startKIT Hardware Manual

IN THIS DOCUMENT

startKIT is a low-cost development board for the configurable xCORE multicore microcontroller products from XMOS. It's easy to use and provides lots of advanced features on a small, extremely low cost platform.

xCORE lets you software-configure the interfaces that you need for your system; so with startKIT you can configure the board to your match your exact requirements. Its 500MIPS xCORE multicore microcontroller has eight 32bit logical cores that perform deterministically, making startKIT an ideal platform for functions ranging from robotics and motion control to networking and digital audio.

startKIT also connects easily to your Raspberry Pi, allowing you to add real-time I/O and communication features to this popular computing platform, and to try out advanced applications for xCORE.

Publication Date: 2013/10/17 REV A

1 Features

A block diagram of the startKIT is shown below:

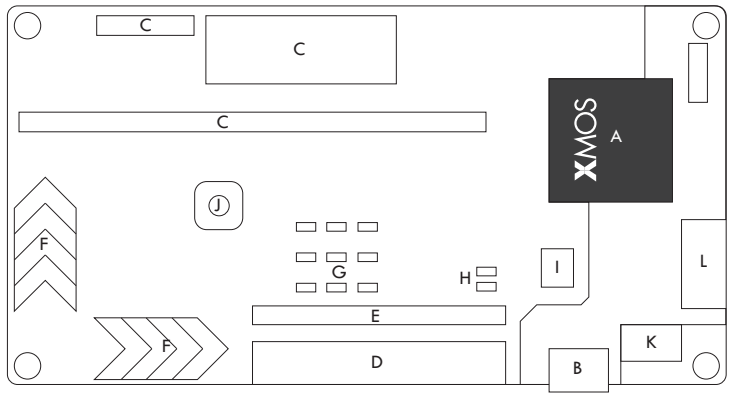

Figure 1: startKIT block diagram

It includes the following features:

- A: xCORE-Analog A8 device with integrated debugger
- B: Micro USB connector for debugger/JTAG
- C: PCIe slot for sliceCARD or 1x24 GPIO header
- D: 2x13 header for GPIO and compatible with Raspberry Pi
- E: 1x13 header providing two XMOS Links
- F: Two four-zone cap sense areas
- G: 3x3 grid green LEDs
- H: Two green LEDs
- I: SPI Flash
- J: Push-button switch
- K: 3x2 analog input header
- L: 24MHz Oscillator

Some of the features share the xCORE port resources as shown below:

2 xCORE-Analog A8-DEV Device

startKIT provides a single programmable 500 MIPS xCORE tile with eight logical cores. All the digital I/O have been brought out to pins providing many combinations of peripherals to be integrated with the startKIT board. The A8-DEV board also includes an integrated debugger.

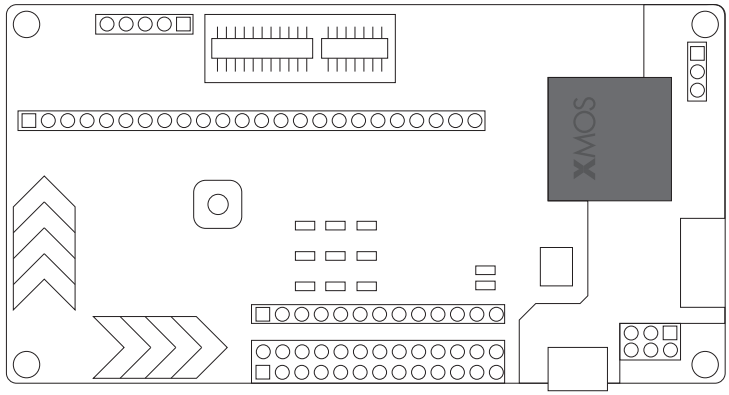

Figure 2: xCORE-Analog device and integrated debugger

2.1 Integrated debugger

The integrated debugger and associated components are positioned at one end of the board. The debugger is accessed by a micro-USB connector connected to the host PC, allowing the xTIMEcomposer tools to interrogate the application running on the device using the XMOS debugger and the xSCOPE library which provides non-intrusive program instrumentation.

