OwlBoard

rev 3.2

Features

- Xilinx® Spartan 6™ XC6SLX9 FPGA
- 8 Mb on-board flash memory
- JTAG programmable
- USB programmable[1]
- 100 MHz on-board crystal clock source
- USB powered
- 8 on-board LEDs
- 4 on-board pushbuttons
- Two 40-pin expansion headers
- Large 14x30 through-hole prototyping/development area

Table of Contents

[1] With additional hardware, which has not been provided or tested. See Programming section.

WARNING

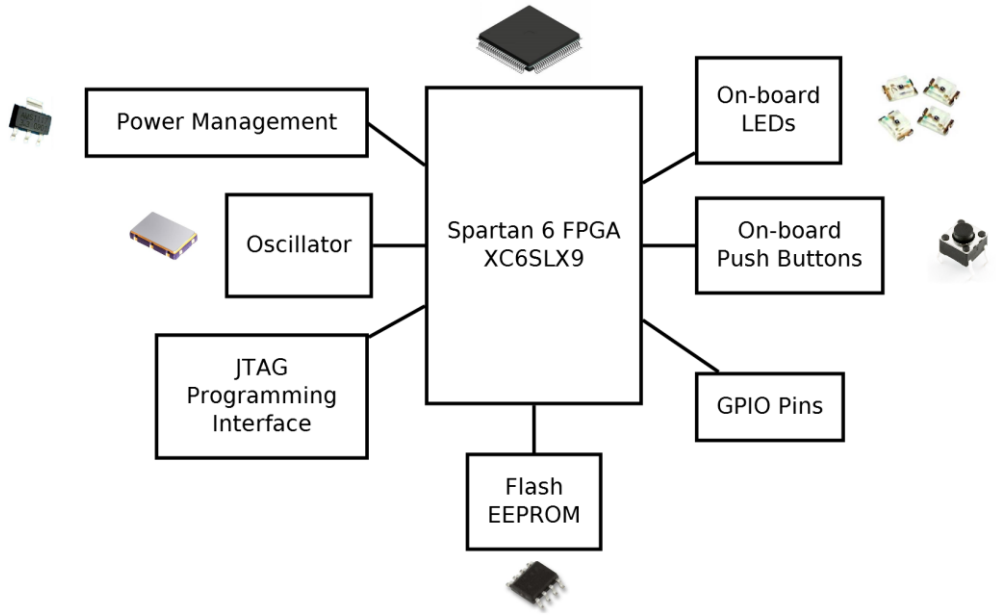
Failure to read and adhere to the instructions in this manual may cause permanent and irreversible damage to your OwlBoard and computing equipment.

Use at your own risk.

1.0 Hardware

This section describes the basic hardware features of the OwlBoard. For more specific part selections and PCB references, consult the BOM in the References section of this document.

1.1 Hardware Overview



The OwlBoard v3 is a solid foundation to explore the fundamentals of digital logic, configurable hardware, and programmable logic devices. Not only does this board provide basic I/O components like buttons and LEDs, the board offers a large prototyping area for custom circuits, greatly increasing the number of possibilities for expansion.

Two 40-pin GPIO headers on either end of the board allow for direct connection to the FPGA's logic pins. Headers on both sides of the prototyping area of the board offer 3.3V and GND busses.

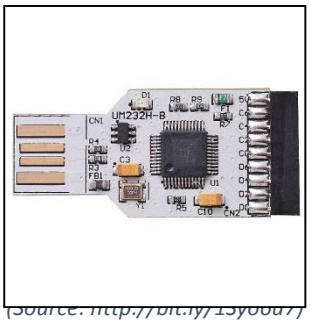
Because the Flash EEPROM chip is connected to the FPGA in a 1-wire SPI configuration, it has been noted that non-volatile programming times can take a significant amount of time. However, because this configuration is so simple to implement, it is still currently used on the OwlBoard.

1.2 Additional hardware for USB compatibility

As noted previously, the OwlBoard v3 does support JTAG communication over USB with the additional hardware. This hardware is included but has been greyed out in the Bill of Materials. It should be noted that for Oregon Tech classes, this additional hardware will not be provided to you. Also note that this hardware and programming configuration has not been thoroughly tested and should be used at the user's own risk.

2.0 Programming

The OwlBoard v3.0 features both JTAG and USB programming capabilities; however, the hardware components for the USB programming are not provided and are currently untested. Instead, to program the OwlBoard, we will use an external FTDI to JTAG adapter. The recommended adapter is the UM232 (Digi-Key Part# 768-1159-ND).



The UM232 features the FT232H integrated circuit, which can be used as a JTAG, SPI, and I2C adapter. By installing XCS3PROG (Xilinx FPGA programming tool) and FTDI D2XXX, we can use the UM232 as a USB to JTAG converter in order to program the OwlBoard v3.0.

2.1 Software Requirements

Whether using Windows or Linux, a valid copy of Xilinx ISE is required to synthesize HDL code into a binary file, readable by the FPGA. Consult Xilinx's online resources to determine which version of ISE is compatible with the FPGA being used on the OwlBoard.

In addition to Xilinx ISE, the following software is required.

Install the following for Windows machines:

- XC3SPROG - *http://sourceforge.net/projects/xc3sprog/*
- FTDI D2XXX drivers - *http://www.ftdichip.com/Drivers/D2XX.htm*

Install the following for Linux machines:

- LIBFTDI - *https://github.com/open-ephys/GUI/wiki/libFTDI*

2.2 Hardware Setup

The following section outlines common hardware setup for both volatile and non-volatile programming modes. Please follow these wiring instructions carefully. Routing connections incorrectly can result in electrical shorts, rendering your OwlBoard and/or computer inoperable.

WARNING: If your FTDI UM232H-B adapter has wires built in, you must remove all unused wires.

WARNING: Do not connect UM232H-B power line and micro USB power at the same time!

Connect the cable as shown in Table 1.

FTDI UM232 USB to JTAG adapter marking	OwlBoard Pin	Warnings
D3	TMS	
D2	TDI	
D1	TDO	
D0	TCK	
Black wire / GND	GND	
Red wire / 5V	5V	Do not use. If wire present, desolder or heat shrink each wire individually
Any other wires	N/A	Do not use. If wire present, desolder or heat shrink each wire individually

Table 1 - USB to JTAG adapter connections

Note that the FT232H used on the UM232 can be used in a wide variety of configurations. See the FT232H datasheet for more information.

A partial copy of the pin configuration from FTDI data sheet is shown in Figure 2 below as a quick reference.

