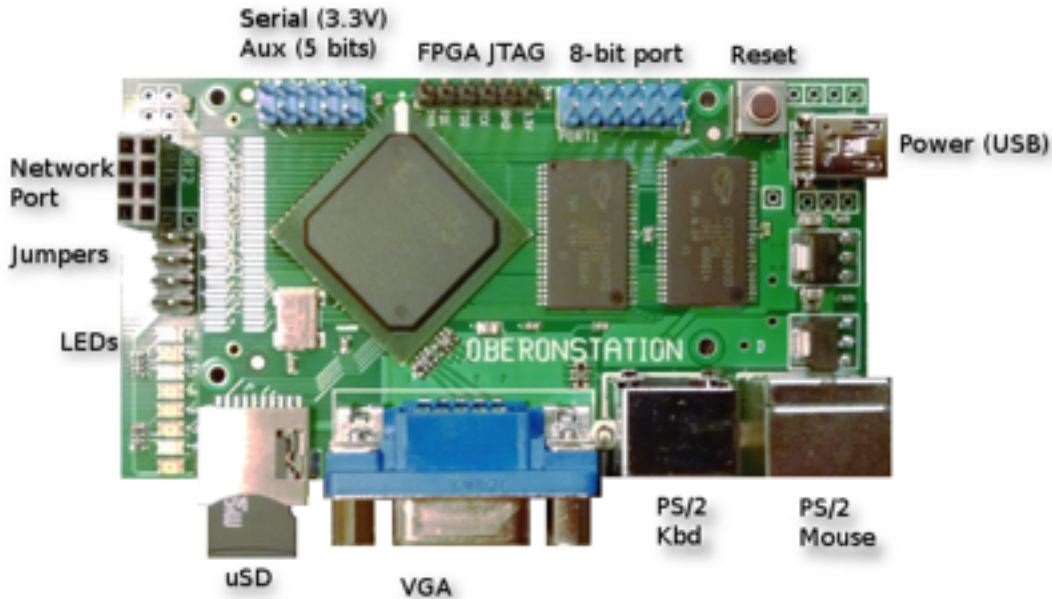


RISC5 OberonStation Quick Reference Guide

PDR rev. B 24.11.2015



Oberon & boot loader LED Codes (*BootLoad.Mod, Oberon.Mod*)

7 6 5 4 3 2 1 0													
■	□	□	□	□	□	□	□	□	□	□	□	80H	Boot loader running, initialising SD-Card
■	□	□	□	□	□	□	■	□	□	□	□	81H	Booting from serial (jumper J0 on)
■	□	□	□	□	□	■	□	□	□	□	□	82H	Booting from SD-Card (J0 off)
■	□	□	□	■	□	□	□	□	□	□	□	84H	Loading successful, transferring control to Oberon
□	□	■	□	□	□	□	□	□	□	□	□	20H	Garbage collection complete (ESC to clear)

Jumpers: read swi data port -60 (0FFFC4H), bits 7, 2..0

☒	□	□	☒	J0 fitted: boot from serial instead of SD-Card (bit 0, SW0 on S3BOARD)
☒	□	□	☒	J1 & J2: unused (bits 1 & 2, SW1 & SW2 on S3BOARD)
J3	J2	J1	J0	J3 fitted: inverse video (bit 7, SW7 on S3BOARD)

(lower row of 4 pins closest to labels is GND, and can be used to 'park' unused jumpers)

SPI: read/write SPI data port -48, read/write SPI control port -44 (0FFFD0H, 0FFFD4H)

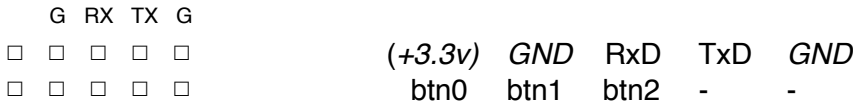
PORT2							
■	■	■	■	MISO	SCLK	NEN	GND
■	■	■	■	-	MOSI	SS	+3.3v

control port bit	write	read
0	select SD-Card	0 = SPI busy, 1 = ready
1	select PORT2 (network)	-
2	0 = slow/byte mode (~400Kbit), 1 = fast/word mode (~8Mbit)	-
3	network enable (NEN)	-

Data port write triggers an SPI read and write of a byte or word (word is littleendian byte-order), default is 0 (no NEN, slow/byte, both slaves disabled). Slave-selects are independent - NEN and GPIO (overleaf) can be used as further device slave-selects if required.

