

# PLD MachXO Mini: Arbeitsanleitung



Version 0.2, Nov. 2010, © by Bruno Wamister

## Inhalt:

<b>Thema:</b>	<b>Seite:</b>
MachXO Mini Evaluation Board	2
My First PLD Design	3
My First PLD Design - That's the Way...	4
Arbeiten mit Bus-Darstellungen	9
Beispiel mit IPexpress	10
Komponenten und Bezeichnungen	12
Installation der ispLEVER-Starter Software	14
ASCII-Tabelle	15
Anschlussbelegung MachXO Mini	16
Multifunktion I/O-Platte	17
4 Digit BCD 7-Segment LED-Anzeige	18

## Inhaltsübersicht:

Dieses Dokument enthält Anleitungen und Tipps für Lernende zum Programmieren des MachXO Mini Development Kits.

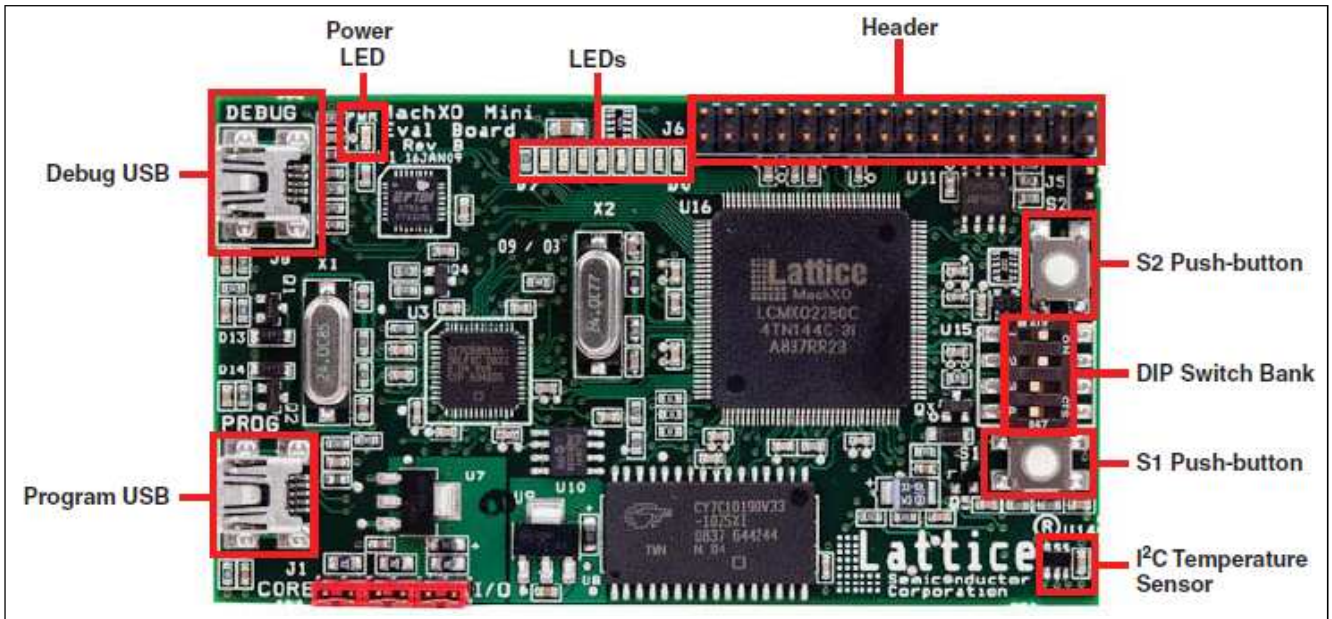
Eine Anleitung zeigt, wie Schemas gezeichnet, kompiliert und in das MachXO gebrannt werden.

Der Umgang mit Bus-Systemen sowie die Anwendung des IPexpress Makrogenerators werden beschrieben.

In einer Liste sind die zur Verfügung stehenden Bauteile (Gatter, FF usw.) aufgeführt.

Im Anhang sind Anschlussbelegungen des MachXO Mini Prints, der MFIO-Platte sowie der 7-Segment Display-Platte zusammengestellt.

## MachXO Mini Evaluation Board



### MachXO Mini Evaluation Board

- MachXO LCMXO2280C-4TN144C
- 2-Mbit SPI Flash memory
- 1-Mbit SRAM
- I2C temperature sensor
- 2 USB connectors (JTAG, RS-232)
- 2x16 expansion header for I/O, I2C, and SPI
- Push-buttons sleep mode and global set/reset
- 4-bit DIP switch
- DAC/ADC circuit
- MachXO Sleep Mode circuit
- Eight status LEDs

### MachXO LCMXO2280C-4TN144C

Herzstück des Boards ist das Mach XO PLD. Die 2280 LUTs (Makrozellen) lassen sich für die unterschiedlichsten Anwendungen programmieren. Der IC verfügt zudem über 27.6Kbit SRAM (kann auch als ROM eingesetzt werden), einen internen Oszillator (20MHz), zwei PLL Schaltungen und 113 I/O Pins.

Eine Schaltung für einen 8 Bit Microcontroller kann im Mach XO programmiert werden. Dabei werden in der Minimalversion ca. 200 LUTs belegt.

### Möglichkeiten des MachXO Mini Evaluation Boards

Mit dem Mini Board können von kombinatorischen Schaltungen über sequentielle Logik bis hin zu komplexen Microcontrollerschaltungen alle Stufen von digitaler Logik gebaut werden. Es ist auch möglich, Anwendungen mit SPI und I2C Schnittstellen zu bauen. Die Debug USB-Schnittstelle kann als RS232 Schnittstelle verwendet werden wenn auf dem PC ein virtuelles COM-Port installiert wird (Treiber vorhanden). Mit den RAM/ROM Speichern können Schaltungen mit Arrays konstruiert werden. Auch AD/DA-Wandler nach dem Delta-Sigma Prinzip können gebaut und eingesetzt werden.

**VORSICHT:** Der MachXO IC wird mit 3.3V betrieben und die I/Os sind **NICHT 5V tolerant!**

**Programmiersoftware ispLEVER Starter** (Download: <http://www.latticesemi.com> )

Das Mach XO kann mit ispLEVER Starter ohne VHDL Kenntnisse programmiert werden. Schemas von digitalen Schaltungen werden gezeichnet und anschliessend durch ispLEVER compiliert. Für das Erstellen der Schemas steht eine Symbolbibliothek mit den wichtigsten digitalen Schaltungen zur Verfügung (Tore, D-FF). Das generierte JEDEC File wird mit dem Programm ispVM über die USB Programm Schnittstelle in den Mach XO gebrannt.

Mit dem Tool IPexpress lassen sich in wenigen Schritten arithmetische Module, Counter und Speicher (RAM, ROM, FIFO) konfigurieren, die anschliessend in die Anwenderschaltung eingebaut werden.

## My First PLD Design

Mit einer ganz einfachen Schaltung soll gezeigt werden, wie das PLD MachXO programmiert werden kann.

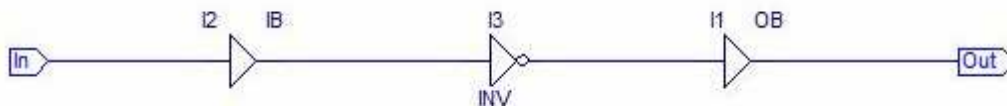
### Aufgabe MyFirst:

Der Taster Tas S1 soll invertiert und auf der MachXO LED-Zeile LED D7 angezeigt werden. Sowohl der Taster als auch die LED sind nullaktiv.

Alle Eingänge einer Schaltung müssen einen Signalnamen, eine Pin-Nummer und einen Input Buffer haben. Alle Ausgänge einer Schaltung müssen einen Signalnamen, eine Pin-Nummer und einen Output Buffer haben.

Für den Taster Tas S1 ist auf dem MachXO bereits ein pullup Widerstand angebracht.

### Schaltung der Aufgabe MyFirst:



Auf den folgenden Seiten ist das Vorgehen schrittweise dokumentiert.

