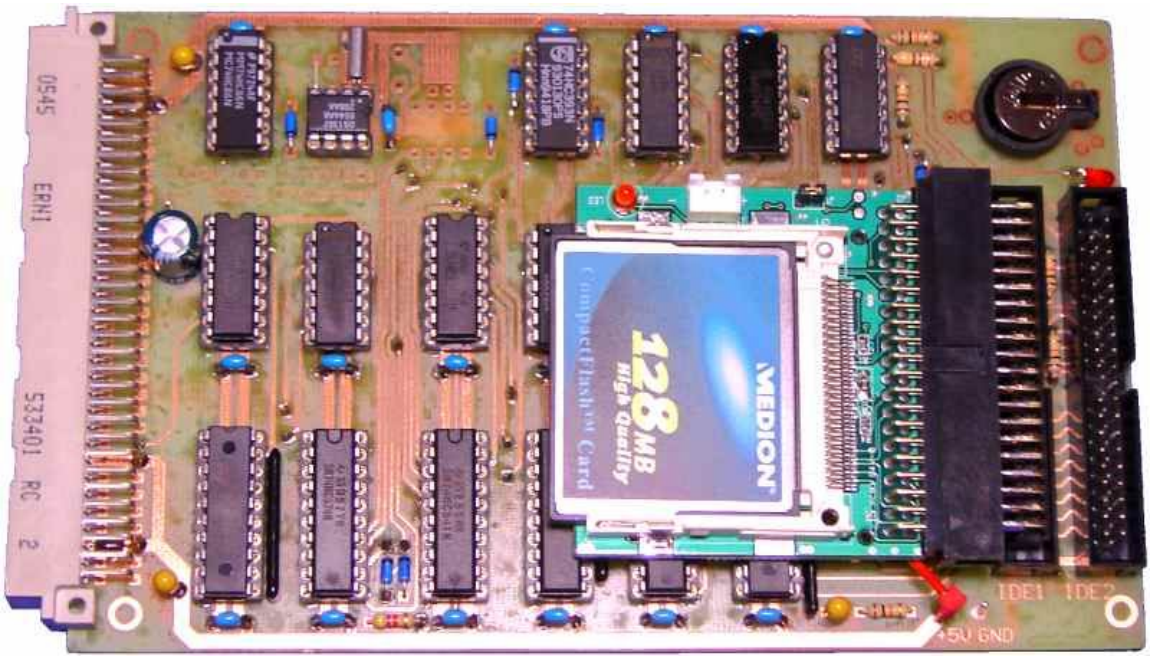
IDE - Controller

For the MyCPU

- Selfbuild Guide -



uses `/IOEN2:2800h-29FFh`

Dennis Kuschel · Corintostraße 21 · 28279 Bremen · Germany · www.mycpu.eu · dennis_k@freenet.de 2015-07-14

Content

Figures

Tables

1 Overview

Features:

- 2 independent IDE channels, up to 4 IDE devices can be connected to the controller
- supports PIO mode 1 (all parallel IDE devices that support LBA can be connected)
- supported devices are: Hard Disk Drives, CD-ROMs, Compact Flash Cards
- A battery buffered RTC (real time clock) provides date and time

Limitations:

- CD-ROM - Drives are still not supported by the MyCPU operating system.
- There is still no FAT / NTFS / EXT2 file system driver available. The CPU uses its own "Micro File System" (MFS) to store the data on the disks. The MFS supports partitions up to 16MB, so it does not make sense to connect large disks to the system. Furthermore, the MFS uses a block size of 256 bytes, whereas a standard sector size is 512 bytes. This means that the half of the disk space remains unused when using MFS. For example, you can store a maximum of 32MB data on a 64MB compact flash card.

WARNING!

This board is really difficult to solder!

Because the circuit of the IDE Controller is really complex, there must be many parts on the printed circuit board. An inexperienced guy may not be able to solder all parts on the board as they should be. It is likely that the IDE Controller will not function as expected.

Hint: You can buy a professional PCB at http://mycpu.thtec.org/ .

Author:

Dennis Kuschel
Corintostraße 21
28279 Bremen
Germany

web : http://www.mycpu.eu
email: dennis_k@freenet.de

2 Board

2.1 IDE Controller Board

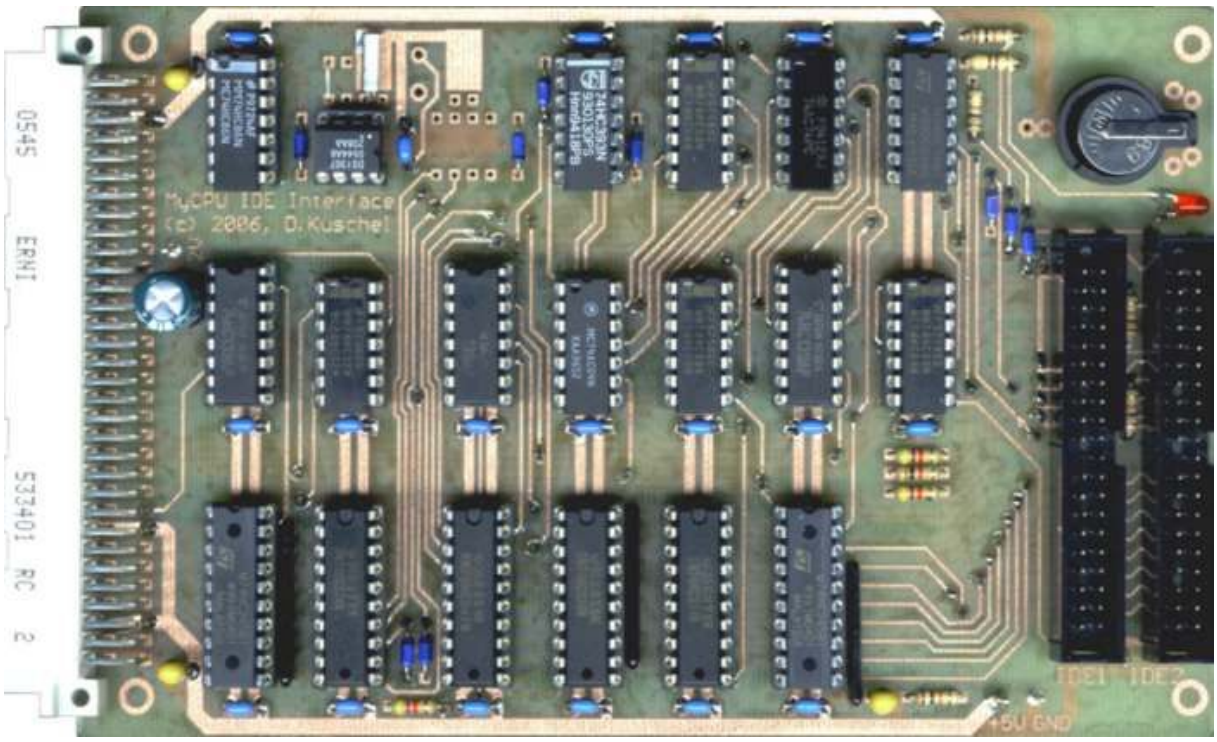


Fig. 1: IDE Controller Board (Prototype)

2.1.1 Description

The IDE Controller Board enables you to connect Mass Storage Devices to the MyCPU21. The Controller provides two IDE channels (40 pin connectors). Compact Flash Cards and 2,5'' harddisks can be directly mounted onto the board.

2.1.2 Selection of Components

There are two 74ACxxx gates used in the circuit. This gates can safely be replaced by their 74HCxxx counterparts, but this may result in a bit unstable behaviour of the IDE controller at high CPU clock frequencies (6MHz and above). Five different RTC chips can be used with the board, please see section 2.1.4 for details.

2.1.3 Placement of Components

After you have soldered all Via's, you can continue with the integrated circuits. I strongly recommend you not to use sockets for the IC's because it is nearly impossible to solder all sockets correctly to a self-etched PCB. A manufactured PCB does not have this limitation. If you wish to use sockets anyway, you must use the high precision sockets. Only the high quality sockets allow you to solder the pads on the top side of the board. I recommend you to follow the placement order I have noted in the placeplan below (see blue numbers, and start with the IC that has the blue number 1). When all IC's are placed and soldered, you can continue to place the capacitors. In the last round the resistors, diodes and board connectors are placed and soldered.

ATTENTION!
Please be careful, and don't forget to solder a pad on the top side of the board. I have marked all critical pads with red colour in the placeplan below. Please check if you have really soldered these pads! The connectors IDE1 and IDE2 are really difficult to solder. If you are not so experienced in soldering, then populate only connector IDE1.