Pin		Pin functions (depends on configuration)								
Pin #	Pin Name	ASYNC Serial (RS232)	SYNC 245 FIFO	STYLE ASYNC 245 FIFO	ASYNC Bit-bang	SYNC Bit-bang	MPSSE	Fast Serial interface	CPU Style FIFO	FT1248
13	ADBUS0	TXD	D0	D0	D0	D0	TCK/SK	FSDI	D0	MIOSI0
14	ADBUS1	RXD	D1	D1	D1	D1	TDI/DO	FSCLK	D1	MIOSI1
15	ADBUS2	RTS#	D2	D2	D2	D2	TDO/DI	FSDO	D2	MIOSI2
16	ADBUS3	CTS#	D3	D3	D3	D3	TMS/CS	FSCTS	D3	MIOSI3

Figure 2 - FTDI Datasheet Table showing pin connections and descriptions

This FTDI datasheet can be found at the following URL:
http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232H.pdf

2.3 Volatile Programming Mode

In this mode of programming, the FPGA is programmed directly. When power is removed after programming, the design configuration will be lost and the FPGA must be reprogrammed.

1) In Xilinx ISE, verify that Generate Programming File completes successfully by checking for a green checkbox as shown in Figure 3.

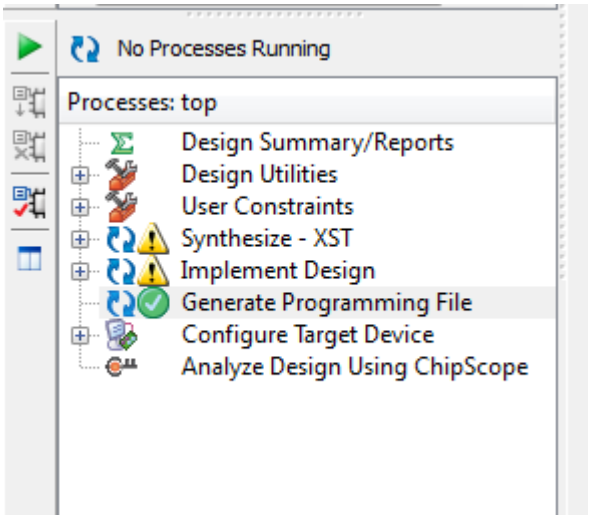


Figure 3 – Bit file generated

2) Navigate to where your project is stored and locate the *.bit* file. The *.bit* file will be named after the top-level file in your project hierarchy.

3) Copy the *.bit* file to the location where you unzipped *xc3sprog.exe*.

4) Open a terminal window. In Windows, do this by pressing ⊞ + R, type “CMD”, then click “OK”.

5) Type the following command in the terminal window. Note that you may need to substitute 'ft232h' for 'ftdi'

<xc3sprog.exe location> -c ftdi –v –p 0 *<.bit filename>*

For this example, the command is shown in Figure 3.

```
Z:\desktop>xc3sprog -c ftdi -v -p 0 top.bit
```

Figure 4 - Command to program the FPGA

-c	Indicates selection of cable
ftdi	Indicates that we are using an FTDI interface
-v	Enables verbose output
-p	Indicates position in JTAG chain
0	We are selecting position 0
top.bit	The name of our *.bit* file generated by Xilinx ISE

Table 2 - Volatile programming command elements description

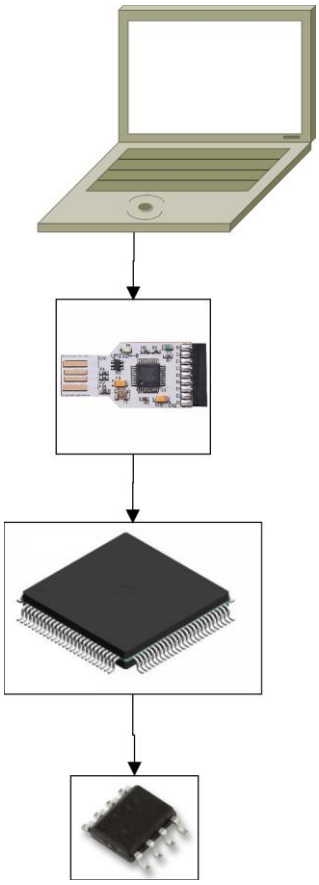
6) After successful programming, you should receive something similar to the following output shown in Figure 5.

```
Z:\desktop>xc3sprog -c ftdi -v -p 0 top.bit
XC3SPROG (c) 2004-2011 xc3sprog project $Rev$ OS: Windows
Free software: If you contribute nothing, expect nothing!
Feedback on success/failure/enhancement requests:
        http://sourceforge.net/mail/?group_id=170565
Check Sourceforge for updates:
        http://sourceforge.net/projects/xc3sprog/develop

Using built-in device list
Using built-in cable list
Cable ftdi type ftdi VID 0x0403 PID 0x6010 dbus data 00 enable 0b cbus data 00
ata 00
Could not open FTDI device (using libftdi): device not found
Using FTD2XX, Using JTAG frequency   1.500 MHz from undivided clock
JTAG chainpos: 0 Device IDCODE = 0x24001093     Desc: XC6SLX9
Created from NCD file: top.ncd;UserID=0xFFFFFFFF
Target device: 6slx9tqg144
Created: 2015/06/05 17:40:35
Bitstream length: 2724832 bits
DNA is 0x1990667bd8ebcaff
done. Programming time 1825.2 ms
USB transactions: Write 180 read 11 retries 0

Z:\desktop>
```

Figure 5 - Volatile programming results

NOTE: This screen shot was taken on a Windows machine. Windows uses the FTD2XXX driver not the libftdi driver, so it displays an error indicating that libftdi cannot be found. This is normal.

2.4 Non-volatile Programming



Because FPGAs do not typically have non-volatile memory to store programs and configuration data more permanently, it is necessary to utilize external memory storage for more permanent applications. The OwlBoard uses a M25P80 Micron 8 Mb flash chip, connected to the Xilinx Spartan 6 FPGA in an SPI configuration. The following instructions outline how to program the SPI flash memory chip.

First, you will load the *bst_bit.bit* file to the FPGA. The *bst_bit.bit* file is available on the course website or as part of the OwlBoard package. The *bst_bit.bit* file contains the logic needed to pass the bit file through to the SPI flash device. The source code for the *bst_bit.bit* file is also available as part of the OwlBoard package. Then, after the *bst_bit.bit* file has been loaded onto the FPGA, the desired bit file to be stored in memory can be sent, as is explained in the steps below.

1) Open a terminal window. In Windows, do this by pressing ⊞ + R, type "CMD", then click "OK".