Serial and Aux port

read/write RS232 data port -56 bits 7..0, read/write RS232 control port -52 (0FFFC8H, 0FFFCCH)
 read aux port -60 (0FFFC4H), bits 10..8



data port write initiates an RS232 transmit of a byte (even if transmitter is already busy)

control port bit	write	read
0	0 = 19200 baud (default), 1 = 115200 baud	0 = no data available 1 = byte received since last data read
1	-	0 = transmitter busy 1 = transmitter ready

RS232 line protocol is 8 bits, one start bit, one stop bit, no parity. **NOTE THIS IS A 3.3v-level PORT, intended for USB-RS232 converter cables - DO NOT CONNECT A STANDARD RS-232 CABLE!**

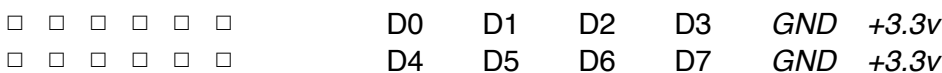
The aux port has 3 internally-pulled-down btn0..2 lines suitable for connecting pushbuttons to +3.3v - these states are available by reading the swi port at -60, bits 8, 9, 10 respectively. (Note this is the opposite of the jumpers, which have internal pullups and inverters, and are connected to GND to activate.)

JTAG programming port



(used for factory-programming the Xilinx Spartan 3AN FPGA with the RISC5 configuration)

8-bit port (GPIO), read/write data port -32, write control port -28 (0FFFE0H, 0FFFE4H), bits 7..0

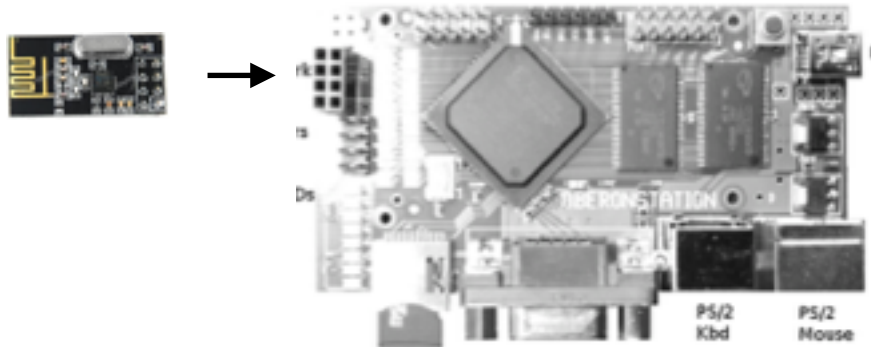


PORT1

data port reads state of pins D7..D0 in bits 7..0, data write bits 7..0 sets D7..D0 output latches
 0 = low (GND), 1 = high (3.3v)

control port write enables output on corresponding bit (bit = 1), or disables output (bit = 0), i.e. port bit is an input - output is tri-state (default)

Nordic nRF24L01+ 2.4GHz Wireless Networking Module



Fit to OberonStation's SPI port2 header (disconnect power first!) with PCB aerial facing *outwards*

SCC Packet Format: `seqno dadr sadr typ len:4 [payload...]`

e.g. TIM time server response: 01 FF 00 47 04 00 00 00 57 AC F0 3E

01	non-zero = valid packet
FF	broadcast destination adr
00	server source address
47	TIM time response packet type
04 00 00 00	payload length 4
57 AC F0 3E	payload: 001111 1011 11000 01010 110001 010111 (15/11/24 10:49:23)

(all values in hex)

nRF24L01+ Initialisation Example for SCC Broadcast Receive

reg	val	comment
00	7F	CONFIG: mask ints, CRC16, power up, receiver
01	00	EN_AA: disable auto-acknowledgement
04	00	SETUP_RETR: disable auto-retransmit
05	channel	set RF channel (2400 + channel MHz)
06	07	RF_SETUP: 1Mb/s, 0dBm
07	70	STATUS: clear flags
11	20	RX_PW_P0: rx pipe 0 payload width 32 bytes

(all values in hex)

Common nRF24L01+ SPI Commands

cmd byte	data bytes	comment
000r rrrr	1 to 5	R_REGISTER: read status and register rrrr (00h-1Fh) value
001r rrrr	1 to 5	W_REGISTER: read status, write value
0110 0001	1 to 32	R_RX_PAYLOAD: read status and payload
1010 0000	1 to 32	W_TX_PAYLOAD: read status, write payload
1110 0001	0	FLUSH_TX: flush tx FIFO
1110 0010	0	FLUSH_RX: flush rx FIFO

(cmd values in binary)