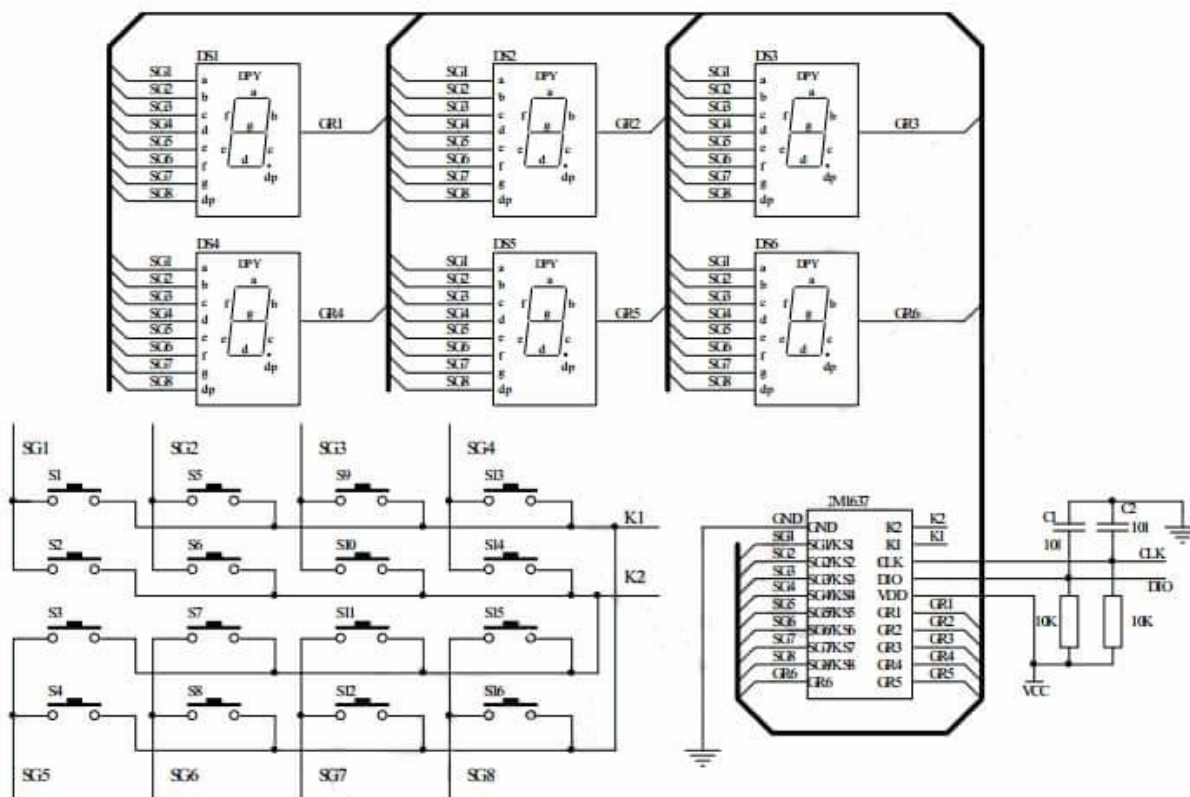


دارای قابلیت ها به شرح زیر **tm1637** یک کنترل کننده یا راه انداز سون سگمنت و کیبورد است . آی سی **tm1637** آی سی

: 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 8*2
- استفاده از پروتکل دو سیمه



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

برجسب	نام	بین	توصیف
	DIO	17	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 .
	CLK	18	ورودی کلاک فعال سازی با لبه بالارونده
	K1-K2	19-20	اسکن صفحه کلید
	SG1-SG8	2-9	خروجی سگمنتها
	GRID6-GRID1	10-15	خروجی هر عدد سون سگمنت (Open drain)
	VDD	16	منبع تغذیه +
	GND	1	زمین -