2) Enter the following command for Windows:

```
Z:\desktop\xc3sprog –c ft232h bst_bit.bit
```

xc3sprog	The loader program
-c ft232h	Use the FT232H chip as USB to JTAG bridge
bst_bit.bit	The bit file containing the SPI loader

Table 3 - Non-volatile command description

3) Wait for bit file to transfer to the FPGA.

```
          :~/Desktop/xc3sprog/xc3sprog$ sudo ./xc3sprog -c ft232h bst_bit.bit
XC3SPROG (c) 2004-2011 xc3sprog project $Rev: 774 $ OS: Linux
Free software: If you contribute nothing, expect nothing!
Feedback on success/failure/enhancement requests:
        http://sourceforge.net/mail/?group_id=170565
Check Sourceforge for updates:
        http://sourceforge.net/projects/xc3sprog/develop

Using Libftdi,
DNA is 0x1990667bd8ebcaff
```

Figure 6 - Non-volatile programming results

NOTE: This screenshot is from a Linux computer, and uses *libftdi* to program the FPGA. Compare with Figure 5, the Windows screenshot of *xc3sprog* which displays an error.

Figure 5 shows that the program was successfully loaded. Xc3sprog will return the Device DNA for the Spartan 6 FPGA.

4) In Xilinx ISE, verify that Generate Programming File completes successfully by checking for a green checkbox as shown in Figure 7.

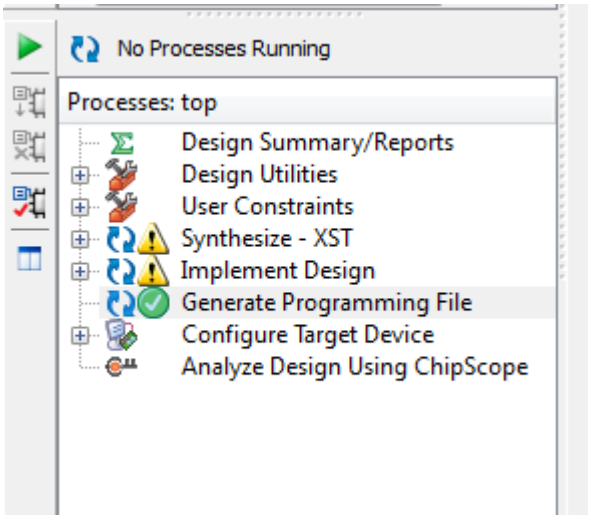


Figure 7 - Bit file generated

5) Navigate to where your project is stored and locate the *.bit* file. The *.bit* file will be named after the top-level file in your project hierarchy.

6) Copy the bit file to the location of *xc3sprog.exe*.

7) Issue the following command

```
xc3sprog -c ft232h -I top.bit
```

xc3sprog	Call the programmer
-c ft232h	Select the FT232H chip (the chip on the UM232 cable)
-I top.bit	Program the bit file (top.bit) to the memory chip

Table 4 - Command to send bit file to non-volitile memory

8) You should see the response shown in Figure 8. If unsuccessful, check solder joints, cable, and verify connections.

```
:~/Desktop/xc3sprog/xc3sprog$ sudo ./xc3sprog -c ft232h -I bst_bit.bi
t
XC3SPROG (c) 2004-2011 xc3sprog project $Rev: 774 $ OS: Linux
Free software: If you contribute nothing, expect nothing!
Feedback on success/failure/enhancement requests:
        http://sourceforge.net/mail/?group_id=170565
Check Sourceforge for updates:
        http://sourceforge.net/projects/xc3sprog/develop

Using Libftdi,
JEDEC: 20 20 0x14 0x10
Found Numonyx M25P Device, Device ID 0x2014
256 bytes/page, 4096 pages = 1048576 bytes total
Verify: Success!
```

Figure 8 - Transmitting bit file for non-volatile programming

9) Issue the following command to reset the Spartan FPGA. When the Spartan resets, it checks for memory devices and loads any configuration data it finds. Because the memory was just written with the desired bit file, the FPGA will load that bit file into its volatile memory.

```
xc3sprog –c ft232h –R
```

xc3sprog	Call the programmer
-c ft232h	Select the FT232H chip (the chip on the UM232 cable)
-R	Reset the FPGA

Table 5 - Command to reset the FPGA

2.5 Troubleshooting

WARNING: Do not connect UM232H-B power line and micro USB power at the same time!

- Check that your TDI, TDO, TMS, and TCK pins are connected to the board correctly.
- Verify that your board is connected to power (5V) and ground.
- Verify that there are no cold or missing solder joints or bridges on the board. Use a microscope to check smaller joints.
- Verify that the UM232 USB to JTAG converter has an amber light.
- Reduce your programmer frequency by appending `-J slowclock` to the end of the `xc3sprog` command. This will reduce the clock frequency of the JTAG circuitry.

Figure 9 indicates a failure to program the onboard flash memory. This is generally caused by a hardware failure. Refer to the items listed above to resolve many common hardware issues.

```
          :~/Desktop/xc3sprog/xc3sprog$ sudo ./xc3sprog -c ft232h -I bst_bit.bi
t
XC3SPROG (c) 2004-2011 xc3sprog project $Rev: 774 $ OS: Linux
Free software: If you contribute nothing, expect nothing!
Feedback on success/failure/enhancement requests:
        http://sourceforge.net/mail/?group_id=170565
Check Sourceforge for updates:
        http://sourceforge.net/projects/xc3sprog/develop

Using Libftdi,
JEDEC: 20 20 0x14 0x10
Found Numonyx M25P Device, Device ID 0x2014
256 bytes/page, 4096 pages = 1048576 bytes total

Verify failed  at flash_page     20
read:00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000
file:00000000000000004000000000000000000000000000000000000000000000000000000000000000
00000040000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000

Verify failed  at flash_page     23
read:00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000
file:00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000100000000000000000000000000000000000000000
00000000000000000000000000000000
```

Figure 9 - FPGA programming failed due to hardware failure

3.0 References

3.1 Supporting Documents

- Xilinx JTAG Programming Cables reference: http://www.xilinx.com/support/documentation/user_guides/xtp029.pdf

- FTDI UM232H-B USB to Multipurpose UART/FIFO datasheet: http://www.digikey.com/product-detail/en/UM232H-B/768-1159-ND/3770836

3.2 GPIO Pinout Descriptions

Group A

Header Pin No.	Pin Description	Spartan 6 (XC6SLX9 TQG144) Pin No.
1	VCCIO	N/A
2	**GND**	**N/A**
3		35
4		34
5		33
6		32
7		30
8		29
9		27
10		26
11		24
12		23
13		22
14		21
15		17
16		16
17		15
18		14
19		12
20		11
21		10
22		9
23		8
24		7
25		6
26		5
27		2
28		1
29		142
30		141
31		140
32		139
33		138
34		137
35		134
36		133
37		132
38		131
39	VCCIO	N/A
40	**GND**	**N/A**

