

به نام خدا

آموزش آردوینو جلسه سوم

بصورت پروژه محور با ربات NK-100



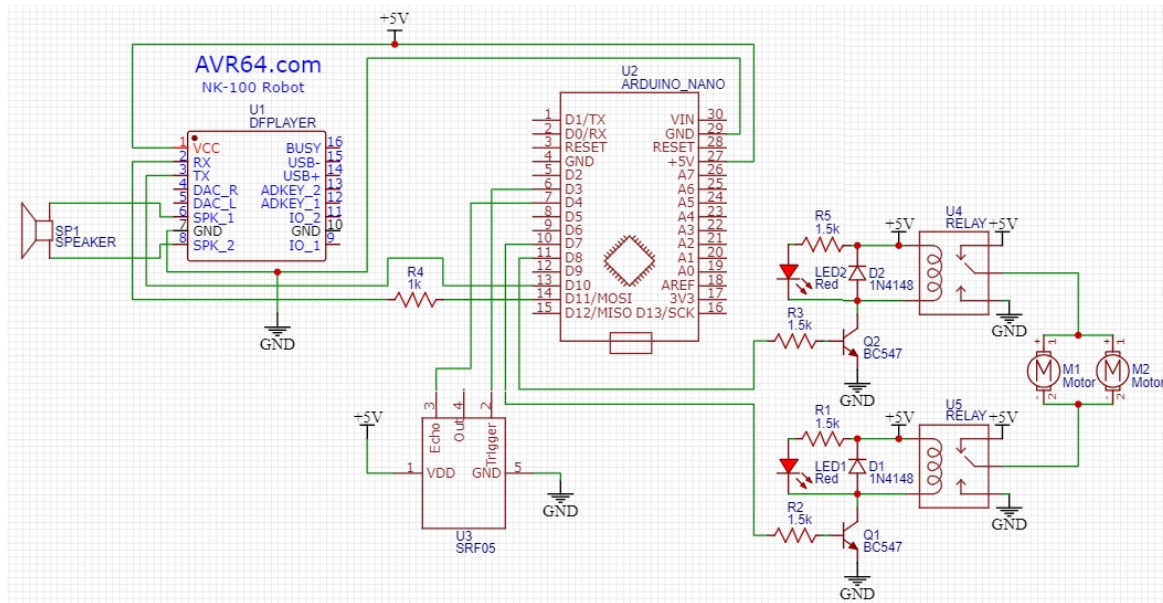
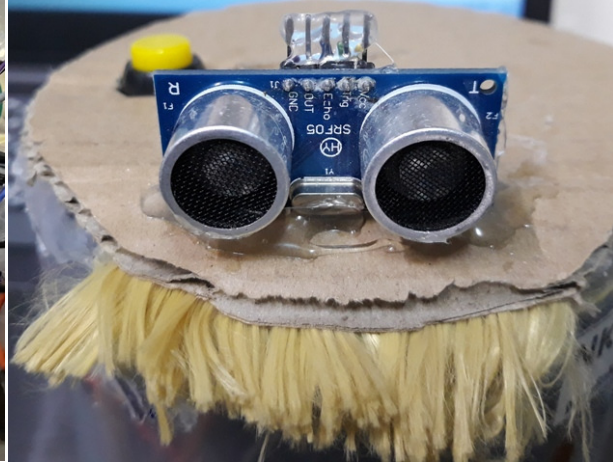
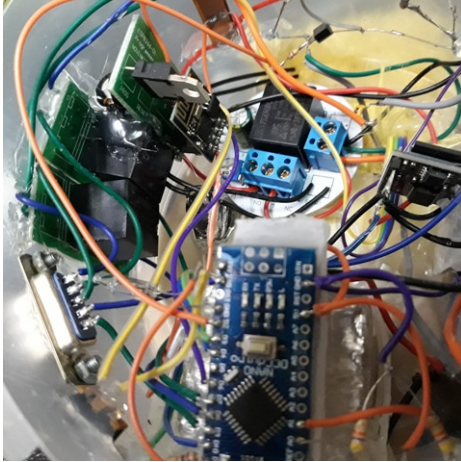
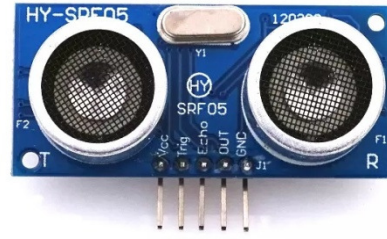
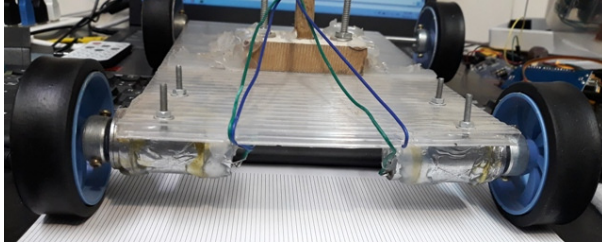
در جلسه قبل با اتصال ماژول DFPlayer روبات را سخنگو کردیم. در این جلسه تصمیم داریم چرخهای روبات را به حرکت درآورده و با اتصال ماژول فاصله سنج آلتراسونیک SRF05 فیدبک لازم را در صورت نزدیک شدن دست به صورت روبات اجرا کنیم.

