BIG BANG
Scientific Series

انتشارات نبض

برنامه نویسی PLC
با زبان SCL
Structured Control Language

تالیف :
مهندس محمدرضا ماهر
مهندس منصور نعیمی

DVD شامل :
آخرین نسخه نرم افزارهای STEP7 و TIA
قابل نصب روی WIN7 (32 / 64 Bit)

بنام یکتا آفریدگار بی همتا

پیشگفتار مولفین

جای بسی خوشوقتی است که در طول چند سال گذشته شاهد آشنایی روز افزون کارشناسان و دانشجویان رشته های فنی بویژه رشته های برق و کامپیوتر با سیستم های کنترل بوده ایم ، به‌گونه ای که اکثر دانشجویان یا فارغ التحصیلان گرایش های مختلف براین رشته ها در جهت آشنایی بیشتر با سیستم های کنترل صنعتی ، بصورت خود جوش گام برداشته اند و آموزش های تخصصی نیز در این زمینه دیده اند .

افراد در طول این آموزش ها بیشتر از همه با زبان های سطح پایین برنامه نویسی LAD یا FBD کار کرده و تعدادی نیز به زبان STL پرداخته اند. متاسفانه زبان سطح بالای SCL در این میان گمنام مانده است و تعداد کمی از افراد با آن آشنا هستند . این در حالیست که زبان SCL به مراتب قویتر از زبان های سطح پایین است . بسیاری از برنامه هایی که با زبان های LAD/STL/FBD بسختی قابل نوشتن است با SCL بسادگی قابل پیاده سازی هستند و حتی سازنده سیستم کنترل نیز از زبان SCL برای طراحی فانکشن های نسبتا پیچیده ای که برای کتابخانه نرم افزار تهیه کرده استفاده کرده است.

با توجه به کمبودی که در این زمینه وجود داشت ، بر آن شدیم که کتابی جامع در زمینه برنامه نویسی با زبان SCL تهیه کنیم که هر دو نرم افزار STEP7 و TIA که برای سیستم کنترل های جدید است را پوشش دهد. سعی بر این بوده است که مطالب همراه با مثال های کاربردی زیاد عرضه شوند تا کاربر بتواند بخوبی از توانایی های این زبان استفاده نماید .

اگر چه در این کتاب قبل از شروع بحث ، بصورت خلاصه مطالبی در مورد نرم افزار و سخت افزار سیستم کنترل آورده شده است ولی باید اذعان کرد که افرادی می توانند از این کتاب بهره کامل ببرند که قبلا با PLC کار کرده و با زبان های سطح پایین ، برنامه نویسی انجام داده باشند. این افراد پس از مطالعه این کتاب قطعاً در خود توانایی زیادی در پیاده سازی انواع برنامه های PLC را حس خواهند کرد. خوشبختانه همه مثال ها و مطالب عرضه شده در کتاب توسط سیمولاتور قابل تست است و سخت افزار واقعی برای آزمایش برنامه مورد نیاز نخواهد بود.

در DVD همراه با کتاب آخرین نسخه نرم افزار های STEP7 و TIA وجود دارد که روی windows 7 نیز قابل نصب هستند و سیمولاتور نیز به همراه آنها نصب می گردد.

با امید به اینکه با ارائه این کتاب گامی در جهت ارتقا سطح دانش اتوماسیون صنعتی در کشور عزیزمان برداشته باشیم اعتراف می کنیم که این کتاب نیز مانند سایر مصنوعات دست بشر خالی از اشکال نیست . با ارائه‌ی نقطه نظرات خود به آدرس ایمیل reza.maher@hotmail.com یا mansoornaeimi@yahoo.com مارا در ارائه‌ی اثری کامل‌تر یاری فرمایید.

در پایان به تعداد معدودی از افراد فرصت طلب که به کپی برداری از دست نوشته های دیگران عادت کرده اند یاد آوری می کنیم که اجازه نقل قول و کپی برداری از هیچکدام از مثال ها و مطالب ارائه شده در این کتاب را عرفاً و قانوناً ندارند.

