

جزوه برنامه‌ریزی میکروکنترلر PIC



by Yesu Thommandru
Iowa State University – ECpE
November 2006

ترجمه: ب. صفری

از آنجا که متن اصلی جزوه رایگان است، به تبع آن ترجمه آن نیز رایگان و هرگونه استفاده از آن مجاز است. پیشنهاد خود را به آدرس safari.b1987@gmail.com ارسال نمایید.

فهرست مطالب

۵	مقدمه	۱
۵	۱.۱. هدف	۱.۱
۵	۲.۱. قراردادهای نوشتار	۲.۱
۵	۳.۱. چه کسانی این کتاب را بخوانند و کتب پیشنهادی	۳.۱
۶	۴.۱. منابع	۴.۱
۷	انتخاب میکروکنترلر PIC	۲
۷	۱.۲. مقدمه	۱.۲
۷	۲.۲. انواع PIC	۲.۲
۸	محیط توسعه مجتمع (IDE)	۳
۸	زبانهای برنامه نویسی	۴
۹	کمپایلر، اسمبلر و لینکر	۵
۱۰	استفاده از MPLab IDE	۶
۱۴	نوشتن نرم افزار	۷
۱۵	Burning فرایند	۸
۲۱	Breadboard بستن مدار PIC بر روی	۹

فهرست اشکال

۱۰	شکل ۱: انتخاب کمپایلر در MPLab IDE
۱۱	شکل ۲: ساخت یک پروژه جدید در MPLab IDE
۱۱	شکل ۳: پنجره خوش آمد گویی برای ساخت پروژه جدید
۱۲	شکل ۴: انتخاب دستگاه
۱۲	شکل ۵: نامگذاری پروژه
۱۳	شکل ۶: آخرین صفحه ساخت پروژه جدید
۱۳	شکل ۷: انتخاب مسیر Workspace
۱۵	شکل ۸: Build کردن پروژه
۱۶	شکل ۹: فایل‌های تولید شده بعد از Build
۱۷	شکل ۱۰: پروگرامر DataMan برای پروگرام کردن PIC
۱۷	شکل ۱۱: PIC بر روی پروگرامر
۱۸	شکل ۱۲: انتخاب دستگاه در نرم افزار Dataman-48XP/UXP
۱۹	شکل ۱۳: بارگذاری فایل Hex
۱۹	شکل ۱۴: ویرایش بافر
۲۰	شکل ۱۵: پیکربندی میکرو
۲۱	شکل ۱۶: پروگرام کردن
۲۲	شکل ۱۷: دیاگرام پایه های PIC16F87A/877A
۲۳	شکل ۱۸: بستن پایه MCLR/Vpp
۲۳	شکل ۱۹: اسپلاتور
۲۳	شکل ۲۰: تراشه تایمر ۵۵۵
۲۴	شکل ۲۱: مدار پروژه LED چشمک زن
۲۴	شکل ۲۲: نمای Breadboard پروژه LED چشمک زن

۱. مقدمه

۱.۱. هدف

هدف این نوشتار فراهم آوردن خودآموزی کوتاه و ساده برای برنامه ریزی میکروکنترلرهای PIC است. این خودآموز با دستورالعمل‌هایی برای انتخاب PIC مناسب آغاز می‌شود و با بستن مدار آن بر روی Breadboard به پایان می‌رسد.

۲.۱. قراردادهای نوشتار

در این نوشتار انواع مختلفی برای نوشته‌های گوناگون انتخاب شده‌است که به خواننده اجازه می‌دهد اطلاعات مختلف را از یکدیگر جدا کند. کدها و شبه کدها با فونت *Courier* و به صورت رنگی نوشته می‌شوند. به عنوان مثال:

```
#include <stdio.h>
void main(int argc, char *argv[])
{
printf("Star Wars!\n");
return;
}
```

دکمه‌ها و منوها با فونت *Arial* و به صورت **Bold** (تو پر) نوشته می‌شوند. نکات مهم نیز با پس زمینه خاکستری به این صورت نوشته می‌شوند.

۳.۱. چه کسانی این کتاب را بخوانند و کتب پیشنهادی

مقصود اصلی این نوشتار دانشجویان رشته‌های برق و کامپیوترند. بعلاوه سایر دانشجویان و کسانی که علاقه‌مند به دانستن مفاهیم اولیه درباره برنامه‌نویسی برای میکروکنترلرهای PIC هستند می‌توانند از این کتاب بهره‌مند شوند.

چندین کتاب پیشنهادی در این رابطه وجود دارد که فهرست زیر بخشی از آنان را معرفی می‌کند:

- ✚ *Introduction to microelectronic systems: the PIC 16F84 microcontroller by Martin Bates.*
- ✚ *PIC microcontroller: an intro to software and hardware interfacing by Han-Way Huang.*
- ✚ *The PIC microcontroller: your personal introductory course by John Morton.*
- ✚ *PIC microcontroller project book by John Lovine.*
- ✚ *Programming and customizing the PIC microcontroller by Myke Predko.*
- ✚ *The quintessential PIC microcontroller by Sid Katzen.*

نکته: بنا به تجربه شخصی بیشتر نرم‌افزارهای این کتاب‌ها به زبان اسمبلی نوشته شده‌است و برای دانشجویانی که قصد برنامه نویسی در سطوح بالا را دارند مفید نیست.

- ✦ Schildt, Herbert. *C/C++ Programmer's Reference 2nd Edition*. McGraw-Hill Publishing. NewYork, 2000.
- ✦ Morton, John. *PIC Your Personal Introductory Course*. Newnes. Boston, 1998.
- ✦ Bergquist, Carl. *Guide to PICMICRO Microcontrollers*. Sams Technical Publications. Indianapolis,2001.
- ✦ Predko, Myke. *Handbook of Microcontrollers*. McGraw-Hill Publishing. New York, 1999.
- ✦ Predko, Myke. *PICMicro Microcontroller Pocket Reference*. McGraw-Hill. New York. 2000.
- ✦ Smith, D.W. *PIC in Practice*. Newnes. Oxford. 2002.
- ✦ Microchip.com. PIC16F877A. 2006. <http://www.microchip.com/>
- ✦ MicrochipC.com PICMicros and C <http://www.microchipc.com/>
- ✦ Best Microcontroller Projects <http://www.best-microcontroller-projects.com/index.html>

۲. انتخاب میکروکنترلر PIC

۱.۲. مقدمه

میکروکنترلرهای PIC پردازنده‌های محبوبی هستند که توسط Microchip Technology با حافظه، Bus و وسایل جانبی داخلی که می‌توانند برای کارهای مختلف مورد استفاده قرار گیرند طراحی شد. PIC مخفف Programmable Intelligent Computer است که امروزه به عنوان Peripheral Interface Controller شناخته می‌شود.

۲.۲. انواع PIC

میکروکنترلرهای PIC به دو دسته بزرگ تقسیم می‌شوند: کنترلهای ۸ بیتی و ۱۶ بیتی. هر دسته خود به زیرشاخه‌های بیشتری که در جدول زیر می‌بینید تقسیم می‌شود.

8 bit	16 Bit
PIC10	PIC24F
PIC12	PIC24H
PIC14	dsPIC30
PIC16	dsPIC33
PIC18	

خانواده‌های PIC10 تا PIC14 از میکروهای low-end هستند و PIC16 تا PIC18 از دسته mid-end و تمامی PICهای ۱۶ بیت جزء کنترلهای High-end هستند.

نکته: اکثریت دانشجویان و پروژه‌ها به کنترلهای mid-level احتیاج دارند. پر استفاده ترین PIC برای اینگونه موارد PIC16F877/A است.

هر PIC ویژگی‌ها خاص خود و تفاوت‌های ریزی با دیگر PICها دارد. انتخاب درست برای پروژه شما به فاکتورهای متعددی وابسته است:

- ✚ آیا این پروژه نیاز به ورودی و خروجی آنالوگ دارد؟
- ✚ آیا این پروژه نیاز به ورودی و خروجی دیجیتال دارد؟
- ✚ چند پین I/O مورد نیاز است؟
- ✚ آیا پروژه به زمانبندی دقیقی نیاز دارد؟
- ✚ چه مقدار حافظه مورد نیاز است؟
- ✚ آیا I/O سریال مورد نیاز است؟
- ✚ ...

همچنین PIC در پکیج‌های مختلفی ارائه شده است:

- ✚ Plastic Dual Inline Package (PDIP)
- ✚ Small-Outline Transistor (SOT)
- ✚ Dual Flat No-lead (DFN)
- ✚ Mini Small Outline Package (MSOP)
- ✚ Thin Quad Flat Pack (TQFP)
- ✚ Plastic Leaded Chip Carrier (PLCC)
- ✚ CERamic QUADpack (CERQUAD)

دلیل وجود پکیج‌های مختلف آن است که بعضی PICها با ۱۰۰ پایه I/O وجود دارد. عمدتاً میکروکنترلرها مستطیلی یا مربعی شکل هستند. راحت‌ترین پکیج برای کار DIP یا PDIP است چراکه قابل بستن روی Breadboard است و برای لحیم‌کاری راحت ترند.

نکته: از یک میکرو mid-level با پکیج DIP استفاده کنید. شما نمی‌توانید برنامه‌تان را روی یک Quad Chip بسوزانید و SOP Chipها نیز نیاز به Schmartboardها دارند.

