



آموزش جامع راه اندازی نمایشگر TFT – نمایش اشکال هندسی و متن



تاريخ انتشار ١١ اسفند, ١۴٠٠ توسط آرمان حاجى علينقى

سلام به همه شما مایکروالکامی ها. همیشه نمایشگر یکی از اجزای ارتباطی میان دستگاه با کاربر نهایی میباشد. نمایشگر ها دارای انواع مختلف و کاربرد های منحصر به فرد خود نیز هستند. در بین نمایشگر های رایج مدل TFT آن دارای کیفیت نمایش بهتر و جذاب تری میباشد. در این مطلب نحوه راه اندازی کامل نمایشگر TFT اینچ با آردوینو بررسی خواهد شد. پس با من تا انتهای مطلب همراه باشید. همچنین شما میتونید سایر مطلب من رو از این





قسمت مطالعه كنيد.

نمایشگر TFT

TFT مخفف Thin Film Transistor و نوعی از دسته نمایشگر های LCD اما با کیفیت نمایش بهتر میباشد. دلیل این مزیت به خاطر وجود خازن در هرپیسکل نمایشگر هست که کنترل بیشتری روی رنگ ها و تصاویر وجود دارد. همچنین در نمایشگر های LCD از نوعی مایع کریستالی تولید شده و زمانی که میخواهید تصویری را به نمایش بگذارید این مایع به آهستگی حرکت کرده و به همین دلیل باعث کدری و تار شدن تصویر خواهد شد. زمانی که از نمایشگر های TFT استفاده میکنیم این سرعت حرکت مایع با سرعت حرکت تصویر تا حدی در تعادل هست که در نتیجه این مشکل تاری و کدری حل میگردد.







نمایشگر *TFT* مدل 1.8 اینچ

این نمایشگر ها از سطر و ستون تشکیل شده اند. بعنوان مثال وقتی میگوییم نمایشگر 168x128 بدین معناست که این نمایشگر از 128 ستون و 168 سطر تشکیل شده و با دادن مختصات همین سطر و ستون ها میتوان مشخص کرد نقطه شروع نمایش تصویر یا متن از کجا باشد.





اتصالات ما ثول TFT 1.8 اینچ به آردوینو

پایه های ماژول را به ترتیب زیر به آردوینو متصل کنید.

آردوينو	نمایشگر TFT
3.3V/5V	LED
pin13	SCK
pin11	SDA
pin9	A0
pin8	RESET
pin10	CS
GND	GND
5V	VCC

نصب کتابخانه Ucglib برای راه اندازی نمایشگر TFT 1.8 در Arduino IDE

برای راه اندازی این ماژول میتوان از کتابخانه های مختلفی استفاده کرد که در این آموزش از کتابخونه ucglib استفاده شده است. ابتدا باید این کتابخانه را بصورت جداگانه دانلود و نصب کرد. اما روش آسان تر و سریع تر این هست که با استفاده از نرم افزار آردوینو کتابخانه را نصب کنید. برای اینکار مسیر زیر را دنبال نمایید و در پنجره باز شده ucglib را سرچ و نصب نمایید.

Sketch --> Include Library --> Manage Libraries





e All 🗸 Topic All	ucglib	
cglib y oliver Version 1.5.2 INSTALL rue color TFT and OLED library, LI9486,LD50T6160, PCF8833, S lepth. Supported display control SD1251, HY2252	ED Up to 18 Bit color depth. Supported display controller: ST7735, iEPS225, SSD1331, SSD1351, HX8352C. True color TFT and OLE ler: ST7735, ILI9163, ILI9325, ILI9341, ILI9486,LD50T6160, P	ILI9163, ILI9325, ILI9341, D library, Up to 18 Bit color CF8833, SEPS225, SSD1331,
<u>tore info</u> Select version ↓ Install	وبلاگ مایکروالکام	

نصب کتابخانه ucglib برای راه اندازی نمایشگر TFT1.8 اینچ

برنامه نویسی و راه اندازی TFT 1.8 اینچ با آردوینو

در ابتدا همان طور که گفته شد کتابخانه Ucglib را باید فراخوانی کرد. همچنین پایه هایی که از نمایشگر به آردوینو متصل شده اند را آدرس دهی خواهیم کرد.