Group B

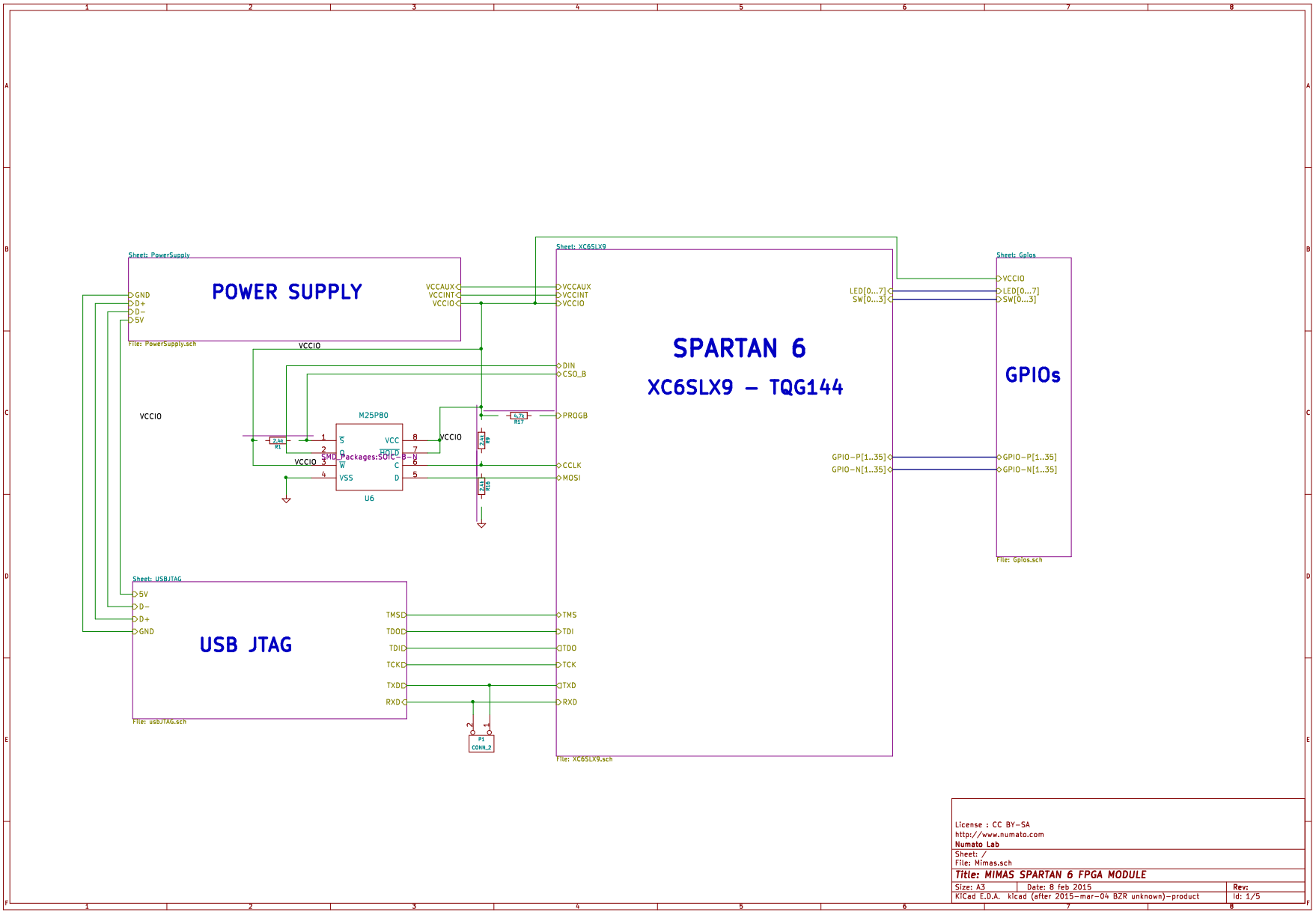
Header Pin No.	Pin Description	Spartan 6 (XC6SLX9 TQG144) Pin No.
1	**GND**	**N/A**
2	VCCIO	N/A
3		43
4		44
5		45
6		46
7		47
8		48
9		50
10		51
11		55
12		56
13		74
14		75
15		78
16		79
17		80
18		81
19	**GND**	**N/A**
20	**GND**	**N/A**
21		82
22		83
23		84
24		85
25		87
26		88
27		92
28		93
29		94
30		95
31		97
32		98
33		99
34		100
35		101
36		102
37		104
38		105
39	**GND**	**N/A**
40	VCCIO	N/A

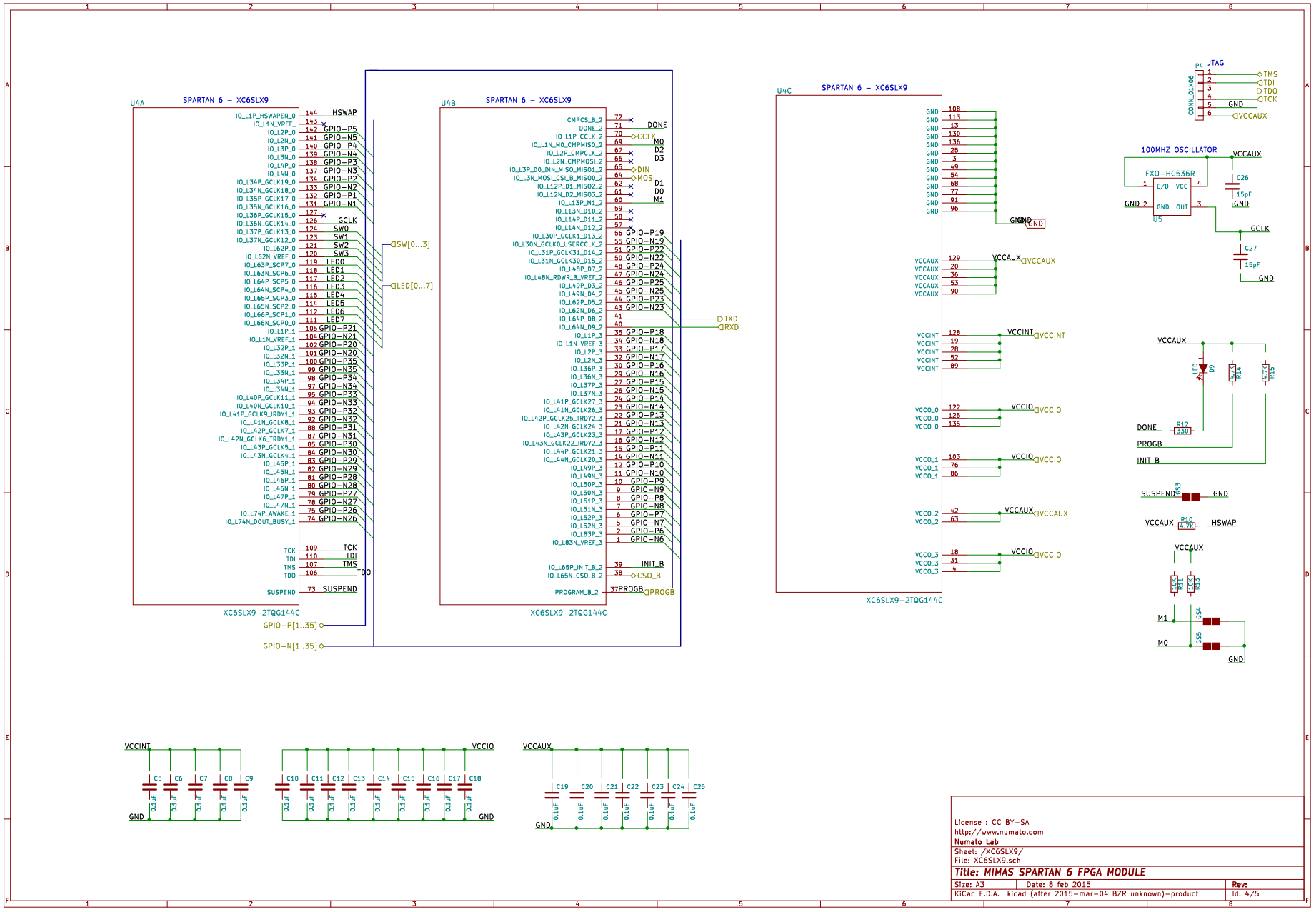
3.2 Bill of Materials (BOM)