Meine Notizen:

# My First PLD Design - That's the Way...

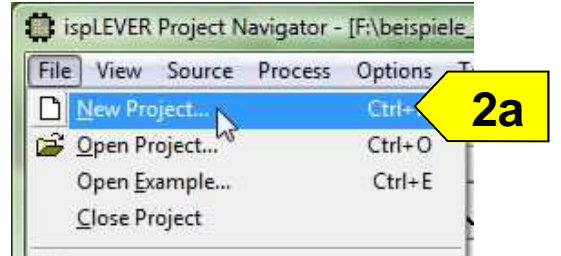
**1**

ispLEVER Project Navigator starten...



**2**

Im ispLEVER Project Navigator => File => New Project..



**3**

**Project Wizard: Pfade und Tools festlegen**

**Tipps:**

Namen für Dateien und Ordner:

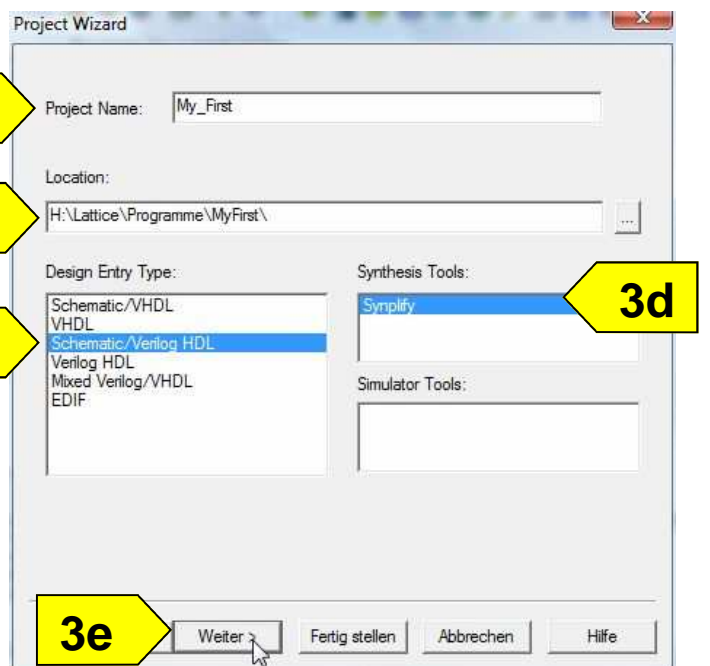
Keine Sonderzeichen, keine Zwischenräume und keine Umlaute verwenden!

Für jedes neue Projekt einen neuen Ordner öffnen.