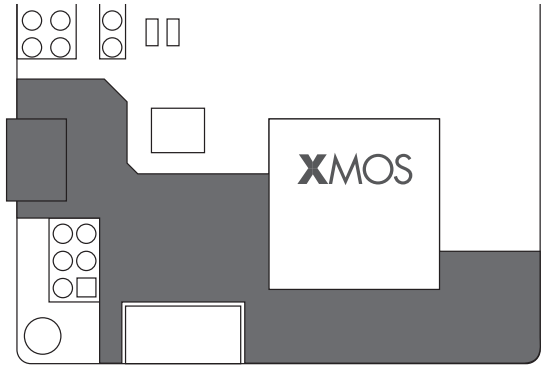

Figure 3: Integrated debugger

3 PCIe connector and GPIO header (J7)

The pins of the PCIe connector and the 1x24 GPIO header are mapped to twelve 1-bit ports and three 4-bit ports. The connector and GPIO header are mutually exclusive. The PCIe connector is suitable for XMOS sliceCARDs such as audio, Ethernet, IS-BUS.

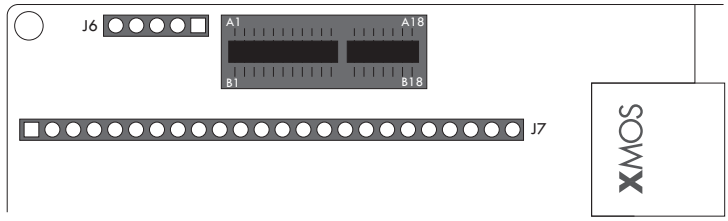

Figure 4: PCIe Connector

The xCORE ports are mapped to the PCIe connector pins as shown in Figure 5:

Port	Pin	PCIe (top)	PCIe (bottom)	Pin	Port
	NC	B1	A1	NC	
P1F0	X0D13	B2	A2	5V	
	GND	B3	A3	X0D12	P1E0
P1G0	X0D22	B4	A4	X0D23	P1H0
	3V3	B5	A5	GND	
P4C0	X0D14	B6	A6	X0D20	P4C2
P4C1	X0D15	B7	A7	X0D21	P4C3
	GND	B8	A8	X0D25	P1J0
P4D0	X0D16	B9	A9	X0D18	P4D2
P1K0	X0D34	B10	A10	GND	
P4D1	X0D17	B11	A11	X0D19	P4D3
P1M0	X0D36	B12	A12	X0D32	P4E2
P1N0	X0D37	B13	A13	X0D33	P4E3
P4D3	CLK(DEBUGGER)	B14	A14	GND	
P1I0	X0D24	B15	A15	X0D35	P1L0
	GND	B16	A16	RST_N(DEBUGGER)	
P1O0	X0D38	B17	A17	X0D26	P4E0
P1P0	X0D39	B18	A18	X0D27	P4E1

Figure 5: PCIe connector

The J6 header provides peripheral support for the PCIe connector as described in Figure 6

Pin	Support
CLK	25 MHz clock, signal generated by debugger
nRST	Reset for PCIe slot, signal generated by debugger
5V0	5V power supply
3V3	3V3 power supply
GND	Ground

Figure 6: J6 header

The GPIO header (J7) provides 24 user configurable GPIO if the PCIe slot is not used - see Figure 7.

Pin	Port	GPIO
P1F	X0D13	1
P1H	X0D23	2
P1G	X0D22	3
P1E	X0D12	4
P4C0	X0D14	5
P4C2	X0D20	6
P4C1	X0D15	7
P4C3	X0D21	8
P4D0	X0D16	9
P1J	X0D25	10
P1K	X0D34	11
P4D2	X0D18	12
P4D1	X0D17	13
P4D3	X0D19	14
P1M	X0D36	15
P4E2	X0D32	16
P1N	X0D37	17
P4E3	X0D33	18
P1L	X0D35	19
P1I	X0D24	20
P1O	X0D38	21
P4E0	X0D26	22
P1P	X0D39	23
P4E1	X0D27	24

Figure 7: J7 Header GPIO

4 Raspberry Pi compatible header and GPIO (J3)

The 2x13 pin 0.1" header is connected to a combination of 1-bit ports and the 32-bit port. It is compatible with a Raspberry Pi connection through the SPI device. Alternatively the header can be used for user configurable GPIO.

