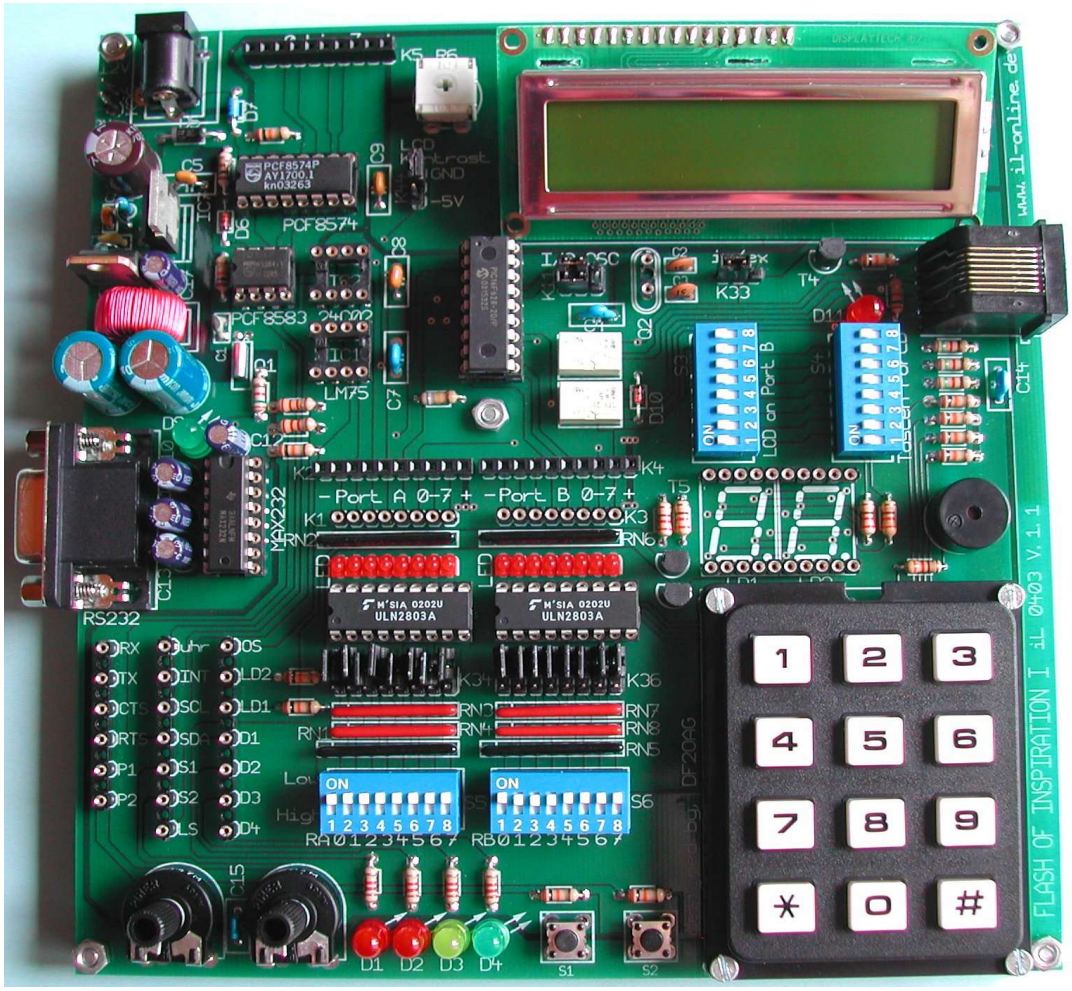
Erste Schritte mit Flash Of Inspiration I

(von der Idee zum fertigen Programm)



Stand 05.04.2005

Nichts ist vergleichbar mit den ersten bewußten Schritten, die man auf unbekannten Terrain macht. Allerdings bereitet der Einstieg in ein neues Thema meistens erhebliche Probleme, da vieles noch unbekannt ist und fremdartig erscheint.