برابر 50 میلی sink می تواند بین 3 تا 5 ولت باشد و جریان آن برای راه اندازی سگمنتها به صورت **tm1637** تغذیه آی سی هستند ، توسط مقاومت با ظرفیت 10 کیلو اهم به تغذیه متصل **open drain** چون **CLK** , **DIO** آمپر می باشد . همچنین پایه های می گردند و جهت جلوگیری از نویز و سیگنال ناخواسته توسط خازن با ظرفیت 100 پیکوفاراد به زمین متصل می شوند . پشتیبانی نمی I2C با اینکه توسط پروتکل دوسیمه راه اندازی می شود ولی متاسفانه توسط پروتکل استاندارد **tm1637** آی سی کار می کند یک آدرس مخصوص به خود را دارد که نهایتا I2C هر دیوایس که توسط پروتکل I2C شود . در پروتکل استاندارد به هم دیگر وصل شوند و ارتباط برقرار کنند I2C تعداد 128 دیوایس می توانند بر روی یک باس I2C برای اینکه تولید کنندگان بتوانند برای دیوایس خود یک آدرس مخصوص به خود را داشته باشند باید مبلغی را به کنسرسیوم آدرس مخصوص به خود را ندارد ، بنابراین از آی **tm1637** پرداخت کنند ، ولی مشاهده می کنیم که سازنده آی سی استفاده کرد ، و باید یک پروتکل مخصوص به خود آی I2C نمی توان در کنار دیگر دیوایس های بر روی باس **tm1637** سی توضیح داده شده است ، استفاده کنیم **tm1637** که در دیتا شیت آی سی **tm1637** سی آی سی CLK و DIO با میکروکنترلر به صورت سریال و تنها به دو پایه جهت اتصال به پایه های **tm1637** راه اندازی آی سی می رود LOW به سطح HIGH از سطح Data زمانی آغاز می شود که خط START نیاز می باشد . شروع ارتباط یا **tm1637** به سطح LOW از سطح Data قرار دارد . پایان ارتباط زمانی به اتمام می رسد که خط HIGH در سطح CLK ، در حالی که خط HIGH قرار دارد HIGH در سطح CLK می رود ، در حالی که خط HIGH یا تایید ، پاسخ می دهد . به این شکل که در لبه پایین رونده ACK با سیگنال **tm1637** اگر دیتا بدرستی انتقال داده شود ، آی سی ، ارتباط پایان CLK می رود . همچنین در لبه پایین رونده از کلاک نهم خط LOW به سطح Data ، خط CLK از کلاک هشتم خط می یابد و خط آزاد شده و آماده ارسال بایت بعدی می شود .

: موضوعاتی که در این مقاله به آنها پرداخته خواهد شد

- **tm1637** در آی سی Data ساختار رجیستر های
- **tm1637** ساختار دستورات در آی سی
- دستورات پردازش داده 1.
- دستور کنترل شدت روشنایی 2.

- دستور آدرس دهی سون سگمنتها . 3
- tm1637 خواندن کلید توسط
- tm1637 بررسی مازول
- با آردوینو tm1637 راه اندازی مازول
- با کدویژن tm1637 راه اندازی مازول
- با بسکام tm1637 راه اندازی مازول

tm1637 در آی سی ساختار رجیستر های :

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

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

tm1637 ساختار دستورات در آی سی :

دستورات به صورت بایت . شروع شوند START در CLK باید در اولین بایت بعد از لبه نزولی **tm1637** دستورات در آی سی 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

tm1367 نوع دستور در آی سی

: بنابراین طبق جدول فوق دستور ها شامل سه دسته می شوند

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

دستورات پردازش داده 1.

: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

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

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

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

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

(0x44 و 0x40 فرمان 0): دو حالت عملی

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

دستور کنترل شدت روشنایی . 2

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

b7	b6	b5	b4	b3	b2	b1	b0	Function	Description
1	0	0 0			0	0	0	brightness	PWM 1/16
1	0				0	0	1		PWM 2/16
1	0				0	1	0		PWM 4/16
1	0				0	1	1		PWM 10/16
1	0				1	0	0		PWM 11/16
1	0				1	0	1		PWM 12/16
1	0				1	1	0		PWM 13/16
1	0				1	1	1		PWM 14/16
1	0			0				Display on/off	Display OFF
1	0			1					Display ON

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

دستور آدرس دهی سون سگمنتها . 3

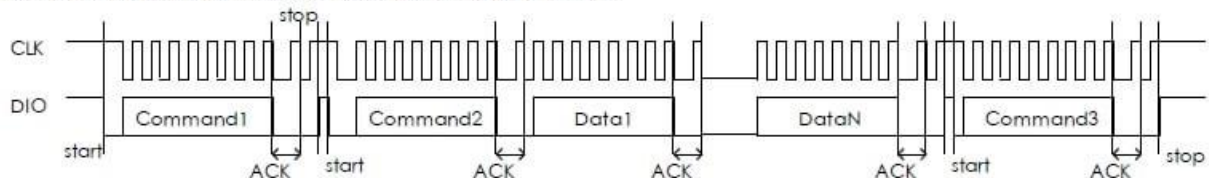
می **COH** زمانی که تغذیه مازول را وصل می کنیم و برای اولین بار روشن می شود ، به طور پیش فرض آدرس انتخاب شده خانه می باشد ، یعنی آخرین عدد از سمت چپ است **C5H** باشد ، یعنی اولین عدد در سمت چپ است . آخرین رقم از سون سگمنتها خانه

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

b7	b6	b5	b4	b3	b2	b1	b0	Indirizzo display
1	1	0 0		0	0	0	0	C0H
1	1			0	0	0	1	C1H
1	1			0	0	1	0	C2H
1	1			0	0	1	1	C3H
1	1			0	1	0	0	C4H
1	1			0	1	0	1	C5H

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

Write SRAM data in address auto increment 1 mode.

