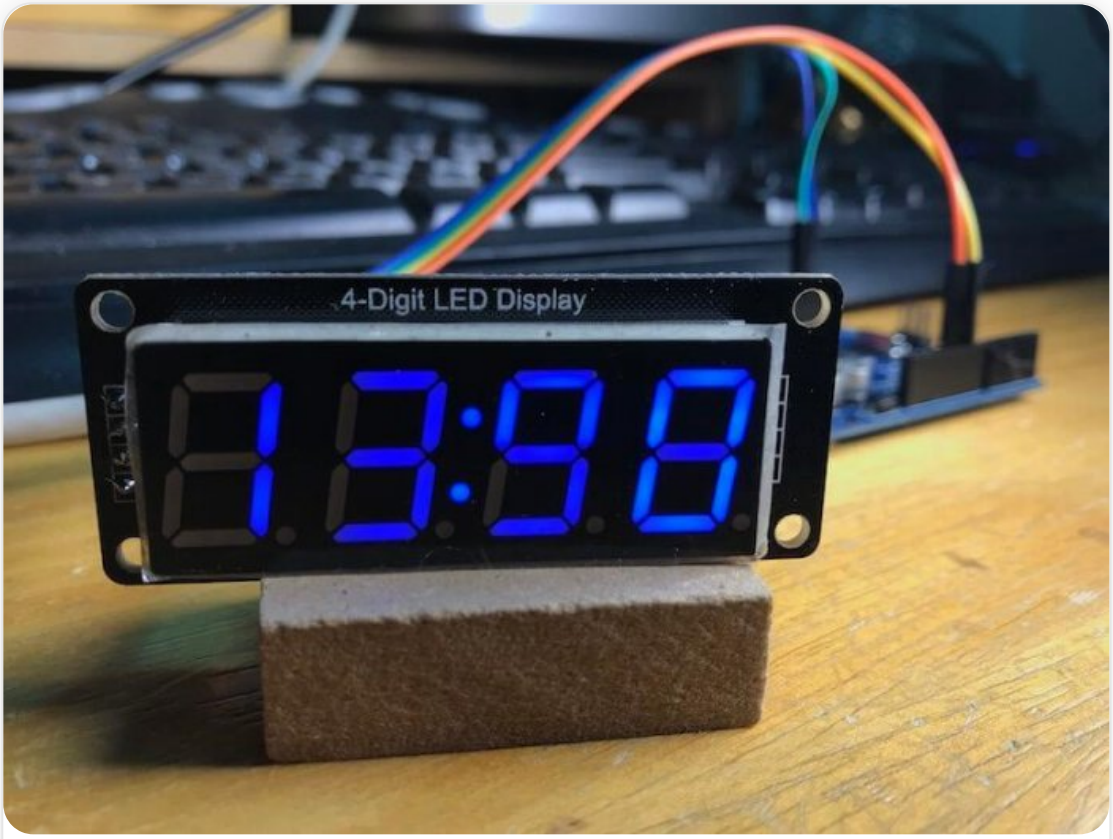


ARDUINO AVR STM32 بسکام کدویژن

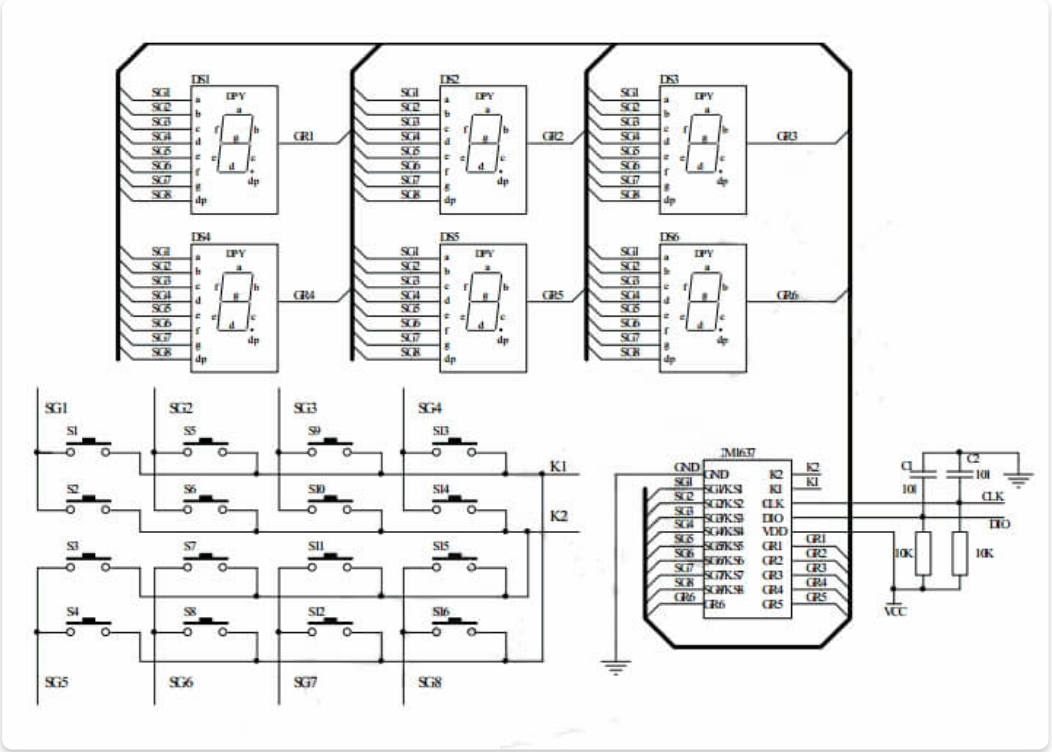
راه اندازی ماژول tm1637 با STM32 – کدویژن – بسکام – آردوینو

40 6,322

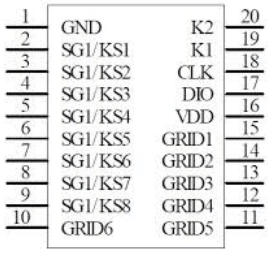
4.9/5 - (102 امتیاز)

آی سی tm1637 یک کنترل کننده یا راه انداز سون سگمنت و کیبورد است . آی سی tm1637 دارای قابلیت ها به شرح زیر می باشد :

- کنترل و راه اندازی 6 عدد سون سگمنت آند مشترک
- کنترل اتوماتیک روشنایی و نور سون سگمنت ها
- اسکن اتوماتیک سون سگمنتها به صورت مولتی پلکسر
- اسکن اتوماتیک 16 کلید به شکل کیبورد 2*8
- استفاده از پروتکل دو سیمه



آی سی tm1637 یک آی سی 20 پایه است ، که در بسته بندی های DIP و SOP موجود می باشد .

توصیف	پین	نام	برچسب	
داده ها و فرمان I / O داده با سطح HIGH پایه CLK ارسال می شوند . در هر 8 بیت آی سی یک ACK تولید می کند .	17	Data I/O	DIO	
فعال سازی با لبه بالارونده	18	ورودی کلاک	CLK	
ورودی برای صفحه کلید	19-20	اسکن صفحه کلید	K1-K2	
راه انداز بخش سگمنت ها (Open drain)	2-9	خروجی سگمنتها	SG1-SG8	
خروجی هر عدد سون سگمنت (Open drain)	10-15	خروجی هر سون سگمنت	GRID6-GRID1	
+	16	منبع تغذیه	VDD	
-	1	زمین	GND	

تغذیه آی سی tm1637 می تواند بین 3 تا 5 ولت باشد و جریان آن برای راه اندازی سگمنتها به صورت sink برابر 50 میلی آمپر می باشد . همچنین پایه های CLK , DIO چون open drain هستند ، توسط مقاومت با ظرفیت 10 کیلو اهم به تغذیه متصل می گردند و جهت جلوگیری از نویز و سیگنال ناخواسته توسط خازن با ظرفیت 100 پیکوفاراد به زمین متصل می شوند .