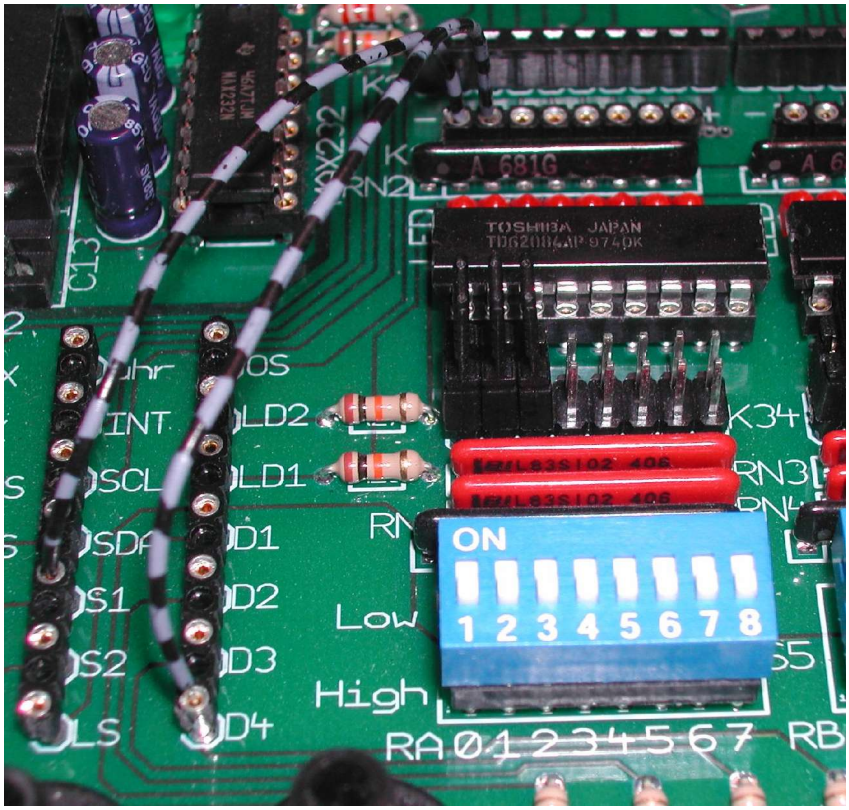
Es wird vorausgesetzt, dass:
Flash of Inspiration liegt fertig aufgebaut vor uns. Die Software und das USB-Programmiergerät wurden erfolgreich installiert.

Das erste kleine Projekt kann beginnen.
Nach dem Drücken des Tasters S1 beginnt die LED D4 für 30 Sekunden zu blinken. Danach kann die Prozedur durch Drücken des Tasters S1 erneut ausgelöst werden.

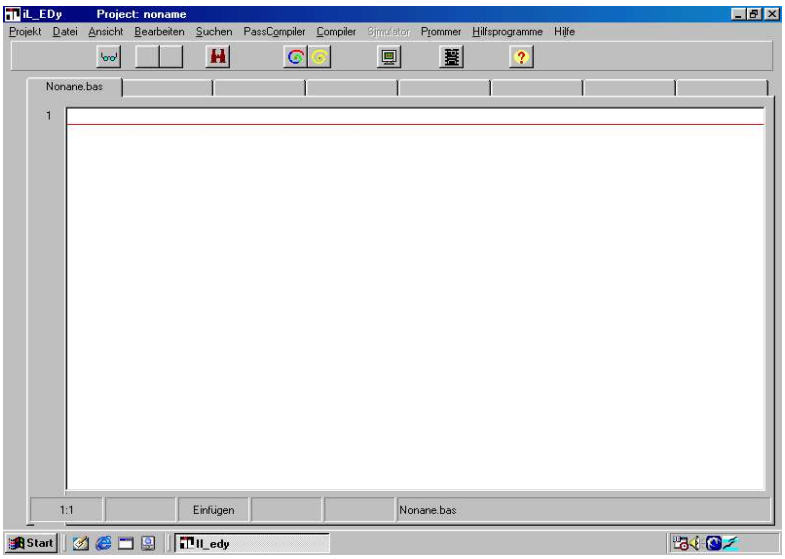
Auf dem Board von „Flash of Inspiration I“ sind folgende Einstellungen und Verbindungen vor zu nehmen:

K11 „I/O OSC“ in Stellung „I/O“
K33 „in/ex“ in Stellungung „in“
S3 alle Schiebeschalter auf „off“
S4 alle Schiebeschalter auf „off“
S5 alle Schiebeschalter auf „off“
S6 alle Schiebeschalter auf „off“

