ICD2

In Circuit Debugger and Programmer for PIC Devices

Jonas Meyer, NebLab

Zurich, Switzerland, December 12, 2004

What am I doing here?

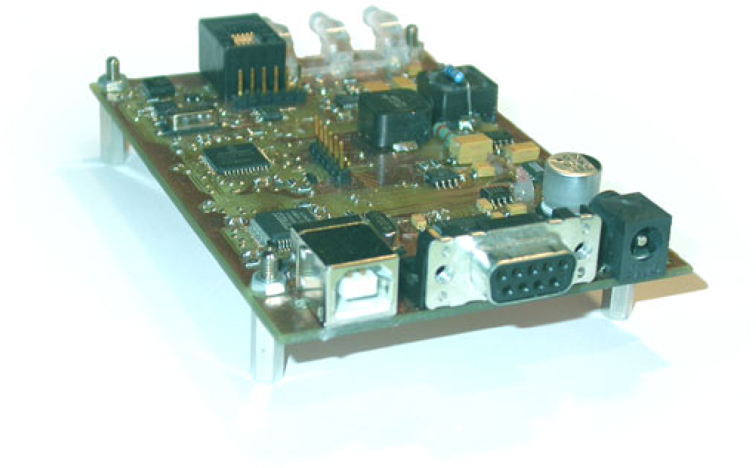
ICD2 is a programmer and debugger for Microchip PIC microcontrollers. It is a follow-up of the vintage ICD. The main features are:

- real time debugger
- full speed USB and RS232 interface
- support of most PIC and dsPIC microcontrollers
- compatibility to MPLAB IDE
- upgradeable firmware

Essentially it is a clone of Microchip's ICD2 based on information of reverse engineered designs found on www.edaboard.com and www.mcu.cz. This project is dedicated to enthusiasts and freaks with rather advanced hardware skills. The PCB is a two layer board and all components except some connectors are SMDs. So it is not a suitable training object for newbies. If you are just looking for a way to get a cheap debugger or programmer, go and buy one!

Contents

Note: If this article was useful to you or if you have any comments or feedback, feel free to contact us. Emails can be sent to: `in_port@nebadje.org`.



1 Circuit

1.1 Power Supply

ICD2 can either be powered by connecting an AC or DC power supply to the power jack J5 or over the USB cable connected to J4. When using an external power supply, its voltage should be above $5.5V$ but not exceed the maximum input voltage of the LDO regulator U4 of $15V$. Also make sure that the input capacitors C16 and C17 are rated for the used supply voltage.

In contrast to other designs, this one offers the possibility to power the target board with $3.3V$ over USB. This is very comfortable when working on small prototypes or doing some code experiments on a simple target board. However, make sure not to overload the USB host controller. In all probability ICD2 will enumerate as a low power bus powered d USB device, allowed it to draw at most $100mA$ from the bus. Since ICD2 needs about $60mA$ supply current the target should not draw more then $40mA$. It has been experienced that in USB powered operation mode the supply voltage fed to the target shows quite a large AC ripple. It doesn't affect digital circuitry but causes considerable conversion errors if this voltage is used as supply or reference for the ADC of the target board.

The USB supply voltage is regulated to $3.3V$ by U10 and fed to the USB controller. If the ICD2 is externally powered, the supplied voltage is regulated to $5V$ and the USB controller is inactive. Diode D3 assures safe operation if both supplies, USB and external, are connected. Either the regulated $3.3V$ or, if present, the $5V$ voltage is then fed to U3 and boosted to about $13V$. This is the programming voltage for the target. It is monitored on ADC channel 3 of U1 and actively controlled by a firmware routine adjusting the value of the digital potentiometer U16. The original Microchip debugger generatea programming voltages between $12.5V$ and $13.5V$, this ICD2 design generates voltages between $12.1V$ and $13.2V$. The programming voltage range can be fitted to the original one, if more precise resistor values for R40 and R41 are used (assembled

$1M\Omega$ and $100k\Omega$, exact $1.16M\Omega$ and $113k\Omega$). In spite of this deviation, no problems were experienced during programming various targets.

The boost regulator is designed to deliver $\approx 75mA$ of supply current. The inductivity `L4` and the capacity `C24` form a second LP filter to attenuate the voltage ripple. These two parts do not have to be assembled. `L4` can be shortened with a 0Ω resistor `RB4`. In this case it is important not to assemble `C24`, otherwise the output capacity of `U3` is too high and the boost regulator will not work properly.