آی سی tm1637 با اینکه توسط پروتکل دوسیمه راه اندازی می شود ولی متاسفانه توسط پروتوکل استاندارد I2C پشتیبانی نمی شود . در پروتکل استاندارد I2C هر دیوایس که

توسط پروتکل I2C کار می کند یک آدرس مخصوص به خود را دارد که نهایتا تعداد 128 دیوایس می توانند بر روی یک باس I2C به هم دیگر وصل شوند و ارتباط برقرار کنند .

برای اینکه تولید کنندگان بتوانند برای دیوایس خود یک آدرس مخصوص به خود را داشته باشند باید مبلغی را به کنسرسیوم I2C پرداخت کنند ، ولی مشاهده می کنیم که سازنده آی سی tm1637 آدرس مخصوص به خود را ندارد ، بنابراین از آی سی tm1637 نمی توان در کنار دیگر دیوایس های بر روی باس I2C استفاده کرد ، و باید یک پروتکل مخصوص به خود آی سی tm1637 که در دیتا شیت آی سی tm1637 توضیح داده شده است ، استفده کنیم .

راه اندازی آی سی tm1637 با میکروکنترلر به صورت سریال و تنها به دو پایه جهت اتصال به پایه های DIO و CLK آی سی tm1637 نیاز می باشد . شروع ارتباط یا START زمانی آغاز می شود که خط Data از سطح HIGH به سطح LOW می رود ، در حالی که خط CLK در سطح HIGH قرار دارد . پایان ارتباط زمانی به اتمام می رسد که خط Data از سطح LOW به سطح HIGH می رود ، در حالی که خط CLK در سطح HIGH قرار دارد .

اگر دیتا بدرستی انتقال داده شود ، آی سی tm1637 با سیگنال ACK یا تایید ، پاسخ می دهد . به این شکل که در لبه پایین رونده از کلاک هشتم خط CLK ، خط Data به سطح LOW می رود . همچنین در لبه پایین رونده از کلاک نهم خط CLK ، ارتباط پایان می یابد و خط آزاد شده و آماده ارسال بایت بعدی می شود .

موضوعاتی که در این مقاله به آنها پرداخته خواهد شد :

ساختار رجیستر های Data در آی سی tm1637 :

آی سی tm1637 قادر است هم داده و هم دستور دریافت کند . دستورات به صورت مستقیم پردازش می شوند و اما داده ها برای ثبت شدن نیاز به رجیستر دارند . آی سی tm1637 قادر است سون سگمنت 6 رقمی را کنترل کند ، بنابراین برای هر رقم از سون سگمنت یک رجیستر وجود دارد .

آدرس رجیستر ها از C0 هگز شروع شده و تا C5 هگز می باشد . هر رجیستر برای نگهداشتن اطلاعات مربوط به یک رقم می باشد . هر رجیستر شامل 8 بیت است ، اگر هر بیت مقدار 1 باشد led آن سگمنت روشن و اگر بیت 0 باشد led مربوط به آن سگمنت خاموش می شود .

seg1	seg2	seg3	seg4	seg5	seg6	seg7	seg8	
b0	b1	b2	b3	b4	b5	b6	b7	
xxHL 4 bit bassi				xxHU 4 bit alti				
C0HL				C0HU				GRID1
C1HL				C1HU				GRID2
C2HL				C2HU				GRID3
C3HL				C3HU				GRID4
C4HL				C4HU				GRID5
C5HL				C5HU				GRID6

ساختار دستورات در آی سی tm1637 :

دستورات در آی سی tm1637 باید در اولین بایت بعد از لبه نزولی CLK در START شروع شوند. دستورات به صورت بایت و گروهی از 8 بیت هستند که بیت 6 و 7 نوع دستور را طبق جدول زیر نشان می دهند:

b7	b6	Comando
0	1	Commands related to data processing
1	0	Display control
1	1	Address setting

نوع دستور در آی سی tm1367

بنابراین طبق جدول فوق دستورها شامل سه دسته می شوند :

1. دستورات مربوط به پردازش داده (b7=0 , b6=1)
2. دستور مربوط به کنترل شدت روشنایی سون سگمنتها (b7=1 , b6=0)
3. دستور مربوط به آدرس دهی سون سگمنتها (b7=1 , b6=1)

1. دستورات پردازش داده

دستورات مربوط به پردازش داده ها عبارتند از:

b7	b6	b5	b4	b3	b2	b1	b0	Funzione	Descrizione
0	1	0	0			0	0	data read/write	Write to display
0	1					1	0		Read keys
0	1				0			Addressing	Auto increment
0	1				1				Fixed address
0	1			0				Test mode	Normal mode
0	1			1					Test mode

در جدول فوق بیت های b0 و b1 نمی توانند مقادیر 01 یا 11 را داشته باشند. بیت های b4 و b5 نیز با مقدار 0 تنظیم می شوند . اما دیتا شیت در مورد بیت هایی که در جدول فضاهای خالی دارند چیزی نگفته است . بنابراین تفاوتی ندارد که مقدار آنها را 0 یا 1 قرار دهیم . در هر صورت اگر فضاهای خالی را با 0 پر کنیم به این معنی است که دستور Normal mode با Write to display و Auto increment یکی است و عمل یکسانی را انجام می دهند .

بنابراین برای نوشتن داده بر روی سون سگمنتها حالت های زیر را می توانیم داشته باشیم :

- 01000000 **0x40** حالت Normal mode است و آدرس دهی به صورت اتوماتیک افزایش می یابد.
- 01000010 **0x42** حالت خواندن ورودی است و برای اسکن کیبورد استفاده می شود .
- 01000100 **0x44** حالت آدرس دهی ثابت یا fix را تنظیم می کند .
- 01001000 **0x48** حالت Test mode می باشد .

هیچ توضیحی در داخل دیتا شیت آی سی tm1367 در مورد Test mode داده نشده ، فقط اشاره شده برای استفاده داخلی می باشد .

دو حالت عملی (فرمان 0x40 و 0x44):

- **0x40** با آدرس مقصد ثابت: داده ها فقط به آدرس انتخاب شده ارسال می شوند .
- **0x44** با افزایش آدرس مقصد خودرو: هر یک از داده ها به صورت پیش فرض در آدرس های متوالی بعد از یکدیگر نوشته می شوند.

2 . دستور کنترل شدت روشنایی

این قسمت مربوط به وضعیت روشن / خاموش و تغییر روشنایی سون سگمنتها است که با PWM کنترل می شود.

- 10000000 **0x80** صفحه نمایش یا سون سگمنتها خاموش است .
- 10001000 **0x88** صفحه نمایش یا سون سگمنتها روشن است با PWM 1/16 .
- 10001001-100001111 **0x89-0x8F** شدت روشنایی در 7 سطح یا پله قابل کنترل است . مقدار 0X8F بیشترین روشنایی و مقدار 0X89 کمترین روشنایی را ایجاد می کند .

3 . دستور آدرس دهی سون سگمنتها

زمانی که تغذیه ماژول را وصل می کنیم و برای اولین بار روشن می شود ، به طور پیش فرض آدرس انتخاب شده خانه C0H می باشد ، یعنی اولین عدد در سمت چپ است . آخرین رقم از سون سگمنتها خانه C5H می باشد ، یعنی آخرین عدد از سمت چپ است . اگر مقدار بالاتر

از 5 رقم فرستاده شود ، آی سی tm1637 آن را نادیده گرفته و تا زمانی که یک آدرس معتبر ارسال نشود ، هیچ تاثیری نخواهد داشت . در حالت افزایش آدرس خودکار که دستور آن معادل 0x44 می باشد ، آدرس اولیه برای سون سگمنتها باید ارسال شود .

تا اینجای کار با دستورات و رجیسترهای آی سی tm1637 آشنا شدیم ، اکنون می خواهیم توسط یک مثال عملی با نحوه ارسال دستور و دیتا به آی سی tm1637 بهتر آشنا شویم . دیاگرام زیر نحوه ارسال دیتا به صورت آدرس دهی افزایشی یا Auto increment نشان می دهد .

