

فایل‌های پروژه‌های برنامه‌نویسی Symbian

موسی مرادی

ایمیل: mousamk@gmail.com

وبلاگ: <http://series60.blogfa.com>

توجه: کپی کردن و انتشار این مقاله آزاد است ولی تغییر آن در انحصار نویسنده است.

/*توضیح: مثال‌های و اسامی که برای مطالب زیر نوشته‌ام، همگی مربوط به یک برنامه به نام MKLightOn هستند.*/

فایل‌های پروژه‌های برنامه‌نویسی Symbian:

همانطور که می‌دانید، اکنون IDEهای زیادی برای ساخت برنامه برای موبایل وجود دارند و این IDEها به طور اتوماتیک تمام فایل‌های لازم را ایجاد می‌کنند و نیازی نیست که آنها را به طور دستی ایجاد کنید، ولی بهتر است که شما آنها را بشناسید و با ساختارشان آشنا شوید تا هم برنامه‌نویس ماهری شوید و هم اینکه در صورت لزوم و عدم پشتیبانی IDE از بعضی موارد، خودتان به طور دستی آنها را انجام دهید.

خب، برای ایجاد هر پروژه برای موبایل باید فایل‌های مختلفی وجود داشته باشد: یک فایل با پسوند `mmp` و یک فایل با پسوند `tsc`؛ این دو مهمترین فایل‌های مربوط به یک پروژه هستند. به جز این دو فایل، فایل‌های دیگری هم مثل `bld.inf` و فایل `aif` و ... باید وجود داشته باشند که توضیح می‌دهیم.

این فایل‌ها هنگام ساخته شدن برنامه استفاده می‌شوند.

فایل `bld.inf`

این فایل، فایل تعریف اجزای پروژه هست. هر پروژه باید این فایل را داشته باشد و این فایل هم باید حاوی آدرس تمامی فایل `mmp`های مربوط به پروژه باشد. البته در برنامه‌های ساده که فقط یک فایل `mmp` دارند، این فایل فقط به آن فایل `mmp` موجود اشاره می‌کند. در برنامه‌های پیچیده که حاوی اجزای دیگری به جز برنامه اصلی (مانند `dll`ها و ...) هستند، باید `mmp`های جداگانه‌ای برای هر کدام وجود داشته باشد و بالتبع باید نام تمام آنها در فایل `bld.inf` آمده باشد.

ساختار یک فایل `bld.inf` ساده که فقط به یک `mmp` اشاره می‌کند، به صورت زیر است:

```
PRJ_MMPFILES
```

```
MK_LightOn.mmp
```

فایل `mmp`

این فایل، فایل تعریف پروژه است. یک فایل `mmp` ویژگی‌های یک پروژه را مشخص می‌کند و توسط ابزار ساخت SDK (`abld.bat`) استفاده می‌شود. در صفحه بعد محتویات فایل `mmp` مربوطه به مثال این مقاله را می‌بینیم.

توضیح ساختار این فایل:

TARGET: نام برنامه را مشخص می‌کند. (توضیح همانطور که فایل‌های اجرایی ویندوز دارای پسوند `exe` هستند، فایل‌های اجرایی `GUI` سیمپین هم دارای پسوند `app` هستند. یعنی بعد از نصب یک برنامه در سیمپین، هنگام اجرای برنامه باید این فایل را اجرا کنید.)

TARGETTYPE: در اینجا `app` است؛ به این معنی که این برنامه، یک برنامه `GUI` (دارای واسط کاربر گرافیکی) است.

TARGETPATH: موقعیت برنامه و اجرای آن را مشخص می‌کند. (البته در موبایل و یا شبیه‌ساز). این موقعیت همیشه باید در پوشه `\system\apps\` باشد.

UID: UID (شماره سیستم منحصر به فرد) برنامه `GUI` را مشخص می‌کند.

SOURCEPATH: مسیر سورس فایل‌هایی را که بعد از این خط ذکر خواهند شد را مشخص می‌کند. (حتماً می‌دانید که دونقطه به معنی یک پوشه بالاتر است؛ توضیح بیشتر این که پوشه پیش‌فرض، پوشه `group` است. ولی برای این که پوشه `src` را بشناسانیم باید به پوشه بالای `group` برویم و داخل پوشه `src` شویم. به همین دلیل ابتدا از دو نقطه استفاده کردیم و بعد وارد `src` شدیم.)