۳. محیط توسعه مجتمع (IDE)

برای توسعه نرم‌افزار خود و سازماندهی فایل‌ها باید از یک IDE استفاده کرد. بهترین IDE برای کار با PICها MPLab IDE است که توسط Microchip Technology نوشته شده است. MPLab IDE رایگان و آسان است. برای دریافت آخرین نسخه آن می‌توانید به تارنمای <http://www.microchip.com> مراجعه کنید.

توجه: آخرین نسخه MPLab IDE با صفر ختم می‌شود. (به عنوان مثال v7.50) فایل‌هایی که با صفر تمام نمی‌شوند نسخه موقتی هستند.

همچنین می‌توانید راهنمای کاربر، نمودار سریع و راهنمای سریع کاربر MPLab IDE را دریافت کنید. پس از دریافت MPLab IDE آن را نصب کنید.

۴. زبان‌های برنامه نویسی

PICها با زبان‌های C، Assembly یا مخلوطی از این دو قابل برنامه‌نویسی است. سایر زبان‌های سطح بالا می‌توانند که به صورت پایه‌ای توسط C نوشته شده‌اند نیز قابل استفاده در Embedded Systemها هستند. در زیر سه نمونه برنامه را خواهید دید:

Example 1 – Assembly

```

MAIN
    Clrf          PORTB          ;Clear PORTB output latches
    Bsf          STATUS,RP0      ;Switch to bank 1
    Movlw       b'11110000'      ;Load value to make lower 4 bits outputs
    Movwf       TRISB           ;Move value to TRISB
    Bcf          STATUS,RP0      ;Switch to bank 0

LOOP
    Bsf          PORTB,0         ;Turn on LED on RB0
    Call         DELAY           ;Call delay routine
    Bcf          PORTB,0         ;Turn off LED on RB0
    Call         DELAY           ;Call delay routine
    Goto        LOOP            ;Repeat main loop

DELAY
    Decfsz      COUNTERL        ;Decrement COUNTERL
    Goto        DELAY           ;If not zero, keep decrementing COUNTERL
    Decfsz      COUNTERH        ;Decrement COUNTERH
    Goto        DELAY           ;If not zero, decrement COUNTERL again
    return

END

```

Example 2 – Assembly and C

```

main()
{
    short first_pass = TRUE;
    //----- Set up port direction reg's -----
    #asm
    movlw 0 // Set port b to outputs

```



```
tris port_b
clrf port_b
movlw 0xff // Set port a to inputs
tris port_a
#endasm
//----- Wait for powerup, Initialize LCD -----
delay_ms(200);
init_lcd();
//----- Write a startup message -----
msg_1();
//----- Write status message -----
msg_2();
...
```

Example 3 – C

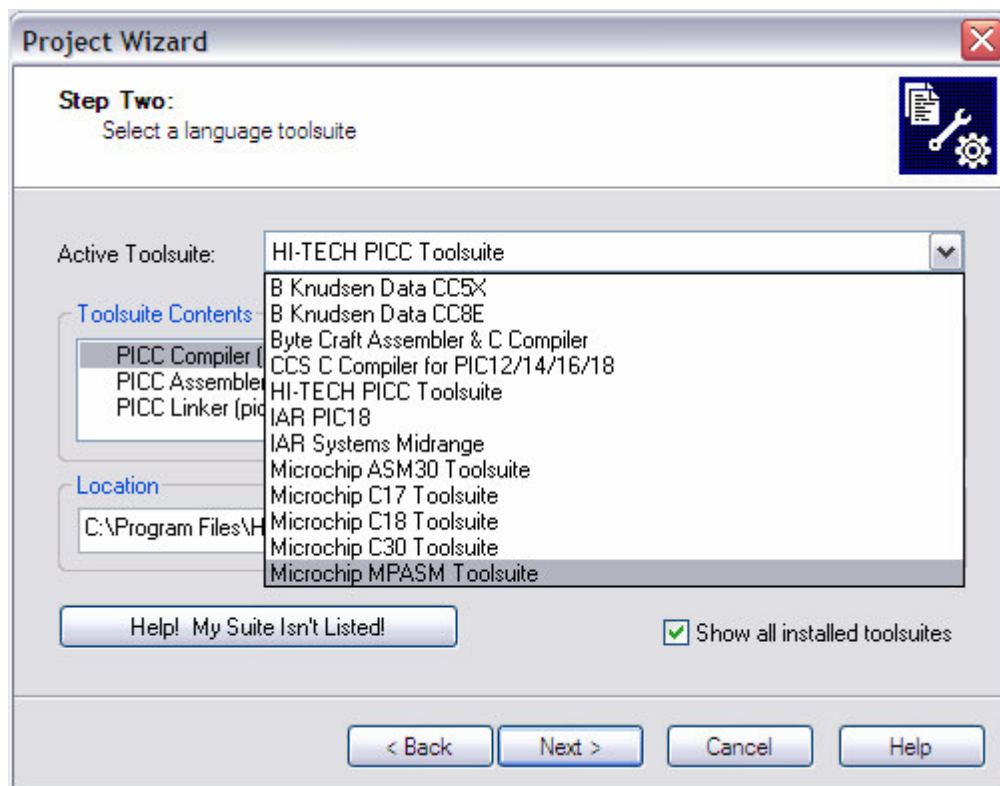
```
void main() {
U8 i = 0; // General purpose loop var.
U16 num ; // General purpose number var.
U8 row = 0; // Current display row.
U16 blinkc = 0; // LED blinker counter.
U16 blink_onoff = 1; // LED state.
U8 bcd_h,bcd_m,bcd_s; // BCD numbers.
init_ports();
init();
enable_interrupts();
ROW_RESET;
////////////////////////////////////
for (;;) { // infinite loop
// FLASH LED @ RA3
if (++blinkc>500) { // time to change state ?
blinkc=0;
...
}
```

پیشنهاد می‌کنیم که کدهای خود را به زبان C بنویسید چرا که نوشتن آن سریع‌تر و راحت‌تر است.

نکته: نسخه‌ای از زبان C که برای نوشتن کدهای خود انتخاب می‌کنید وابسته به کمپایلری است که برای پروژه خود انتخاب کرده‌اید.

۵. کمپایلر، اسمبلر و لینکر

بعد از دریافت، نصب و انتخاب زبان برنامه نویسی مورد نظرتان شما باید یک کمپایلر برگزینید. کمپایلر، اسمبلر و لینکر عموماً در یک بسته واحد قرار دارند. در MPLab IDE از طریق Project Wizard یا انتخاب منو Option Project → Select Language Toolsuite به شکل زیر توجه کنید.



شکل ۱: انتخاب کمپایلر در MPLab IDE

اغلب toolsuiteها بصورت پیش فرض نصب نیستند و گران قیمتند. به عنوان یک دانشجو احتمالاً علاقه‌مند به رایگان‌های رایگان همراه MPLab IDE (Toolsuite Microchip MPASM و CCS C Compiler for PIC12/24/26/18) هستید. سایر کمپایلرهای رایگان که می‌توانند در MPLab IDE مجتمع گردد روی وب وجود دارد.

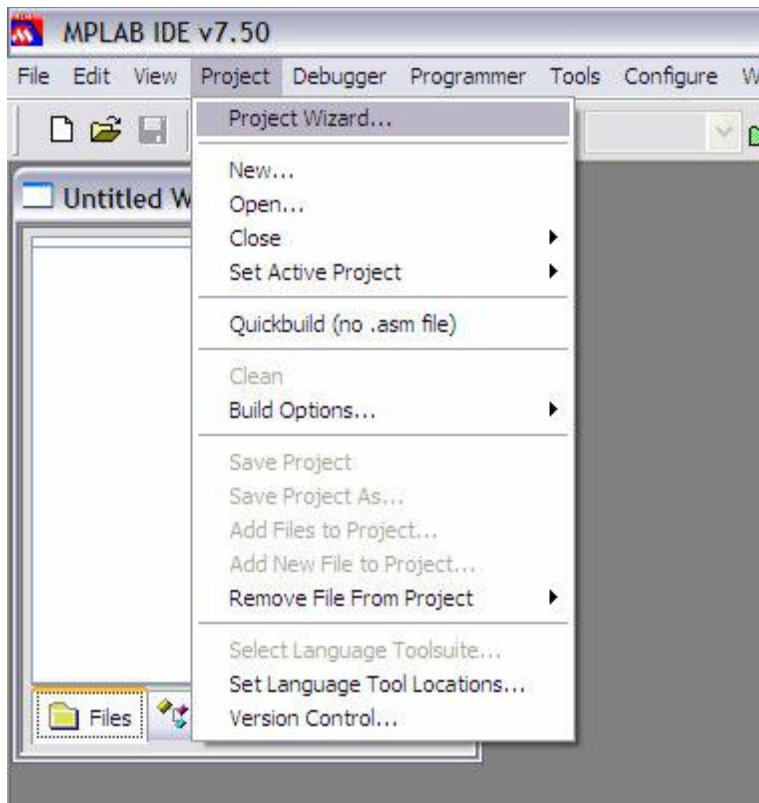
توجه: CCS C Compiler رایگان است اما با خیلی از میکروکنترلرهای PIC ناسازگار است. به دستگاه‌های پشتیبانی شده در لیست زیر توجه کنید: <http://www.ccsinfo.com/devices.php?page=devices>

به دلیل نکته ذکر شده در بالا در اینترنت به دنبال یک کمپایلر رایگان C گشتم. به کمپایلر HI-TECH PICC-Lite رسیدم که در http://htsoft.com/products/PICClite_comparison.php وجود دارد که به راحتی قابل نصب و مجتمع سازی با MPLab IDE است.