Finally, the $13V$ programming voltage is regulated down to $5V$ to supply the ICD2 circuitry. This is quite a long daisy-chain of up and down conversions but this topology was adopted from the original Microchip debugger assuming their engineers knew what they were doing.

1.2 Debugger/Programmer

The central part of ICD2 is `U1`, a `PIC16A877A`. It communicates with both the host and the target. The connection to the host is established using the integrated `USART` of the `PIC`. `U2` drives the RS232 interface. In this design a `MAX232A` is used which needs $100nF$ capacities only, compared to $1\mu F$ when using a standard `MAX232`. `U1` directly interfaces with the USB controller `U12`. The target interface is eqipped with bus buffers, driving the serial programming lines. Standard high speed CMOS (`HC/HCT`) buffers can be used, although advanced high speed CMOS (`AHC/AHCT`) are to be prefered. The reason why two input buffers are connected in parallel is not known. The only argument I found was that the layout looks so nice ;). The status LEDs are designed to feed a light pipe which makes the design really fancy and well suited for integration into a housing.

Programming Connectors

The connector `J3` to the target is a standard RJ12 connector. Additionally, a 6 pin header `J8` is provided for custom programming cables. Resistors `R42` to `R45` protect the buffers against over current conditions. These resistors can cause communication problems depending on target device and circuitry connected to the target programming lines. In this case the resistors should be shortened. Connector `J1` can be used to write the bootloader firmware to the `PIC U1`. Care should be taken when using connectors J1 and J6 since they are not polarized.

1.3 USB Interface

The USB interface is implemented in `U12`. This Cypress USB controller `CY7C64613` manages the whole USB communication. It directly interfaces to the `PIC` PSP (Parallel Slave Port).

The external EEPROM `U13` stores the VID PID and device revision number. Its data is used to load the correct USB driver on the host.

The shielding of the USB connector is not connected since the ICD2 is intended to be mounted into a non metallic housing.

If no USB functionality is needed or if the Cypress USB controller is not available, the USB section can be omitted without affecting the remaining circuit. When building the ICD2, a good strategy is to first not assemble the USB parts and debug the circuit using the RS232 interface. If everything works fine, the USB interface can be assembled.

2 PCB

The PCB is a two layer board. Narrowest tracks measure 10 mil. It is not trivial to etch such small structures (experiment with exposure time, base concentration and temperature). Also consider contacting a PCB manufacturer. The via drills are $0.6mm$. To make the connections from one side of the PCB to the other $0.4/0.6mm$ hollow rivets are preferably used. If not available, short wires can be used to make the connections instead. For debugging purposes some 0Ω resistors have been introduced, (`RB0` to `RB7`). It is not necessary to assemble them. The corresponding pads can just be soldered together.

3 Firmware

3.1 PIC Bootloader

The microcontroller on the original ICD2 is a `PIC16F877`. Because this device is obsolete, it has been replaced by a `PIC16F877A`. The main difference between the two is the write procedure to the internal FLASH memory. The `877A` writes 4-word blocks at once to the memory, the `876` writes only one word. Therefore the ICD2 firmware had to be modified. This job has been done by *zaphod42* or at least he provided the firmware file. It is called `ICD2_BL.HEX` and is included in `ICD2_FW.zip`. The file can also be found at www.edaboard.com.

The bootloader has to be written to the `PIC18F77A` using a common programmer supporting `PIC` devices.

3.2 USB ID's

To get the programmer working on the USB port the EEPROM `U13` connected to the `CY7C64613` has to be programmed with the correct VID/PID/DRev. It is possible to either use a standard EEPROM programmer or the development tools provided by Cypress ($\approx 60MB$ download). The correct EEPROM data is stored in the `ICD2_EEPROM.HEX` (also included in `ICD2_FW.zip`). The first 9 bytes of the EEPROM have to contain the values `0xb4 0xd8 0x04 0x00 0x80 0x01 0x00 0x06 0x00` (VID/PID/DRev).

If using the Cypress development tools, EZ-USB FX has to be installed first. After connecting ICD2 to the USB port, the host recognizes the `CY7C64613` (with empty EEPROM) as a Cypress device. Now Cypress drivers have to be installed. Then the Cypress USB control panel can be started to program the EEPROM with the corresponding values.

