

Escher Manual

Version: 1.0,

Winfried Ritsch, Atelier Algorithemics

(last updated November 1, 2011)

November 1, 2011

This manual is a short description of the escher board v1.0 and is published under <http://algo.mur.at/projects/microcontroller/escher>. Please feel free to report other tests and send me correction etc. The board was made with open hardware concept in mind and was released as “CC by NC-SA” (creative commons by non-commercial use attribution), which means hopefully it will be developed further on each release.

1 Introduction

The escher is a simple micro-controller board for the use of *dsPIC33F128MC708* or any other PIN-compatible PIC with 80 pins like *PIC32*. The purpose of the board is to interface different hardware boards like escherFET and sensors with partly or fully assembled board.

The board can be interfaced by ethernet over the [ENC28J60](#) or one of the serial interfaces from a host and programmed with the ICSP from the ICD2 or any other debugger and programmer, including boot-loader.

It has a jumper-boards for configuration like addressing or using them as dedicated interfaces for testing hardware and four IDC ribbon-connectors each 10, 14 and 2 with 40-pins (IDE-Cables). Here I will describe the main design goals and usages of the board as an example. The board is highly configurable and can be used for many purposes.

2 dsPIC Controller

The dsPIC Controller used is mounted on as 80-PIN so it is easy to solder. The board is targeted to *dsPIC33FJ128MC708* but also a *Pin compatibles* can be used. In figure 2 the pin assignment is shown. The Reset is connected to button and also is used by the ICSP¹. The board is derived from the microchip development board Explorer 16 with the PICtail Ethernet interface.

The PIC can be clocked with a Oscillator, which is also used for the Ethernet processor ENC28J60, therefore 25MHz is used, but also the internal RC can be used. I recommend the oscillator since seems to be the most stable solution. The frequency can be chosen with internals PLLs and Prescaler.

3 The ISCP connector

On the ISCP-Connector a Debugger or Programmer can be attached for in-circuit programming or debugging the dsPIC.

The pin assignment is made so it is compatible with the PICStart+² and compatible Kits³.

Programming could also be done over a boot-loader (see software description).

4 Power Supply

As a power supply 3.3V, 5V stabilized can be used or 7-12V if the second regulator is soldered. Also these voltages can be supplied over the PIN-boards from addons, which often has higher power drain than the processor. Care should be taken considering supplying add-on boards over this board, since the regulators can be too weak and a voltage breakdown can reset the board.