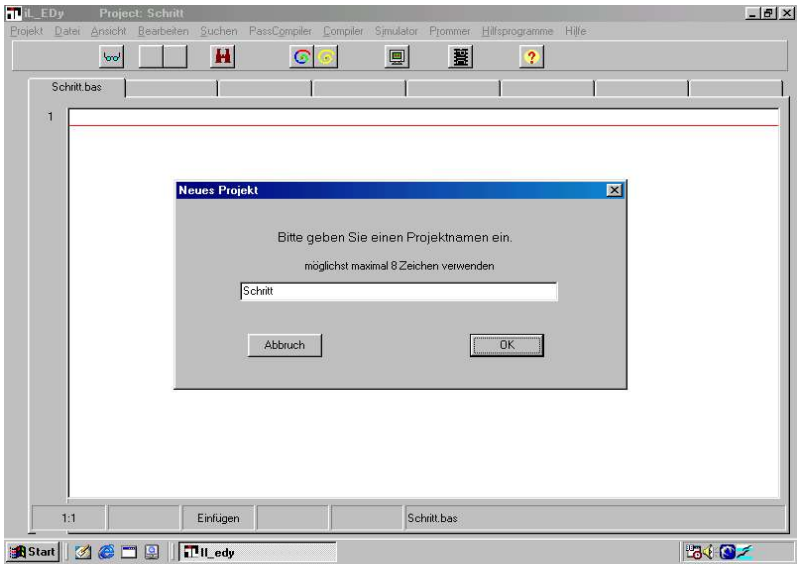
Drahtverbindungen:
Schaltdraht verbinden von „S1“ im Schaltfeld mit K1 Pin ganz links (Port A, RA,0)
Schaltdraht verbinden von „D4“ im Schaltfeld mit K1 Pin 2 von links (Port A, RA,1)



Das Programm iL_BAS-SES starten. Nach Vergabe eines Projekt Namens kann mit dem Programmieren begonnen werden.



Links oben Projekt anklicken, dann „neu“ auswählen und einen Projektnamen eingeben.
In unserem Beispiel „Schritt“



In das aktuelle Fenster des Editors wird der Programmcode geschrieben. Der Editor erleichert die Eingabe des Programms, da er korrekt eingegebene Basic-Befehle rot markiert und gültige Parameter grün einfärbt. Das Programm sieht folgender Maßen aus:
Der rot markierte Text dient nur zur Erläuterung, bitte nicht in den Editor eingeben.

schwarze Schrift bedeutet,	dies ist der Programmcode.
grüne Schrift bedeutet,	dies ist ein Kommentar zum Programmcode. Kann in den Programmcode eingefügt werden.
rote Schrift bedeutet,	dies ist eine Erläuterung, darf nicht im Programmcode aufgenommen werden. Führt beim Compilieren zu Fehlermeldungen.

_____Hier beginnt das Programm „schritt.bas“_____

```
rem      Programm erste Schritte mit Flash of Inspiration I
rem
        define device 16f628 wdt_off,irc_osc,cmcfg7,mclr_int,osc2_io
```

Mit „define device“ erhält der Compiler die Anweisung, für welchen PIC-Prozessor der Code erstellt werden muss und mit welcher Konfiguration er betrieben wird.
In unserem Fall:
16f628 = PIC 16F628,
wdt_off = Watchdog Timer aus,
irc_osc = interner RC Oszillator eingeschaltet (es wird kein Quarz benötigt),
cmcfg7 = Port RA hat nur digitale I/O,
mclr_int = Reset beim Einschalten wird intern erzeugt ein I/O Pin mehr,
osc2_io = Der PIN OSC2 steht als I/O Pin zur Verfügung

```
        tris ra,%00000001   rem RA,0 = Eingang; alle anderen PINS Ausgänge
        tris rb,0           rem RB   = Alle PINS Ausgänge
```