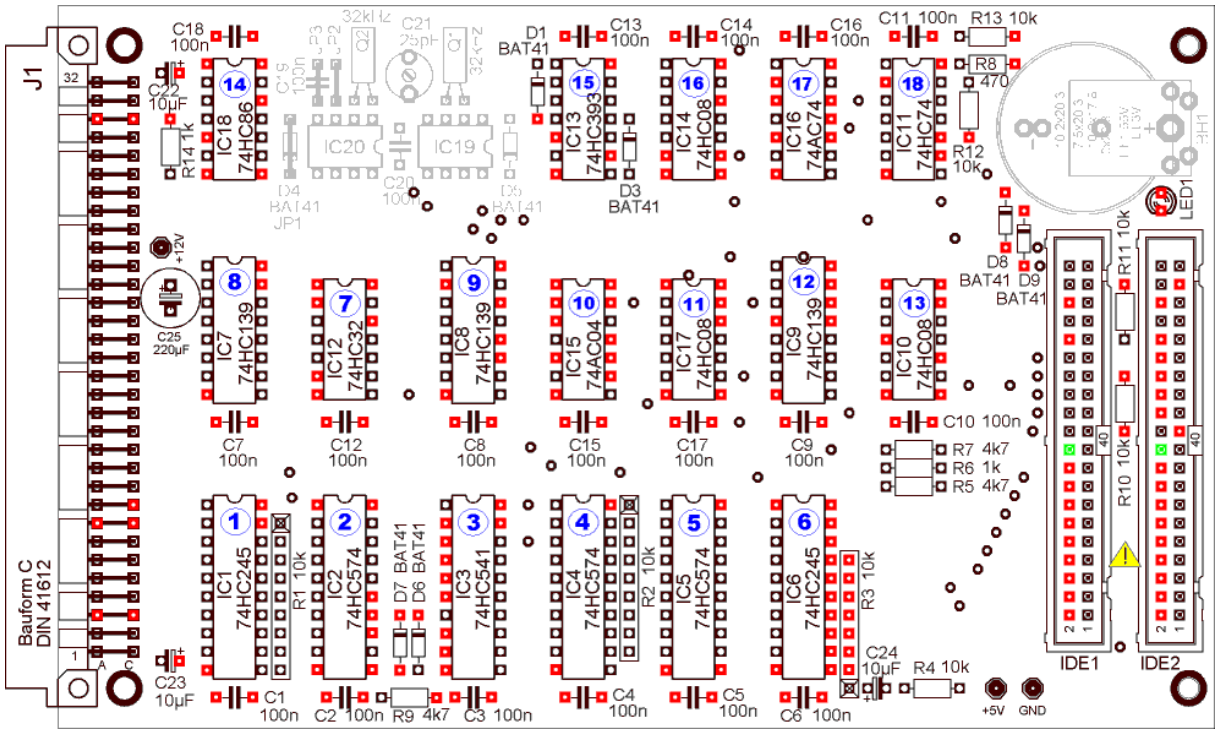


Fig. 2: IDE Controller Board - Placeplan

2.1.4 Five possible Real Time Clocks

You can choose between five different RTC chips for the IDE Controller. The table below lists all possible RTC's and shows which other components must be populated.

	DS1302	DS1307	DS1337	DS1678	PCF8385
Battery	3.0 V	3.0 V	1.55 V	3.0 V	1.55 V
IC19	X				
IC20		X	X	X	X
D4	X		X		X
D5	X		X		X
C19		X		X	
C20	X	X	X	X	X
C21					X
Q1	6 pF				
Q2		12,5 pF	6 pF	12,5 pF	12,5 pF
JP1		X		X	
JP2		X		X	
JP3					X