**3a**

**3b**

**3c**



**4**

**Project Wizard: PLD auswählen**

Family: MachXO

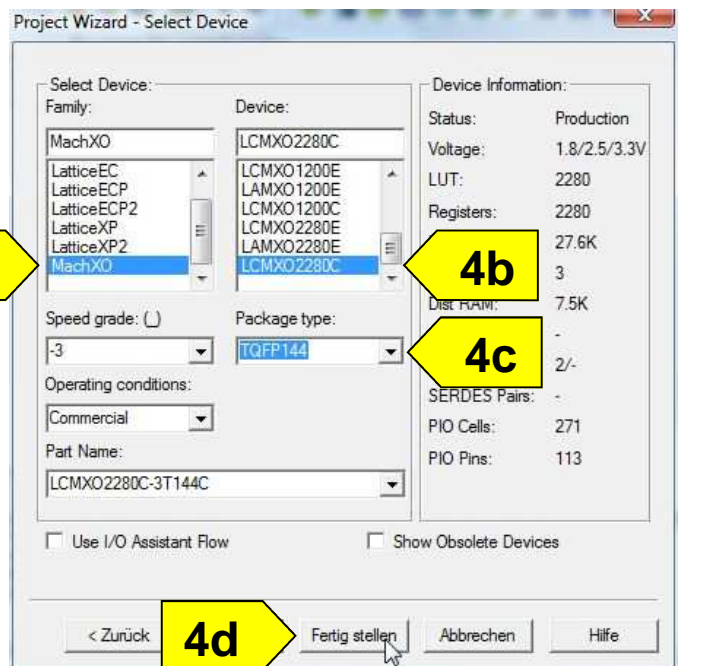
Device: LCMXO22880C

Package: TQFP144

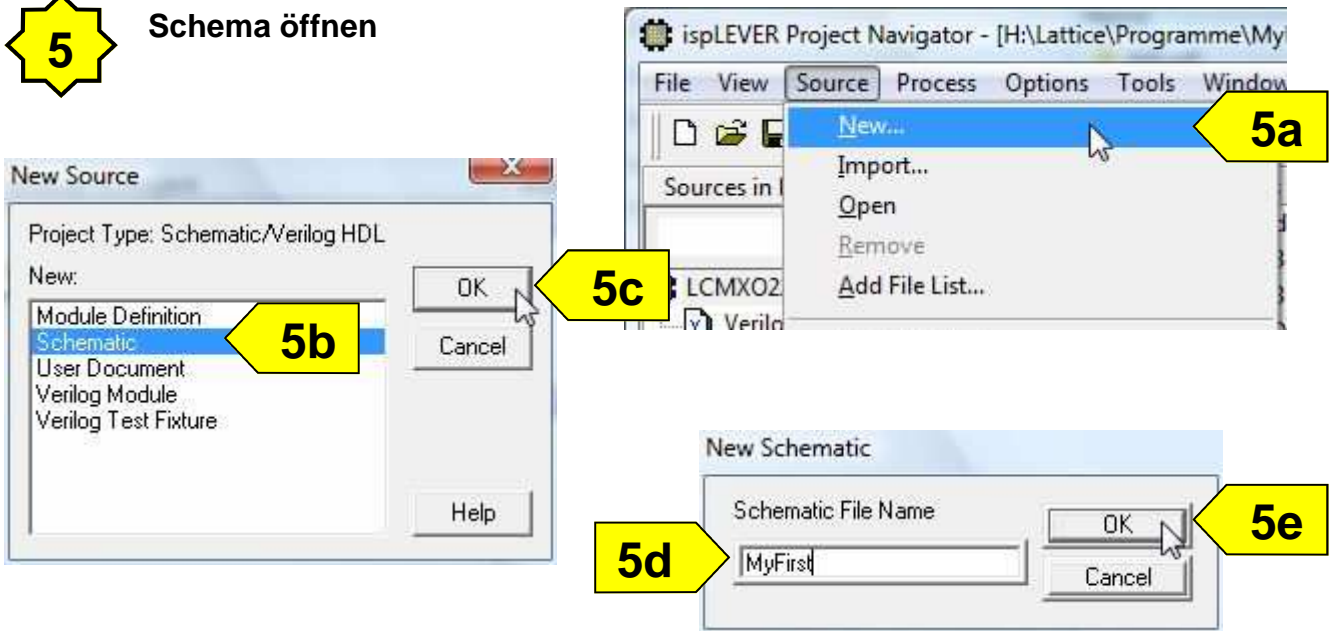
**4a**

**4b**

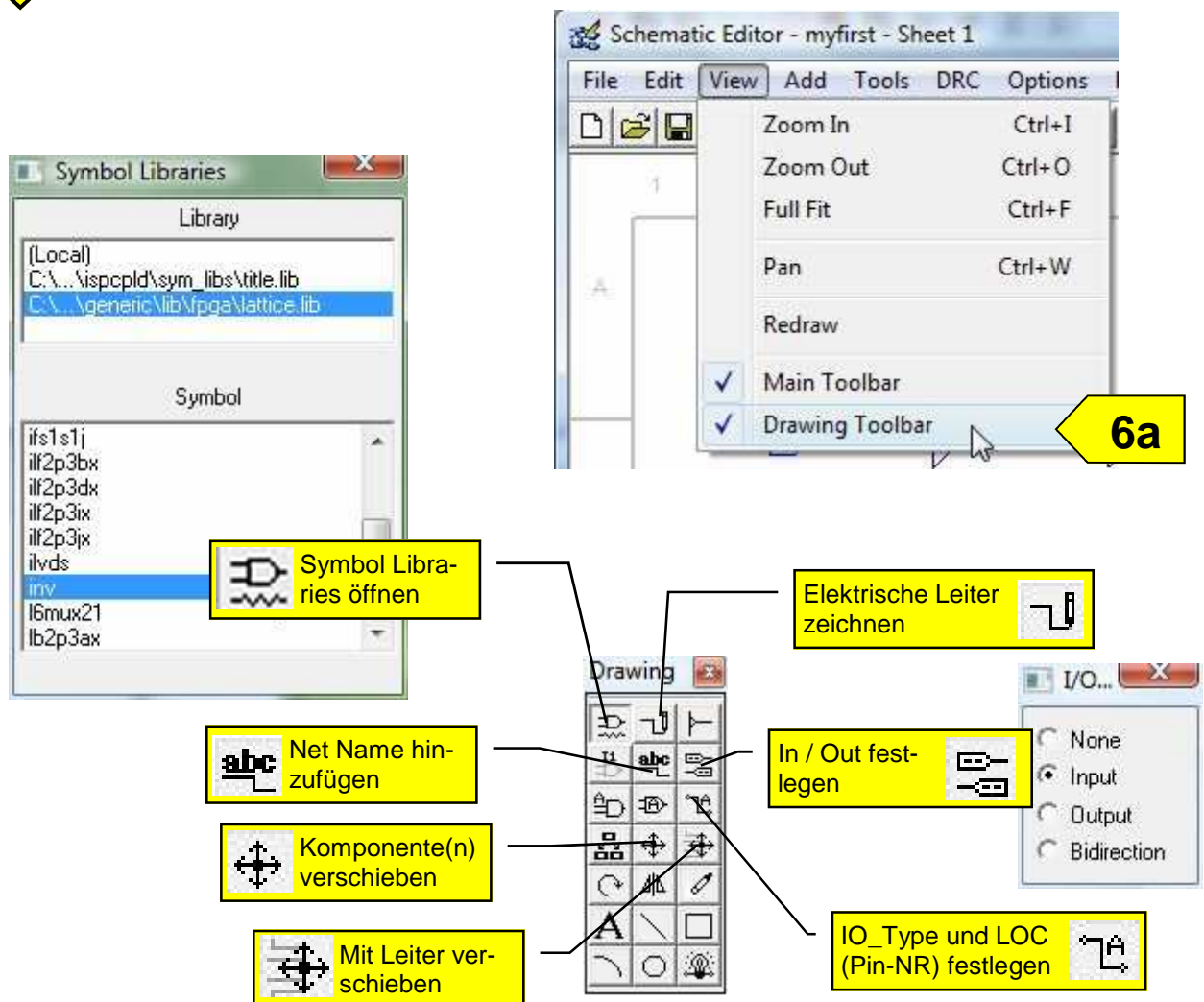
**4c**



## 5 Schema öffnen

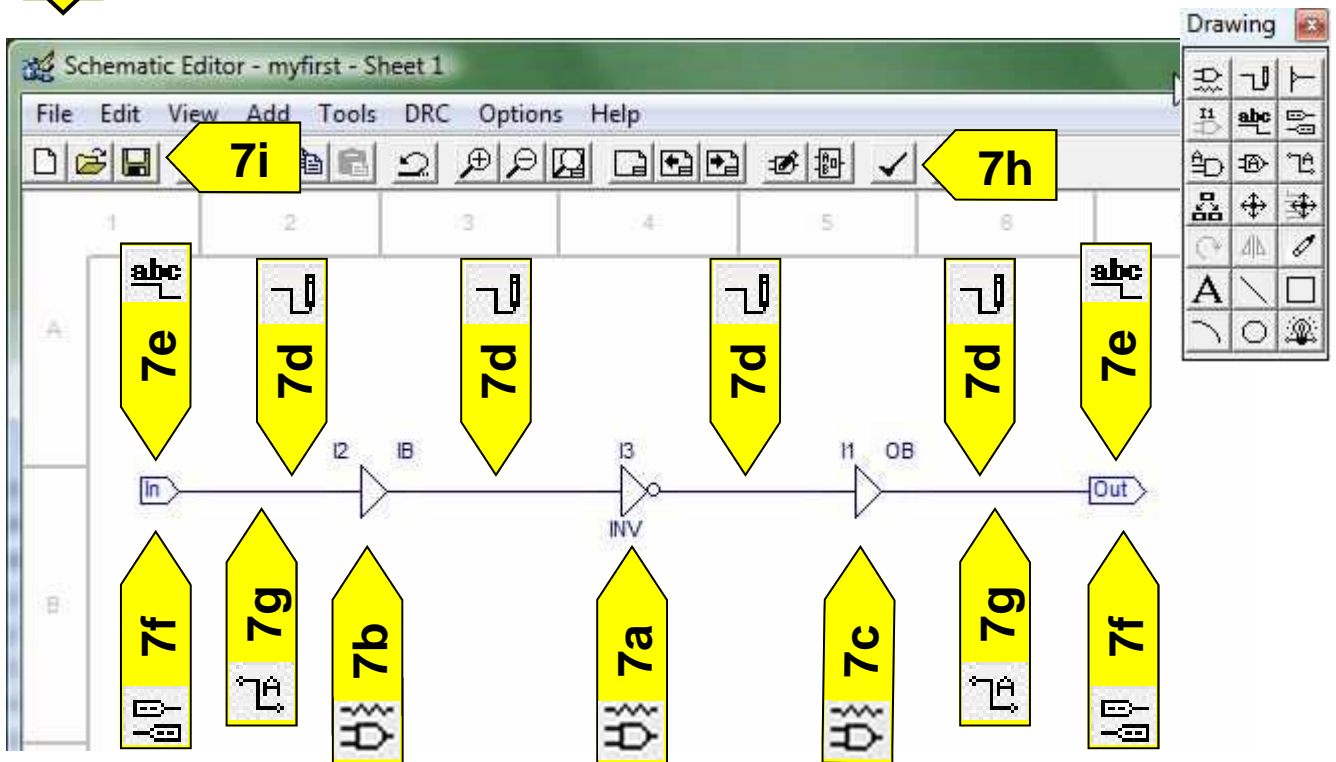


## 6 Schema: Wichtigste Werkzeuge



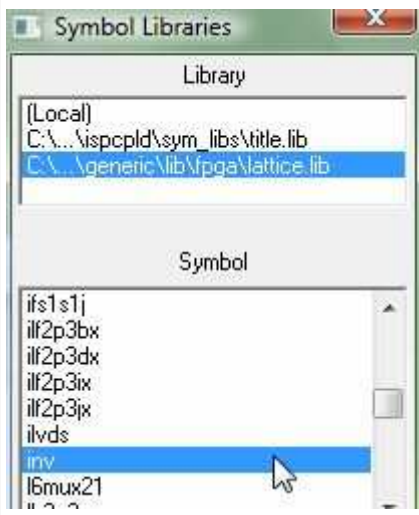
**7**

**Schema: Zeichnen**



**a, b, c : Schaltsymbol auswählen**

- INV Inverter
- IB Input Buffer
- OB Output Buffer



**d : Leiter zeichnen**

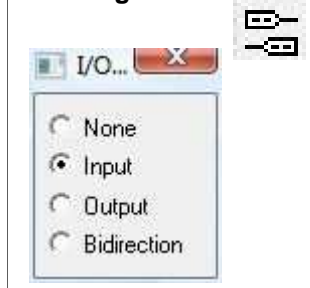
Zwischen allen Schaltsymbolen muss ein Stück Leiter gezeichnet werden.