The position of the header on the startKIT board is shown below:

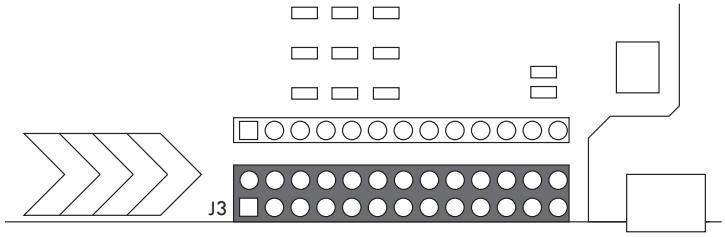

Figure 8: Raspberry Pi compatible header and GPIO

The xCORE ports are connected to the header as shown in Figure 7:

Port	Pin	Header IO		Pin	Port
	NC	1	2	NC	
P32A0	X0D49	3	4	NC	
P32A19	X0D70	5	6	GND	
P32A18	X0D69	7	8	X0D68	P32A17
	GND	9	10	X0D63	P32A12
P32D10	X0D61	11	12	X0D62	P32D11
P32A9	X0D58	13	14	GND	
P32D8	X0D57	15	16	X0D56	P32A7
	NC	17	18	NC	
P1A0	X0D0	19	20	GND	
P1D0	X0D11	21	22	NC	
P1C0	X0D10	23	24	X0D51	P32A2
	GND	25	26	X0D50	P32A1

Figure 9: GPIO (J3)

Notes:

- The compatible Raspberry Pi connections are shown on the back of the startKIT board.
- If you use the Raspberry Pi header the LEDs and push button switch are not available. You can still use the links on the J8 header.

5 XMOS Links and GPIO header (J8)

startKIT has a 1x13 pin GPIO header that includes two 2-wire XMOS Links (Link C/D), which can be used for connecting multiple startKITs together. Alternatively the header can be used to provide an additional eight GPIO pins connected to the 32-bit port.

The position of the header on the startKIT board is shown below:

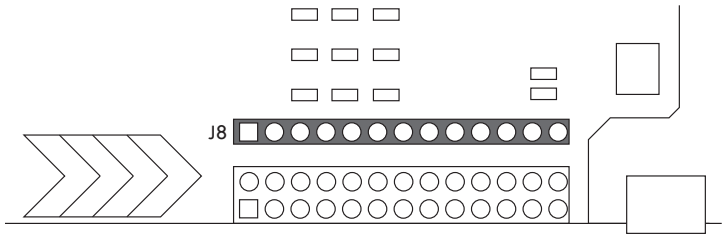

Figure 10: XMOS Links and GPIO header (J8)

Port	Pin	Position
	GND	1
	GND	2
P32A3	X0D52	3 - Link C: 1 Out
P32A4	X0D53	4 - Link C: 0 Out
P32A5	X0D54	5 - Link C: 0 In
P32A6	X0D55	6 - Link C: 1 In
P32A13	X0D64	7 - Link D: 1 Out
P32A14	X0D65	8 - Link D: 0 Out
P32A15	X0D66	9 - Link D: 0 In
P32A16	X0D67	10 - Link D: 1 In
	GND	11
	3V3	12
	5V0	13

Figure 11: J8 header ports

Note that the XMOS Links connections are shown on the back of the startKIT card.

