Bitfire Development Kit
Hardware Reference Guide
Version 1.12

Bitfire

ARROW ENGINEERING

This page is intentionally left blank.

Revision History		
Version	**Date**	**Updates**
1.10	050530	Document public release.
1.11	051019	New HW release V1.11 Some pictures changed Description of new housekeeping MCU ATMega48. Added information on interface traffic indicators.
1.12	070212	New HW release V1.12 Power design changed to National Semiconductor New housekeeper concept / functionality Onboard debug selection mechanism changed Pictures updated

Arrow Engineering Sweden, Timmernabben, Sweden

1 Overview

Thank you for choosing an Arrow Engineering development platform, we hope and belive that it will meet your expectations. We appreciate your honest feedback and suggestions for improvements for future revisions and new developments, please mail any feedback you might may have to; turnkey@arrownordic.com. This will go directly to makers of this development package and is highly appreciated.

Bitfire provides the advanced embedded systems developer with an excellent low-cost ARM-7 based development kit. The kit also sports a low-cost FPGA, which opens possibilities for a wide variety of applications. This guide explains how to use the various functions onboard the Bitfire such as on chip debug tools, interfaces etc. Further you find in-depth explanation of the various functions in the design.

2 Getting Started

The Bitfire ships with a complete set of development and debugging tools. No additional software or hardware is required. It's all based around the GCC/GDB tool suite thus basically all platforms are supported for development. The suite has been tested under Windows© and Linux operating systems, but it should work on all platforms supported by GCC. You can optionally use external hardware such as professional J-Tag tools.

Note: For complete design reference, review the files '*Bitfire Schematic.pdf*' and '*Bitfire Layout.pdf*' files supplied on the CD.

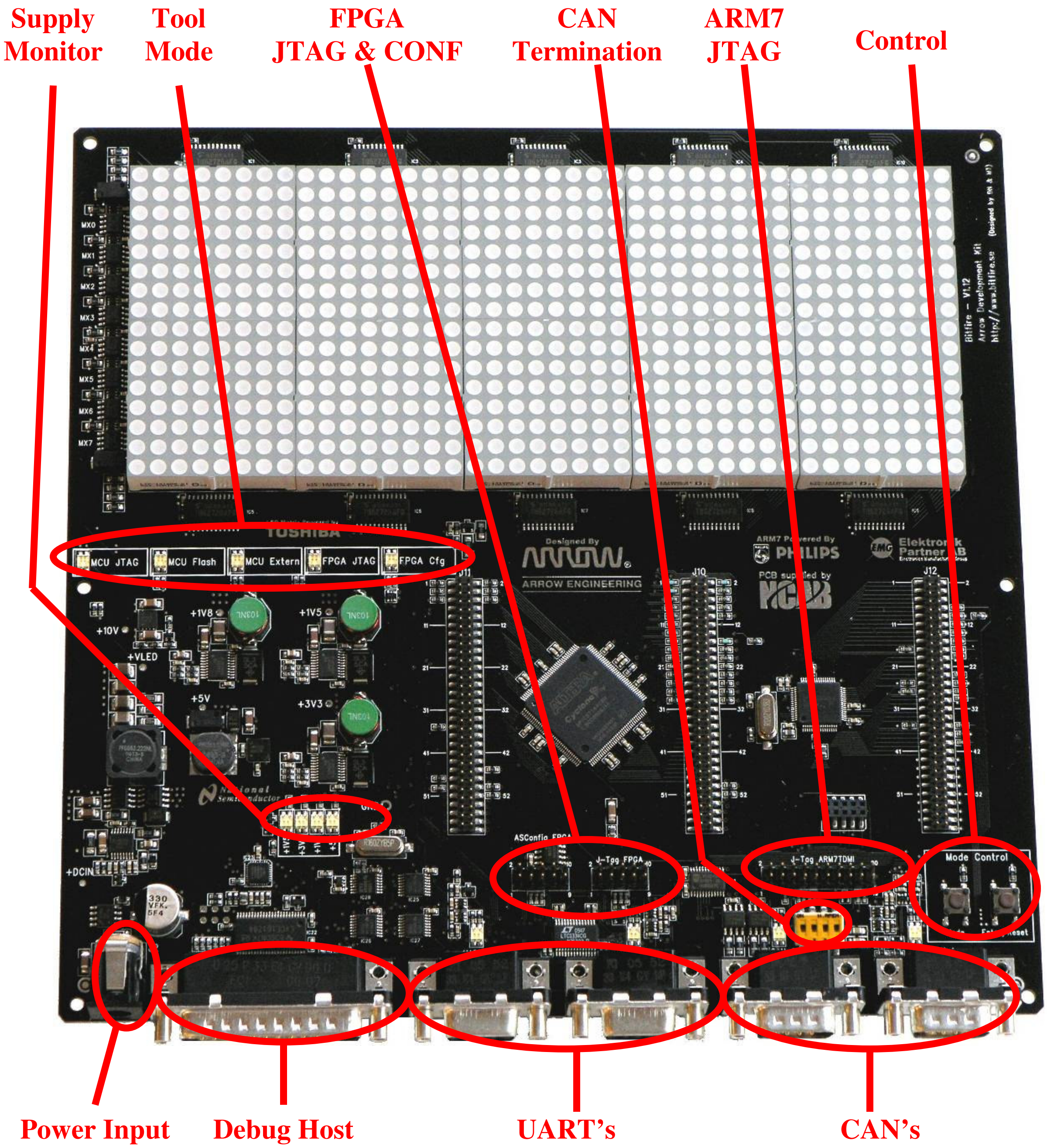
2.1 Unpacking the box

The Bitfire ships with necessary power supply and cables. You should be able to find these items in your Bitfire box:

- Bitfire development board
- Power supply with additional sockets and voltage plugs
- 1.8m 9 way F/M D-SUB cable.
- 1.8m 25 wat F/M D-SUB cable.
- Software CD-ROM

Caution: switching PC connectors while powered may damage your PC and/or Bitfire

2.2 Board overview



Power input

The input power required to fully charge the LED matrix is required to be in the range of 10-23VDC stabilized with at least 50W of power. The power supply shipped with Bitfire is preset to provide correct output voltage. The centerpin is the positive terminal. A lossless polarity protection prevents failure due to wrong polarity. The board is NOT over voltage protected. Additional programming plugs are provided between 5-24VDC to enable the user to reprogram the supply. Consult the power supply manual for details.

2.2.1 Debug Host

Connection to HOST PC for the onboard debugging tools, *see section 3* for more information on how to operate these tools.

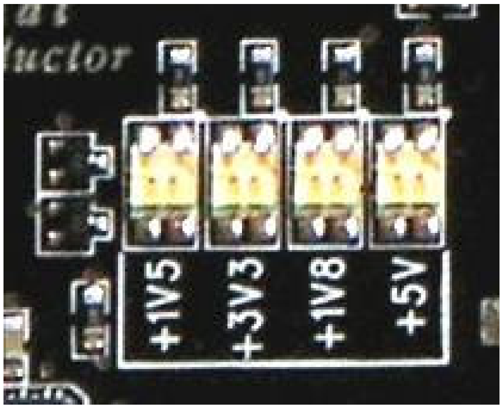
2.2.2 UART Connections

These ports are connected to the LPC2129's UART0 and UART1 serial interfaces; they have two modes of signal operation RS232 or RS485. Coming out of reset the FPGA defaults these to RS232, the same is true for a non configured FPGA. Consult the documentation for the Bitfire BSP in the *Bitfire Software Developers Guide* on how to reconfigure to RS485 signal mode by software.

2.2.3 CAN Connections

These connectors give access to the LPC2129 CAN channels. CAN-0 is connected to LPC2129 CAN-module 1 and CAN-1 to CAN-module 2. Transceivers are of high speed type and has a programmable termination. Pulling dipswitch position 1 & 2 in "on" state activates 136Ω termination for channel 1 and position 3 & 4 does the same on channel 2.

2.2.4 Supply monitor



This block of LED's gives a visual indication that all core and I/O voltages generated onboard are present and within tolerance.

2.2.5 ARM7 JTAG

This connector makes it possible to connect an external tool to access the ARM7 JTAG interface. For details on how to use this connector see *section 3* for more details

2.2.6 FPGA JTAG & CONF

These connector gives external access to the FPGA JTAG and configuration memory. For operation on how to use these connectors see *section 3* for more details.

2.2.7 Control

The on-board debug controller provides you functions to conveniently set the board into different modes by using the two operation buttons. For operation see *section 3* Onboard debug control. Tool mode indicates the current mode selected.

2.2.8 Traffic indicators

There are four traffic indicators each connected to each interface connectors. These will indicate interface traffic going in or out from the Bitfire. Green indicates packets going out trough the interface and red indicate incoming traffic.

3 Onboard debug control

The onboard debug tools both support the ARM target and the FPGA. Basically there are two HW tools integrated, but they support 3 different connection points. For the FPGA we have a ByteBlaster II compliant circuit and for the MCU target we support Macraigor™ Wiggler compliance.

3.1 How to use the debug control

Five leds located on the left hand indicate witch mode the onboard tools are in, you simply select the mode you want by pressing '*mode*' key on bottom right and activate you selection by pressing '*Enter/Reset*'. You can also select Idle mode, which disengage the debug controller from the chips debug ports and sets the debug controller into Idle state, this is done by pressing mode until non of the modes are indicated by the leds, this is the default mode upon start up. This will allow you to hook up external tools to the connectors described above in *section 2.2.6 and 2.2.7*. In active and idle mode you can reset the targets by pressing '*Enter/Reset'* all leds will flash red. The reset has slightly different functionality in different operating modes. See table below for the different modes. For more information on the debug and programming of FPGA and MCU please refer to the *Bitfire Software Developers Guide*.

Mode	**Description**	**Pressing reset will**
MCU J-Tag	ARM7 wiggler debugger engaged	Reset MCU
MCU Flash	Forcing bootloader on MCU if avalible	Exit bootloader and reset MCU
FPGA Cfg	FPGA byteblaster for configuration	Reset FPGA
FPGA J-Tag	FPGA byteblaster for debugging engaged	Reset FPGA
MCU Extern	ARM7 external tool	Reset MCU
IDLE	IDLE, onboard debugger disengaged (no leds lit)	Reset MCU & FPGA

3.1.1 More on "MCU Extern" mode

The MCU external mode described above will activate a pulldown on RTCK pin. During reset this pin has a very special purpose on the LPC2129, if sampled low this will automatically turn on the JTAG port on the device. This mode is only used if the external tool does not have built in support for driving this pin. The reason for having this mechanism is that the JTAG pins are multiplexed with GPIO pins on the LPC2129.

3.1.2 More on "MCU Flash"

In the boot loader mode the LPC2129 starts an internal boot loader that can be used to program the onboard flash using the Philips LPC Flash Utility or similar. For information on how to use this utility, consult the *Bitfire Software Developers Guide.*

3.2 MagicConfig and Firmware

This section describes the configuration options and commands trough MagicConfig available on the Bitfire. For more information on details of the configuration utility MCF and firmware upload etc please refer to the *'MagicConfig Handbook'*.

Use 'MCF -?' and 'MCF -?:bitfire' to get a flying start.

3.2.1 Implementation

The configuration MCU on Bitfire is based on MagicConfig code base, this is a common shell and utility for configuration of the board functionality without the need of using jumpers. The MCF utility itself has built-in documentation for all available settings. It connects to UART-0 and acts as I2C bridge config MCU giving access to the expansion board stackup.

3.2.2 Application Commands

There are a few unique commands used for controlling the behaviour of the Bitfire. This section describes the native commands trough the communication tunnel. MCF utility gives an abstraction of these and is easier to use.