Tab. 1: Five different RTC's

2.1.4.1 DS1302

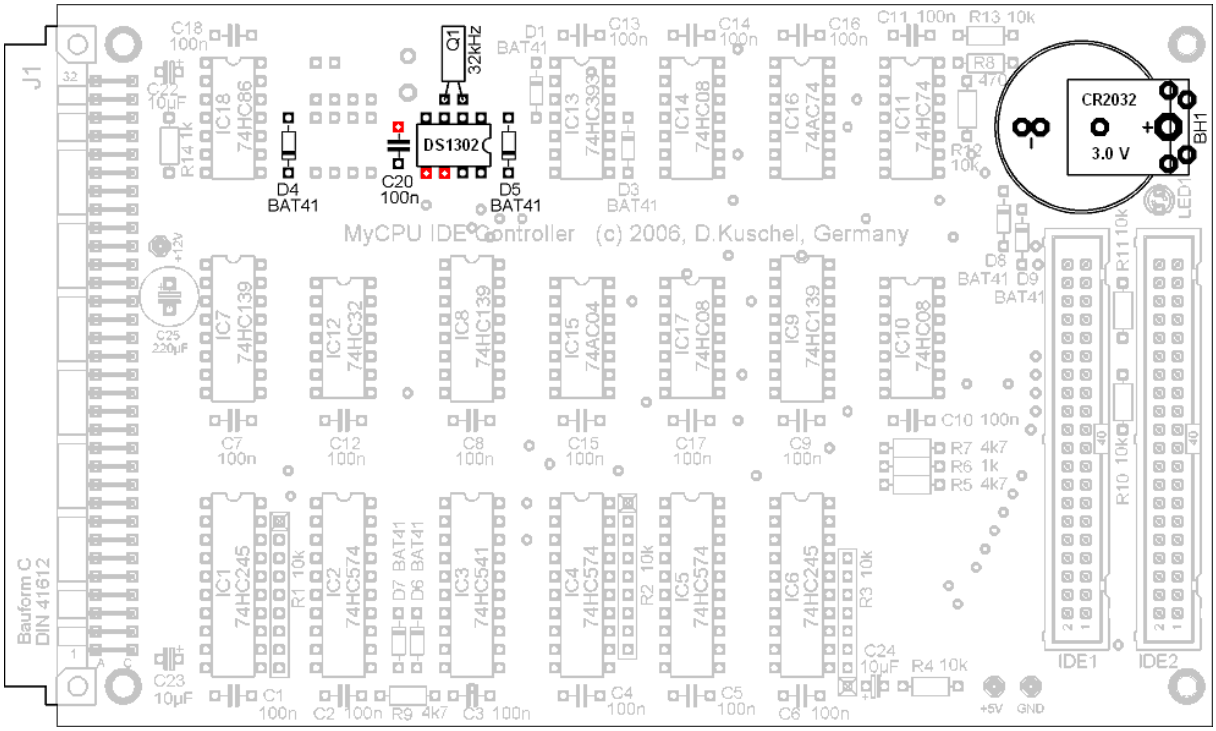


Fig. 3: Placeplan of Components required for RTC DS1302

2.1.4.2 DS1307

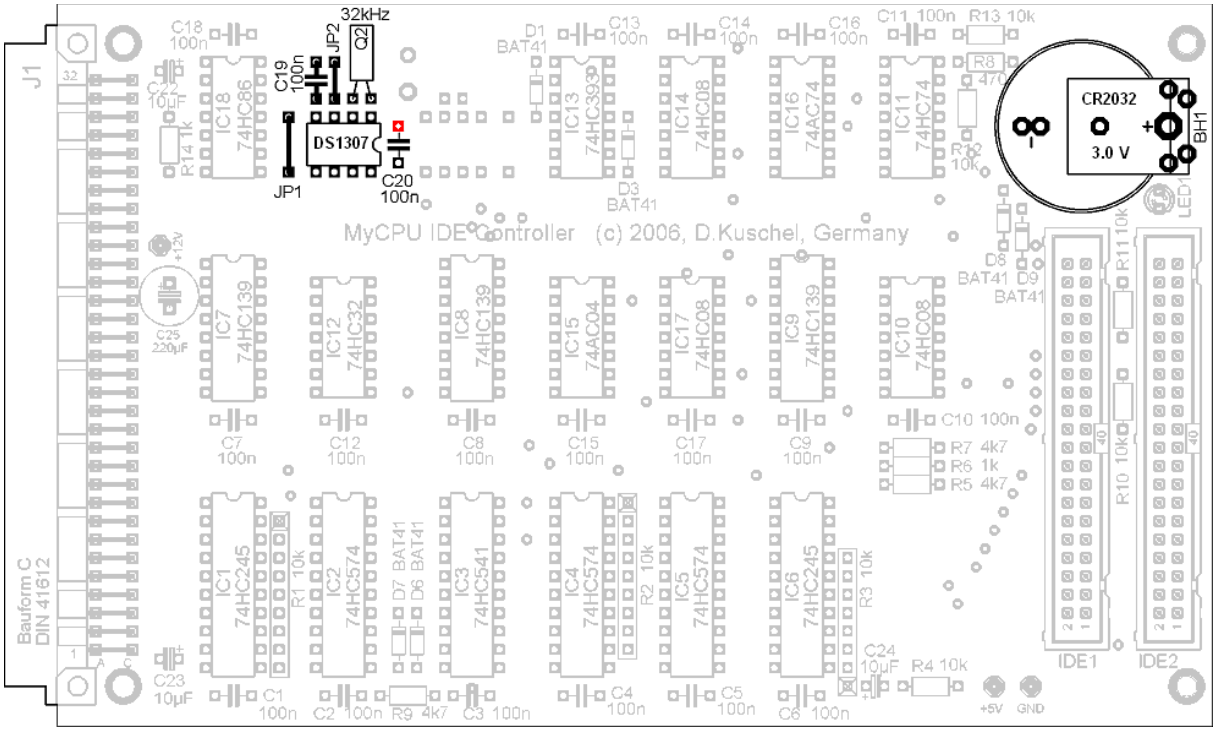


Fig. 4: Placeplan of Components required for RTC DS1307

2.1.4.3 DS1337

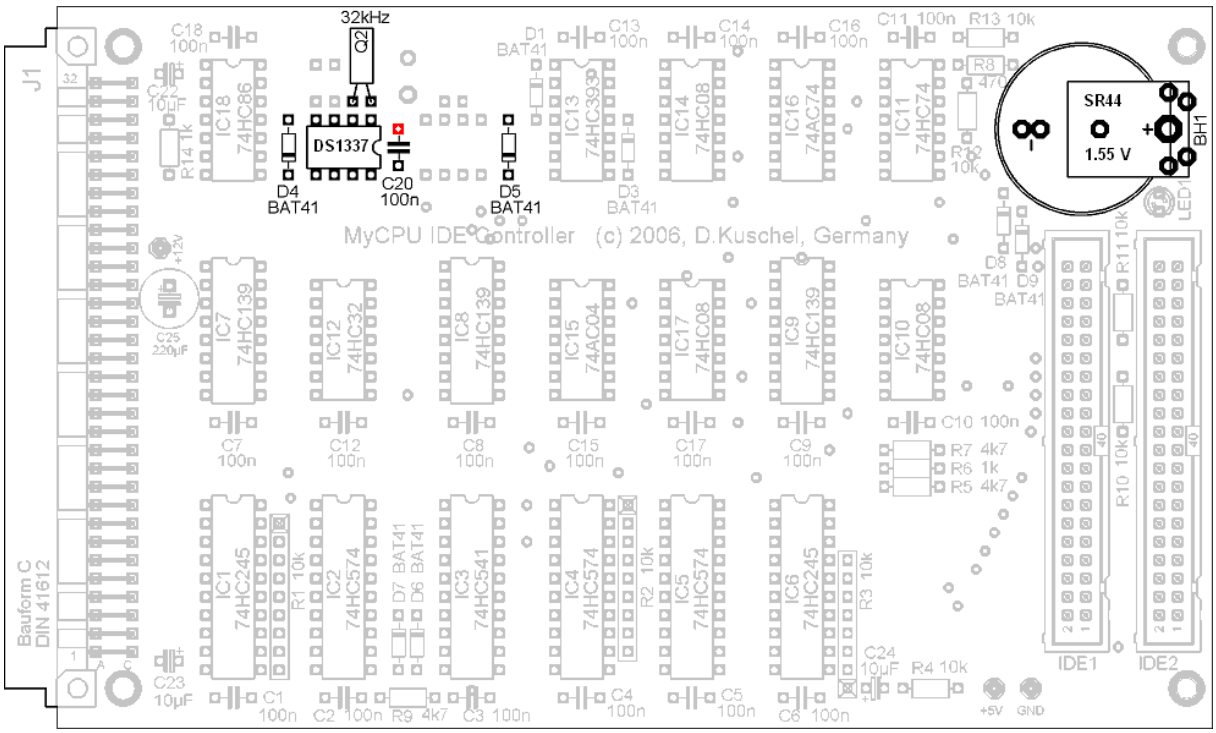


Fig. 5: Placeplan of Components required for RTC DS1337

2.1.4.4 DS1678

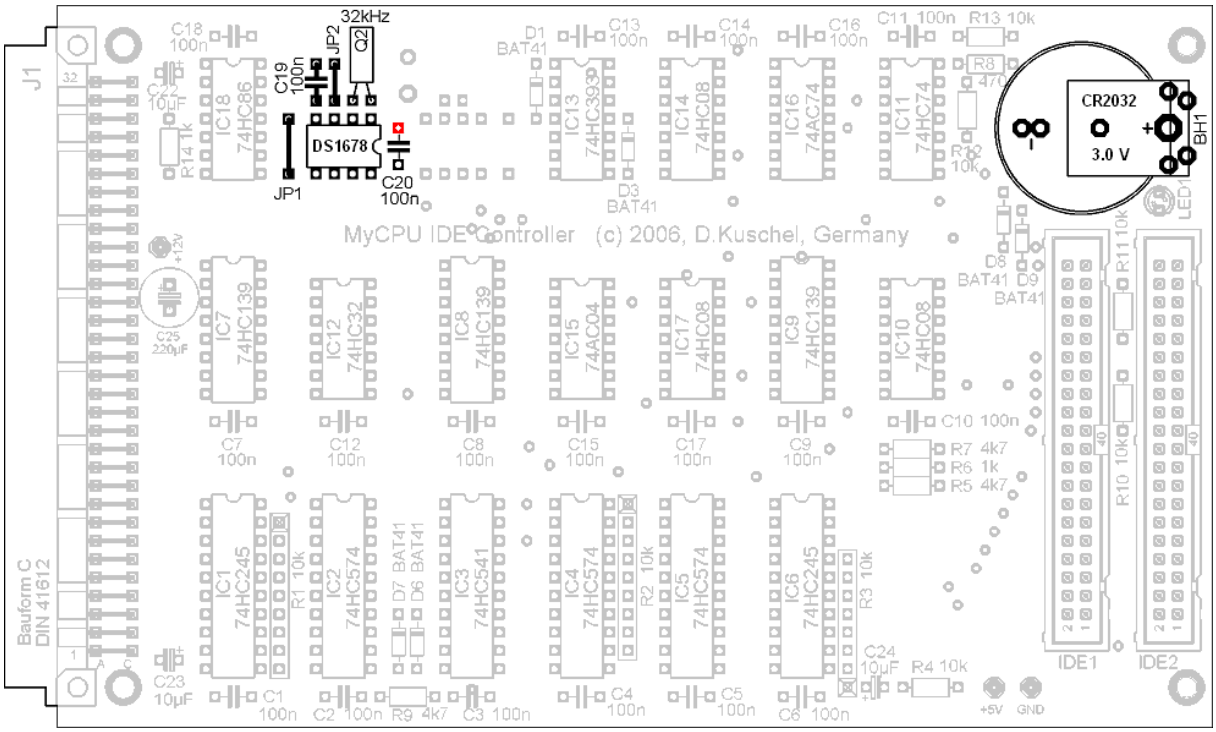


Fig. 6: Placeplan of Components required for RTC DS1678

2.1.4.5 PCF8583

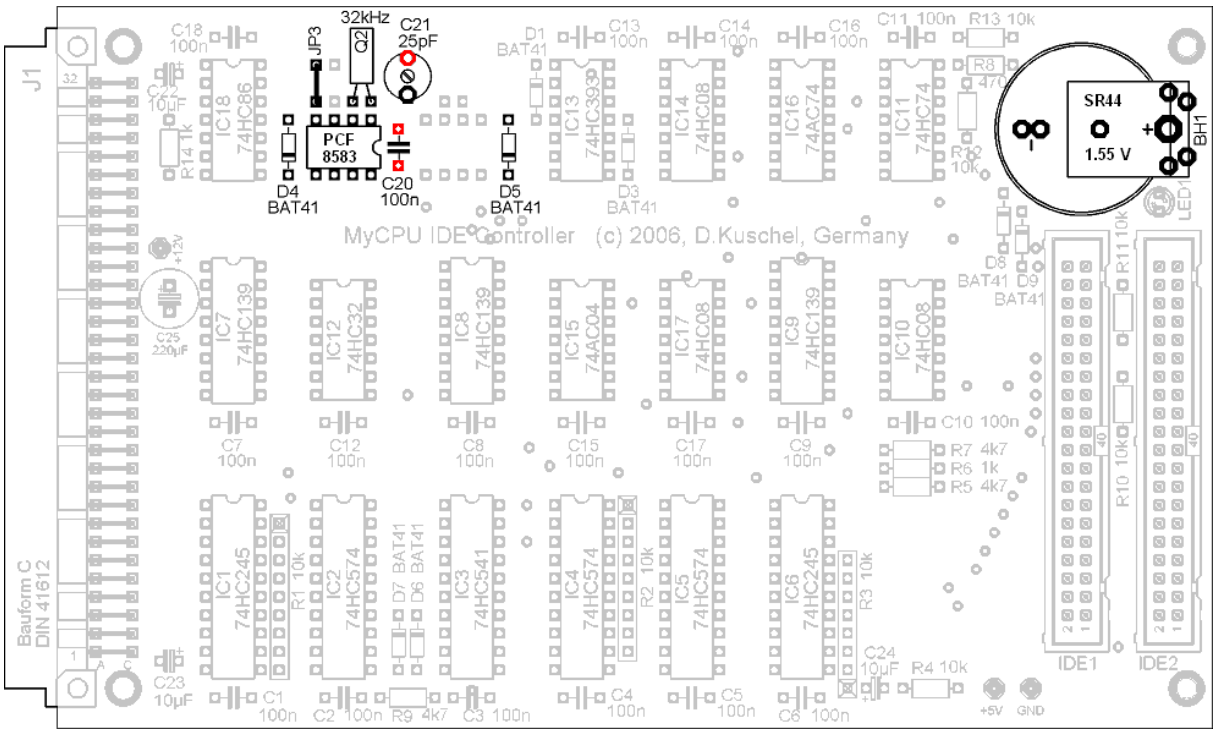


Fig. 7: Placeplan of Components required for RTC PCF8583

2.1.5 Partlist: Common Parts

74AC04	IC15	
74HC08	IC10, IC14, IC17	
74HC32	IC12	
74AC74	IC16	
74HC74	IC11	
74HC86	IC18	***must be a 74HC type***
74HC139	IC7, IC8, IC9	
74HC245	IC1, IC6	
74HC393	IC13	
74HC541	IC3	
74HC574	IC2, IC4, IC5	
BAT41	D1, D3, D6, D7, D8, D9	
red LED	LED1	
470 Ohm	R8	
1 kOhm	R6, R14	
4.7 kOhm	R5, R7, R9	
10 kOhm	R4, R10, R11, R12, R13	
SIL 8 x 10 kOhm	R1, R2	
SIL 7 x 10 kOhm	R3	
100nF ceramic capacitor	C1 - C18	
10µF / 16V, tantal	C22, C23, C24	
220µF / 16V, radial	C25	
40 pin header for ribbon cable	IDE1, IDE2	Note: Pin 20 removed (“key pin”)
DIN 41612 Connector	J1	

2.1.6 Required Parts for RTC DS1302

Dallas DS1302 (or DS1202)	IC19
BAT41	D4, D5
100nF ceramic capacitor	C20
Crystal 32kHz / 6pF	Q1
3.0V Lithium Battery Holder	BH1

2.1.7 Required Parts for RTC DS1307

Dallas DS1307	IC20
100nF ceramic capacitor	C19, C20
Crystal 32kHz / 12.5pF	Q2
piece of wire	JP1, JP2
3.0V Lithium Battery Holder	BH1

2.1.8 Required Parts for RTC DS1337

Dallas DS1337	IC20
BAT41	D4, D5
100nF ceramic capacitor	C20
Crystal 32kHz / 6pF	Q2