6 Touch Sliders

The startKIT provides two four-zone cap sense areas. The layout of the touch areas is shown below:

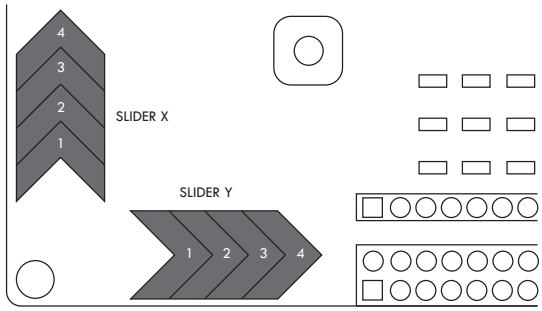

Figure 12: Cap sense touch zones

The touch areas are connected to pins driven by two 4-bit ports as described in Figure 13:

Pin	Port	Slider
P4A1	X0D4	X1
P4A2	X0D5	X2
P4A3	X0D6	X3
P4A4	X0D7	X4
P4B1	X0D2	Y1
P4B2	X0D3	Y2
P4B3	X0D8	Y3
P4B4	X0D9	Y4

Figure 13: Touch sliders

The touch areas must be polled to measure any touch.

7 User LEDs

startKIT provides nine green LEDs arranged in a 3x3 grid as shown below:

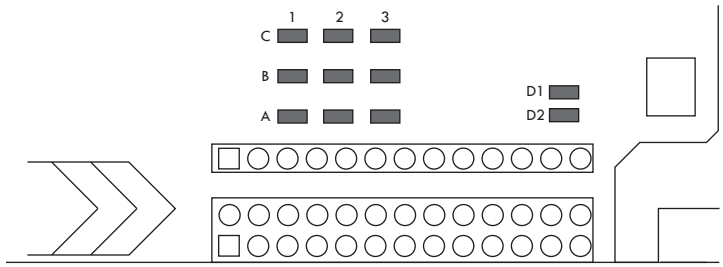

Figure 14: 3x3 grid LEDs

Each LED is connected to a different pin, all of which are mapped to bits on a 32-bit port as described in Figure 15:

Port	Pin	LED
P32A19	X0D70	A1
P32A18	X0D69	A2
P32A17	X0D68	A3
P32A12	X0D63	B1
P32A11	X0D62	B2
P32A10	X0D61	B3
P32A9	X0D58	C1
P32A8	X0D57	C2
P32A7	X0D56	C3

Figure 15: 3x3 grid LEDs

Two additional green LEDs are connected to pins driven by 1-bit ports as described in Figure 16:

Port	Pin	Processor
P1A	X0D0	LED-D1
P1D	X0D11	LED-D2

Figure 16: User LEDs

Notes

- The LEDs are not available if the J3 (Raspberry Pi) header is in use.
- If the LEDs/button are in use, you cannot use the J8 header.
- LED pins are active high.

8 SPI Flash

startKIT provides 256 Kbytes of Serial Peripheral Interface (SPI) FLASH memory, which is interfaced by the four 1-bit connections shown in Figure 17:

Figure 17: SPI Flash

Port	Pin	Processor
P1A0	X0D0	MISO
P1B0	X0D1	CS_N
P1C0	X0D10	M_CK
P1D0	X0D11	MOSI

The xTIMEcomposer tools include the xFLASH utility for programming compiled programs into the flash memory. startKIT designs may also access the FLASH memory at run-time by interfacing with the above pins.

9 Push-button switch

startKIT includes one push-button switch whose states can be samples at any time by software. The position of the switch is shown below.

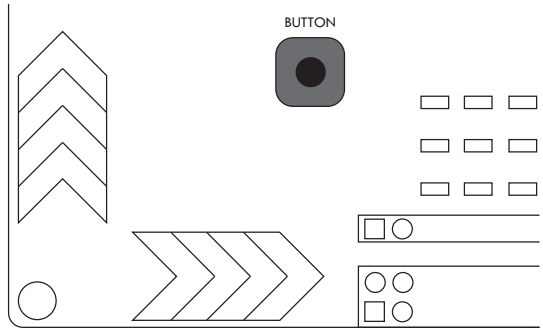

Figure 18: Push button switch

The switch is connected to a pin which is mapped to one bit of the 32-bit port as described in Figure 19:

Figure 19: Button

Port	Pin	Processor
P32A0	X0D49	BUTTON

Notes:

- The push-button switch pin is active low.
- The push-button switch is not available if the J3 (Raspberry Pi) header is in use.