**e : Signalnamen zuordnen**

Jeder Ein- und Ausgang muss einen Signalnamen haben

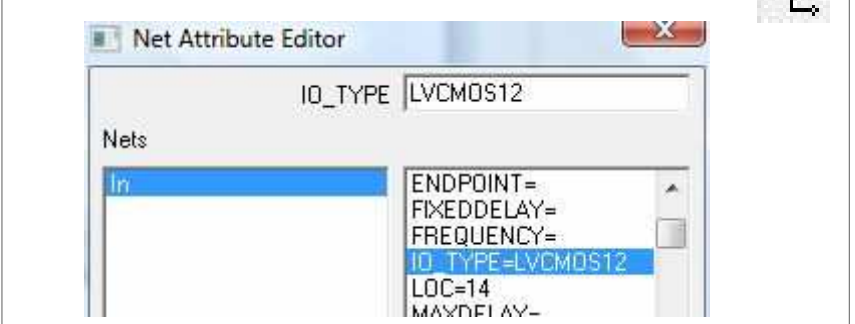


**f : Eingang / Ausgang festlegen**



**g : Eingang / Ausgang Typ und Pin-Nummer definieren**

IO\_TYPE=LVCMOS12 und Pin-Nr LOC= definieren



**h, i : Schema kontrollieren und speichern**



## 8 Schema compilieren, JEDEC Datei generieren

### Schema compilieren:

8a: IC MachXo markieren

8b: JEDEC File anklicken

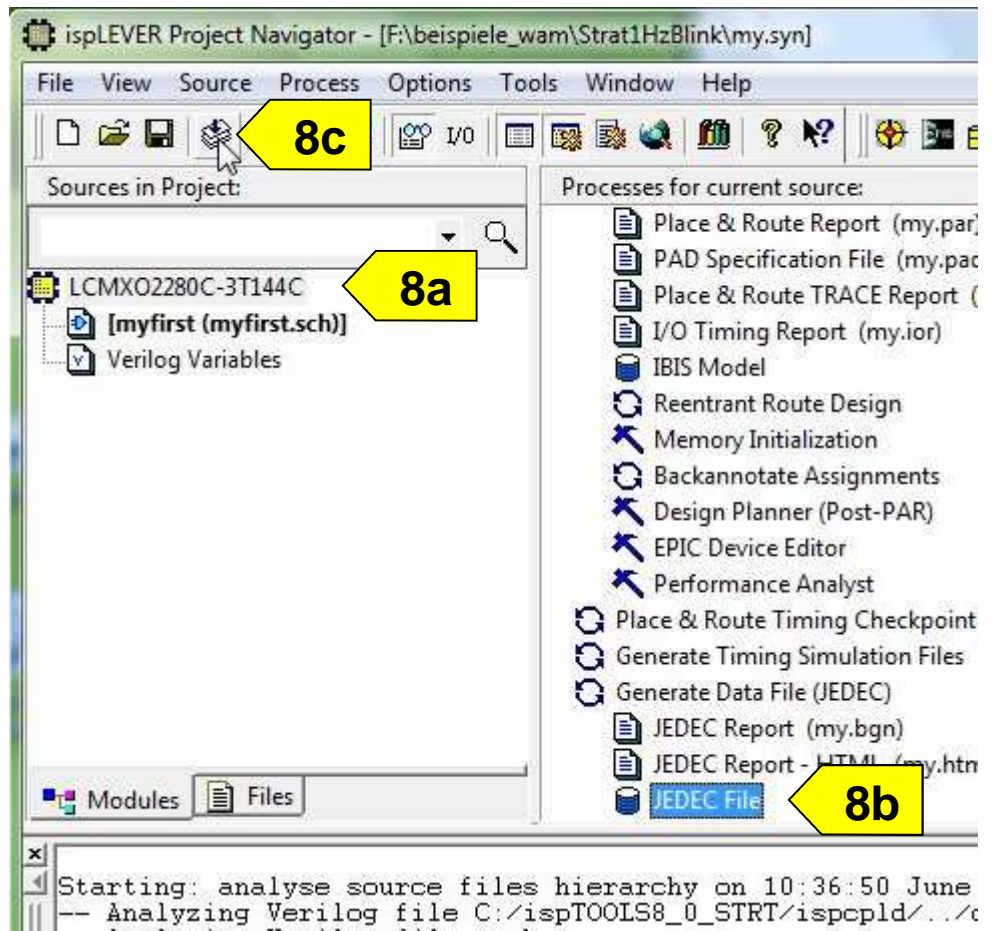
8c: Start der Compilierung

### Bemerkungen:

Das JEDEC File wird nur erstellt, wenn während dem Compilieren keine Fehler auftreten. Die \*.jed Datei wird im aktuellen Arbeitsordner gespeichert.

Warnungen sind in den meisten Fällen nicht von Bedeutung und können ignoriert werden.

Bei Windows Vista und 7 muss evtl. eine Programmausführung bestätigt werden.



## 9 MachXO anschliessen und ispVM System starten

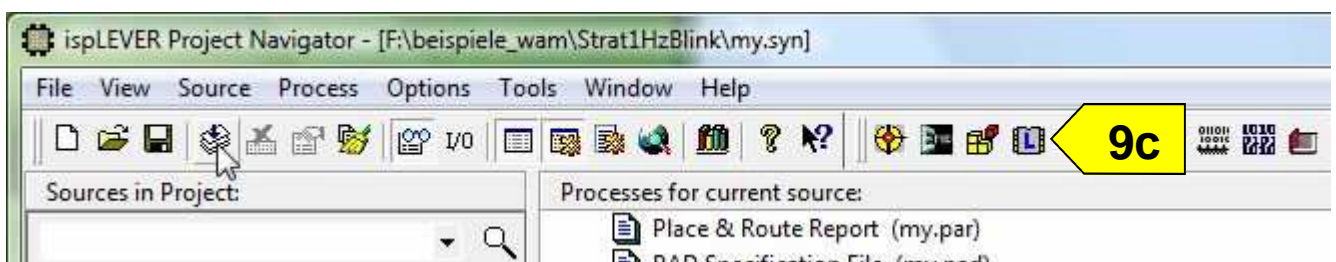
• 9a: MachXO Mini USB Programmierschnittstelle am PC anschliessen

9a

• 9b: Speisung des MachXO Mini erfolgt über die Debug USB Schnittstelle. Diese muss am PC oder an einen 5V Netzadapter angeschlossen werden

9b

• 9c: In der Toolbar des ispLEVER Project Navigators ispVM System starten



## 10 ispVM System konfigurieren, MachXO brennen

Falls der Treiber für die USB-Schnittstelle noch installiert werden muss. Den Treiberpfad selber wählen und nicht Windows automatisch suchen lassen! C:\ispTOOLS\ispvmsystem

**Vorgehen:**

**10a:** Alle evtl. bereits vorhandenen 'Devices' löschen.

**10b:** Neues Device hinzufügen

**10c:** Device Description (frei)

**10d:** Select new Device: MachXO LCMXO2280C auswählen

**10e:** Pfad zum zu brennenden jed File angeben

**10f:** Fenster mit OK schliessen

**10g:** Mit GO Brennvorgang starten

**10h:** Warten bis Brennvorgang abgeschlossen ist

Index	Device List	FileName/IR-Length	Operation	Status
1	LCMXO2280C	F:\beispiele_wam\MyFirst\myfirst.jed	FLASH Erase, Progra...	N/A

Nach dem Brennen kann nun die Schaltung im MachXO getestet werden... Viel Erfolg!

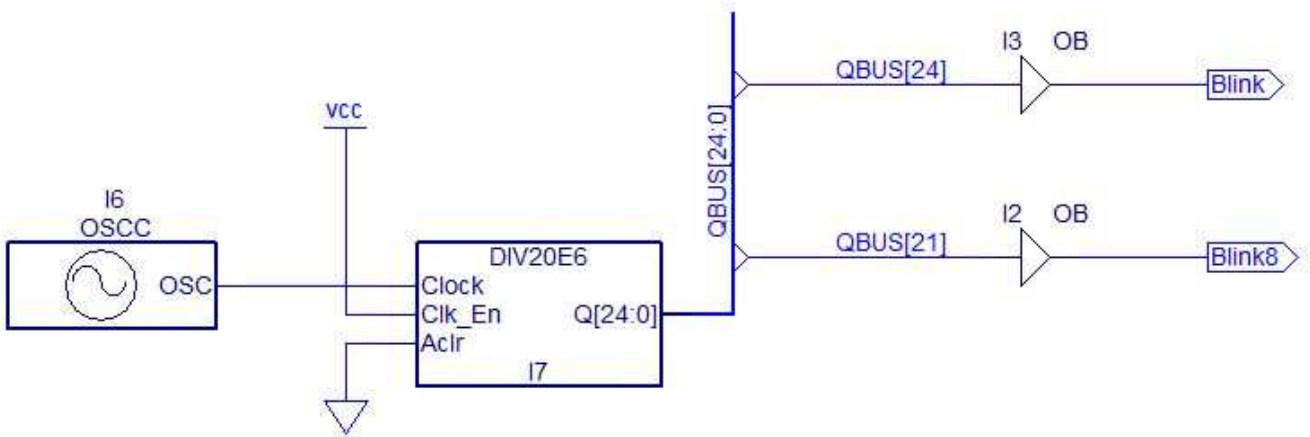
# Arbeiten mit Bus-Darstellungen

Im ursprünglichen datentechnischen Sinne versteht man unter einem Bus mehrere parallel geführte Datenleitungen, die als eine breitere Leitung gezeichnet wird. Bus Darstellungen in komplexen Schaltungen vereinfachen die Lesbarkeit und machen Schemas übersichtlicher. Hier noch zwei Begriffe, die im Zusammenhang mit der Organisation von Bus-Systemen verwendet werden:

- Big Endian:        QBus[MSB:LSB]    Das MSB wird an der höheren Adresse gespeichert
- Little Endian:    QBus[LSB:MSB]    Das MSB wird an der niedrigsten Adresse gespeichert

## Beispiel mit Bus:

Die 20MHz Frequenz des internen Oszillators wird mit dem Teiler DIV20E6 auf eine Frequenz von 1 Hz geteilt. Der Teiler DIV20E6 hat einen Ausgangs-Bus mit 25 parallelen Leitungen, von denen 2 Leitungen (24 und 21) abgenommen werden.



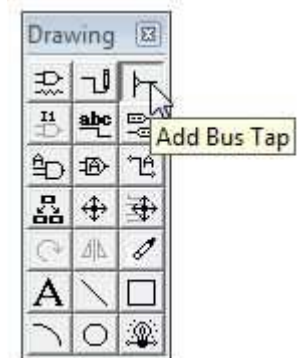
### Bus-Gesamleitung zeichnen und bezeichnen

- Bus wie eine Normale Verbindung zeichnen
- Bus markieren (mit linker Maustaste auf Bus klicken)
- Mit *Add Net Name* Bus bezeichnen. Angabe mit Doppelpunkt in eckigen Klammern [X:Y] definieren die Leitung als Bus-Gesamleitung



### Bus-Anschlüsse zeichnen und bezeichnen

- Im Draw-Menü *Add BusTap* anklicken
- Den Bus-Anschluss beim Bus beginnend zeichnen
- Bus-Anschluss markieren (mit linker Maustaste auf Bus klicken)
- Mit *Add Net Name* Bus-Anschluss bezeichnen. Angabe in eckigen Klammern [Z] mit nur einer Zahl, definieren die Leitung als Bus-Anschluss



## Beispiel mit IPexpress

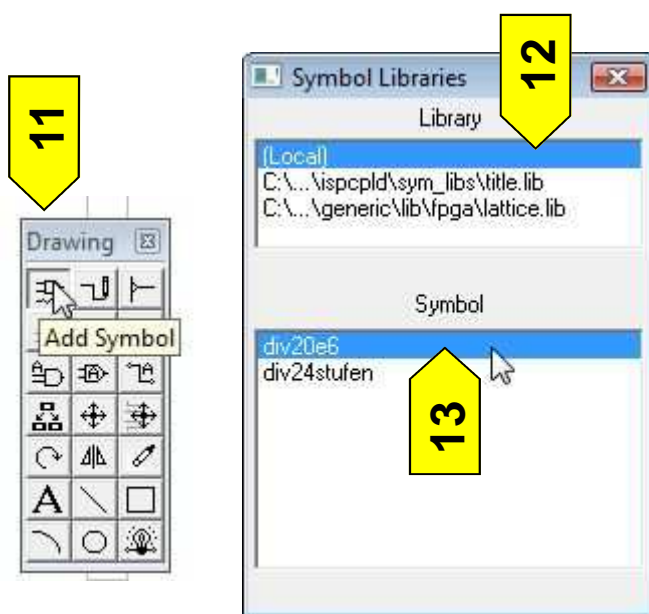
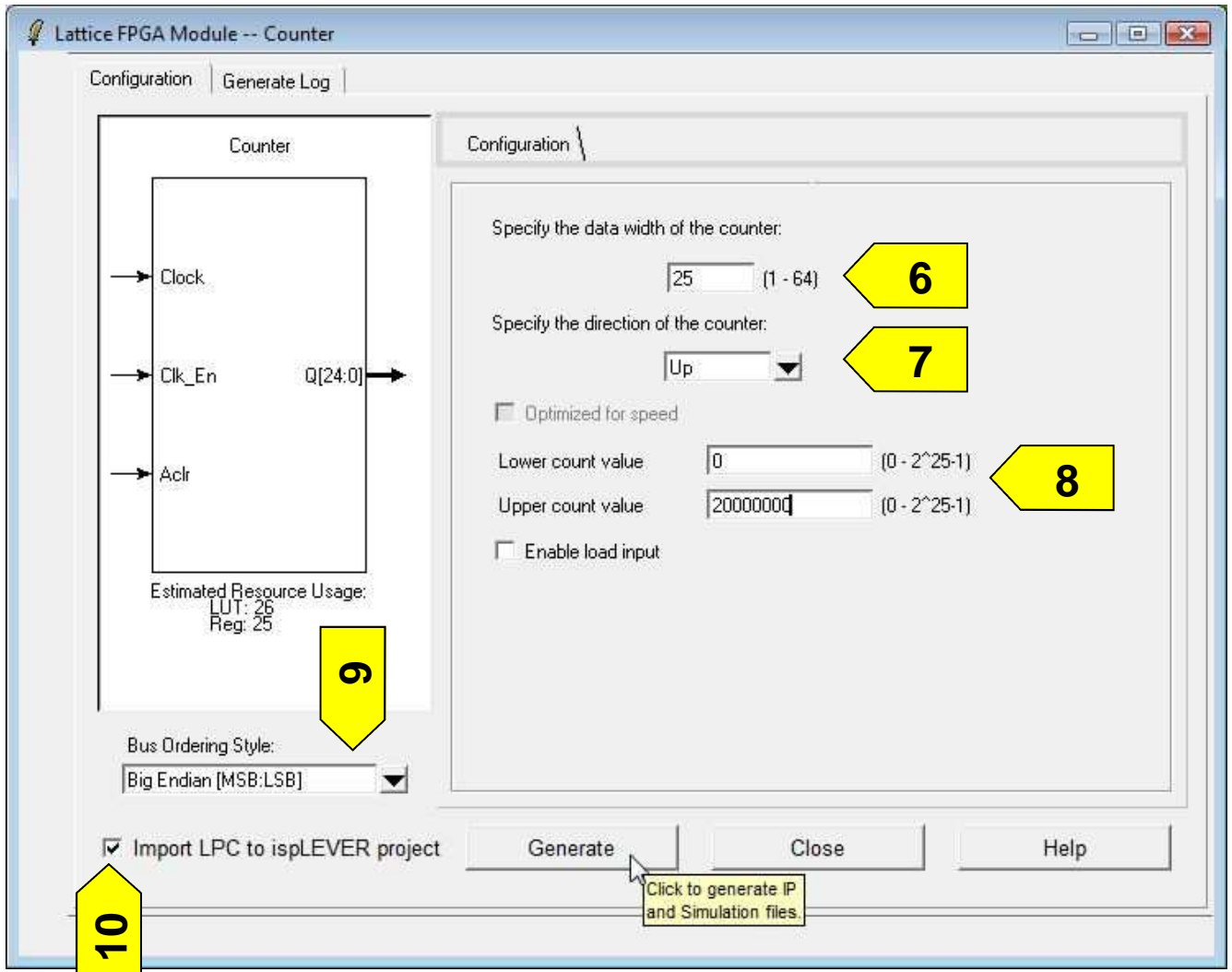
Mit IPexpress lassen sich komplexe logische und arithmetische Schaltungskomponenten einfach generieren. IPexpress erstellt von der definierten Schaltung ein Makro, welches anschliessend in ein Schema eingebunden werden kann.

**Problem:** Mit einem Teiler :20Mio soll die 20MHz Frequenz des internen Oszillators des MachXO auf 1Hz geteilt werden.

The image shows two screenshots from the ispLEVER Project Navigator and IPexpress software. The top screenshot shows the Project Navigator with the IPexpress icon highlighted by a yellow arrow labeled '1'. The bottom screenshot shows the IPexpress window with the 'Arithmetic\_Modules' folder expanded, the 'Counter' module selected (labeled '3'), the 'File Name' field set to 'Div20E6' (labeled '4'), and the 'Customize' button highlighted by a yellow arrow labeled '5'.

