Erste Schritte mit Flash Of Inspiration I

(von der Idee zum fertigen Programm)



Stand 05.04.2005

Seite 1 von 10

Ing. Büro Stefan Lehmann www.il-online.de

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"

iL_E	Dy	Projec	st: Schritt											_8>
Projekt	Dater	Ansicht	Bearbeiten	Suchen	PassCompiler	Compiler	Simulator	Prommer	Hilfspro	gramme	Hi∥e			
		lorol		H	0	<u>s</u>		薑		?				
	Schritt.	bas		1	1		1		1		1		1	1
	1	1,15			1999						1.01		5.07	
			Ne	ues Proje	kt							×		
					-									
					Bitte g	eben Sie	e einen Pro	ijektnam	en ein.					
				-	mi	öglichst ma	aximal 8 Zeich	ien verwer	iden					
				s	Schritt									
						-								
					Abbruch			L	OK]				
	1:1			Finfligen			Sel	uitt has						
<u> </u>				Linagen	1		SCI	inc.pds					1 Internet	
🔀 Sta	art 🛛 🙋	1 🈂 🗆	ם 🖳 🔛	ll_edy									3 €5	

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 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.

	lool	H	00		?	
Scł	nritt.bas	1	1	1	1	l l
1	rem	Programm erste S	Chritte mit	Flash of Ins	piration I	
2	rem					
3		define device 16	f628 wdt_off	,irc_osc,cmc	fg7,mclr_int	,osc2_io
4						
5		tris ra,%0000000	1 rem RA,0	= Eingang;	alle anderen	n PINS Ausgänge
6		tris rb,0	rem RB	= Alle PINS	Ausgänge	
7		TI NC	Error		_ 🗆 🗵	4 Company and the second seco second second sec
8		define tast				PIN RA,0
9		define LED	no er	rors found		Contraction of the second
10		- 140 B				
11		define zael	1024 (0400) 05-2		5.5	elle 20hex des PIC
12		Usin	g 1024 (0400H) Word	15 of 2048 (0800H) V	vords	
13		output ra,				LOW= U VOIT
14	start:			OK		danas carc
10		IL LASLE -				druckt, gene zu Start
10		For zeebler=0 to	29 rem Prog	ranne chleife	20 mal dure	hfahren
18		set LFD	rem T.FD	einschalten	So war dure	ant dur en
19		wait 500	rem wart	e 500ms		
20		res LED	rem LED	ausschalten		
21		wait 500	rem wart	e 500ms		
22		next zachler	rem Prog	rammschleife	wiederholer	2
23		goto start	rem Prog	rammschleife	30 mal auso	eführt, beginne von vor
24		The second			and the second second second second	
23	-86	Finfligen		Schritt has		

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:

LiL_P datei	RG16 Einstellu	angen	<u>B</u> austi	ein Le	eitest	Lesen	⊻ergle	eichen	Progra	ammiere	n Info	<u>H</u> ilfe	1 1100	hinitai	nines r	nine.		-Konfigu	rationswort: 3D58H
																		☐ Cod	e Protection 1 tchdog Timer Enable
Date	n Kalina ina	codo																Pow	/er-up Timer Enable
riug	00	01	02	03	04	05	86	07	08	09	0A	OB	0C	DD	ØE	OF		Brov	wn Out Detect Enable
0000	1D83	280F	280E	0000	OOFF	0E03	0183	OOED	0804	OOEF	080A	OOEE	018A	2815	282D	1A03		Date	a FE Memory Protect
0010	282D	0871	008A	0870	0082	0000	0009	3080	2819	04FB	3400	0870	0471	1903	282C	0872		Prog	g.Memory Write Enable
0020	00F3	0000	0000	03F3	1D03	2821	03F0	0A70	1D03	281B	03F1	281B	3400	0064	30FF	1683		🗖 Deb	ug Enable
0030	0081	1283	3007	009F	1683	3001	0085	1283	1683	3000	0086	1283	3000	0085	0805	00F0		∏ Mas	ter Clear Pin Enable
0040	1C70	2843	283E	01A0	1485	30F4	00F0	3001	00F1	30A5	00F2	201B	1085	30F4	00F0	3001		- Oszillato	avtyp
0050	00F1	30A5	00F2	201B	301D	00F8	0820	0AA0	0278	1D03	2844	283E	3FFF	3FFF	3FFF	3FFF		C LP	INTRC I/O
0060	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF		CXT	C INTRC CLKOU
0070	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF		C HS	C EXTRC I/O
0080	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF			
0090	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF			16F628
00A0	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF			101.020
0080	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF			<u> </u>
UUCU	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF			•
UUDU	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF		<u> </u>	
DOLO	3111	3FFF	3111	3FFF	3111	3FFF	3FFF	3FFF	3111	3111	3FFF	3FFF	3FFF	3FFF	3111	3FFF			• ה
UUFU	эггг	эггг	JEFF	эггг	эггг	SELE	ərrr	эггг	эггг	ərrr	arre	эггг	эггг	SELE	ərrr	эггг	-		
								CO	M4			O	NLIN	E	5			Vpp	=13.2V Vcc=5.00V
1.000	D 1																		Senaimode
crinitt. U	DJ																		