1. شروع ارتباط یا START زمانی آغاز می شود که خط DIO از سطح HIGH به سطح LOW می رود ، در حالی که خط CLK در سطح HIGH قرار دارد .
2. مرحله بعد 8 بیت مربوط به فرمان یا Command1 ارسال می شود و باید با کلاک CLK همزمان باشد . مقدار فرکانس کلاک باید کمتر از 250 کیلو هرتز باشد . Command1 نوع آدرس دهی افزایشی (0x40) یا آدرس دهی ثابت (0x44) را مشخص می کند .
3. اگر دیتا بدرستی انتقال داده شود ، آی سی tm1637 با سیگنال ACK یا تایید ، پاسخ می دهد . به این شکل که در لبه پایین رونده از کلاک هشتم خط CLK ، خط DIO به سطح LOW می رود . همچنین در لبه پایین رونده از کلاک نهم خط CLK ، ارتباط STOP و پایان می یابد و خط آزاد شده و آماده ارسال بایت بعدی می شود .
4. ارسال بایت بعدی با START آغاز می شود و 8 بیت مربوط به Command2 ارسال می شود . Command2 مربوط به آدرس سون سگمنت است و مقدار آن می تواند 0XC0 الی 0XC5 باشد .
5. در مراحل بعد به صورت متوالی مقادیر Data1 الی DataN ارسال می شود . این دیتا ها بر روی سون سگمنتها به نمایش در می آیند .
6. بایت آخر یا Command3 مربوط به کنترل شدت روشنایی یا Brightness می باشد . مقدار آن از 0x80 الی 0x8F می باشد .

اگر ما آدرس دهی ثابت یا Fixed address را انتخاب کرده باشیم . نحوه ارسال دیتا به صورت دیاگرام زیر انتقال می یابد .

توسط روش آدرس دهی ثابت در آی سی tm1637 ما می توانیم به صورت انتخابی ارقام دلخواه را بر روی صفحه نمایش با مقدار جدید بروزرسانی کنیم و نیاز نیست همه ارقام ارسال شود .

نکته : اگر در هنگام انتقال یک دستور ، سیگنال STOP توسط میکروکنترلر ارسال شود، دستورالعمل و داده های ارسال شده قبل از آن اجرا شده و معتبر است . بنابراین ما می توانیم مقدار تنظیم brightness را یک بار در ابتدای برنامه انجام دهیم و دیگر نیاز نیست هر بار که مقداری را جهت نمایش ارسال می کنیم مقدار کنترل brightness را نیز ارسال کنیم .

خواندن کلید توسط tm1637

آی سی tm1637 قادر است دو بلوک 8 کلیدی را بخواند . طبق تصویر زیر پایه های - SG1 SG8 جهت اتصال به کلیدها و پایه های K1 , K2 جهت اتصال دو بلوک 8 تایی کلید به آی سی tm1637 استفاده می شوند .

آی سی tm1637 از وضعیت فشرده شدن همزمان بیش از یک کلید پشتیبانی نمی کند . ما می توانیم توسط دستور 0x42 درخواست خواندن کیبورد آی سی tm1637 را بدهیم . اگر هیچ کلیدی فشرده نشده باشد آی سی tm1637 مقدار 1111-1111 یا 0xff را بر می گرداند . اما اگر کلیدی فشرده شده باشد داده برگشتی مطابق جدول زیر خواهد بود . هنگام انتقال ابتدا bit0 یا کم ارزشترین بیت ارسال می شود .

مطابق جدول فوق 5 بیت اول یعنی b0 الی b4 از اهمیت برخوردار هستند . بیتهای b4 , b3 کلیدها را در خط مشترک K1 , K2 تشخیص می دهند و بیتهای b0 تا b2 کلید فشرده شده

را رمزگذاری می کنند . جدول زیر مقدار داده بازگشتی را با توجه به کلید فشار داده شده نشان می دهد .

بررسی ماژول tm1637

متاسفانه آی سی tm1637 در بازار ایران بسیار کمیاب است . اما نگران نباشید ، آی سی tm1637 را می توان در قالب ماژولی که به همراه یک عدد سون سگمنت 4 digit با جداکننده ساعت یا کولون(:) دار که مخصوص ساخت ساعت می باشد ، در بازار ایران با نام ماژول tm1637 براحتی تهیه کرد . سون سگمنت استفاده شده در ماژول tm1637 از نوع آند مشترک می باشد و نحوه اتصالات سون سگمنت به آی سی tm1637 در شماتیک زیر مشاهده می شود :

ماژول tm1637 دارای 4 پایه است ، که پایه CLK جهت ایجاد کلاک و پایه DIO جهت ارسال و دریافت دیتا به ماژول tm1637 و پایه های GND , VCC مربوط به تغذیه ماژول می باشد . تغذیه ماژول tm1637 بین 3 الی 5 ولت است ، اما توجه داشته باشید وقتی از سون سگمنت های آبی یا سبز رنگ استفاده می کنید ، تغذیه ماژول حتما باید 5 ولت باشد .

نکته بعد جداکننده ساعت یا کولون (:) می باشد . همانطور که در شماتیک مشخص است ، کولون به بیت هفتم از 2 digit متصل شده و با مقدار منطقی 0 خاموش و با مقدار منطقی 1 روشن می شود . نکته بعد در pcb ماژول tm1637 این است که پایه های مخصوص به اتصال کیبورد وجود ندارد و نمی توان در ماژول tm1637 از قابلیت خواندن کلید استفاده کرد .

راه اندازی ماژول tm1637 با آردوینو

برای راه اندازی ماژول tm1637 با آردوینو ابتدا نرم افزار آردوینو را اجرا کنید و از منوی Sketch گزینه Include Library و سپس Add .ZIP library را انتخاب کنید و فایل کتابخانه راه اندازی ماژول tm1637 که با نام TM1637.ZIP می باشد را به آردوینو اضافه نمائید .

کتابخانه راه اندازی ماژول tm1637 با آردوینو که در سایت قرار داده شده با بقیه کتابخانه های موجود در اینترنت متفاوت است و دارای امکانات بیشتری می باشد . در ادامه به بررسی و نحوه استفاده از توابع برای راه اندازی ماژول tm1637 با آردوینو می پردازیم . کتابخانه راه اندازی ماژول tm1637 با بورد آردوینو uno تست شده و توسط نرم افزار پروتئوس نیز شبیه سازی شده است .

ابتدا باید فایل کتابخانه در اول برنامه اضافه شود و پایه هایی که می خواهیم ماژول tm1637 را به آردوینو متصل کنیم را تعریف می کنیم .

```
#include <TM1637.h>

#define CLK 13//Pins for TM1637
#define DIO 12
TM1637 tm1637(CLK,DIO);
```

در قسمت setup برنامه آردوینو ، توسط دستور begin تنظیمات اولیه جهت راه اندازی ماژول tm1637 با آردوینو را اعمال می کنیم . در ابتدای آموزش گفتیم که شدت روشنایی یا brightness ماژول tm1637 دارای 7 سطح می باشد ، که سطح 7 حداکثر روشنایی را اعمال می کنید . با اجرای دستور begin به صورت پیشفرض حداکثر brightness یعنی مقدار 7 اعمال می شود .

