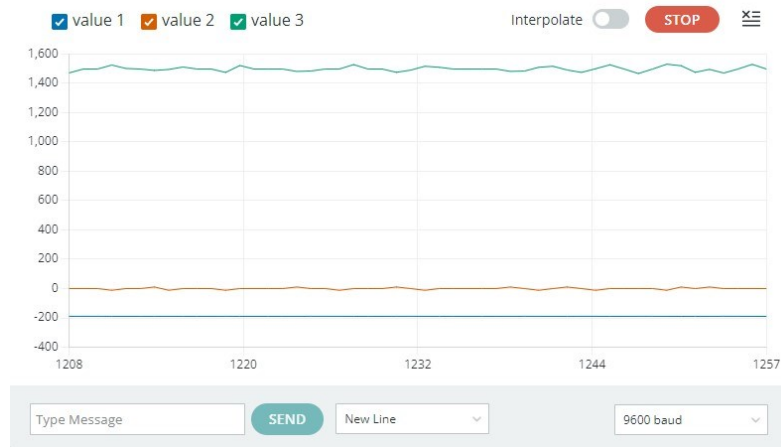
تنظیم ضرایب کنترلر:

برای تنظیم ضرایب کنترلی، ابتدا ضریب K_p نسبت را مقدار یک می‌دهیم و میبینیم که سیستم تغییر سریعی دارد اما هیچ وقت به مقدار خواسته شده نرسیده و همواره آفست دارد. پس باید ضریب K_i انتگرالگیر را نیز تعیین کنیم. با افزایش ضریب انتگرالگیر می‌توانیم ببینیم که در نهایت به مقدار سرعت خواسته شده می‌رسیم. اما زمان نشست زیاد خواهد بود، برای همین نسبت K_p و K_i باید تنظیم شود. در نهایت با انجام سعی و خطا می‌توانیم به مقادیر زیر برای ضرایب کنترلر PI برسیم.

```
Kp=0.3;  
Ki=0.0007;
```

```
if(start_signal){ui+=(velocity-vel)*Ki*t/1000;}  
up=Kp*(velocity-vel);  
u =int( up + ui );  
if(u>255){u=255;}
```

همچنین باید دقت کرد که ورودی $drive$ مجموع دو مقدار up و ui می‌باشد. همینطور برای انتگرال باید بازه زمانی یک گردش کامل t را به عنوان المان زمان در انتگرالگیری در نظر بگیریم و برای تبدیل میلی ثانیه به ثانیه، آن را بر 1000 تقسیم کنیم. در عین حال چون PWM مقدار بالاتر از 255 را نمیگیرد، یک فیلتر اشباع تعریف می‌کنیم تا بر u اثر کرده و مقدار غیرقابل درک به $drive$ ندهیم. برای بازه های سرعت مختلف از 500 تا 3800 دور بر دقیقه، موتور در زمانی کمتر از 5 ثانیه، به مقدار نهایی می‌رسد. (زمان نشست کمتر از 5 ثانیه است).



همانطور از که از نمودار مشاهده می شود، زمانی که سرعت دلخواه 1500rpm را به کنترلر می دهیم، با خطای رندوم کمتر از 50rpm روبهرو می شویم (value3) که به علت اغتشاش ، نویز و رزولوشن IR و PWM می باشد. همینطور ورودی up(value2) و ورودی ui(value1) قابل مشاهده هستند.