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.

ren Ernsteinungen Baustein Leeftest Fesen ⊼ergleichen j										ammiere ode aten ode <u>+</u> Da	iten							Konfig C C C C	gurationswort: 3D58H- ode Protection 0 ode Protection 1 (atchdog Timer Epable
ater		code							<u>K</u> c	onfigura	tionswo	ort						F P	ower-up Timer Enable
Ug	00	01	02	03	84	05	86	07	Löschen				OC.	OD	ÐE	OF		₽ Bi	rown Out Detect Enabl
000	1D83	280F	280E	0000	OOFF	0E03	0183	OOED	0804	OOEF	080A	OOEE	018A	2815	282D	1A03			ow vallage Programmir ata FF Memory Protect
010	282D	0871	008A	0870	0082	0000	0009	3080	2819	04FB	3400	0870	0471	1903	282C	0872		E P	rog.Memory Write Enat
020	00F3	0000	0000	03F3	1D03	2821	03F0	0A70	1D03	281B	03F1	281B	3400	0064	30FF	1683		ГD	ebug Enable
030	0081	1283	3007	009F	1683	3001	0085	1283	1683	3000	0086	1283	3000	0085	0805	00F0		ГМ	aster Clear Pin Enable
040	1C70	2843	283E	01A0	1485	30F4	00F0	3001	00F1	30A5	00F2	201B	1085	30F4	OOFO	3001		- O szilla	atortun
050	00F1	30A5	00F2	201B	301D	00F8	0820	0AA0	0278	1D03	2844	283E	3FFF	3FFF	3FFF	3FFF		CLE	
060	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF		CX	T C INTRO CLKI									
070	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF		СН	S C EXTRCI/O									
080	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF		CR										
090	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF			165600									
0A0	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF		10	101 020									
OBO	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF			0									
OCO	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF												
ODO	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF		c										
OEO	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF												
OFO	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	-	1	
		CC								M4 ONLINE ⁵							V	pp=13.2V Vcc=5.00V Serialmode	

Die erfolgreiche Programmierung zeigt das Programmiermodul an.

HL_F	RG16		Dente	Street 194		1.0.0					2011 - 1070	a tot raisa							
ater)ater 2root	n Churchen Churchen Churchen	ungen Icodie	Dause	en Le	eŭesi	Lesen	Yergi		Eloĝis		n Inic							Konfigu Coo Coo Wa Pov	irationswort: 3D5 de Protection 0 de Protection 1 itchdog Timer En wer-up Timer Ena
	00	01	02	03	04	05	06	07	08	09	0A	OB	0C	BD	0E	OF		Lov	wh Out Detect E v Valtage Program
0000	1D83	280F	280E	0000	OOFF	0E03	0183	OOED	0804	OOEF	080A	OOEE	018A	2815	282D	1A03		T Dat	ta EE Memory Pri
0010	282D	0871	008A	0870	0082	0000	0009	3080	2819	04FB	3400	0870	0471	1903	282C	0872		E Pro	g Memory Write I
0020	00F3	0000	0000	03F3	1D03	2821	03F0	NA70	1003	281R	0.3F1	281R	3400	1064	30FF	1683		I Ma	ster Clear Pin En
0030	0081	1283	3007	009F	1683	3001		onnaci	on.					<u>e</u>	0805	UUFU			
JU4U	1070	2843	283E	01A0	1485	30F4	01	1	Progra	ammieru	ng erfol	lgreich		Ē	OUFU	3001		Oszillat	ortyp
0050	DUFI	JUAD	OUF 2	2018	3010	OULS SEEE	21	~						Ē	SFFF	SELE		CLP	
0000	SELE	SELE	SELE	OFFF	SELE	SELE	21			[1		E	OFFF	SELE		CHS	C EXTRC
0070	3FFF	SEFE	SEFE	SEFE	3FFF	SEFE	31			<u> </u>	JIV.			-	SEFE	3FFF		C RC	C EXTRC
0000	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3EEE	3FFF	3FFF	3FFF	3FFF			
DOAD	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF			16F628
DOBO	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF			
DOCO	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF			
00D0	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF			
DOEO	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF			
DOFO	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	3FFF	-		
								CO	M4			0	NLIN	E				Vpj	p=13.2V Vcc=5 Serialmode

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 Programme	ode)
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: <u>flash@t555.de</u> © DF2OAG 2005