سخت افزار:

اساس این جلسه بر راه اندازی ماژول آلتراسونیک یا مافوق صوت SRF-05 که با نام های دیگر از قبیل HY-SRF05، SR05 یا SR-05 نیز شناخته می شود استوار است. ماژول SRF05 برای تشخیص فاصله از یک مانع بکار می رود. این ماژول دارای ۵ پایه بوده که تنها از ۴ پایه آن استفاده می کنیم. دو پایه مربوط به ولتاژ (۵ولت) بوده و دو پایه دیگر Trig و Echo نام دارند که یکی ورودی و دیگری خروجی می باشد. با اعمال یک پالس به پایه Trig یک صدای مافوق صوت توسط یکی از مولدهای آلتراسونیک روی ماژول ایجاد شده و به محض برخورد با مانع، انعکاس آن توسط گیرنده آلتراسونیک روی ماژول دریافت می شود و یک پالس روی پایه Echo ایجاد می شود، در صورتی که مانعی در جلوی ماژول وجود نداشته باشد یا فاصله آن بیشتر از چند متر باشد هیچ سیگنالی دریافت نمی شود.

سخت افزار بعدی مربوط به کنترل جهت موتورهای چرخ های ربات می باشد. در این ربات چرخ های جلو محرک بوده و هر محور چرخ به طور مستقیم به یک موتور DC گیربکس دار دورپایین متصل شده است. موتورهای ۱۲ ولت بوده ولی با ۵ ولت نیز بخوبی حرکت می کنند. مطابق با شماتیک برای کنترل جلو و عقب روبات از دو رله معمولی استفاده کرده ایم که با ترانزیستور BC547 و مقاومت 1.5K به پایه های D7 و D8 آردوینو متصل شده اند.

اگر هر دو پایه آردوینو در وضعیت LOW یا صفر باشند هر دو رله خاموش بوده و ربات حرکت نخواهد کرد. در صورتی که یکی از پایه ها HIGH شود یک رله جذب شده و ولتاژ در یک جهت به موتورهای موازی متصل شده و ربات به یک سمت حرکت می کند. در صورتی که پایه دیگر HIGH شود رله دیگر وصل شده و ولتاژ به صورت معکوس به موتورهای متصل شده و ربات در سمت مخالف حرکت می کند. در صورتی که هر دو پایه باهم LOW یا HIGH شوند ربات متوقف می شود.



شماتیک جلسه سوم

کدنویسی:

قبل از شروع کد نویسی اجازه دهید تا یک سناریو و الگوریتم برای ربات خود پیاده‌سازی کنیم. ما قصد داریم روبات به محض روشن شدن سلام کرده و به سمت جلو بیاید و در صورتی که دستمان را جلوی صورتش گرفتیم اخطار داده و سپس به عقب بازگردد تا به به دست ما یا هرمانع دیگری برخورد نکند. (با توجه به این که این جلسات آموزشی در ادامه یکدیگر هستند حتما باید جلسه دوم را اجرا و کتابخانه مربوط به DFPlayer را نیز نصب کرده باشید). کدهای زیر را داخل کامپایلر Arduino وارد نمایید:

```
const int LED = 2;

#include <SoftwareSerial.h>
#include "DFRobotDFPlayerMini.h"

SoftwareSerial mySoftwareSerial(10, 11); // RX, TX
DFRobotDFPlayerMini myDFPlayer;

const int TRIG_PIN = 3;
const int ECHO_PIN = 4;
int ultraDist();
long duration, distanceCm;

const int jolo = 7;
const int aghab = 8;

void setup() {
  // put your setup code here, to run once:
  pinMode(LED,OUTPUT);
  digitalWrite(LED,HIGH);

  pinMode(TRIG_PIN,OUTPUT);
  pinMode(ECHO_PIN,INPUT);

  pinMode(jolo,OUTPUT);
  pinMode(aghab,OUTPUT);

  Serial.begin(9600);

  Serial.println();
  Serial.println(F("DFRobot DFPlayer Mini Demo"));
  Serial.println(F("Initializing DFPlayer ... (May take 3~5 seconds)"));
```

```

mySoftwareSerial.begin(9600);

if (!myDFPlayer.begin(mySoftwareSerial)) { //Use softwareSerial to communicate with mp3.
  Serial.println(F("Unable to begin:"));
  Serial.println(F("1.Please recheck the connection!"));
  Serial.println(F("2.Please insert the SD card!"));
  while(true);
}
Serial.println(F("DFPlayer Mini online."));

myDFPlayer.setTimeout(500); //Set serial communictaion time out 500ms

//----Set volume----
myDFPlayer.volume(30); //Set volume value (0~30). 20 is good

//----Set different EQ----
myDFPlayer.EQ(DFPLAYER_EQ_NORMAL);

//----Set device we use SD as default----
myDFPlayer.outputDevice(DFPLAYER_DEVICE_SD);

myDFPlayer.play(1); //salam
delay(14000);

}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(jolo,HIGH);
  delay(500);
  int cm = ultraDist();
  if(cm>=2 && cm<=15){
    Serial.println(cm);
    myDFPlayer.play(2); //Ekhtar
    digitalWrite(jolo,LOW);
    delay(3000);
    digitalWrite(aghah,HIGH);
    delay(3000);
    digitalWrite(aghah,LOW);
    delay(2000);
  }
}

int ultraDist(){
  // Give a short LOW pulse beforehand to ensure a clean HIGH pulse:
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);

```

```

digitalWrite(TRIG_PIN, HIGH);
delayMicroseconds(10);
digitalWrite(TRIG_PIN, LOW);
duration = pulseIn(ECHO_PIN,HIGH);
// convert the time into a distance
distanceCm = duration / 29.1 / 2 ;
return distanceCm;
}

```

کد این جلسه با ویرایش کد جلسه دوم نوشته شده بنابراین فقط به شرح قسمتهای جدید می پردازیم. در خطوط ۹ و ۱۰ دو ثابت به نام TRIG_PIN و ECHO_PIN تعریف کرده و مقدار آنها را برابر با ۳ و ۴ قرار داده ایم. این مقادیر درواقع پایه هایی از آردوینو هستند که به پایه های Trig و Echo ماژول آلتراسونیک SRF-05 متصل شده اند.

در خط ۱۱ با دستور int ultraDist() تابع سنسور آلتراسونیک را معرفی کرده ایم. این تابع در انتهای برنامه نوشته شده است.

در خط ۱۲ با دستور long duration, distanceCm دو متغیری که در داخل تابع مربوط به سنسور آلتراسونیک به کار رفته است را معرفی نموده ایم، البته می توانید متغیرها را در داخل تابع مربوطه نیز تعریف کنید.

در خطوط ۱۴ و ۱۵ دو ثابت به نام های jolo و aghab تعریف کرده ایم که مربوط به پایه های رله های درایور موتورهای ربات بوده و به پایه های ۷ و ۸ آردوینو متصل شده اند. با HIGH کردن هر کدام از این پایه ها می توانیم ربات را به سمت جلو یا عقب هدایت کنیم.