3.2.2.1 Reset

Command: "reset T\r"
Response: "OK\r\n"
Desc: Resets the given target T can be any of the following:
'c' – Resets the MCU
'b' – Resets the MCU and activates the bootloader
'f' – Resets the FPGA

3.2.2.2 SetMode

Command: “setmode N\r”
Response: “OK\r\n”
Desc: Sets the current toolmode of the Bitfire, N can be any of the following:
‘0’ – MCU JTag
‘1’ – MCU Flash
‘2’ – MCU Extern
‘3’ – FPGA JTAG
‘4’ – FPGA Config

3.2.2.3 Bitfire Specifics

The Equipment ID for Bitfire V1.12 is 0x01. The hardware condition to force boot monitor lock is holding both buttons depressed during boot up.

4 Board description

The development board you now possess has dual targets, FPGA and a ARM7 MCU. Basic HW development tools are supported to both targets directly on the board. The only thing you have to do is to connect the board to a host PC in order to have full debug control over each target. No additional hardware is needed.

4.1 FPGA: Altera low Cost Cyclone Family

The Cyclone™ field programmable gate array family is based on a 1.5-V, 0.13-µm, all-layer copper SRAM process, with 20,060 logic elements (LEs) and up to 288 Kbits of RAM. With features like phaselocked loops (PLLs) for clocking and a dedicated double data rate (DDR) interface to meet DDR SDRAM and fast cycle RAM (FCRAM) memory requirements, Cyclone devices are a cost-effective solution for data-path applications. Cyclone devices support various I/O standards, including LVDS at data rates up to 640 megabits per second (Mbps), and 66- and 33-MHz, 64- and 32-bit peripheral component interconnect (PCI), for interfacing with and supporting ASSP and ASIC devices. Altera also offers new low-cost serial configuration devices to configure Cyclone devices.

Onboard we have a EP1C6 with 6000 logic elements in 144-pin TQFP package.

4.2 ARM7 TDMI-S: Philips LPC2129

The LPC2119/LPC2129 are based on a 16/32 bit ARM7TDMI-S™ CPU with real-time emulation and embedded trace support, together with 128/256 kilobytes (kB) of embedded high speed flash memory. A 128-bit wide memory interface and a unique accelerator architecture enable 32-bit code execution at maximum clock rate. For critical code size applications, the alternative 16-bit Thumb Mode reduces code by more than 30% with minimal performance penalty.

With their compact 64 pin package, low power consumption, various 32-bit timers, 4-channel 10-bit ADC, 2 advanced CAN channels, PWM channels and 46 GPIO lines with up to 9 external interrupt pins these microcontrollers are particularly suitable for automotive and industrial control applications as well as medical systems and fault-tolerant maintenance buses. With a wide range of additional serial communications interfaces, they are also suited for communication gateways and protocol converters as well as many other general-purpose applications.

Key features

- 16/32-bit ARM7TDMI-S microcontroller in a tiny LQFP64 package.
- 16 kB on-chip Static RAM.
- 128/256 kB on-chip Flash Program Memory. 128-bit wide interface/accelerator enables high speed 60 MHz operation.
- In-System Programming (ISP) and In-Application Programming (IAP) via on-chip boot-loader software. Flash programming takes 1 ms per 512 byte line. Single sector or full chip erase takes 400 ms.
- EmbeddedICE-RT interface enables breakpoints and watch points. Interrupt service routines can continue to execute whilst the foreground task is debugged with the on-chip RealMonitor software.
- Embedded Trace Macrocell enables non-intrusive high speed real-time tracing of instruction execution.
- Two interconnected CAN interfaces with advanced acceptance filters.
- Four channel 10-bit A/D converter with conversion time as low as 2.44 ms.
- Multiple serial interfaces including two UARTs (16C550), Fast I2C (400 kbits/s) and two SPIs™.
- 60 MHz maximum CPU clock available from programmable on-chip Phase-Locked Loop.
- Vectored Interrupt Controller with configurable priorities and vector addresses.
- Two 32-bit timers (with 4 capture and 4 compare channels), PWM unit (6 outputs), Real Time Clock and Watchdog.

4.3 The huge LED-matrix

A central part of the board is the high performance bi-color LED matrix made up by 40x16 pixels in 10 x 3.7” display modules. The FPGA design deployed from factory on this board contains a high performance LED display driver that support 8-bit intensity depth on each color and in each pixel. As the driver is driven with serial bit streams, this makes a good test bench and usage of the raw parallel power from the FPGA. See *section 8.5* for more details.

4.4 Expansion

The purpose of the board is to supply a good platform for design with MCU and FPGA using components from Arrows linecard. Expansion modules enable the board to function as a baseboard and can be equipped with different MCU’s and circuit design in a modular way. By defining the the standard for the expansion connector signals and defining a way to automatically configuring the board stackup you will be able to stack up expansion boards for different purposes and by software configure the behaviour of these.

Various processor-modules can replace the MCU fitted onboard by plugging on a module above the base connectors. In addition, more boards can be stacked above to supply application support for different topologies. These topologies can be analog functions, network interface and motor control.

4.5 Expansion headers

Almost all pins from FPGA and MCU are connected to 3 x 60 pins 2mm expansion connectors for user expansion and future expansion boards. All connections that share board functions can be isolated by removing jumper or termination resistors if it conflicts with the users needs, in addition there are plenty of free GPIO to use at your own discretion. See *section 6* for details on how to isolate a specific board function.

4.6 Power supplies

The power supply design is made up from several high performance step down switching regulators, which delivers more than enough current for board and combinations of future expansion boards. The LED-matrix is supplied from a synchronous step-down controller, which delivers the high current demand from the LED-matrix.

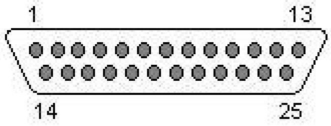
5 Connector pinouts

Below you will find pinout information on all the onboard connectors. You should consult the schematics and *section 6* for more detailed information on isolating board functions.

5.1 Interface connectors

5.1.1 Debug host Connector

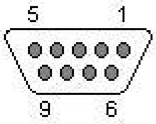
Connector type: D-SUB 25 way male
Connects to: Host debugging system like a PC
Reference: J13



Pin	Dir		Mode	
		FPGA J-Tag	FPGA Conf	ARM7 Wiggler
2	In	TCK	DCLK	Reset
3	In	TMS	nConfig	TMS
4	In	-	nCS	TCK
5	In	-	nCE	TDI
6	In	Loop pin 10	Loop pin 10	Loop pin 10
8	In	TDI	ASDI	Loop pin 15
10	Out	Loop pin 6	Loop pin 6	Loop pin 6
11	Out	TDO	CONF_DONE	TDO
13	Out	-	DATAOUT	-
14	In	Logic Enable	Logic Enable	-
15	Out	Low	Low	Loop pin 8
18-22	-	GND	GND	GND
Others	-	-	-	-

5.1.2 UART-0 & UART-1 Connectors

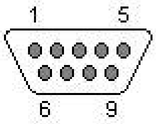
Connector type: D-SUB 9 way female
Connects to: Asynchronous serial device
Reference: J3, J4



Pin	Dir	Mode	
		RS232	RS485
2	Out	RxD	T-
3	In	TxD	R-
5	-	GND	GND
7	In	RTS	R+
8	Out	CTS	T+

5.1.3 CAN-0 & CAN-1 Connectors

Connector type: D-SUB 9 way male
Connects to: CAN network connection
Reference: J14, J15

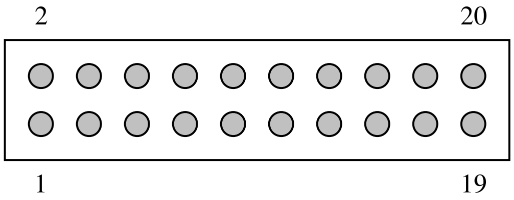


Pin	Dir	Signal
2	Bi	CANL
3	-	GND
5	-	GND
6	-	GND
7	Bi	CANH

5.2 Debug connectors

5.2.1 LPC2129 JTAG Connector

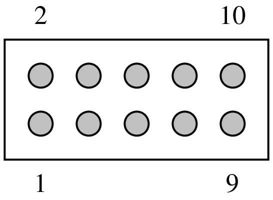
Connector type: 20pin 2.54mm pin-header
Connects to: External JTAG debug or test tool
Reference: J16



Pin	Dir	Signal
1-2	-	VCCIO
3	In	TRST
5	In	TDI
7	In	TMS
9	In	TCK
11	In	RTCK
13	Out	TDO
15	In	RESET
17	In	(DBGRQ) 10K Pull-down
19	Out	(DBGACK) 10K Pull-down
4, 6, 8, 10, 12, 14, 16, 18, 20	-	GND

5.2.2 FPGA JTAG Connector

Connector type: 10pin 2.54mm pin-header
Connects to: External JTAG debug or test tool
Reference: J17

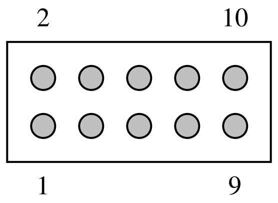


Pin	Dir	Signal
1	In	TCK
2	-	GND
3	Out	TDO

4	-	VCCIO
5	In	TMS
9	In	TDI
10	-	GND
Others	-	-

5.2.3 FPGA Config Connector

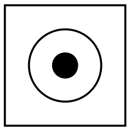
Connector type: 10pin 2.54mm pin-header
Connects to: External JTAG debug or test tool
Reference: J17



Pin	Dir	Signal
1	In	DCLK
2	-	GND
3	Out	CONF_DONE
4	-	VCCIO
5	In	nConfig
6	In	nCE
7	Out	DATAOUT
8	In	nCS
9	In	ASDI
10	-	GND

5.3 Power Connector

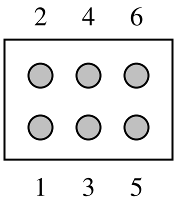
Connector type: DIN45323, Laptop connector DC connector
Connects to: Input power supply
Reference: J2



Pin	Dir	Signal
Centertap	-	DC+ in
Shield	-	GND

5.4 ISP ATMega48 Connector

Connector type: Footprint of 2x3 pin 2.54mm header (mount from bottom side)
Connects to: Programmer or debug tool for Atmel AVR ATMega48
Reference: J5



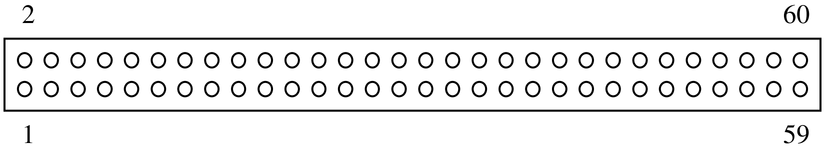
Foot print view from bottom side

Pin	Dir	Signal
1	Out	MISO (PGM Master In Slave Out)
2	-	Vcc (3.3V)
3	-	SCK (Programming Clock)
4	In	MOSI (PGM Master Out Slave In)
5	In	\Reset (Chip Reset)
6	-	GND

5.5 Expansion connectors

5.5.1 LPC2129 MCU Expansion Connector

Connector type: 60 pin 1 mm pin-header, SAMTEC TMx
Connects to: Expansion card and/or user application
Reference: J12