**Vorgehen:**

- 1: IPexpress im ispLEVER Project Navigator starten
- 2: Arithmetic\_Modules öffnen
- 3: Counter auswählen
- 4: File Name angeben z.B. Div20E6
- 5: Button Customize drücken...
- 6: Anzahl binäre Stufen (FF) angeben
- 7: Zählrichtung definieren
- 8: Zählbereich einstellen
- 9: Big/Little Endian festlegen
- 10: Makro automatisch in das Projekt importieren
- 11: Im Schema Add Symbol wählen
- 12: Symbol Library Local öffnen
- 13: Makro div20e6 auswählen und im Schema einfügen



**Vorgehen:**

- 1: IPexpress im ispLEVER Project Navigator starten
- 2: Arithmetic\_Modules öffnen
- 3: Counter auswählen
- 4: File Name angeben z.B. Div20E6
- 5: Button Customize drücken...
- 6: Anzahl binäre Stufen (FF) angeben
- 7: Zählrichtung definieren
- 8: Zählbereich einstellen
- 9: Big/Little Endian festlegen
- 10: Makro automatisch in das Projekt importieren
- 11: Im Schema Add Symbol wählen
- 12: Symbol Library Local öffnen
- 13: Makro div20e6 auswählen und im Schema einfügen

## Komponenten und Bezeichnungen (Auszug aus Lattice Document FPGAlibrary.pdf)

### Logic Gates:

AND2	2 Input AND Gate
AND3	3 Input AND Gate
AND4	4 Input AND Gate
AND	5 Input AND Gate
ND2	2 Input NAND Gate
ND3	3 Input NAND Gate
ND4	4 Input NAND Gate
ND5	5 Input NAND Gate
OR2	2 Input OR Gate
OR3	3 Input OR Gate
OR4	4 Input OR Gate
OR5	5 Input OR Gate
NR2	2 Input NOR Gate
NR3	3 Input NOR Gate
NR4	4 Input NOR Gate
NR5	5 Input NOR Gate
XNOR2	2 Input Exclusive NOR Gate
XNOR3	3 Input Exclusive NOR Gate
XNOR4	4 Input Exclusive NOR Gate
XNOR5	5 Input Exclusive NOR Gate
XOR2	2 Input Exclusive OR Gate
XOR3	3 Input Exclusive OR Gate
XOR4	4 Input Exclusive OR Gate
XOR5	5 Input Exclusive OR Gate
XOR11	11 Input Exclusive OR Gate
XOR21	21 Input Exclusive OR Gate

### Miscellaneous Logic

INV	Inverter
VHI	Logic High Generator
VLO	Logic Low Generator
VCC	+ Supply Voltage
GND	Ground

### Multiplier

MULT2	2x2 Multiplier
-------	----------------

### Comperators:

AGEB2	"A" Greater Than Or Equal To "B" (2 bit)
ALEB2	"A" Less Than Or Equal To "B" (2 bit)
ANEB2	"A" Not Equal To "B" (2 bit)

### Adders/Subtractors

FADD2	2 Bit Fast Adder
FSUB2	2 Bit Fast Subtractor (two's comple-
FADSU2	2 Bit Fast Adder/Subtractor (two's complement)

### Multiplexer

MUX161	16 to 1 Mux
MUX321	32 to 1 Mux
MUX21	2 to 1 Mux
MUX41	4 to 1 Mux
MUX81	8 to 1 Mux

### Input Buffers

IB	Input Buffer
IBPD	Input Buffer with Pull-down
IBPU	Input Buffer with Pull-up

### Output Buffers

OB	Output Buffer
OBW	Output Buffer with Tristate
OBZPD	Output Buffer with Tristate and Pull-down
OBZPU	Output Buffer with Tristate and Pull-up

### Special Cells

OSCC	Internal Oscillator (ca. 20MHz)
GSR	Global Set/Reset
PUR	Power Up Set/Reset
GOE	Global Ouput Enable
TBUF	Internal Buffer with Tristate

**Flip-Flops:**

FD1P3AX	Positive Edge Triggered D Flip-Flop with Positive Level Enable, GSR used for Clear
FD1P3AY	Positive Edge Triggered D Flip-Flop with Positive Level Enable, GSR used for Preset
FD1P3BX	Positive Edge Triggered D Flip-Flop with Positive Level Enable and Positive Level Asynchronous Preset
FD1P3DX	Positive Edge Triggered D Flip-Flop with Positive Level Enable and Positive Level Asynchronous Clear
FD1P3IX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Clear and Positive Level Enable (Clear overrides Enable)
FD1P3JX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Preset and Positive Level Enable (Preset overrides Enable)
FD1S3AX	Positive Edge Triggered D Flip-Flop, GSR Used for Clear
FD1S3AY	Positive Edge Triggered D Flip-Flop, GSR Used for Preset
FD1S3BX	Positive Edge Triggered D Flip-Flop with Positive Level Asynchronous Preset
FD1S3DX	Positive Edge Triggered D Flip-Flop with Positive Level Asynchronous Clear
FD1S3IX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Clear
FD1S3JX	Positive Edge Triggered D Flip-Flop with Positive Level Synchronous Preset
FL1P3AY	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, and Positive Level Enable, GSR used for Preset
FL1P3AZ	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, and Positive Level Enable, GSR used for Clear
FL1P3BX	Positive Edge Triggered D Flip-Flop with 2 Input Data Mux, Positive Level Data Select, Positive Level Asynchronous Preset, and Positive Level Enable

**Latches:**

FD1S1A	Positive Level Data Latch with GSR Used for Clear
FD1S1AY	Positive Level Data Latch with GSR Used for Preset
FD1S1B	Positive Level Data Latch with Positive Level Asynchronous Preset
FD1S1D	Positive Level Data Latch with Positive Level Asynchronous Clear
FD1S1I	Positive Level Data Latch with Positive Level Synchronous Clear
FD1S1J	Positive Level Data Latch with Positive Level Synchronous Preset
FL1S1A	Positive Level Loadable Latch with Positive Select and GSR Used for Clear
FL1S1AY	Positive Level Loadable Latch with Positive Select and GSR Used for Preset
FL1S1B	Positive Level Loadable Latch with Positive Select and Positive Level Asynchronous Preset

# Installation der ispLEVER-Starter Software

<http://www.latticesemi.com> => Downloads => Software => ispLEVER-Starter. Die beiden Dateien herunterladen und installieren (falls noch nicht vorhanden, muss vorab ein persönlicher Lattice Account eingerichtet werden):

**Zuerst:** Installieren von Primary Module: **isp8\_0m1.FPGA.exe**

**TIPPS:** Die Software im Root-Verzeichnis installieren, wie es vom Installationsprogramm vorgegeben wird! Während der Installation werden auch die Treiber für die Schnittstelle installiert. Dazu muss das MachXO Mini bereit sein.

**Danach:** Installieren von Synplify Synthesis Module: **SYN\_starter\_80\_120309.exe**

**ACHTUNG:** ispLEVER **nicht starten**, bevor das Lizenz-File in die entsprechende Datei kopiert wurde!

**Zuletzt:** Bei Lattice Lizenz-File beantragen:

<http://www.latticesemi.com> => Products => Design Software => Licensing => ispLever-Starter => Request a license

Hier wird die Physikalische Adresse der Netzwerkkarte verlangt. Man findet diese Nummer auf dem PC, auf welchem die SW installiert werden soll unter: Start => Alle Programme => Zubehör => Eingabeaufforderung Im DOS-Fenster nun eingeben: ipconfig/all



Die Physikalische Adresse

00-1D-60-64-44-50

nun ohne Bindestriche im Fenster links eingeben, E-Mailadresse eingeben und *Generate License* klicken....

Ein License File wird an die E-Mailadresse gesendet.