1.55V Lithium Battery Holder	BH1

2.1.9 Required Parts for RTC DS1678

Dallas DS1678	IC20
100nF ceramic capacitor	C19, C20
Crystal 32kHz / 12.5pF	Q2
piece of wire	JP1, JP2
3.0V Lithium Battery Holder	BH1

2.1.10 Required Parts for RTC PCF8385

Philips PCF8385	IC20
BAT41	D4, D5
100nF ceramic capacitor	C20
4-25 pF (trim capacity)	C21
Crystal 32kHz / 12.5pF	Q2
piece of wire	JP3
1.55V Lithium Battery Holder	BH1

3 Using the IDE Controller (first test)

3.1 Test Setup

You need a working MyCPU System including the **Interrupt Controller Board**. Please check your ROM version, the **Kernel version 2.7** and **MicroFS version 1.3 are required**.

3.2 Running the Test

3.2.1 Basic Test

For the first test you should NOT connect any harddisk or other IDE device to the IDE controller. Switch on the MyCPU and observe the printing. You should see the following:

```
FOUND: Memory Extension Board
FOUND: Interrupt Controller
FOUND: RS232 Interface (UART)
FOUND: Parallel Printer Port
FOUND: AT Keyboard Controller
FOUND: IDE Interface Controller

Installing IDE Controller:
  Primary   Master :  not present
  Primary   Slave  :  not present
  Secondary Master :  not present
  Secondary Slave  :  not present
```

3.2.2 RTC Test

When you see the prompt, you can use the commands "`date`" and "`time`" to set the system time and date. For example, "`date 2006-03-25`" sets the date to year 2006, March the 25th, and "`time 15:47:20`" sets the time to 3 PM, 47 minutes, 20 seconds.