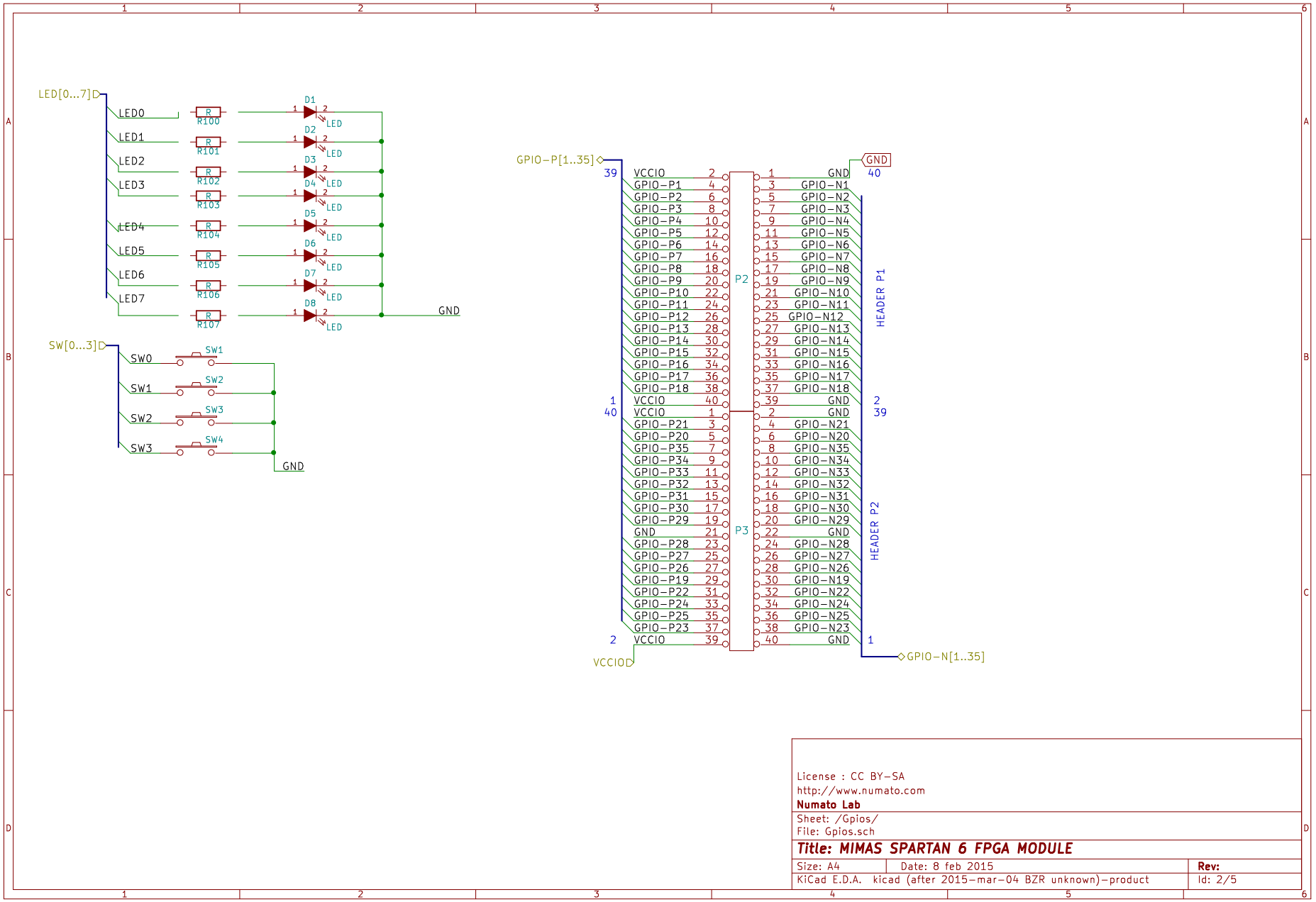
NOTE: *DNP = Do Not Populate*

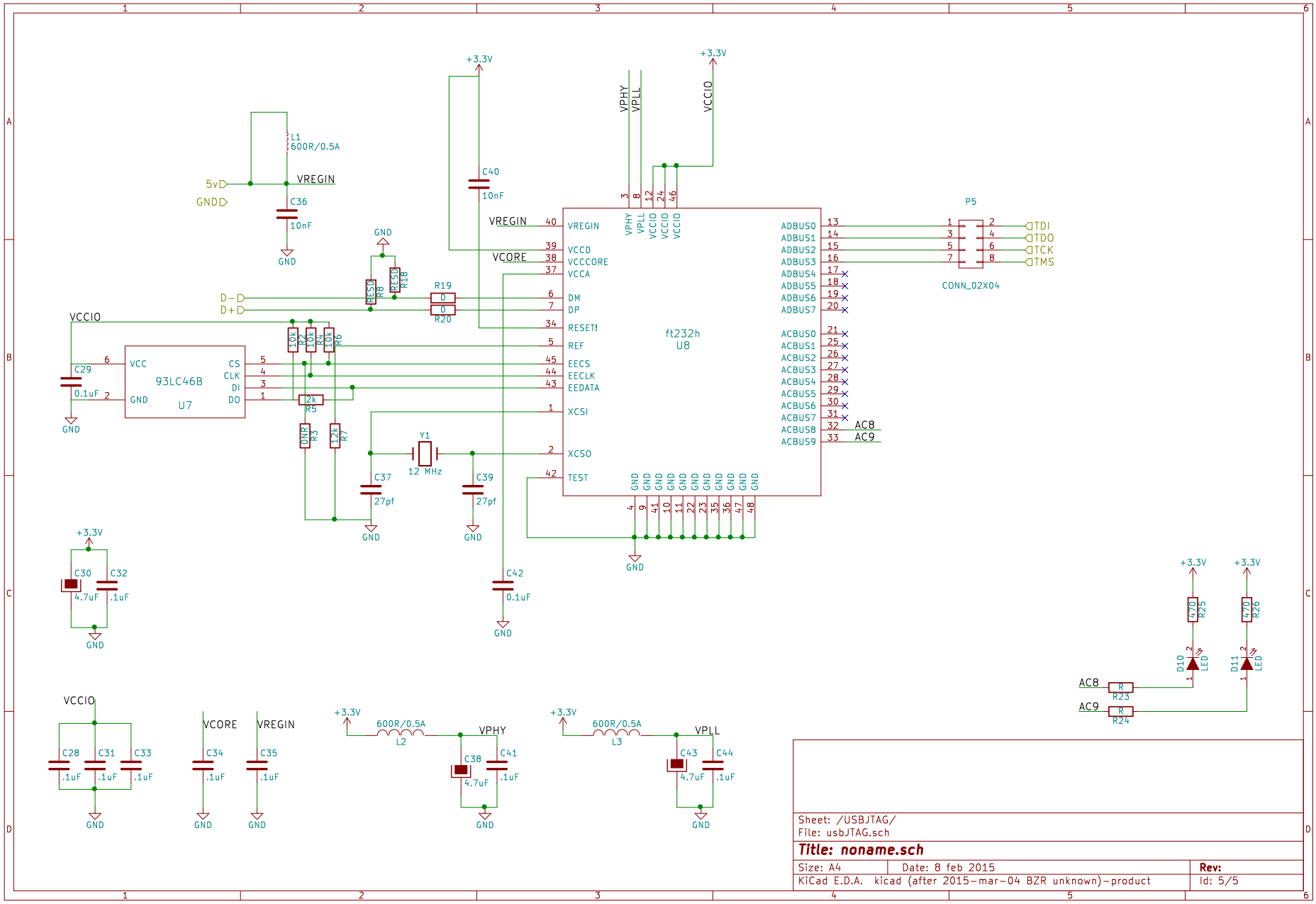
PCB Reference	Qty.	Value	Footprint	Notes
C1	1	47uF	Capacitors_SMD:C_1210_HandSoldering	
C26-27	2	15pF	Capacitors_SMD:C_0603_HandSoldering	
C2-C4	3	10uF	Capacitors_SMD:C_0805_HandSoldering	
C31-32	2	27pF	Capacitors_SMD:C_0603_HandSoldering	DNP (USB JTAG hardware)
C5-C25, C28-C30, C33-C35	27	0.1uF	Capacitors_SMD:C_0603_HandSoldering	Can use 0.001uF
D1-D9	11	LED	LEDs:LED-0805	
GS3, GS5		GS2	N/A	DNP
GS4		0 Ω	N/A	Short
P1	1	CONN_2	Pin_Headers:Pin_Header_Straight_1x02	
P100	1	GND	Pin_Headers:Pin_Header_Straight_1x12	
P101	1	VCC AUX +3.3 V	Pin_Headers:Pin_Header_Straight_1x12	
P23		CONN_4	Pin_Headers:Pin_Header_Straight_1x04	
P2-P3	2	CONN_20X2	Pin_Headers:Pin_Header_Straight_2x20	
P4	1	CONN_01X06	Pin_Headers:Pin_Header_Straight_1x06	
R1, R9, R16	3	2.4 kΩ	Resistors_SMD:R_0603_HandSoldering	
R10, R14, R15, R17	4	4.7 kΩ	Resistors_SMD:R_0603_HandSoldering	