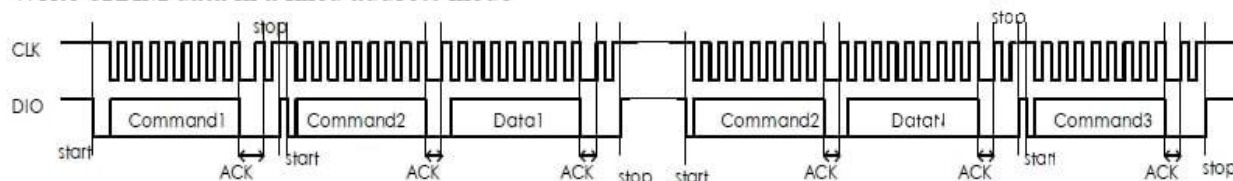


1. CLK می رود ، در حالی که خط LOW به سطح HIGH از سطح DIO زمانی آغاز می شود که خط **START** شروع ارتباط یا . قرار دارد HIGH در سطح .
2. همزمان باشد . مقدار فرکانس CLK ارسال می شود و باید با کلاک Command1 مرحله بعد 8 بیت مربوط به فرمان یا (0x44) یا آدرس دهی ثابت (0x40) نوع آدرس دهی افزایشی Command1 . کلاک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

- ارسال می شود . این دیتا ها بر روی سون سگمنتها به نمایش DataN الی Data1 در مراحل بعد به صورت متوالی مقادیر در می آیند .
- می 8F الی 80x می باشد . مقدار آن از 0 Brightness مربوط به کنترل شدت روشنایی یا Command3 بایت آخر یا باشد .

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

Write SRAM data in a fixed address mode

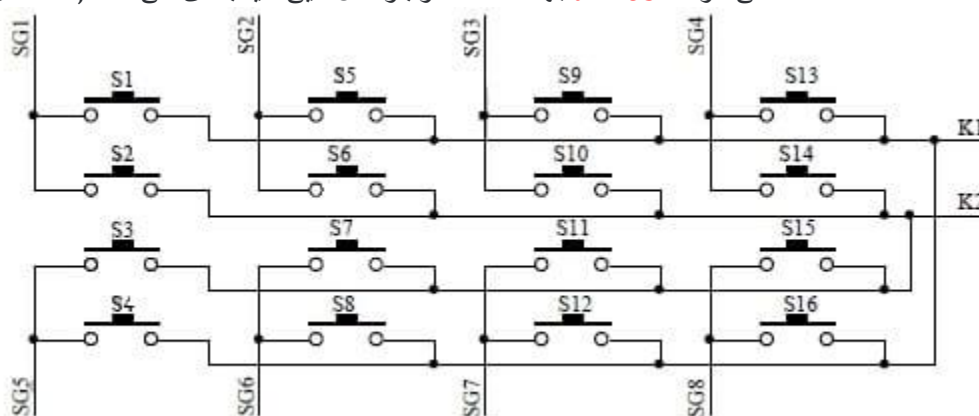


ما می توانیم به صورت انتخابی ارقام دلخواه را بر روی صفحه نمایش با **tm1637** توسط روش آدرس دهی ثابت در آی سی مقدار جدید بروزرسانی کنیم و نیاز نیست همه ارقام ارسال شود .

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

tm1637 خواندن کلید توسط

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



42x از وضعیت فشرده شدن همزمان بیش از یک کلید پشتیبانی نمی کند . ما می توانیم توسط دستور 0 **tm1637** آی سی مقدار 1111-1111 **tm1637** را بدهیم . اگر هیچ کلیدی فشرده نشده باشد آی سی **tm1637** درخواست خواندن کیبورد آی سی یا bit0 را بر می گرداند . اما اگر کلیدی فشرده شده باشد داده برگشتی مطابق جدول زیر خواهد بود . هنگام انتقال ابتدا 0xff یا 0 کم ارزشترین بیت ارسال می شود .