محمدرضا ماهر **منصور نعیمی**

پاییز ۱۳۹۲

فهرست

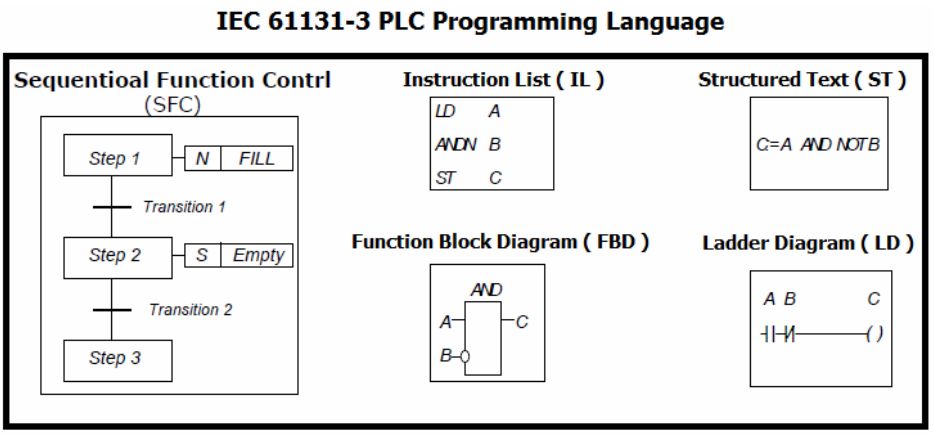
فصل ۱
مروری بر نکات PLC زیمنس

در این فصل نکات کلی مربوط به نرم افزار و سخت افزار PLC زیمنس یاد آوری و مرور می‌شود. افرادی که با این مفاهیم آشنا هستند می توانند می‌توانند مطالعه کتاب را از فصل ۲ شروع کنند.

۱-۱ مقدمه

زبان برنامه نویسی SCL نسبت به زبان های LAD/FBD/STL تا حد زیادی ناشناخته مانده است و بسیاری از برنامه نویسان PLC از این زبان و امکانات و توانایی های آن به اندازه کافی اطلاع ندارند. این در حالیست که وقتی فانکشن های موجود در کتابخانه های مختلف نرم افزار PLC را بررسی می‌کنیم، می‌بینیم که سازنده آن‌ها را با زبان SCL طراحی کرده است و در عین حال، توصیه‌ی سازنده به کاربران برای نوشتن فانکشن های پیچیده ، استفاده از همین زبان است.

SCL برگرفته از سه کلمه Structured Control Language بوده و زبان برنامه نویسی PLC های زیمنس است که بسیار شبیه پاسکال می‌باشد . طبق استاندارد IEC61131 که استاندارد PLC هاست ، سازندگان PLC بایستی زبان ساختار یافته موسوم به ST مخفف Structured Text را ارائه دهند که کاربر بتواند مانند زبان های برنامه نویسی ازجمله C و Pascal از دستورات سطح بالا مانند IF..THEN و حلقه FOR و امثال آن‌ها برای برنامه نویسی سیستم کنترل استفاده نماید.



شکل ۱-۱

برنامه نویسی Structured Text توسط سایر سازندگان PLC نیز ارائه شده است و مختص زیمنس نیست .به عنوان مثال ، شرکت Allen Bradley در نرم افزار RSLogix ،

فصل ۲

استفاده از SCL در STEP7

در این فصل ، ابتدا محیط نرم افزار SCL که همراه با STEP7 عرضه شده است ، تشریح شده و سپس چگونگی ایجاد بلوک های مختلف و برنامه نویسی آن‌ها با مثال های کاربردی به‌تفصیل شرح داده شده است.

۲-۷ مثال های کاربردی با SCL

مثال ۲-۵۷: کنترل دمای موتور

در یک پروسه یک سنسور PT100 برای اندازه گیری دمای یک موتور استفاده شده است. این سنسور به کارت های آنالوگ PLC متصل می باشد. هدف این است که موتور را در برابر گرمای بیش از حد حفاظت کنیم. برای این کار ، اگر دمای موتور به بالای ۴۰ درجه برسد ،باید فن خنک کننده موتور وارد مدار شود و اگر دما بیش از ۶۵ درجه شود ، چراغ خطر روشن شود و زمانی که دمای موتور به ۷۰ رسید ، موتور خاموش گردد.

جدول ۲-۲۲ متغیرهای مورد استفاده در این پروژه را نشان می‌دهد.