Mit der „tris“ Anweisung legt man den I/O-Pin auf Eingang oder Ausgang fest.
Eine 0 bestimmt den betreffenden Pin als Ausgangspin.
Eine 1 bestimmt den betreffenden Pin als Eingangspin.
Mit dem „%“ Zeichen kann jeder Pin sichtbar (Binär) eingestellt werden.
Die Zählweise ist: % 0 0 0 0 0 0 0 1
Portpin: 7 6 5 4 3 2 1 0
Mit einer Dezimalzahl (0-255) wird entprechend des Binärwertes der Pin als Eingang oder Ausgang geschaltet. Die Ziffer Null bedeutet alles Ausgänge. Die Ziffer 5 hat zum Beispiel den Binärwert %00000101. Dies bedeutet: Pin 0 und Pin 2 sind Eingänge, die restlichen Pin sind Ausgänge.

```
define taste = ra,0 rem Tastendruck wird abgefragt am PIN RA,0
define LED  = ra,1 rem LED leuchtet, wenn RA,1 high
```

„define“ weist in diesem Fall einem Portpin einem Namen zu.
Erleichtert das Lesen des Programms.

```
define zaehler= $20 rem Variable Zaehler in Speicherzelle 20hex des PIC
```

„define“ reserviert die Speicherzelle 20hex für die Variable „Zaehler“
Die für den jeweiligen zur Verfügung stehenden Speicherbereiche erfahren
Sie im Handbuch zum Programm iL_BAS-SES.

```
output ra,0         rem Alle Ausgangspins des RA auf Low= 0 Volt
```

Die Anweisung „output“ weist den nach dem Komma stehen Dezimalwert (0-255) dem vor dem Komma stehen Port zu. In diesem Beispiel werden alle Pins des Port A (RA) auf Low gesetzt.

```
start:              rem Einsprungpunkt = Label; Programmstart
```

Der Label „start:“ erlaubt einen gezielten Sprung zu den nachfolgenden Programmzeilen. Der Doppelpunkt nach dem Label ist wichtig. In der Sprung-Anweisung „goto start“ darf kein Doppelpunkt verwendet werden.

```
if taste = 1 then goto start  rem Wenn Taste NICHT gedrückt, gehe zu Start
```

Bei der „if“-Abfrage wird der Logikpegel von „Taste“ (ra,0) mit dem vorgegeben Wert (in diesem Fall 1=high) verglichen. Im Ruhezustand, Taste nicht gedrückt, liegt an ra,0 High-Pegel. Low-Pegel liegt nach Druck auf die Taste an.
In diesem Fall wartet das Programm, bis die Taste S1 gedrückt wird.

```
For zaehler=0 to 29 rem Programmschleife 30 mal durchfahren
```

Mit der Anweisung „For“ beginnt eine „For-Next-Schleife“. Diese Programmschleife führt die Programmzeilen zwischen „For“ und „Next“ beliebig oft aus. In diesem Fall werden die Programmzeilen 30 mal ausgeführt. Der Variablen „zaehler“ wird zu Beginn der Wert „0“ zugewiesen. Bei jedem Durchgang erhöht sich der Wert der Variablen um 1. Erreicht die Variable den vorgegeben Wert, in diesem Fall 29, wird die „For-Next-Schleife“ beendet und der Programmcode nach der „Next“ Anweisung fortgesetzt.

```
set LED             rem LED einschalten
```

„Set“ setzt den Portpin ra,1 auf Highpegel. Die LED D4 beginnt zu leuchten.

wait 500 rem warte 500ms

Die Programmausführung wird für 500ms angehalten. Die LED leuchtet für diese Zeit.

res LED rem LED ausschalten

„Res“ setzt den Portpin ra,1 auf Lowpegel. Die LED D4 leuchtet nicht mehr.

wait 500 rem warte 500ms

Die Programmausführung wird für 500ms angehalten. Die LED bleibt während Zeit nicht.