#include <Ucglib.h>
Ucglib_ILI9341_18x240x320_HWSPI ucg(A0, CS, RESET);





این دو عدد مربوط به سایز نمایشگر شما میباشد که در هنگام خرید نوشته شده نمایشگر مورد نظر چند میلی متر در چند میلی متر هست .

در نهایت در قسمت void setup کد زیر را برای شروع به کار کردن نمایشگر خواهیم نوشت. در ادامه برخی از توابع این کتابخانه معرفی خواهد شد.

ucg.begin(UCG_FONT_MODE_SOLID);

پا کردن صفحه با ()ucg.clearScreen

با این دستور همه متن و اشکال در نمایشگر به طور کامل پاک میشود و نمایشگر را آماده میکنیم که دیتا جدید را نمایش دهد .







عملکرد تابع ucg.clearScreen در نمایشگر TFT

(ucg.setColor(R, G, B انتخاب رنگ با

با استفاده از این تابع شما میتوانید رنگ مورد نظرتون برای انجام هر کاری مثل رسم شکل یا نوشتن متنی جایگزین مقادیر ورودی کنید. مقدار دهی برای هر کدام از رنگ ها از 0 تا 255 میباشد بر همین اساس میتوانید رنگ دلخواه را انتخاب نمایید.

تعیین زاویه برای شروع به کار

قبل از شروع کردن با دستورات مختلف برای رسم اشکال و یا نوشتن متون مختلف باید مشخص کنید که میخواهید در چه زاویه ای با نمایشگر کار کنید به عبارتی افقی یا عمودی بودن نمایش را مشخص کنید. برای مشخص کردن این زوایا از دستورات زیر استفاده میشود.

حالت عمودى

ucg.undoRotate();

چرخش 90 درجه _ افقی

ucg.setRotate90();

چرخش 180 درجه _ عمودی

ucg.setRotate180();

چرخش 270 درجه _ افقی

ucg.setRotate270();





اگر میخواهید چندین نوشته در چند زاویه مخالف بنوسید بهتره از دستور زیر استفاده کنید. مقادیر مختلف Option شامل حالات زیر میباشد.

ucg.setPrintDir(Option);

- اگر مقدار Opiotn برابر 0 باشد: با زاویه 0 درجه متن شما نوشته میشود.
- اگر مقدار Opiotn برابر 1 باشد: با زاویه 90 درجه متن شما نوشته میشود.
- اگر مقدار Opiotn برابر 2 باشد: با زاویه 180 درجه متن شما نوشته میشود.
- اگر مقدار Opiotn برابر 3 باشد: با زاویه 270 درجه متن شما نوشته می شود.







زوایای نمایش برروی نمایشگر *TFT 1.8* اینچ





رسم فريم 1

ucg.drawFrame(x, y, x', y');

- x: از چه ستونی شروع به رسم فریم انجام شو
- y: از چه سطری شروع به رسم فریم انجام شود
 - 'x: مشخص کردن طول فریم
 - 'y' مشخص کردن عرض فریم

رسم فريم 2

با استفاده از دستور زیر که در واقع مقادیر آرگومان آن همانند دستور قبل است میتوان فریم خود را با لبه های زاویه دار رسم کرد. کافی است مقدار مورد نظر در لبه را با z جایگزین کرد. طبعا هر چه این عدد بیشتر باشد، گوشه های منحنی تری خواهیم داشت.

ucg.drawRFrame(x, y, x', y', z);







رسم فریم زاویه دار در نمایشگر *TFT*

تغییر فونت و نوشتن

یکی از مزایای خوب این کتابخانه داشتن فونت های مختلف در اندازه های متفاوت هست که شما میتوانید از هرکدام آنها برای نوشتن متن خود استفاده کنید. با استفاده از <mark>این لینک</mark> میتوانید شکل فونت و کد آن فونت رو یک جا ببینید و استفاده کنید. با استفاده از دستور زیر میتوانید فونت مورد نظر خود را انتخاب و سپس با تابعی که در ادامه معرفی میشود متن مورد نظر خود را نمایش دهید.





