



خط کش دیجیتال با استفاده از ماژول التراسونیک SRF-04/SRF-05



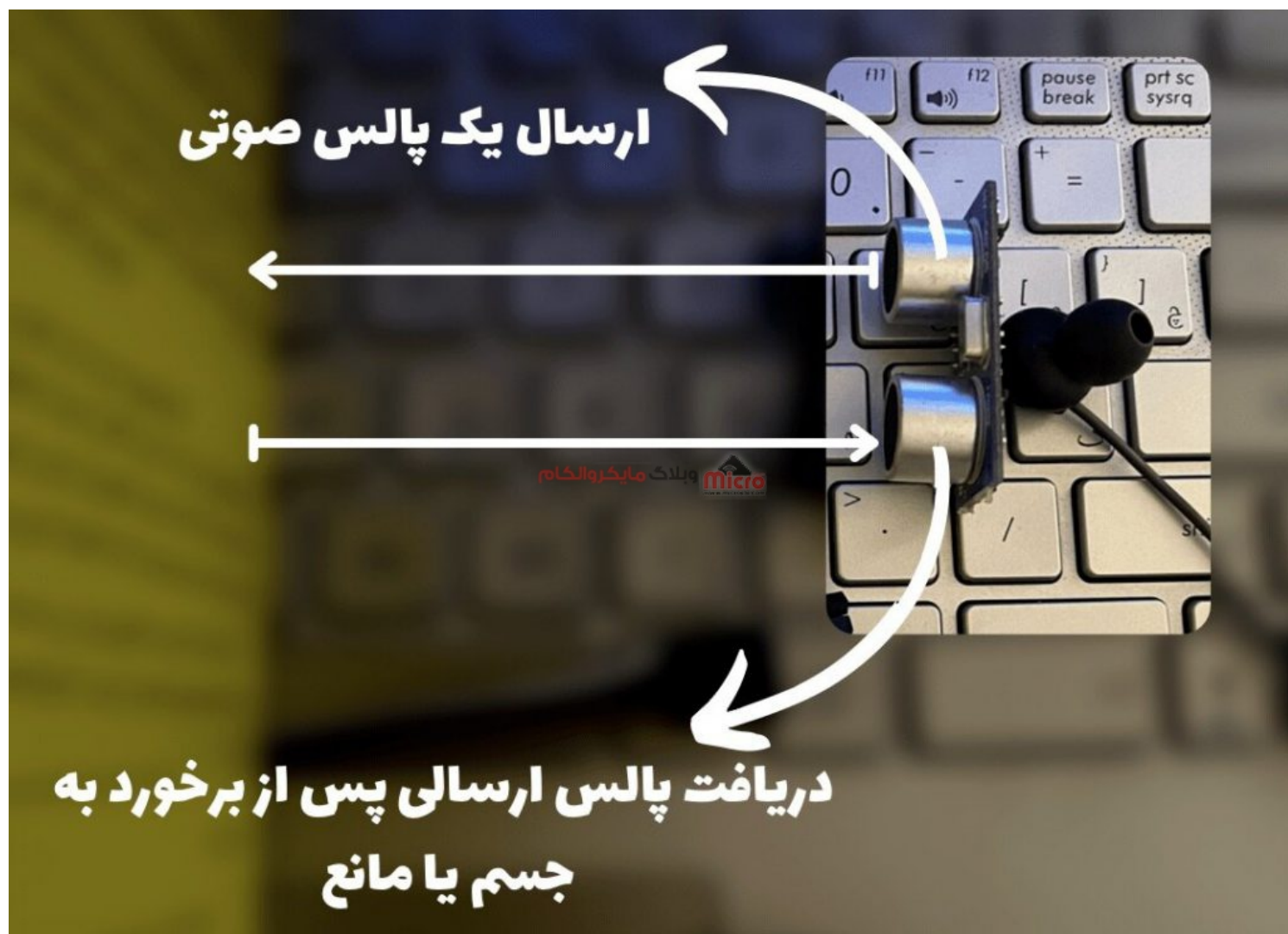
تاریخ انتشار ۱۲ تیر، ۱۴۰۱ توسط آرمان حاجی علینقی