همچنین توسط دستور tm1637.brightness می توانیم مقدار brightness را به صورت دلخواه تنظیم کنیم ، عدد 0 با عث خاموش شدن سون سگمنتها و عدد 1 الی 7 به ترتیب مقدار روشنایی را افزایش می دهد . اگر مقداری بیشتر از عدد 7 در تابع قرار داده شود ، حداکثر همان عدد 7 اعمال می شود . همچنین توسط دستور begin مقدار تاخیر برای اسکرول کردن متن بر روی ماژول tm1637 به صورت پیشفرض 300 میلی ثانیه در نظر گرفته می شود . می توان توسط فرمان tm1637.delayscroll مقدار تاخیر دلخواه را قرار داد .

```
void setup()
{
  tm1637.begin();
  tm1637.brightness(6); // 0~7 -> 0 = off -> 7 = max light
  tm1637.delayscroll(150);  // mili second  ,  defult = 300 ms
}
```

تابع بعدی جهت کنترل pointer یا : میان اعداد است ، که برای ساخت و نمایش ثانیه در ساخت ساعت دیجیتال می توانیم از آن استفاده کنیم . در صورت نوشتن عبارت POINT_ON یا عدد 1 درون تابع ، علامت : بر روی ماژول TM1637 روشن شده و با نوشتن عبارت POINT_OFF یا عدد 0 درون تابع ، علامت : بر روی ماژول tm1637 خاموش می شود .

```
tm1637.point(POINT_ON);
tm1637.point(POINT_OFF);
```

تابع بعدی برای راه اندازی ماژول tm1637 با آردوینو ، تابع tm1637.display می باشد . توسط این دستور ما می توانیم یک مقدار بر روی آدرس یا دیجیت دلخواه بنویسیم و همچنین می توانیم اعداد و آرایه و string بر روی ماژول tm1637 با آردوینو بنمایش درآوریم .

```
tm1637.display("duno");
tm1637.display(-123);
tm1637.display(9999);

int digitoneT = temp / 10;
int digittwoT = temp % 10;

tm1637.display(0,digitoneT);
tm1637.display(1,digittwoT);
tm1637.display(2,38); // put degree
tm1637.display(3,12);  // put a C at the end
```

تابع بعدی برای راه اندازی ماژول tm1637 با آردوینو ، می تواند رشته مورد نظر ما را بر روی سون سگمنت به حرکت درآورد یا اسکرول کند . این تابع زمانی مفید است که طول رشته ما بیشتر از 4 رقم باشد و ماژول tm1637 تنها 4 کاراکتر را می تواند نمایش بدهد . به عنوان مثال ما می خواهیم تاریخ و یا رشته ای را بر روی ماژول tm1637 نمایش دهیم از این تابع استفاده می کنیم .

در صورتی که از کاراکترها و حروفی که قابلیت نمایش بر روی سون سگمنت را ندارند استفاده کنیم ، بجای آن کاراکترها سون سگمنت خاموش شده و چیزی نمایش داده نمی شود .

```
void loop()
{
 tm1637.scroll("    ----HELLO----    Arduino  uno    ");
}
```

توسط کتابخانه راه اندازی ماژول tm1637 با آردوینو ، می توان اعداد و حروفی را که قابلیت ایجاد بر روی سون سگمنت را دارند به نمایش در آورد .تمامی اعداد و حروف قابل نمایش در

جدول زیر وجود دارد . همچنین می توان با ویرایش کتابخانه راه اندازی ماژول tm1637 اشکال یا حروف دیگری اضافه نمود .

```
// 0=0      A=10      L=20          _=30      h=40
// 1=1      b=11      n=21          ]=31      i=41
// 2=2      C=12      O=22          [=32
// 3=3      D=13      P=23          e=33
// 4=4      E=14      r=24          a=34
// 5=5      F=15      u=25          q=35
// 6=6      G=16      t=26          c=36
// 7=7      H=17      U=27          y=37
// 8=8      I=18    Off=28    degree=38
// 9=9      J=19      -=29          l=39
```

به همراه کتابخانه راه اندازی ماژول tm1637 با آردوینو چندین مثال وجود دارد که از منوی file و قسمت Examples قابل دسترسی می باشند .

در اولین مثال توسط یک سنسور دما و رطوبت dht22 یا dht11 مقادیر رطوبت و دما قرائت شده و بر روی ماژول tm1637 به نمایش در می آید . مطابق جدول فوق برای نشان دادن علامت درجه باید از عدد 38 دسیمال استفاده شود .

```
#include <TM1637.h>
#include <DHT.h>
#define DHTPIN 7
#define DHTTYPE DHT11   // DHT 11
//#define DHTTYPE DHT22   // DHT 22
DHT dht(DHTPIN, DHTTYPE);

#define CLK 13//Pins for TM1637
#define DIO 12
TM1637 tm1637(CLK,DIO);

void setup(){
  dht.begin();
  tm1637.begin();
  tm1637.delayscroll(150);  // mili second  ,  defult = 300 ms
  tm1637.scroll("    ----HELLO----    ");
}

void loop(){

   int humidity = dht.readHumidity();
```

```
    int temp = dht.readTemperature();

    int digitoneT = temp / 10;
    int digittwoT = temp % 10;
    int digitoneH = humidity / 10;
    int digittwoH = humidity % 10;

     tm1637.display(1,digitoneT);
     tm1637.display(2,digittwoT);
     tm1637.display(3,38); // put degree
     tm1637.display(4,12);  // put a C at the end

     delay (3000);

     tm1637.display(1,digitoneH);
     tm1637.display(2,digittwoH);
     tm1637.display(3,24);  // r
     tm1637.display(4,40);  // h

     delay(3000);
}
```

مثال بعدی برای راه اندازی ماژول tm1637 با آردوینو ، نمایش ساعت دیجیتالی بر روی ماژول tm1637 می باشد . در این مثال ابتدا باید کتابخانه TimerOne برای ایجاد زمان ساعت به آردوینو اضافه شود . توسط timer1 زمان نیم ثانیه را ایجاد می کنیم ، در هر نیم ثانیه یک بار علامت : خاموش و روشن می شود .

```
#include <TM1637.h>
#include <TimerOne.h>
#define ON 1
#define OFF 0

int8_t TimeDisp[] = {0x00,0x00,0x00,0x00};
unsigned char ClockPoint = 1;
unsigned char Update;
unsigned char halfsecond = 0;
unsigned char second;
unsigned char minute = 4;
unsigned char hour = 15;

#define CLK 13//pins definitions for TM1637 and can be changed
#define DIO 12
TM1637 tm1637(CLK,DIO);
```

```
void setup()
{
  tm1637.begin();
  Timer1.initialize(500000);//timing for 500ms
  Timer1.attachInterrupt(TimingISR);//declare the interrupt se
}
void loop()
{
  if(Update == ON)
  {
    TimeUpdate();
    tm1637.display(TimeDisp);
  }
}
void TimingISR()
{
  halfsecond ++;
  Update = ON;
  if(halfsecond == 2){
    second ++;
    if(second == 60)
    {
      minute ++;
      if(minute == 60)
      {
        hour ++;
        if(hour == 24)hour = 0;
        minute = 0;
      }
      second = 0;
    }
    halfsecond = 0;
  }
 // Serial.println(second);
  ClockPoint = (~ClockPoint) & 0x01;
}
void TimeUpdate(void)
{
  if(ClockPoint)tm1637.point(POINT_ON);
  else tm1637.point(POINT_OFF);
  TimeDisp[0] = hour / 10;
  TimeDisp[1] = hour % 10;
  TimeDisp[2] = minute / 10;
  TimeDisp[3] = minute % 10;
  Update = OFF;
}
```

مثال بعدی به حرکت درآوردن یک رشته یا اسکرول کردن متن بر روی ماژول tm1637 با آردوینو می باشد . مقدار تاخیر بین حرکت متن به صورت پیش فرض 300 میلی ثانیه می باشد . یعنی اگر از تابع delayscroll استفاده نکنیم ، مقدار تاخیر 300 میلی ثانیه خواهد بود .

```
#include "TM1637.h"

#define CLK 13//Pins for TM1637
#define DIO 12
TM1637 tm1637(CLK,DIO);

void setup(){
tm1637.begin();
}

void loop()
{
tm1637.delayscroll(250); //defult = 300 ms
tm1637.scroll("    ----HELLO----   Arduino  uno    ");
}
```

در مثال بعدی کدی نوشته شده است که مقدار یک عدد در حال افزایش را بر روی ماژول tm1637 به نمایش در می آورد . حداکثر عددی تا 9999 و -999 را می تواند نمایش دهد .

```
#include

#define CLK 13//Pins for TM1637
#define DIO 12
TM1637 tm1637(CLK,DIO);

void setup(){
tm1637.begin();
}

void loop(){
int numCounter = 0;
for(numCounter = 0; numCounter < 110; numCounter++)
{
tm1637.display(numCounter); //Display the numCounter value;
```

```
delay(300);
}
}
```

راه اندازی ماژول tm1637 با کدویژن

کتابخانه راه اندازی ماژول tm1637 با کدویژن دارای دو فایل tm1637.h و tm1637.lib می باشد . ابتدا باید فایل tm1637.h را در مسیری که نرم افزار کدویژن نصب شده است داخل پوشه INC و فایل tm1637.lib را داخل پوشه LIB انتقال دهید . چندین مثال متنوع برای آشنایی و کار با کتابخانه راه اندازی ماژول tm1637 با کدویژن آورده شده است .