. فونت مورد نظر خود را باید داخل آرگومان این تابع وارد نمایید// ; (ucg.setFont (ucg_font_ncenR08_hf)

تنظيم مختصات براى نوشتن

ucg.setPrintPos(x, y);

نوشتن و نمایش متن مورد نظر

ucg.print ("متن مورد نظر خود را بین دابل کوتیشن وارد نمایید");

نکته: اکثر فونت هایی که برای این کتابخانه قرار دارن مربوط به زبان انگلیسی هست و اگر شما بخواهید متن فارسی رو بنویسید باید از فونت فارسی یا عربی این کتابخانه استفاده کنید که در بالاتر منبع فونت و کد ها داخل لینک قرار گرفته شده.

مثال

```
#include <Ucglib.h>
Ucglib_ILI9341_18x240x320_HWSPI ucg(9, 10, 8);
void setup()
{
    ucg.begin(UCG_FONT_MODE_SOLID);
    ucg.clearScreen();
}
void loop()
{
    ucg.setFont(ucg_font_ncenR10_hf);
    ucg.setPrintPos(4,40);
    ucg.print("Mental relaxation");
```





}



نمایش متن روی نمایشگر *TFT*

نکته: در دستور ucg.setColor(option,R, G, B) پارامتر اختیاری به نام آپشن وجود داریم. اگر این مقدار 1 باشد رنگ خود نوشته را میتوان تعیین و اگر مقدار آن برابر 0 قرار باشد رنگ بک گراند نوشته را میتوان مشخص نمود. به مثال زیر توجه نمایید.





```
#include <Ucglib.h>
Ucglib_ILI9341_18x240x320_HWSPI ucg(9, 10, 8);
void setup() {
    ucg.begin(UCG_FONT_MODE_SOLID);
    ucg.clearScreen();
    ucg.setColor(1, 0, 0, 0);
}
void loop() {
    ucg.setFont(ucg_font_ncenR14_hf);
    ucg.setColor(0,255 , 0, 255);
    ucg.setPrintPos(10, 30);
    ucg.print("Hello world!");
}
```







نمایش متن با بک گراند دلخوا در نمایشگر *TFT*





رسم خط افقی

ucg.drawHLine(x, y, line);

- x: ستون
- y: سطر
- line: تا ستون شماره چند رسم شود یا به عبارت دیگر طول خط افقی را مشخص میکنیم.

رسم خط عمودی

ucg.drawVLine(x, y, line);

- x: ستون
- y: سطر
- line: تا سطر شماره چند رسم شود یا به عبارت دیگر طول خط عمودی را مشخص میکنیم.

رسم خط مورب

در حقيق اين تابع ادغام دو تابع قبل است يعنى مىتوان علاوه بر خط افقى و عمودى، خط مورب هم رسم كرد.

ucg.drawLine(x1, y1, x2, y2);

- x1: از چه ستونی شروع به رسم خط انجام شود.
- y1 : از چه سطری شروع به رسم خط انجام شود.
 - x2: تا سطر شماره چند خط ادامه یابد.
 - y2 : تا ستون شماره چند این خط ادامه یابد.







رسم خط در نمایشگر *TFT*

مثال

#include <Ucglib.h>
Ucglib_ILI9341_18x240x320_HWSPI ucg(9, 10, 8);





```
void setup()
{
    ucg.begin(UCG_FONT_MODE_SOLID);
    ucg.clearScreen();
    ucg.setColor(1, 0, 0, 0);
}
void loop() {
    ucg.setColor(0, 0, 255, 0); // دن رنگ // (0, 255, 0);
    ucg.drawHLine(60, 40, 40);
    ucg.drawVLine(60, 64, 40);
    ucg.drawLine(40, 10, 10, 40);
}
```

رسم شکل _ مستطیل

قبل از توضیحات مربوط به این بخش باید یاد آوری کنم که پیشتر رسم مستطیل تو خالی که همان رسم فریم هست بیان شد. برای رسم مستطیل تو پر هم روند کار مشابه قبل است. برای رسم مستطیل تو پر 2 دستور داریم که به شرح زیر میباشند.

ucg.drawBox(x1,y1,x2,y2);

ucg.drawRBox(x1,y1,x2,y2,C);

(ucg.drawBox(x1,y1,x2,y2 دستور

- x1: از چه ستونی شروع به رسم مستطیل انجام شود
- y1: از چه سطری شروع به رسم مستطیل انجام شود
 - x2: نقطه پایانی یا طول مستطیل
 - y2: نقطه پایانی یا عرض مستطیل





(ucg.drawRBox(x1,y1,x2,y2,C دستور

از این دستور برای رسم مستطیل استفاده میشود. تفاوتی که بین این دستور و قبلی وجود دارد این است که در این دستور میتوان گوشه های مستطیل را زاویه دار کرد.