جدول ۲-۲۲

توضیح	آدرس	نام سمبلیک
موتور	Q0.0	Motor
سنسور دما	PIW256	Motor_Temp
چراغ خطر	Q0.1	Dang_Lamp
استارت موتور	I0.0	Start
فن خنک کننده	I0.1	Fan_Motor

حل:

ابتدا متغیرهای مورد نظر را در جدول سمبل ها تعریف می‌کنیم.

	Statu	Symbol	Address	Data type
1		Start	I 0.0	BOOL
2		Motor_Temp	PIW 256	INT
3		Motor	Q 0.0	BOOL
4		Fan_Motor	Q 0.1	BOOL
5		Dang_Lamp	Q 0.2	BOOL

شکل ۲-۲۰۸

سپس ، برنامه را به شکل زیر می نویسیم:

```
ORGANIZATION_BLOCK OB1
VAR_TEMP
    info : ARRAY[0..19] OF BYTE;
    Temp_Centigrade:INT;
END_VAR

    Temp_Centigrade :=  Motor_Temp/10; //1
    IF Start THEN                      //2
            Motor := True;             //3
        ELSIF Temp_Centigrade>40 THEN  //4
            Fan_Motor := True;         //5
        ELSIF Temp_Centigrade<=40 THEN //6
            Fan_Motor := False;        //7
    END_IF;                            //8
    IF Temp_Centigrade>65 THEN         //9
            Dang_Lamp :=  true;        //10
        ELSIF Temp_Centigrade<=65 THEN //11
            Dang_Lamp :=  False;       //12
        ELSIF Temp_Centigrade>70 THEN  //13
            Motor := False;            //14
   END_IF;                             //15

END_ORGANIZATION_BLOCK                 //16
```

شکل ۲-۲۰۹

تشریح برنامه:

بعد از تعریف OB1 ، متغیری به نام Temp_Centigrad را تعریف می‌کنیم تا دمای کالیبره شده را داخل آن بریزیم.

نکته: زمانی که دما از PT100 خوانده می‌شود ، برای کالیبره کردن کافی است تا عدد خوانده شده توسط PLC را بر ۱۰ تقسیم کنیم تا دمای واقعی به دست آید.

عمل کالیبراسیون در بند ۱ انجام شده است.

در سطر ۲ دستور IF را فراخوانی کرده و استارت شدن موتور را منوط به فشردن شستی I0.0 کرده ایم ، یعنی اگر I0.0 یک شد ، موتور روشن شود(سطر۳) .

در سطر ۴ گفته شده است که اگر دما بیشتر از ۴۰ درجه بود ، فن خنک کننده روشن شود(سطر۵) .

طبق سطرهای۶ و ۷ اگر دما کمتر از ۴۰ درجه شود ، فن خنک کننده خاموش می‌شود.

در سطر۹ تا ۱۲ دستورات کنترل مربوط به چراغ خطر نوشته شده است.

و درسطرهای ۱۳ و ۱۴ دستور مربوط به خاموش شدن موتور ، درصورت بالاتر رفتن دما از ۷۰ درجه ، نوشته شده است.

مثال ۲-۵۸ : کنترل دستگاه در مد اتومات و مد دستی

دستگاهی را می خواهیم هم در مد دستی و هم در مد اتوماتیک راه اندازی کنیم. مراحل کار این دستگاه در دو مد به شرح زیر می باشد:

مد دستی:

تا زمانی که اپراتور ورودی I0.0 را فعال نگه می‌دارد ، خروجی Q0.0 که مربوط به حرکت نوار نقاله است ، فعال شود و زمانی که اپراتور ورودی I0.1 را فعال نگه می‌دارد ، خروجی Q0.1 که مربوط به نازل آب برای پر کردن بطری ها است ، فعال شود. توجه داشته باشید که برنامه باید طوری نوشته شود که اپراتور مجبور باشد برای فعال کردن خروجی ها ، کلید های ورودی را نگه دارد و وقتی که دست خود را از روی کلید های ورودی برداشت ، خروجی غیر فعال شود.