ابتدای هر برنامه لازم است که کتابخانه ماژول tm1637 به برنامه اضافه و معرفی گردد . سپس باید پورت و شماره پایه هایی که ماژول tm1637 به میکرو متصل می شود را تعریف کنیم . همانطور که می دانید در نرم افزار کدویژن شماره پورت و پایه ها به زبان اسمبلی به برنامه معرفی می شوند .

```
#include <tm1637.h>

#asm
 .equ __tm1637_port=0X1b; //PORTA
 .equ __clk_bit=0;
 .equ __dio_bit=1;
#endasm
```

آدرس رجیستر پورتهای میکروکنترلر avr به شرح زیر است ، در این مثال توسط کد 0x1b پورت PORTA و برای clk=porta.0 و dio=porta.1 انتخاب می شوند .

```
PORTD=0x12;
PORTC=0x15;
PORTB=0x18;
PORTA=0x1b;
```

برای راه اندازی ماژول tm1637 با کدویژن چندین تابع وجود دارد که با نحوه استفاده از این توابع آشنا می شویم . اولین تابع که حتما باید در تابع main برنامه استفاده شود تابع tm1637_init می باشد ، این تابع ماژول tm1637 را پیکربندی و آماده کار می کند .

```
tm1637_init();
```

توسط تابع زیر می توانیم شدت روشنایی یا برایتنس سون سگمنتها را کنترل کنیم . ورودی این تابع عدد 0 الی 7 می پذیرد . عدد 0 باعث خاموش شدن سون سگمنتها می شود . عدد 1 الی 7 به ترتیب باعث افزایش برایتنس می شوند . اگر از این تابع در برنامه استفاده نشود ، ماژول tm1637 به صورت پیش فرض حداکثر برایتنس یعنی عدد 7 را دارد .

```
tm1637_brightness(7); // 0~7 -> 0 = off -> 7 = max light
```

تابع زیر زمانی که از تابع اسکرول استفاده می کنیم کاربرد دارد و مقدار تاخیر حرکت متن یا اسکرول شدن را بر حسب میلی ثانیه تایین می کند . اگر از این تابع استفاده نشود مقدار پیش فرض 300 میلی ثانیه می باشد .

```
tm1637_delayscroll(150);  // mili second  ,  defult = 300 ms
```

تابع زیر جهت نمایش چهار مقدار به ترتیب بر روی چهار سون سگمنت می باشد . این تابع تنها می تواند مقادیر داخل جدول را نمایش دهد . در مثال زیر ابتدا عدد 2 بعد 5 ، سپس علامت درجه و آخر حرف C را بر روی سون سگمنتهای ماژول tm1637 نمایش می دهد و به معنی دمای 25 درجه سانتیگراد می باشد .

```
tm1637_display_all(2,5,38,12);
```

تمامی اعداد و حروف قابل نمایش در جدول زیر وجود دارد . همچنین می توان با ویرایش کتابخانه راه اندازی ماژول tm1637 کاراکترها یا حروف دیگری اضافه نمود .

```
// 0=0      A=10      L=20        _=30      h=40
// 1=1      b=11      n=21        ]=31      i=41
// 2=2      C=12      O=22        [=32
// 3=3      D=13      P=23        e=33
// 4=4      E=14      r=24        a=34
// 5=5      F=15      u=25        q=35
// 6=6      G=16      t=26        c=36
// 7=7      H=17      U=27        y=37
// 8=8      I=18    Off=28    degree=38
// 9=9      J=19      -=29        l=39
```

تابع بعدی جهت نوشتن دیتا بر روی یکی از سون سگمنتها که در تابع انتخاب شده است می باشد .در این تابع آرگومان اول مقدار آدرس و آرگومان دوم مقدار دیتا را برای ماژول tm1637 مشخص می کند . در مثال زیر عدد 2 بر روی سون سگمنت اول و عدد 5 بر روی سون سگمنت دوم ، علامت درجه بر روی سون سگمنت سوم و حرف C بر روی سون سگمنت چهارم نمایش داده می شود . مقدار آدرس باید 1 الی 4 باشد اگر عددی غیر از آن وارد کنیم نادیده گرفته می شود .

```
//tm1637_display(address,data);

tm1637_display(1,2);
tm1637_display(2,5);
tm1637_display(3,38);
tm1637_display(4,12);
```

دو تابع بعدی برای نمایش یک رشته با طول چهار کاراکتر بر روی ماژول tm1637 می باشد . اگر طول رشته بیشتر از چهار کاراکتر باشد ، فقط چهار کاراکتر اول نمایش داده می شود و بقیه کاراکترها نادیده گرفته می شوند . تابع tm1637_putsf یک رشته که بر روی حافظه flash ذخیره شده نمایش می دهد و tm1637_puts یک رشته که بر روی حافظه ram ذخیره شده نمایش می دهد . مثال زیر نحوه استفاده از توابع را نشان می دهد .

```
flash unsigned char s1[]={"duno"};
unsigned char s2[]={"duno"};
tm1637_putsf(s1);
tm1637_puts(s2);

tm1637_putsf("duno");
tm1637_puts("duno");
```

تابع بعدی برای راه اندازی ماژول tm1637 با کدویژن ، می تواند رشته با طول بیشتر از چهار کاراکتر را بر روی سون سگمنتها اسکرول کند . به عنوان مثال ما می خواهیم تاریخ و یا رشته ای را بر روی ماژول tm1637 نمایش دهیم از این تابع استفاده می کنیم . در صورتی که از کاراکترهایی که داخل جدول کاراکترها وجود ندارد استفاده شود ، بجای آن کاراکترها سون سگمنت خاموش شده و چیزی نمایش داده نمی شود .

```
tm1637_scroll("    ----HELLO----     ");
```

آخرین تابع برای کنترل pointer یا : است ، که برای نمایش ثانیه ساعت دیجیتال می توانیم از آن استفاده کنیم . در صورت نوشتن عبارت POINT_ON یا عدد 1 درون تابع ، علامت : بر روی ماژول tm1637 روشن شده و با نوشتن عبارت POINT_OFF یا عدد 0 درون تابع ، علامت : بر روی ماژول tm1637 خاموش می شود .

```
tm1637_point(POINT_ON);
tm1637_point(POINT_OFF);
```

تا اینجای کار که با توابع آشنا شدیم با انجام چند مثال عملی به طور کامل با راه اندازی ماژول tm1637 با کدویژن آشنا می شویم . در اولین مثال می خواهیم مقادیر رطوبت و دما را از طریق سنسور dht22 قرائت و بر روی ماژول tm1637 با کدویژن نمایش دهیم . شما می توانید آموزش کامل راه اندازی سنسور dht22 را در سایت مطالعه کنید .

```
#include <mega16.h>
#include <stdio.h>
#include <math.h>
#include <tm1637.h>
#include "DHT22.h"

#asm
 .equ __tm1637_port=0X1b;
 .equ __clk_bit=0;
 .equ __dio_bit=1;
#endasm

unsigned char str[]={"    ----HELLO-----    "};

void main(void)
{
float temperature,humidity;
char t,h;
char dig1T,dig2T,dig1H,dig2H;
tm1637_init();
tm1637_scroll(str);
while (1)
      {
      if(dht22_read(&temperature,&humidity) == 0)
        {
           tm1637_scroll("    Error  dht22    ");
        }
      else
        {
```

```
        t=ceil(temperature);
        h=ceil(humidity);
        dig1T = t / 10;
        dig2T = t % 10;
        dig1H = h / 10;
        dig2H = h % 10;
        tm1637_display_all(dig1T,dig2T,38,36); // (degree = 38
        delay_ms(2500);
        tm1637_display_all(dig1H,dig2H,24,40); // ("r" = 24)
        delay_ms(2500);
        }
      }
}
```