- x1: از چه ستونی شروع به رسم مستطیل انجام شود
- y1 : از چه سطری شروع به رسم مستطیل انجام شود
 - x2: نقطه پایانی یا طول مستطیل
 - y2 : نقطه پایانی یا عرض مستطیل
 - C: مقدار گرد بودن راس هر مستطیل



رسم اشکال هندسی در نمایشگر *TFT*

مثال نمونه از تصوير بالا:





```
#include <Ucglib.h>
Ucglib_ILI9341_18x240x320_HWSPI ucg(9, 10, 8);
void setup() {
    ucg.begin(UCG_FONT_MODE_SOLID);
    ucg.clearScreen(); ucg.setColor(1, 0, 0, 0);
    }
    void loop()
    {
        ucg.setColor(255, 0, 255);
        ucg.drawBox(20, 20, 40, 15);
        ucg.drawRbox(20, 45, 40, 15, 4);
        ucg.drawRBox(20, 70, 40, 15, 10);
    }
}
```

رسم شکل _ شبه مستطیل

دستور دیگه ای برای رسم مستطیل وجود دارد که شبیه به مستطیل بوده اما خود مستطیل نیست. رسم این مستطیل شبیه به مستطیلی که در قبل تر بیان شد هست اما شکل و رنگی که در نمایشگر رسم میشود کاملا متفاوت با حالت عادی هست. برای درک بهتر این مطلب به عکس زیر توجه کنید. برای رسم این شکل از دستور زیر استفاده میشود.

ucg.drawGradientBox(x0, y0, x1, y2);

- x1: از چه ستونی شروع به رسم مستطیل انجام شود
- y1 : از چه سطری شروع به رسم مستطیل انجام شود
 - x2: نقطه پایانی یا طول مستطیل
 - y2 : نقطه پایانی یا عرض مستطیل







رسم مستطیل در نمایشگر TFT

#include <Ucglib.h>
Ucglib_ILI9341_18x240x320_HWSPI ucg(9, 10, 8);
void setup()
{





```
ucg.begin(UCG_FONT_MODE_SOLID);
ucg.clearScreen();
ucg.setColor(1, 0, 0, 0);
ucg.setColor(255, 0, 255);
}
void loop() {
ucg.setColor(255, 0, 255);
ucg.drawBox(70, 10, 30, 60);
ucg.drawGradientBox(10, 10, 30, 60);
}
```

در کد بالا برای رسم مستطیل ها از رنگ بنفش با کمک دستور ucg.setColor(255, 0, 255) استفاده شده. مستطیل رسم شده جدید ظاهری 3 بعدی دارد.

رسم شکل _ مثلث

برای رسم مثلث از دستور زیر استفاده میشود.

ucg.drawTriangle(x0, y0, x1, y1, x2, y2);

- X0, Y0: مختصات راس اول مثلث
- X1, Y1: مختصات راس دوم مثلث
- X2, Y2: مختصات راس سوم مثلث

مثال

#include <Ucglib.h>
Ucglib_ILI9341_18x240x320_HWSPI ucg(9, 10, 8);

void setup()





```
{
    ucg.begin(UCG_FONT_MODE_SOLID);
    ucg.clearScreen();
    ucg.setColor(1, 0, 0, 0);
}
void loop() {
    ucg.setColor(255, 0, 255);
ucg.drawTriangle(60, 30, 10, 100, 110, 100);
```

}











رسم شکل _ چهار ضلعی

ucg.drawTetragon(x0, y0, x1,y1, x2, y2, x3,y3);

- X0, Y0: مختصات راس اول
- X1, Y1: مختصات راس دوم
- X2, Y2: مختصات راس سوم
- X3, Y3: مختصات راس چهارم

با استفاده از این تابع میتوانیم تمام اشکال 4 ضلعی را رسم کنیم مثل : مربع، مستطیل، ذوزنقه، لوزی و …. شاید سوال پیش آید که ترتیب راس ها به چه صورت هست. برای درک بهتر به تصویر زیر دقت نمایید.