	SG1	SG2	SG3	SG4	SG5	SG6	SG7	SG8
K1	1110_1111	0110_1111	1010_1111	0010_1111	1100_1111	0100_1111	1000_1111	0000_1111
K2	1111_0111	0111_0111	1011_0111	0011_0111	1101_0111	0101_0111	1001_0111	0001_0111

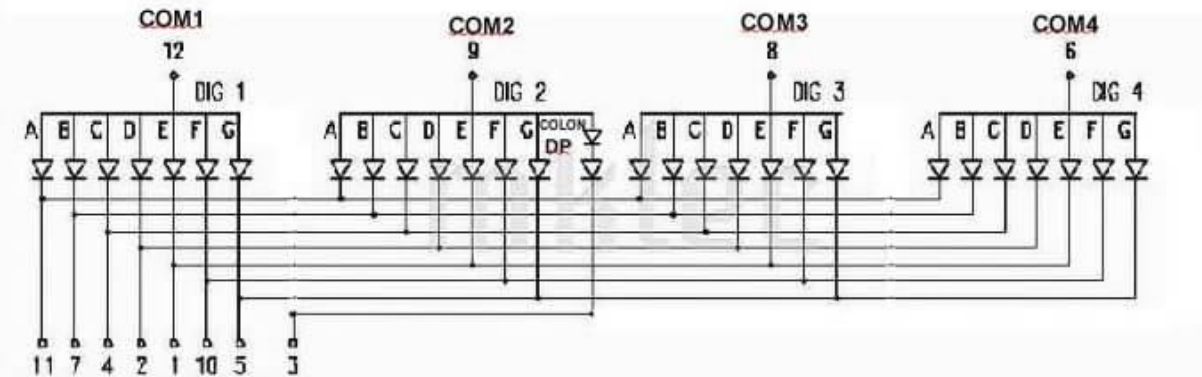
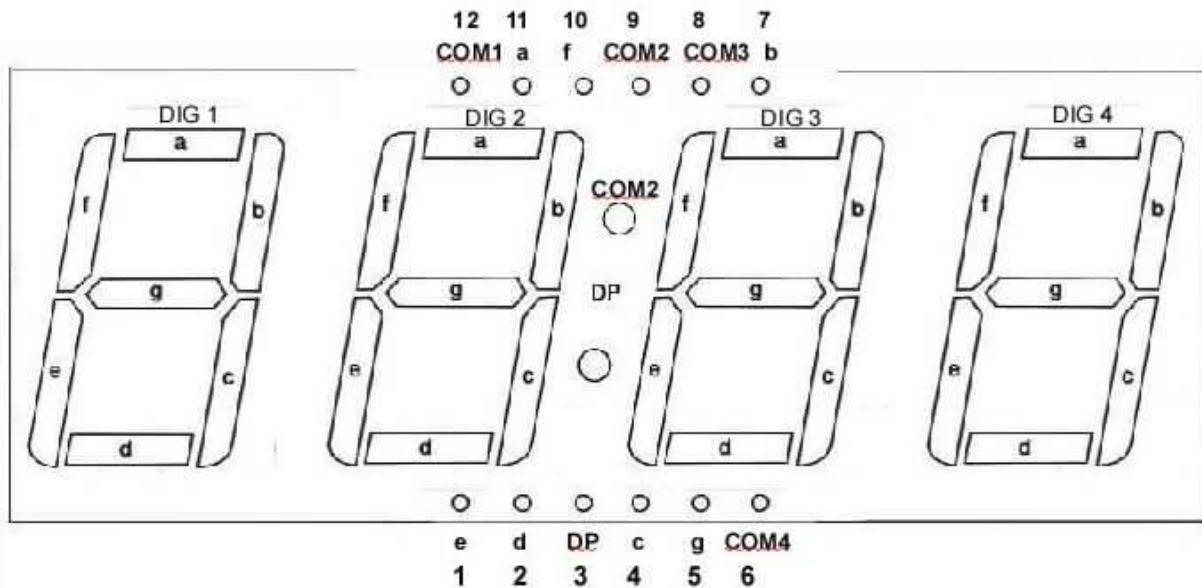
K1									K2								
b7	b6	b5	b4	b3	b2	b1	b0	Tasto	b7	b6	b5	b4	b3	b2	b1	b0	Tasto
1 1 1			1	0	1	1	1	S1	1 1 1			0	1	1	1	1	S2
			1	0	1	1	0	S5				0	1	1	1	0	S6
			1	0	1	0	1	S9				0	1	1	0	1	S10
			1	0	1	0	0	S13				0	1	1	0	0	S14
			1	0	0	1	1	S4				0	1	0	1	1	S3
			1	0	0	1	0	S8				0	1	0	1	0	S7
			1	0	0	0	1	S12				0	1	0	0	1	S11
			1	0	0	0	0	S16				0	1	0	0	0	S15