When the EEPROM is programmed correctly ICD2 will be recognized by the OS as a Microchip device. Now the Microchip USB drivers have to be installed. After starting MPLAB it first downloads the CY7C64613 firmware. If the download was successful, MPLAB connects to ICD2 and downloads the appropriate `PIC` firmware. Now ICD2 is ready to connect to the target board and download or debug your code.

3.3 USB Drivers

Microchip USB drivers seem to be kind of buggy since they provide a USB driver removal tool `MPUsbClean.exe`. It is included in the MPLAB distribution. After installing MPLAB it is located in the installation directory in `/Utilities/MPUsbClean/`.

Sometimes it happens that the USB drivers will no longer recognize the ICD2. If so, the drivers have to be reinstalled. First, the old installation has to be removed using the provided cleaning tool. It also deletes all registry entries made during installation. After a reboot of the host, the drivers have to be reinstalled following the standard installation procedure.

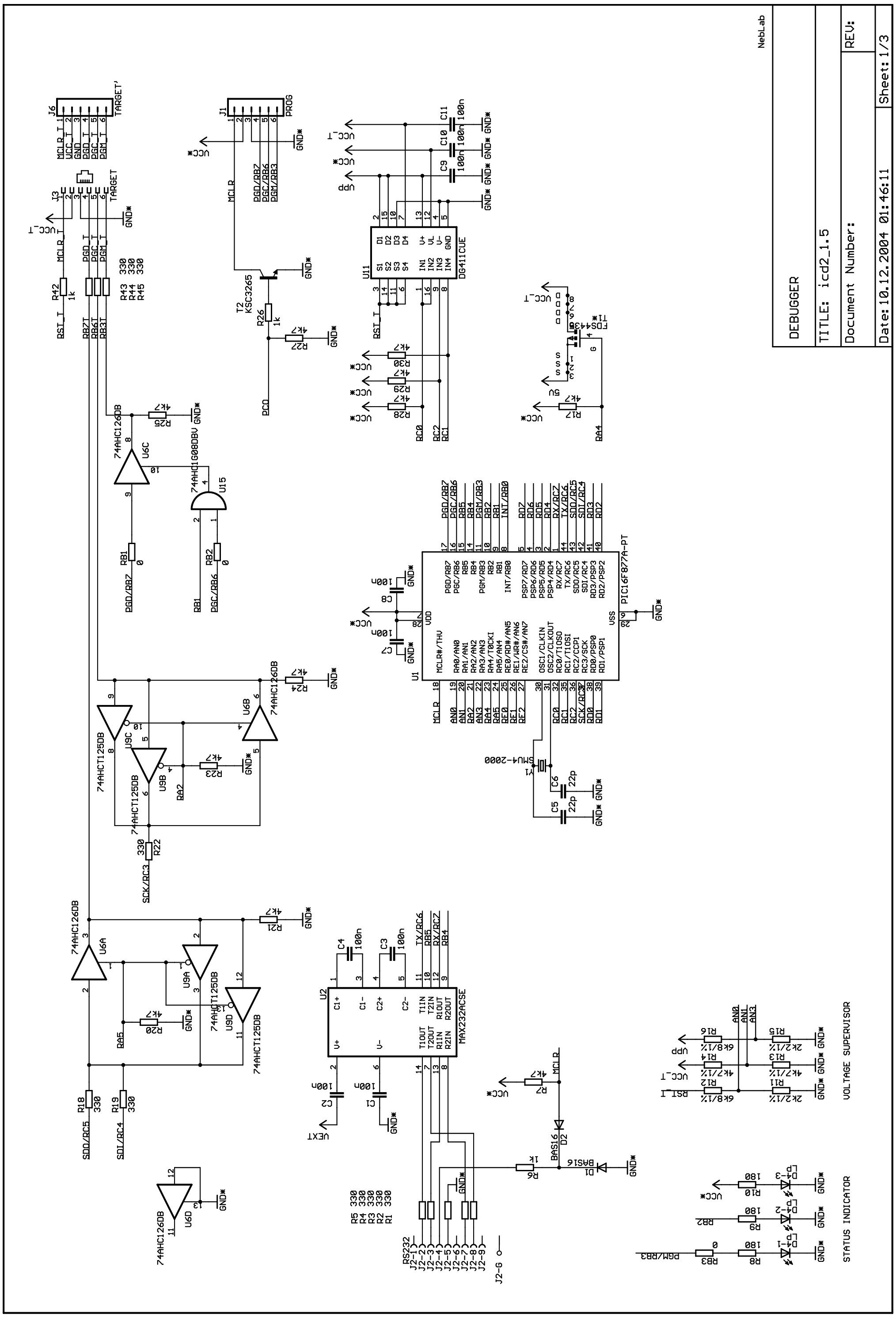
4 Thanks

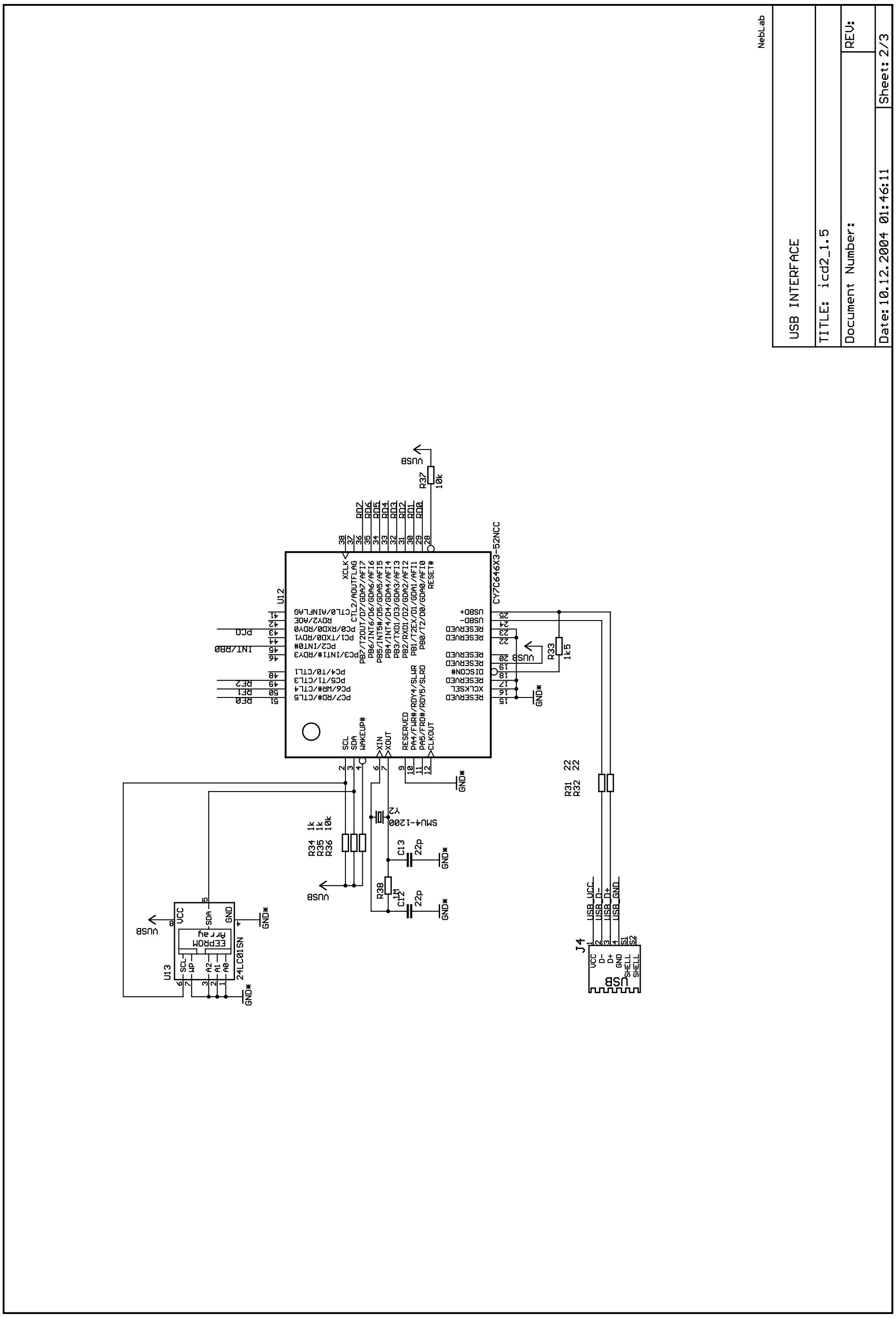
Thanks to everyone who helped realizing this project. Special thanks go to *zaphod42* and *crazyduck* living at www.edaboard.com.