Note that the date and time command will also work when no IDE Controller / RTC is installed, because the Operating System has a software clock. To see if the hardware RTC works properly, power cycle the CPU (do a hardware reset or type "`coldstart`" at the prompt). When you now type "`time`" [plus enter, no parameters] after the system has booted again, the system should display the correct daytime. Otherwise, you should check the RTC chip and the periphery components (don't forget the bridges).

Important Note:
After the system has booted, the Operating System uses only the software clock for time services. After a while the software clock may drift from the RTC clock. To resynchronize the clocks, it is required to reboot the system.
The basic variables "`TI`" and "`TI$`" get their time from the software clock, thus they get inaccurate over time. This is important to know for data logging applications that require time stamping.

3.2.3 IDE Controller Test

Connect an IDE device (Harddisk device preferred) to the IDE connector labelled "IDE1". When you power on the system, you should see something like this:

```
FOUND: IDE Interface Controller

Installing IDE Controller:
  Primary   Master :  SAMSUNG CF/ATA / 62MB
  Primary   Slave  :  not present
  Secondary Master :  not present
  Secondary Slave  :  not present


No MFS Partition found. Please choose an existing partition that
will be deleted and used for MyCPU MicroFS on IDE Primary Master:

  [A]  FAT16 (type 06)

Please choose a partition: _
```

In this example, there is one FAT16 partition on the Compact Flash Card. When you now press the "A" the FAT16 partition will be deleted and formatted for MyCPU. The MyCPU automatically creates up to 8 new partitions within the space of the FAT16 partition. Please note that the last 4 partitions can not be accessed by default (they are marked "hidden").