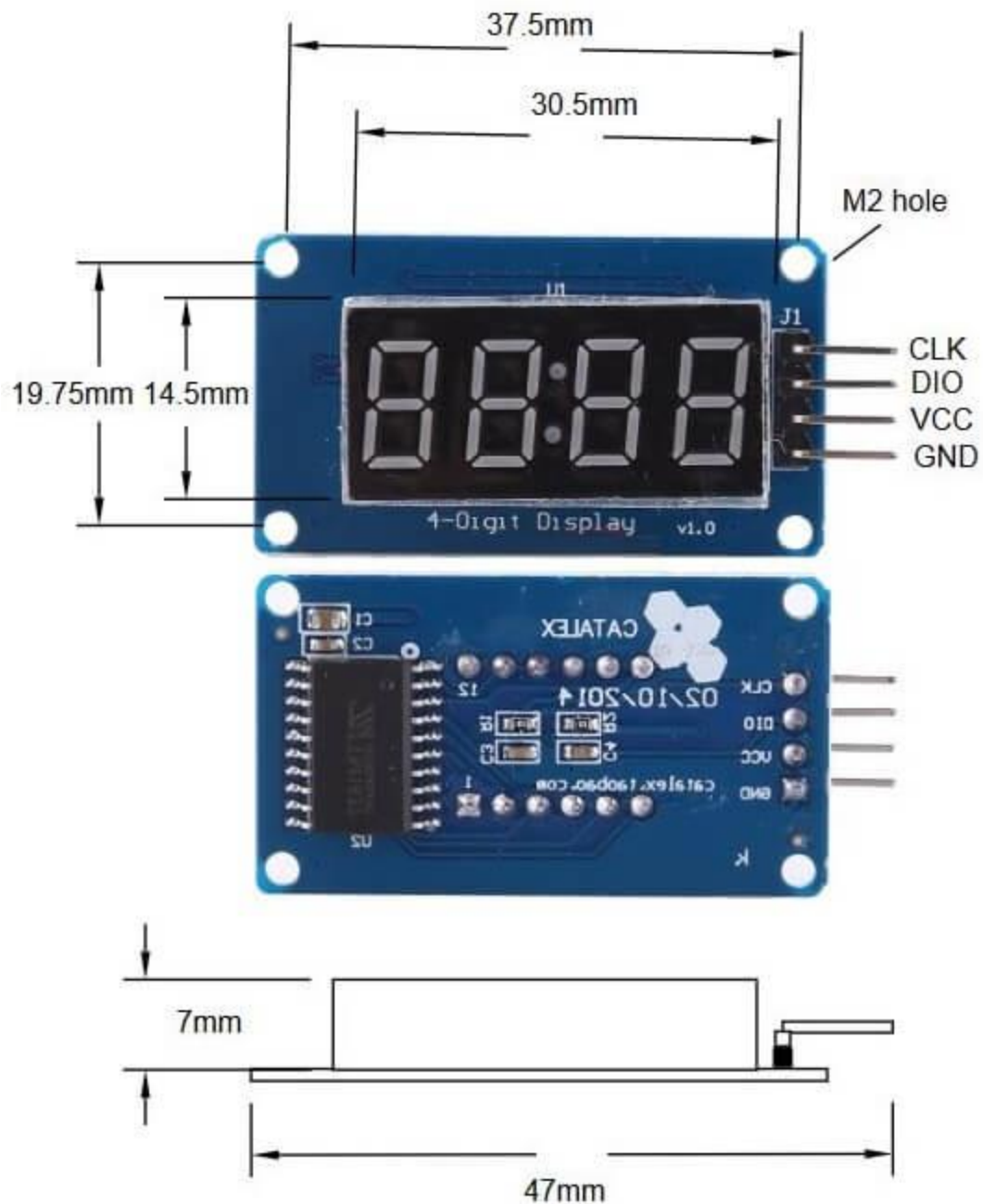
K1 , K2 کلیدها را در خط مشترک b4 , b3 از اهمیت برخوردار هستند . بیت‌های b4 الی b0 مطابق جدول فوق 5 بیت اول یعنی کلید فشرده شده را رمزگذاری می کنند . جدول زیر مقدار داده بازگشتی را با توجه به کلید b2 تا b0 تشخیص می دهند و بیت‌های فشار داده شده نشان می دهد .

S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	دکمه
SG1 K1	SG1 K2	SG5 K2	SG5 K1	SG2 K1	SG2 K2	SG6 K2	SG6 K1	SG3 K1	SG3 K2	SG7 K2	SG7 K1	SG4 K1	SG4 K2	SG8 K2	SG8 K1	ماتریس
F7	EF	EB	F3	F6	EE	EA	F2	F5	ED	E9	F1	F4	EC	E8	F0	هگزا