SOURCE: سورس فایل‌ها را مشخص می‌کند.

RESOURCE: به سورس فایل‌هایی اشاره می‌کند که حاوی اکثر اجزای رابط کاربری مانند منوها، پنجره‌های محاوره، متن‌ها و غیره

هستند.

USERINCLUDE: مسیر فایل‌های headerی که خودمان نوشته‌ایم و ضمیمه خواهیم کرد را مشخص می‌کند.

SYSTEMINCLUDE: همانطور که خودتان هم متوجه شده‌اید، مسیر فایل‌های سیستم را که از قبل وجود دارند و ما ضمیمه خواهیم

کرد، مشخص می‌کند.

```
TARGET      MK_LightOn.app
TARGETTYPE  app
UID         0x100039CE 0x20001568
TARGETPATH  \system\apps\MK_LightOn
```

```
LANG        01
```

```
SOURCEPATH  ..\src
SOURCE      MK_LightOnmain.cpp
SOURCE      MK_LightOnapp.cpp
SOURCE      MK_LightOndocument.cpp
SOURCE      MK_LightOnappui.cpp
SOURCE      MK_LightOnview.cpp
SOURCE      MK_LightOncontainer.cpp
```

```
RESOURCE    MK_LightOn.rss
```

```
USERINCLUDE ..\inc
USERINCLUDE ..\src
```

```
SYSTEMINCLUDE \epoc32\include
SYSTEMINCLUDE \epoc32\include\borland
```

```
LIBRARY      euser.lib
LIBRARY      apparc.lib
LIBRARY      cone.lib
LIBRARY      eikcore.lib
LIBRARY      avkon.lib
LIBRARY      egul.lib
LIBRARY      eikcoctl.lib
LIBRARY      bafl.lib
LIBRARY      eikctl.lib
LIBRARY      fbscli.lib
LIBRARY      etext.lib
LIBRARY      form.lib
LIBRARY      gdi.lib
LIBRARY      insock.lib
LIBRARY      aknotify.lib
LIBRARY      aprgrfx.lib // Using TApTask - for sending app to background
```

```
AIF MK_LightOn.aif ../aif MK_LightOn.rss c12 Main_Icon.bmp Main_Icon_Mask.bmp List_Icon.bmp
List_Icon_Mask.bmp
```

LIBRARY: کتابخانه‌های ضروری برای برنامه را مشخص می‌کند.

AIF: به فایل اطلاعاتی برنامه (Application Information File) اشاره می‌کند که حاوی آیکن‌ها و نام برنامه به زبان‌های مختلف و ... است. همانطور که در مثال بالا می‌بینید، برای تعریف فایل aif، بعد از عبارت AIF، نام فایل را می‌نویسیم؛ سپس نام فایل‌های منبع و بالاخره نام آیکن‌ها و ماسک‌هایشان را می‌نویسیم.

فایل‌های منبع (Resource Files):

در سیستم عامل سیمبین، فایل‌های منبع برای تعریف ظاهر برنامه در صفحه نمایش به کار می‌روند. بیشتر اطلاعاتی که ظاهر، رفتار و کارکرد یک برنامه را مشخص می‌کنند، در فایل‌های منبع ذخیره می‌شوند. به جز قسمت اصلی برنامه، همه چیز از قسمت وضعیت و نوار منو گرفته تا پنجره‌های محاوره (Dialog Boxes) می‌توانند در فایل منبع تعریف شوند. منابعی که فقط در یک فایل ذخیره شده‌اند، به راحتی هنگام لزوم، لود می‌شوند و از این رو اشغال حافظه نیز به حداقل می‌رسد.

منابع برنامه در فایل‌های متنی تعریف می‌شوند (با پسوند rsc) و هنگام ساخت کامپایل شده و به صورت فایل‌های باینری فشرده شده و ذخیره می‌شوند (به طور پیش فرض با پسوند rsc) و بالاخره هنگام اجرای برنامه (run time) استفاده می‌شوند. برای راحتی کار، بهتر است تمام متن‌های رابط کاربری، به طور جداگانه در یک فایل header با پسوند loc قرار داده شوند؛ که به فایل منبع اصلی ضمیمه (include) خواهند شد. برای ترجمه برنامه به زبان‌های مختلف از فایل‌های loc استفاده می‌شود.