سلام به همه شما میکروالکامی ها. در **مطلب قبلی به راه اندازی نمایشگر TFT** پرداخته شد و مفصل در باره آن صحبت شد. در این مطلب به نحوه استفاده از سنسور التراسونیک (SRF-05 / SRF-04) برای ساخت یک خط کش دیجیتال و نمایش اندازه یا فاصله بر روی نمایشگر رنگی TFT پرداخته خواهد شد. پس با من تا انتهای مطلب همراه باشید. همچنین شما میتونید سایر مطلب من رو از **این قسمت** مطالعه کنید.



ماژول التراسونیک (SRF-05 یا SRF-04)

در این ماژول دو قطعه وجود دارد که شبیه به بلندگو هستند. یکی وظیفه ارسال پالس صوتی را داشته و دیگری وظیفه دریافت پالس صوتی ارسال شده را دارد. این ماژول یک پالس صوتی 8 بیتی به کمک آیسی ماکس 3232 تولید می‌کند. پس از ارسال پالس صوتی تولید شده و بعد از برخورد به یک جسم، این پالس مجدداً به ماژول باز خواهد گشت. گیرنده موجود روی ماژول این پالس را دریافت کرده و با استفاده از آیسی LM234 سیگنال دریافتی را تقویت کرده که در خروجی ماژول قابل دریافت و پردازش باشد.



نحوه عملکرد ماژول التراسونیک SRF-04, SRF-05

اندازه گیری فاصله با ماژول التراسونیک SRF-04/SRF-05

در اصل ما با این ماژول زمان رفت و برگشت پالس صوتی را اندازه گیری می‌کنیم. به همین دلیل سرعت و پارامتر های مربوط به سرعت (دما و رطوبت محیط) برای ما حائز اهمیت هستند. لذا سیگنالی که از ماژول دریافت می‌شود در یک عدد مناسبی ضرب شده و حاصل را تقسیم بر 2 خواهیم کرد. (آن عدد ضرب شده همان عدد مربوط به سرعت بوده و برای به دست آوردن اندازه دقیق در دمای محیط معمولی است).



نکته: چون زمان رفت و برگشت را محاسبه می‌کنیم، باید عدد را تقسیم بر 2 کنیم.

اتصالات ماژول التراسونیک SRF-04/SRF-05 به آردوینو

آردوینو	ماژول التراسونیک
GND	GND
+5V	VCC
6	Trig
5	Echo

نمایشگر TFT

TFT مخفف Thin Film Transistor و نوعی از نمایشگر های با کیفیت بالا می‌باشند. لذا با توجه حساسیت کار یا نوع کاربرد می‌توان از این نمایشگر ها بهره برد. البته قیمت این نوع نمایشگر ها از مدل های کاراکتری بیشتر است. در مطلب قبلی بطور کامل نمایشگر TFT و نحوه راه اندازی آن و نمایش اشکال مختلف و متن بر روی آن توضیح داده شده است. برای دسترسی به این مطلب از [این لینک](#) استفاده کنید.



