Development of electronic compounds for DFPOP Prototype III

Exemplary new designed and developed components:

Basic Tiger board

Stepper-motor-control

new Photo counter and control module with a modern Mitsubishi Microcontroller M16C62P

by

Christof Ermer

University Regensburg / Germany

Basic Tiger Modul Board

Reflection:

Not everybody can be a good scientist, a good C-language programmer, a good electronic specialist, a good concept designer and, of course, a mechanic genie. But the practical requests are really complicated and it is necessary to bring together this requests.

(one possible)

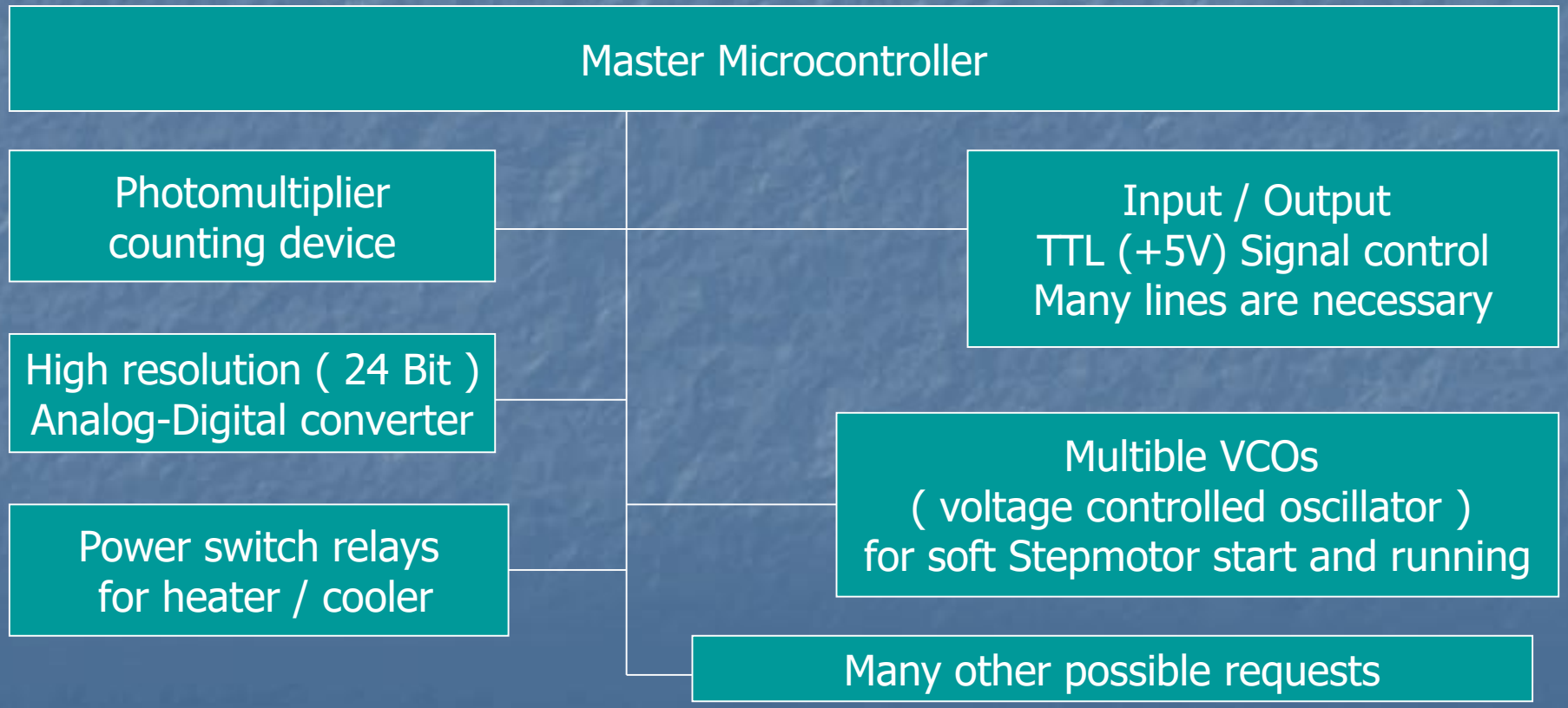
Solution:

We need also a simple to use Controller

That is now realized with the Basic Tiger Module board with a included ready to use, self kitted software (Firmware).

Environment and components in measure-instruments

In modern measure-instruments we have to do with the controlling of very different components.
This job is done by a microcontroller.



A reason to be happy !

The Tiger-Basic-board talks with you, doing what you want (really !) and doesn‘t contradict you !!

(…most of the time)

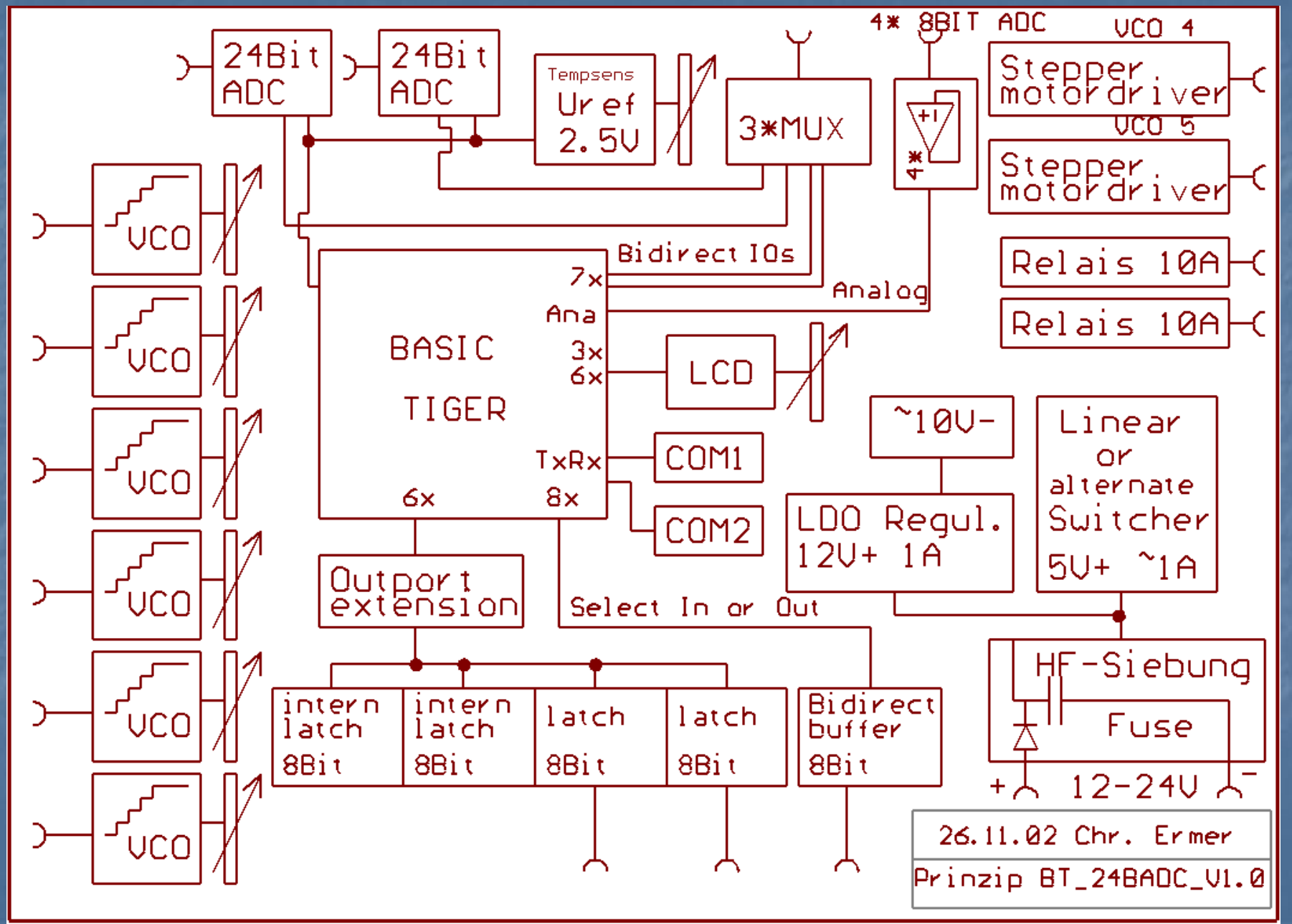
How it works:

- The Tiger Basic Module (in future called „TB“) is an integrated Device which provides a series of standard requests in electronic environment.
- Like: Input-Output lines with TTL-Level (+5V), Analog-Digital converter, LCD Display. Frequency generator and much more.
- {The wheel is made up. You must only use it.}

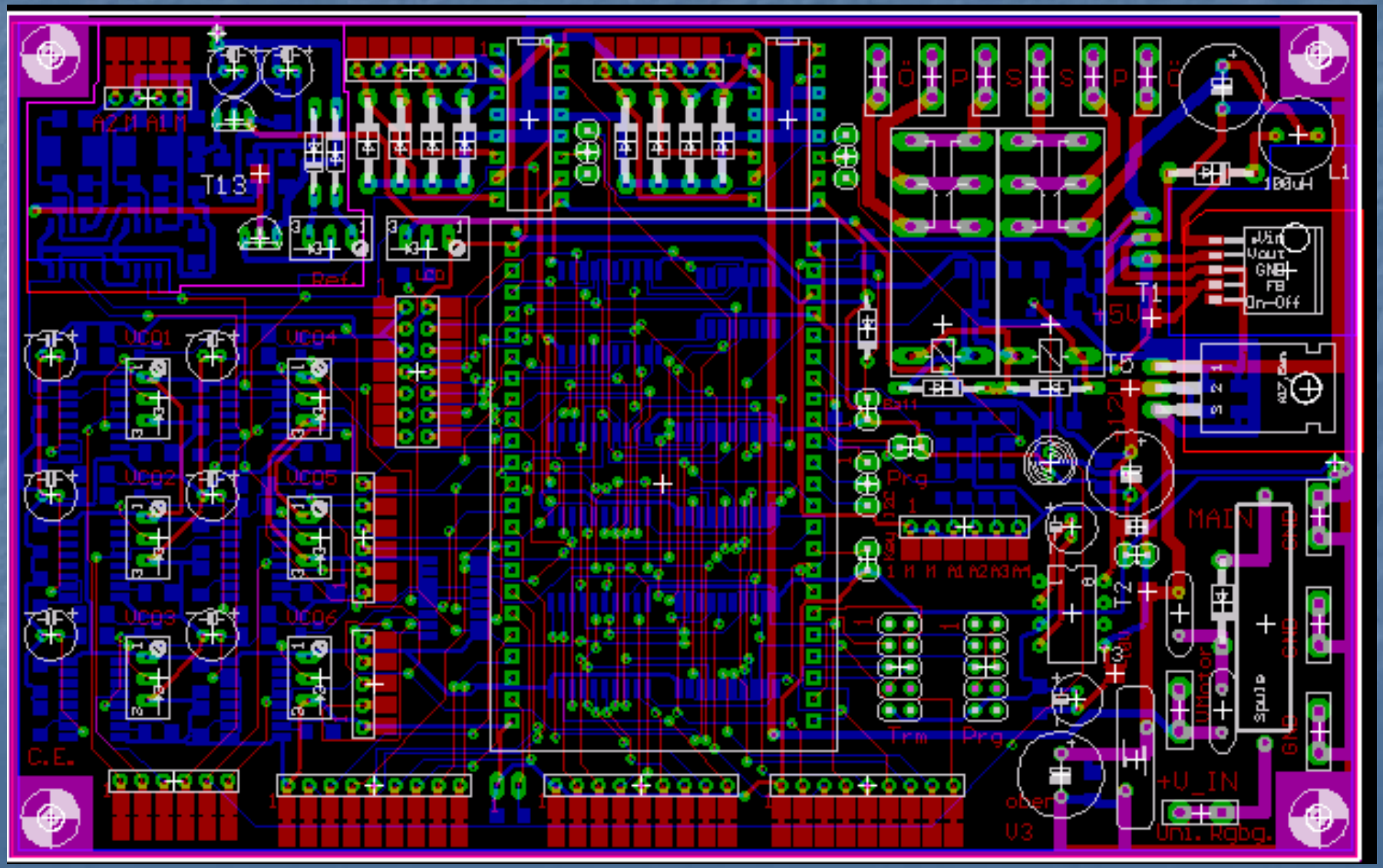
The self-developed control board with the TB Controller and the self written and adapted integrated software contains most of the requests for the DFPOP measure instrument.