¹In-Circuit Serial Programming

²Programmer/Debugger by Microchip, Pin 6 not used

³for PicKit3 a adapter is needed

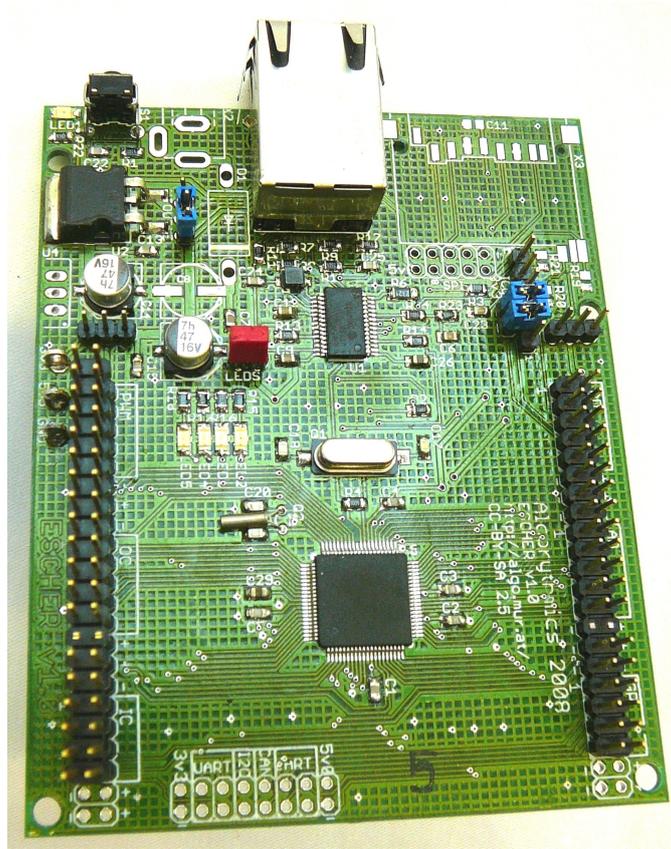


Figure 1: escher with ethernet interface

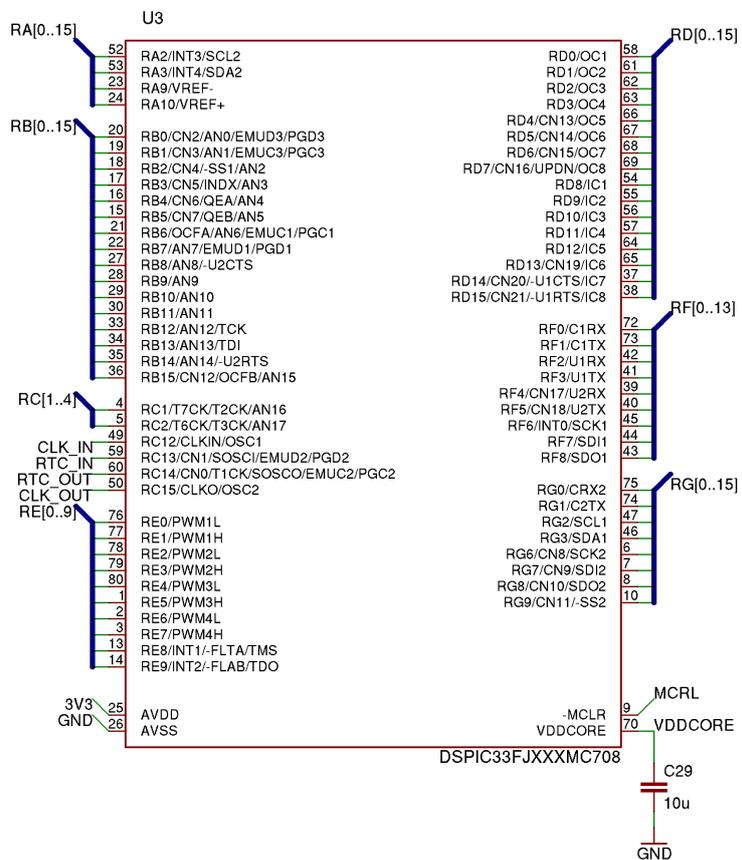


Figure 2: dsPIC33FJ128MC708 pin assignment

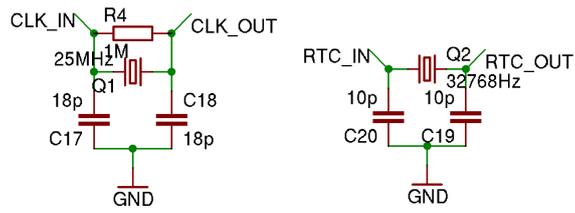


Figure 3: mcu, ethernet clock and rtc clock

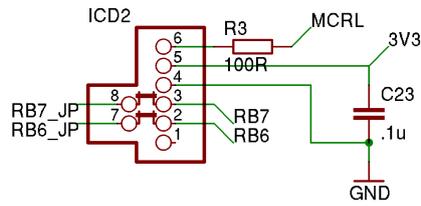


Figure 4: ICD connector

Pin	ICSP
1	Reset
2	+5V
3	ground
4	PGD of PIC
5	PGC of PIC

Table 1: Pinout ICSP (PGD and PGC must not be jumpered)