فایل‌های منبع به ظاهر پیچیده به نظر می‌رسند ولی با تمرین زیاد و استفاده کردن از مثال‌ها و تیوتوریال‌های مختلف، می‌توانید بر آنها تسلط پیدا کنید. البته همانطور که در ابتدا هم گفتم اگر از یک IDE مثل C++BuilderX استفاده می‌کنید، نیازی به ویرایش دستی فایل‌های منبع ندارید و فقط کافی است که منابع برنامه را توسط ابزار خود IDE طراحی کنید. IDE خودش کدهای لازم را تولید خواهد کرد.

سورس فایل‌ها (Source Files):

سورس فایل‌های زیر کدهای برنامه را تشکیل می‌دهند:

MK_LightOnapp.cpp
MK_LightOnappui.cpp
MK_LightOncontainer.cpp
MK_LightOndocument.cpp
MK_LightOnmain.cpp
MK_LightOnview.cpp

ساختار این فایل‌ها دقیقاً مثل سورس فایل‌های زبان C++ هستند و هیچگونه تفاوتی ندارند.

MK_LightOn.rsc: فایل منبع استاندارد که حاوی تعاریف UI (رابط کاربری) زیادی است. این فایل توسط کامپایلر منبع استفاده می‌شود که به طور اتوماتیک قبل از کامپایل شدن سورس کدها اجرا می‌شود. فایل خروجی، یک فایل باینری است که هنگام اجرای برنامه استفاده می‌شود.

فایل‌های header:

MK_LightOn.hrh: حاوی تعاریف تعدادی از ثوابت است که هم در سورس‌های (cpp) و هم در فایل‌های منبع (rsc) استفاده می‌شوند. این ثوابت عبارتند از Command ID ها، Key ID ها یا View ID های مشخص شده برای برنامه. این ثوابت معمولاً برای مدیریت و کنترل کلیدها یا دستورات در پنجره‌های محاوره به کار می‌روند.

توجه داشته باشید که فایل avkon.hrh (واقع در مسیر \Epo32\Include) حاوی تعاریف ثوابت زیادی هست که توسط framework رابط کاربری سری 60 تعریف شده‌اند. این فایل، یک فایل خیلی مهم برای برنامه‌های سری 60 است. برای مثال یکی از ثوابت تعریف شده،

Exit منوی EAKnCmdExit است. باید همیشه هنگام اجرا شدن این دستور را تولید کند.

مقدار استاندارد ی که باید هنگام خروج از یک برنامه به سیستم برگردانده شود، EEikCmdExit است که در فایل uikon.hrh تعریف شده است.

فایل‌های header مثال ما به قرار زیر هستند:

MK_LightOndocument.cpp	فایل header برای	:MK_LightOndocument.h
MK_LightOnapp.cpp	فایل header برای	:MK_LightOnapp.h
MK_LightOnappui.cpp	فایل header برای	:MK_LightOnappui.h
MK_LightOncontainer.cpp	فایل header برای	:MK_LightOncontainer.h
MK_LightOnview.cpp	فایل header برای	:MK_LightOnview.h

فایل‌های اجرایی (Executable Files):

در این قسمت فایل‌های اجرایی ایجاد شده توسط پروسه ساخت، نوع آنها، هدف آنها و مسیر آنها در شبیه‌ساز و موبایل را توضیح می‌دهیم.

برنامه‌های کنسولی:

فایل اجرایی برنامه‌های کنسولی یا همان برنامه‌های دارای رابط کاربر دستوری (CDI) در سیمین دارای پسوند exe هستند. این فایل‌ها در شبیه‌ساز در مسیرهای زیر قرار دارند:

در حالت wins\udeb (این حالت برای اشکالزدایی است) :Epo32\Release\wins\UDEB

در حال wins\urel (این حالت برای ساخت نهایی و انتشار برنامه است) :Epo32\Release\wins\UREL

در موبایل به طور قراردادی فایل‌های exe در پوشه \system\programs قرار می‌گیرند، ولی در هر مسیر دیگری هم می‌توانند اجرا شوند. البته چون هدف ما برنامه‌های GUI هستند، توضیح بیشتری نمی‌دهیم و به بحث خودمان برمی‌گردیم.