The electronic components like VCO, I-O Lines, ADC, LCD Display and Power supply are arranged around the Tiger Basic module. →see following diagram

Conception of the Basic Tiger board



Viewing of board design

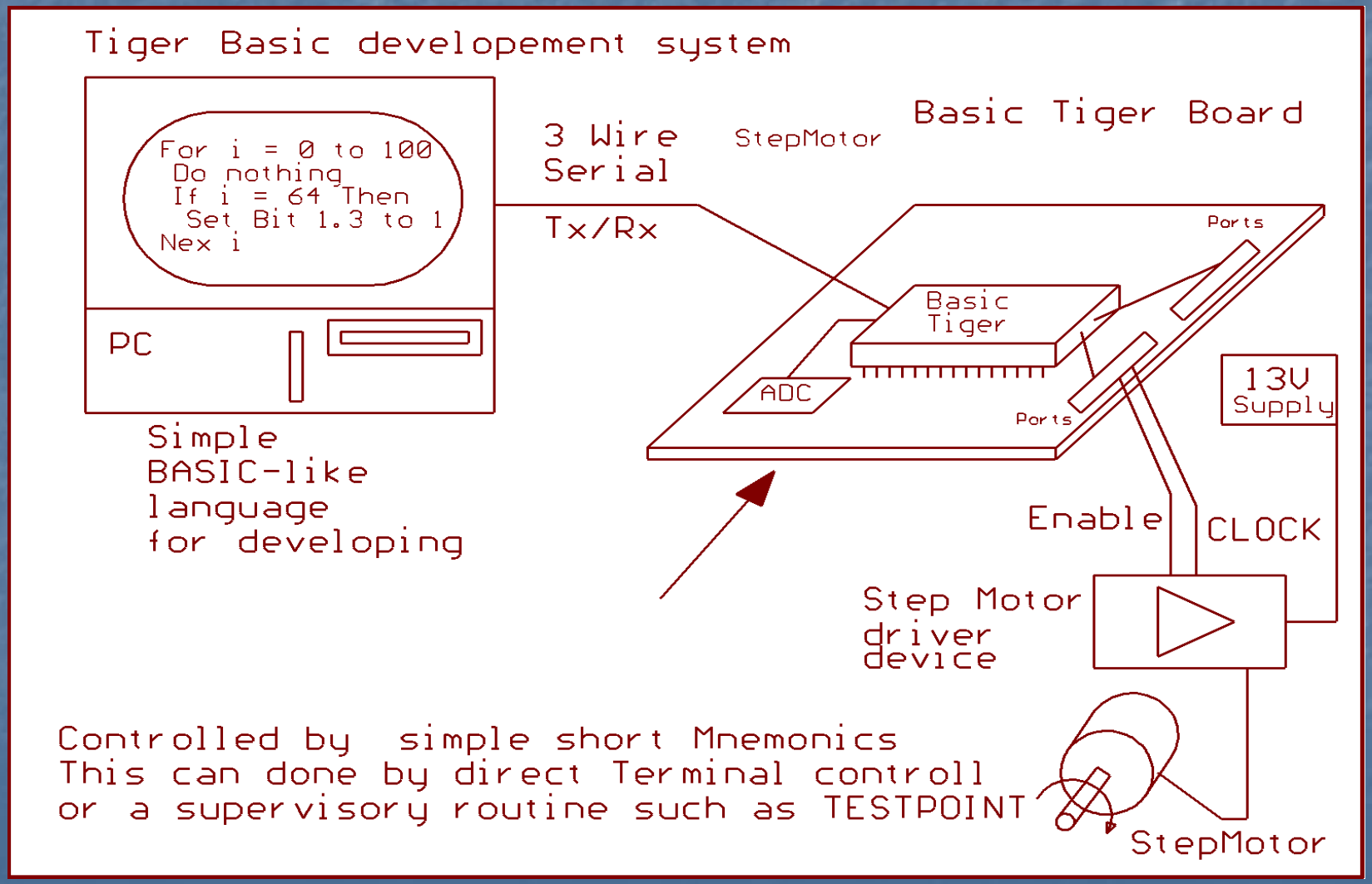


3D-Model of Basic Tiger Bord

Working and controlling using the Tiger-Board

- A simple 3-wire and string based serial communication between the TB-Bord is necessary and established.
- You talk to the TB in the simplest case with an ordinary terminal program.
- A windows control software like „Testpoint“ is a good solution to automate working steps.
- The results can be displayed and interpreted simply on the local PC.

Simple steps to using the Tiger Basic Board.



Example: Using included ordercodes (mnemonics)

- We want to check the availability or the lifesigns of the TB Board
- You send the order
- „Ping“ and a „carriage return“ (CR).
- Under correct conditions the TB answers with
- „Pong“ and a „carriage return“ (CR) too.
- Afterwards the TB waits for the next order signing a questionmark
- „?“ (means ...awaiting next order)
-

Oders are listed in the build-in menu to prevent loss of documentaion

- A big selection of orders are realised in the current firmware-
- To show the possible orders type „??“+CR
- Then you get a list of all most used orders and sequences

Some Order examples

- oXb => (X=Out Port 1..2)put BYTE on external port
- M1LO M1RO MO1F, M2LO M2RO MO2F, M4LO M4RO
- uXu => (X=ADC1..2) 1* voltage measurement
- vXo => (X=VCO 1..6) ON
- mXf => (X=MOTOR 1..2) OFF
- iadX => (X=ADC1..4) intern 10Bit
- rXo => (X=Relays 1..2)ON
- rXf => (X=Relays 1..2)OFF
- INP8P = { whole byte } of fixed direction InPort_8

How it looks in BASIC code

```
PRINT #SER, #0, " MENUE "
'PRINT #SER, #0, " PCOOLO => Cooler ON, PCOOLF => Cooler OFF"
'PRINT #SER, #0, " HEATO => HEAT ON HEATF => HEAT OFF"
PRINT #SER, #0, " TCOH  => Anzeige 'ToCoolOrHeat  <> 20 Grad OR OK"
PRINT #SER, #0, " M1LO M1RO MO1F, M2LO M2RO MO2F, M4LO M4RO MO4F" ' Motor 1,2,4 LINKS/RECHTS On OFF
PRINT #SER, #0, " SNNCLK=>Set -->NN*CLK Clocks on M3. VORHER 1* EINSTELLEN !! (default 180)"
PRINT #SER, #0, " NNSPEED => Set -->SNN. VORHER 1* EINSTELLEN !!(default 0)Speed of M3)"
PRINT #SER, #0, " M3LO, M3RO => MOTOR 3 LeftSpin ON mit NN*CLK. Stop=MO3F"
PRINT #SER, #0, " scoolontime or scoolofftime => in MS. Default 30s 20s is used"
PRINT #SER, #0, " shysterese =>(0..255)(default=2 NTC Vals)"
PRINT #SER, #0, " S20GRADNTC => of ADC1. Default = 584"
PRINT #SER, #0, " gettemp => Grad C of NTC"
PRINT #SER, #0, " uXu => (X=ADC1..2) 1* voltage measurement"
PRINT #SER, #0, " uXl => (X=ADC1..2) 1* TX Long CVal voltage "
PRINT #SER, #0, " rXo => (X=Relais 1..2)ON"
PRINT #SER, #0, " rXf => (X=Relais 1..2)OFF"
PRINT #SER, #0, " vXo => (X=VCO 1..6) ON"
PRINT #SER, #0, " vXf => (X=VCO 1..6) OFF"
PRINT #SER, #0, " mXo => (X=MOTOR 1..2) ON"
PRINT #SER, #0, " mXf => (X=MOTOR 1..2) OFF"
PRINT #SER, #0, " oXb => (X=OutPort 1..2)put BYTE on external port (1 Steps)"
PRINT #SER, #0, " oXt => (X=OutPort 1..2) put Bit on external port (2 Steps)"
PRINT #SER, #0, " cXY => (X=OutPort 1..2, Y=Biz 0..7) Clear bit Y on outport X"
PRINT #SER, #0, " SXY => (X=OutPort 1..2, Y=Biz 0..7) Set bit Y on outport X"
PRINT #SER, #0, " iadX => (X=ADC1..4) intern 10Bit"
PRINT #SER, #0, " #iadX => (X=ADC1..4) intern 10Bit, Reine Zahl ohne Liste"
PRINT #SER, #0, " inp8X => (X=bit 0..7) 1Bit of fixed direction InPort_8"
PRINT #SER, #0, " INP8P = ganzes BYte } of fixed direction InPort_8"
PRINT #SER, #0, " frq7XY = {X=2.3} {Y=A..B} Frequency of Port72/72 A or B"
PRINT #SER, #0, " cpwm7X => (X=PWM Port 2..3) Clear switch L72/L73 on J16"
PRINT #SER, #0, " spwm7X => (X=PWM Port 2..3) Set switch L72/L73 on J16"
PRINT #SER, #0, " pwmX => (X=1..2) PWM an Port L72/73) (2 Steps) (2.=Bytewert)"
PRINT #SER, #0, " Q => Quit"
PRINT #SER, #0, " "
PRINT #SER, #0, "Order with Pre # Bsp: #U1L send value without delimiter list"
PRINT #SER, #0, " ?? => Show command menue again. OR press m,?,??"
PRINT #SER, #0, " "
PRINT #SER, #0, " Please select command"
PRINT #SER, #0, " "
```

A closer look to the electronic schematic

Some is simple, some not.

That‘s live.

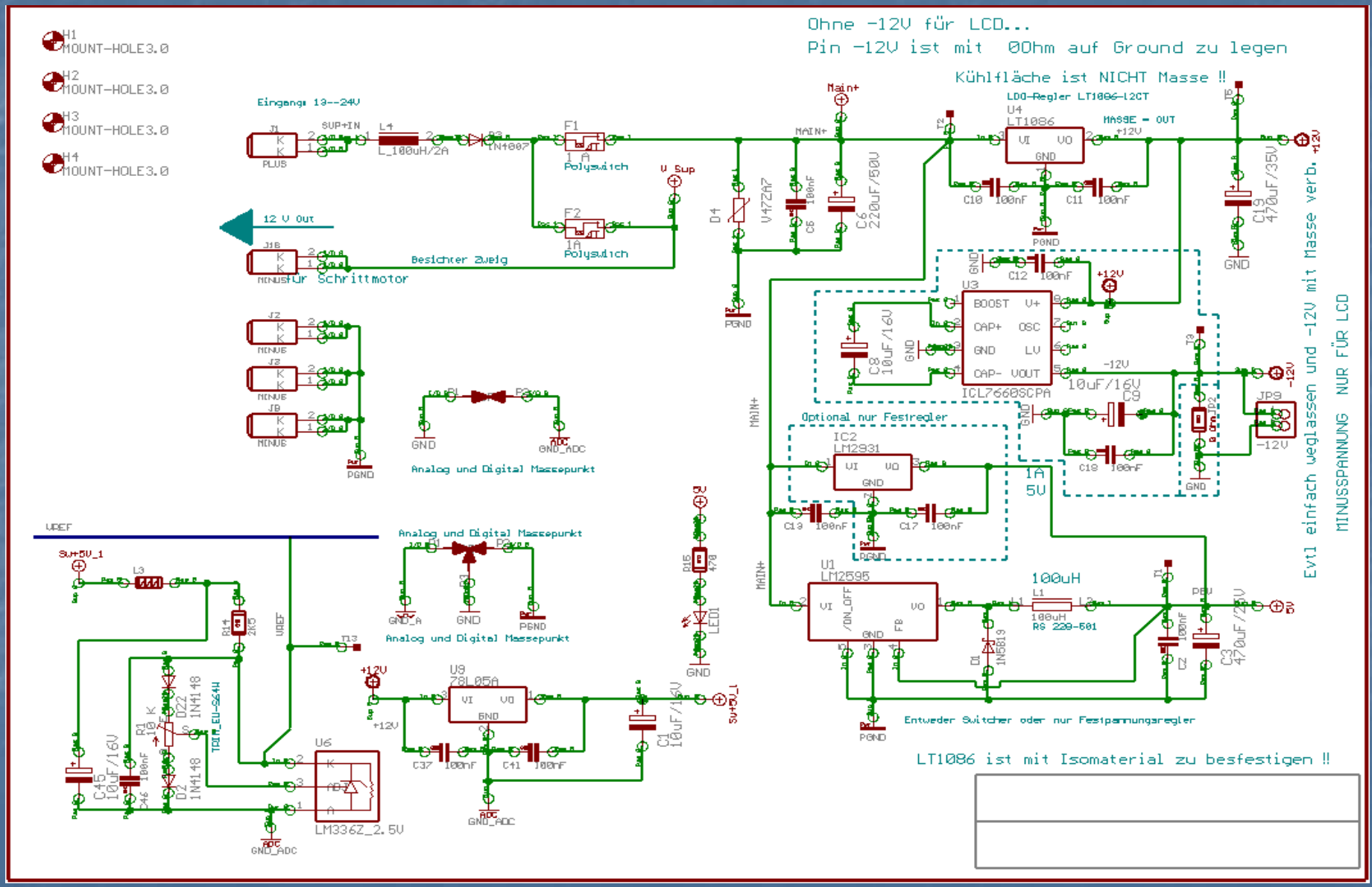
Don‘t be frightened. !

All people are cooking with water.