next zaehler rem Programmschleife wiederholen

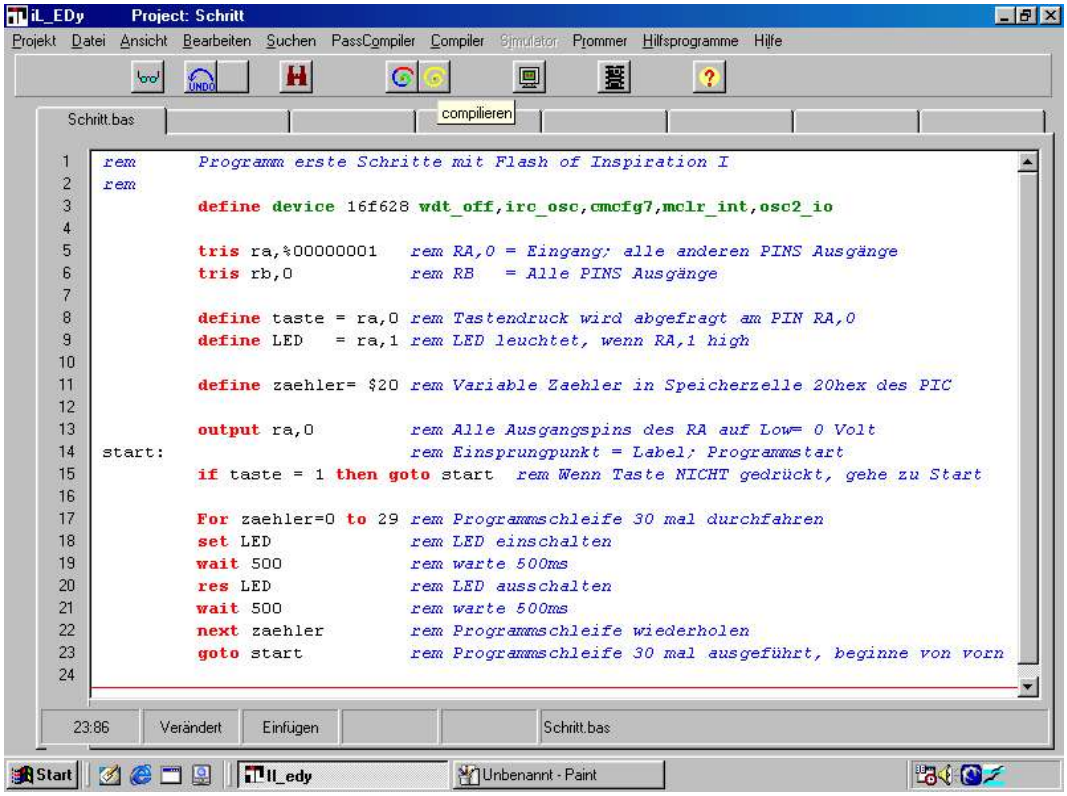
Die „For-Next-Schleife“ ist hier zu Ende. Der nachfolgende Befehl wird erst nach dem 30. Durchlauf der Schleife ausgeführt.

goto start rem Programmschleife 30 mal ausgeführt, beginne von vorn

„Goto“ ist ein unbedingter Sprungbefehl. Er wird immer ausgeführt. Das Programm beginnt von vorn.