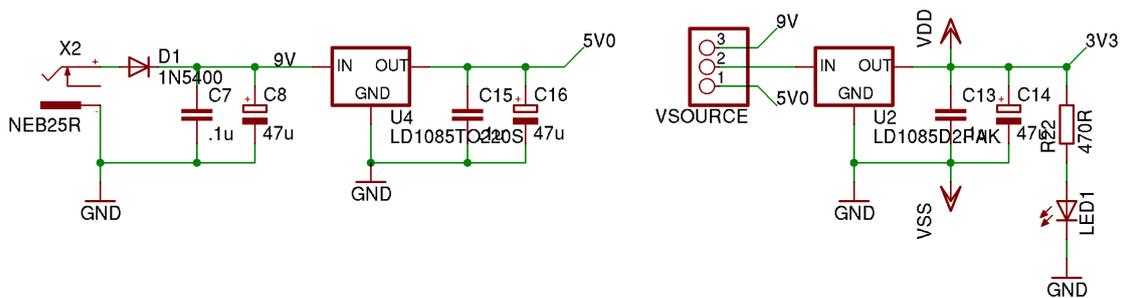


Figure 5: power supply circuits for many options

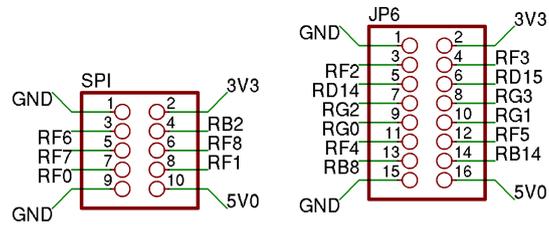


Figure 8: serial Ports

Pin	Port	UART, CAN, I2C	alternate
1	-	GND	
2	-	+3V	
3	RF2	U1RX	
4	RF3	U1TX	
5	RD14	-U1CTS	CN20, IC7
6	RD15	-U1RTS	CN21, IC8
7	RG2	SCL1	
8	RG3	SDA1	
9	RG0	CRX2	
10	RG1	C2TX	
11	RF4	U2RX	CN17
12	RF5	U2TX	CN18
13	RB8	-U2CTS	AN8
14	RB14	-U2RTS	AN14
15	-	GND	
16	-	+5V	

Figure 9: Pinout BUS with UART, CAN and I2C as main functions

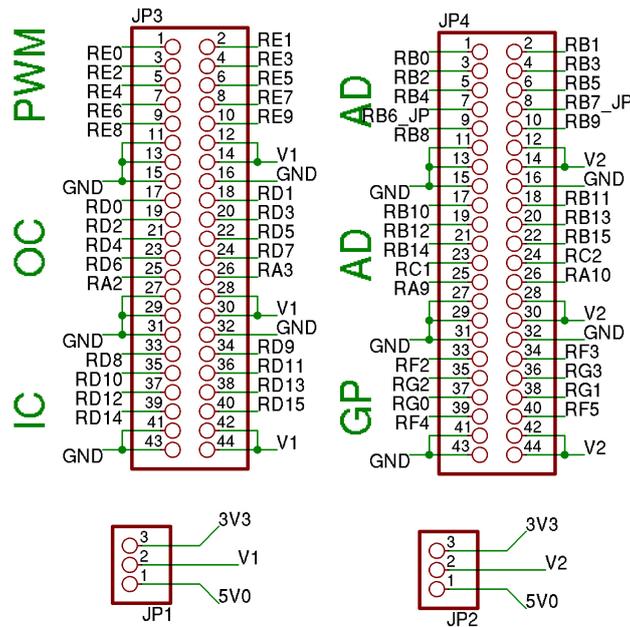


Figure 10: PWM and AD or GPIO ports

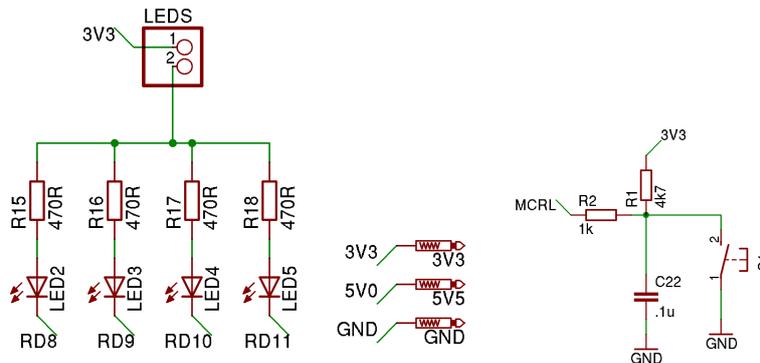


Figure 11: LEDs, PADS, Reset etc,

8 PWM and AD-Ports

These ports are made for multipurpose use and special use for planned projects. The trick is that on these ports power supply is switchable either supplied by escher or the the addon can supply escher. Therefore for each of the ports there is a jumper, which can select a 3V or 5V supply for V1, V2 on the ports.

These ports can be used with different sizes of ribbon cable with connectors for easy wiring. If 26-Pin ribbon connectors are used, PIN 27,28 can be cut out so it fits. Also for 10-Pin connectors 4 of the pins can be cut so they fit. Generally 40-PIN ribbon cables (easy accessible IDE-cables) will use all of them.