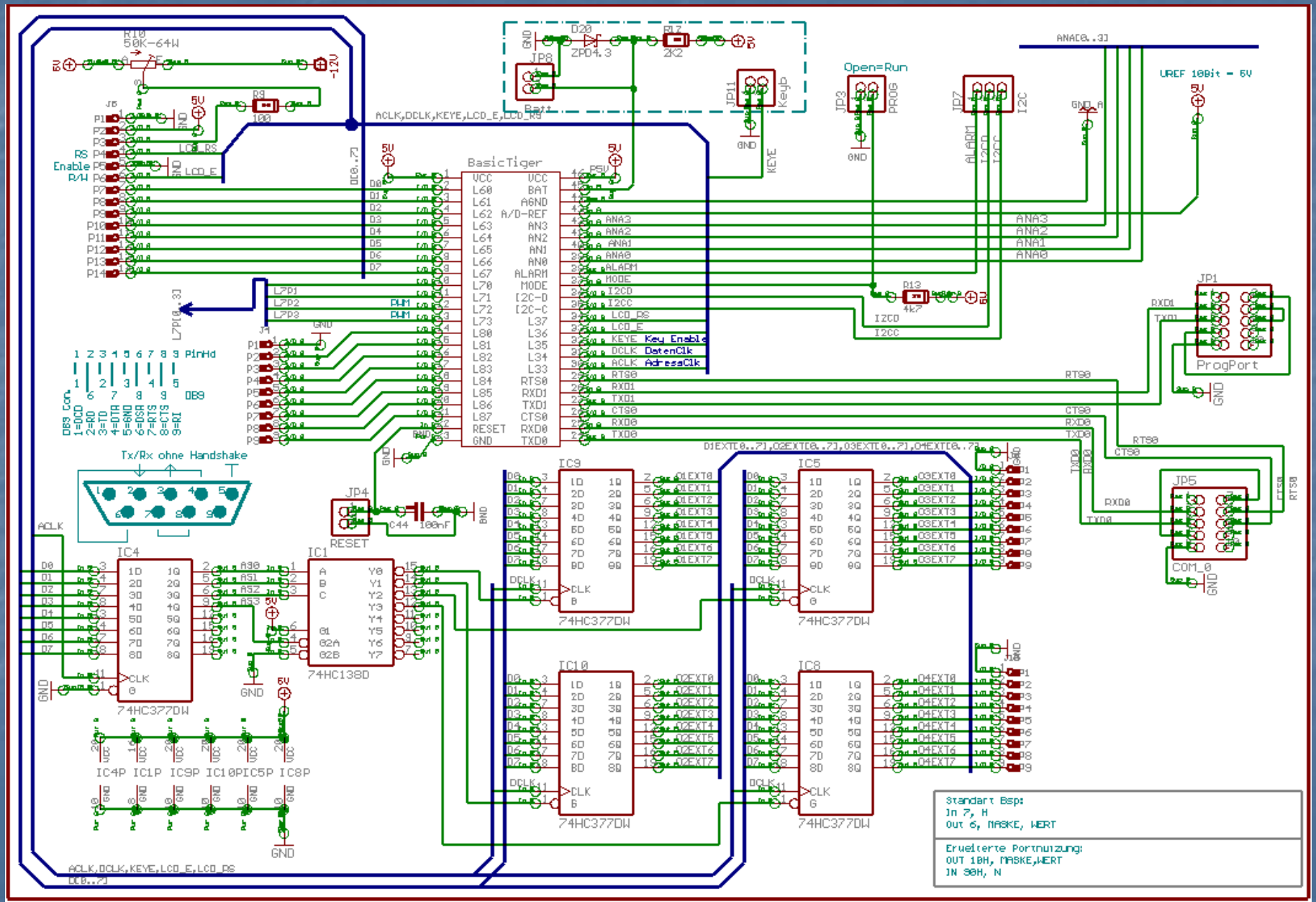
No magic is necessary.

A good design is beautiful and looks clear.

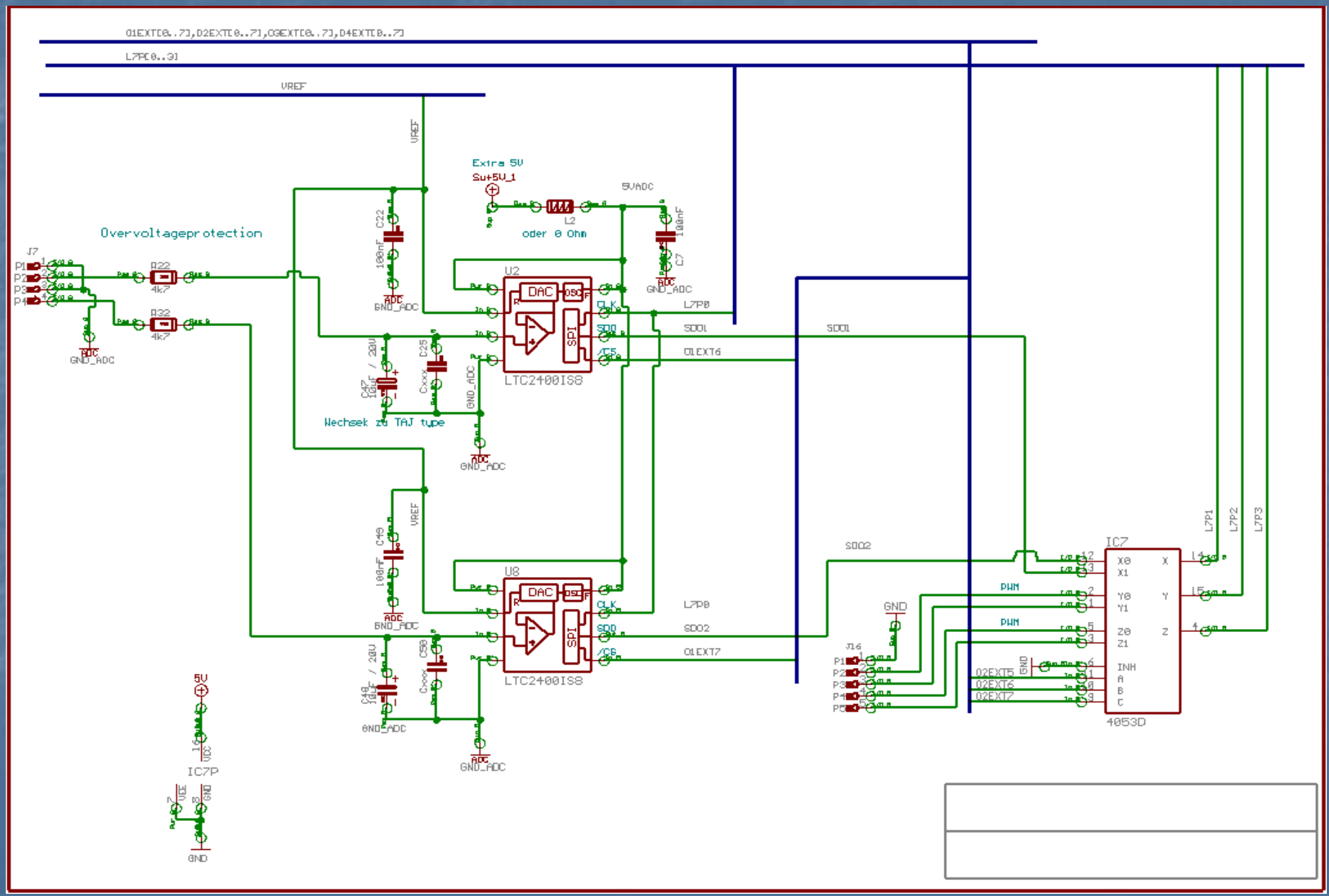
Schematic View 1of 7

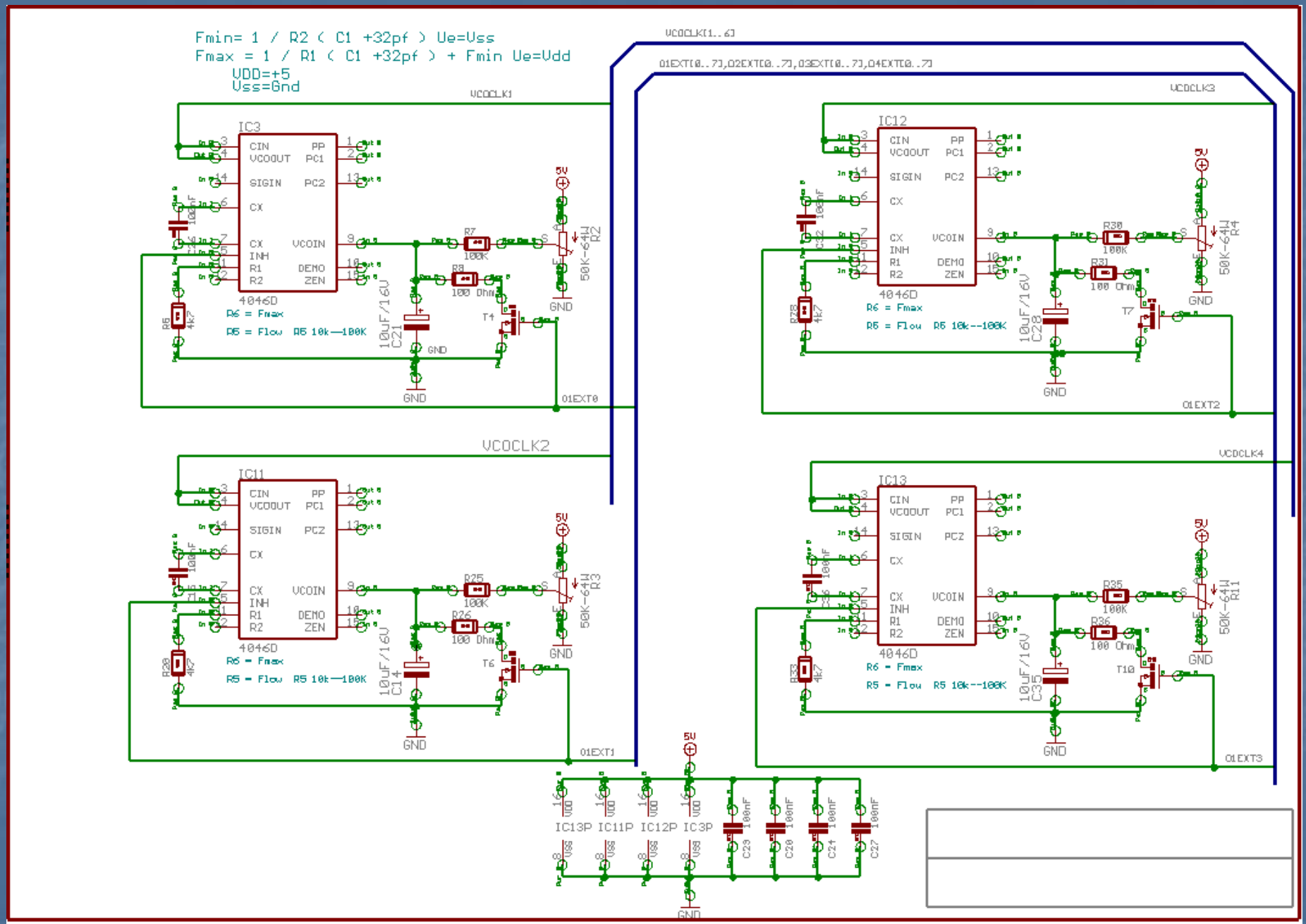


2 of 7

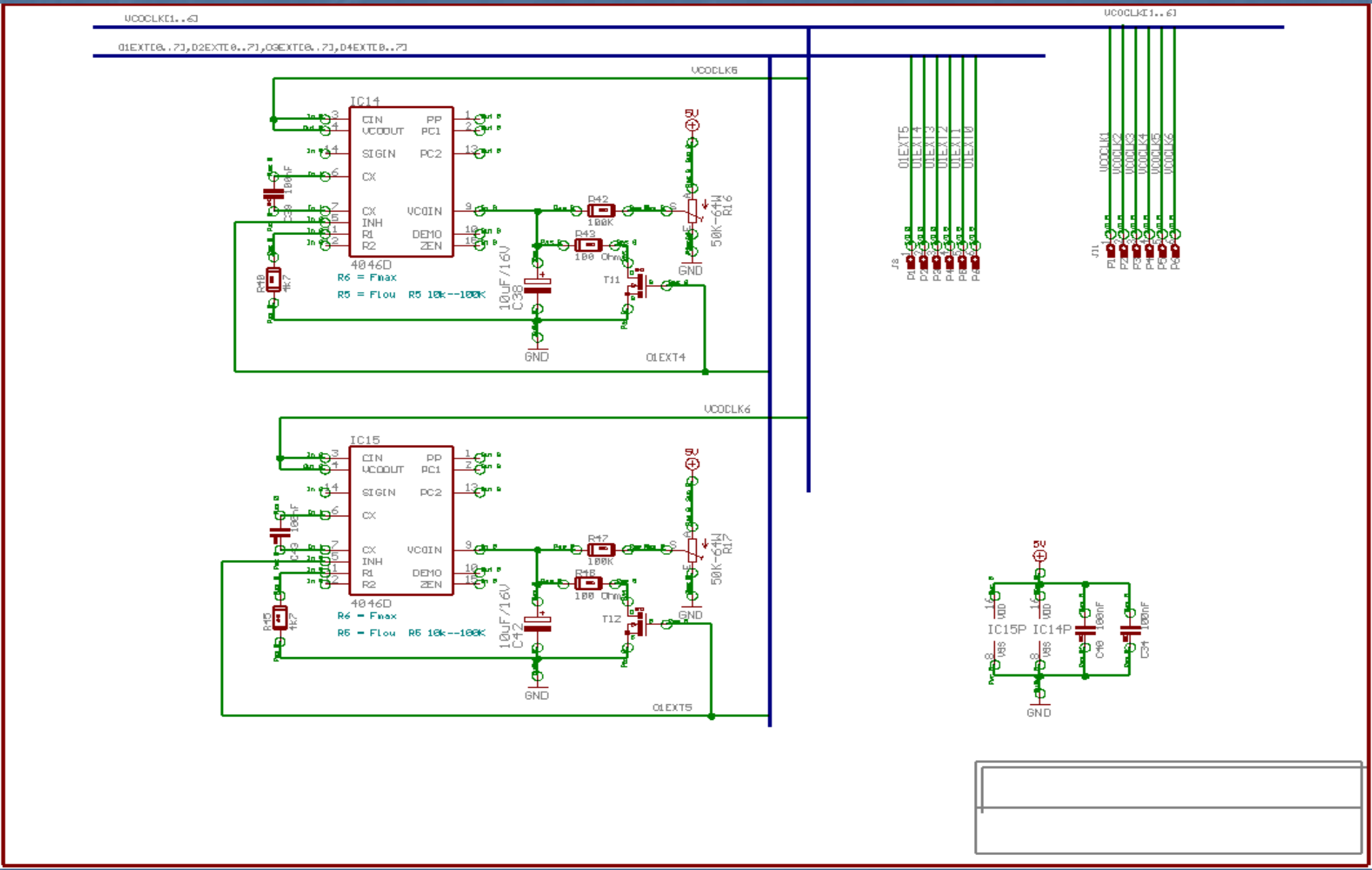

BasicTiger
ACLK,DCLK,KEYE,LCD_E,LCD_RS
ANA0..3
UREF 10Bit = 5V
Open=Run
Key Enable
DatenClk
AdressClk
ProgPort
Tx/Rx ohne Handshake
74HC377DW
74HC138D
IC4P IC1P IC9P IC10PIC5P IC8P
Standart Bsp:
In 7, H
Out 6, MASKE, WERT
Erweiterte Portnutzung:
OUT 1BH, MASKE,WERT
IN 30H, N