۶. استفاده از MPLab IDE

بیباید با ساخت یک پروژه جدید در MPLab IDE یک برنامه به زبان C بنویسیم. MPLab را باز کنید و Workspace و پنجره‌های خروجی را مشاهده کنید. پنجره Workspace فایل‌های پروژه را به صورتی سلسله مراتبی و راحت سازماندهی می‌کند.

برای ساخت یک پروژه جدید مسیر Project → Project Wizard را دنبال کنید.



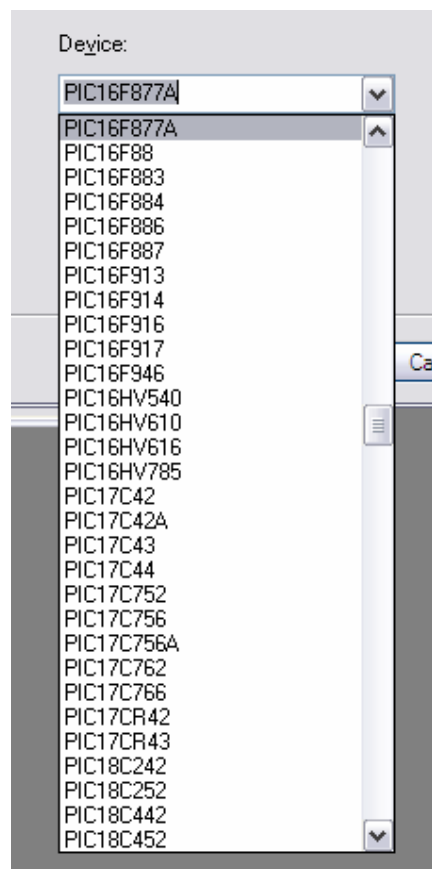
شکل ۲: ساخت یک پروژه جدید در MPLab IDE

پنجره خوش آمدگویی مطابق شکل زیر باز خواهد شد



شکل ۳: پنجره خوش آمدگویی برای ساخت پروژه جدید

دکمه Next را بزنید و سپس وارد بخش انتخاب دستگاه خواهید شد. در این پنجره تنها یک فهرست بازشدنی با تعداد زیادی میکروکنترلر PIC وجود دارد که باید دستگاه مورد نظرتان را از میان آن‌ها انتخاب کنید.

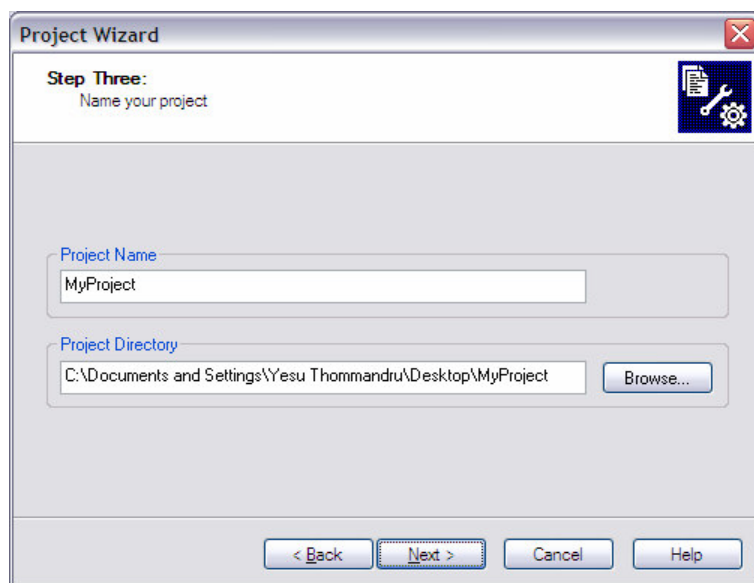


شکل ۴: انتخاب دستگاه

بعد از انتخاب دستگاه مورد نظر دکمه Next را بزنید تا وارد مرحله دوم که انتخاب زبان toolsuite است می‌شوید. کمپایلر مورد نظر خود را انتخاب کنید و دکمه Next را بزنید.

در مرحله سوم که انتخاب نام پروژه است باید نام و یک مسیر برای پروژه انتخاب کنید و دکمه Next را بزنید.

در این مثال نام MyProject و فولدر Desktop انتخاب شده‌اند:

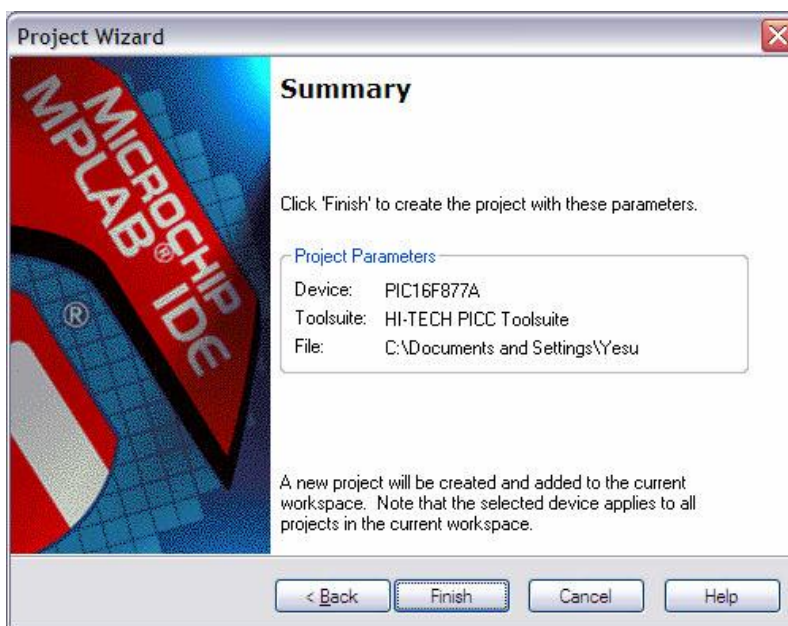


شکل ۵: نام‌گذاری پروژه

مرحله چهارم: افزودن فایل‌های موجود به پروژه به شما اجازه می‌دهد که فایل‌های مورد نیاز خود را به مسیر پروژه اضافه کنید. مثال فایل‌های مورد نیاز <pic.h>، <stdlib.h> و <string.h> هستند. هر فایل مورد نیاز را به پروژه خود اضافه کنید و دکمه Next را بزنید. (در این مثال این مرحله را رد کردیم)

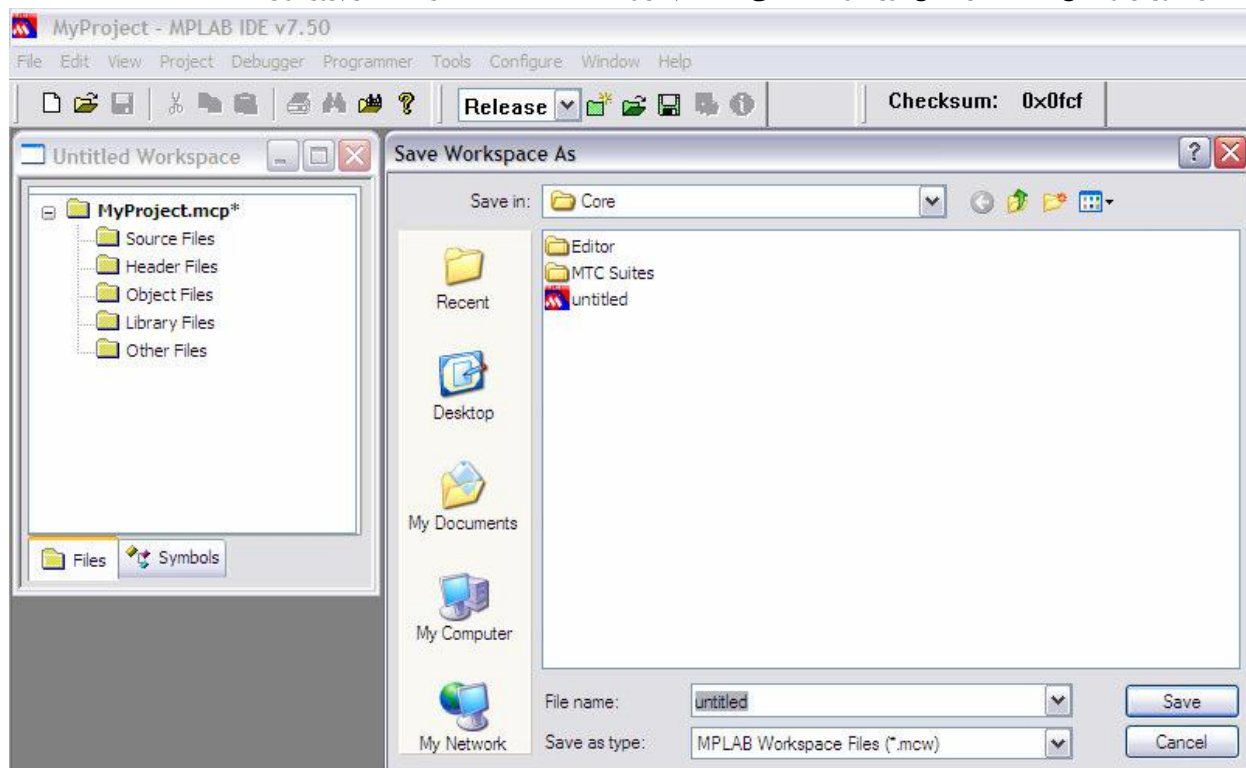
نکته: اگر از کمپایلر HI-TECH PICC-Lite استفاده می‌کنید فایل‌های مفید بسیاری مانند delay.c, delay.h, lcd.c, lcd.h در مسیر HI-TECH Software\PICC-Lite\9.50\samples وجود دارد که می‌تواند شما را در پروژه کمک کند.