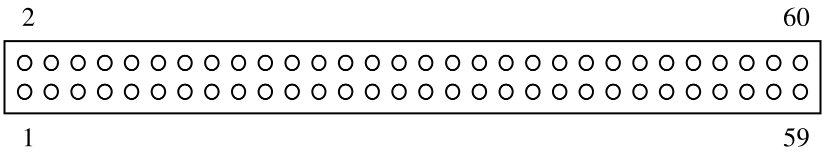
Pin	Dir	GPIO	Primary	Secondary	Third
1	Bi	P0.18	CAP1.3	MISO1	MAT1.3
2	Bi	P0.19	MAT1.2	MOSI1	CAP1.2
3	Bi	P0.20	MAT1.3	SSEL1	EINT3
4	***In***	***-***	***\RESET***	***-***	***-***
5	Bi	P1.20	TRACESYNC	-	-
6	Bi	P0.17	CAP1.2	SCK1	MAT1.2
7	Bi	P0.16	EINT0	MAT0.2	CAP0.2
8	***Bi***	***P0.15***	***RI1***	***EINT2***	***-***
9	Bi	P1.21	PIPESTAT0	-	-
10	***Bi***	***P0.14***	***DCD1***	***EINT1***	***-***
11	Bi	P1.22	PIPESTAT1	-	-
12	***Bi***	***P0.13***	***DTR1***	***MAT1.1***	***-***
13	***Bi***	***P0.12***	***DSR1***	***MAT1.0***	***-***
14	***Bi***	***P0.11***	***CTS1***	***CAP1.1***	***-***
15	Bi	P1.23	PIPESTAT2	-	-
16	***Bi***	***P0.10***	***RTS1***	***CAP1.0***	***-***
17	***Bi***	***P0.9***	***RxD1***	***PWM6***	***EINT3***
18	***Bi***	***P0.8***	***TxD1***	***PWM4***	***-***
19	Bi	P1.24	TRACECLK	-	-
20	Bi	P0.7	SSEL0	PWM2	EINT2
21	***Bi***	***P0.6***	***MOSI0***	***CAP0.2***	***-***
22	***Bi***	***P0.5***	***MISO0***	***MAT0.1***	***-***
23	Bi	P1.25	EXTIN0	-	-
24	***Bi***	***P0.4***	***SCK0***	***CAP0.1***	***-***
25	***Bi***	***P0.3***	***SDA***	***MAT0.0***	***EINT1***

26	***Bi***	***P0.2***	***SCL***	***CAP0.0***	***-***
27	***Bi***	***P0.1***	***RxD0***	***PWM3***	***EINT0***
28	***Bi***	***P0.0***	***TxD0***	***PWM1***	***-***
29	Bi	P1.16	TRACEPKT0	-	-
30	Bi	P0.30	AIN3	EINT3	CAP0.0
31	Bi	P0.29	AIN2	CAP0.3	MAT0.3
32	Bi	P0.28	AIN1	CAP0.2	MAT0.2
33	Bi	P1.17	TRACEPKT1	-	-
34	Bi	P0.27	AIN0	CAP0.1	MAT0.1
35	***Out***	***-***	***TD1***	***-***	***-***
36	***Bi***	***P0.25***	***RD1***	***-***	***-***
37	Bi	P1.18	TRACEPKT2	-	-
38	***Bi***	***P0.24***	***TD2***	***-***	***-***
39	Bi	P1.19	TRACEPKT3	-	-
40	***Bi***	***P0.23***	***RD2***	***-***	***-***
41	Bi	P0.22	CAP0.0	MAT0.0	-
42	Bi	P0.21	PWM5	CAP1.3	-
44, 46, 48, 50		-	+3V3	-	-
52, 54, 56, 58, 60		-	+1V8	-	-
43, 45, 47, 49, 51, 53, 55, 57, 59			GND	-	-

NOTE: IOs that have shared functions onboard are marked in bold italic in the list, precautions and information how to isolate these signals can be found in *section 6*.

5.5.2 FPGA Expansion connector #1

Connector type: 60 pin 1 mm pin-header, SAMTEC TMx
Connects to: Expansion card and/or user application
Reference: J10



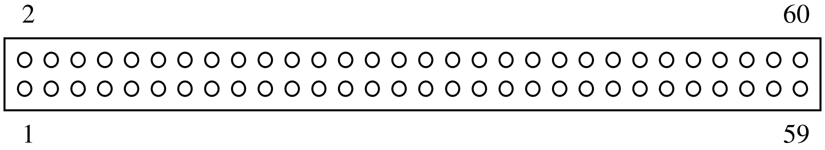
Pin	Dir	GPIO	Function	Config	Bank	VREF Bank
1	Bi	P108	LVDS35p	-	B3	VREF0B3
2	Bi	P107	LVDS35n	-	B3	VREF0B3
3	Bi	P106	LVDS36p	-	B3	VREF0B3
4	Bi	P105	LVDS36n	-	B3	VREF0B3
5	Bi	P104	VREF0B3	-	B3	VREF0B3
6	Bi	P103	-	-	B3	VREF0B3
7	Bi	P100	DPCLK4	-	B3	VREF0B3
8	Bi	P99	LVDS38p	-	B3	VREF0B3
9	Bi	P98	LVDS38n	-	B3	VREF0B3
10	Bi	P97	LVDS39p	-	B3	VREF0B3
11	Bi	P96	VREF1B3	-	B3	VREF1B3
12	Bi	P85	LVDS47n	-	B3	VREF2B3
13	Bi	P84	LVDS48p	-	B3	VREF2B3
14	Bi	P83	LVDS48n	-	B3	VREF2B3
15	Bi	P82	DPCLK5	-	B3	VREF2B3
16	Bi	P79	VREF2B3	-	B3	VREF2B3
17	Bi	P78	LVDS49p	-	B3	VREF2B3
18	Bi	P77	LVDS49n	-	B3	VREF2B3
19	Bi	P76	LVDS50p	-	B3	VREF2B3
20	Bi	P75	LVDS50n	-	B3	VREF2B3
21	Bi	P74	LVDS51p	-	B3	VREF2B3
22	Bi	P73	LVDS51n	-	B3	VREF2B3
23	Bi	P72	LVDS52n	-	B4	VREF0B4
24	Bi	P71	LVDS52p	-	B4	VREF0B4
25	Bi	P70	LVDS54n	-	B4	VREF0B4
26	Bi	P69	LVDS54p	-	B4	VREF0B4
27	Bi	P68	LVDS55n	-	B4	VREF0B4
28	Bi	P67	LVDS55p	-	B4	VREF0B4

29	***Bi***	***P62***	***DPCLK6***	***-***	***B4***	***VREF0B4***
30	***Bi***	***P61***	***VREF0B4***	***-***	***B4***	***VREF0B4***
31	Bi	P60	-	-	B4	VREF0B4
32	Bi	P59	LVDS59n	-	B4	VREF1B4
33	Bi	P58	LVDS61n	-	B4	VREF1B4
34	Bi	P57	LVDS61p	-	B4	VREF1B4
35	Bi	P56	VREF1B4	-	B4	VREF1B4
36	Bi	P53	LVDS62n	-	B4	VREF1B4
37	Bi	P52	LVDS62p	-	B4	VREF1B4
38	Bi	P51	LVDS63n	-	B4	VREF2B4
39	Bi	P50	LVDS63p	-	B4	VREF2B4
40	Bi	P49	LVDS67p	-	B4	VREF2B4
41	Bi	P48	VREF2B4	-	B4	VREF2B4
42	Bi	P47	DPCLK7	-	B4	VREF2B4
43	Bi	P42	LVDS68n	-	B4	VREF2B4
44	Bi	P41	LVDS68p	-	B4	VREF2B4
45	Bi	P40	LVDS69n	-	B4	VREF2B4
46	Bi	P39	LVDS69p	-	B4	VREF2B4
47	Bi	P38	LVDS71n	-	B4	VREF2B4
48	Bi	P37	LVDS71p	-	B4	VREF2B4
49, 51, 53, 55, 57, 59			GND	-	-	-
54, 56, 58, 60			+1V5	-	-	-
50			VCCIO_B4	-	B4	-
52			VCCIO_B1	-	B1	-

NOTE: IOs that have shared functions onboard are marked in bold italic in the list, precautions and information how to isolate these signals can be found in *section 6*.

5.5.3 FPGA Expansion connector #2

Connector type: 60 pin 1 mm pin-header, SAMTEC TMx
Connects to: Expansion card and/or user application
Reference: J11



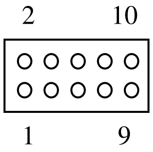
Pin	Dir	GPIO	Function	Config	Bank	VREF Bank
1	***Bi***	***P109***	***LVDS34n***	***-***	***B2***	***VREF0B2***
2	***Bi***	***P110***	***LVDS34p***	***-***	***B2***	***VREF0B2***
3	***Bi***	***P111***	***LVDS32n***	***-***	***B2***	***VREF0B2***
4	***Bi***	***P112***	***LVDS32p***	***-***	***B2***	***VREF0B2***
5	***Bi***	***P113***	***LVDS31n***	***-***	***B2***	***VREF0B2***
6	***Bi***	***P114***	***LVDS31p***	***-***	***B2***	***VREF0B2***
7	Bi	P119	DPCLK3	-	B2	VREF0B2
8	Bi	P120	VREF0B2	-	B2	VREF0B2
9	Bi	P121	-	-	B2	VREF0B2
10	Bi	P122	LVDS27n	-	B2	VREF1B2
11	Bi	P123	LVDS25n	-	B2	VREF1B2
12	Bi	P124	LVDS25p	-	B2	VREF1B2
13	Bi	P125	VREF1B2	-	B2	VREF1B2
14	Bi	P128	LVDS24n	-	B2	VREF1B2
15	Bi	P129	LVDS24p	-	B2	VREF1B2
16	Bi	P130	LVDS23n	-	B2	VREF2B2
17	Bi	P131	LVDS23p	-	B2	VREF2B2
18	Bi	P132	LVDS19p	-	B2	VREF2B2
19	Bi	P133	VREF2B2	-	B2	VREF2B2
20	Bi	P134	DPCLK2	-	B2	VREF2B2
21	Bi	P139	LVDS18n	-	B2	VREF2B2
22	Bi	P140	LVDS18p	-	B2	VREF2B2
23	Bi	P141	LVDS17n	-	B2	VREF2B2
24	Bi	P142	LVDS17p	-	B2	VREF2B2
25	Bi	P143	LVDS15n	DEV_OE	B2	VREF2B2
26	Bi	P144	LVDS15p	DEV_CLRn	B2	VREF2B2
27	Bi	P1	LVDS14p	INIT_DONE	B1	VREF0B1

28	Bi	P2	LVDS14n	-	B1	VREF0B1
29	Bi	P3	LVDS13p	CLKUSR	B1	VREF0B1
30	Bi	P4	LVDS13n	-	B1	VREF0B1
31	Bi	P5	VREF0B1	-	B1	VREF0B1
32	Bi	P6	-	-	B1	VREF0B1
33	Bi	P7	LVDS12p	-	B1	VREF0B1
34	Bi	P10	DPCLK1	-	B1	VREF0B1
35	***Bi***	***P11***	***VREF1B1***	***-***	***B1***	***VREF1B1***
36	Bi	P26	PLL1_OUTp	-	B1	VREF1B1
37	Bi	P27	PLL1_OUTn	-	B1	VREF1B1
38	Bi	P28	DPCLK0	-	B1	VREF2B1
39	Bi	P31	VREF2B1	-	B1	VREF2B1
40	Bi	P32	-	-	B1	VREF2B1
41	Bi	P33	LVDS1p	-	B1	VREF2B1
42	Bi	P34	LVDS1n	-	B1	VREF2B1
43	Bi	P35	LVDS0p	-	B1	VREF2B1
44	Bi	P36	LVDS1n	-	B1	VREF2B1
45	***In***	***-***	***CLK0***	***LVDSCLK0P***	***B1***	***VREF1B1***
46	***In***	***-***	***CLK1***	***LVDSCLK0N***	***B1***	***VREF1B1***
47	***In***	***-***	***CLK2***	***LVDSCLK2P***	***B3***	***VREF1B3***
48	In	-	CLK3	LVDSCLK2N	B3	VREF1B3
49, 51, 53, 55, 57, 59			GND	-	-	-
54, 56, 58, 60			+1V5	-	-	-
50			VCCIO_B3	-	B3	-
52			VCCIO_B2	-	B2	-

NOTE: IOs that have shared functions onboard are marked in bold italic in the list, precautions and information how to isolate these signals can be found in *section 6*.