در داخل تابع setup و در خطوط ۲۲ تا ۲۶ پایه های سنسور آلتراسونیک و موتور را با دستور pinMode ورودی و خروجی کرده ایم، بدین صورت که پایه Echo بعنوان ورودی و بقیه بصورت خروجی تعریف شده اند.

در خط ۵۵ با دستور `myDFPlayer.play(1)` اولین فایل داخل مموری کارت ماژول DFPlayer را که حاوی یک معرفی کوتاه در مورد ربات است پخش می‌کنیم و در خط بعدی با دستور `delay` به مدت ۱۴ ثانیه صبر می‌کنیم تا این فایل به طور کامل پخش شود و بعد از آن عملیات بعدی شروع شود.

بقیه برنامه در داخل حلقه `loop` نوشته شده چرا که باید بطور دائم اجرا شود. در خط ۶۳ با دستور `digitalWrite(jolo,HIGH)` رله متصل به پایه ۷ آردوینو روشن شده و سبب حرکت موتورهای روبات به سمت جلو می‌شود. بعد از آن یک دستور تاخیر نیم ثانیه‌ای قرار داده‌ایم تا نویز احتمالی یا فشاری که موقع استارت موتور به تغذیه وارد می‌شود روی بخش‌های دیگر مثل پالس‌های دریافتی سنسور آلتراسونیک اثری نداشته باشد.

در خط ۶۵ با دستور `int cm = ultraDist()` یک متغیر تعریف کرده و مقدار آن را برابر با نتیجه تابع خواندن از سنسور فاصله یاب SRF05 قرار می‌دهیم، این تابع فاصله تا مانع را بر حسب سانتی‌متر برمی‌گرداند.

در خط ۶۶ با دستور شرطی `if(cm>=2 && cm<=15)` بررسی می‌کنیم که هرگاه جسمی در فاصله بین حدود ۲ الی ۱۵ سانتی متری صورت روبات قرار گرفت دستورات داخل شرط را اجرا کند.

در خط ۶۷ با دستور `Serial.println(cm)` مقدار این فاصله را روی پورت سریال می‌فرستیم که موقع دیباگ کردن پروژه می‌توانید این مقدار را در سریال مانیتور آردوینو مشاهده نمایید.

در خط ۶۸ با دستور `myDFPlayer.play(2)` فایل دوم کپی شده داخل مموری متصل به ماژول DF Player را پخش می‌کنیم که یک صدای اخطار است و با کمک نرم افزار خریداری شده آریانا فایل شرکت عصر گویش پرداز تولید شده است.

در خط ۶۹ با دستور `digitalWrite(jolo,LOW)` رله متصل به پایه ۷ آردوینو LOW شده و روبات فوراً متوقف می‌شود تا به مانع برخورد نکند. و با دستور تاخیر ۳ ثانیه‌ای خط بعدی به مدت ۳ ثانیه در همین وضعیت توقف باقی می‌ماند تا صدای اخطار نیز به طور کامل پخش شود.

در خط ۷۱ با دستور `digitalWrite(aghah,HIGH)` پایه شماره ۸ آردوینو HIGH شده و ربات به سمت عقب حرکت می‌کند و با دستور تاخیر ۳ ثانیه‌ای خط بعدی، به مدت ۳ ثانیه به عقب می‌رود.

در نهایت در خط ۷۲ با دستور `digitalWrite(aghah,LOW)` روبات متوقف شده و با تاخیر ۲ ثانیه‌ای خط بعدی، به مدت ۲ ثانیه در همین وضعیت توقف باقی می‌ماند.

درنهایت برنامه به پایان شرط رسیده و حلقه از بالا مجدداً تکرار می‌شود؛ یعنی ربات مجدداً به سمت جلو حرکت می‌کند تا اینکه به مانع برخورد کند و برگردد.

و اما در آخر به شرح تابع `ultraDist` که در خطوط ۷۹ تا ۹۰ تعریف شده می‌پردازیم. در خط ۸۱ با دستور `digitalWrite(TRIG_PIN, LOW)` پایه `Trig` ماژول `SRF05` را LOW کرده و با دستور `delayMicroseconds(2)` به مدت ۲ میکروثانیه صبر می‌کنیم. سپس در خط بعدی پایه را HIGH کرده و ۱۰ میکروثانیه منتظر می‌مانیم و سپس پایه را مجدداً LOW می‌کنیم. با این دستورات یک پالس مثبت با عرض ۱۰ میکروثانیه به پایه تریگر ماژول آلتراسونیک ارسال می‌کنیم.