آخرین پنجره در Wizard ساخت پروژه، خلاصه‌ایست از آنچه که انتخاب کرده‌اید. اگر همه چیز مناسب است دکمه Finish را فشار دهید.



شکل ۶: آخرین صفحه ساخت پروژه جدید

پس از مرحله پایانی پنجره‌ای باز می‌شود که مسیر فایل Workspace را برای ذخیره‌سازی سوال می‌کند. این فایل به کاربر اجازه می‌دهد چندین سورس و منبع مختلف را جمع‌آوری و ساماندهی کند. نام مربوط به Workspace را با مسیر پروژه وارد کنید.



شکل ۷: انتخاب مسیر Workspace

پنجره Workspace اکنون شامل نمای سلسله‌مراتبی فولدرها برای پروژه شماست.

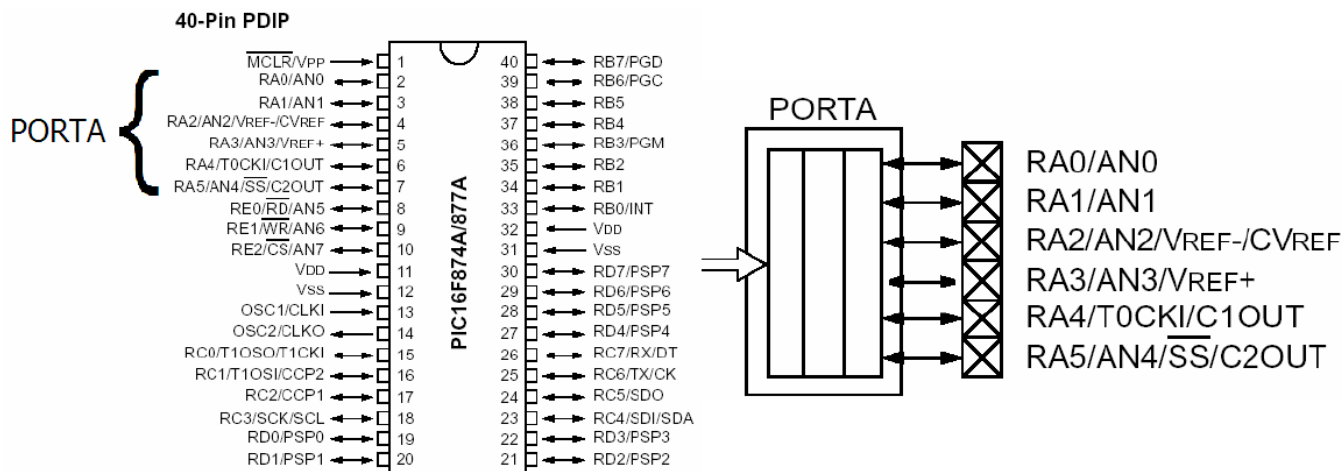
۷. نوشتن نرم افزار

اکنون یک برنامه پایه به زبان C با استفاده از MPLab IDE می نویسیم. برنامه زیر یک LED را که به PORTA از پایه های میکروکنترلر PIC وصل شده است، را چشمک می زند. (به دیتا شیت PIC نگاه کنید)

main.c

```
//LED example program written for
//PIC programming tutorial.
//standard include files
#include <stdlib.h>
#include <pic.h>
#include "delay.h"
//main function
void main()
{
    PORTA = 0x00; //set RA0-RA5 low
    TRISA = 0x00; //set PORTA to output
    //superloop
    while(1)
    {
        PORTA = !PORTA;
        DelayMs(250);
    }
}
```

برای عملیات پایه ای C و تاخیرها فایل های `stdlib.h` و `delay.h` مورد نیاز هستند. فایل `pic.h` برای دسترسی به پایه های I/O میکروکنترلرهای PIC، فضاهای حافظه و سایر اجزاء لازم است. دیگرام زیر نشان می دهد که هر پایه I/O روی PIC16F877A به کدام متغیر در برنامه نویسی مربوط است.



PORTA در `main.c` به ۶ بیت I/O روی میکروکنترلر PIC است. هر پایه می تواند با استفاده از دستورات زیر پایین (صفر) یا بالا (یک) شود. `PORTA=0x01` پایه RA0 را یک می کند. چند پایه می توانند مانند دستور `PORTB=0xFF` مقدار دهی شود. برای اطلاع از اندازه و تعداد درگاهها (پورتها) به دیتا شیت میکروکنترلر PIC مراجعه کنید.

TRISA یک رجیستر کنترلی جهت مربوط به PORTA است. TRIS تعداد بیت‌هایی برابر با درگاه مربوطه دارد. اگر در TRIS یک بیت ۱ قرار داده شود به معنای آن است که این پایه ورودی خواهد بود و صفر بودن آن به معنای خروجی بودن پایه است.

نکته: معمولا اولین تابعی که در قسمت main برنامه صدا زده می‌شود تابعی برای تنظیمات اولیه درگاه‌ها است.

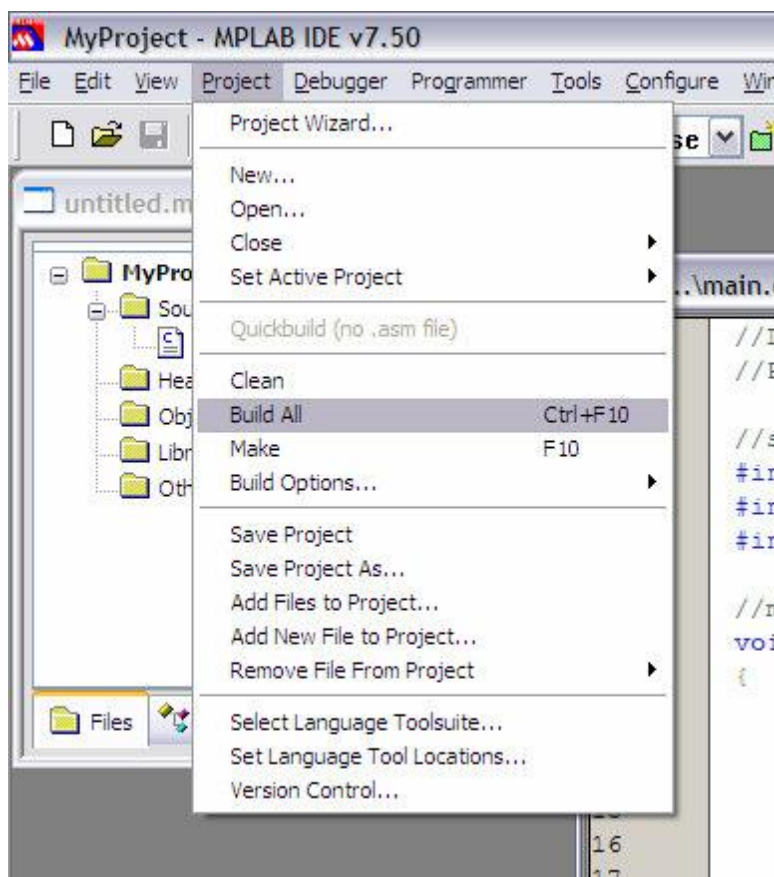
سپس برنامه وارد یک حلقه اصلی می‌شود و مقدار درگاه A را تغییر داده و ۲۵۰ میلی ثانیه تاخیر ایجاد می‌کند. ساختار، هدف و پیچیدگی نرم‌افزار شما وابسته به کارکرد میکرو شما است. یک منبع مناسب برای برنامه نویسی C برای میکروکنترلر ها به آدرس زیر مراجعه نمایید.

<http://www.best-microcontroller-projects.com/programming-microcontrollers-in-c.html>

توجه: توابع استاندارد C مانند printf() و scanf() در برنامه‌های embedded بی معناست. برای آزمایش کارکرد ورودی/خروجی باید PIC روی Breadboard قرار گرفته باشد(مدار بسته شده باشد)

۸. فرایند Burning

هنگامی که برنامه خود را نوشتید باید ابتدا آن را کمپایل کنید تا مطمئن شوید که Syntscs آن ایرادی نداشته باشد. اولین گام برای Burn کردن برنامه Build کردن (ساختن) برنامه است. قبل از ساختن پروژه خود اطمینان حاصل کنید که بیت‌های پیکربندی به درستی اختصاص داده شده‌اند. برای این کار مسیر Configure→Configuration Bits را انتخاب کنید. در پایان Project→Build All را انتخاب کنید و یا Ctrl+F10 را بزنید.