R19-R21	3	68 Ω	Resistors_SMD:R_0603_HandSoldering	DNP (USB JTAG hardware)
R2, R11, R13	3	10 kΩ	Resistors_SMD:R_0603_HandSoldering	DNP R2 (USB JTAG hardware)
R3,R4	2	27 Ω	Resistors_SMD:R_0603_HandSoldering	DNP (USB JTAG hardware)
R5	1	2.2 kΩ	Resistors_SMD:R_0603_HandSoldering	DNP (USB JTAG hardware)
R6	1	470 Ω	Resistors_SMD:R_0603_HandSoldering	DNP (USB JTAG hardware)
R7	1	1 kΩ	Resistors_SMD:R_0603_HandSoldering	DNP (USB JTAG hardware)
R8, R12, R18, R100-R107	11	330 Ω	Resistors_SMD:R_0603_HandSoldering	
SW1-SW4	4	SW_PUSH	kevin:SW_PUSH_FSMJ	
U1	1	AMS1117 1.2	SOT223	
U2, U3	2	AMS1117 3.3	SOT223	
U4	1	XC6SLX9-2TQG144C	TQFP144	
U5	1	FXO-HC536R	kevin:crystal_FXO-HC53	
U6	1	M25P80	SMD_Packages:SOIC-8-N	Consider Winbond W25Q
U7	1	93LC46B	SMD_Packages:SOIC-8-N	DNP (USB JTAG hardware)
U9	1	FT2232D	Housings_QFP:LQFP-48_7x7mm_Pitch0.5mm	DNP (USB JTAG hardware)
X1	1	USBMICRO	kevin:MICRO-B_USB	
X2	1	CRYSTAL	Crystals:Crystal_HC49-SD_SMD	DNP (USB JTAG hardware)