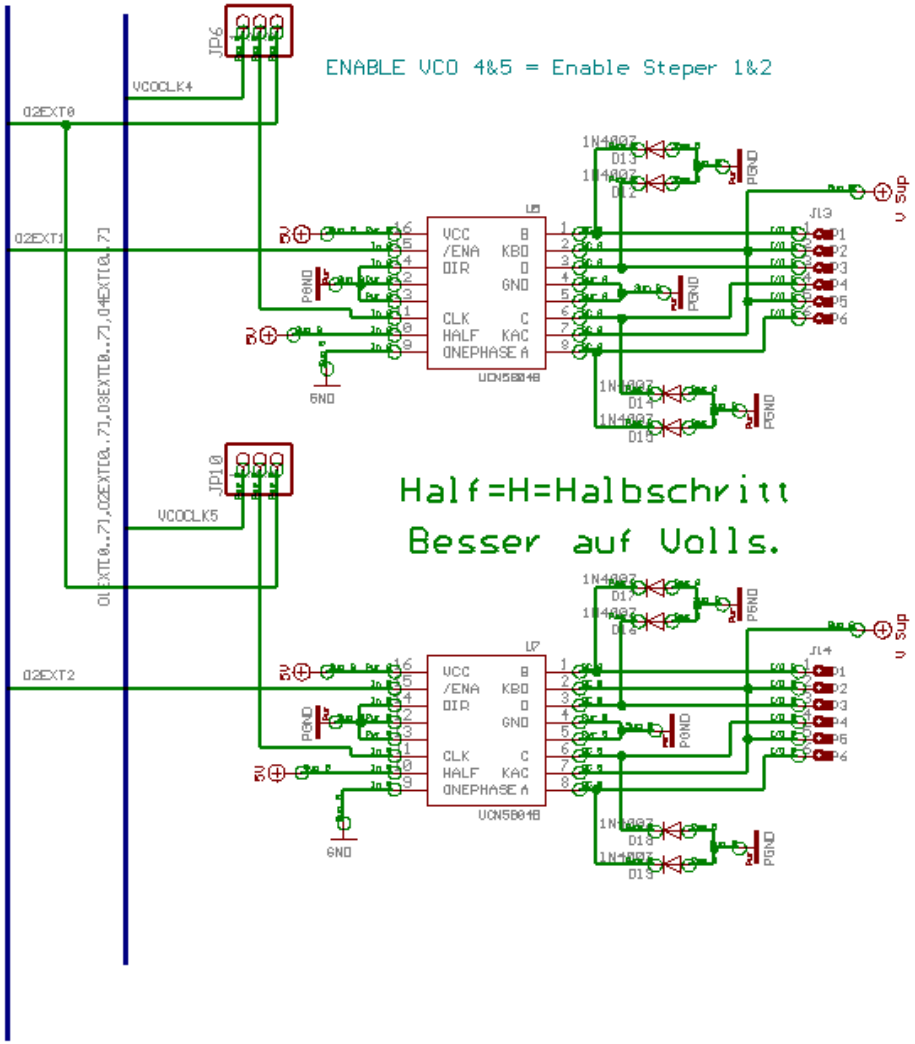
3 of 7

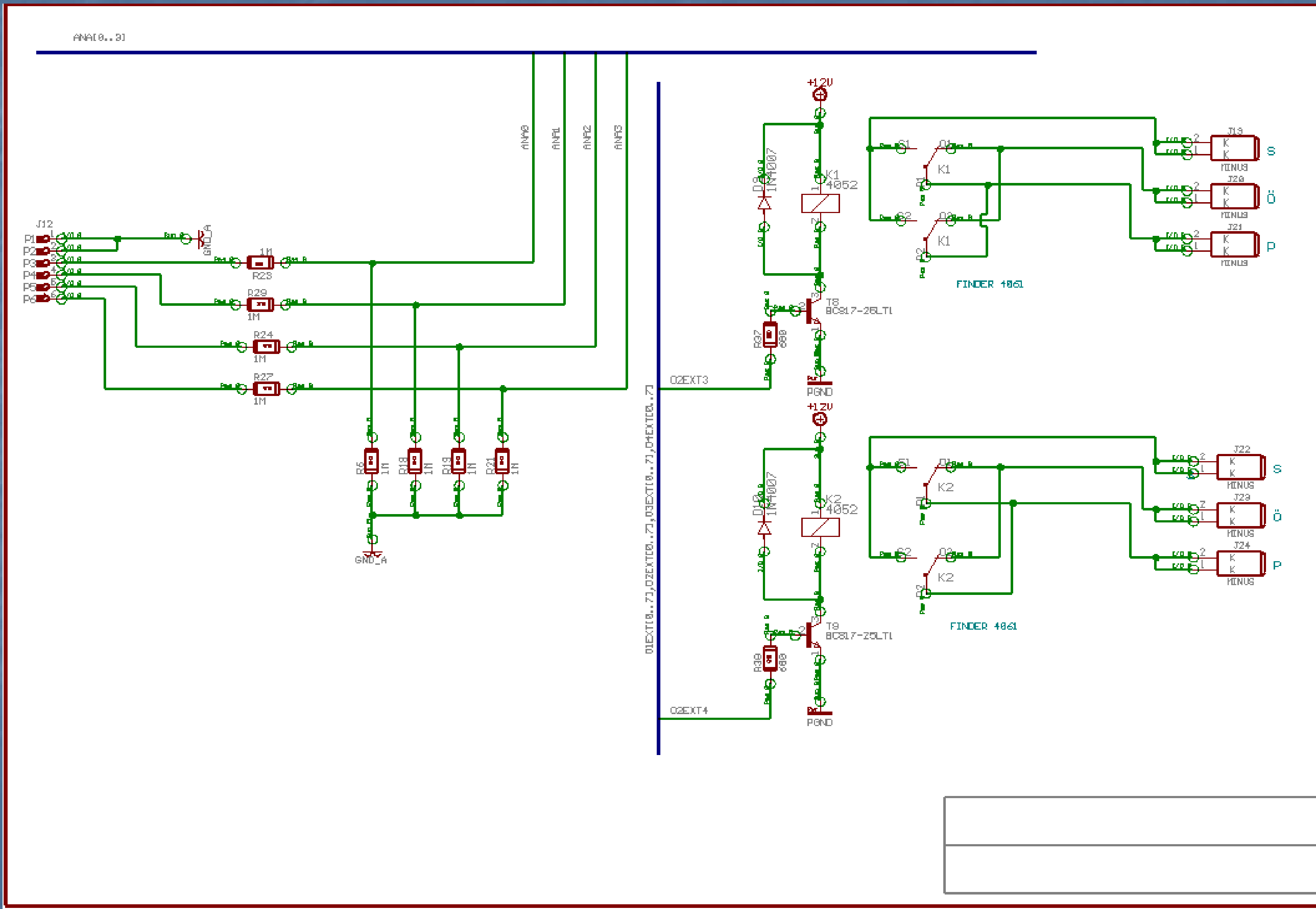

01EXT[0..7],02EXT[0..7],03EXT[0..7],04EXT[0..7]
L7P[0..3]
VREF
Extra 5V
Su+5V_1
oder 0 Ohm
5VADC
Overvoltageprotection
Wechsek zu TAJ type
U2
U8
LTC2400IS8
DAC
SPI
L7P0
SDO1
SDO2
01EXT6
01EXT7
PWM
GND
GND_ADC
IC7
4053D
IC7P
L7P1
L7P2
L7P3
02EXT5
02EXT6
02EXT7


Fmin= 1 / R2 (C1 +32pf) Ue=Vss
Fmax = 1 / R1 (C1 +32pf) + Fmin Ue=Vdd
VDD=+5
Vss=Gnd
VCOCLK1
VCOCLK2
VCOCLK3
VCOCLK4
VCOCLK[1..6]
01EXT[0..7],02EXT[0..7],03EXT[0..7],04EXT[0..7]
IC3
IC11
IC12
IC13
CIN
VCOOUT
SIGIN
CX
INH
R1
R2
PP
PC1
PC2
VCOIN
DEMO
ZEN
4046D
R6 = Fmax
R5 = Flow R5 10k—100K
R5 = Flow R5 10k--100K
10uF/16V
100nF
100K
100 Ohm
4k7
50K-64W
GND
5V
01EXT0
01EXT1
01EXT2
01EXT3
IC13P IC11P IC12P IC3P
VDD
Vss

5 of 7


VCOCLK[1..6]
D1EXT[0..7],D2EXT[0..7],D3EXT[0..7],D4EXT[0..7]
VCOCLK5
IC14
CIN
VCOOUT
SIGIN
CX
CX
INH
R1
R2
PP
PC1
PC2
VCOIN
DEMO
ZEN
4046D
R6 = Fmax
R5 = Flow R5 10k--100K
R42
100K
R43
100 Ohm
R16
50K-64W
5V
GND
10uF/16V
C38
T11
D1EXT4
VCOCLK6
IC15
R47
100K
R48
100 Ohm
R17
50K-64W
10uF/16V
C42
T12
D1EXT5
D1EXT5
D1EXT4
D1EXT3
D1EXT2
D1EXT1
D1EXT0
VCOCLK1
VCOCLK2
VCOCLK3
VCOCLK4
VCOCLK5
VCOCLK6
J8
J11
IC15P IC14P
VDD
VSS
C40 100nF
C34 100nF


ENABLE VCO 4&5 = Enable Steper 1&2
Half=H=Halbschritt
Besser auf Volls.
JP6
JP10
VCOCLK4
VCOCLK5
O2EXT0
O2EXT1
O2EXT2
O1EXT0..7],O2EXT0..7],O3EXT0..7],O4EXT0..7]
U5
U7
VCC
/ENA
DIR
CLK
HALF
ONEPHASE
B
KBD
D
GND
C
KAC
A
UCN5804B
1N4007
PGND
GND
J13
J14
Sup