_____hier endet das Programm „schritt.bas“_____

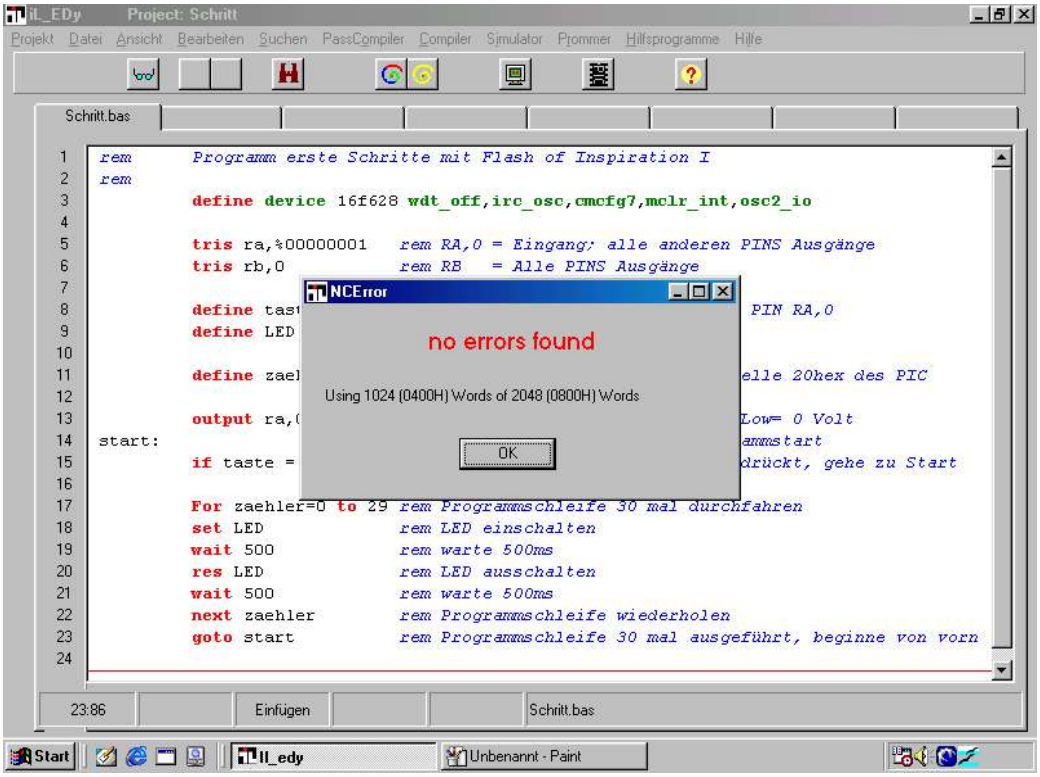
Das Basic-Programm ist nun komplett. Der nächste Schritt ist der Compilerlauf. Dazu bitte die „gelbe Schnecke“ anklicken.



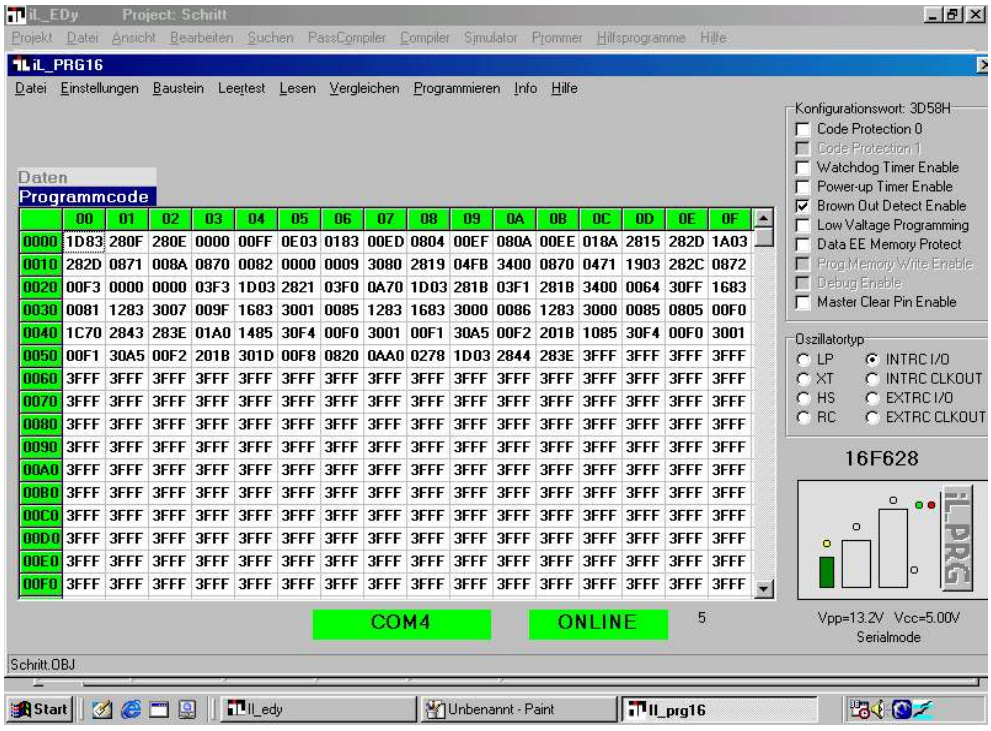
Nach dem Compilerlauf gibt es im Prinzip zwei Möglichkeiten:

1. Der Compiler läuft ohne Probleme durch, da der Programmcode ohne Syntaxfehler geschrieben wurde. Dies ist hoffentlich bei unserem Beispielprogramm der Fall.
2. Erkennt der Compiler Syntaxfehler, so muss der Programmierer den Fehler suchen. Die Fehlersuche kann auch schwierig werden, wenn z.B. ein Komma, ein Doppelpunkt oder ein Parameter vergessen wurde.