برنامه‌های گرافیکی:

حداقل دو فایل app و rsc یک برنامه GUI را می‌سازند (مثلاً فایل‌های MK_LightOn.app و MK_LightOn.rsc). فایل app اجرایی برنامه است و محصول پروسه کامپایل است. فایل rsc هم که فایل منبع است و به صورت باینری است. معمولاً یک فایل اطلاعات برنامه (aif) هم وجود دارد که حاوی آیکن‌ها و عناوین برنامه و بعضی قابلیت‌های دیگر برنامه است. یک برنامه بدون وجود فایل aif هم می‌تواند کار کند، ولی در این صورت سیستم یک آیکن پیش‌فرض برای برنامه در نظر می‌گیرد.

در سیستم عامل سیمین یک برنامه GUI، حالت خاصی از یک کتابخانه اتصال پویا (dll) است که همیشه دارای پسوند app است. یک پروسه ویژه در سیستم تدارک دیده شده است که وظیفه‌اش لود کردن درست برنامه‌های گرافیکی است. برنامه‌نویس‌ها باید برنامه‌هایشان را با ملزومات این پروسه مطابقت دهند و باید تعدادی تابع مجازی هم تدارک ببینند تا برنامه‌شان به درستی اجرا شود.

برای این که برنامه‌ها توسط سیستم به طور اتوماتیک‌وار شناخته شوند، همه آنها باید در پوشه‌ای مثل \system\apps\appname قرار داشته باشند. بنابراین در مثال بالا، فایل‌هایی که ما شرح دادیم باید در پوشه‌ای مثل \system\apps\MK_LightOn قرار داده شوند.

ممکن است برای درست کار کردن برنامه، فایل‌های دیگری هم مثل فایل‌های تصویری فشرده (mbm)، فایل‌های اطلاعاتی و ... نیاز باشند که این‌ها هم بهتر است در همان پوشه فایل اصلی برنامه قرار داده شوند.

فایل‌های AIF:

فایل‌های اطلاعاتی برنامه، هنگام اجرای برنامه استفاده می‌شوند و حاوی اطلاعاتی هستند که عبارتند از:

آیکن‌ها در سایزهای مختلف تا سیستم بتواند توسط آنها برنامه را نشان دهد.

عناوین برنامه در زبان‌های پشتیبانی شده.

قابلیت‌های برنامه، مثلاً قابلیت اضافه کردن document، ایجاد فایل جدید، این که برنامه مخفی است یا نه و ...

هر برنامه باید فایل aif شخصی خودش را داشته باشد. که حاوی تصاویر و عناوین است و همانطور که گفتیم در صورت نبود این فایل، سیستم آیکن‌های پیش‌فرض خودش را برای برنامه در نظر می‌گیرد. فایل‌های aif را می‌توان توسط ابزار Aif builder همراه SDK هم ایجاد کرد. اطلاعات داخل این فایل به صورت متنی نیست و نمی‌توان آن را با ابزارهای متنی ویرایش کرد.

فایل‌های PKG:

این فایل‌ها (که البته برای هر پروژه فقط یک فایل وجود دارد) حاوی دستوراتی است که برای ساخت فایل نهایی SIS باید استفاده شوند. در این فایل اطلاعاتی از قبیل زبان‌های پشتیبانی شده توسط برنامه، فایل‌هایی که باید به دستگاه هنگام نصب کپی شوند، ورژن برنامه، UID برنامه و ... وجود دارد. من قبلاً در وبلاگم مطلبی در مورد ساختار این نوع فایل‌ها نوشته بودم و الان هم تکمیل شده همان مطلب را در زیر می‌خوانید: به علت این که این فایل خیلی مهم است، ساختار این فایل را به طور تقریباً کامل توضیح می‌دهم. در زیر فایل pkg مربوط به برنامه MKLightOn را می‌بینید.

```
&EN
```

```
#{"MKLightOn"}, (0x20001568), 1, 1, 0  
(0x101F6F88), 0, 0, 0, {"Series60ProductID"}
```

```
"install.txt"-"", FILETEXT, TEXTCONTINUE
```

```
"D:\Symbian\7.0s\Series60_v21\Epoc32\release\armi\urel\MK_LIGHTON.APP"-
```

```
"!:\system\apps\MK_LightOn\MK_LIGHTON.APP"
```

```
"D:\Symbian\7.0s\Series60_v21\Epoc32\data\z\system\apps\MK_LIGHTON\MK_LIGHTON.R01"-
```

```
"!:\system\apps\MK_LIGHTON\MK_LIGHTON.R01"
```

```
"D:\Symbian\7.0s\Series60_v21\Epoc32\data\z\system\apps\MK_LIGHTON\MK_LIGHTON.AIF"-
```

```
"!:\system\apps\MK_LightOn\MK_LIGHTON.AIF"
```

```
"blog.txt"-"", FILETEXT, TEXTCONTINUE
```