5.5.4 Stack up debug connector

Connector type: 10 pin 1 mm socket, SAMTEC SMM
Connects to: Expansion card stack up
Reference: J6

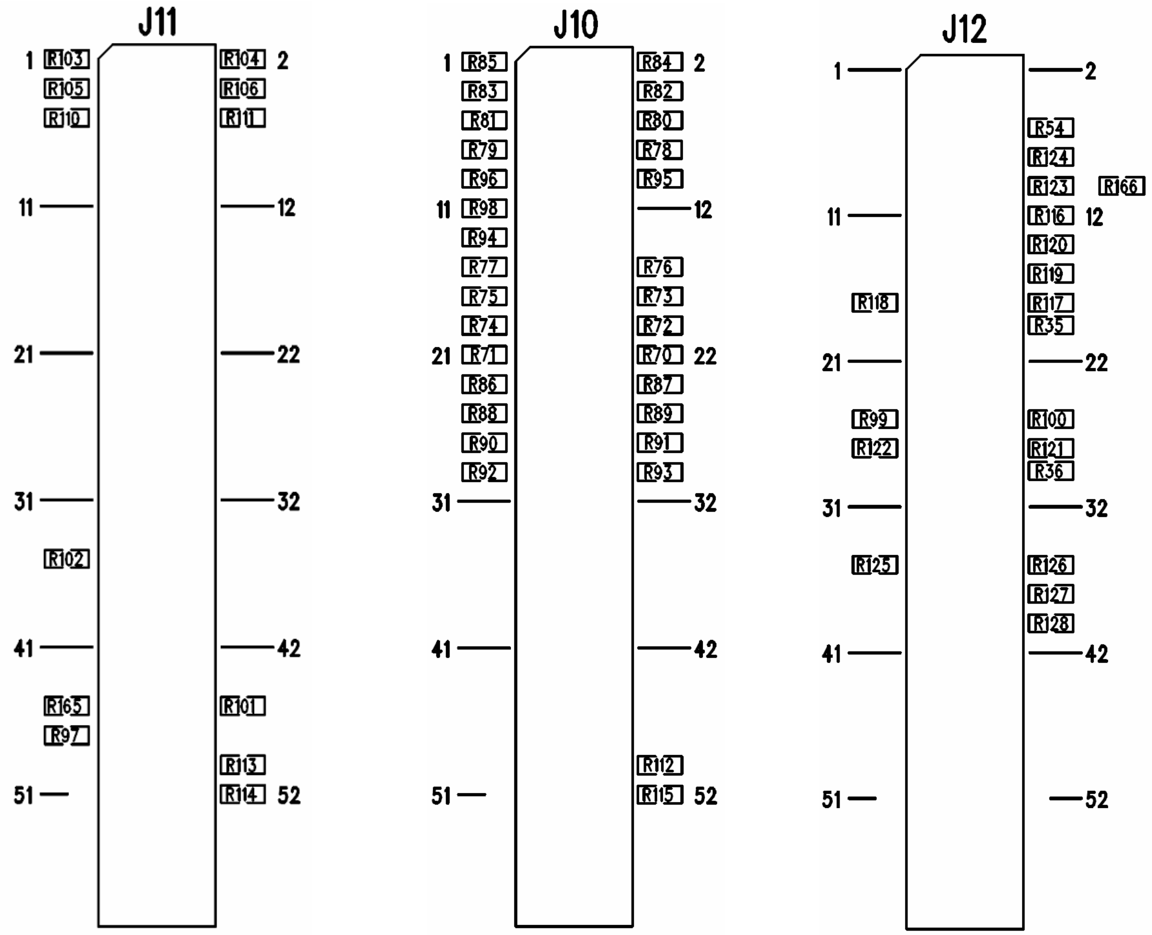


Pin	Dir	Signal
1	Out	\TRST
2	Out	TDI
3	-	+3V3
4	Out	TMS
5	Out	\Reset
6	Out	TCK
7	-	GND
8	In	TDO
9	-	GND
10	Out	DBGRQS

6 Signal sharing and isolation

Pins used for kit onboard functions to/from the LPC2129 and/or FPGA can be isolated by removing jumpers or termination resistors. This section contains information on how to do this and what the precautions are if you use the shared function pins without isolating them from the board application. Also refer to *Bitfire Schematics.pdf* that can be found on the CD for more details.

5.6 Resistor location reference



5.7 General on signals to the FPGA from the LPC2129

Signals connected from the LPC2129 to the FPGA have some leakage even if it's configured in input mode. Therefore you are encouraged to isolate these signals if used by off board applications.

5.8 Shared and isolated

Definition of an isolated signal is when it is disconnected from its default onboard function. Removing a specific resistor for the function does this.

The definition of a shared signal is that it still has its function connected but are used for some other purpose, this perfectly fine for a number of functions.

Below sections will take you trough the details on what you need to be aware of when isolating or sharing pins that have dedicated functions onboard.

5.9 LPC2129: Port0 signals

5.9.1 P0.0 – CPU_TxD0

This is the asynchronous transmission signal for UART0.

Precautions when shared
UART-0 is used for the LPC2129 flash boot-loader. Interception of this signal will make it impossible to send data via the UART-0 I/F connector.

Direction:	Out
Isolation resistor:	R121 (0R)

5.9.2 P0.1 – CPU_RxD0

This is the asynchronous receive signal for UART0.

Precautions when shared
UART-0 is used for the LPC2129 flash boot-loader. Interception of this signal will make it impossible to receive data trough the UART-0 I/F connector. This is actively driven by B1 output of IC24 giving a possible collision if not isolated.

Direction:	In
Isolation resistor:	R122 (47R)

5.9.3 P0.2 – CPU_I2C_SCL

This is the I2C interface clock-signal. It is connected to the FPGA, but has no given functionality by default. It will be used to assist the configuration for expansion boards.

To reduce the current leakage you can isolate the signal from the FPGA. A 10K pull-up resistor is present on this pin, even after isolation.

Direction:	Out/In
Isolation resistor:	R101 (0R)

5.9.4 P0.3 – CPU_I2C_SDA

This is the I2C interface data-signal. It is connected to the FPGA, but has no given functionality by default. It will be used to assist the configuration for expansion boards. To reduce the current leakage you can isolate the signal from the FPGA. A 10K pull-up resistor is present on this pin, even after isolation.

Direction:	Out/In
Isolation resistor:	R102 (0R)

5.9.5 P0.4 – CPU_SPI_CLK

This is the master SPI0 output clock, it is used to communicate with FPGA for the LED-matrix driver application.

Precautions when shared
Intercepting or isolating this signal will remove the main data-link to the FPGA, making it impossible to run the LED-matrix application. If the FPGA is running it can also alter the operation of LED-matrix spurious and/or the UART-0 & UART-1 mode configurations.

Direction:	Out
Isolation resistor:	R97 (0R)

5.9.6 P0.5 – CPU_MISO

Master-In-Slave-Out signal of the SPI0 used to communicate with the FPGA. However, in our LED-matrix application we are not currently using the read-back channel but it is reserved to be used in future FPGA / MCU code revisions.

Precautions when shared
If function is implemented in the future, this pin will be a driven output from the FPGA causing a collision if not isolated.

Direction:	In
Isolation resistor:	R96 (0R)

5.9.7 P0.6 – CPU_MOSI

Master-Out-Slave-In signal of SPI0 used to communicate with the FPGA.

Precautions when shared

Intercepting or isolating this signal will remove the main data-link to the FPGA, making it impossible to run the LED-matrix application. If the FPGA is running and this signal is not isolated, but shared it can also alter the operation of LED-matrix spurious and/or the UART-0 & UART-1 mode configurations.

Direction:	In
Isolation resistor:	R95 (0R)

5.9.8 P0.7 – SSEL0

This is not a pure shared board function but it has a very important meaning for the SPI0 interface.

Precautions when shared

SPI0 is used as the main pipe for communicating with the FPGA. The nature of the SPI peripheral in the LPC2129 prescribes that this pin MUST be pulled high in order for the SPI0 interface to work. So this pin has a 10K pull-up resistor and if pulled low by external application the SPI0 will cease operation, making the LED-matrix impossible to access from the MCU.

Direction:	In
Isolation resistor:	N/A

5.9.9 P0.8 – CPU_TxD1

This is the asynchronous transmission signal for UART1.

Precautions when shared

Interception of this signal will make it impossible to send data via the UART-1 I/F connector.

Direction:	Out
Isolation resistor:	R117 (0R)

5.9.10 P0.9 – CPU_RxD1

This is the asynchronous receive signal for UART1.

Precautions when shared

Interception of this signal will make it impossible to receive data via the UART-1 I/F connector. This is actively driven by B3 output of IC24 giving a possible collision if not isolated.

Direction:	In
Isolation resistor:	R118 (47R)

5.9.11 P0.10 – CPU_RTS_DE1

This is the hardware handshake request-to-send signal for UART1 in RS232 mode or driver enable for RS485 mode.

Precautions when shared
Interception of this signal will make it impossible to send handshake signal onto the UART-1 I/F connector or controlling the driver enable.

Direction:	Out
Isolation resistor:	R119 (0R)

5.9.12 P0.11 – CPU_CTS1

This is the hardware handshake clear-to-send signal for UART1.

Precautions when shared
Interception of this signal will make it impossible to receive handshake signal via the UART-1 I/F connector. This is actively driven by B2 output of IC24 giving a possible collision if not isolated.

Direction:	In
Isolation resistor:	R120 (47R)

5.9.13 P0.12 – CPU_SPI_SYNC

This is the SPI0 software sync pin used to synchronize the communication to the FPGA for the LED-matrix application driver.

Precautions when shared
Intercepting or isolating this signal will remove the main data-link to the FPGA, making it impossible to run the LED-matrix application. If the FPGA is running and this signal is not isolated but shared it can also alter the operation of LED-matrix spurious and/or the UART-0 & UART-1 mode configurations.

Direction:	Out
Isolation resistor:	R98 (0R)

5.9.14 P0.13 – CPU_SPI_SEL

This is the SPI0 software command/data selection pin to select data type sent to the FPGA for the LED-matrix application driver.

Precautions when shared
Intercepting or isolating this signal will remove the main data-link to the FPGA making it impossible to run the LED-matrix application. If the FPGA is running and this signal is

not isolated but shared it can also alter the operation of LED-matrix spurious and/or the UART-0 & UART-1 mode configurations.

Direction:	Out
Isolation resistor:	R116 (0R)

5.9.15 P0.14 – CPU_CTS0

This is the software driven hardware handshake clear-to-send signal for UART0.

Precautions when shared
Great care must be taken when using this pin as shared or isolated; a low signal on this pin when LPC2129 is coming out of reset will activate the onboard boot-loader. This pin is also driven by the onboard debug controller when a boot-loader activation request is issued.

Interception of this signal will make it impossible to receive handshake signal via the UART-0 I/F connector. This is actively driven by B0 output of IC24 trough a 2K2 resistor that possibly could affect user signal.

Direction:	In
Isolation resistor:	R123 (2K2)

5.9.16 P0.15 – CPU_RTS_DE0

This is the software driven hardware handshake request-to-send signal for UART1 in RS232 mode or driver enable for RS485 mode.