tm1637 بررسی ماژول

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



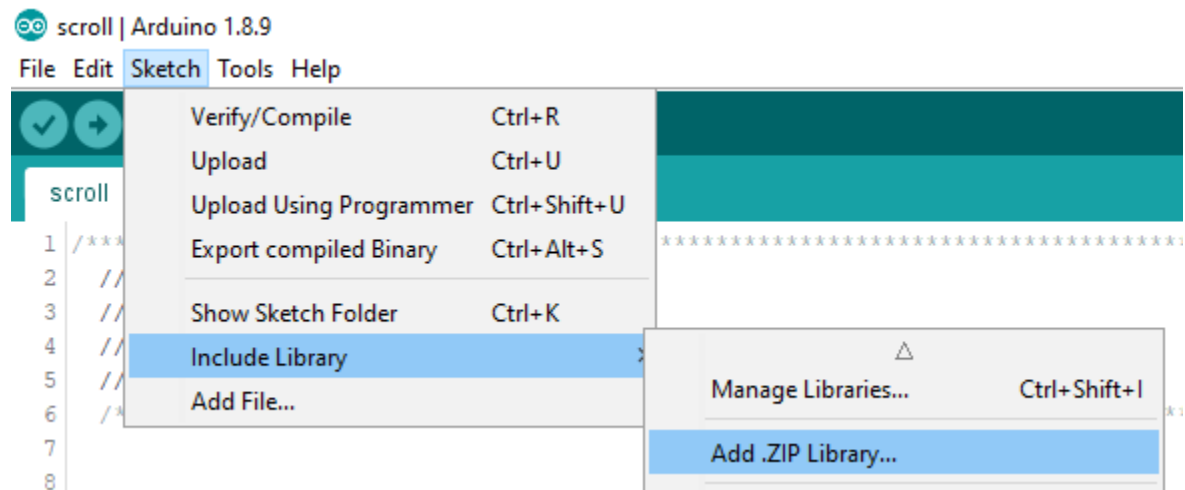


جهت ارسال و دریافت دیتا به DIO جهت ایجاد کلاک و پایه CLK دارای 4 پایه است ، که پایه **tm1637** ماژول بین 3 الی 5 ولت است ، **tm1637** مربوط به تغذیه ماژول می باشد . تغذیه ماژول VCC , GND و پایه های **tm1637** ماژول . اما توجه داشته باشید وقتی از سون سگمنت های آبی یا سبز رنگ استفاده می کنید ، تغذیه ماژول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

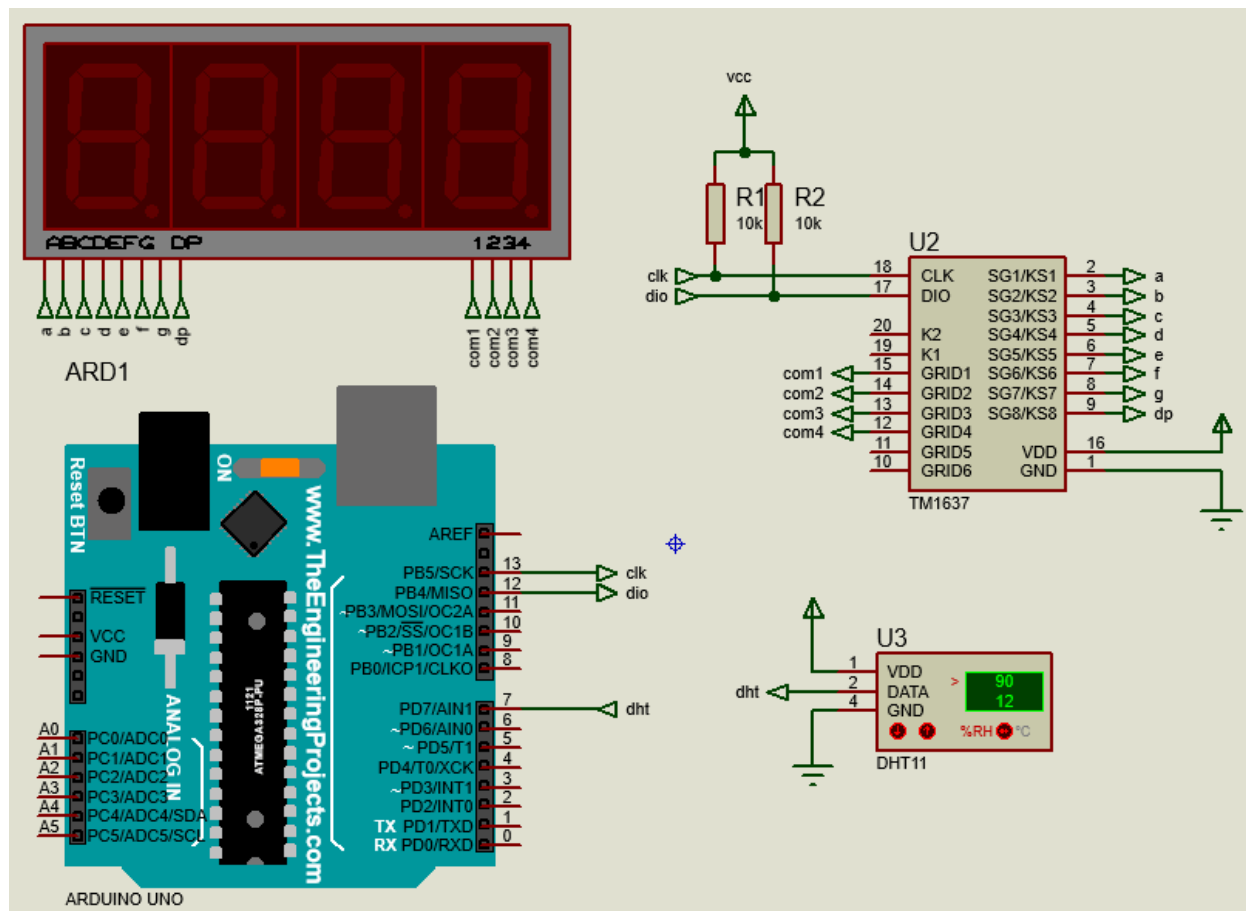
متصل digit 2 نکته بعد جداکننده ساعت یا کولون (:) می باشد . همانطور که در شماتیک مشخص است ، کولون به بیت هفتم از این است که پایه های **tm1637** ماژول pcb شده و با مقدار منطقی 0 خاموش و با مقدار منطقی 1 روشن می شود . نکته بعد در . از قابلیت خواندن کلید استفاده کرد **tm1637** مخصوص به اتصال کیبورد وجود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