After the partitions are formatted, and after any subsequent boot, you will see the following:

```
Mounting IDE drives:
  Drive  8: Partition 0 on IDE Primary Master, size 2048KB
  Drive  9: Partition 1 on IDE Primary Master, size 5120KB
  Drive 10: Partition 2 on IDE Primary Master, size 5120KB
  Drive 11: Partition 3 on IDE Primary Master, size 5120KB
  3 hidden partitions not mounted
```

In this example, the CF card has 7 partitions and only the first 4 are mounted to the drive numbers 8: to 11:. The first partition has a size of 2MB, and the others 5MB. Big partitions are really slow on the MyCPU, thus the first partition that usually holds the Operating System is set to the smaller size of 2MB. Note: The maximum size a partition can have is 16MB. With a maximum of eight partitions the MyCPU can use up to 128MB of a IDE device with MicroFS. (With a FAT32 file system driver all a the disk space would be available.)

After the system has started, you need to copy some basic files to drive 8. Please execute the following steps for this purpose:

- start the RemoteFS server program on the PC
- enter "`romdrive`" at the MyCPU prompt
- change to the drive where the romdrive is installed to (for example, enter "`12:`")
- enter "`remotefs com2:`"
- change to the drive where the remotefs is installed to (for example, enter "`13:`")
- type "`copy8`" to start the script that copies some files to drive 8.

You may now change the keyboard layout for your purpose. If you prefer a German keyboard layout, type "`edit 8:/init`", and uncomment the line "`# console gr`" (remove the "#").

4 Registers

4.1 Overview

Address	Register Name	Rd/Wr	Function
2800h	REG_IDE_DATA*	R/W	IDE Device: Data register / Data Port
2801h	REG_IDE_ERROR*	R	IDE Device: Error information register
2801h	REG_IDE_FEATURE*	W	IDE Device: Feature register
2802h	REG_IDE_SECTOR*	R/W	IDE Device: Sector Count register
2803h	REG_IDE_LBA_LOW*	R/W	IDE Device: LBA Low register
2804h	REG_IDE_LBA_MID*	R/W	IDE Device: LBA Mid register
2805h	REG_IDE_LBA_HIGH*	R/W	IDE Device: LBA High register
2806h	REG_IDE_DEVICE*	R/W	IDE Device: Device register
2807h	REG_IDE_COMMAND*	W	IDE Device: Command register
2807h	REG_IDE_STATUS*	R	IDE Device: Status register
280Eh	REG_IDE_ASTATUS*	R	IDE Device: Alternate Status register
280Eh	REG_IDE_CONTROL*	W	IDE Device: Device Control register
2900h	REG_CTRL_HIGHBYTE	R/W	IDE Controller: Bits 8-15 for 16bit transfer
2901h	REG_CTRL_OUTPUT	W	IDE Controller: Multi Output register
2901h	REG_CTRL_STATUS	R	IDE Controller: Line Status register

* This registers are part of the IDE devices. They are documented in the IDE specification (the ATA/ATAPI specification can be downloaded from www.t13.org).

4.2 REG_CTRL_HIGHBYTE

This register enables the access to the higher data bits on the IDE bus. The IDE-Bus has a width of 16 bits. The lower 8 bits can be directly accessed by reading/writing the REG_IDE_DATA-register, and the higher 8 bits can be accessed through this register.

Address: 2900h (read / write)

MSB(D7)							LSB(D0)
DD15	DD14	DD13	DD12	DD11	DD10	DD9	DD8

DD15 – DD8 IDE-Bus data lines DD15 – DD8. Before writing bits DD7 – DD0 to the IDE-Bus REG_IDE_DATA-register, the high bits are written to this register. Vice versa, after reading the low bits DD7 – DD0 from the REG_IDE_DATA-register, the bits DD15 – DD8 can be read from this register.

4.3 REG_CTRL_OUTPUT

This register controls several IDE steering lines.

Address: 2901h (write only)

MSB(D7)							LSB(D0)
ISEL	IRES	CHK		SEN	SCLK	SOD2	SOD1

`ISEL` IDE interface select. When this bit is set to 0, the bus IDE0 can be accessed through the registers `REG_IDE_xxx`. When this bit is set to 1, the bus IDE1 can be accessed.

`IRES` This bit can be used to reset all connected IDE devices (hardware reset). This bit is active low, the normal (inactive) state is high (=1).

`CHK` This bit is directly connected to the both `CHK`-bits in the `REG_CTRL_STATUS`-regsiter. It can be used to detect the presence of the IDE controller board.

`SEN` Serial bus: enable signal. This bit controls the chip-enable line on the serial SPI bus. The SPI bus is used to connect a RTC chip.

`SOD2, SOD1` Serial bus: Output data signals for SPI / I^2C bus. See Tab. 2 for details.

`SCLK` Serial bus: clock signal. This bit controls the SCLK (SPI bus) or SCL (I^2C bus) signal.