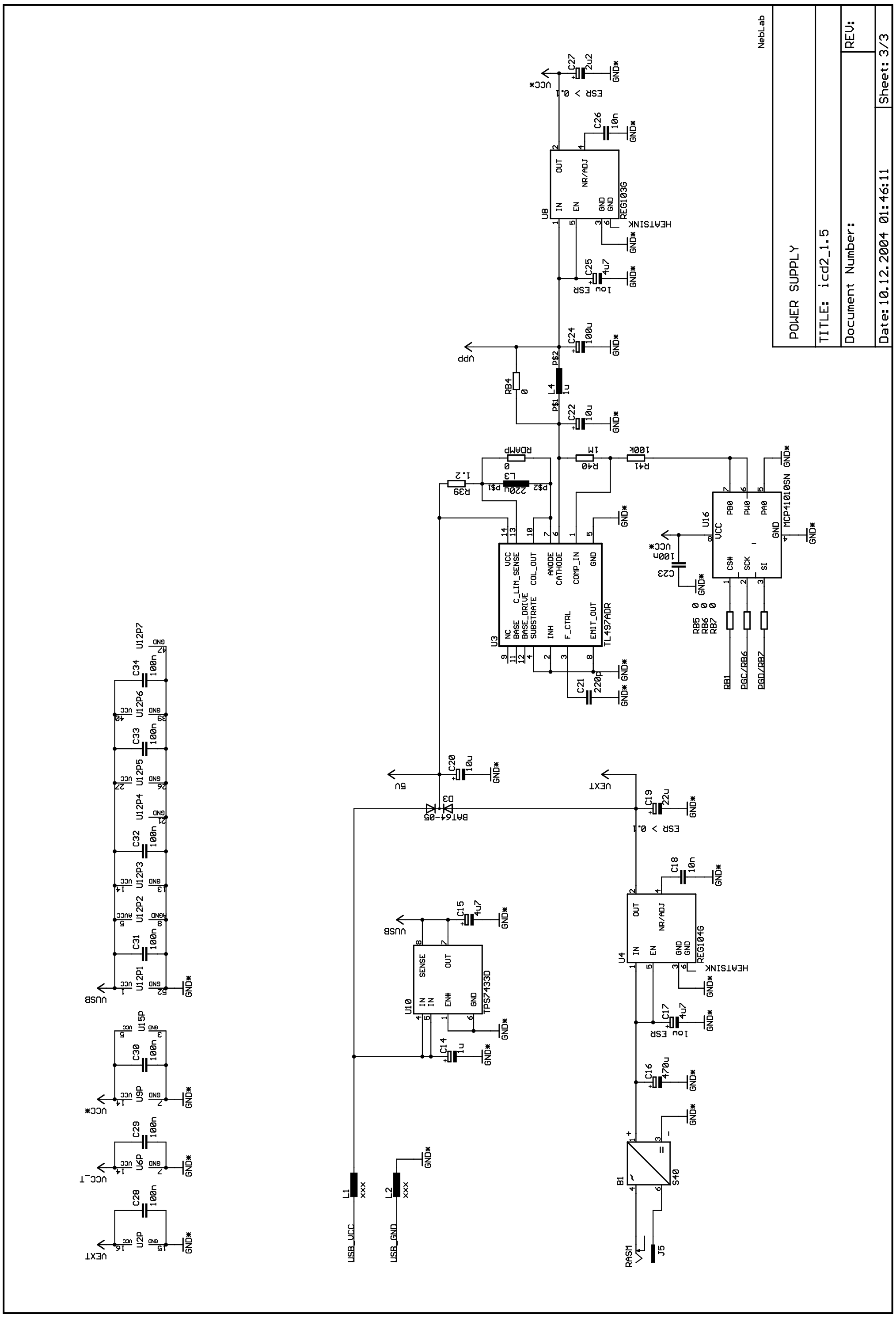
ROCK ON!

A Appendix

A.1 Schematic




NebLab
USB INTERFACE
TITLE: icd2_1.5
Document Number:
REV:
Date: 10.12.2004 01:46:11
Sheet: 2/3
CY7C646X3-52NCC
U12
U13
24LC01SN
EEPROM
Array
J4
USB
Y2
SMU4-1200
R37 10k
R33 1k5
R31 22
R32 22
R34 1k
R35 1k
R36 10k
R38
C13 22p
C12 22p
VUSB
GND*
PC0
INT/RB0
RE2
RE1
RE0
USB_VCC
USB_D-
USB_D+
USB_GND


NebLab
POWER SUPPLY
TITLE: icd2_1.5
Document Number:
REV:
Date: 10.12.2004 01:46:11
Sheet: 3/3