رسم چهار ضلعی در نمایشگر *TFT*

مثال

#include <Ucglib.h>





```
Ucglib_ILI9341_18x240x320_HWSPI ucg(9, 10, 8);
void setup() {
    ucg.begin(UCG_FONT_MODE_SOLID);
    ucg.clearScreen();
    ucg.setColor(1, 0, 0, 0);
}
void loop() {
    ucg.setColor(255, 0, 255);
    ucg.drawTetragon(40, 30, 5, 100, 120, 100, 90, 30);
}
```

رسم شکل _ دایرہ

ucg.drawCircle(x0, y0, rad,option);

- x1: نقطه شروع رسم دایره
 - y1: نقطه پایان
 - rad: شعاع دایره
- Option: با تنظيم عدد بين 1 تا 15 قابليت رسم 15 حالت مختلف را دارد.







رسم دایره در نمایشگر *TFT*





مثال

```
#include <Ucglib.h>
Ucglib_ILI9341_18x240x320_HWSPI ucg(9, 10, 8);
void setup() {
  ucg.begin(UCG_FONT_MODE_SOLID);
  ucg.clearScreen();
  ucg.setColor(1, 0, 0, 0);
}
void loop() {
  ucg.setColor(255, 0, 255);
  ucg.setFont(ucg_font_ncenR08_hf);
  ucg.setColor(0, 20, 255, 20);
  ucg.setPrintPos(14, 20);
  ucg.print("in the name of GOD");
  ucg.drawDisc(20, 20, 10, 1);
  ucg.drawDisc(60, 20, 10, 2);
  ucg.drawDisc(100, 20, 10, 3);
  ucg.drawDisc(20, 50, 10, 4);
  ucg.drawDisc(60, 50, 10, 5);
  ucg.drawDisc(100, 50, 10, 6);
  ucg.drawDisc(20, 100, 10, 7);
  ucg.drawDisc(60, 100, 10, 8);
  ucg.drawDisc(100, 100, 10, 9);
```

}







رسم دایره در نمایشگر *TFT*

```
#include <Ucglib.h>
Ucglib_ILI9341_18x240x320_HWSPI ucg(9, 10, 8);
void setup() {
    ucg.begin(UCG_FONT_MODE_SOLID);
    ucg.clearScreen();
    ucg.setColor(1, 0, 0, 0);
}
void loop() {
    ucg.setColor(255, 0, 255);
```





```
ucg.drawDisc(20, 20, 12, 1);
ucg.drawDisc(60, 20, 12, 2);
ucg.drawDisc(100, 20, 12, 3);
ucg.drawDisc(20, 100, 12, 4);
ucg.drawDisc(60, 100, 12, 5);
ucg.drawDisc(100, 100, 12, 6);
}
```







رسم دایره در نمایشگر *TFT*

```
#include <Ucglib.h>
Ucglib_ILI9341_18x240x320_HWSPI ucg(9, 10, 8);
void setup() {
  ucg.begin(UCG_FONT_MODE_SOLID);
  ucg.clearScreen();
  ucg.setColor(1, 0, 0, 0);
```

}





```
void loop() {
   ucg.setColor(255, 0, 255);
   ucg.drawDisc(70, 30, 15, 7);
   ucg.drawDisc(40, 80, 15, 8);
   ucg.drawDisc(100, 100, 15, 9);
```

}



رسم دایره در نمایشگر *TFT*





```
#include <Ucglib.h>
Ucglib_ILI9341_18x240x320_HWSPI ucg(9, 10, 8);
void setup() {
  ucg.begin(UCG_FONT_MODE_SOLID);
 ucg.clearScreen();
  ucg.setColor(1, 0, 0, 0);
}
void loop() {
  ucg.setColor(255, 0, 255);
ک
  ucg.drawDisc(20, 20, 12, 10);
  ucg.drawDisc(60, 20, 12, 11);
  ucg.drawDisc(100, 20, 12, 12);
  ucg.drawDisc(20, 100, 12,13);
  ucg.drawDisc(60, 100, 12, 14);
  ucg.drawDisc(100, 100, 12, 15);
```

رسم شکل _ دایرہ تو خالی

ucg.drawCircle(x0, y0, rad, option);

}

تمامی آرگومان های ورودی این تابع مشابه رسم دایره است که پیشتر ذکر گردید.