Precautions when shared
Interception of this signal will make it impossible to send handshake signal onto the UART-0 I/F connector or controlling the driver enable.

Direction:	Out
Isolation resistor:	R124 (0R)

5.9.17 P0.23 – CPU_CANRX1

MCU CAN interface CAN2 receive channel.

Precautions when shared
Interception of this signal will make it impossible to receive data via the CAN-1 I/F connector. This is actively driven by B5 output of IC24 giving a possible collision if not isolated.

Direction:	In
Isolation resistor:	R128 (47R)

5.9.18 P0.24 – CPU_CANTX1

MCU CAN interface CAN2 transmission channel.

Precautions when shared
Interception of this signal will make it impossible to send data via the CAN-1 I/F connector.

Direction:	Out
Isolation resistor:	R127 (0R)

5.9.19 P0.25 – CPU_CANRX0

MCU CAN interface CAN1 receive channel.

Precautions when shared
Interception of this signal will make it impossible to receive data via the CAN-0 I/F connector. This is actively driven by B4 output of IC24 giving a possible collision if not isolated.

Direction:	In
Isolation resistor:	R126 (47R)

5.9.20 CPU_CANTX0

MCU CAN interface CAN1 transmission channel. This pin is dedicated to CAN transmission, it has no port functionality.

Precautions when shared
Interception of this signal will make it impossible to send data via the CAN-0 I/F connector.

Direction:	In
Isolation resistor:	R125 (0R)

5.10 LPC2129: Port1 signals

Port1 on LPC2129 is mainly used to multiplex the debug interface; the signals dedicated for running and activating the JTAG interface are strictly dedicated to the debug connector/controller onboard. If you still want to use P1.26 – P1.31 you need to access that trough the JTAG connector. See schematic for details.

5.11 LPC2129: Special pins during reset

There are three port pins that need extra care when performing or coming out of a reset, these are used to activate functions such as debug interface and boot-loader.

P0.14 described above activate the boot-loader if sampled low when coming out of a reset.

P1.26 activates the JTAG interface multiplexed on port pins P1.26 – P1.31 if sampled low during a reset.

P1.20 activates the trace-port interface multiplexed on port pins P1.16 - P1.25 if sampled low during a reset.

5.12 FPGA pins

The pins connected to the LPC2129 are described in detail under *section 6.4*. Below you find a table for an overview which signals are connected between the FPGA and LPC2129. These will not be described further under this section.

Netname	FPGA	LPC2129	Description
CPU_I2C_SCL	CLK1	P0.2	I2C clock signal
CPU_I2C_SDA	P11	P0.3	I2C data signal
CPU_SPI_CLK	CLK2	P0.4	MCU SPI0 clock signal
CPU_MISO	P98	P0.5	MCU SPI0 MISO signal
CPU_MOSI	P97	P0.6	MCU SPI0 MOSI signal
CPU_SPI_SYNC	P96	P0.12	MCU SPI0 soft sync signal
CPU_SPI_SEL	P85	P0.13	MCU SPI0 soft cmd/data selection

5.12.1 Pin 82 – CLK_SPI0

Clock signal for SPI blast interface 0 (upper half of display).

Precautions when shared
Interception of this signal will make it impossible to operate the LED-matrix driver.

Direction:	Out
Isolation resistor:	R77 (22R)

5.12.2 Pin 79 – \ENABLE_SPI0

Enable signal for SPI blast interface 0 (upper half of display).

Precautions when shared
Interception of this signal will make it impossible to operate the LED-matrix driver.

Direction:	Out
Isolation resistor:	R76 (22R)

5.12.3 Pin 78 - \LATCH_SPI0

Latch signal for SPI blast interface 0 (upper half of display).

Precautions when shared
Interception of this signal will make it impossible to operate the LED-matrix driver.

Direction:	Out
Isolation resistor:	R75 (22R)

5.12.4 Pin 77 – D0_SPI0

Data channel 0 signal for SPI blast interface 0 (upper half of display, LD1).

Precautions when shared
Interception of this signal will make it impossible to operate the LED-matrix driver.

Direction:	Out
Isolation resistor:	R73 (47R)

5.12.5 Pin 76 – D1_SPI0

Data channel 1 signal for SPI blast interface 0 (upper half of display, LD2).

Precautions when shared
Interception of this signal will make it impossible to operate the LED-matrix driver.

Direction:	Out
Isolation resistor:	R74 (47R)

5.12.6 Pin 75 – D2_SPI0

Data channel 2 signal for SPI blast interface 0 (upper half of display, LD3).

Precautions when shared
Interception of this signal will make it impossible to operate the LED-matrix driver.

Direction:	Out
Isolation resistor:	R72 (47R)

5.12.7 Pin 74 – D3_SPI0

Data channel 3 signal for SPI blast interface 0 (upper half of display, LD4).

Precautions when shared
Interception of this signal will make it impossible to operate the LED-matrix driver.

Direction:	Out
Isolation resistor:	R71 (47R)

5.12.8 Pin 73 – D4_SPI0

Data channel 4 signal for SPI blast interface 0 (upper half of display, LD5).

Precautions when shared
Interception of this signal will make it impossible to operate the LED-matrix driver.

Direction:	Out
Isolation resistor:	R70 (47R)

5.12.9 Pin 108 – CLK_SPI1

Clock signal for SPI blast interface 1 (lower half of display).

Precautions when shared
Interception of this signal will make it impossible to operate the LED-matrix driver.

Direction:	Out
Isolation resistor:	R85 (22R)

5.12.10 Pin 107 – \ENABLE_SPI1

Enable signal for SPI blast interface 1 (lower half of display).

Precautions when shared
Interception of this signal will make it impossible to operate the LED-matrix driver.

Direction:	Out
Isolation resistor:	R74 (22R)

5.12.11 Pin 106 - \LATCH_SPI1

Latch signal for SPI blast interface 1 (lower half of display).

Precautions when shared
Interception of this signal will make it impossible to operate the LED-matrix driver.

Direction:	Out
Isolation resistor:	R83 (22R)

5.12.12 Pin 105 – D0_SPI1

Data channel 0 signal for SPI blast interface 1 (lower half of display, LD6).

Precautions when shared

Interception of this signal will make it impossible to operate the LED-matrix driver.

Direction:	Out
Isolation resistor:	R82 (47R)

5.12.13 Pin 104 – D1_SPI1

Data channel 1 signal for SPI blast interface 1 (lower half of display, LD7).

Precautions when shared

Interception of this signal will make it impossible to operate the LED-matrix driver.

Direction:	Out
Isolation resistor:	R81 (47R)

5.12.14 Pin 103 – D2_SPI1

Data channel 2 signal for SPI blast interface 1 (lower half of display, LD8).

Precautions when shared

Interception of this signal will make it impossible to operate the LED-matrix driver.

Direction:	Out
Isolation resistor:	R80 (47R)

5.12.15 Pin 100 – D3_SPI1

Data channel 3 signal for SPI blast interface 1 (lower half of display, LD9).

Precautions when shared

Interception of this signal will make it impossible to operate the LED-matrix driver.

Direction:	Out
Isolation resistor:	R79 (47R)

5.12.16 Pin 99 – D4_SPI1

Data channel 4 signal for SPI blast interface 1 (lower half of display, LD10).

Precautions when shared

Interception of this signal will make it impossible to operate the LED-matrix driver.

Direction:	Out
Isolation resistor:	R78 (47R)

5.12.17 Pin 61 – EN0_MUX

Multiplex signal for display row 0.

Precautions when shared
Interception of this signal will make it impossible to operate the LED-matrix driver.

Direction:	Out
Isolation resistor:	R83 (0R)

5.12.18 Pin 61 – EN0_MUX

Multiplex signal for display row 0.

Precautions when shared
Interception of this signal will make it impossible to operate the LED-matrix driver.

Direction:	Out
Isolation resistor:	R93 (0R)

5.12.19 Pin 62 – EN1_MUX

Multiplex signal for display row 1.

Precautions when shared
Interception of this signal will make it impossible to operate the LED-matrix driver.

Direction:	Out
Isolation resistor:	R92 (0R)

5.12.20 Pin 67 – EN2_MUX

Multiplex signal for display row 2.

Precautions when shared
Interception of this signal will make it impossible to operate the LED-matrix driver.

Direction:	Out
Isolation resistor:	R91 (0R)

5.12.21 Pin 68 – EN3_MUX

Multiplex signal for display row 3.

Precautions when shared
Interception of this signal will make it impossible to operate the LED-matrix driver.

Direction:	Out
Isolation resistor:	R90 (0R)

5.12.22 Pin 69 – EN4_MUX

Multiplex signal for display row 4.

Precautions when shared
Interception of this signal will make it impossible to operate the LED-matrix driver.

Direction:	Out
Isolation resistor:	R89 (0R)

5.12.23 Pin 70 – EN5_MUX

Multiplex signal for display row 5.

Precautions when shared
Interception of this signal will make it impossible to operate the LED-matrix driver.

Direction:	Out
Isolation resistor:	R88 (0R)

5.12.24 Pin 71 – EN6_MUX

Multiplex signal for display row 6.

Precautions when shared
Interception of this signal will make it impossible to operate the LED-matrix driver.

Direction:	Out
Isolation resistor:	R87 (0R)

5.12.25 Pin 72 – EN7_MUX

Multiplex signal for display row 7.

Precautions when shared
Interception of this signal will make it impossible to operate the LED-matrix driver.

Direction:	Out
Isolation resistor:	R86 (0R)

5.12.26 Pin 84 – DIMMERPWM

Programmable display current PWM output.

Precautions when shared
Interception of this signal will interfere with the LED-matrix driver operation (light intensity).

Direction:	Out
Isolation resistor:	R94 (0R)

5.12.27 Pin 109 – ON_RSX

Signal for controlling on/off status of UART-0 and UART-1 transceiver interface.

Precautions when shared
Interception of this signal will interfere with the operation of both UART interface connections.

Direction:	Out
Isolation resistor:	R103 (0R)

5.12.28 Pin 110 – \LB_RSX

Signal for controlling hardware loop back mode of UART-0 and UART-1 transceiver interface.

Precautions when shared
Interception of this signal will interfere with the operation of both UART interface connections.

Direction:	Out
Isolation resistor:	R104 (0R)

5.12.29 Pin 111 – SEL1_RSX

Signal for controlling the mode in which UART-0 interface connection work in (low RS232 and high RS485).

Precautions when shared
Interception of this signal will interfere with the operation of UART-0 interface connection.

Direction:	Out
Isolation resistor:	R105 (0R)

5.12.30 Pin 112 – SEL1_RSX

Signal for controlling the mode in which UART-1 interface connection work in (low RS232 and high RS485).

Precautions when shared
Interception of this signal will interfere with the operation of UART-1 interface connection.

Direction:	Out
Isolation resistor:	R106 (0R)

5.12.31 Pin 113 – \SHDN_MUXSUP

Signal to control on/off for the special +10V supply used to drive the row multiplexer of the LED-matrix.

Precautions when shared
Interception of this signal will make it impossible to operate the LED-matrix driver. You can isolate and override this function by hardwire the signal high at the DC/DC converter side.

Direction:	Out
Isolation resistor:	R110 (0R)

5.12.32 Pin 113 – SHDN_LEDSUP

Signal to control on/off for the main VLED supply used to drive the LED-matrix.

Precautions when shared
Interception of this signal will make it impossible to operate the LED-matrix driver. You can isolate and override this function by hardwire the signal low at the DC/DC converter side.

Direction:	Out
Isolation resistor:	R111 (0R)

5.12.33 Pin 16 – CLK0

Main input clock, an onboard crystal oscillator drives this pin with a 16MHz TTL signal. This is used as a main clock in the LED-matrix driver application.