توضیح سطرهای فایل:

- در سطر اول زبان‌های پشتیبانی شده برنامه نوشته می‌شوند، به این ترتیب که ابتدا یک علامت & قرار می‌گیرد، سپس نام اختصاری زبان پیش‌فرض، سپس در صورت وجود زبان‌های دیگر هم، کاما می‌گذاریم و علامت اختصاری آنها را می‌نویسیم، مثلاً اگر برنامه‌ای سه زبان انگلیسی (پیش‌فرض)، فرانسوی و فارسی را پشتیبانی کند، این عبارت باید به صورت &EN,FR,FA باشد. (البته نوع دیگر برنامه‌های چند زبانه هم برای موبایل وجود دارد که در آن خط مربوط به زبان را کلاً نمی‌نویسند؛ در مطالب بعد آن را توضیح خواهیم داد)
- سطر بعد مربوط به مشخصات برنامه است. ابتدا علامت # قرار می‌گیرد، سپس نام برنامه داخل آکولاد و نقل قول دوتایی قرار می‌گیرد، سپس UID برنامه بعد از آن سه عدد وجود دارد که ورژن برنامه را مشخص می‌کنند.
- سطر بعدی مربوط به مشخصات دستگاه مقصد است. در ابتدای سطر، داخل پرانتز کدی وجود دارد که شبیه UID است. این کد مشخص کننده نوع سیستمی است که برنامه می‌تواند در آن نصب شود. IDE به طور خودکار این کد را وارد می‌کند، شما آن را تغییر ندهید. با سه

عدد بعدی کاری نداشته باشید و آنها را هیچوقت تغییر ندهید. این سه عدد همیشه باید صفر باشند. عبارت آخری هم مشخص می‌کند که این برنامه روی گوشی‌های با رابط سری ۶۰ قابل نصب است.

• سطر بعدی مشخص می‌کند که در هنگام نصب برنامه و رسیدن به این مرحله، باید محتویات فایلی با نام `install.txt` نمایش داده شود. محتویات فایل متنی هر چیزی می‌تواند باشد و به خودتان بستگی دارد. می‌توانید اطلاعاتی درباره برنامه و کپی رایت و یا چیزهای دیگر به کاربر بدهید. در اینجا فقط باید یک نکته را مد نظر داشته باشید. اگر متنی که می‌خواهید نشان دهید، فارسی است، هنگام نوشتن متن در Notepad باید آن را به صورت Unicode و یا UTF-8 ذخیره کنید. اگر به حالت عادی یا همان ANSI ذخیره کنید. متن فارسی درست نشان داده نخواهد شد. و البته متأسفانه یک اشکال دیگر هم وجود دارد. اگر متن شما حاوی حروف «گ چ پ ژ» تنها باشد، درست نشان داده نمی‌شود. یعنی اگر در متن کلمه‌ای مثل «وبلاگ» داشته باشید، این کلمه به صورت «وبلا□□» نشان داده می‌شود. ولی اگر حرف مورد نظر تنها نباشد، مثلاً در کلمه «رایگان»، هیچ مشکلی وجود نخواهد داشت. پس سعی کنید از کلماتی استفاده کنید که حاوی حروف مورد نظر به صورت تنها نباشند. مثلاً می‌توانید به جای کلمه «وبلاگ» از کلمه «وبلاگم» و یا ... استفاده کنید. البته امیدواریم این مشکل در ورژن‌های آینده سیمبین رفع شود.

نکته دیگری که باید مد نظر قرار داده شود، این است که باید فایل متنی شما در همان پوشه‌ای باشد که فایل `pkg` در آن است، در غیر این صورت، باید مسیر فایل را هم مشخص کنید؛ مثلاً به این صورت `"Text\install.txt"` و ضمناً این فایل متنی در موبایل کپی نخواهد شد و بعد از نصب برنامه از بین خواهد رفت. زیرا پس از نام برنامه، عبارت `"-"` قرار گرفته است.