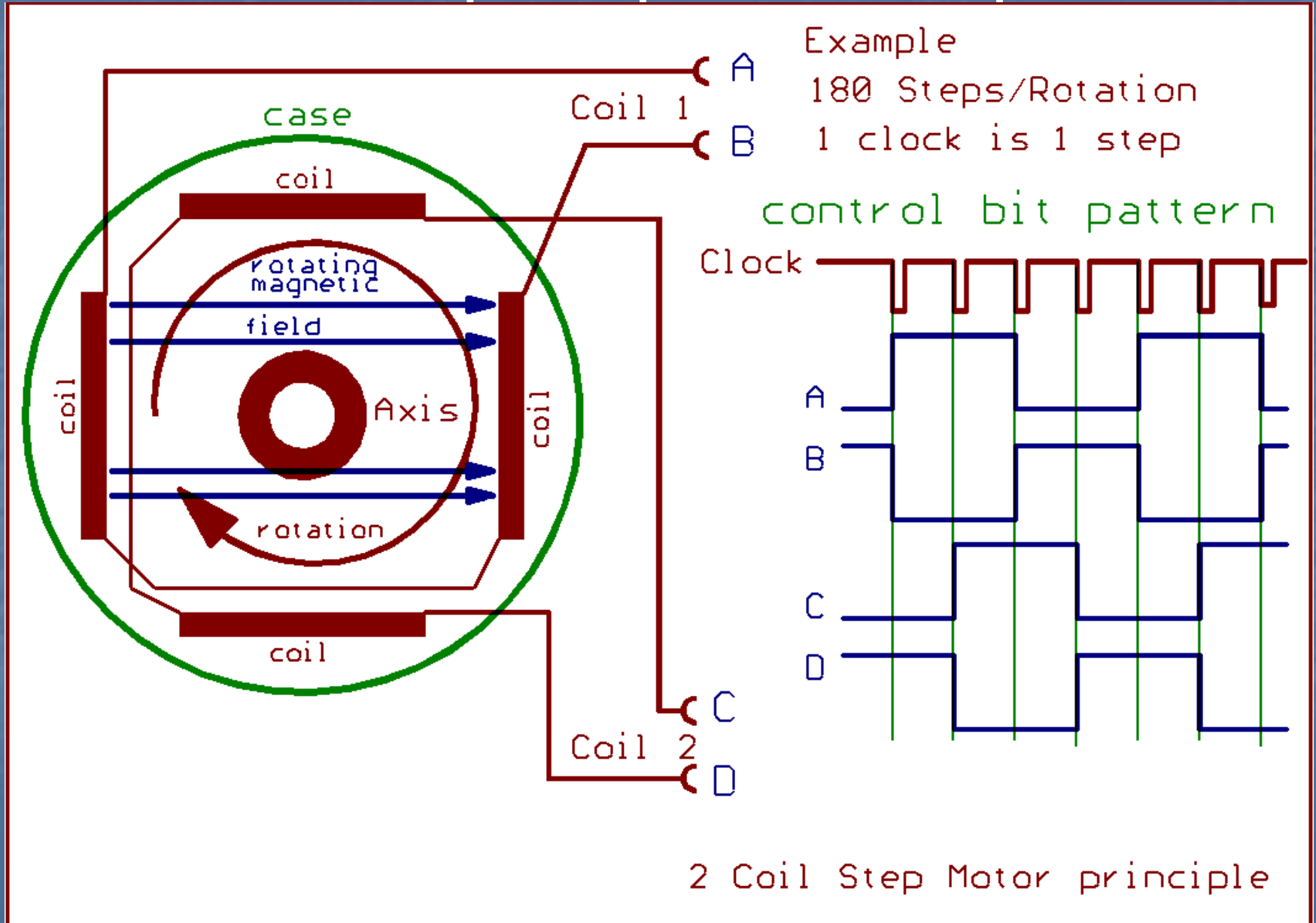

ANA[0..3]
ANA0
ANA1
ANA2
ANA3
J12
P1
P2
P3
P4
P5
P6
GND_A
R23
1M
R29
1M
R24
1M
R27
1M
R6
R18
R19
R21
O2EXT3
O2EXT4
+12V
K1
4052
K2
4052
1N4007
T8
BC817-25LT1
T9
BC817-25LT1
R37
R30
PGND
K1
K2
FINDER 4061
FINDER 4061
J13
J20
J21
J22
J23
J24
MINUS
S
Ö
P

Reasons for a self-developed Step-Motor power driver device

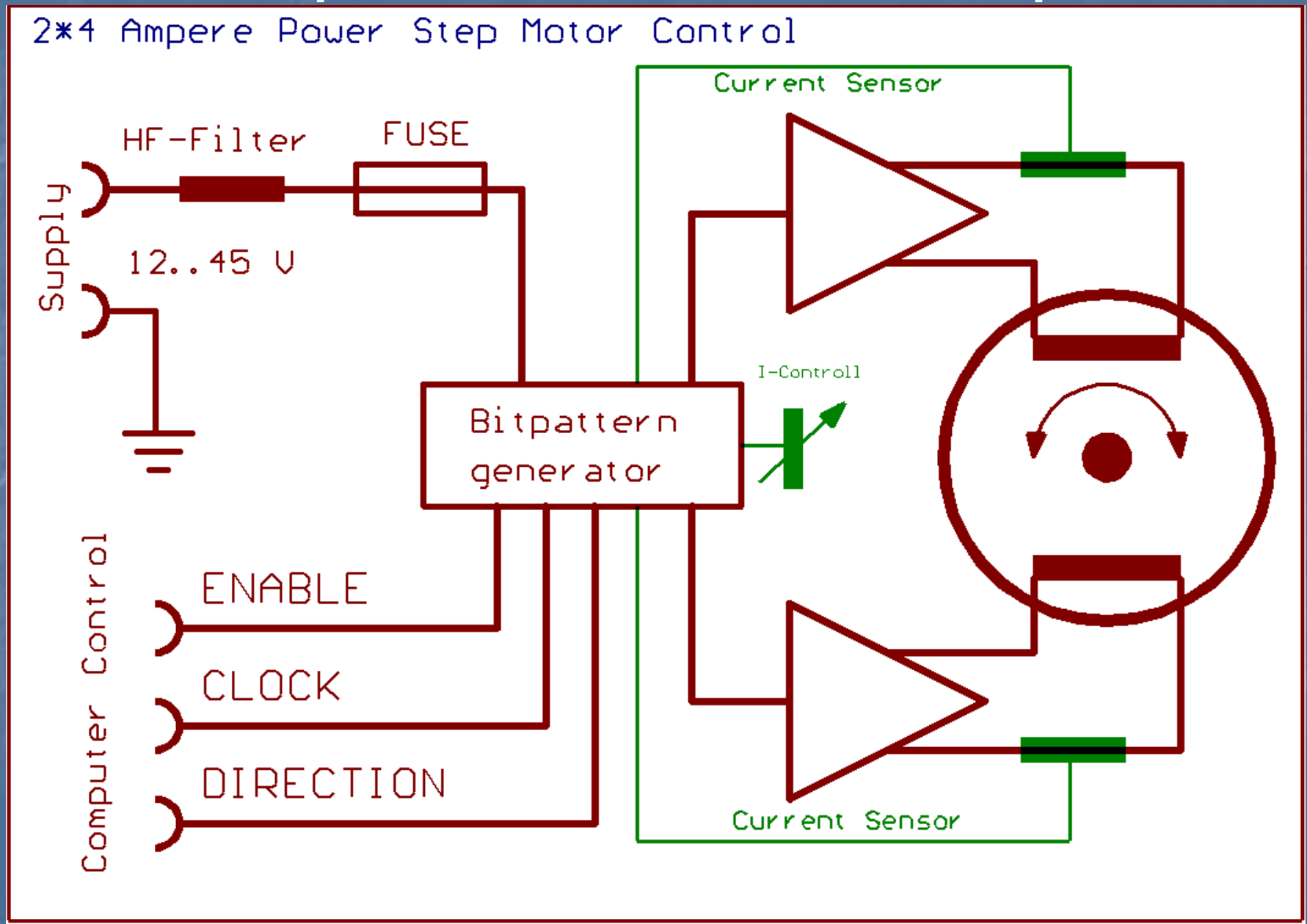
Purchasable step motor drivers are very expensive.

- The possible coil-current of many of these devices is to low for the used power supply
- The quality has in many cases not the demanded conditions.
- The mechanical dimensions are in many cases not optimal.
- To prevent start problems by the step motor, the internal resistance of the motor driver must be very low. That allows high current pulses for better starting conditions !.
- Resolution:
- If you want something to be done, do it yourself.