با آردوینو **tm1637** راه اندازی ماژول

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



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



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

```
#include <TM1637.h>

#define CLK 13//Pins for TM1637

#define DIO 12

TM1637 tm1637 (CLK,DIO) ;
```

با آردوینو را اعمال **tm1637** تنظیمات اولیه جهت راه اندازی ماژول begin برنامه آردوینو ، توسط دستور setup در قسمت دارای 7 سطح می باشد ، که سطح 7 **tm1637** ماژول **brightness** می کنیم . در ابتدای آموزش گفتیم که شدت روشنایی یا

یعنی مقدار 7 اعمال **brightness** به صورت پیشفرض حداکثر **begin** حداکثر روشنایی را اعمال می کنید . با اجرای دستور می شود .

را به صورت دلخواه تنظیم کنیم ، عدد 0 با عث **brightness** می توانیم مقدار **tm1637.brightness** همچنین توسط دستور 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 مقدار تاخیر برای اسکرول کردن متن بر **begin** قرار داده شود ، حداکثر همان عدد 7 اعمال می شود . همچنین توسط دستور به صورت پیشفرض 300 میلی ثانیه در نظر گرفته می شود . می توان توسط فرمان **tm1637** روی مازول مقدار تاخیر دلخواه را قرار داد **tm1637.delayscroll** .

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

    tm1637.brightness(6); // 0~7 -> 0 = off -> 7 = max light

    tm1637.delayscroll(150); // mili second , default = 300 ms
}
```

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

```
tm1637.point(POINT_ON);

tm1637.point(POINT_OFF);
```

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

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

}
```

با آردوینو ، می توان اعداد و حروفی را که قابلیت ایجاد بر روی سون سگمنت را **tm1637** توسط کتابخانه راه اندازی ماژول دارند به نمایش در آورد . تمامی اعداد و حروف قابل نمایش در جدول زیر وجود دارد . همچنین می توان با ویرایش کتابخانه راه اشکال یا حروف دیگری اضافه نمود **tm1637** اندازی ماژول .

```
// 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
```

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

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

```
#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 , default = 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);

}

```

می باشد. در این **tm1637** با آردوینو، نمایش ساعت دیجیتالی بر روی ماژول **tm1637** مثال بعدی برای راه اندازی ماژول زمان نیم ثانیه را ایجاد می **timer1** برای ایجاد زمان ساعت به آردوینو اضافه شود. توسط **TimerOne** مثال ابتدا باید کتابخانه 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 to other ports

#define DIO 12

```

```
TM1637 tm1637(CLK,DIO);

void setup()

{

    tm1637.begin();

    Timer1.initialize(500000); //timing for 500ms

    Timer1.attachInterrupt(TimingISR); //declare the interrupt serve
routine:TimingISR

}

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;

}
```

با آردوینو می باشد . مقدار تاخیر بین **tm1637** مثال بعدی به حرکت در آوردن یک رشته یا اسکرول کردن متن بر روی مازول استفاده نکنیم ، مقدار تاخیر 300 **delayscroll** حرکت متن به صورت پیش فرض 300 میلی ثانیه می باشد . یعنی اگر از تابع 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); //default = 300 ms

tm1637.scroll("    ---HELLO---    Arduino uno    ");

}
```