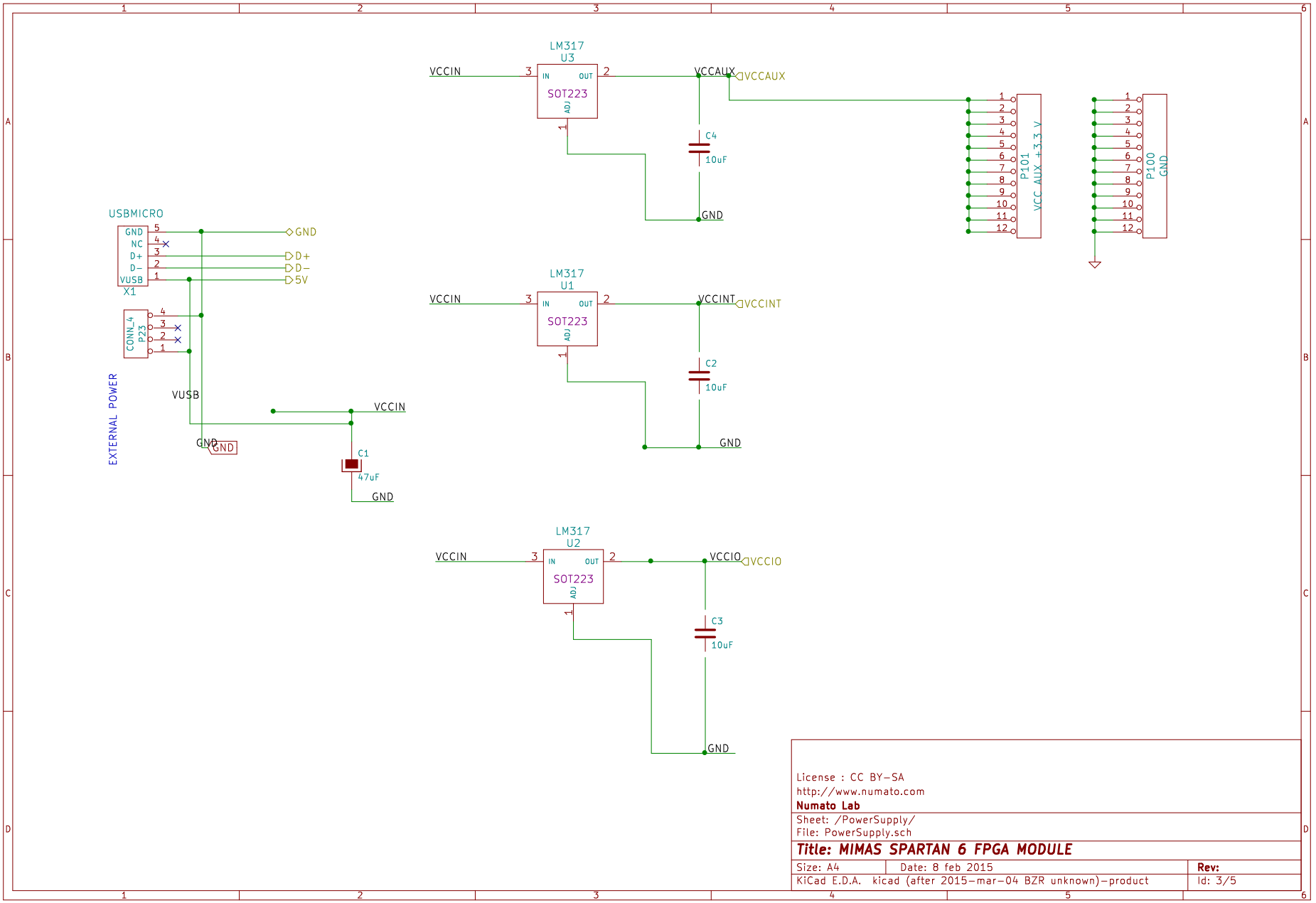
3.3 Schematics



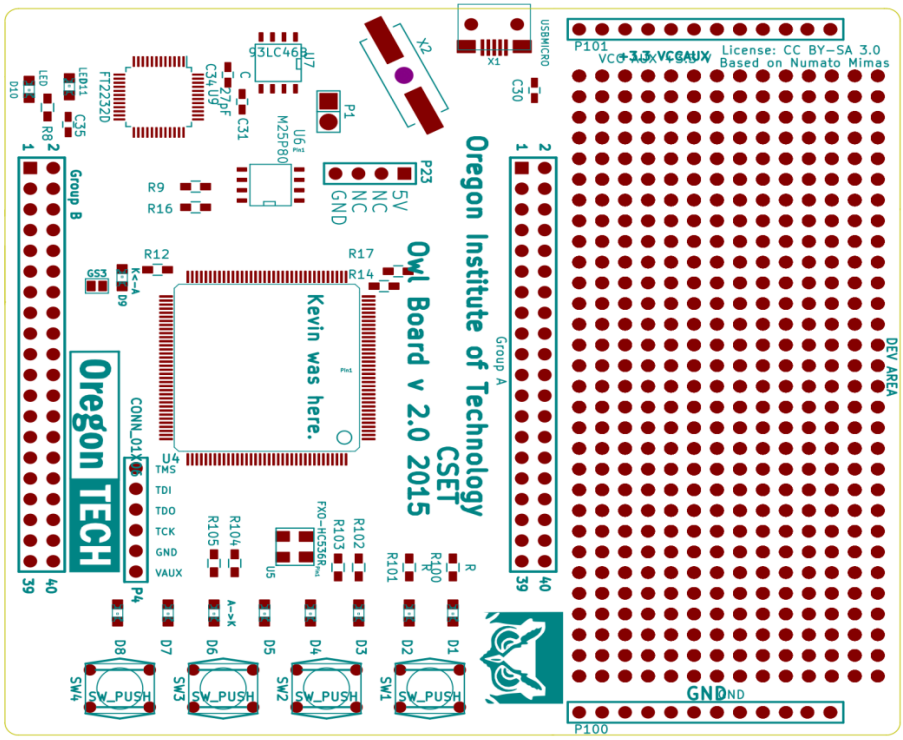

U4A
SPARTAN 6 – XC6SLX9
XC6SLX9-2TQG144C
U4B
SPARTAN 6 – XC6SLX9
XC6SLX9-2TQG144C
U4C
SPARTAN 6 – XC6SLX9
XC6SLX9-2TQG144C
P4 JTAG
TMS
TDI
TDO
TCK
GND
VCCAUX
100MHZ OSCILLATOR
FXO-HC536R
U5
C26
15pF
C27
15pF
GCLK
SW[0...3]
LED[0...7]
TXD
RXD
GPIO-P[1..35]
GPIO-N[1..35]
DONE
INIT_B
PROGB
SUSPEND
HSWAP
M1
M0
VCCINT
VCCIO
VCCAUX
C5 C6 C7 C8 C9
C10 C11 C12 C13 C14 C15 C16 C17 C18
C19 C20 C21 C22 C23 C24 C25
0.1uF
License : CC BY-SA
http://www.numato.com
Numato Lab
Sheet: /XC6SLX9/
File: XC6SLX9.sch
Title: MIMAS SPARTAN 6 FPGA MODULE
Size: A3
Date: 8 feb 2015
Rev:
KiCad E.D.A. kicad (after 2015-mar-04 BZR unknown)-product
Id: 4/5