مد اتومات:

زمانی که ورودی I0.0 فعال شود ، توسط اپراتور نوار نقاله شروع به حرکت می‌کند (خروجی Q0.0 فعال شود) و زمانی که بطری به زیر سنسور با آدرس I0.1 رسید ، ۵ ثانیه برای بارگیری توقف می‌کند. همزمان با توقف بطری خروجی Q0.1 که فرمان پرکردن بطری را به نازل آب می‌دهد باید فعال شود. این پروسه تا زمانی که اپراتور ورودی I0.2 مربوط به توقف پروسه را فعال کند ، ادامه دارد. توجه داشته باشیدکه برعکس حالت دستی ، برنامه طوری نوشته شود که وقتی که یک بار اپراتور فرمان استارت را داد ، پروسه شروع به استارت کند و برای توقف هم یک بار زدن کلید stop کافی باشد.

ورودی I0.2 برای تعیین مد دستی و اتوماتیک می باشد که اگر یک باشد ، مد دستی و اگر صفر باشد ، مد اتومات انتخاب می‌شود.

حل:

ابتدا به آدرس های داده شده اسامی سمبلیک اختصاص می‌دهیم.

	Statu	Symbol	Addres	Data type	Comment
1		M_start	I 0.0	BOOL	
2		N_start	I 0.1	BOOL	
3		Mode	I 0.2	BOOL	
4		Flag1	M 7.0	BOOL	
5		Flag2	M 7.1	BOOL	
6		Flag3	M 7.2	BOOL	
7		Motor	Q 0.0	BOOL	
8		Nazel	Q 0.1	BOOL	

شکل ۲-۲۱۰

برای سادگی ، برنامه را به سه قسمت تقسیم کرده ایم و هر قسمت را جداگانه توضیح داده ایم.

```
ORGANIZATION_BLOCK OB1
    VAR_TEMP
    info  : ARRAY[0..19] OF BYTE;        //1
    Timer1: S5TIME;                      //2
    Motor   : BOOL;                      //3
    Nazel : BOOL;                        //4
    M_Start   :BOOL;                     //5
    N_Start :BOOL;                       //6
    Mode : BOOL;                         //7
    i :INT;                              //8
    Flag1:BOOL;                          //9
    Flag2:BOOL;                          //10
    Flag3:BOOL;                          //11

    END_VAR                              //12
        IF Mode THEN                     //13
        i := 1;                          //14
        ELSE                             //15
        i :=0;                           //16
        END_IF;                          //17
        MW4:=INT_TO_WORD(i);             //18
    CASE i OF                            //19
        0 : IF M_Start THEN              //20
                Motor := true;           //21
                ELSE                     //22
                Motor := false;          //23
            END_IF;                      //24
```

شکل ۲-۲۱۱

همانطور که در بخش اول می بینید ، از سطر اول تا یازدهم جنس متغیرهایی که در جدول سمبل ها تعریف کرده ایم ، مشخص شده است. در سطر ۱۳ تا ۱۷ دستور IF آورده شده و وظیفه‌ی این بخش از برنامه تعیین این است که تعیین کند آیا مد دستی باشد ، یا مد اتوماتیک.

فصل ۳
استفاده از SCL در محیط TIA

در این فصل نحوه‌ی برنامه نویسی SCL در محیط TIA با ارائه‌ی مثال‌های کاربردی فراوان تشریح شده است.

۳-۱ مقدمه

در فصل قبل با نحوه‌ی استفاده از SCL و برنامه نویسی آن در محیط STEP7 V5.x آشنا شدیم . از آنجایی که سیستم های کنترل جدید زیمنس نظیر S7-1200 و S7-1500 را نمی‌توان توسط نرم افزار قبلی پیکر بندی و برنامه نویسی نمود وبرای این‌کار نیاز به نرم افزار TIA می‌باشد لازم دیدیم در این فصل خواننده را با نحوه استفاده از SCL در محیط TIA آشنا کنیم. البته با استفاده از TIA می‌توان کنترلر های S7-300 و S7-400 را نیز پیکر بندی و برنامه نویسی نمود.

پس از مطالعه این فصل خواهید دید که نحوه‌ی استفاده از ابزارهای SCL در TIA نسبت به نرم افزار قبلی کمی متفاوت و تاحدی نیز ساده تر است با این وجود اصول برنامه نویسی در هر دو نرم افزار یکسان می‌باشد.

آخرین نسخه این نرم افزار در DVD همراه کتاب موجود است و می‌توان آن را در کنار STEP7 V5.x نصب نمود و تداخلی با یکدیگر نخواهند داشت.