شکل ۸: Build کردن پروژه

پنجره خروجی نتایج هر مرحله را در فرایند ساختن چاپ می کند. ممکن است چند Warning (اخطار) یا advisory messages (پیغام-های مشورتی) دریافت کنید. اگر فرایند ساخت موفقیت آمیز باشد صفحه خروجی نقشه حافظه استفاده شده را نمایش خواهد داد که چیزی مشابه زیر خواهد بود:

Memory Usage Map:

Program space:

CODE used 21h (33) of 800h words (1.6%)
 CONST used 0h (0) of 800h words (0.0%)
 ENTRY used 0h (0) of 800h words (0.0%)
 STRING used 0h (0) of 800h words (0.0%)

Data space:

BANK0 used 3h (3) of 60h bytes (3.1%)
 BANK1 used 0h (0) of 50h bytes (0.0%)
 COMBANK used 0h (0) of 10h bytes (0.0%)

EEPROM space:

EEDATA used 0h (0) of 100h bytes (0.0%)

ID Location space:

IDLOC used 0h (0) of 4h bytes (0.0%)

Configuration bits:

CONFIG used 0h (0) of 1h word (0.0%)

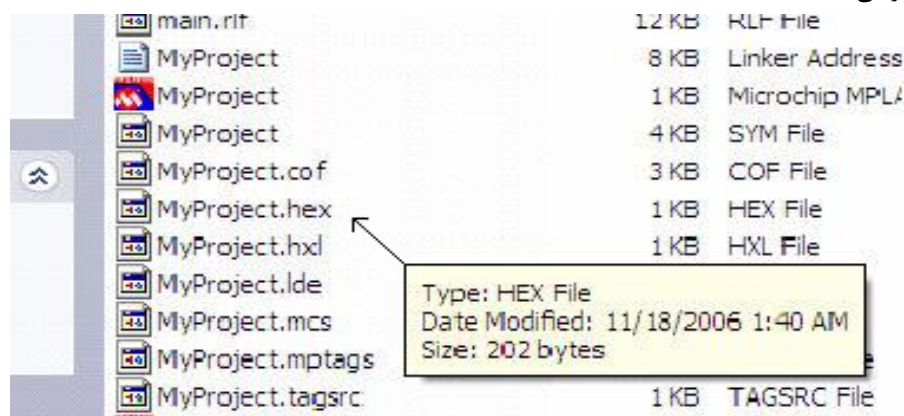
Summary:

Program space used 21h (33) of 800h words (1.6%)
 Data space used 3h (3) of B0h bytes (1.7%)
 EEPROM space used 0h (0) of 100h bytes (0.0%)
 ID Location space used 0h (0) of 4h bytes (0.0%)
 Configuration bits used 0h (0) of 1h word (0.0%)

Loaded C:\MyProject\MyProject.cof.

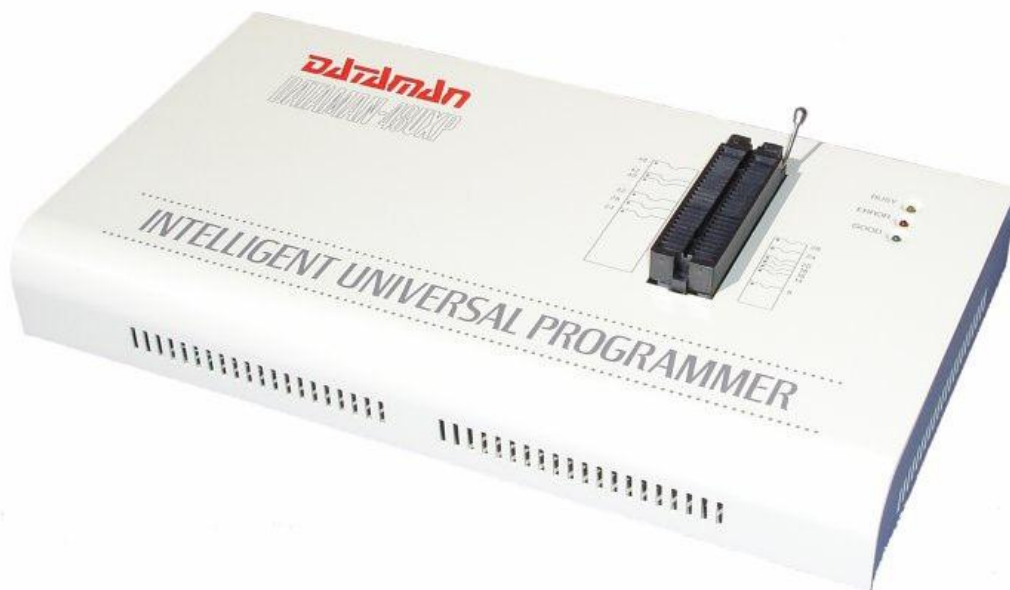
BUILD SUCCEEDED: Sat Nov 18 01:34:23 2006

بعد از ساختن برنامه خود، تعداد زیادی فایل در مسیر پروژه تان تولید خواهد شد. مهمترین فایلی که تولید می شود فایل Hexadecimal است که در تصویر زیر می بینید.



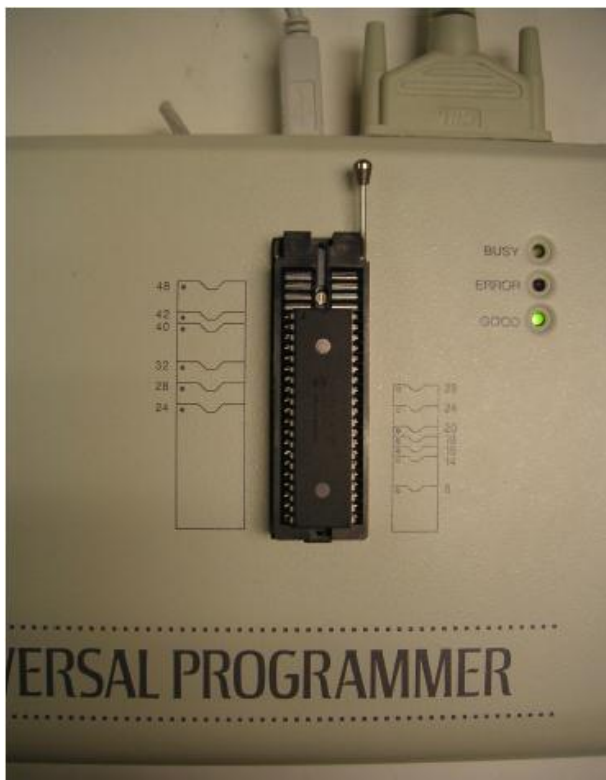
شکل ۹: فایل های تولید شده بعد از Build

این فایل است که روی PIC شما Burn خواهد شد. این فایل را روی کامپیوتری که به Programmer شما وصل است کپی کنید.



شکل ۱۰ پروگرامر DataMan برای پروگرام کردن PIC

میکرو خود را در سوکت ZIF سیاه رنگ قرار دهید و دسته نقره‌ای را در وضعیت پایین قرار دهید تا پایه‌های I/O درگیر شود.



شکل ۱۱: PIC بر روی پروگرامر

کار با نرم افزار DataMan Programmer را با مسیر زیر آغاز کنید:

Start → All Programs → Dataman Programmers → Dataman-48XP

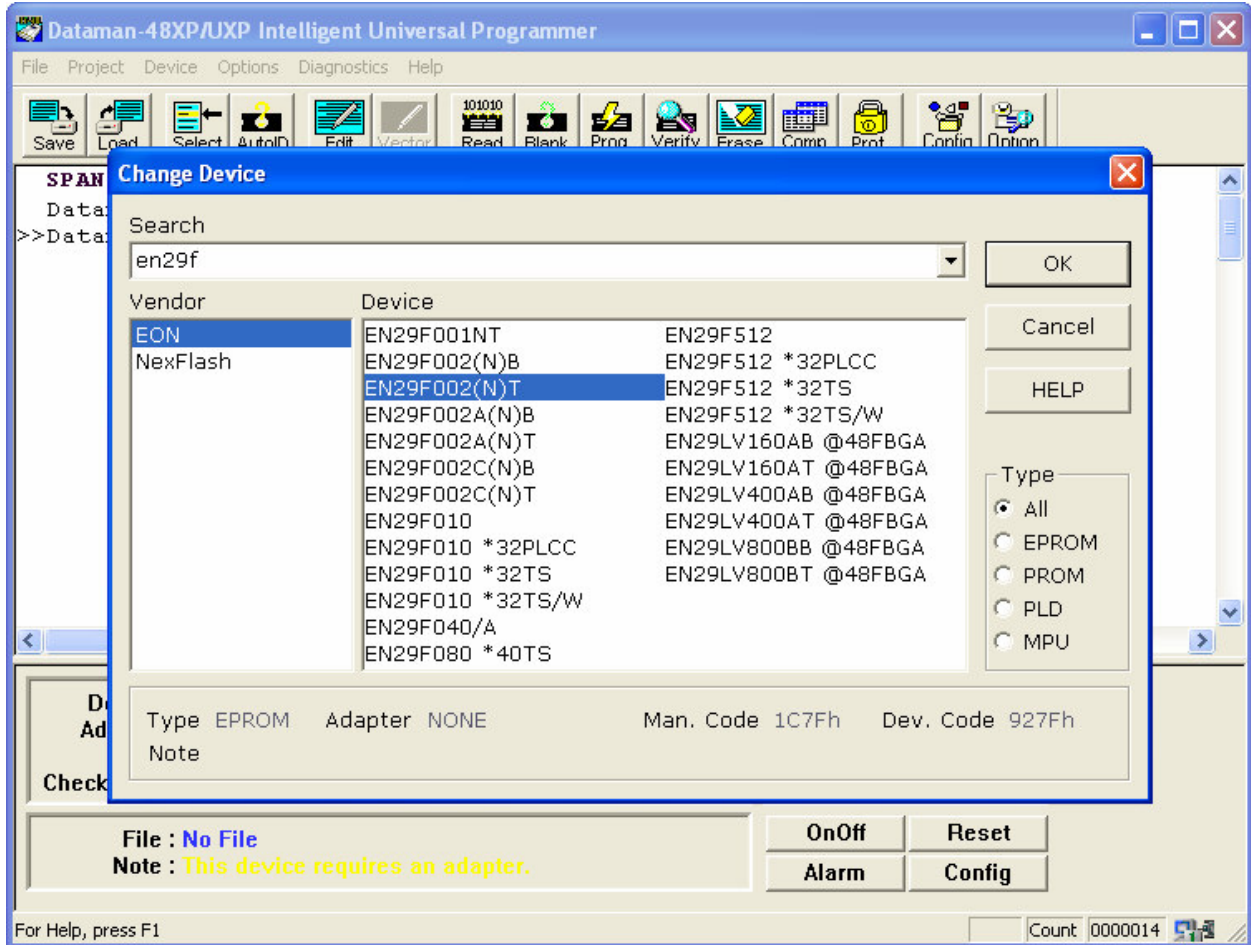
برای فرایند Burn سه مرحله اصلی وجود دارد:

انتخاب دستگاه

بارگذاری فایل HEX

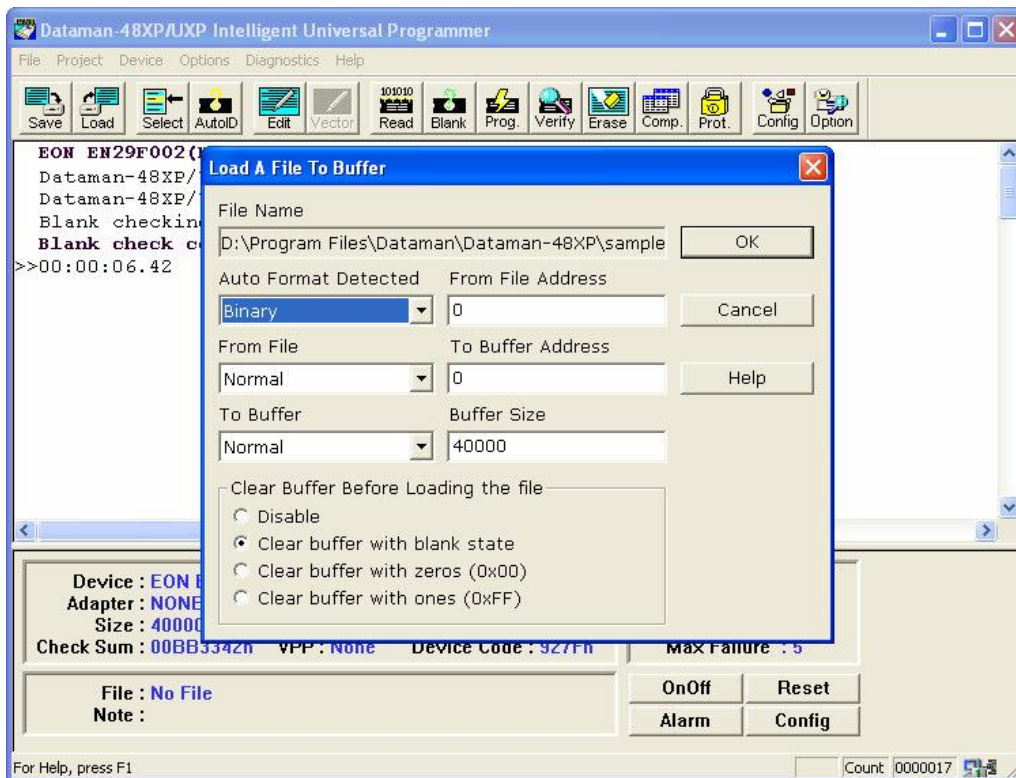
برنامه ریزی PIC

مسیر Device → Select Device را از منو انتخاب کنید و یا Alt+C را برای انتخاب PIC بزنید. فهرست چندین میکرو ظاهر می شود. PIC مورد نظر را انتخاب کند و OK را بزنید.



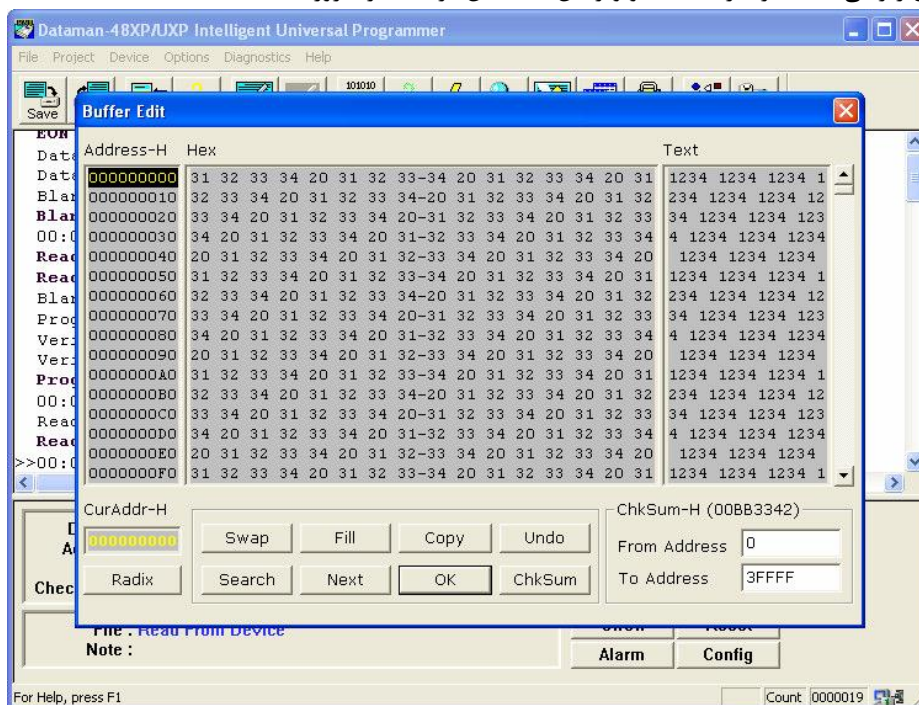
شکل ۱۲: انتخاب دستگاه در نرم افزار Dataman-48XP/UXP

مسیر File → Load File را انتخاب کنید یا دکمه Load را بزنید و یا Alt+L را فشار دهید تا فایل Hex پروژه خود را بارگذاری کنید. نرم افزار DataMan باید به صورت خودکار فایل را به عنوان یک فایل Intel Hex شناسایی کند. مطمئن شوید که یکی از Radio Button های Clear Buffer انتخاب شده است و سپس OK را فشار دهید.