نمایشگر TFT

اتصالات نمایشگر TFT به آردوینو

آردوینو نانو	نمایشگر TFT
3.3v / +5v	LED



PIN13	SCK
PIN11	SDA
PIN9	A0
PIN8	RESET
PIN10	CS
GND	GND
+5V	VCC

پس از برقرای اتصالات مطابق جدول بالا، کدی که در ادامه مطلب قرار داده شده است را بر روی برد آردوینو آپلود نمایید.

```
#include <Ucglib.h>

int trig = 6;
int echo = 5;
int a = 0;
Ucglib_ILI9341_18x240x320_HWSPI ucg(9, 10, 8);

void setup()
{
  Serial.begin(9600);
  pinMode(trig, OUTPUT);
  pinMode(echo, INPUT);
  // ucg.begin(UCG_FONT_MODE_TRANSPARENT);
}
```



```
ucg.begin(UCG_FONT_MODE_SOLID);
ucg.clearScreen();
ucg.setColor(255, 255, 255);
ucg.drawRFrame(2, 2, 124, 124, 8);

ucg.setFont(ucg_font_ncenR08_hf);
ucg.setColor(0, 20, 255, 20);
ucg.setPrintPos(14, 20);
ucg.print("in the name of GOD");
//ucg.setColor(0, 255, 255, 255);

ucg.setFont(ucg_font_ncenR08_hf);
ucg.setPrintPos(14, 45);
ucg.print("product name:");
ucg.setFont(ucg_font_ncenR12_hf);
ucg.setPrintPos(14, 64);
ucg.print("digital ruller");

ucg.setFont(ucg_font_ncenR08_hf);
ucg.setPrintPos(14, 88);
ucg.print("Direct by:");
ucg.setFont(ucg_font_ncenR10_hf);
ucg.setPrintPos(10, 106);
ucg.print("Arman.H.R");

delay(2000);
//*****
ucg.clearScreen();
ucg.begin(UCG_FONT_MODE_SOLID);
ucg.clearScreen();
ucg.setColor(255, 255, 255);
ucg.drawRFrame(2, 2, 124, 124, 8);
```



```
ucg.setFont(ucg_font_inr16_mr);
ucg.setColor(0, 0, 0, 255);
ucg.setColor(1, 0, 0, 0);
ucg.setPrintPos(4, 25);
ucg.print("Hello");

ucg.setFont(ucg_font_courR12_mr);
ucg.setColor(0, 255, 0, 0);
ucg.setPrintPos(60, 30);
ucg.print("world!");
}

void loop()
{
  digitalWrite(trig, LOW);
  delay(0.002);
  digitalWrite(trig, HIGH);
  delay(0.01);
  digitalWrite(trig, LOW);
  int duration = pulseIn(echo, HIGH);
  int distances = (duration / 2) * 0.0343;
  Serial.println(distances);

  ucg.setFont(ucg_font_ncenR10_hf);
  ucg.setColor(0, 255, 255, 255);
  ucg.setPrintPos(10, 74);
  ucg.print("distances...");

  ucg.setFont(ucg_font_courB18_mr);
  ucg.setColor(0, 50, 255, 50);
  ucg.setPrintPos(10, 100);
  ucg.print(distances) && ucg.print("cm");
```




```
if (distances > 99)
    a = 1;

if (distances < 99 && a == 1)
{
    Clear();
    a = 0;
}
}
void Clear() {
    ucg.setColor(0, 0, 0, 0);
    ucg.drawBox(10, 80, 86, 20);
    //return;
}
```

مشاهده نتیجه



اندازه گیری فاصله با ماژول التراسونیک و نمایش بر روی نمایشگر TFT

تحلیل کد

```
#include <Ucglib.h>

int trig = 6;
int echo = 5;
int a = 0;
Ucglib_ILI9341_18x240x320_HWSPI ucg(9, 10, 8);
```



در ابتدا کتابخانه مربوط راه اندازی نمایشگر TFT را فراخوانی می‌کنیم. سپس پایه های Trig و Echo مربوط به ماژول التراسونیک (SRF04 / SRF-05) را آدرس دهی خواهیم کرد. همچنین پایه های RESET، CS، A0 مربوط به نمایشگر را نیز آدرس دهی می‌کنیم. در ادامه درباره متغیر a توضیح داده خواهد شد.

```
Serial.begin(9600);  
pinMode(trig, OUTPUT);  
pinMode(echo, INPUT);
```

در ابتدای void setup مقدار باودریت برای سریال مانیتور را مشخص و در ادامه پایه Trig را خروجی و پایه Echo را ورودی تعریف کردیم. این پایه ها همان فرستنده و گیرنده ماژول التراسونیک هستند. به طوری که پایه Trig فرستنده و پایه Echo گیرنده می‌باشند.

```
ucg.begin(UCG_FONT_MODE_SOLID);  
ucg.clearScreen();  
ucg.setColor(255, 255, 255);  
ucg.drawRFrame(2, 2, 124, 124, 8);
```