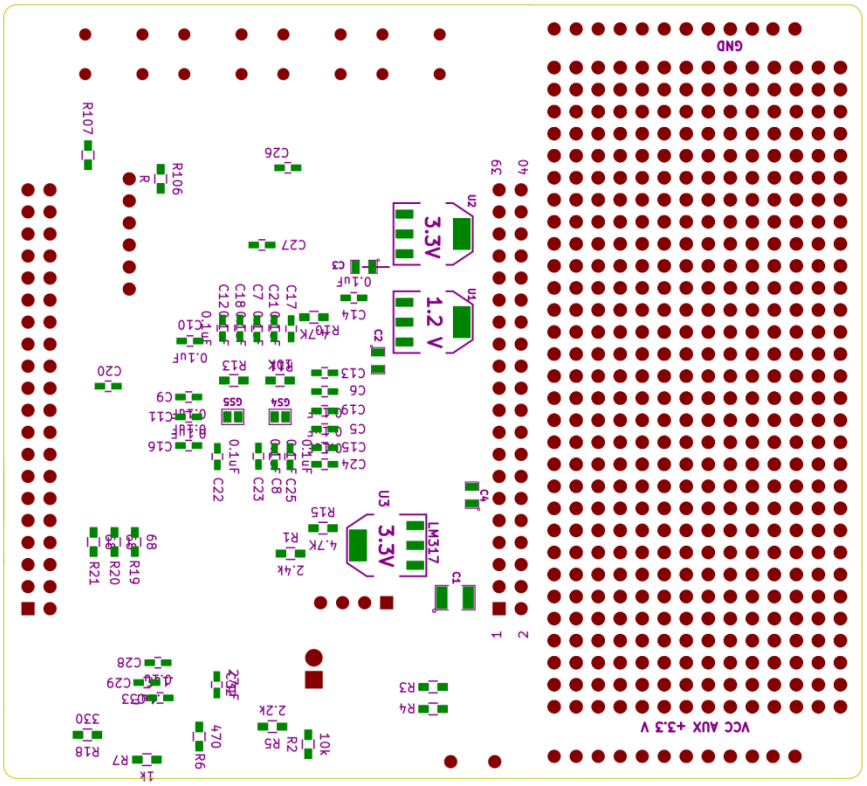

LED[0...7]
LED0
LED1
LED2
LED3
LED4
LED5
LED6
LED7
R100
R101
R102
R103
R104
R105
R106
R107
D1
D2
D3
D4
D5
D6
D7
D8
LED
GND
SW[0...3]
SW0
SW1
SW2
SW3
SW4
GND
GPIO-P[1..35]
VCCIO
GPIO-P1
GPIO-P2
GPIO-P3
GPIO-P4
GPIO-P5
GPIO-P6
GPIO-P7
GPIO-P8
GPIO-P9
GPIO-P10
GPIO-P11
GPIO-P12
GPIO-P13
GPIO-P14
GPIO-P15
GPIO-P16
GPIO-P17
GPIO-P18
GPIO-P21
GPIO-P20
GPIO-P35
GPIO-P34
GPIO-P33
GPIO-P32
GPIO-P31
GPIO-P30
GPIO-P29
GPIO-P28
GPIO-P27
GPIO-P26
GPIO-P19
GPIO-P22
GPIO-P24
GPIO-P25
GPIO-P23
P2
P3
GND
GPIO-N1
GPIO-N2
GPIO-N3
GPIO-N4
GPIO-N5
GPIO-N6
GPIO-N7
GPIO-N8
GPIO-N9
GPIO-N10
GPIO-N11
GPIO-N12
GPIO-N13
GPIO-N14
GPIO-N15
GPIO-N16
GPIO-N17
GPIO-N18
GPIO-N21
GPIO-N20
GPIO-N35
GPIO-N34
GPIO-N33
GPIO-N32
GPIO-N31
GPIO-N30
GPIO-N29
GPIO-N28
GPIO-N27
GPIO-N26
GPIO-N19
GPIO-N22
GPIO-N24
GPIO-N25
GPIO-N23
HEADER P1
HEADER P2
GPIO-N[1..35]
License : CC BY-SA
http://www.numato.com
Numato Lab
Sheet: /Gpios/
File: Gpios.sch
Title: MIMAS SPARTAN 6 FPGA MODULE
Size: A4
Date: 8 feb 2015
Rev:
KiCad E.D.A. kicad (after 2015-mar-04 BZR unknown)-product
Id: 2/5


+3.3V
L1
600R/0.5A
5v
GND
VREGIN
C36
10nF
C40
10nF
VPHY
VPLL
VCCIO
VREGIN
VCORE
GND
RESD
R8
R18
R19
0
R20
D−
D+
VCCIO
10k
R2
10k
R4
10k
R6
C29
0.1uF
93LC46B
U7
VCC
GND
CS
CLK
DI
DO
2k
R5
0R
R3
12k
R7
Y1
12 MHz
C37
27pf
C39
27pf
C42
0.1uF
ft232h
U8
VREGIN
VCCD
VCCCORE
VCCA
DM
DP
RESET!
REF
EECS
EECLK
EEDATA
XCSI
XCSO
TEST
ADBUS0
ADBUS1
ADBUS2
ADBUS3
ADBUS4
ADBUS5
ADBUS6
ADBUS7
ACBUS0
ACBUS1
ACBUS2
ACBUS3
ACBUS4
ACBUS5
ACBUS6
ACBUS7
ACBUS8
ACBUS9
AC8
AC9
P5
CONN_02X04
TDI
TDO
TCK
TMS
C30
4.7uF
C32
.1uF
C28
C31
C33
C34
C35
600R/0.5A
L2
L3
C38
C41
C43
C44
R25
470
R26
D10
D11
LED
R23
R24
Sheet: /USBJTAG/
File: usbJTAG.sch
Title: noname.sch
Size: A4
Date: 8 feb 2015
Rev:
KiCad E.D.A. kicad (after 2015−mar−04 BZR unknown)−product
Id: 5/5


LM317
U3
SOT223
VCCIN
VCCAUX
VCCAUX
C4
10uF
GND
P101
VCC_AUX +3.3_V
P100
GND
LM317
U1
SOT223
VCCIN
VCCINT
VCCINT
C2
10uF
GND
LM317
U2
SOT223
VCCIN
VCCIO
VCCIO
C3
10uF
GND
USBMICRO
GND
NC
D+
D−
VUSB
X1
GND
D+
D−
5V
CONN_4
P23
EXTERNAL POWER
VUSB
GND
VCCIN
C1
47uF
GND
License : CC BY−SA
http://www.numato.com
Numato Lab
Sheet: /PowerSupply/
File: PowerSupply.sch
Title: MIMAS SPARTAN 6 FPGA MODULE
Size: A4
Date: 8 feb 2015
Rev:
KiCad E.D.A. kicad (after 2015−mar−04 BZR unknown)−product
Id: 3/5

3.4 PCB Layout





4.0 Document History

Document Revision	Date	Author
v. 0.1 - Initial write-up	June 1, 2015	Alexander Hogen
v. 0.2 - Programming instructions, BOM	July 19, 2015	Alexander Hogen
v. 0.8 - Schematics, pinout tables, images	July 21, 2015	Alexander Hogen
v. 3.2 – Revise BOM, add additional warnings	January 29, 2015	Kevin Pintong

5.0 Important Notes

- Xilinx, Spartan, ISE, iMPACT and any other related tools and trademarks are property of Xilinx Incorporated (www.xilinx.com)
- FTDI, UM232, and any other related tools and trademarks are property of Future Technology Devices International Ltd. (http://www.ftdichip.com/)