Die erhaltene Datei license.dat in den Pfad \license der installierten ispLEVER-Starter Software kopieren

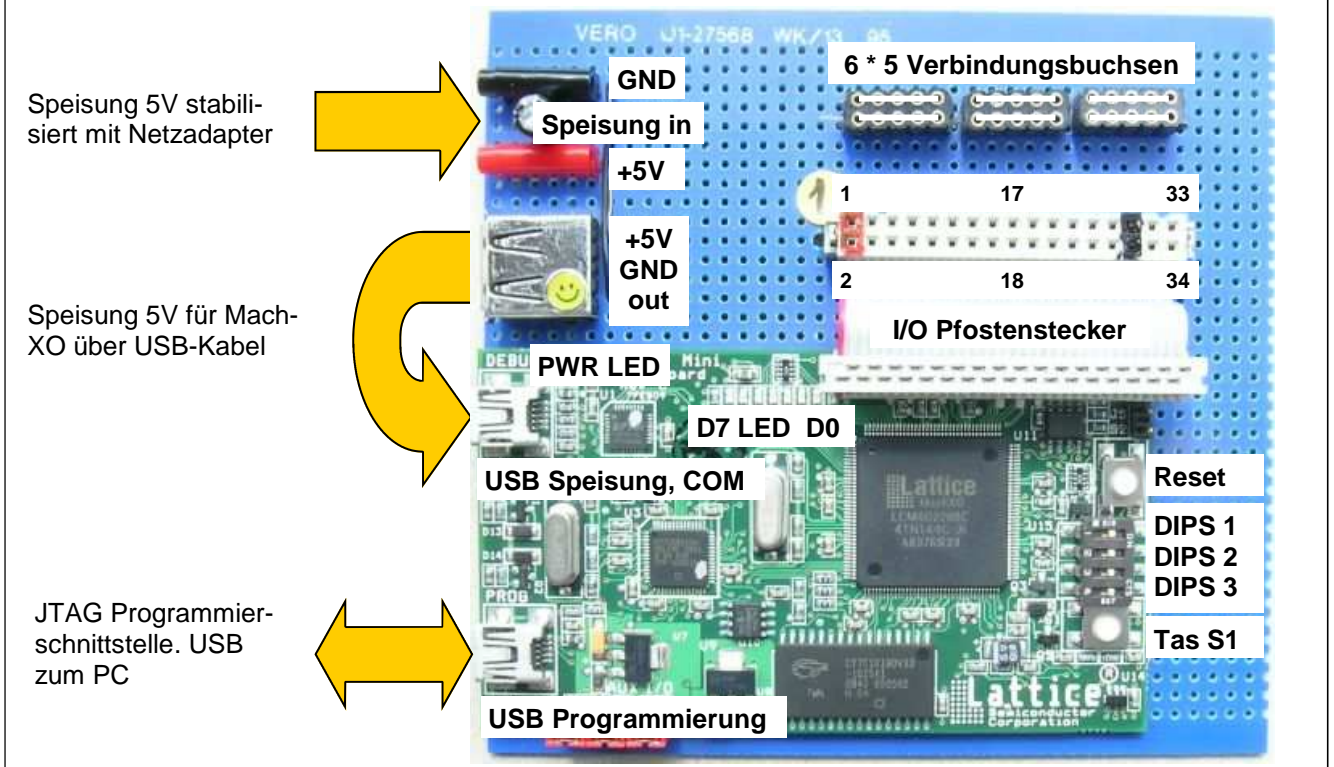
Nun ist die Software betriebsbereit. Die Lizenz ist für ein halbes Jahr gültig und muss danach erneuert werden.

# ASCII Tabelle

## ASCII (American Standard Code for Information Interchange)

Char	Hex	Dez	Char	Hex	Dez	Char	Hex	Dez	Char	Hex	Dez
NUL	0	0	Space	20	32	@	40	64	`	60	96
SOH	1	1	!	21	33	A	41	65	a	61	97
STX	2	2	"	22	34	B	42	66	b	62	98
ETX	3	3	#	23	35	C	43	67	c	63	99
EOT	4	4	\$	24	36	D	44	68	d	64	100
ENQ	5	5	%	25	37	E	45	69	e	65	101
ACK	6	6	&	26	38	F	46	70	f	66	102
Beep	7	7	'	27	39	G	47	71	g	67	103
Back sp	8	8	(	28	40	H	48	72	h	68	104
Tab	9	9	)	29	41	I	49	73	i	69	105
Linefeed	A	10	*	2A	42	J	4A	74	j	6A	106
VT	B	11	+	2B	43	K	4B	75	k	6B	107
Formfeed	C	12	,	2C	44	L	4C	76	l	6C	108
Return	D	13	-	2D	45	M	4D	77	m	6D	109
SO	E	14	.	2E	46	N	4E	78	n	6E	110
SI	F	15	/	2F	47	O	4F	79	o	6F	111
DLE	10	16	0	30	48	P	50	80	p	70	112
DC1	11	17	1	31	49	Q	51	81	q	71	113
DC2	12	18	2	32	50	R	52	82	r	72	114
DC3	13	19	3	33	51	S	53	83	s	73	115
DC4	14	20	4	34	52	T	54	84	t	74	116
NAK	15	21	5	35	53	U	55	85	u	75	117
SYN	16	22	6	36	54	V	56	86	v	76	118
ETB	17	23	7	37	55	W	57	87	w	77	119
CAN	18	24	8	38	56	X	58	88	x	78	120
EM	19	25	9	39	57	Y	59	89	y	79	121
SUB	1A	26	:	3A	58	Z	5A	90	z	7A	122
Escape	1B	27	;	3B	59	[	5B	91	{	7B	123
FS	1C	28	<	3C	60	\	5C	92		7C	124
GS	1D	29	=	3D	61	]	5D	93	}	7D	125
RS	1E	30	>	3E	62	^	5E	94	~	7E	126
US	1F	31	?	3F	63	-	5F	95	DEL	7F	127

# Anschlussbelegung MachXO Mini



Belegung I/O Pfostenstecker mit MachXO Pins					
XOPin	Signal	Stecker		Signal	XOPin
3.3V	Vcc	2	1	Vcc	3.3V
119	I/O	4	3	I/O	80
116	I/O	6	5	I/O	78
114	I/O	8	7	I/O	79
113	I/O	10	9	I/O	76
112	I/O	12	11	I/O	77
111	I/O	14	13	I/O	74
110	I/O	16	15	I/O	75
109	I/O	18	17	I/O	73
127	Clk 48MHz	20	19	I2C Alert	120
87	SPI SS	22	21	I2C Dat	122
85	SPI CLK	24	23	I2C Clock	121
81	SPI MOSI	26	25	Reserved	
84	SPI MISO	28	27	Reserved	
0V	GND	30	29	GND	0V
(108)	PWM out	32	31	Analog IN	(106)
	leer	34	33	leer	

Weitere MachXO Anschlüsse	
XOPin	Signal
137	LED rot D0 (nullaktiv)
138	LED rot D1 (nullaktiv)
139	LED rot D2 (nullaktiv)
140	LED rot D3 (nullaktiv)
141	LED rot D4 (nullaktiv)
142	LED rot D5 (nullaktiv)
143	LED rot D6 (nullaktiv)
144	LED rot D7 (nullaktiv)
14	Taster Tas S1 PullUp (nullaktiv)
70	Taster Tas S2 Reset
92	DIP Switch DIPS 1
91	DIP Switch DIPS 2
90	DIP Switch DIPS 3
7	FTDI 232 TXD
6	FTDI 232 RXD
5	FTDI 232 RTS#
4	FTDI 232 CTS#

# Multifunktion I/O-Platte

## Ausgänge:

Impuls Generator mit verschiedenen Frequenzen (2MHz, 125Hz, 2Hz), 2 Taster mit Umschaltkontakt, 8 Bit Schal-terzeile.