A.2 PCB Masks

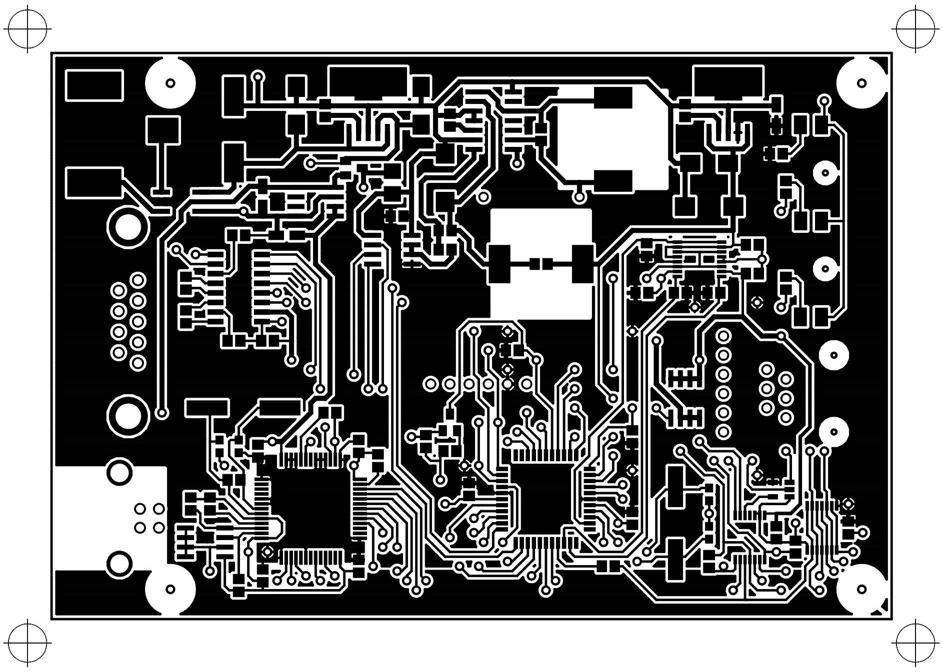


Figure 1: Top (mirrored)

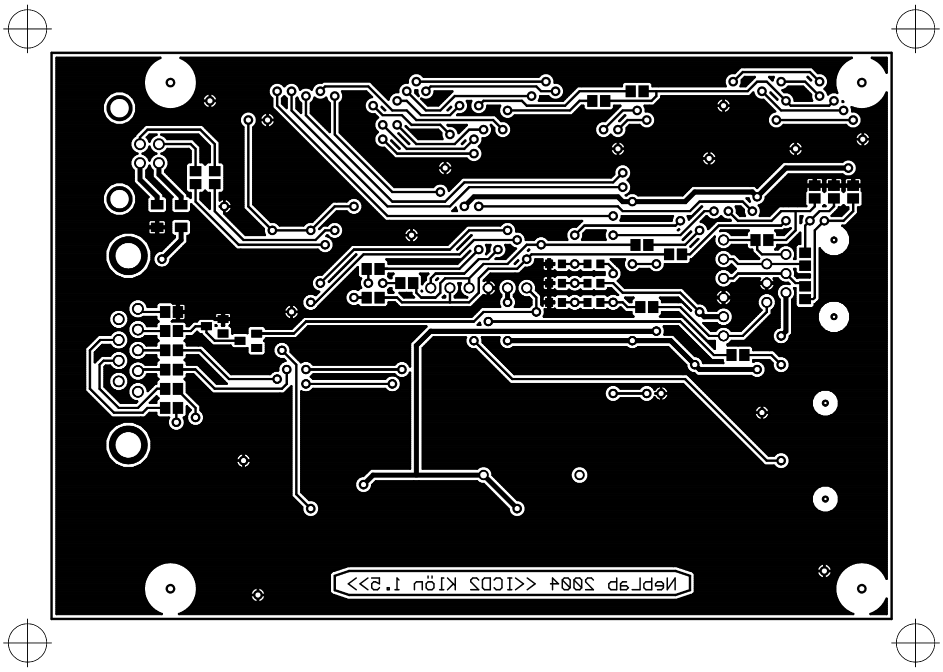


Figure 2: Bottom (mirrored)