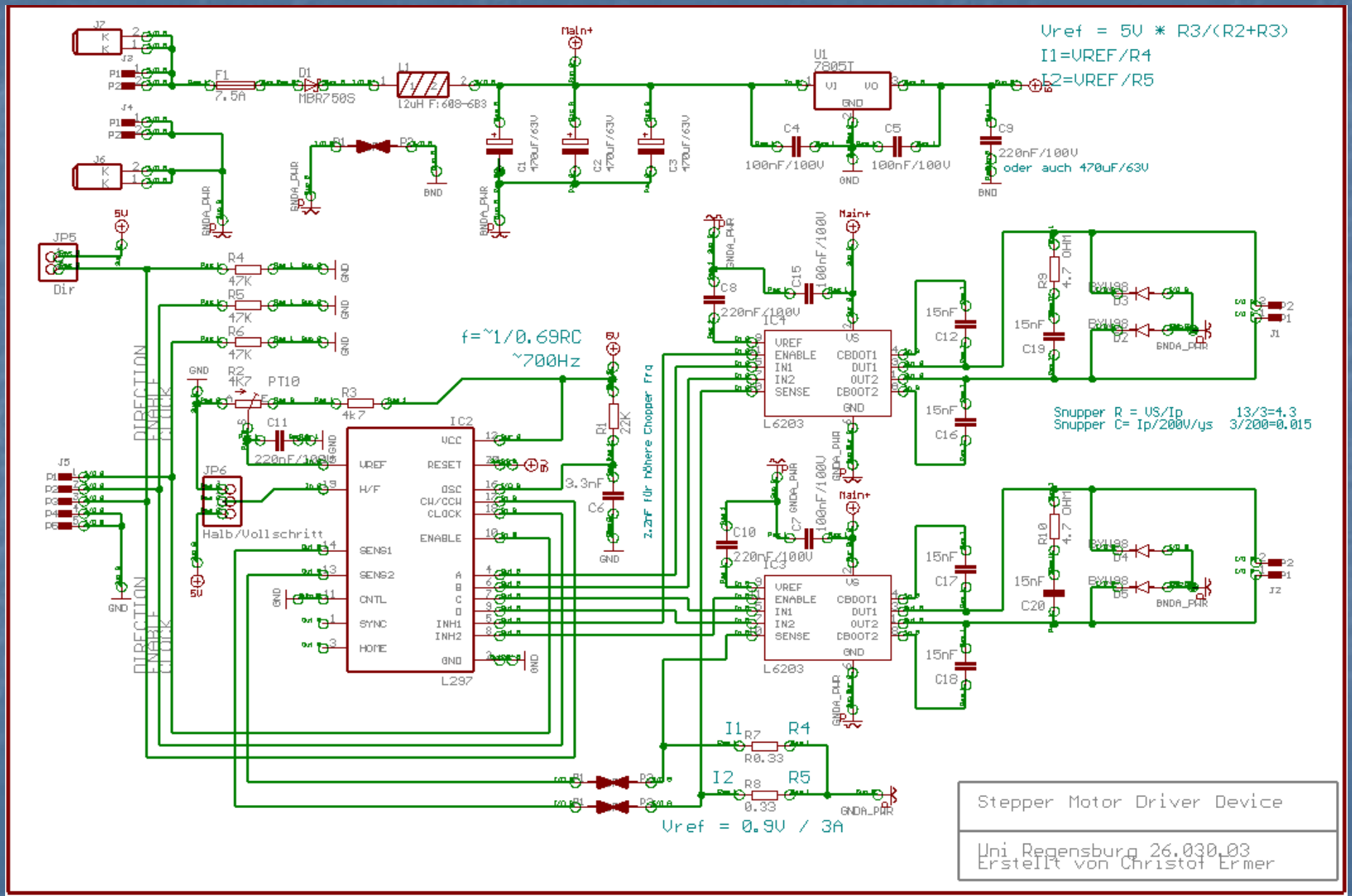
Construction principle of a step motor



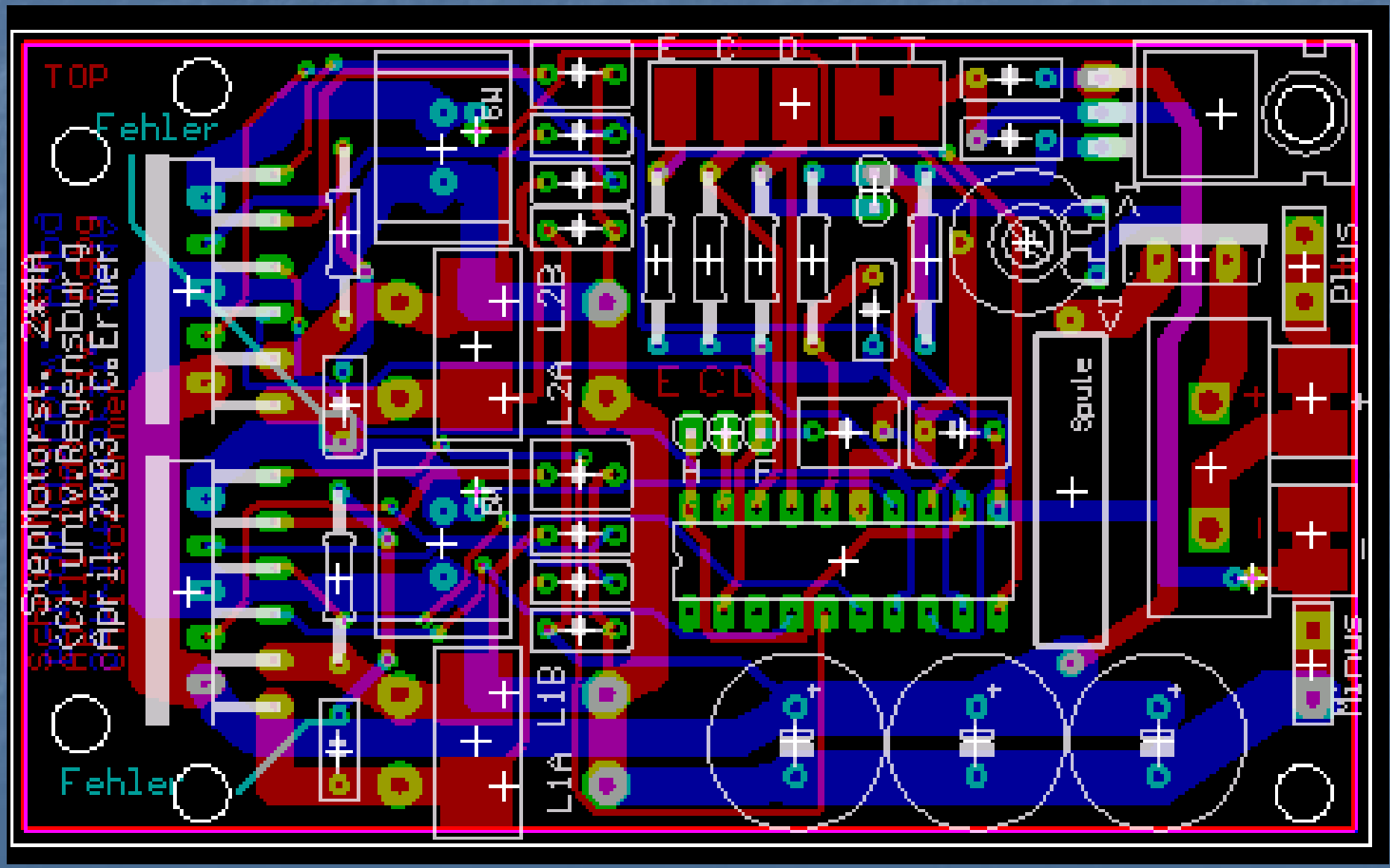
Step motor driver concept



Stepmotor driver schematic

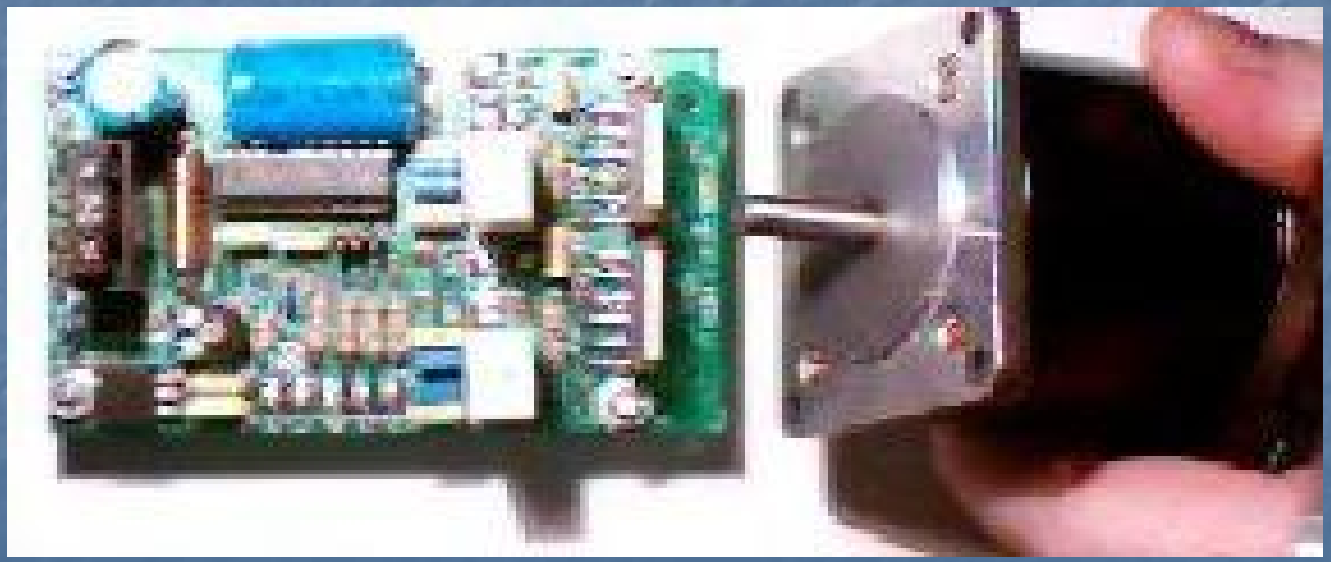


Board design



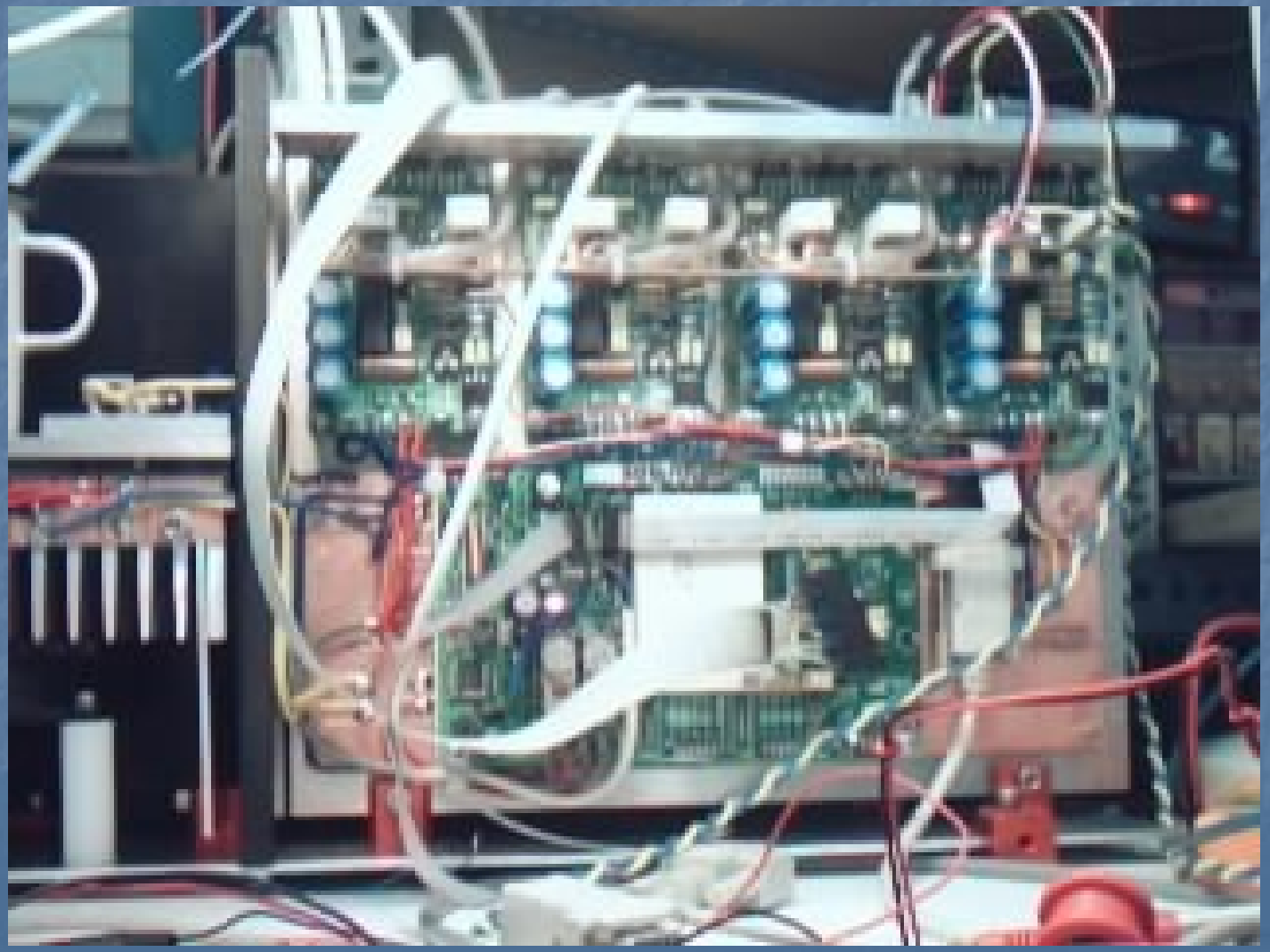
3D Viewing of the Stepmotor board

View on the 2*4 Ampere Step motor control



The control has the same width as the most used step motors

Example: Device with the Basic-Tiger board and 4 stepmotor driver in developing state



The new CPU Mitsubishi M16C62P Microcontroller

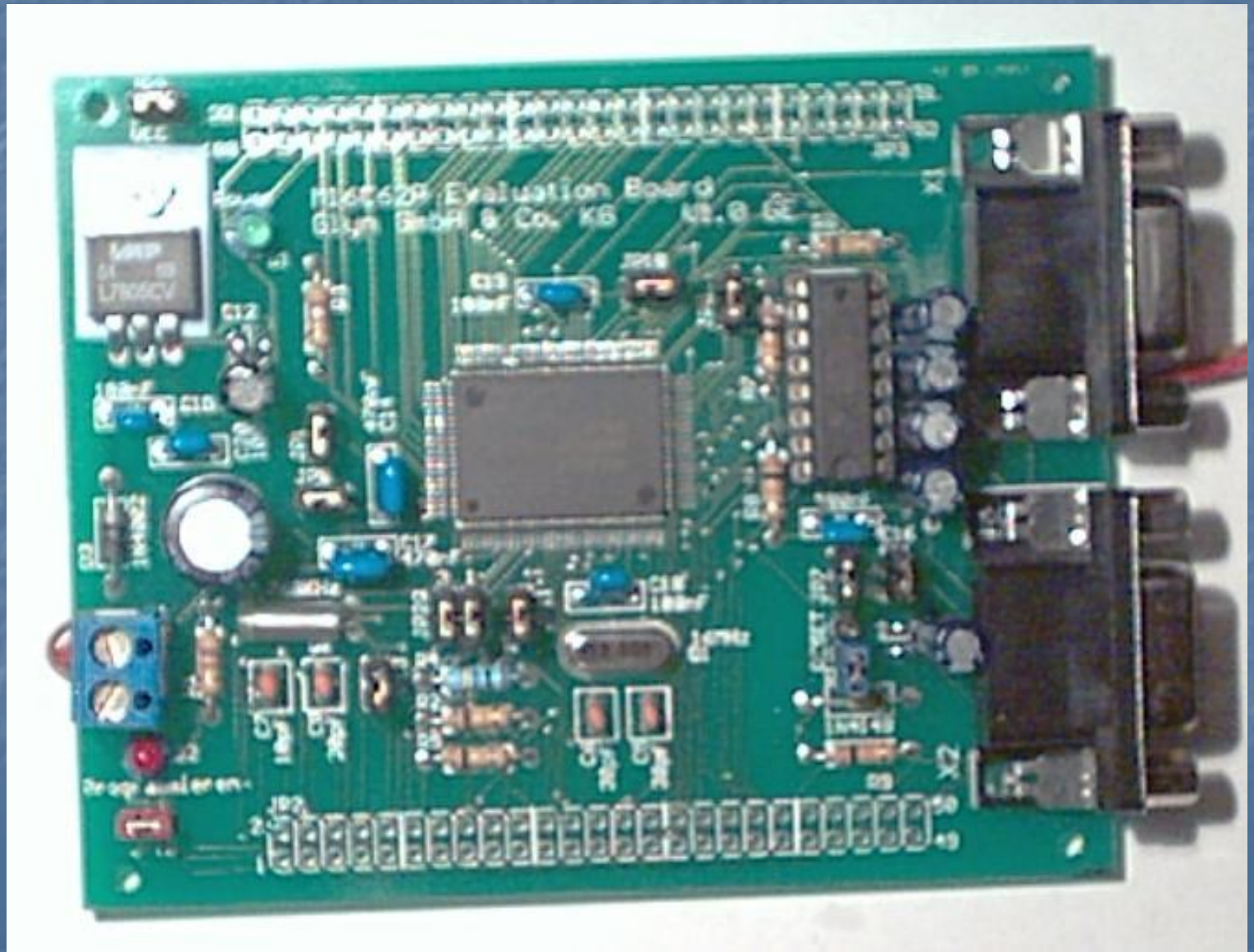
- The fast developing in electronic requires from time to time chances in the construction of electronic. In this case we searched for a substitute for an obsolete, antiquated and old-fashioned counter chip from AMD and additional the present 8 Bit CPU.