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

```
#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) ;

}

}

```

با کدویژن tm1637 راه اندازی ماژول

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

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

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

```
PORTD=0x12;

PORTC=0x15;

PORTB=0x18;

PORTA=0x1b;
```

با کدویژن چندین تابع وجود دارد که با نحوه استفاده از این توابع آشنا می شویم . اولین تابع که **tm1637** برای راه اندازی ماژول را پیکربندی و آماده کار **tm1637** می باشد ، این تابع ماژول **tm1637_init** برنامه استفاده شود تابع **main** حتما باید در تابع 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 به صورت پیش فرض حداکثر برایتنس یعنی عدد 7 را دارد **tm1637** استفاده نشود ، ماژول

```
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 , default = 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 را بر روی سون سگمنتهای C نمایش دهد . در مثال زیر ابتدا عدد 2 بعد 5 ، سپس علامت درجه و آخر حرف . نمایش می دهد و به معنی دمای 25 درجه سانتیگراد می باشد **tm1637**ماژول

```
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 کاراکترها یا حروف دیگری اضافه نمود **tm1637**ماژول

// 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 مشخص می کند . در مثال زیر عدد 2 بر روی سون سگمنت اول و **tm1637**آدرس و آرگومان دوم مقدار دیتا را برای ماژول

بر روی سون سگمنت چهارم نمایش داده C عدد 5 بر روی سون سگمنت دوم ، علامت درجه بر روی سون سگمنت سوم و حرف 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);
```

می باشد . اگر طول رشته بیشتر از چهار **tm1637** دو تابع بعدی برای نمایش یک رشته با طول چهار کاراکتر بر روی ماژول یک **tm1637_putsf** کاراکتر باشد ، فقط چهار کاراکتر اول نمایش داده می شود و بقیه کاراکترها نادیده گرفته می شوند . تابع ذخیره شده ram یک رشته که بر روی حافظه **tm1637_puts** ذخیره شده نمایش می دهد و **flash** رشته که بر روی حافظه 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");
```

با کدویژن ، می تواند رشته با طول بیشتر از چهار کاراکتر را بر روی سون **tm1637** تابع بعدی برای راه اندازی ماژول نمایش دهیم از این تابع **tm1637** سگمنتها اسکرول کند . به عنوان مثال ما می خواهیم تاریخ و یا رشته ای را بر روی ماژول 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----    ");
```

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

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

با کدویژن آشنا می شویم tm1637 تا اینجای کار که با توابع آشنا شدیم با انجام چند مثال عملی به طور کامل با راه اندازی ماژول با کدویژن tm1637 قرائت و بر روی ماژول dht22. در اولین مثال می خواهیم مقادیر رطوبت و دما را از طریق سنسور . را در سایت مطالعه کنید dht22نمایش دهیم . شما می توانید آموزش کامل راه اندازی سنسور

```
#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) - ("c" =
36)

delay_ms(2500);

```

```

        tm1637_display_all(dig1H,dig2H,24,40); // ("r" = 24) - ("h" = 40)

        delay_ms(2500);

    }

}

}

```

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

```

#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;
            }
        }
    }
}

```

با بسکام tm1637 راه اندازی ماژول

با **بسکام** دارای دو فایل با نام **tm1637** کتابخانه راه اندازی ماژول می باشد ، فایل های کتابخانه باید در کنار برنامه اصلی قرار داده **tm1637_function.bas** و **tm1637_config.bas** های باز کنید و چند خط کد که زیر را تنظیم و notpad را توسط نرم افزار بسکام یا **tm1637_config.bas** شوند . ابتدا باید فایل اتصال دهید را تعریف کنید **tm1637** را که می خواهید به ماژول avr پیکربندی کنید . ابتدا پایه هایی از میکروکنترلر سپس مقدار کنتر است یا شدت روشنایی سون سگمنتها را تنظیم می کنیم . مقدار کنتر است 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
```

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

```
Tm1637_init
```

با بسکام ، می تواند رشته با طول بیشتر از چهار کاراکتر را بر روی **tm1637** تابع بعد برای راه اندازی ماژول نمایش دهیم از **tm1637** به حرکت درآورد . به عنوان مثال ما می خواهیم تاریخ و یا متنی را بر روی ماژول **tm1637** 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 کاراکترها یا حروف دیگری را اضافه کنید **tm1637**ماژول

```
// 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
```

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

```
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

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

```
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. بر روی سون سگمنت چهارم نمایش داده می شود C درجه بر روی سون سگمنت سوم و حرف

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

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


Dht_put Alias PortB.0 : Set Dht_put 'Sensor pins

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"

```

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

```
$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 , _min mod 10,pointer)
```

```
RETURN
```

```
$include "tm1637_function.bas"
```

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