A.3 Component Placement Drawing

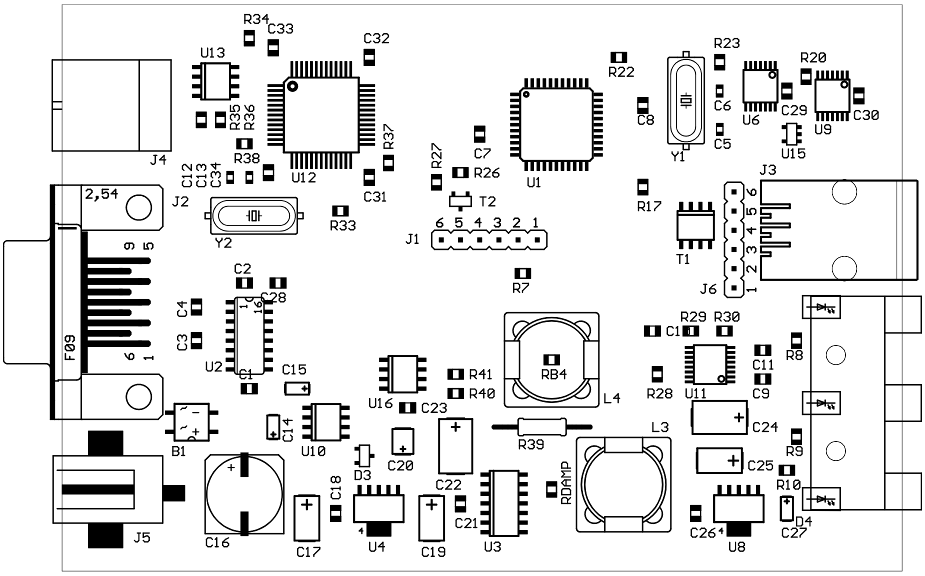


Figure 3: Top

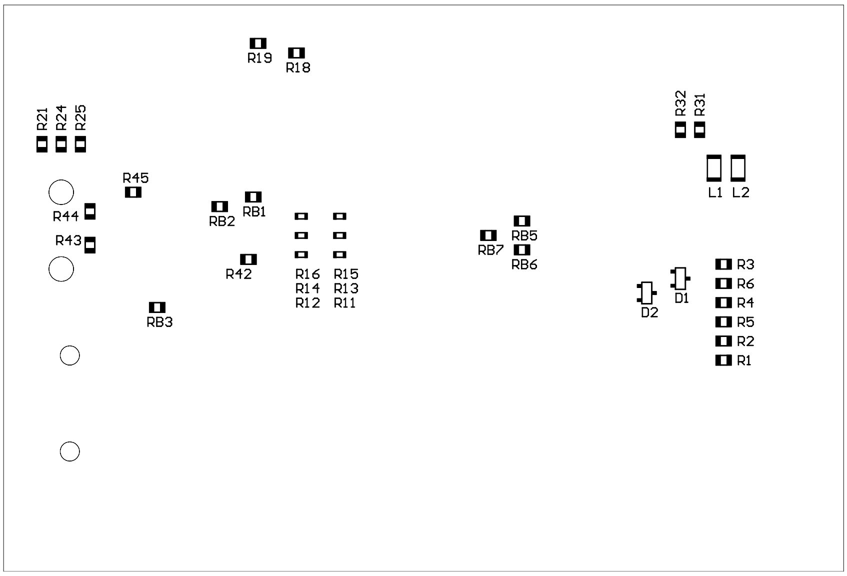


Figure 4: Bottom

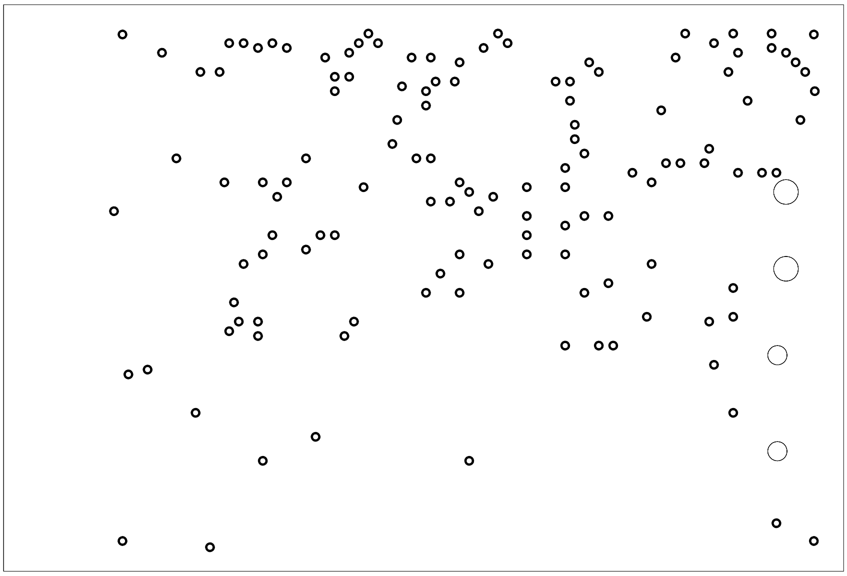


Figure 5: Vias

A.4 Bill of Material

Design Name.	Component Value/Description	Component Package
B1	Bridge 0.8A	SOP6
C14	1u	CT3216
C27	2u2	CT3216
C15	4u7	CT3216
C17	4u7 low ESR	CT6032
C25	4u7 low ESR	CT6032
C20	10u	CT3528
C22	10u	CT7343
C19	22u	CT6032
C24	100u	CT7343
C16	470u	Q10X10.5
C5	22p	C0603
C6	22p	C0603
C12	22p	C0603
C13	22p	C0603
C21	220p	C0805
C18	10n	C0805
C26	10n	C0805
C1	100n	C0805
C2	100n	C0805
C3	100n	C0805
C4	100n	C0805
C7	100n	C0805
C8	100n	C0805
C9	100n	C0805
C10	100n	C0805
C11	100n	C0805
C23	100n	C0805
C28	100n	C0805
C29	100n	C0805
C30	100n	C0805
C31	100n	C0805
C32	100n	C0805
C33	100n	C0805
C34	100n	C0805
D1	BAS16	SOT23
D2	BAS16	SOT23
D3	BAT64-05	SOT23
D4-1	LED red	
D4-2	LED green	
D4-3	LED yellow	
LP	Lightpipe	
J1	PROG	IDC6
J6	TARGET	IDC6
J2	RS232	SUBD25

J3	TARGET	RJ12
J4	USB Connector	
J5	Power Jack	
L1	EMV Ferrite Beads	3216
L2	EMV Ferrite Beads	3216
L3	220uH	DR
L4	1uH	DR
RB1	0	R0805
RB2	0	R0805