9 LEDs, PADS, Reset, etc

The LEDs are on board for Debugging and also status on Ethernet connections, data parsed or anything else. LEDs can be powered off by a jumper to reduce power supply. The pads are for debugging and testing and the reset for resetting the mcu, which also can be done by powering off, but sometimes when FETs or any other power electronics is depended its often safer to reset.

10 Board Layout and Devices

Please for more information look at the eagle files, schematics. For better understanding, the board layout has are shown at the end and a list of devices used.

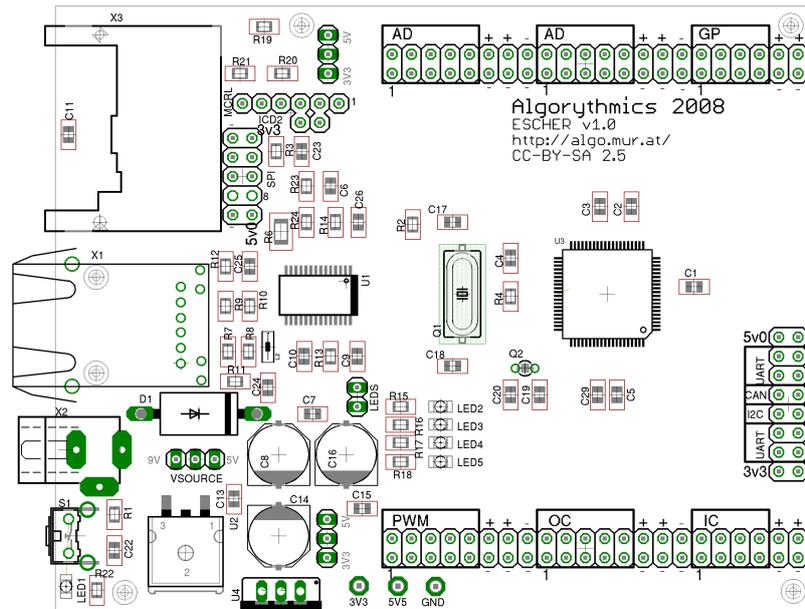


Figure 12: escher Board with devices

Menge	Wert	Device	Bauteile
1		31-XX	S1
5		LEDSMT1206	LED1, LED2, LED3, LED4, LED5
1		PINHD-1X2	LEDS
3		PINHD-1X3	JP1, JP2, VSOURCE
1		PINHD-2X5	SPI
1		PINHD-2X8	JP6
2		PINHD-2X22	JP3, JP4
1		WE-CBF_1210	L2
13	0.1u	C-EUC0805	C1, C2, C3, C4, C5, C6, C7, C13, C15, C22, C23, C24, C25
2	1M	R-EU_M0805	R4, R13
1	1N5400	1N5400	D1
2	1k	R-EU_M0805	R2, R24
3	1u	C-EUC0805	C9, C10, C11
1	2k32+/-1%	R-EU_M1206	R6
2	47k	R-EU_M0805	R19, R20
2	4k7	R-EU_M0805	R1, R23
2	10p	C-EUC0805	C19, C20
2	10u	C-EUC0805	C26, C29
2	18p	C-EUC0805	C17, C18
1	25MHz	CRYSTALHC49UP	Q1
3	47u	CPOL-EUE	C8, C14, C16
4	49k9 +/-1%	R-EU_M0805	R7, R8, R9, R10
1	100R	R-EU_M0805	R3
9	470R	R-EU_M0805	R11, R12, R14, R15, R16, R17, R18, R21, R22
1	32768Hz	CRYSTALTC26V	Q2
1	AMPHENOL_RJ45JACK	AMPHENOL_RJ45JACK	X1
1	DSPIC33FJXXMC708	DSPIC33FJXXMC708	U3
1	ENC28J60	ENC28J60	U1
1	FPS009-3001	FPS009-3001	X3
1	ICD2	ICD2	ICD2
1	LD1085D2PAK	LD1085D2PAK	U2
1	LD1085TO220S	LD1085TO220S	U4
1	NEB25R	NEB25R	X2
3	PTR1PAD1-17	PTR1PAD1-17	3V3, 5V5, GND

Figure 13: List of devices for escher

11 Disclaimer

I cannot be held responsible for the information contained within this docs and hardware documented here, even I tried hard not to make failures but nobody is perfect. Any uses and/or misuse of the information that could result in bodily and/or mental harm to you or any other person that came in contact with the information cannot hold me or any company or anyone else responsible.