در مثال بعدی قصد داریم بوسیله ماژول tm1637 با کدویژن یک ساعت دیجیتال بسازیم . در این مثال تایمر دو میکروکنترلر avr در مد آسنکرون پیکربندی شده است بنابراین باید از کریستال ساعت ۳۲/۷۶۸KHZ بر روی پایه های TOCS1 TOCS2 جهت تامین کلاک تایمر استفاده شود . شما می توانید آموزش پیکر بندی تایمر دو در مد آسنکرون را در سایت و از اینجا مطالعه کنید .

```
#include <mega16.h>
#include <tm1637.h>

#asm
 .equ __tm1637_port=0X1b; //PORTA
 .equ __clk_bit=0;
 .equ __dio_bit=1;
#endasm

_Bool Update_Time=0;
_Bool ClockPoint=0;
signed char second=0,minute=0,hour=0;

void main(void)
{
ASSR=0x08;
TCCR2=0x05;
TCNT2=0x00;
OCR2=0x00;
TIMSK=0x40;
#asm("sei")

tm1637_init();
```

```
while (1)
      {
        if(Update_Time)
        {
        Update_Time =0;
            if(ClockPoint)
            {
            tm1637_point(POINT_ON);
            }else
            {
            tm1637_point(POINT_OFF);
            };
        tm1637_display_all(hour/10,hour%10,minute/10,minute%10
        }
      }
}
//************************************************************
interrupt [TIM2_OVF] void timer2_ovf_isr(void)
{
Update_Time =1;
ClockPoint =~ClockPoint;
second++;
    if (second > 59)
    {
    second=0;
    minute++;
            if (minute>59)
            {
            minute=0;
            hour++;
                    if(hour>23)
                    {
                    hour=0;
                    }
            }
    }
}
```

راه اندازی ماژول tm1637 با بسکام

کتابخانه راه اندازی ماژول tm1637 با بسکام دارای دو فایل با نام های tm1637_config.bas و tm1637_function.bas می باشد ، فایل های کتابخانه باید در

کنار برنامه اصلی قرار داده شوند . ابتدا باید فایل tm1637_config.bas را توسط نرم افزار بسکام یا notpad باز کنید و چند خط کد که زیر را تنظیم و پیکربندی کنید . ابتدا پایه هایی از میکروکنترلر avr را که می خواهید به ماژول tm1637 اتصال دهید را تعریف کنید .

سپس مقدار کنتراست یا شدت روشنایی سون سگمنتها را تنظیم می کنیم . مقدار کنتراست از 0 الی 7 می باشد ، که مقدار صفر باعث خاموش شدن و مقدار 7 حداکثر نور را تنظیم می کند .

گزینه بعدی میزان تاخیر اسکرول متن را تایین می کند که بر حسب میلی ثانیه می باشد ، به صورت پیش فرض این تاخیر 300 میلی ثانیه می باشد . اگر مقدار تاخیر اسکرول کم باشد ، حرکت متن بر روی سون سگمنتها آهسته و اگر مقدار تاخیر اسکرول بیشتر باشد سرعت حرکت متن بر روی سون سگمنتها بیشتر می باشد .

```
Clk Alias Porta.0
Dio Alias Porta.1

brightness alias 7 ' 0~7 >>  0=off >>  7=Max light
delay_scroll alias 300 ' ms
```

برای راه اندازی ماژول tm1637 با بسکام چندین تابع نوشته شده است . اولین تابع که حتما باید قبل از تابع های دیگر نوشته شود ، تابع tm1637_init می باشد ، این تابع ماژول tm1637 را پیکربندی و آماده کار می کند .

```
Tm1637_init
```

تابع بعد برای راه اندازی ماژول tm1637 با بسکام ، می تواند رشته با طول بیشتر از چهار کاراکتر را بر روی ماژول tm1637 به حرکت درآورد . به عنوان مثال ما می خواهیم تاریخ و یا متنی را بر روی ماژول tm1637 نمایش دهیم از این تابع استفاده می کنیم . در صورتی که از کاراکترهایی که داخل جدول کاراکترها وجود ندارد استفاده شود ، بجای آن کاراکترها سون سگمنت خاموش شده و چیزی نمایش داده نمی شود .

```
tm1637_scroll("----HELLO----")
```

تمامی اعداد و حروف قابل نمایش در جدول زیر وجود دارد . همچنین می توان با ویرایش کتابخانه راه اندازی ماژول tm1637 کاراکترها یا حروف دیگری را اضافه کنید .

```
// 0=0      A=10      L=20        _=30      h=40
// 1=1      b=11      n=21        ]=31      i=41
// 2=2      C=12      O=22        [=32
// 3=3      D=13      P=23        e=33
// 4=4      E=14      r=24        a=34
// 5=5      F=15      u=25        q=35
// 6=6      G=16      t=26        c=36
// 7=7      H=17      U=27        y=37
// 8=8      I=18    Off=28    degree=38
// 9=9      J=19      -=29        l=39
```

توسط تابع زیر می توانیم یک رشته چهر کاراکتری را بر روی ماژول tm1637 نمایش دهیم . اگر طول رشته بیشتر از چهار کاراکتر باشد ، فقط چهار کاراکتر اول بر روی ماژول tm1637 نمایش داده می شود و بقیه رشته نادیده گرفته می شود .

```
Tm1637_print("duno")
```

تابع زیر برای نمایش یک عدد دسیمال می باشد . زمانی که می خواهیم یک شمارنده یا کرنومتر بسازیم . این تابع می تواند مفید باشد . حداکثر عددی که می توان نمایش داد 9999 می باشد .

```
dim n as word
for n = 1 to 9999
call Tm1637_num(n)
next n
```

تابع زیر جهت نمایش چهار مقدار به ترتیب بر روی چهار سون سگمنت ماژول tm1637 می باشد . این تابع تنها می تواند مقادیر داخل جدول را نمایش دهد . در مثال زیر ابتدا عدد 4 بعد عدد 1 ، سپس کاراکتر r و کاراکتر h را بر روی ماژول tm1637 نمایش می دهد و به معنی 41rh درصد رطوبت می باشد . آخرین مقدار تابع برای خاموش و روشن کردن پوینتر یا : بر روی ماژول tm1637 می باشد ، اگر 0 باشد خاموش و اگر 1 باشد علامت : روشن خواهد شد .

```
call disp_1_2_3_4_dot(4,1,24,40,0)
```

تابع بعدی جهت نمایش تنها یک عدد یا کاراکتر می باشد . بر روی یکی از چهار سون سگمنتی که ما انتخاب می کنیم . آرگومان اول تابع مقدار آدرس و آرگومان دوم مقدار دیتا می باشد . مقدار آدرس عددی بین 1 تا 4 باید باشد . اگر عددی غیر از این اعداد استفاده شود نادیده گرفته می شود . در مثال زیر عدد 2 بر روی سون سگمنت اول و عدد 5 بر روی سون سگمنت دوم ، علامت درجه بر روی سون سگمنت سوم و حرف C بر روی سون سگمنت چهارم نمایش داده می شود .

```
disp(1,2);
disp(2,5);
disp(3,38);
disp(4,12);
```