در حلقه void setup کانفیگ اولیه نمایشگر برای شروع به کار را انجام می‌دهیم. در ادامه با دستور ucg.clearScreen() گفتیم نمایشگر از نمایش هر چیزی خالی شود و سپس با دستور setColor رنگ پس زمینه که در اینجا مشکی است را مشخص کردیم. با دستور drawFrame مربعی در مختصات (2,2) به طول و عرض 124 پیکسل رسم شد. پارامتر آخر در این دستور که عدد 8 هست مربوط به این هست که راس های مربع را بصورت منحنی و به اندازه 8 پیکسل رسم کند که راس های مربع ما نوک تیز نباشد.

به طور کل

با دستور seFont فونت مورد نظر برای نوشتن متن را انتخاب می‌کنیم. با دستور (setColor(0, R, G, B), رنگ متن را انتخاب کردیم که به هر کدام از مقادیر R G B اعدادی بین 0 تا 255 داده می‌شود. با دستور (setPrintPos(x,y), مختصات شروع نوشتن متن مشخص می‌شود. در نهایت با دستور print() متن دلخواه را بر روی نمایشگر، خواهیم نوشت.

نکته: اگر می‌خواهید یک متن نمایش دهید باید آن را داخل "" قرار داد. اما در صورتی که بخواهیم اطلاعات حاوی یک



سنسور یا ماژول که در متغیری ذخیره شده را نمایش دهیم، مستقیماً مقدار آن متغیر را در آرگومان تابع print نمایشگر قرار خواهیم داد.

```
ucg.setFont(ucg_font_ncenR08_hf);
ucg.setPrintPos(14, 45);
ucg.print("product name:");
ucg.setFont(ucg_font_ncenR12_hf);
ucg.setPrintPos(14, 64);
ucg.print("digital ruller");

ucg.setFont(ucg_font_ncenR08_hf);
ucg.setPrintPos(14, 88);
ucg.print("Direct by:");
ucg.setFont(ucg_font_ncenR10_hf);
ucg.setPrintPos(10, 106);
ucg.print("Arman.H.R");

delay(2000);
//*****
ucg.clearScreen();
ucg.begin(UCG_FONT_MODE_SOLID);
ucg.clearScreen();
ucg.setColor(255, 255, 255);
ucg.drawRFrame(2, 2, 124, 124, 8);

ucg.setFont(ucg_font_inr16_mr);
ucg.setColor(0, 0, 0, 255);
ucg.setColor(1, 0, 0, 0);
ucg.setPrintPos(4, 25);
ucg.print("Hello");

ucg.setFont(ucg_font_courR12_mr);
```



```
ucg.setColor(0, 255, 0, 0);  
ucg.setPrintPos(60, 30);  
ucg.print("world!");
```

اندازه گیری فاصله با ماژول التراسونیک

داستان اصلی پروژه و اندازه گیری فاصله و نمایش آن بر روی نمایشگر، در حلقه Void loop صورت خواهد گرفت. در ابتدا پایه Trig را خاموش می‌کنیم. سپس با ایجاد یک تاخیر پایه Trig را فعال می‌کنیم. نهایتاً با یک تاخیر دیگر مجدداً آن را قطع خواهیم کرد.

```
digitalWrite(trig, LOW);  
delay(0.002);  
digitalWrite(trig, HIGH);  
delay(0.01);  
digitalWrite(trig, LOW);
```

اندازه گیری زمان

• با استفاده از دستور PulseIn مدت زمانی کاری که داخل پرانتز است را اندازه گیری می‌کنیم.

```
int duration = pulseIn(echo, HIGH);
```

• در این قسمت مشخص کردیم مدت زمانی که پایه Echo دارای مقدار HIGH است (یعنی در حال دریافت پالس برگشتی) را محاسبه و آن را داخل متغیر duration ذخیره نماید.