- *CPU = Central processor unit

The next generation...(not Star Trek)

- In earlier times we needed for every job much separate devices, like amplifiers, discrete transistors, one-job-chips, extra memory and much more ....
- Modern microcontrollers integrate much of this separate units.
- We found a very cheap and at the same time extremely powerful ready-to-use board with the Mitsubishi 16 Bit RISC controller from the 16C62 family
- Additional is a free of charge and size unlimited C-Compiler available. That opens the way to a professional software developing and using of very fast interrupt routines.
- Remember. A quality C-compiler costs normally way from 2000$...>

Ready-to-use M16C62P evaluation board by Glyn with the M30626FHPFP processor

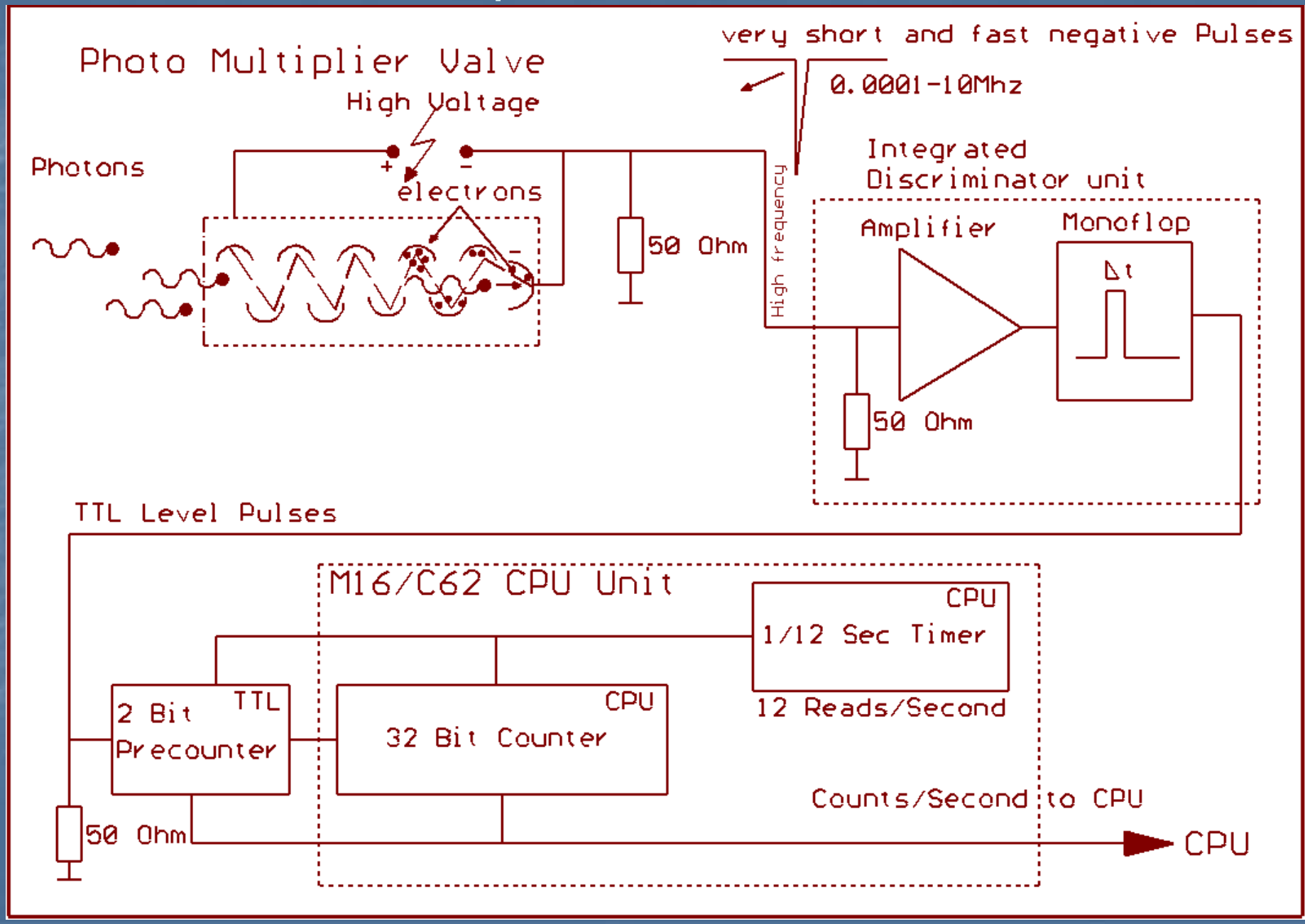


Additional features

- Internal flashmemory (reloadable resident program storage)
- Immense space of dynamic memory (RAM) is on chip .
- 11 Counter/Timer on chip
- 8 Channel 10Bit ADC on Chip
- Much 8 Bit Ports for Input / OutPut

That sounds like this controller does what we need to do for our work. !

Components of DFPOP

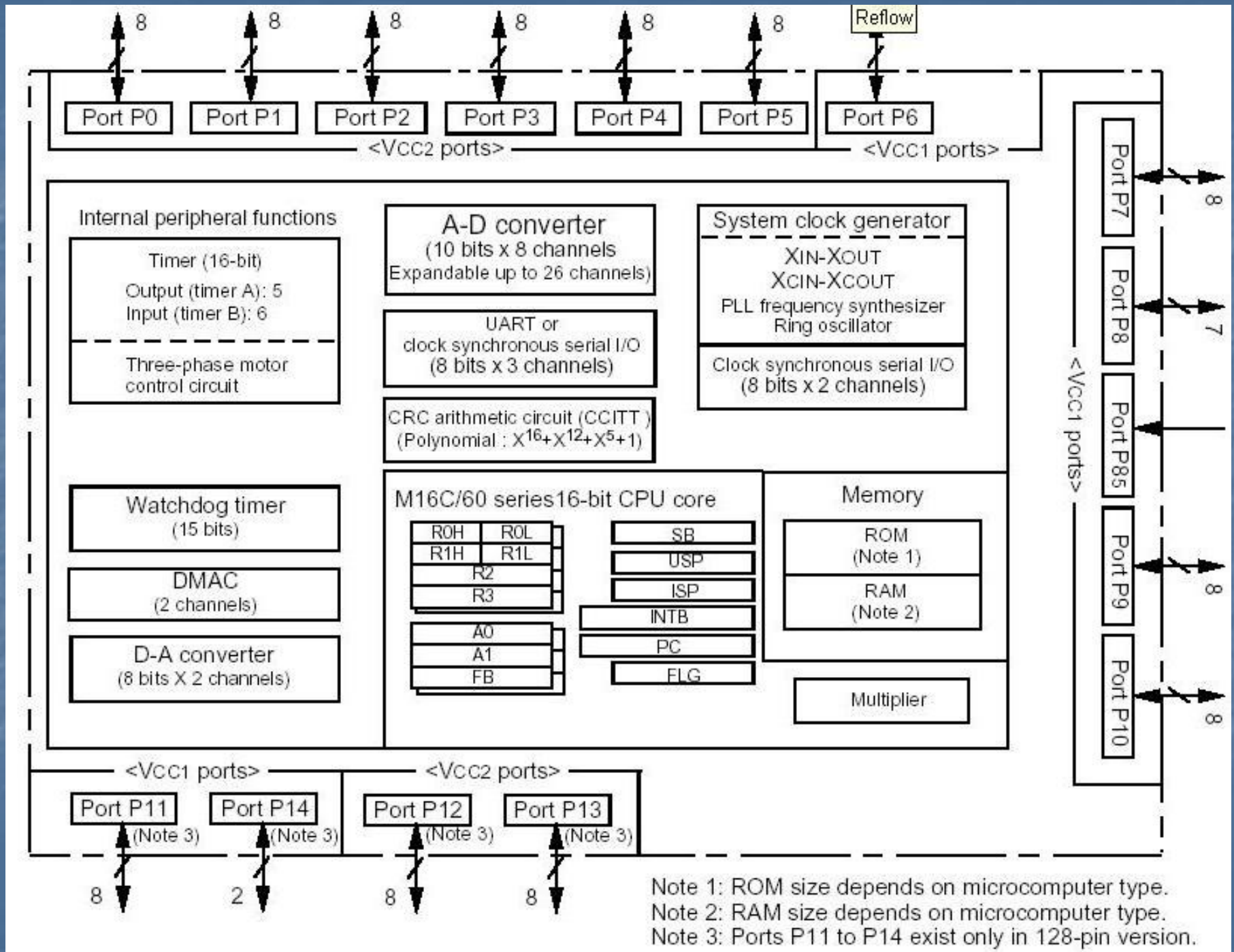


A little introduction to the inner life of the M16C62 controller

We write a C-language software that allows the use of the controller in the same “mnemonic” code control technique like by the Tiger-Basic module.

That makes the using in the same way simple.

But now we can use the inner components of the controller.



Note 1: ROM size depends on microcomputer type.
Note 2: RAM size depends on microcomputer type.
Note 3: Ports P11 to P14 exist only in 128-pin version.

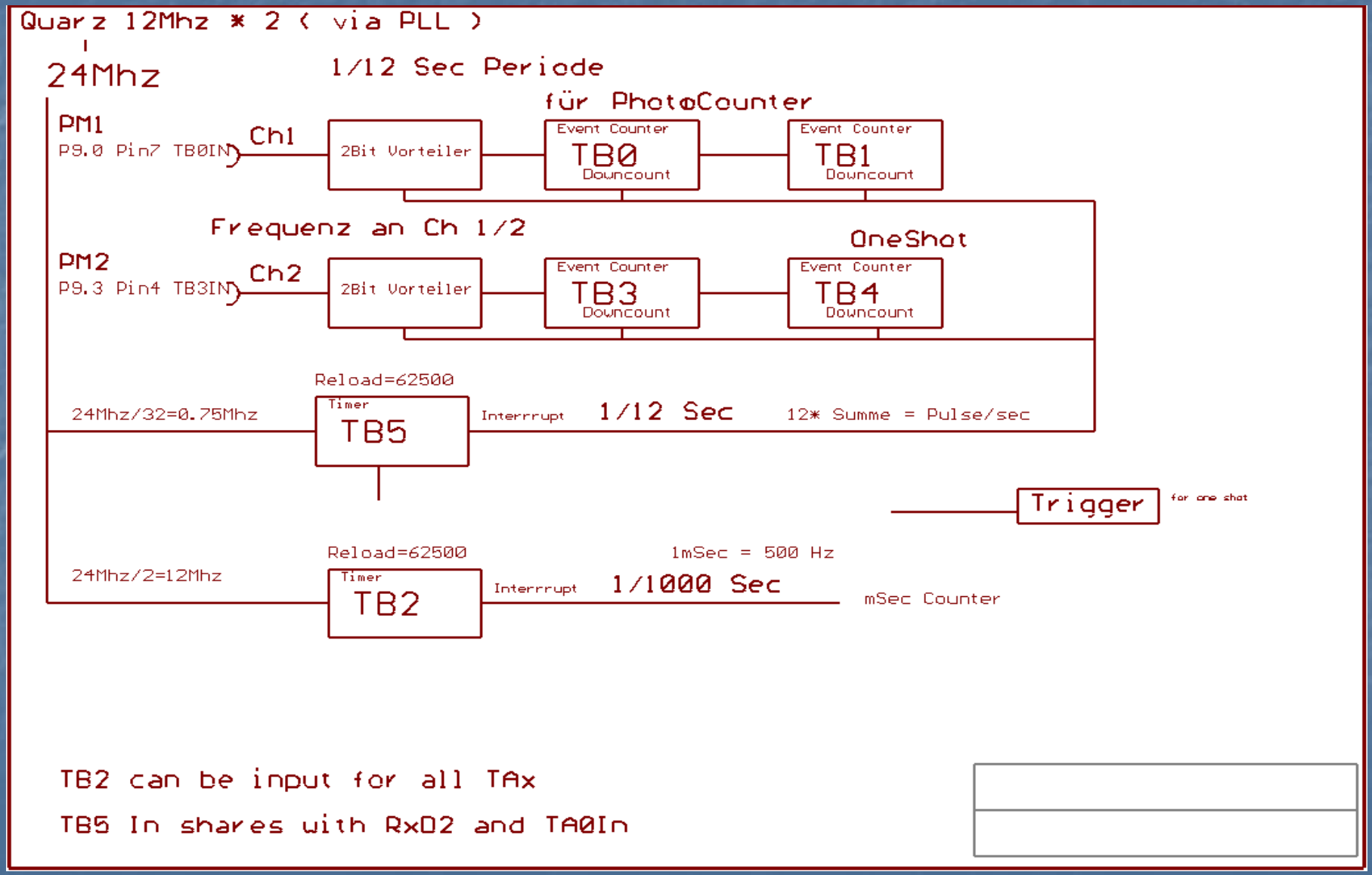
The photocounter section

The C-language firmware configurates the controller to use the timer “B” section for a 2 channel photocounter. That is done by a 1/12 second, interrupt driven reading routine.

Additional a 1 millisecond pulse is generated.

It is used as a system counter.