در خط ۸۶ با دستور `duration = pulseIn(ECHO_PIN,HIGH)` تابع `pulseIn` را که از توابع داخلی `Arduino` می‌باشد را فراخوانی کرده‌ایم و دو آرگومان به این تابع داده‌ایم. آرگومان اول پایه‌ای است که قصد داریم پالس مدت زمان ورودی روی آن را بسنجیم و آرگومان دوم نیز وضعیت پالس مورد نظر می‌باشد که با عبارت `HIGH` جایگزین کرده‌ایم، یعنی به `Arduino` دستور می‌دهیم که پایه `ECHO_PIN` یا همان پایه شماره ۴ آردوینو که به پایه `Echo`ی ماژول فاصله سنج آلتراسونیک متصل شده را اسکن کند و مدت زمانی که از وضعیت `LOW` به `HIGH`

رفته و مجدداً LOW شده را اندازه‌گیری نماید (اندازه عرض پالس را حساب کند) و اندازه این پالس را بر حسب میکروثانیه در متغیر duration قرار بدهد. توجه داشته باشید که ماژول آلتراسونیک SRF05 بعد از تریگر شدن یک صوت فرکانس بالا ارسال کرده و پس از برخورد به مانع و برگشت به گیرنده، زمان این رفت و برگشت را محاسبه و بر حسب میکروثانیه بصورت یک پالس روی پایه Echo تولید می‌کند. اگر مانعی جلوی سنسور نباشد هیچ پالسی تولید نمی‌شود و دستور pulseIn پس از Timeout شدن (یک ثانیه پس از اینکه پالسی دریافت نکرد) با مقدار صفر خارج می‌شود. (آرگومان سوم دستور pulseIn میزان تایم‌اوت است که بصورت پیش‌فرض روی یک ثانیه می‌باشد و در صورت تمایل می‌توانید با ارسال آرگومان سوم این مقدار را بازنویسی نمایید).

در خط ۸۸ با دستور $\text{distanceCm} = \text{duration} / 29.1 / 2$ این پالس را بر ۲ تقسیم می‌کنیم تا زمان رفت و برگشت را به زمان رفت تبدیل کنیم و با تقسیم بر 29.1 نیز با توجه به سرعت صوت در هوا این زمان را تبدیل به سانتی‌متر می‌نماییم. در خط آخر نیز مقدار را به خروجی تابع برمی‌گردانیم.

به این ترتیب با ترکیب ماژول تشخیص فاصله SR05 و DFPlayer و نیز قابلیت حرکت توانستیم یک روبات خودکار طراحی کنیم که تا حدی هوشمند بوده و می‌تواند از برخورد با موانع جلوگیری کرده و فیدبک صوتی مناسب را به کاربر ارائه دهد و به تعامل با محیط بپردازد. با تغییرات بیشتر در نرم‌افزار و سخت‌افزار می‌توانید قابلیت یادگیری و حافظه به روبات بدهید تا در صورت تکرار چند باره جلوگیری از حرکت در بازه زمانی کوتاه، پیام‌های دیگری به کاربر ارائه دهد تا از حالت یکنواخت و تکراری خارج شود. برای افزودن قابلیت یادگیری به روبات باید مطالبی در خصوص هوش مصنوعی و یادگیری عمیق یا Deep Learning مطالعه نمایید.

منابع:

<https://www.aliexpress.com/item/32828459924.html> [SRF05 Picture]

در این جلسه با ماژول SRF05 و نحوه فاصله‌یابی و تشخیص مانع و نیز کنترل جهت موتورهای گیربکس‌دار ربات NK-100 آشنا شدیم. در جلسه بعد ان‌شاءالله با Servo موتور آشنا می‌شویم و گریپر یا دست روبات را که از یک سروو موتور تشکیل شده کنترل خواهیم کرد.

مolf بهنام زکی‌زاده – ۱۴۰۰/۰۱/۰۲

AVR64.com