Precautions when shared
Loss of this signal or unclocked pin will make it impossible to run the LED-matrix driver application.

Direction:	Out
Isolation resistor:	R165 (56R)

6 Future expansion

Some signals are predefined for use in expansion boards making a flexible solution and possibility to stack up several expansion modules without conflict. Part of the expansion connectors has also predefined signals and electrical signal types.

The Bitfire baseboard can take a new MCU module to replace the existing LPC2129 by putting this into reset and overriding the signals making the Bitfire a general purpose MCU evaluation platform.

6.1 Different board types

Five different board types are defined:

- Processor module with basic expansion connector (PMBE)
- Processor module with extended expansion connector (PMEE)
- Application module half size (AMHS)
- Application module full size (AMFS)
- Application module full size with extended expansion (AMFSE)

6.1.1 Processor module – PMBE

This module fits on top of Bitfire J10, J11, J12 and J6 expansion connectors. The module takes control over board functions by automatically putting the Bitfire CPU into reset. It also replaces the IO passed on up-streams the stack with its own. The Reset signal is clamped below 0.5V to keep the board LPC2129 in reset but is sensed for low-pulls and if detected regenerates a new reset signal for the processor module. It is prohibited to actively drive force reset signal high by external tool. Some obscure tools on the market actually do this by brute force and substantial current injection.

6.1.2 Processor module – PMEE

The main difference between PMBE and PMEE modules are that PMEE introduces a fourth expansion header next to J10 in the upstream path. This gives additional expansion possibilities including a 16-bit external bus interface.

6.1.3 Application module – AMHS

This application module is connecting only to J10 and J12 for basic IO functionality. These boards need to be placed on the top of stack up as they will not offer J11 pass trough.

6.1.4 Application module – AMFS

Module that covers J10, J11 and J12 connectors including pass trough of signals.

6.1.5 Application module – AMFSE

These application modules make use of the extended bus connector on extended processor modules. Typically these application modules utilizes external memory interface of the processor or have a large IO need.

6.2 Concept of configuration

The I2C interface has a central role for the configuration for the stack of expansion boards. There are 4 dedicated analog signal pins and 6 additional analog pins in the bus connector. Application modules cross connect the rest of IOs locally, there might also be analog multiplexing depending on application module. Housekeeper on the Bitfire features a special I2C bridge to the stack trough the UART-0 interface to a PC running MCF configuration utility see MagicConfig dedicated section and *'MagicConfig Handbook'* for more details. To use the onboard wiggler tool connector J6 passes the signals from the Bitfire to the application module. Cross connection of IOs can also be done by hardware if special needs exist. Processor modules will form the standard for where different peripherals will have allocated pins in the connector. As base we will start from the current setup.

6.2.1 Reserved signal pins

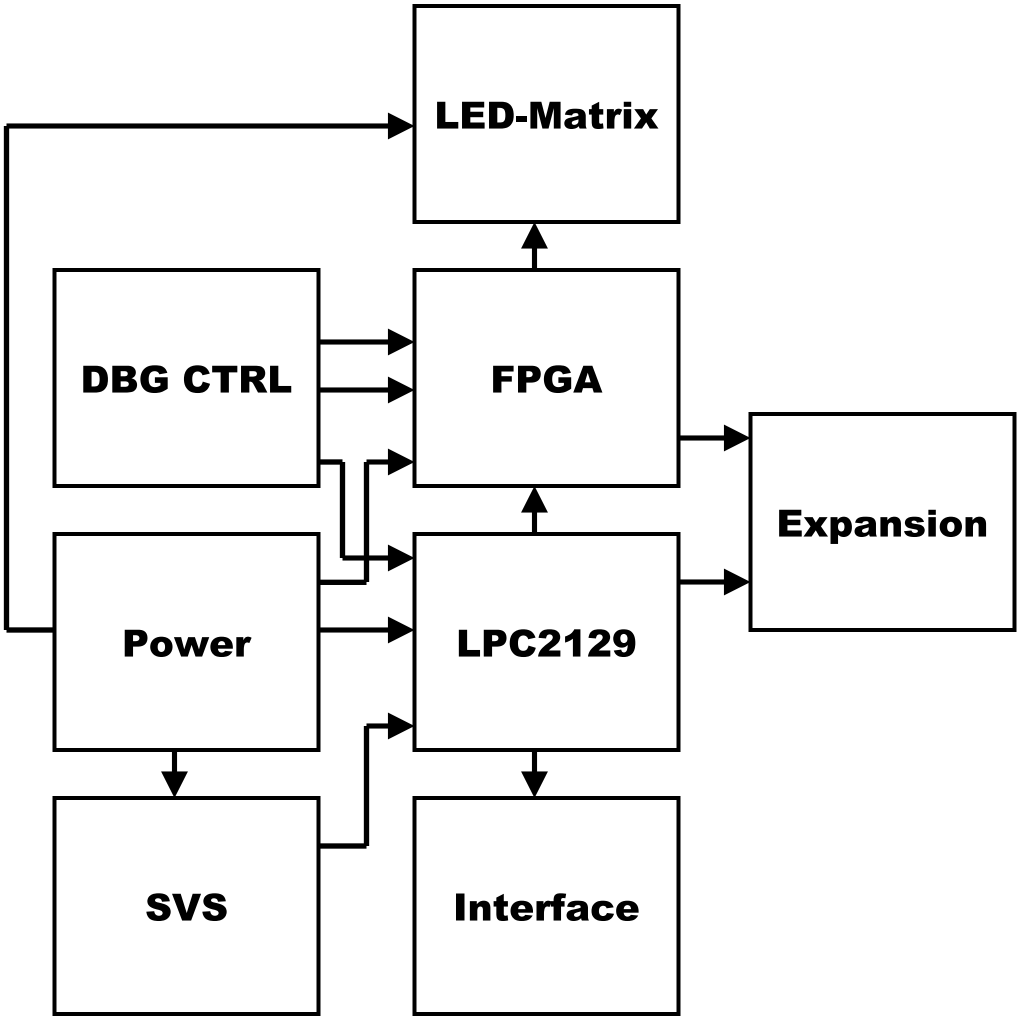
Connector	Description
J12-25	Configuration I2C – SDA
J12-26	Configuration I2C – SCL
J12-30:33	Analog 0 … 3
J12-43:60	Supply
J13-1:16	D0:15
J13-17:37	A0:19
J13-38	\WE
J13-39	\OE
J13-40	\UBYTE
J13-41	\LBYTE
J13-42:47	\CS0:5
J13-48:53	Analog 4 … 9
J13-54:60	Supply

7 Design reference

In this section we will go trough and comment the hardware design, block by block. Refer to the '*Bitfire Schematic.pdf*' and '*Bitfire Layout.pdf*' that can be found on the CD as overall reference for the following sections. You will also find gerber there, if you want the design files we can on a case by case basis supply these.

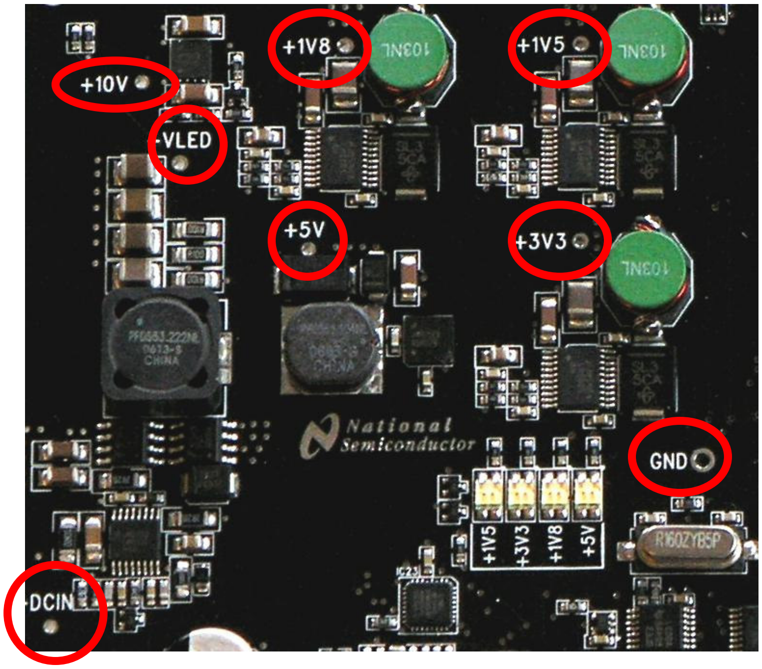
This schematic symbol indicates that there is a test point available on the PCB for easy access.

7.1 Simple block diagram



7.2 Testpoints

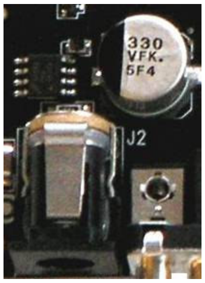
As the layer stack has embedded power and ground planes in the inner layers we have provided a number of test points in addition to connectors for easy low ohm access of supply voltages and ground.

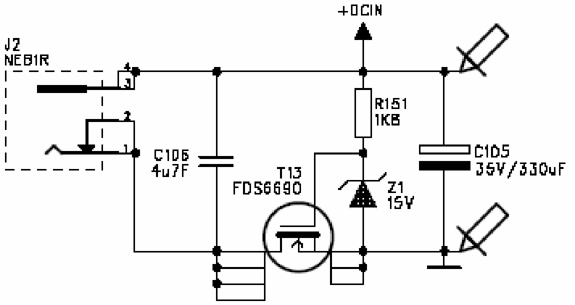


7.3 Power

The power design consists of several switched DC/DC converters supplying core, IO and LED-matrix. I addition to the converters the power design features a 4-channel voltage supervisor to ensure that all voltages are present and within tolerance. On the input circuit we have a lossless supply polarity protection. For spreadsheet calculations on inductors and caps, please refer to the *'SMPS_Calculations.xls'* file and datasheet of each converter supplied on the CD.

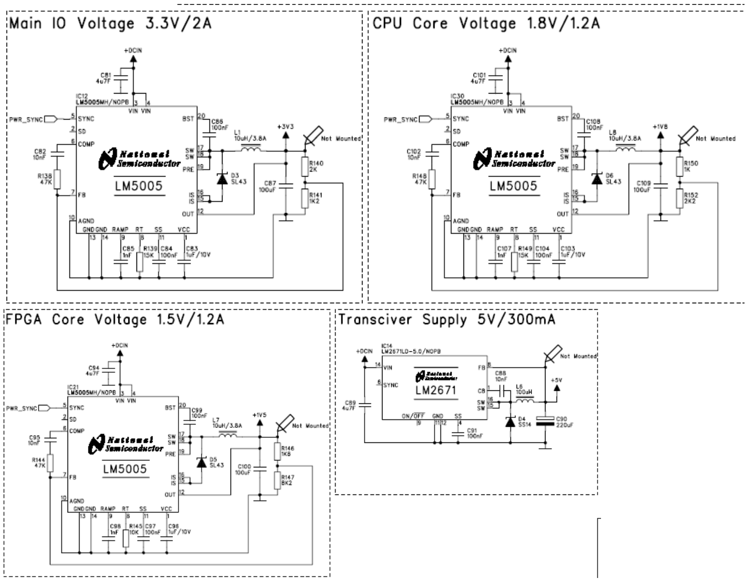
7.3.1 Polarity protection

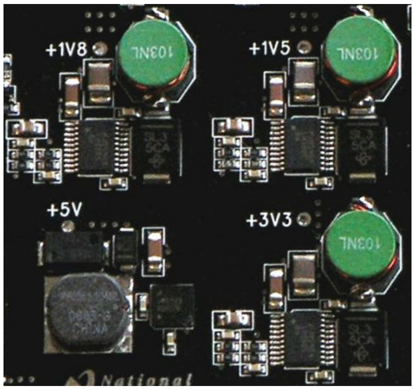




The polarity protection uses the internal parasitic diode in the N-channel MOSFET as a DC blocker. When wrong polarity is applied on the input terminal the transistor is off and the internal diode is blocking the voltage. When the correct polarity is applied the diode conducts and the transistor gate opens which then short circuits diode (transistor working in reverse). This method makes it possible to have a very low loss.