## Eingänge:

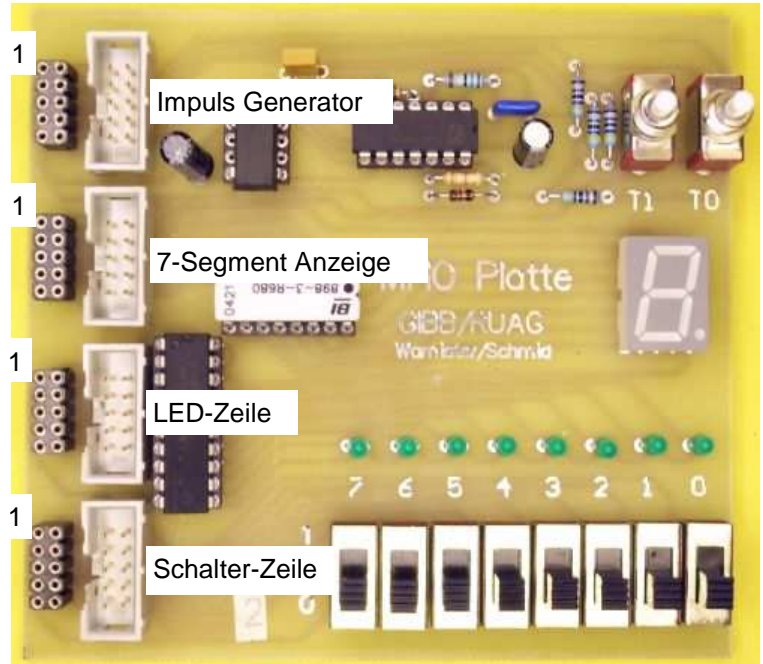
7-Segment Anzeige mit Dezimalpunkt, 8 Bit LED-Zeile

## Speisung:

3.3V bis 5V

## Anschlüsse:

10 polige Stecker mit parallel ange-schalteten Buchsen für den Anschluss mit Kabeln



### Impuls Generator

2	4	6	8	10
+5V	T0on	T1on	F2	F2M
1	3	5	7	9
GND	T0off	T1off	F125	PonR

1	GND	GND	
2	+5V	+5V	
3	T0off	Taste 0 aus	2 mal Taster 
4	T0on	Taste 0 ein	
5	T1off	Taste 1 aus	T0on, T1on
6	T1on	Taste 1 ein	T0off, T1off
7	F125	Astabiler MV, f ~ 125 Hz	
8	F2	Astabiler MV, f ~ 2 Hz	
9	Pon	Power on: '1' während ca. 300ms	
10	F2M	Astabiler Quarz-MV, 2MHz	

### 7-Segment Anzeige

2	4	6	8	10
+5V	b	d	f	dp
1	3	5	7	9
GND	a	c	e	g

1	GND	GND	
2	-	leer	7 LED-Segmente und LED-DP 
3	a	Segment a	
4	b	Segment b	
5	c	Segment c	
6	d	Segment d	
7	e	Segment e	
8	f	Segment f	
9	g	Segment g	
10	dp	Dezimalpunkt	

### LED-Zeile

2	4	6	8	10
+5V	L1	L3	L5	L7
1	3	5	7	9
GND	L0	L2	L4	L6

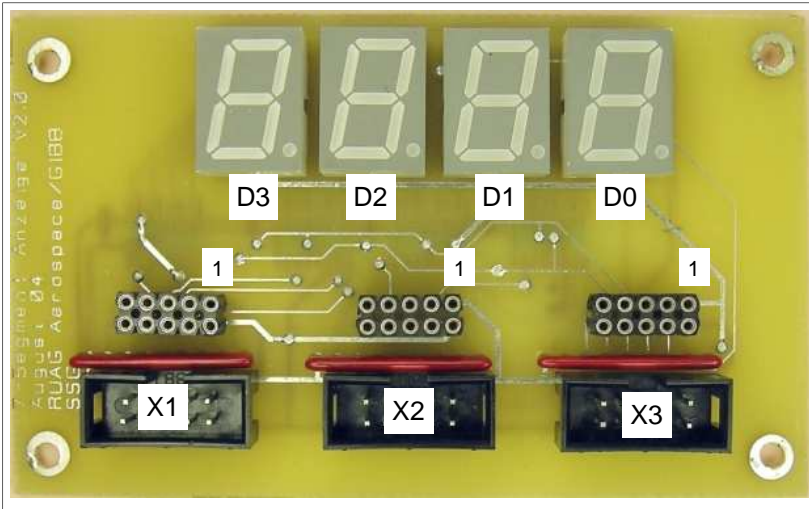
1	GND	GND	
2	+5V	+5V	8 mal Treiber und LED grün 5V, mit integriertem Vorwid. (15mA pro LED) 
3	L0	LED 0	
4	L1	LED 1	
5	L2	LED 2	
6	L3	LED 3	
7	L4	LED 4	
8	L5	LED 5	
9	L6	LED 6	
10	L7	LED 7	

### Schalter-Zeile

2	4	6	8	10
+5V	S1	S3	S5	S7
1	3	5	7	9
GND	S0	S2	S4	S6

1	GND	GND	
2	+5V	+5V	8 mal Schiebeschalter 
3	S0	Schiebeschalter 0	
4	S1	Schiebeschalter 1	
5	S2	Schiebeschalter 2	
6	S3	Schiebeschalter 3	
7	S4	Schiebeschalter 4	
8	S5	Schiebeschalter 5	
9	S6	Schiebeschalter 6	
10	S7	Schiebeschalter 7	

# 4 Digit BCD 7-Segment LED-Anzeige



D	C	B	A	Digit
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9

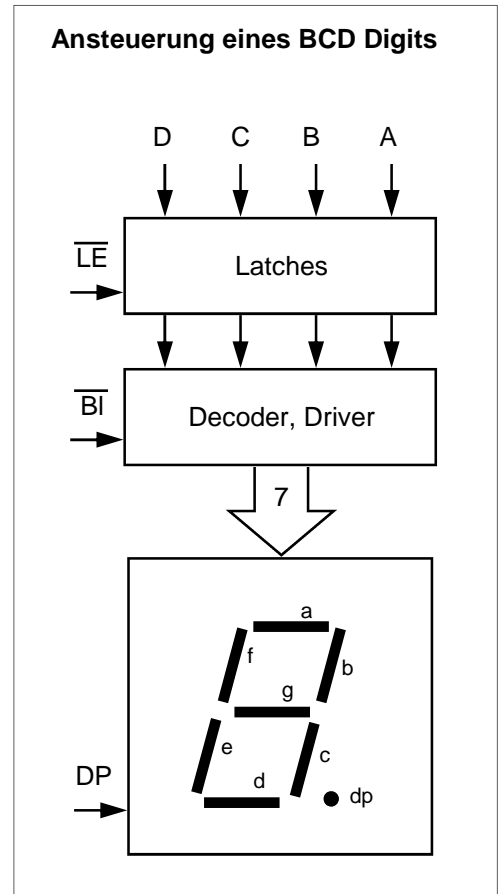
X1 MSB Digits					1	GND
2	4	6	8	10	2	+5V
+5V	D2 B	D2 D	D3 B	D3 D	3	D2 A
1	3	5	7	9	4	D2 B
GND	D2 A	D2 C	D3 A	D3 C	5	D2 C
					6	D2 D
					7	D3 A
					8	D3 B
					9	D3 C
					10	D3 D

X2 LSB Digits					1	GND
2	4	6	8	10	2	+5V
+5V	D0 B	D0 D	D1 B	D1 D	3	D0 A
1	3	5	7	9	4	D0 B
GND	D0 A	D0 C	D1 A	D1 C	5	D0 C
					6	D0 D
					7	D1 A
					8	D1 B
					9	D1 C
					10	D1 D

X3 Dezimalpunkte DP, Control					1	GND
2	4	6	8	10	2	+5V
+5V	DP 1	DP 3	LE'	-	3	DP 0
1	3	5	7	9	4	DP 1
GND	DP 0	DP 2	BI'	-	5	DP 2
					6	DP 3
					7	BI'
					8	LE'
					9	-
					10	-



**Signale: Bezeichnungen, Funktion:**

- D3 B** Digit 3, BCD Signal B
- DP 0** Dezimalpunkt am Digit D0 (aktiv 1)
- BI'** Blank Input BI' = 0: Digits aus, BI' = 1: Digits ein (für Helligkeitssteuerung mit PWM)
- LE'** Latch enable. LE' = 0: D-FF sind transparent, LE' = 1: Eingangszustände bei der positiven Flanke werden in den D-FF gespeichert und erscheinen am Ausgang.
- Bem:** BI und LE sind für alle 4 Digits gemeinsam (parallel) geschaltet.