اکنون با توابع آشنا بالا می خواهیم با انجام چند مثال عملی به طور کامل با راه اندازی ماژول tm1637 با بسکام آشنا شویم . در اولین مثال می خواهیم مقادیر رطوبت و دما را از طریق سنسور dht22 یا dht11 بخوانیم و بر روی ماژول tm1637 با بسکام نمایش دهیم . شما می توانید آموزش کامل راه اندازی سنسور dht22 و dht11 را در یکی از پست های سایت مطالعه کنید .

```
$regfile = "m16def.dat"
$crystal = 8000000
$hwstack = 80
$swstack = 100
$framesize = 100
$baud = 9600

$include "tm1637_config.bas"
declare function dht_read( Dht_HUM As Single , Dht_TEMP As Sin


Dht_put Alias PortB.0 : Set Dht_put                          'S
Dht_get Alias PinB.0
Dht_io_set Alias DdrB.0

Dim Temperature As String * 6 , Humidity As String * 5
dim temp As Single , hum As Single , b as Byte
dim t as Byte , h as Byte



Tm1637_init
Do
 b=dht_read(hum,temp)
```

```
 if b=0 then

 tm1637_scroll("    Error dht22    ")
 print "error"

 else

 h=fix(hum)
 t=fix(temp)


call disp_1_2_3_4_dot(h / 10 ,h mod 10 ,24,40,0)
wait 3
call disp_1_2_3_4_dot(t / 10 ,t mod 10 ,38,12,0)
wait 3

   Humidity = Fusing(HUM , "#.#") + "%"
   Temperature =Fusing( TEMP, "#.#") + "C"

   print "Temp=" ; Temperature;"    ";
   print "Hum=" ; Humidity
   print "type sensor : dht";str(b)

 end if

Loop
End

$include "tm1637_function.bas"
```

مثال بعد یک ساعت دیجیتال توسط ماژول tm1637 با بسکام می باشد .

```
$regfile = "m16def.dat"
$crystal = 8000000
$hwstack = 80
$swstack = 100
$framesize = 100
$include "tm1637_config.bas"

CONFIG CLOCK = SOFT , GOSUB = SECTIC
ENABLE INTERRUPTS
dim pointer as Boolean
tm1637_init
```

```
DO

LOOP
END

SECTIC:
toggle pointer
call disp_1_2_3_4_dot( _hour / 10 , _hour mod 10 , _min / 10 ,
RETURN

$include "tm1637_function.bas"
```

انتهای برنامه حتما باید فایل کتابخانه راه اندازی ماژول tm1637 با بسکام tm1637_function.bas اضافه شود .

راه اندازی ماژول tm1637 با stm32

کتابخانه راه اندازی ماژول tm1637 با STM32 شامل دو فایل TM1637.h و TM1637.c می باشد و همراه یک پروژه به همراه چندین مثال جهت نمایش ساعت ، دما و رطوبت و اسکرول متن می باشد .
کتابخانه TM1637 توسط توابع HAL نوشته شده است و سازگار با نرم افزار CubeMX می باشد .
مثال همراه پروژه بصورت صد در صد عملی بر روی بورد بلوپیل با چیپ STM32F103C8T6 تست شده است .
قابلیت تنظیم کتابخانه برای ماژولهای tm1637 که دارای چهار سون سگمنت هستند و کسانی که می خواهند خود آی سی tm1637 را برای راه اندازی شش عدد سون سگمنت استفاده کنند.

توابع کتابخانه TM1637 برای STM32 به شرح زیر می باشد :

```
#define TM1637_4DIGIT
//#define TM1637_6DIGIT


#define TM1637_DELAY_US             1

#define ADDR_AUTO  0x40   //40H address is automatically incre
#define ADDR_FIXED 0x44   // 44H fixed address mode


#define STARTADDR  0xc0
```

```
#define OFF  28        //digit off


enum COLON     // COLON = (:)
{
  POINT_OFF=0,
  POINT_ON=1
};

extern unsigned char _point;
extern unsigned int  _scrolldelay; //ms
extern unsigned char _brightness; //0 ~ 7 >>>  0=off  ---  7=m

extern const char Table_7s[42];

void tm1637_init(void);
#ifdef TM1637_4DIGIT
void tm1637_display_all(unsigned char d1,unsigned char d2,unsi
#endif
#ifdef TM1637_6DIGIT
void tm1637_display_all(unsigned char d1,unsigned char d2,unsi
#endif
void tm1637_display(unsigned char addr,unsigned char data);
void tm1637_point(unsigned char point);
void tm1637_brightness(unsigned char brightness);
void tm1637_num(unsigned int num);
void tm1637_puts(unsigned char *str);
void tm1637_putsf(unsigned char const *str);
void tm1637_scroll(char *str);
void tm1637_scrolldelay(unsigned int scrolldelay);
```

راه اندازی ماژول tm۱۶۳۷ ۰۰:۴۲

توضیحات

نویسنده : حسین غیاثوند

خرید و دانلود در لحظه :

- [x] Arduino_کتابخانه و مثال – ~~89,000 تومان~~ 69,000 تومان
- [ ] Bascom_کتابخانه و مثال – ~~89,000 تومان~~ 69,000 تومان
- [ ] CodeVision_کتابخانه و مثال – ~~89,000 تومان~~ 69,000 تومان
- [ ] STM32_KEIL_کتابخانه و مثال – ~~89,000 تومان~~ 69,000 تومان
- [ ] STM32CubeIDE_کتابخانه و مثال – ~~89,000 تومان~~ 69,000 تومان

اضافه کردن به سبد خرید

بعد از پرداخت ، لینک دانلود فایل برای شما نمایش داده می شود . همزمان لینک دانلود فایل به ایمیل شما ارسال می شود .

در صورت عدم دریافت لینک دانلود در قسمت Inbox ایمیل ، قسمت Spam ایمیل خود را بررسی کنید .

موبایل : 09120197955
واتساپ : wa.me/989120197955
آموزش خرید و دانلود فایل

برچسب ها
راه اندازی ماژول TM1637 با آردوینو
راه اندازی ماژول TM1637 با بسکام
راه اندازی ماژول TM1637 با کدویژن

40 دیدگاه

قدیمی ترین

امینی

2 سال پیش

باسلام میتوانید یک کتابخانه برای tm1623 بنویسید در صورت امکان هزینه و زمانشو برام ایمیل کنید لطفا

0 پاسخ

soheil

1 سال پیش

سلام چطور میتونم با استفاده از زبان سی و بدون اسمبلی این رو تعریف کنم؟ چون هر چی به سی مینویسم ارور میده؟

```
#asm
 .equ __tm1637_port=0X1b; //PORTA
 .equ __clk_bit=0;
 .equ __dio_bit=1;
#endasm
```

1 پاسخ

Admin نویسنده

پاسخ دادن به *soheil* 1 سال پیش

سلام
پورت و پایه های i2c ماژول به این روش (روش استاندارد کدویژن) باید معرفی شود .
چه خطایی دریافت می کنید؟ از چه میکرویی استفاده می کنید ؟
لطفا بفرمایید تا مشکل شما را رفع کنیم .

0 پاسخ

soheil

پاسخ دادن به *Admin* 1 سال پیش

ممنونم...از atmega32 .. از همین برنامه کیپد شما برای کدویژن استفاده کردم، منتها وقتی زیر برنامه کیپد رو فعال کردم توی اون حلقه مربوط به کیپید، در مورد تعریف dio رو خطا داد که نمیشناسه.

0 پاسخ

Admin نویسنده

پاسخ دادن به *soheil* 1 سال پیش

اگر می خواهید از کتابخانه کیپد و کتابخانه tm1637 همزمان در یک پروژه استفاده کنید . باید هر کدام را بر روی پورتی مجزا تعریف کنید . مثلاً کیپد بر روی پورت A و tm1637 را بر روی پورت B پیکربندی کنید . برای دستیابی به آدرس پورت‌ها فایل mega32.h را باز کنید و آدرس هر پورت را مشاهده کنید

1 پاسخ

soheil

پاسخ دادن به *Admin* 1 سال پیش

مگه کیپد و سگمنتها به tm1637 وصل نیستند و tm1637 از طریق پایه های DIO و CLK که در برنامه اتمگا تعریف کردیم استفاده نشده؟ سوالم این هستش که در سورسی که گذاشتید پایه های DIO و CLK را از طریق برنامه اسمبلی تعریف کردید و این پایه ها به tm1637 وصل میشن، و از آنطریق به سگنت و کی پد....اتصال مستقیم از کیپد مگه به میکرو داریم؟ خود ماژول tm1637

مگه برای درایو کردن کردن کیپد و سگمنت همزمان نیست؟

0 پاسخ

Admin نویسنده

پاسخ دادن به soheil 1 سال پیش

با سلام
ماژول tm1637 که تصویر آن در همین صفحه قرار داده شده است ، فقط برای درایو چهار عدد سون سگمنت طراحی شده است و پایه ای برای اتصال کیپد قرار داده نشده . من فکر کردم از کتابخانه کیپد که در این آدرس https://micronik.ir/scan-keypad4x4 وجود دارد استفاده نموده اید .
لطفا خطایی که دریافت می کنید را قرار دهید تا بررسی شود .