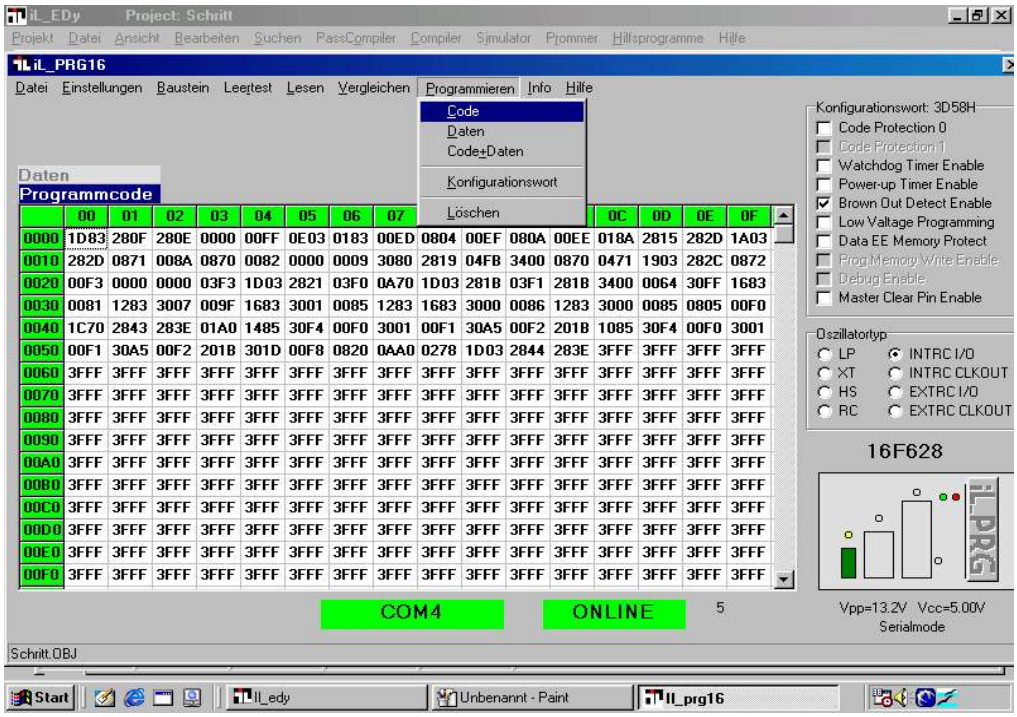
Diesmal wurde das Programm erfolgreich compiliert. Dies sieht dann so aus.