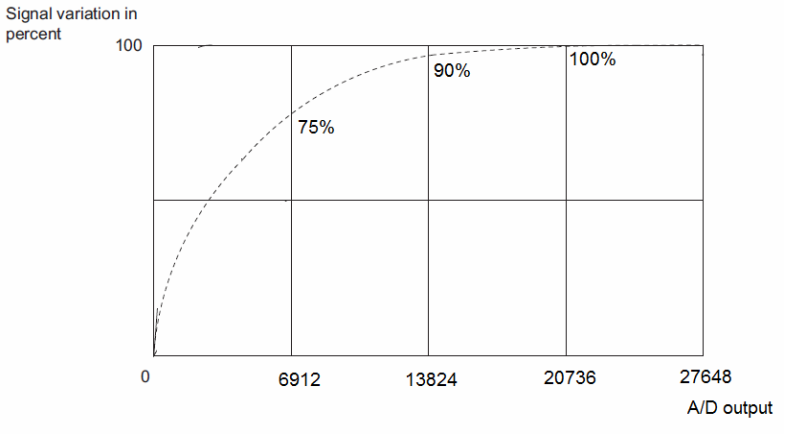
۳-۲ شروع کار در محیط TIA

با ایجاد یک پروژه‌ی جدید توسط TIA ، پنجره شکل ۳-۱ را داریم .

- در قسمت Configure a device پیکر بندی سخت افزار سیستم کنترل که می‌تواند S7-300 ، S7-400 یا S7-1200 باشد ، انجام می‌گیرد.
- در قسمت Write PLC program می‌توان هر نوع بلوک برنامه نویسی مانند OB ، FB یا FC را ایجاد کرد و با انتخاب زبان برنامه نویسی دلخواه مانند LAD ، FBD یا STL و ... آن را برنامه نویسی نمود.
- در قسمت Configure an HMI screen می‌توان پنل اپراتوری TP ، OP یا MP را به پروژه اضافه کرد و تنظیمات آن و طراحی گرافیک آن را با winccFlexible انجام داد.

مثال ۳-۱۰ : Scale غیر خطی با ۴ نقطه

اگر تغییرات سیگنال آنالوگ بصورت خطی و مانند نمودار شکل ۳-۸۸ باشد ، نمی‌توان از فانکشن Scale قبلی استفاده نمود ، زیرا فانکشن قبلی دو نقطه (0 , 0) و (100 , 27648) را با یک خط مستقیم تقریب می زند . این تقریب اختلاف زیادی با واقعیت دارد و قابل استفاده نیست.



شکل ۳-۸۸

دقیق‌ترین روش این‌است که تابع y=f(x) را داشته باشیم و فرمول آن را در برنامه پیاده سازی کنیم ولی اگر تابع بطور دقیق مشخص نباشد، نیاز به تبدیل منحنی به چند پاره خط داریم. در واقع منحنی شکل فوق را به چهار قسمت تقسیم می‌کنیم:

- ناحیه ۱ : وقتی مقدار سیگنال A/D یعنی مقدار x از 6912 کمتر است . در این ناحیه ، تقریب براساس خطی است که از نقطه (0 , 0) و (75 , 6912) عبور می‌کند.
- ناحیه ۲ : وقتی مقدار سیگنال A/D یعنی مقدار x از 6912 بیشتر و از 13824 کمتر است . در این ناحیه ، تقریب براساس خطی است که از نقطه (75 , 6912) و (90 , 13824) عبور می‌کند.

- ناحیه ۳ : وقتی مقدار سیگنال A/D یعنی مقدار x از 13824 بیشتر و از 20736 کمتر است . در این ناحیه ، تقریب براساس خطی است که از نقطه (90 , 13824) و (100 , 20736) عبور می‌کند.
- ناحیه ۴ : وقتی مقدار سیگنال A/D یعنی مقدار x از 20736 بیشتر است . در این ناحیه ، تقریب نیاز نیست و مقدار خروجی روی 100 ثابت می ماند.

در این مثال ، فرض شده است که مقادیر داده شده روی محور x و محور y نمودار قبلی ثابت است. بنابراین فانکشن مورد نظر فقط دارای یک ورودی Input از جنس integer است که مقدار سیگنال خام را به برنامه می‌دهد و یک خروجی Output از جنس Real که مقدار سیگنال Scale شده را بر می‌گرداند.