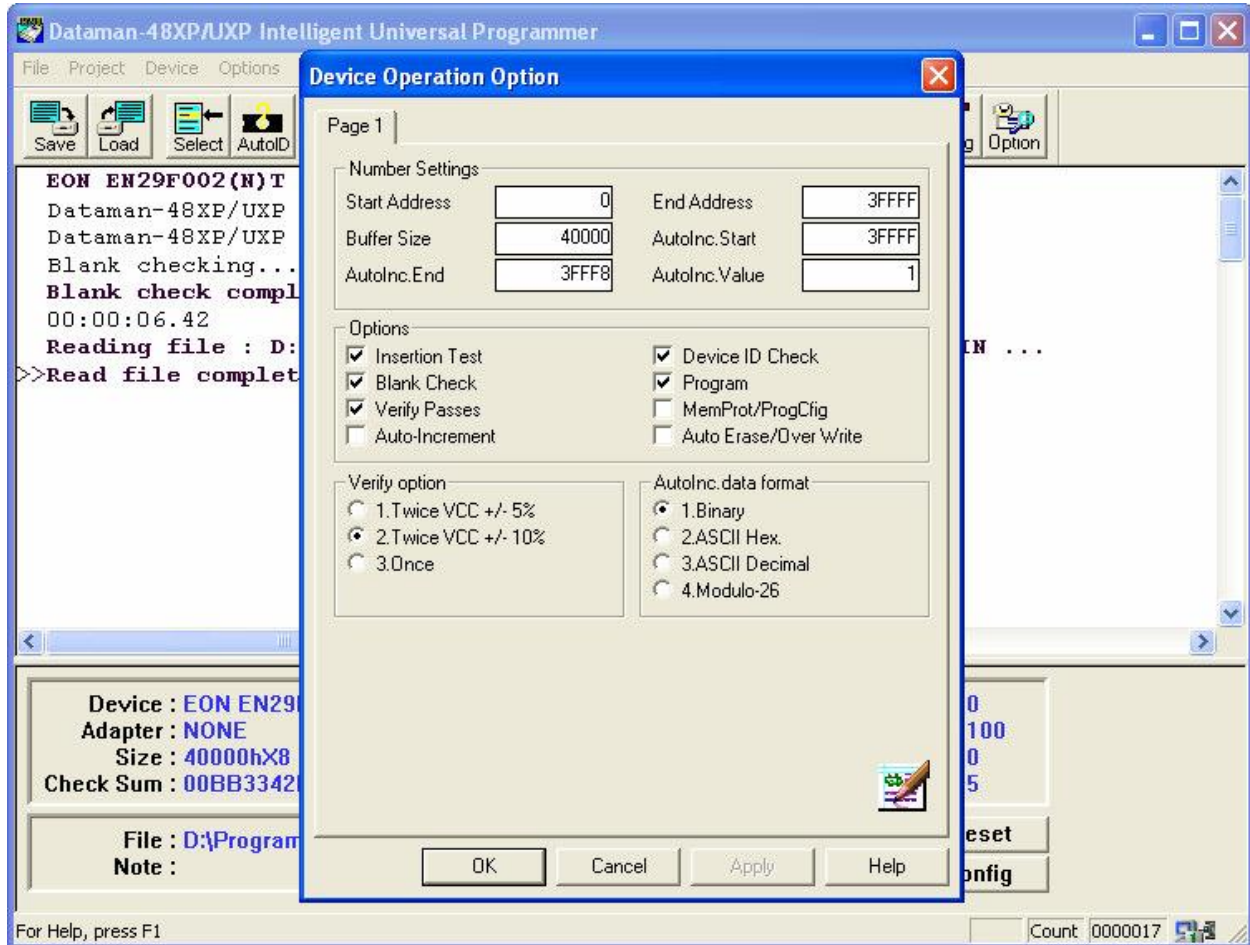
شکل ۱۳: بارگذاری فایل Hex

اکنون فایل Hex را روی بافرهای DataMan بارگذاری کردید اما هنوز PIC را پروگرام نکرده‌اید. نرم افزار DataMan به شما اجازه می‌دهد که همانطور در شکل زیر می‌بینید بافرها را مشاهده و ویرایش کنید. این مرحله غیر ضروری است.



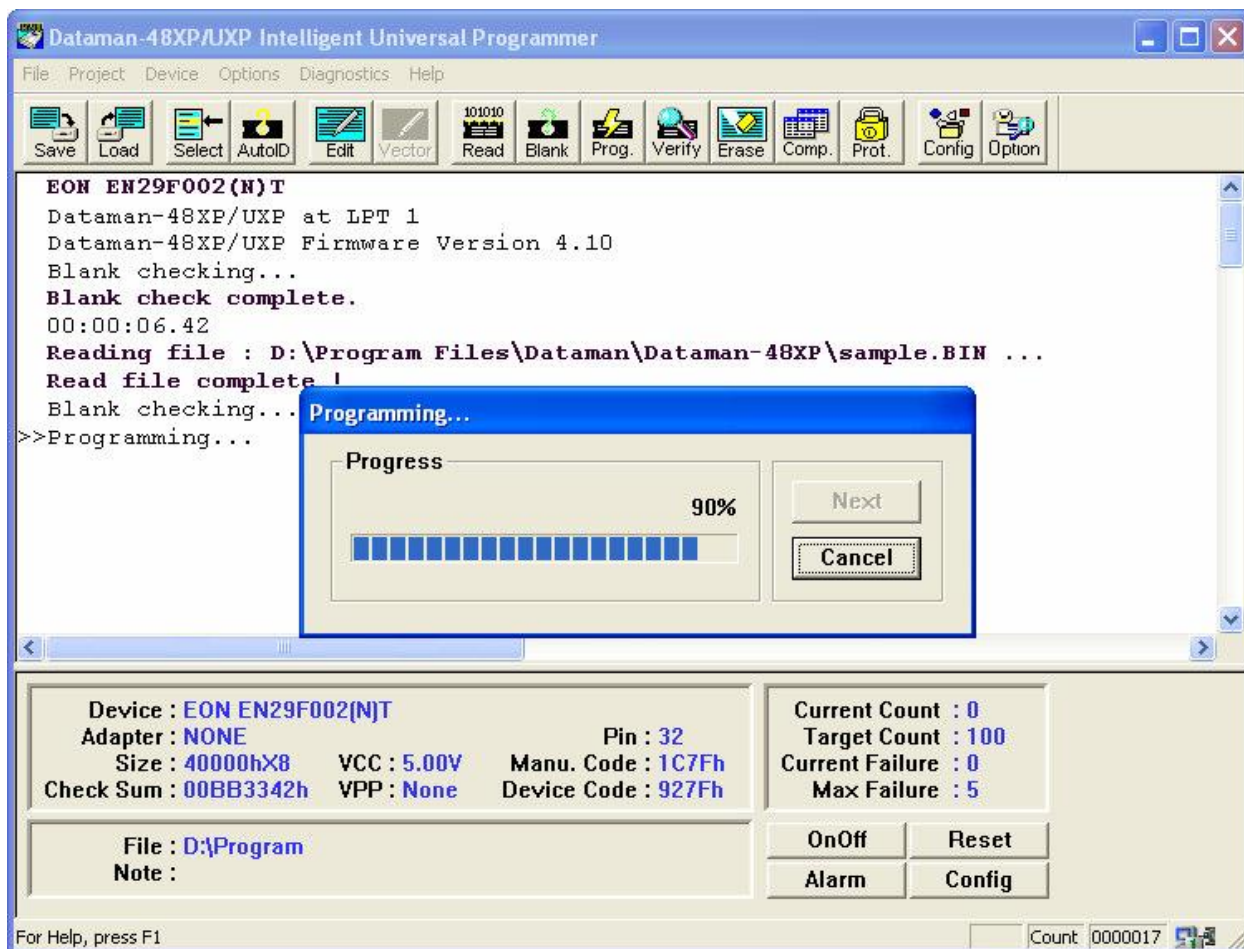
شکل ۱۴: ویرایش بافر

قبل از پروگرام کردن PIC می‌توانید ویژگی‌های خاص دستگاه را در نرم افزار DataMan با فشار دادن دکمه Config در منو تعیین کنید. پنجره‌ای با بخش‌های مختلف از CheckBoxها و RadioButtonها برای یک دستگاه خاص ظاهر خواهد شد. برای درک دقیق این بیت-های پیکر بندی دیتا شیت میکرو مورد نظر خود را مطالعه کنید. از انتخاب‌های خود اطمینان حاصل کنید و سپس دکمه OK را فشار دهید.



شکل ۱۵: پیکربندی میکرو

اکنون وقت آن فرا رسیده که میکرو را پروگرام کنید. دستگاه مورد نظر انتخاب شده، فایل Hex بارگذاری شده و بیت‌های پیکربندی مقدردهی شده‌اند. مسیر Device→Program→Auto را انتخاب کنید و یا Alt+P را بزنید. یک میله پیشرفت (Progress Bar) در مرکز تصویر ظاهر می‌شود.



شکل ۱۶: پروگرام کردن

اگر فرایند Burn کاملاً موفقیت آمیز باشد پیام Programming Complete بر روی صفحه چاپ خواهد شد. کارکردهای بسیار دیگری توسط این نرم افزار فراهم آورده شده است. برای اطلاعات بیشتر توضیحات این نرم افزار را در آدرس زیر بخوانید:

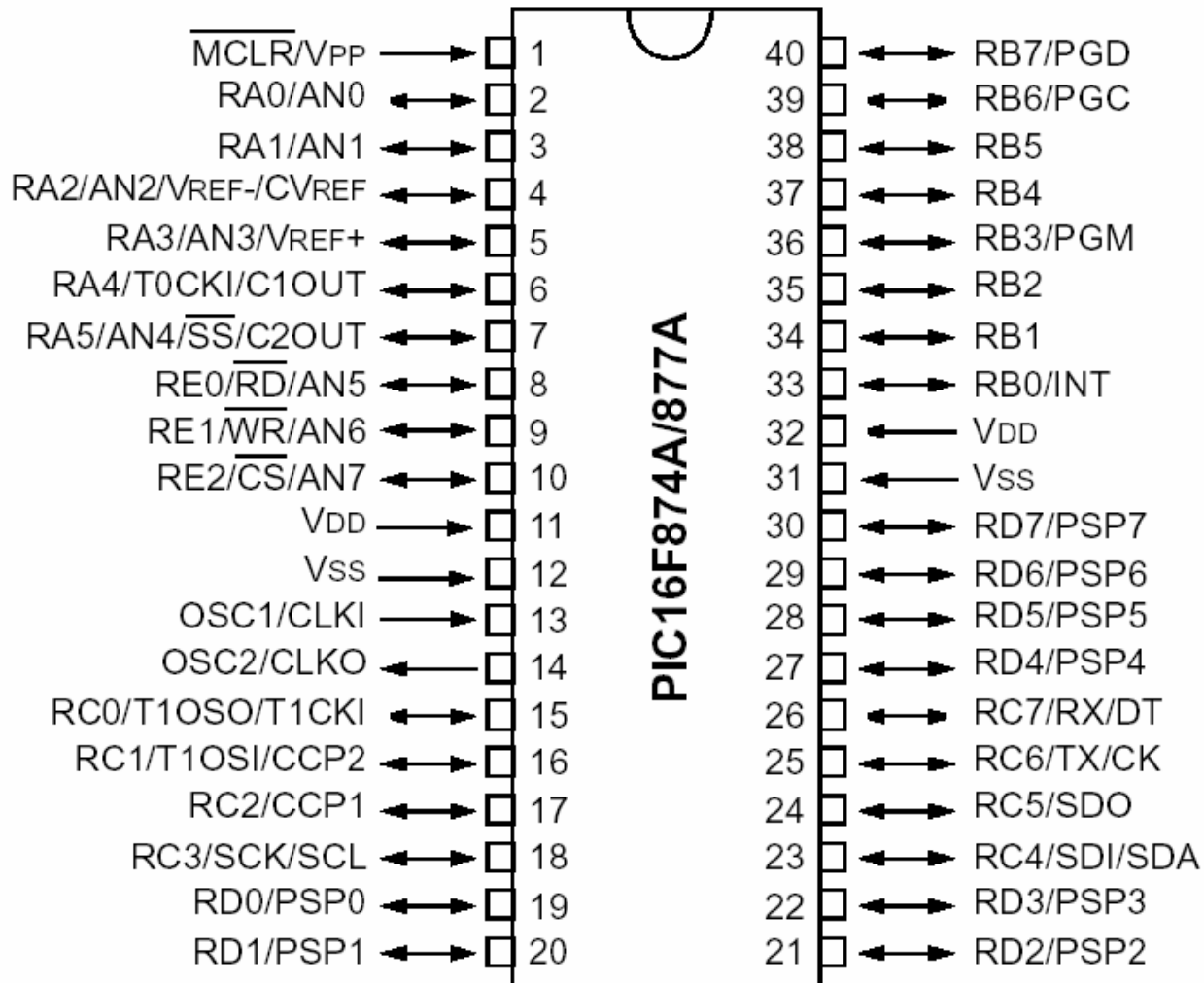
<http://dataman.com/Webpages/Programmers/Product48UXPInformation.aspx>