RB3	0	R0805
RB4	0	R0805
RB5	0	R0805
RB6	0	R0805
RB7	0	R0805
R39	1.2	0207/10
R31	22	R0805
R32	22	R0805
R8	180	R0805
R9	180	R0805
R10	180	R0805
R1	330	R0805
R2	330	R0805
R3	330	R0805
R4	330	R0805
R5	330	R0805
R18	330	R0805
R19	330	R0805
R22	330	R0805
R43	330	R0805
R44	330	R0805
R45	330	R0805
R6	1k	R0805
R26	1k	R0805
R34	1k	R0805
R35	1k	R0805
R42	1k	R0805
R33	1k5	R0805
R7	4k7	R0805
R17	4k7	R0805
R20	4k7	R0805
R21	4k7	R0805
R23	4k7	R0805
R24	4k7	R0805
R25	4k7	R0805
R27	4k7	R0805
R28	4k7	R0805
R29	4k7	R0805
R30	4k7	R0805

R36	10k	R0805
R37	10k	R0805
R41	100k	R0805
R40	1M	R0805
R38	1M	R0805
R11	2k2/1%	R0603
R15	2k2/1%	R0603
R13	4k7/1%	R0603
R14	4k7/1%	R0603
R12	6k8/1%	R0603
R16	6k8/1%	R0603
T1	FDS4435	SOT-23
T2	KSC3265YMTF	SO-8
U1	PIC16F877A	TQFP44
U2	MAX232A	SOIC16
U3	TL497AIDR	SOIC14
U4	REG104GA-5	SOT223-5
U5	SN74AHC1G04DBVR	SOT23-3
U6	SN74AHC126DBR	SSOP14
U7	SN74AHC1G04DBVR	SOT23-3
U8	REG103UA-5	SOT223-5
U9	SN74AHC125DBR	SSOP14
U10	TPS7433	SOIC8
U11	DG411CUE	SSOP16
U12	CY7C64613	SQFP-S-10X10-52
U13	24LC01B-I/SN	SOIC8
U14	X	
U15	SN74AHC1G08DBVR	SOT23-3
U16	MCP41010-I/SN	SOIC8
Y1	Crystal	20MHz
Y2	Crystal	12MHz