1 پاسخ

soheil

پاسخ دادن به Admin 1 سال پیش

من هیچ تغییری در برنامه شما ندادم و از همین برنامه ای که شما در زیر برنامه tm1637.lib گذاشته بودید استفاده کردم، فقط زیر روال مربوط به کیپد غیرفعال بود و من فعالش کردم، با فعال کردن و ران کردن برنامه این ارور رو داد:

Library error:
C:\Users\m\Desktop\tm1637.lib(256):
'undefined symbol 'dio

دقیقا همین اررو رومیده که براتون کپی پیست کردم.

0 پاسخ

Admin نویسنده

پاسخ دادن به soheil 1 سال پیش

تاکید می کنم توسط ماژول tm1637 چون پایه ای برای اتصال کیپد وجود ندارد ، درون کتابخانه ، تابع خواندن کیپد غیرفعال می باشد .

اگر از آی سی tm1637 استفاده می کنید و پایه های کیپد آی سی را در اختیار دارید . می توانید داخل تابع کیپد به جای عبارت dio پایه ای که به dio آی سی متصل شده را قرار دهید .
به عنوان مثال :
dio=1 برابر PORTA.1=1
clk=0 برابر PORTA.0=0

3 پاسخ

soheil

پاسخ دادن به Admin 1 سال پیش

متشکرم

1 پاسخ

Admin نویسنده

پاسخ دادن به soheil 1 سال پیش

موفق باشید

1 پاسخ

Sajjad Rafiei

1 سال پیش

سلام
آیا میشه از واحد سخت افزاری i2c که همون twi هست برای ماژول استفاده کرد یا حتما باید i2c نرم افزاری باشه؟
در ضمن
ممنون از لایبری خوبتون

1 پاسخ

Admin نویسنده

پاسخ دادن به Sajjad Rafiei 1 سال پیش

سلام
خیر ، نمی توان از پروتکل i2c استاندارد میکرو استفاده کرد ، علت این موضوع در همین آموزش توضیح داده شده ، لطفا مطالعه نمایید .

برای راه اندازی این آی سی باید از پروتکل خاصی که خود کارخانه سازنده ارایه کرده ، و ما در کتابخانه استفاده کرده ایم را بکار ببرید .

0 پاسخ

محمد

1 سال پیش

سلام برای مگا64 و پورت E چی میشه اون اولش برای تعریف پورت؟

0 پاسخ

Admin نویسنده

پاسخ دادن به محمد

1 سال پیش

سلام

برای بدست آوردن آدرس پورت مورد نظر از میکروکنترلر در نرم افزار کدویژن ، به روش زیر عمل کنید . برای هر میکروکنترلر یک هدر وجود دارد ، زمانی که پروژه را کامپایل می کنیم ، هدر میکروکنترلر در منوی سمت چپ به قسمت Headers اضافه می شود . به عنوان مثال برای میکروکنترلر atmega64 هدر mega64.h را باز کنید و آدرس پورتها را می توانید ملاحضه کنید . آدرس پورت E برای مگا64 برابر عدد 3 می باشد .

1 پاسخ

محمد

پاسخ دادن به Admin

1 سال پیش

ممنون

0 پاسخ

Admin نویسنده

پاسخ دادن به محمد

1 سال پیش

خواهش می کنم ، موفق باشید

0 پاسخ

علی

1 سال پیش

سلام. این tm1637 تا ۶ دیجیت رو پشتیبانی میکند، من در یکی از کارام هر ۶ دیجیت رو گذاشتم،آیا میشود با کتابخانه ی شما اون دوتا دیجیت دیگر هم عدد روش انداخت و راه اندازی نمود؟

آخرین ویرایش 1 سال پیش توسط Admin

0 پاسخ

Admin نویسنده

پاسخ دادن به علی

1 سال پیش

با سلام ، این آی سی قابلیت نمایش ۶ دیجیت را دارد ، با کمی تغییرات در برنامه می توان ۶ دیجیت را راه اندازی نمود .

0 پاسخ

امینی

1 سال پیش

سلام مهندس امکان داره این کتابخانه را برای راه اندازی tm1623 استفاده کرد

0 پاسخ

Admin نویسنده

پاسخ دادن به امینی

1 سال پیش

سلام

دیتاشیت tm1637 در همین آموزش بصورت کامل توضیح داده شده است .
شما باید دیتاشیت tm1623 را با tm1637م مقایسه نمایید .

0 پاسخ

امینی

11 ماه پیش

با سلام استاد عزیز من این کتابخانه رو برا کدویژن تهیه کردم چطور میتونم از همه قابلیت های tm 1637 در کدویژن استفاده کنم 2 تا سگمنت آخر و رو ای سی نمیشناسه
یعنی فقط سگمنت 1.2.3.4 رو تابع قبول میکنه

0 پاسخ

Admin نویسنده

پاسخ دادن به امینی 11 ماه پیش

با سلام
ماژول های موجود در بازار با ۴ سون سگمنت ارائه شده اند، به همین دلیل کتابخانه برای راه اندازی تا دیجیت نوشته شده است ، برای استفاده تا ۶ دیجیت باید کتابخانه را ویرایش کنید .

0 پاسخ

امینی

پاسخ دادن به Admin 11 ماه پیش

اگه امکان داره راهنمایی بفرمایید
به نظرم 4 تابع آخری که به اسمبلی هستش اونا ثابت باشن و روی توابع بالایی که مربوط به display هستش تغیرات و اعمال کنم

0 پاسخ

Admin نویسنده

پاسخ دادن به امینی 11 ماه پیش

سلام
بزودی کتابخانه را مجدد بازنویسی می کنم و برای نمایش ۶ دیجیت هم سورس کد را داخل سایت بارگزاری می کنم .

2 پاسخ

امینی

پاسخ دادن به Admin 11 ماه پیش

مرسی

0 پاسخ

motahhar

11 ماه پیش

سلام.من این ساعت رو با atmega8 راه اندازی کردم.
portc0=clk , portc1=dio و fuses=int.8 MHZ
کریستال مورد نظر را هم در محل خود جاگذاری کردم.اما پس از اتصال تغذیه ،فقط صفر نمایش داده میشه و تغییری هم رخ نمیده.(تنظیمات TIMER2 رو هم طبق راهنما انجام دادم).
لطفا راهنمایی کنید که چگونه مشکل رو برطرف کنم.ممنون

0 پاسخ

Admin نویسنده

پاسخ دادن به motahhar

11 ماه پیش

سلام
احتمال زیاد مشکل از سخت افزار شما می باشد ، وضعیت اتصال یک کریستال ساعت 32768 هرتز به پایه های TOSC2 , TOSC1 را بررسی کنید .

0 پاسخ

toto

10 ماه پیش

سلام
ممنون میشم بیشتر توضیح بدین کیپد تو کدویژن عمل نمیکنه فقط FF برگشت میده
پایه dio وclk هم طبق فرمایش ست شدن

0 پاسخ

بهروز

6 ماه پیش

سلام
برنامه شما رو آپلود کردم فقط صفر نشون میده
برنامه scroll رو

0 پاسخ

Admin نویسنده

6 ماه پیش

پاسخ دادن به بهروز

سلام ، ابتدا بفرمایید از کدامیک برنامه های کدویژن ، بسکام یا آردوینو استفاده می کنید؟ برنامه نمایش ساعت و دما درست کار می کند ؟ فقط برنامه اسکرول درست کار نمی کند ؟

0 پاسخ

بارگزاری دیدگاه های بیشتر