آپشن های 1 تا 6



رسم دایره در نمایشگر *TFT*

```
#include <Ucglib.h>
Ucglib_ILI9341_18x240x320_HWSPI ucg(9, 10, 8);
void setup() {
    ucg.begin(UCG_FONT_MODE_SOLID);
    ucg.clearScreen();
    ucg.setColor(1, 0, 0, 0);
    }
    void loop() {
    ucg.setColor(255, 255, 0);
    ucg.drawCircle(20, 20, 12, 1);
}
```





ucg.drawCircle(60, 20, 12, 2); ucg.drawCircle(100, 20, 12, 3); ucg.drawCircle(20, 100, 12, 4); ucg.drawCircle(60, 100, 12, 5); ucg.drawCircle(100, 100, 12, 6); }

آپشن های 7 تا 9



رسم دایره در نمایشگر *TFT*





```
#include <Ucglib.h>
Ucglib_ILI9341_18x240x320_HWSPI ucg(9, 10, 8);
void setup() {
    ucg.begin(UCG_FONT_MODE_SOLID);
    ucg.clearScreen();
    ucg.setColor(1, 0, 0, 0);
}
void loop() {
    ucg.setColor(255, 255, 0);
    ucg.drawCircle(70, 30, 18, 7);
    ucg.drawCircle(40, 80, 18, 8);
    ucg.drawCircle(100, 100, 18, 9);
}
```

آپشن های 10 تا 15







رسم دایره در نمایشگر *TFT*

```
#include <Ucglib.h>
Ucglib_ILI9341_18x240x320_HWSPI ucg(9, 10, 8);
void setup() {
    ucg.begin(UCG_FONT_MODE_SOLID);
    ucg.clearScreen();
    ucg.setColor(1, 0, 0, 0);
    }
```





```
void loop() {
ucg.setColor(255, 255, 0);
ucg.drawCircle(20, 20, 12, 10);
ucg.drawCircle(60, 20, 12, 11);
ucg.drawCircle(100, 20, 12, 12);
ucg.drawCircle(20, 100, 12, 13);
ucg.drawCircle(60, 100, 12, 14);
ucg.drawCircle(100, 100, 12, 15);
}
```

خب حالا با چیز هایی که یاد گرفتیم بایید یه طرح خلاقانه درست کنیم و این آموزش رو به پایان برسونیم.



راه اندازی نمایشگر TFT مدل **1.8** اینچ





```
#include <Ucglib.h>
Ucglib_ILI9341_18x240x320_HWSPI ucg(9, 10, 8);
void setup()
{
  ucg.begin(UCG_FONT_MODE_SOLID);
  ucg.clearScreen();
  ucg.setColor(1, 0, 0, 0);
  ucg.setColor(0, 0, 255);
  ucg.drawCircle(35, 50, 20, 15);
  ucg.drawCircle(95, 50, 20, 15);
  ucg.setColor(255, 255, 255);
  ucg.drawDisc(35, 50, 5, 15);
  ucg.drawDisc(95, 50, 5, 15);
  ucg.setColor(0, 0, 255);
  ucg.drawVLine(65,50,90);
 ucg.drawDisc(65, 90, 30, 12);
 ucg.setColor(255, 255, 255);
  ucg.drawDisc(
    65, 95, 17, 12);
 ucg.setFont(ucg_font_ncenR08_hf);
  ucg.setColor(0, 255, 255, 255);
  ucg.setPrintPos(50, 15);
  ucg.print("Hello...");
}
void loop() {
```





}

نتيجه گيرى

در این مطلب بطور کامل و گام به گام نحوه راه اندازی نمایشگر TFT مدل **1.8** اینچ آموزش داده شد. با استفاده از کتابخانه معرفی شده و دستورات مربوط براحتی میتوان از این نمایشگر استفاده کرده و به هدف نهایی خود رسید.

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 پیچ مایکروالکام(microelecom) رو هم منشن کنید.