منطق برنامه

اگر از مقادیر نشان داده شده در نمودار قبلی معادله خط را بدست آوریم ، خواهیم داشت:

معادله خط ناحیه ۱

Y-0 = (75-0)/(6192-0)* (X -0)

که بصورت ساده شده خواهیم داشت:

Y=(75/6192) *X

معادله خط ناحیه ۲

Y-75 = (90-75) /(13824-6192) *(X-6192)

که بصورت ساده شده خواهیم داشت :

Y= (15/6192)*X + 60

معادله خط ناحیه ۳

Y-90 = (100-90) /(20736-13824) *(X-13824)

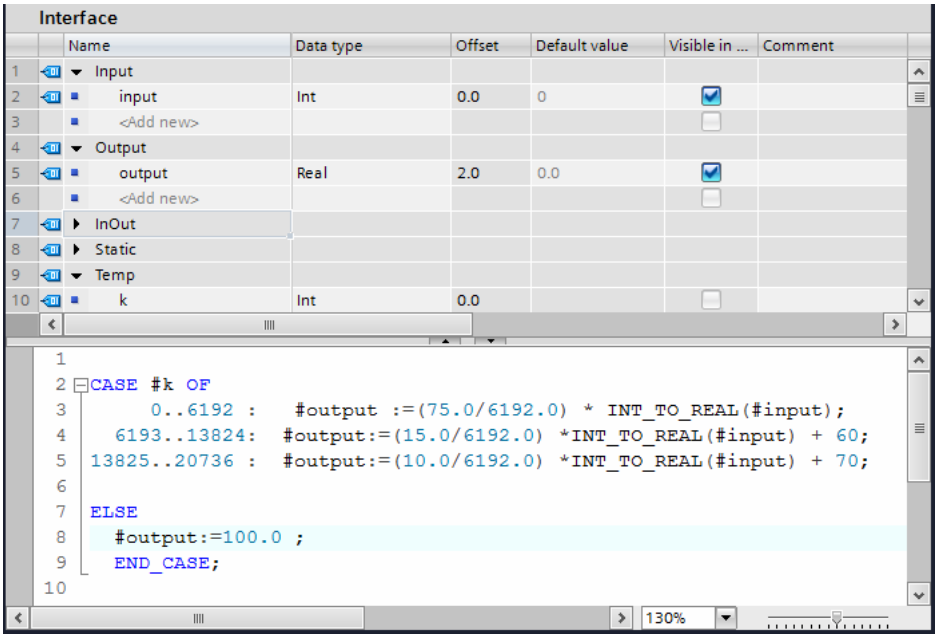
که بصورت ساده شده خواهیم داشت:

Y= (10/6192)*X + 70

در ناحیه ۴ مقدار Y=100 ثابت است و نیاز به معادله خط ندارد.

برنامه با استفاده از دستور Case بصورت شکل ۳-۸۹ خواهد بود. توجه داشته باشید که دستور CASE نیاز به یک متغیر دارد که آن را در قسمت temp با نام k تعریف کرده ایم.

در متن برنامه نیز دقت شود که چون ورودی از جنس Integer است ، نیاز به تابع تبدیل خواهیم داشت.



شکل ۳-۸۹

تمرین ۳-۱۰

مثال ۳-۱۰ را در حالتی که حدود Scale بین 0 تا 100 ثابت نیست ، باز نویسی کنید . در واقع ، این حدود به عنوان ورودی فانکشن با پایه های Out_L و Out_H و از جنس Real می‌باشند.

در بین روش‌های برنامه نویسی PLC روش SCL تا حدی گمنام مانده است، در حالی که توانایی این زبان نسبت به زبان‌های سطح پایین نظیر LAD/FBD/STL به مراتب بیشتر است و در سیستم‌های کنترل پیشرفته کاربرد زیادی دارد. مولفین با تشریح محیط این نرم افزار و دستورات برنامه نویسی آن، تلاش کرده‌اند با ذکر مثال‌های کاربردی فراوان، دانش و مهارت خواننده را در این زمینه ارتقاء دهند.

دفتر مرکزی و مرکز پخش: ۶۶۴۱۱۴۸۱-۶۶۴۰۳۵۴۸
۰۹۱۲۲۷۰۴۸۳۷-۰۹۱۹۵۵۰۷۱۲۵
www.Qeddis.com