• چون مدت زمان ارسال پالس و دریافت پالس را محاسبه می‌کنیم و در اصل 2 زمان است، برای تفکیک این دو از هم و فقط بررسی زمان برگشت عدد حاصل شده را بر 2 باید تقسیم کرد و عدد بدست آمده را در سرعت صوت ضرب می‌کنیم. نهایتاً عددی که بدست خواهد آمد همان فاصله ماژول تا جسم مورد نظر برحسب سانتی متر خواهد بود.

```
int distances = (duration / 2) * 0.0343;
```



- در این قسمت مشخص شد که با فونت ncnR10 و رنگ سفید در مختصات (10,74)، عبارت "distances..." نوشته شود.

```
ucg.setFont(ucg_font_ncnR10_hf);  
ucg.setColor(0, 255, 255, 255);  
ucg.setPrintPos(10, 74);  
ucg.print("distances...");
```

- در این قسمت هم گفته شده با فونت courB18 و با رنگ سبز در مختصات (10,100)، عددی که در متغیر distances ذخیره شده (همان عدد فاصله) بر روی نمایشگر نوشته شود و در کنار آن نیز "cm" نمایش داده شود.

```
ucg.setFont(ucg_font_courB18_mr);  
ucg.setColor(0, 50, 255, 50);  
ucg.setPrintPos(10, 100);  
ucg.print(distances) && ucg.print("cm");
```

بررسی دقیق سورس کد

تا این جا برنامه کامل بوده اما شاهد یک مشکل خواهیم بود. مشکل این است که وقتی عدد جدیدی روی نمایشگر نشان داده می شود عدد قدیمی پاک نمی شود. برای رفع این مشکل باید هر وقت تعداد ارقام زیاد شد، مثلا از عدد 2 رقمی به 3 رقمی، یا به عبارتی دیگر از عدد 99 به بالا، یک مستطیل سیاه رنگ، دقیقا هم رنگ پس زمینه (مشکی) رسم کرده که عملا مقدار دیتای قبلی رو پوشش داده و دیگر عدد جدید با عدد قبلی تداخل نیابد و روی همدیگر نیفتد. لذا از کد زیر به این منظور استفاده می شود.

```
if (distances > 99)  
    a = 1; // مشخص کردن اعداد 3 رقمی  
  
if (distances < 99 && a == 1)  
{
```



```
Clear();  
a = 0; // مشخص کردن اعداد 2 رقمی  
}
```

تابع Clear

هنگام فراخوانی این تابع در کدهای خود، مستطیلی به رنگ مشکی در مختصات ذکر شده و به طول 86 در 20 رسم خواهد شد و عدد نمایش داده قبلی بر روی نمایشگر را محو خواهد کرد.

```
void Clear() {  
    ucg.setColor(0, 0, 0, 0);  
    ucg.drawRect(10, 80, 86, 20);  
}
```

نتیجه گیری

در این مطلب نحوه استفاده و کار با ماژول التراسونیک SRF-04/SRF-05 و اندازه گیری فاصله مورد بررسی قرار گرفت. همچنین جهت نمایش فاصله بدست آمده بر روی نمایشگر نیز از یک نمایشگر TFT استفاده شد و فاصله بر حسب سانتی متر محاسبه و بر روی آن نمایش داده شد.

امیدوارم از این آموزش کمال بهره را برده باشید. در صورت داشتن هرگونه نظر یا سوال درباره این آموزش اون رو در انتهای همین صفحه در قسمت دیدگاه ها قرار بدید. در کوتاه ترین زمان ممکن به اون ها پاسخ خواهم داد. اگر این مطلب براتون مفید بود، اون رو حتما به اشتراک بگذارید. همینطور میتونید این آموزش را پس از اجرای عملی توی اینستاگرام با هشتگ #microelecom به اشتراک بگذارید و **پیج مایکروالکام** (@microelecom) رو هم منشن کنید.