در این سطر معنی عبارت TEXTCONTINUE این است که در هنگام نمایش این متن، فقط یک انتخاب با نام Continue یا OK به کاربر داده می‌شود که که با انتخاب آن توسط کاربر، نصب برنامه ادامه می‌یابد. بقیه انتخاب‌های ممکن به این صورت است:

TEXTSKIP: در این صورت دو انتخاب Yes و No به کاربر داده می‌شود که در صورت انتخاب Yes نصب ادامه می‌یابد و در صورت انتخاب No نصب برنامه ادامه می‌یابد ولی فایل بعدی موجود در `pkg` نصب نخواهد شد.

TEXTTEXT: دو انتخاب Yes و No به کاربر می‌دهد. اگر Yes انتخاب شود، نصب برنامه ادامه می‌یابد. ولی اگر No انتخاب شود، نصب برنامه متوقف می‌شود و فایل‌های نصب شده هم پاک می‌شوند.

TEXTABORT: دو انتخاب Yes و No به کاربر می‌دهد که در صورت انتخاب Yes نصب ادامه می‌یابد و در صورت انتخاب No نصب برنامه متوقف می‌شود، ولی فایل‌های که قبلاً نصب شده‌اند، پاک نخواهند شد. (در واقع یک نصب ناقص اتفاق می‌افتد).

• در سطرهای بعد، فایل‌هایی که باید داخل فایل `sis` قرار داده شوند، مشخص می‌شود. فرمت این سطرها به این صورت است که ابتدا داخل علامت نقل قول، نام و مسیر فایل مورد نظر در کامپیوتر وارد می‌شود. (که معمولاً داخل `Epoc32` - شبیه‌ساز - هستند.) سپس یک خط تیره، سپس مسیری که باید فایل در گوشی در آنجا کپی شود، مشخص می‌شود؛ که همانطور که خودتان هم می‌دانید معمولاً در مسیر `\system\apps\appname` قرار می‌گیرند. علامت ! هم به این معنی است که نام درایو (C یا E) به انتخاب کاربر است. مثلاً در مثال بالا ما سه فایل `MK_LightOn.app`، `MK_LightOn.r01` و `MK_LightOn.aif` را به فایل `sis` ضمیمه می‌کنیم. (فایل `r01` فایل `string`های برنامه در زبان انگلیسی است. `01` کد زبان انگلیسی است.) قسمت دوم این سطر بسیار مهم می‌باشد زیرا نه تنها مقصد فایل را مشخص می‌کند، بلکه مشخص می‌کند که هنگام `UnInstall` کردن برنامه، این فایل‌ها پاک شوند. حال ما اگر برنامه‌ای بنویسیم که یک فایل را هنگام اجرای برنامه ایجاد می‌کند (مثلاً فایل ذخیره اطلاعات یا ...) و از قبل وجود ندارد می‌توانیم از این ویژگی استفاده کنیم تا هنگام پاک کردن برنامه از گوشی، آن فایل هم پاک شود. برای انجام این کار قسمت اول را خالی می‌گذاریم. مثلاً فرض کنید برنامه ما یک فایل هم با نام `MK_LightOnData.dat` ایجاد خواهد کرد، در این صورت سطر زیر را هم باید اضافه کنیم:

`"!:\system\apps\MK_LIGHTON\MK_LightOnData.dat"`

• و باز در سطر آخر متنی را به کاربر نشان می‌دهیم. مشاهده می‌کنید که می‌توان چند بار توضیحاتی به کاربر نشان داد. من در اینجا از کاربر درخواست کرده‌ام که به وبلاگ من مراجعه کند.

یک نکته دیگر هم اینکه اگر خواستید در این فایل کامنت یا توضیحات بنویسید، نباید از علامت // استفاده کنید بلکه باید از علامت ; استفاده

کنید.

/ در صورتی که سؤال و یا مشکلی داشتید و یا به نظرتان مطلبی در این مقاله برایتان گنگ بود و یا خواستید در مورد مقاله نظرتان را بگویید، می‌توانید با بنده از طریق ایمیل ارتباط داشته باشید/

پایان

موسی مرادی

mousamk@gmail.com

<http://series60.blogfa.com>