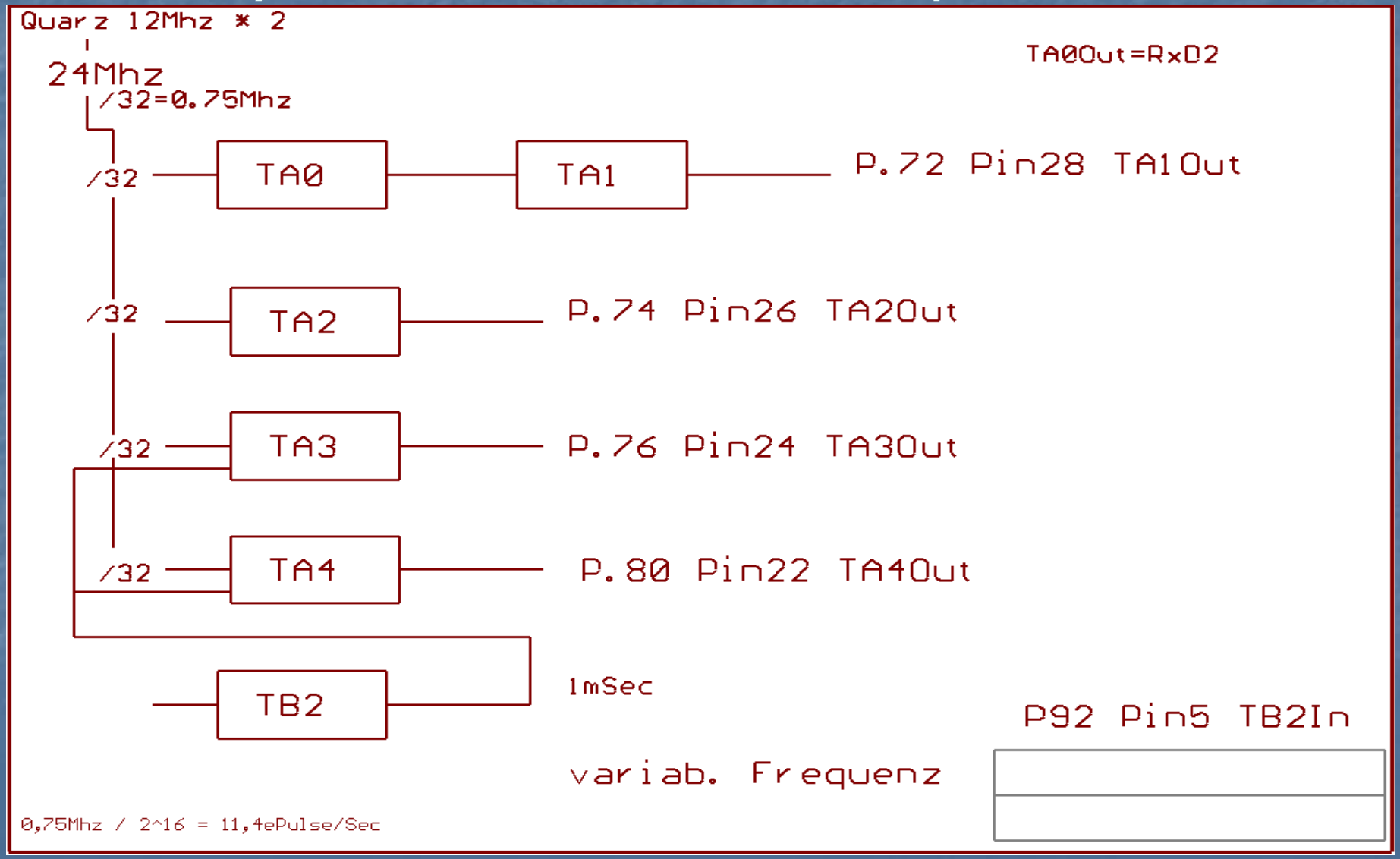
Counter using for photo counting



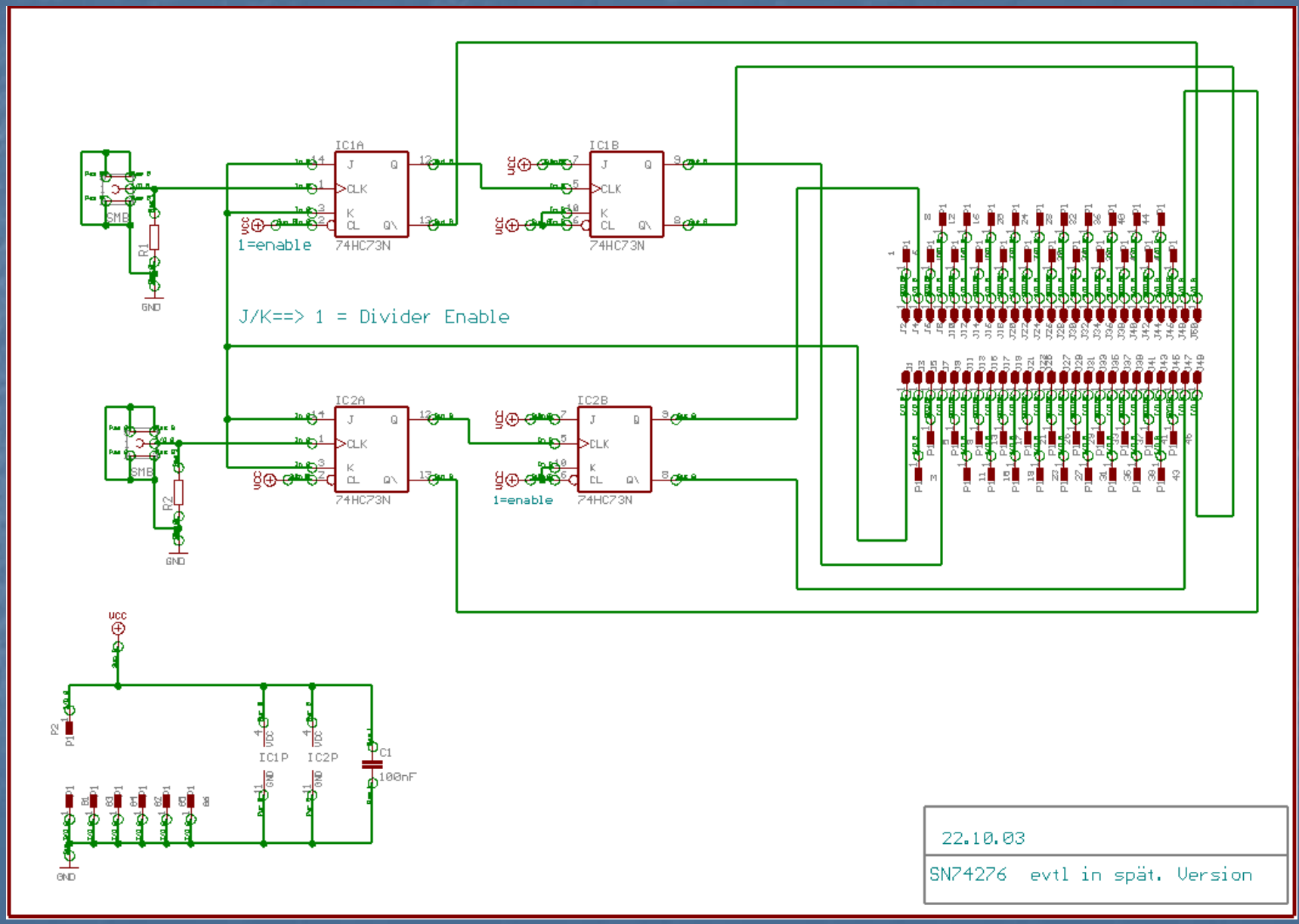
Signal and frequency synthesis section

- The five timers/counters “A” have enough possibilities to realize exact pulsecounts, and frequencies. TA0-TA1 are cascaded for better precision of the synthesized frequency.

Timerusing for different jobs like Frequency synthesis and controlled pulseout



Precounter schematic



2 channel precounter module for direct soldering to M16C62 doard

- C) Universitaet Regensburg
- Version:11.00 vom 10.12.2003
- Glyn Bord with 30624 FGAFP
- MEANING:(x,y,z)=Selctable,[x]=Default
- sampleset =>1sec=[1]; 1/4sec=4; 1/2sec=2; 3/4sec =3 ([1],2,3,4)
- spwmX (X=1,2,3)=>Set PWM & Start
- offpwmX (X=1,2,3)
- p0in => 8 Bit input
- pXsb =>X=(1,2,3) Port X Set Bit 2.Eingabe erford.(0..7)
- pXcb =>X=(1,2,3) Port X Clear Bit 2.Eingabe erford.(0..7)
- pXout =>X=(1,2,3) BYTESET 2.Eingabe erford.
- getphotocnt{gpt} =>returns akt.PhotoCnt, Chan1<CR>Chan2<CR>
- mprc => Set MP-Repeats[1]<CR>
- mp =>returns akt.PhotoCnt, <TAB>Chan1<TAB>Chan2<CR>
- cntch12o; cntch12f => Polling Counter 1+2 ON/OFF
- frqset =>Frq set
- frqon,frqof =>Frequenz TA1_OUT->P28
- adcX =>X=1..7
- p4ps p4_pulsstart
- p4spb =>p4_setpulsbit, Port4 Bit 0..3], -->P4[0]Default
- p4spc =>p4_setpulscnt, NN Pulse [1]
- p4spf p4_setpulsfrq, (1..500),[10], Counts/sec
- qq =>qs, =>START sq=>STOP Polling Counter
- xxx => General Break of Loops
- ??, ???, ???? => Menue, Pinbeschreibung, MP-Mnue

PinBelegung

2*25pol. Steckerleiste unten=A, oben=B

Frequenz Out = TA1Out = Port.72 = Pin A28

PWM 1 Out = TA2Out = Port_74 = Pin A26

PWM 2 Out = TA3Out = Port_76 = Pin A24

PWM 3 Out = TA4Out = Port_80 = Pin A22

sonstige Verwendung

TB2=1mSec.Generator

TB5=1/12Sec.Generator

ADC(0..7)= Port_10(0..7) Pin B47..B40

PORT_4(Bit0..3) = NN_Pulse Out = PIN B0..3

PORT_0 = 8Bit Input = PIN B38.. B31

PORT_1..3 = 8Bit Output = Pin B30..B05

mp =>returns akt.PhotoCnt, <TAB>Chan1<TAB>Chan2<CR>

mprc => Set MP-Repeats[1]

stmep =>Set StepMotor Enalbe Port {[1](1..3)

stmeb =>Set StepMotor Enalbe BIT {[0](0..7)

stmdp =>Set StepMotor Direction Port {[1](1..3)

stmdp =>Set StepMotor Direction Bit {[0](0..7)

p4spb p4_setpulsbit =>Port4 Bit 0..3], -->P4[0]Default

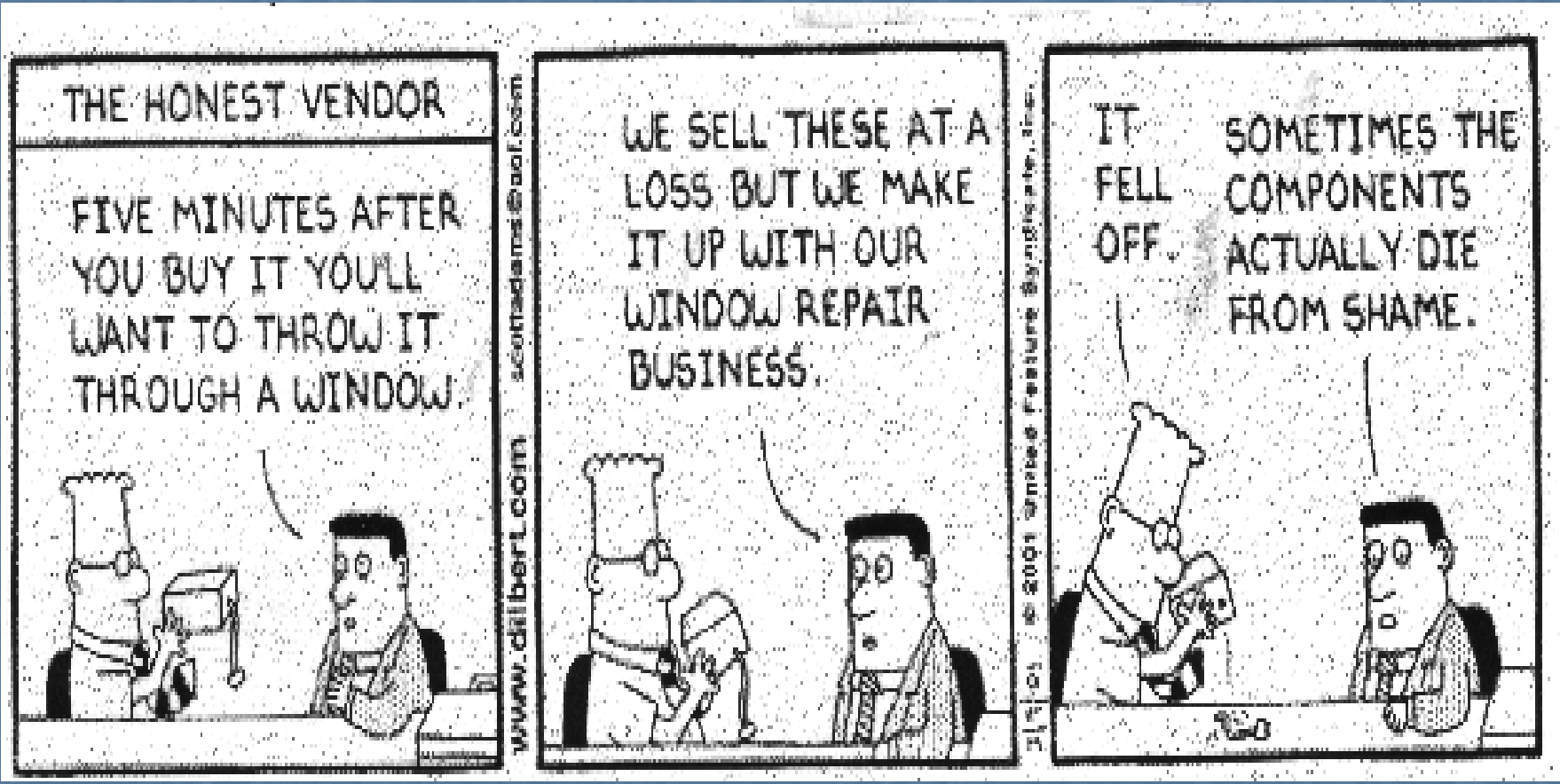
p4spc p4_setpulscnt =>NN Pulse [1]

p4spf p4_setpulsfrq =>(1..500),[10], Counts/sec

xxx => General Break of Loops

??, ??? => Menue, Pinbeschreibung ???? MP-Menue

Quality control



We can do much to avoid this situation.

We work for clean and healthy water.

Fine, but if you see two dolphins you are all right,

if not, you have probably a burn out.