10 Analog input header

startKIT provides support for analog device input. The location of the 2x3 input header is shown below:

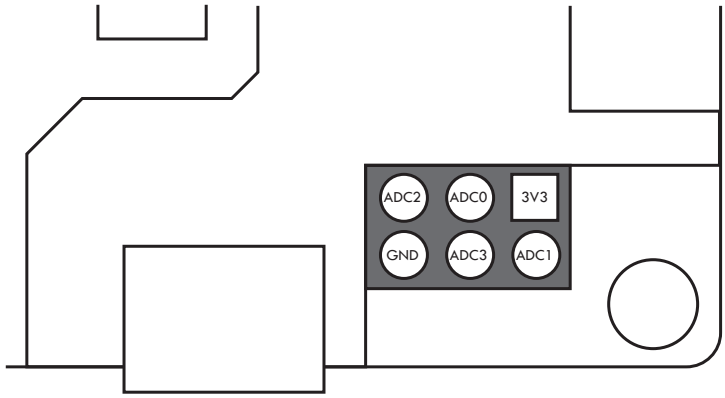

Figure 20: Analog input header

Analog inputs can be can be connected to the xCORE-Analog device using the four ADC pins as shown in Figure 21.

Pin	Header IO
3V3	1
ADC1	2
ADC0	3
ADC3	4
ADC2	5
GND	6

Figure 21: ADC input header

The analog input can be sampled using a 1-bit port as defined in Figure 22:

Port	Pin	Procesor
P1A0	X0D00	ADC_Sample

Figure 22: ADC sample

11 24MHz Crystal Oscillator

The startKIT board is clocked at 24MHz by a crystal oscillator. The A8-DEV xcORE tile is clocked at 500 MIPS, and the I/O ports are 100MHz. The debugger generates an additional 25MHz clock for the PCIe slot which can be accessed using the J6 header.

12 Power connector

startKIT requires a 5V power source input via the micro-USB cable.

The voltage is converted by the on-board regulator to the 1V and 3V3 supplies used by the components.

13 Dimensions

The startKIT dimensions are 94 x 50mm. The mounting holes are 2mm in diameter.

14 startKIT Portmap

The table below provides a full description of the port-pin mappings described throughout this document.