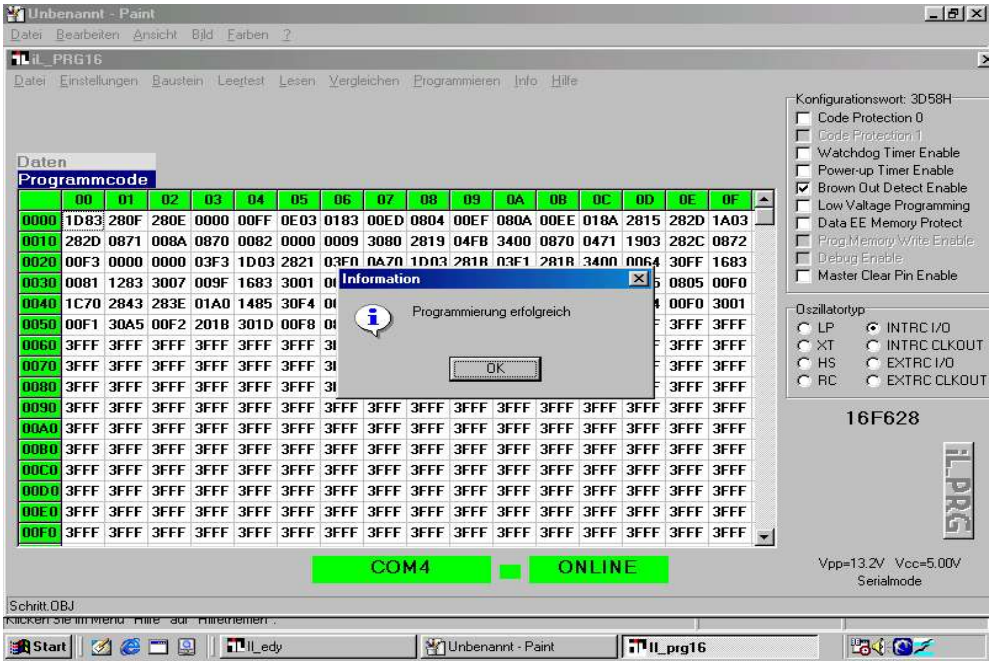
Nach dem der Compiler erfolgreich das Basic-Programm verarbeitet hat, kann der PIC-Prozessor des „Flash of Inspiration I“ mit der neu erstellten Objektdatei programmiert werden. Keine Angst, es ist ganz einfach. Der PIC-Baustein kann während der Programmierung im Board bleiben. Das Programmiergerät wird mit der RJ45-Buchse des „Flash of Inspiration I“ verbunden. Ein kurzes , ca.1m langes Netzwerkkabel (bitte nicht länger und kein Crossover), reicht für die Verbindung mit dem USB-Programmiergerät. Das Programmiermodul startet, wenn man das IC-Symbol mit dem Cursor anklickt. Beim ersten Start muss noch die richtige Schnittstelle für das Programmiergerät ausgewählt werden. Ist das Programmiergerät gefunden, sieht der Bildschirm wie folgt aus:



Das eigentliche Programmieren wird durch die Auswahl „Programmieren“ und dann „Code“ ausgelöst. Der Vorgang startet nach der Bestätigung, einer Sicherheitsabfrage, automatisch. „Flash of Inspiration I“ muss während der Programmierung mit Strom versorgt werden. Die rote LED D11 (links neben der RJ45-Buchse) leuchtet während des Programmiervorgannges.



Die erfolgreiche Programmierung zeigt das Programmiermodul an.



Jetzt kann das neu in „Flash of Inspiration I“ geschickte Programm getestet werden, da der Prozessor nach dem Programmieren sofort das neue Programm ausführt. Die Verbindung zwischen dem Programmiergerät und „Flash of Inspiration I“ kann bestehen bleiben. Fällt der Test des Programms nicht zur Zufriedenheit aus, Programm im Editor verändern und nach dem Compilieren in den PIC senden und erneut testen. Dieser Vorgang lässt sich beliebig oft wiederholen.

Tipps zum erfolgreichen Umgang mit „Flash of Inspiration I“:

Geben Sie bitte das kurze Programm „schritt.bas“ selbst über die Tastatur ein. Sie lernen so gleich die erforderliche Schreibweise der Befehle und das notwendige Format der Parameter kennen.
In Zukunft entwickeln Sie Ihre Programme auch selbst, da ist ein wenig Übung am Anfang gar nicht schlecht.

Wiederholungen im Programmcode sind zu meiden, da Wiederholungen kostbaren Speicherplatz kosten. In diesem Beispielprogramm finden Sie zweimal die „wait“ Anweisung, da das Programm so leichter lesbar ist. Besser ist es, das „wait“ in eine Subroutine (Unterprogramm) zu setzen.
Dies sieht so aus (Programmauszug „schritt.bas“):

```
        set LED             rem LED leuchtet
        gosub warte         rem Aufruf der Subroutine mit dem Label„warte“
        res LED             rem LED leuchtet nicht mehr
        gosub warte         rem Aufruf der Subroutine mit dem Label„warte“
        (weiterer Programmcode)

warte:                      rem Einsprungziel (Label) „warte“
        wait 500            rem warte 500ms
        return              rem Beende Subroutine und setze Programm an alter Stelle fort
                            (return ist am Ende einer Subroutine wichtig, bitte nicht vergessen)
```

Beim Vergeben der Labelnamen bitte die Liste der reservierten Wörter beachten. Siehe Handbuch des iL-_BAS-SES. Hilfe anklicken und Handbuch öffnen.

Fragen und Anregungen bitte per eMail an: flash@t555.de