توجه: در هنگام پروگرام کردن مجدد ممکن است با پیغام خطای Poor contact at pin 13 یا Over current detected مواجه شوید. این بدان معناست که شما PIC خود را خراب کرده‌اید.

۹. بستن مدار PIC بر روی Breadboard

در هنگام بستن مدار علاوه بر اینکه به راحتی PIC ممکن است خراب شود باید به اتصال پایه‌های اجباری توجه داشت. بدین سبب به دیتا شیت میکرو توجه کنید.

به عنوان مثال برای استفاده از PIC16F877A در پروژه LED چشمک زن در پروژه MyProject شرح داده شده است. شکل ۱۷ نمودار پایه‌های PIC16F877A را نشان می‌دهد.

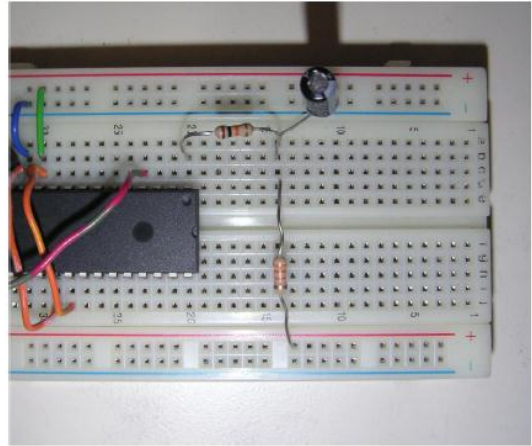
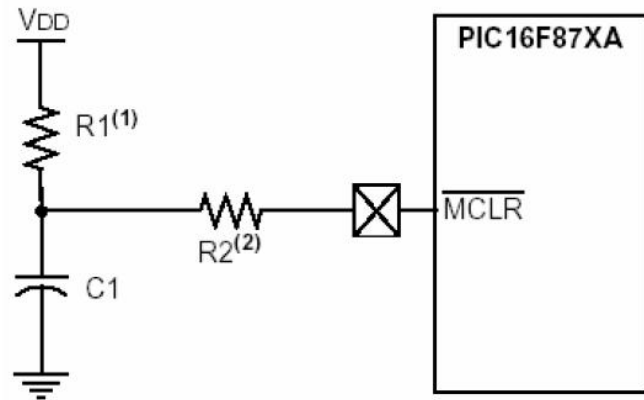


شکل ۱۷: دیاگرام پایه‌های PIC16F874A/877A

با توجه به اینکه منبع ولتاژ، پایه‌های ۱، ۱۱، ۱۲، ۳۲ و ۳۳ همیشه باید وصل باشد. Vdd منبع ولتاژ مثبت و Vss زمین است. پایه MCLR/Vpp پایه ویژه‌ایست که PIC را در وضعیت Reset نگه می‌دارد تا زمانی که منبع ولتاژ مناسب یافت شود. این پایه به طریق ویژه‌ای توسط دو مقاومت و یک خازن بسته می‌شود. (شکل ۱۸)

مقادیر مقاومت‌ها و خازن‌های شکل ۱۸ در جدول زیر آمده‌است:

مقدار	قطعه
۳۳KΩ	R1
۱۰KΩ	R2
۰.۱μF	C1



شکل ۱۸: بستن پایه MCLR/Vpp

برای مقادیر دقیق تر بخش ۱۴.۰ از Special Features of CPU در دیتا شیت میکرو را مطالعه کنید. مرحله بعدی در بستن مدار ساخت کلاک مورد نیاز برنامه است. چون اغلب PICها دارای کلاک داخلی نیستند کلاک خارجی مورد نیاز است. پایه ۱۳ OSC1 باید برای اجرای برنامه وصل شده باشد. برای فاز آزمایشی یک موج مربعی از یک Arbitrary Waveform Generator می توان به عنوان منبع اوسیلاتور استفاده کرد. فرکانس مناسب بسته به PIC و نیازهای نرم افزار است.

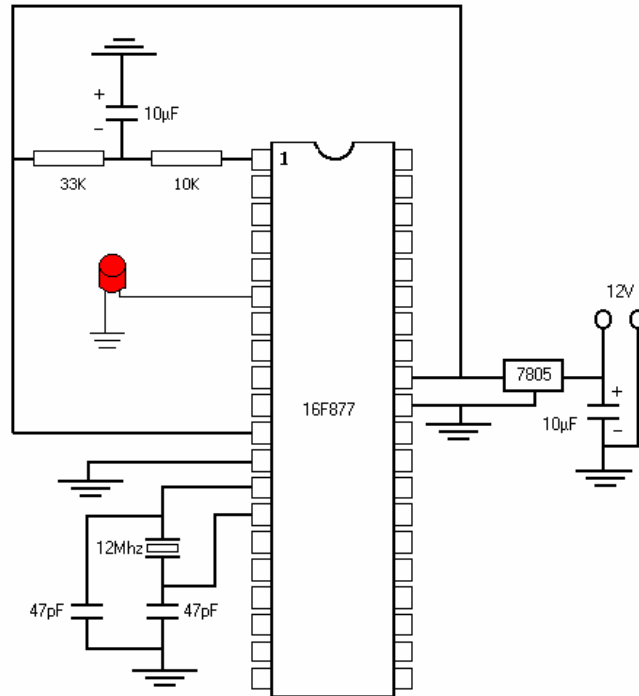


شکل ۱۹: اسیلاتور

شکل ۲۰: تراشه تایمر ۵۵۵

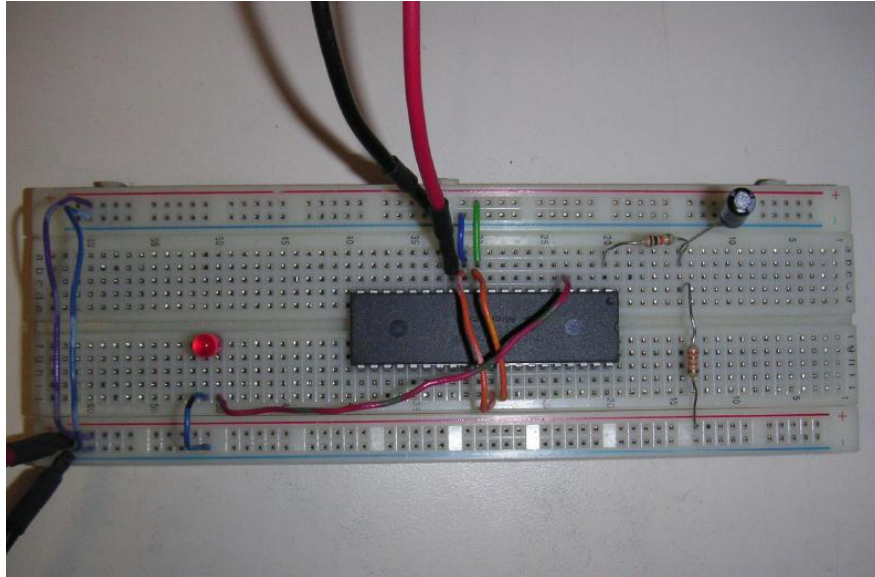
هنگامی که قطعات برای لحیم کاری روی PCB آماده شد یک تراشه Timer مورد نیاز است. یک تایمر ۵۵۵ ارزان قیمت کار ما را راه خواهد انداخت.

دیگرام زیر شماتیک مدار اصلی برای بستن یک میکروکنترلر جهت اجرای برنامه LED چشمک زن است. شما می توانید تراشه ۷۸۰۵ را حذف کرده و از ولتاژ ۵V استفاده کنید. همانطور که می بینید LED قرمز رنگ به یکی از پایه های درگاه A متصل شده است. LED با توجه به فرکانسی که شما انتخاب می کنید چشمک خواهد زد.



شکل ۲۱: مدار پروژه LED چشمک زن

مدار شما در نهایت مانند شکل زیر خواهد شد.



شکل ۲۲: نمای Breadboard پروژه LED چشمک زن

برای اطلاعات بیشتر در مورد بستن مدار روی Breadboard دیتا شیت PIC را مطالعه کنید و سری هم به لینک آموزش اولین پروژه با 16F877 در آدرس <http://members.home.nl/b.vandam/lonely/pagina000.html> بزنید.