Pin	link	1-bit	4-bit	8-bit	32-bit	GPIO	SPI	USER IO	ANALOG
X0D00		$1A$				$J3/rPI^{19}$	$MISO$	LED^{D1}	ADC_SAMPLE
X0D01	A^4 out	$1B$					SS		
X0D02	A^3 out		$4A^0$	$8A^0$	$32A^{20}$			$TOUCH^{Y1}$	
X0D03	A^2 out		$4A^1$	$8A^1$	$32A^{21}$			$TOUCH^{Y2}$	
X0D04	A^1 out		$4B^0$	$8A^2$	$32A^{22}$			$TOUCH^{X1}$	
X0D05	A^0 out		$4B^1$	$8A^3$	$32A^{23}$			$TOUCH^{X2}$	
X0D06	A^0 in		$4B^2$	$8A^4$	$32A^{24}$			$TOUCH^{X3}$	
X0D07	A^1 in		$4B^3$	$8A^5$	$32A^{25}$			$TOUCH^{X4}$	
X0D08	A^2 in		$4A^2$	$8A^6$	$32A^{26}$			$TOUCH^{Y3}$	
X0D09	A^3 in		$4A^3$	$8A^7$	$32A^{27}$			$TOUCH^{Y4}$	
X0D10	A^4 in	$1C$				$J3/rPI^{23}$	CLK		
X0D11		$1D$				$J3/rPI^{21}$	$MOSI$	LED^{D2}	
X0D12		$1E$				$J7^4$ + $PCIe^{A3}$			
X0D13	B^4 out	$1F$				$J7^1$ + $PCIe^{B2}$			
X0D14	B^3 out		$4C^0$	$8B^0$	$32A^{28}$	$J7^5$ + $PCIe^{B6}$			
X0D15	B^2 out		$4C^1$	$8B^1$	$32A^{29}$	$J7^7$ + $PCIe^{B7}$			
X0D16	B^1 out		$4D^0$	$8B^2$		$J7^9$ + $PCIe^{B9}$			
X0D17	B^0 out		$4D^1$	$8B^3$		$J7^{13}$ + $PCIe^{B11}$			
X0D18	B^0 in		$4D^2$	$8B^4$		$J7^{12}$ + $PCIe^{A9}$			
X0D19	B^1 in		$4D^3$	$8B^5$		$J7^{14}$ + $PCIe^{A11}$			
X0D20	B^2 in		$4C^2$	$8B^6$	$32A^{30}$	$J7^6$ + $PCIe^{A6}$			
X0D21	B^3 in		$4C^3$	$8B^7$	$32A^{31}$	$J7^8$ + $PCIe^{A7}$			
X0D22	B^4 in	$1G$				$J7^3$ + $PCIe^{B4}$			
X0D23		$1H$				$J7^2$ + $PCIe^{A4}$			
X0D24		$1I$				$J7^{20}$ + $PCIe^{B15}$			
X0D25		$1J$				$J7^{10}$ + $PCIe^{A8}$			
X0D26			$4E^0$			$J7^{22}$ + $PCIe^{A17}$			
X0D27			$4E^1$			$J7^{24}$ + $PCIe^{A18}$			
X0D32			$4E^2$			$J7^{16}$ + $PCIe^{A12}$			
X0D33			$4E^3$			$J7^{18}$ + $PCIe^{A13}$			
X0D34		$1K$				$J7^{11}$ + $PCIe^{B10}$			
X0D35		$1L$				$J7^{19}$ + $PCIe^{A15}$			
X0D36		$1M$				$J7^{15}$ + $PCIe^{B12}$			
X0D37		$1N$				$J7^{17}$ + $PCIe^{B13}$			
X0D38		$1O$				$J7^{21}$ + $PCIe^{B17}$			
X0D39		$1P$				$J7^{23}$ + $PCIe^{B18}$			
X0D49	C^4 out				$32A^0$	$J3/rPI^3$		$BUTTON$	
X0D50	C^3 out				$32A^1$	$J3/rPI^{26}$			
X0D51	C^2 out				$32A^2$	$J3/rPI^{24}$			
X0D52	C^1 out				$32A^3$	$J8^3$			
X0D53	C^0 out				$32A^4$	$J8^4$			
X0D54	C^0 in				$32A^5$	$J8^5$			
X0D55	C^1 in				$32A^6$	$J8^6$			
X0D56	C^2 in				$32A^7$	$J3/rPI^{16}$		$LED3x3^{C3}$	
X0D57	C^3 in				$32A^8$	$J3/rPI^{15}$		$LED3x3^{C2}$	
X0D58	C^4 in				$32A^9$	$J3/rPI^{13}$		$LED3x3^{C1}$	
X0D61	D^4 out				$32A^{10}$	$J3/rPI^{11}$		$LED3x3^{B3}$	
X0D62	D^3 out				$32A^{11}$	$J3/rPI^{12}$		$LED3x3^{B2}$	
X0D63	D^2 out				$32A^{12}$	$J3/rPI^{10}$		$LED3x3^{B1}$	
X0D64	D^1 out				$32A^{13}$	$J8^7$			
X0D65	D^0 out				$32A^{14}$	$J8^8$			
X0D66	D^0 in				$32A^{15}$	$J8^9$			
X0D67	D^1 in				$32A^{16}$	$J8^{10}$			
X0D68	D^2 in				$32A^{17}$	$J3/rPI^8$		$LED3x3^{A3}$	
X0D69	D^3 in				$32A^{18}$	$J3/rPI^7$		$LED3x3^{A2}$	
X0D70	D^4 in				$32A^{19}$	$J3/rPI^5$		$LED3x3^{A1}$	

Figure 23: startKIT Portmap

15 startKIT schematics

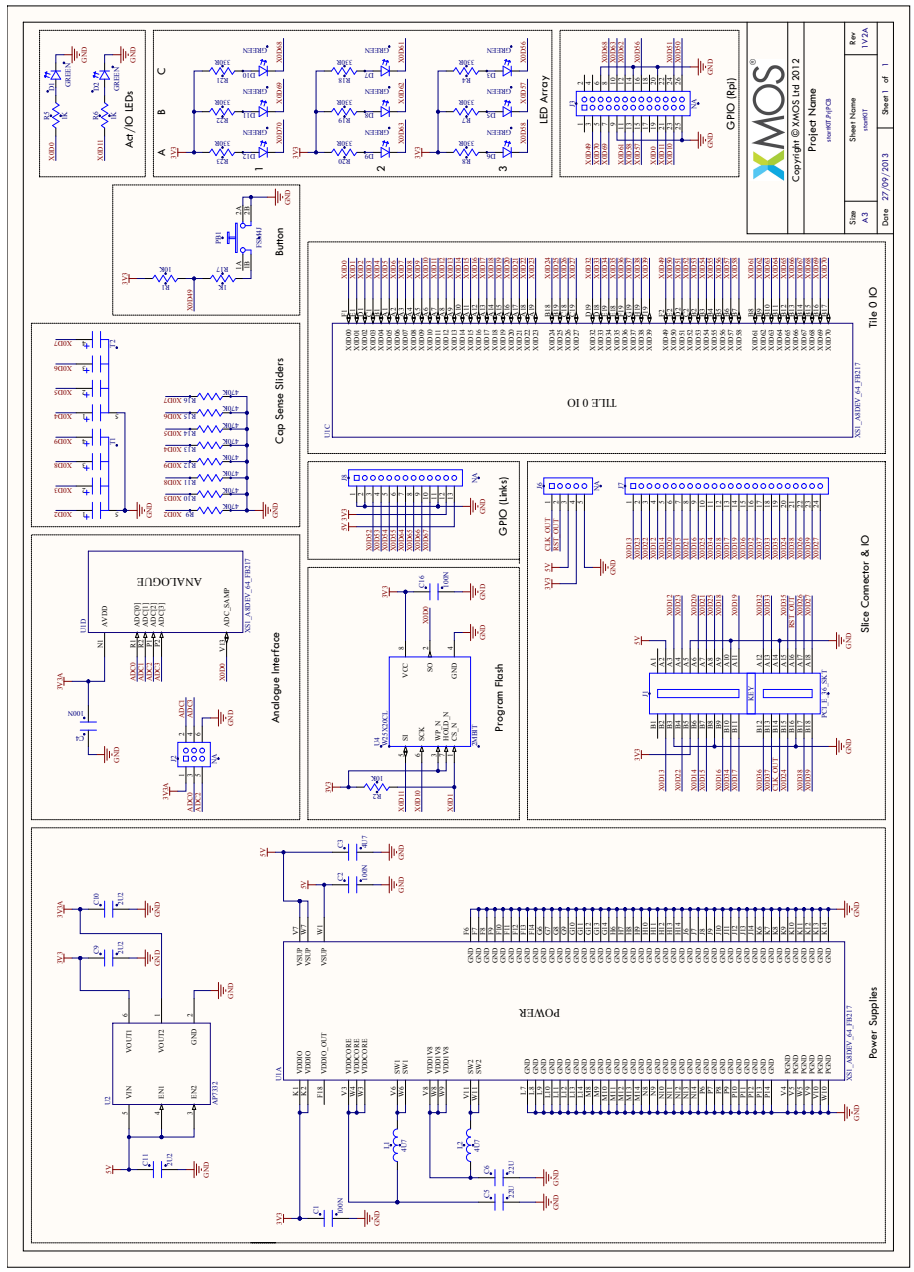

Figure 24: startKIT schematic

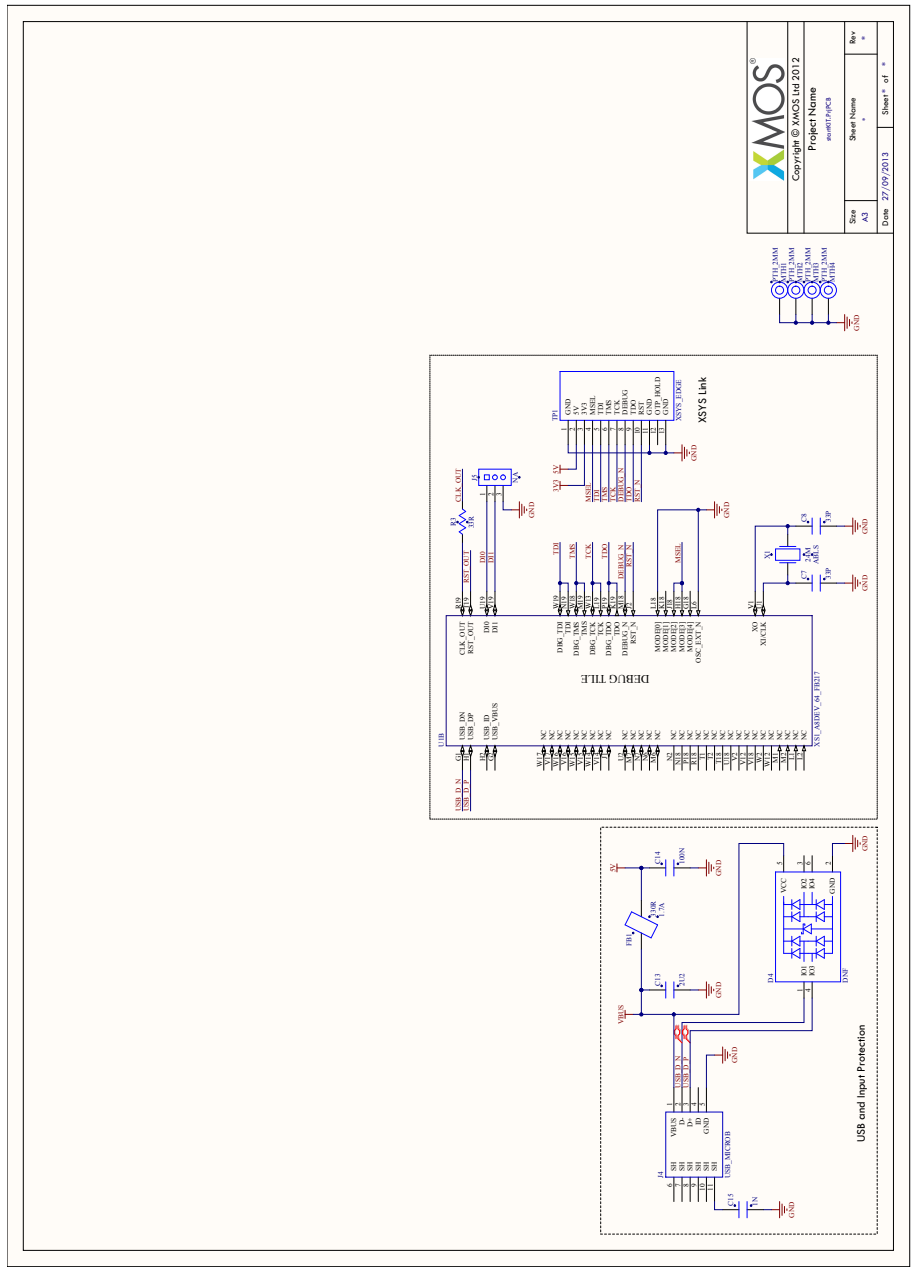

Figure 25: startKIT debugger schematic