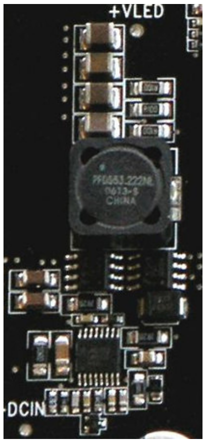
7.3.2 Core and IO supply

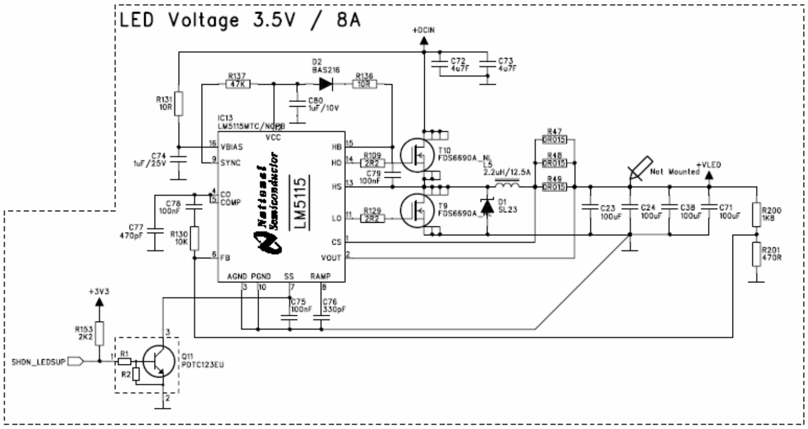




3 high performance LM5005 is used to regulate 3v3, 1v8 and 1v5 voltages. LM5005 and the brand new LM5576 gives a economic stepdown solution covering wide range input voltages and features softstart, shutdown programmable compensation and more in a very attractive footprint size. For more information on Simple Switcher™ visit National Semiconductors homepage. http://www.national.com

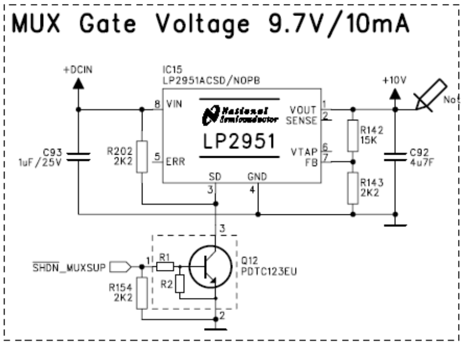
7.3.3 +VLED voltage





The supply for the big LED-matrix is designed around a LM5115 synchronous stepdown controller, this gives a very economic and solution delivering the high current output needed to drive the led display. It delivers 8A at 3.5V over the whole input voltage range, and it generates almost no heat in operating state. The benefit with having the MOSFET switches externally is that you can optimize the performance. With output currents as high as this it's very important to have short loops on the output filter, with external switches the layout becomes almost ideal.

In order to drive the multiplexing MOSFETs of the display a 9.70V (+10VDC) tiny linear LP2951 regulator is used. This regulator has its own enable signal from the FPGA; \SHDN_MUXSUP.

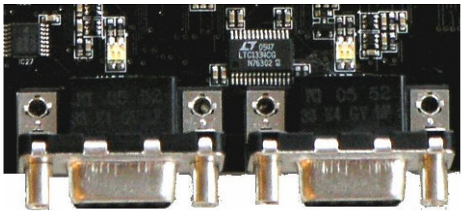


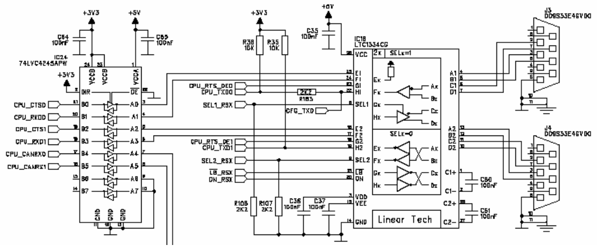
7.4 Interfaces

All interfaces have level translators as required to get clean logic signals to the MCU without injecting current into the body diodes. The UART-0 and UART-1 has a combined RS232 and RS485 modes on their transceiver, these are selectable by software by setting commands to the FPGA. The UART transceiver also features a hardware loop back mode and power down/standby operation. CAN channels has a termination network that can be activated on each channel trough a dipswitch.

Each interface channel has a two color LED indicating traffic activity. Green indicates traffic going outbound and red indicates incoming traffic. Yellow indicates traffic going in both directions.

7.4.1 UART Transceiver

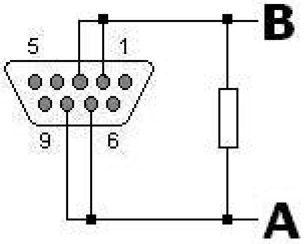




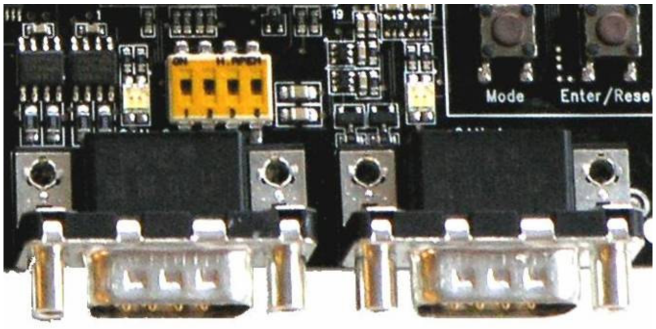
The LTC1334CG has dual signal modes, the same transceiver can act as a RS232 or RS485 physical interface on the same pins. IC24 is translating the incoming 5V logic signals from the transceivers to board IO voltage of 3.3V. R107 and R108 make sure that if the outputs from the FPGA are floating (non configured) the SELx signals forces the transceiver into RS232 mode. As the transition level is well below 3.3V on 5V logic inputs we don't need to run level translations for the output signals. Refer to the datasheet of LTC1334CG for details.

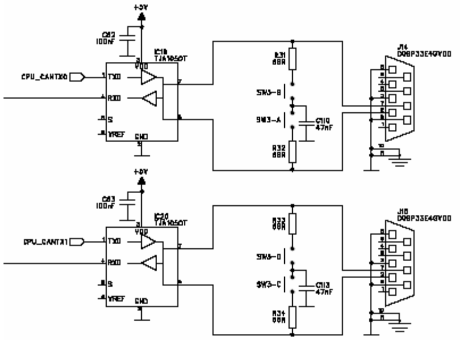
7.4.1.1 RS485 termination connection

For proper connection of RS485, transmitter and receiver pins must be tied together and terminated in the d-sub connector. As these pins are shared with RS232 and the termination will differ for most users this was not suitable to have onboard. See also *section 5.1.2* for detailed pinout information.



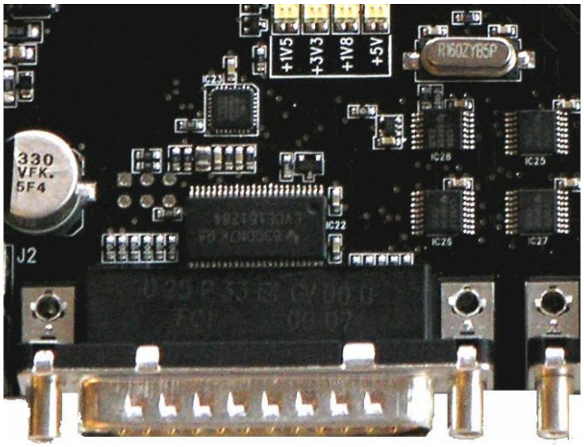
7.4.2 CAN Transceivers

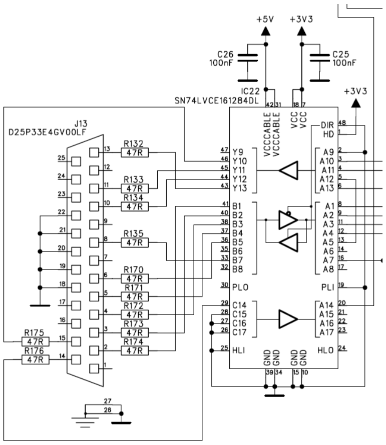




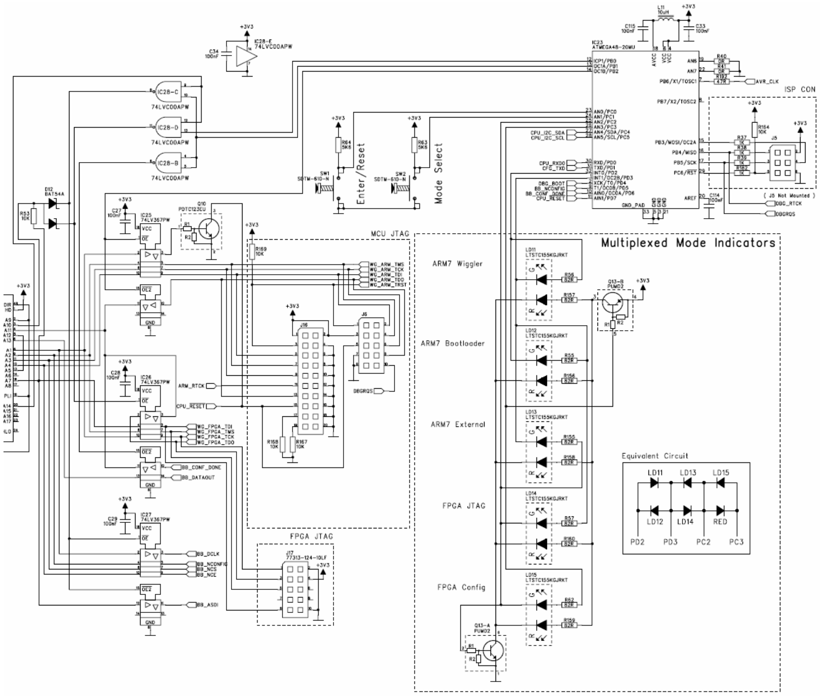
The CAN transceivers RX signals are also run trough IC24. It's possible to enable / disable termination on each channel, normally you only have termination in the outer two ends of a CAN network. TJA1050 is a low EMI high speed CAN-transceiver that is very popular in the industry. The S pin is a sleep select pin, which we don't utilize in this application; refer to the datasheet for more detailed information. The pinout of the 9 way DSUBs are what has become to be a standard for DSUB based CAN connectors.

7.4.3 Host debug interface (IEEE1284)





This is the connection to the host PC used to debug the application. For proper operation, good signal quality and ruggedness we have chosen to use an IEEE1284 transceiver and level translator. This device translates any centronics interface to good quality 3V3 logic signals both ways. Termination resistors improve the signal integrity even more over medium long cables.



The debug controller is actively reconfiguring the centronics port to match the behaviour of ByteBlaster II or Wiggler hardware. The signals can be software routed for FPGA JTAG, LPC2129 JTAG or FPGA configuration programmer or tristate to emulate cable disconnection and enable external tools to take control over the target(s). The debug controller MCU controls the debug tool behaviour, MCU reset, FPGA reconfiguration, boot loader activation and user control interface, see *section 3* for more details on how to operate this.