SOD2	SOD1	Function
0	0	I/O / SDA –signal is LOW (low impedance state)
0	1	forbidden state – shortcut, may destroy IC2
1	0	I/O / SDA –signal is HIGH through 4.7kOhm PU (high impedance state)
1	1	I/O / SDA –signal is HIGH (low impedance state)

Tab. 2: Function of SOD1 and SOD2

4.4 REG_CTRL_STATUS

The state of several status lines can be read back through this register.

Address: 2901h (read only)

MSB(D7)							LSB(D0)
IO16	DA1	DA0	CHK	CHK	SID	0	0

`IO16` This bit represents the state of the IDE /IOCS16 wire. This bit is set to 1 when the /IOCS16 signal goes low. The bit is reset to 0 when a write to register `REG_CTRL_HIGHBYTE` is performed.

`DA1` Current state of the /DASP signal of connector IDE1.

`DA0` Current state of the /DASP signal of connector IDE0.

`CHK` This two bits are directly connected with the `CHK` bit in register `REG_CTRL_OUTPUT`. See register `REG_CTRL_OUTPUT` for details.

`SID` Input data signal of SPI / I^2C bus. This bit represents the current state of the I/O / SDA -signal of the serial communication port.

5 Schematics

On the following page you will find the schematics of the IDE Controller Board.
The PCB layouts can be found in a separate PDF file (see file IDE_Layouts.pdf).

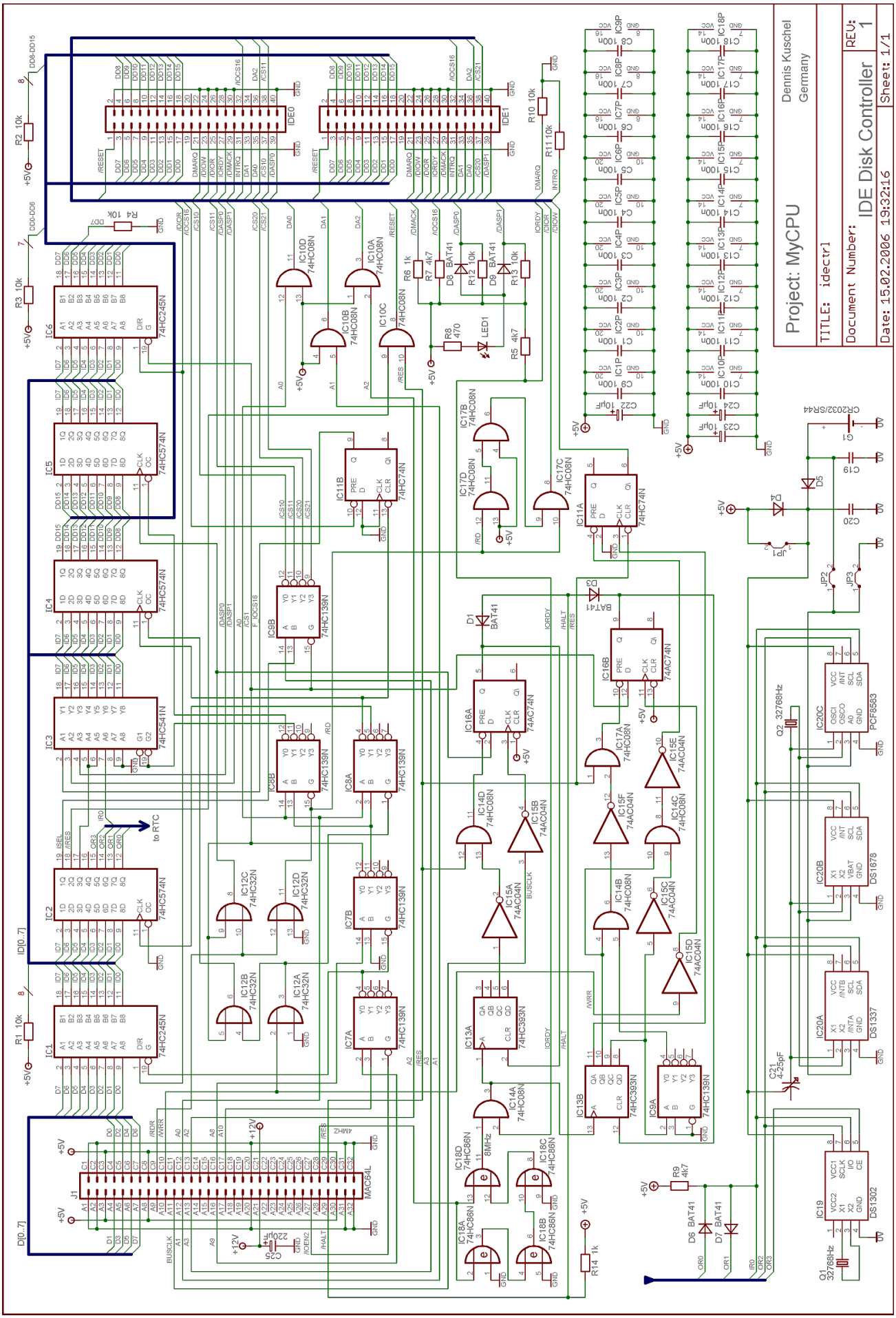


Fig. 8: Schematic of the IDE Controller Board

6 Change Log

6.1 Changes in the IDE-Controller design

Date	Name	Chapter	Description
2008-12-22	D.Kuschel	all	Document converted to OpenOffice format
2015-07-14	D.Kuschel		Document revised