In the debug logic a discrete built crystal oscillator is built from a free NAND-gate in the control logic. This is used to clock the FPGA with a reliable and stable clock. This is then fed to CLK0 input of the FPGA.

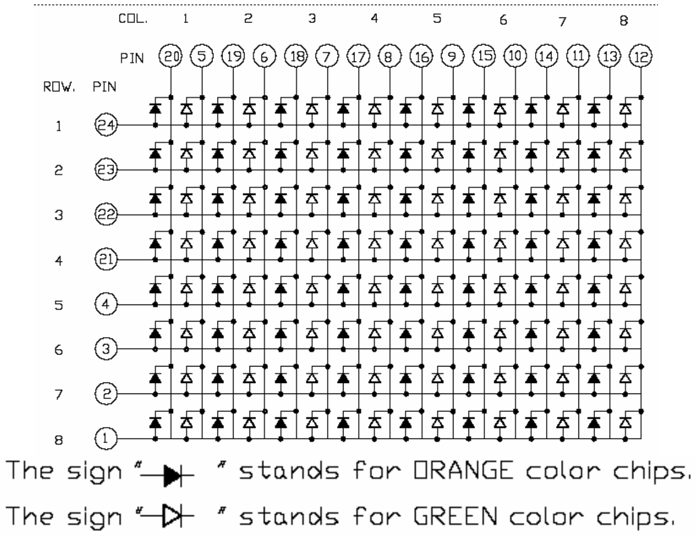
7.5 Display driver

This section explains how the LED-matrix display is driven. We use a fairly complex way of multiplexing the matrix. Each pixel in the 40x16 dot matrix has two colors; red and green. Each of these LEDs can be programmed with a 8-bit value that will result in a pulsed duty cycle of constant current, this results in 255 steps of current setting individually for on all 1280 LED's in the matrix. This gives the FPGA a meaningful and demanding task to perform.

7.5.1 Overview

With the signal setup we have, there are two possible ways of driving the matrix, 1/16 or 1/8 duty cycle. We will only describe the method of driving with 1/8 duty as this is the method used by the example FPGA driver, supplied with the kit.

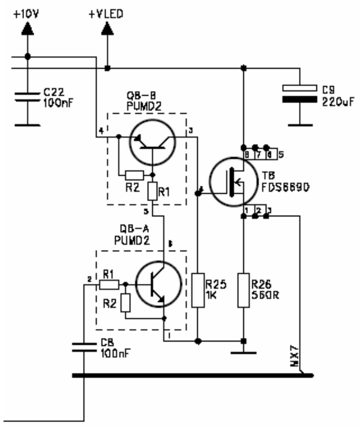
The display is divided into two parts; upper and lower. Each half has 5 led modules consisting of 8x8 bi-color pixels.


COL. 1 2 3 4 5 6 7 8
PIN 20 5 19 6 18 7 17 8 16 9 15 10 14 11 13 12
ROW. PIN
1 24
2 23
3 22
4 21
5 4
6 3
7 2
8 1
The sign "—▶|—" stands for ORANGE color chips.
The sign "—▷|—" stands for GREEN color chips.

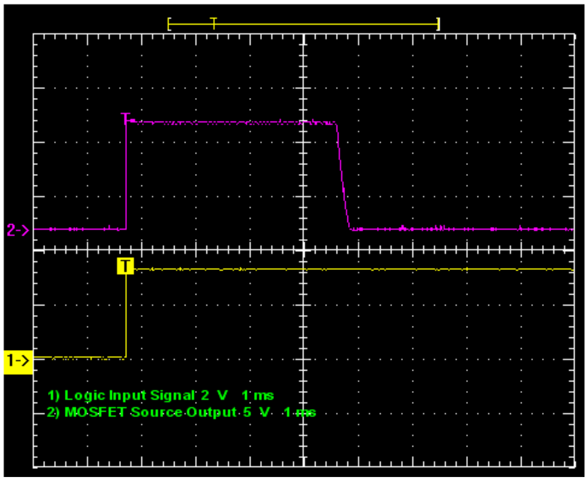
There are two high speed 5 channel SPIs associated with each half one channel per module.

7.5.2 Row multiplexer

The row multiplexer is feeding current to a single selected row in each half, each row will be given a timeslot of 1/8 and rows will be sequentially updated at a frequency well above the limit being possible for the human eye to see.



Above you can see the power stage available for each ROW selector. The MOSFET drain is supplied by the VLED voltage (around 4.5V). FPGA logic inputs a row select signal trough C8 activating Q8-A that will pull the base of Q8-B low. When Q8-B opens, the +10V will charge the gate of T8. The difference in gate drive voltage and source voltage ensures that the MOSFET will saturate. When deselecting the channel R25 ensures that the gate is pulled to the ground and deactivates the multiplex channel. If an input signal longer than approx 3.5ms is applied, C8 will discharge and the channel will self-close. This is a protection for halted multiplexing which could destroy the LEDs due to over current.

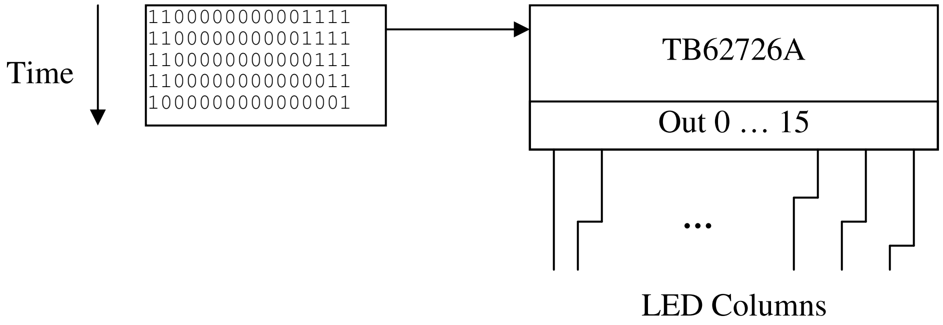


Row input signal timeout

7.5.3 Row update and pulse generation

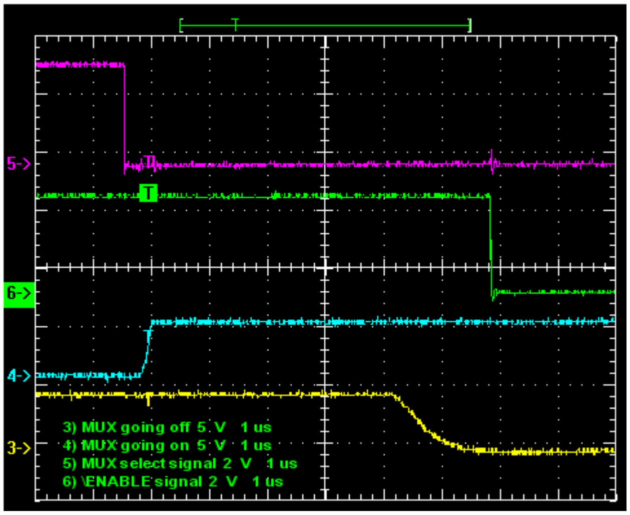
When each row is lit from the multiplexer the column constant current drivers are responsible for generating a single pulse of the programmed length. Each column driver responsible for a 8x8 module has its own SPI channel which feeds 16 bits of data corresponding to red and green half of the selected 8 pixel row.

The 16-bit data word is shifted out and latched 256 times on each module during each row update making time 256 timeslots. These timeslots are used to adjust the point where the light pulse is turned off. The speed of the SPI channels is currently 16MHz.



7.5.4 Control signals

The \ENABLE signal has an important task, when the multiplexer is switching from one row to another an overlap will occur. This is due to row multiplexer MOSFET driver will be faster switching on the channel than switching off. To prevent shadows in the matrix meaning two rows simultaneously driven for a short duration. The \ENABLE signal provides tri-state control for the column drivers during the overlap period.

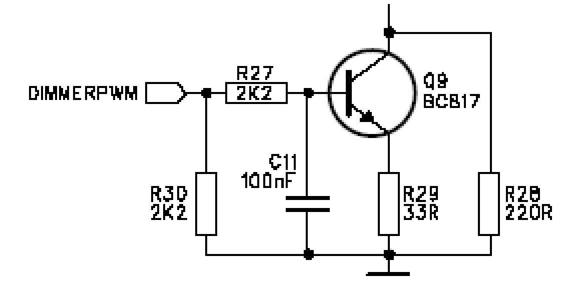


Enable signal in overlap period

The \LATCH signal is different compared to clocked latches. \LATCH is pulsed high after data is shifted into the shift register. High level makes the data going trough the latch and low signal again freezes the latch. This is not totally obvious in the datasheet of TB62726A.

7.5.5 Common current setting (DIMMERPWM)

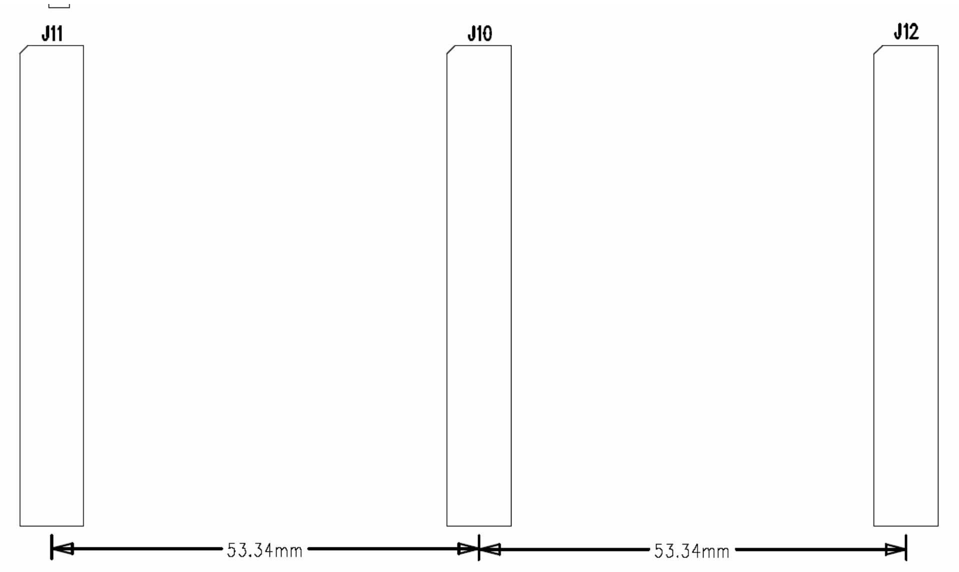
All the column drivers has a common current setting, all the current programming pins (R-ext) are tied together trough 100R resistors and then fed to a digital controlled constant current generator setting the overall driver current for the columns. This gives a way to control the overall display current/intensity in a precise and even way, without loosing dynamic on the intensity setting for each pixel.



DIMMERPWM signal is a 100 kHz pulse width signal fed from the FPGA. The low pass filter R27 and C11 converts this into a DC voltage on the base of Q9. This converts into a constant current of I = (Vbase-0.6)/R29. R28 is base setting for the display current if the current generator is not providing any current sink. With 3.3V I/O the circuit saturates at around 50% duty cycle and this is also giving the maximum allowed current feed to the LED's for a 1/8 multiplex drive. The DIMMERPWM is controlled trough commands to the FPGA, please consult the *BSP reference* in the *Bitfire Software Developers Guide*.

8 Expansion connector placement

Below you find pitch information for the baseboard expansion connectors see http://www.samtec.com for mating parts. Bitfire is equipped with TMM-130-01-F-D-SM.



This page is intentionally left blank.

Notes: