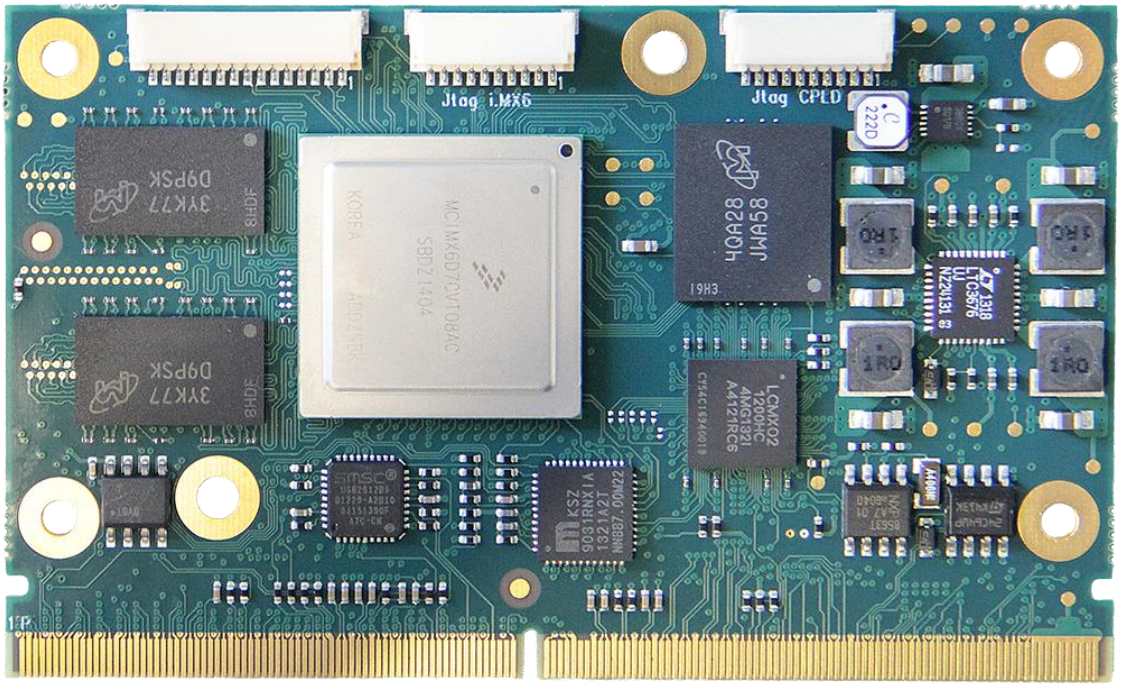
EZ-ROJ-1 SMARC Module

Reference Manual



ROJ s.r.l.
Via Vercellone, 11
13900 Biella, Italy
Tel. +39 015 8480111
Fax +39 015 xxx
EMail: info@roj.com
Web: www.roj.com

Rev: 1
Date: 8/5/2015
Classification: Public

Trademarks

Visit our web page at www.roj.com
For support requests, contact us at info@roj.com
For documentation suggestions, corrections, or requests, contact info@roj.com

Revision History

Revision	Date	Description
0	16/3/2015	First Draft
1	8/5/2015	Added notes on 1.8V only pins; added warranty terms and design services chapters

1 Introduction

Document Scope

This document covers technical features, pin assignment, software and hardware guidelines for the implementation of EZ-ROJ-1 SMARC module.

References

1. **SGeT.** SMARC Hardware Specification V1.1. [Online]
http://www.sget.org/uploads/media/SMARC_Hardware_Specification_V1p1.pdf
2. —. SMARC Design Guide V1.0. [Online]
http://www.sget.org/uploads/media/SMARC_DG_V1p0.pdf
3. **YoctoProject.** Documentation. [Online]
https://www.yoctoproject.org/documentation
4. —. Quick Start. [Online]
http://www.yoctoproject.org/docs/current/yocto-project-qs/yocto-project-qs.html

Definitions, Symbols, Acronyms

Acronyms

API	Application Programming Interface
COM	Computer On Module
CVBS	Composite Video Baseband Signal
DHCP	Dynamic Host Configuration Protocol
DSP	Digital Signal Processor
DVD	Digital Versatile Disc
DVI	Digital Visual Interface
FTP	File Transfer Protocol
FPGA	Field Programmable Gate Array
HDA	High Definition Audio
HMI	Human Machine Interaction
HSMC	Host Mass Storage Class
LCD	Liquid Crystal Display
MCU	Micro Controller Unit
NFC	Near Field Communication
PCB	Printed Circuit Board
RAM	Random Access Memory
SD	Secure Digital
SoC	System On Chip
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SPDIF	Sony/Philips Digital Interface
USB	Universal Serial Bus
VGA	Video Graphics Array

Introduction

EZ-ROJ-1 is a SMARC module based on iMX6 processor with ARM® Cortex™ A9 architecture, a complete 64-bit data bus, Quad/Dual/Single Core 1GHz speed SoC engine.
General specifications for SMARC (1) and SMARC Design Guide (2) are available publicly for free however, in order to speed up time to market, in addition to the companion Carrier board, ROJ is able to provide you several hardware design utilities including Schematics Checklist and Layout Checklist for Carrier board design.
ROJ also offers Evaluation Carrier Board for every single module product and brings you the necessary equipment and S/W test utility that will help reduce design efforts and speed up application development.

Note

The EZ-ROJ-1 module has been designed following the SGeT standard (1) however since design has been started with version 1.0, the board is still able to provide I/O compatibility with 1.8 and 3.3V whereas the new standard mandates for 1.8V only. See section 4.21.4 for details.

Features

EZ-ROJ-1 adopts Freescale i.MX6 Quad/Dual/Single Core Processor - ARM® Cortex™ A9 architecture as its SoC solution. The main features of this platform are followed by SMARC 1.1 standard (1) and can be used with or without heatsink, depending on the version of the CPU and on the software application being used.
EZ-ROJ 1 is perfectly suited for applications such as HMI (Human Machine Interface), Industrial control, Multimedia and digital signage and many more.
The main features of Freescale i.MX6 processors are shown as follows:

- ARM Cortex™-A9 high performance processor, available in quad, dual and single core at up to 1GHz
- Supports OpenGL ES 2.0 and OpenVG™ 1.1 hardware accelerators, full HD 1080p video codec
- Freescale Smart Speed™ Technology support low power consumption
- Industrial temperature range (-40 ~ 85°C) (operation temperature)

EZ-ROJ-1 features the following on board peripherals:

Peripheral	Notes
CPU	Single, Dual or Quad, depending on assembly
RAM	Up to 4GByte on single or dual channel, depending on assembly
eMMC	4GByte standard, other sizes on request.
Ethernet PHY	Micrel KSZ9031 10/100/1000 GBit Ethernet PHY
RTC	NXP PCF8563T for lowest power time retention
Power Management	Linear Technologies LT3676UI

CPLD	Lattice LCMXO2-1200ZE-3MG132I
Design Security	Atmel SHA204 for secure key storage

EZ-ROJ-1 features the following I/O ports:

Port	Quantity	Notes
Parallel Display	1x 24 bit	LVTTL
LVDS Display	2x 4 channel	First channel on SMARC connector, second on module through expansion connector
HDMI	1	
Parallel Camera interface	Up to 2	Configurable as 1x16 bit, 1x10 bit, 2x 8 bit depending on hardware and software options
Serial Camera Interface	1	2 lane MIPI, available with single port 10 bit parallel camera interface only
SDIO	1	4 bit SDIO port for SD cards and SDIO modules
eMMC	1	8 bit eMMC port for eMMC or additional SD card ports
SPI	2	2x SPI ports with 2 hardware controlled chip selects each
I2S	3	3x independent I2S ports with Audio MCK output support (no HDA support)
SPDIF	1	Support for independent input and output
I2C	4+1	Up to 5 I2C interfaces including: • Power Management • General Purpose • Camera • LCD (Software/GPIO based) • HDMI
Serial Ports	2+2	2 TX/RX only + 2 with handshake controls (RTS/CTS)
CAN	2	
USB	1+2	1 OTG + 2 Host ports. Host ports are derived through a hub from a single processor port
PCIe	1	Gen2 (5Gbps) Single lane available on PCIe link A
SATA	1	Single Sata II port at 3 Gbps
Ethernet	1	10/100/1000 ethernet phy with auto negotiation and auto MDIX
Watchdog	1	On chip watchdog with dedicated watchdog output pin
GPIO	12	In addition to serving as GPIO these pins can also be used, in conjunction with others, to implement iMX6 EIM interface. PWM and Tachometer can be implemented using the on board CPLD
AFB	1	Hosting a total of 10 additional GPIOs and 5 differential pairs. Differential pairs can be hardware configured as • MLB+MIPI CSI • MIPI DSI Note that MIPI CSI consists in 2 additional lanes to be used with the first 2 lanes available in the serial

		camera interface

Order code information

EZ-ROJ-1 can be ordered in several assembly configurations. Below you can find a simple part decoder useful for ordering the board code you require

Code example	Description	Notes
EZ-ROJ-1	Product prefix	
-S	Processor cores	Can be specified as (S)ingle, (D)ual and (Q)uad
-1	Processor Memory	Can have the following values -0 512 Mbytes (Single core only) -1 1 GByte (Preferred) -2 2 GBytes -4 4 GBytes (Quad core only)
-S	Parallel Camera	Can have the following values -S Serial Camera Option (Preferred) -P Parallel Camera Option
-M	AFB	Can have the following values -M MLB + MIPI CSI (Preferred) -P MIPI DSI + MIPI CSI

Design Services

More and more often new applications in the industrial field require the computational capacity and the versatility that are getting closer the performance of a personal computer, including the need to use modern operating systems in order to allow customers to concentrate on the development of its specific application.
Today the introduction of low cost and increasingly complex devices with "industrial" characteristics such as longevity, extended temperature and so on makes possible the realization of embedded products at costs that were impossible to imagine a few years ago. This also thanks to the success obtained by the ARM architecture in the recent years becoming a valid alternative to more traditional X86 architecture.
In this view ROJ, with its module EZ-ROJ-1, wants to provide its customers a development base for their applications. The module contains all the basic elements needed in an embedded application:
processor, storage and volatile memory, connectivity (USB, CAN, Ethernet, various types of serial interface, inputs for camera sensors and display out, etc.), clock, power circuits; all tested and working with the associated Linux board support package.

The user simply has to design (possibly in cooperation with ROJ) the carrier board customization and its SW application.

HW and SW design

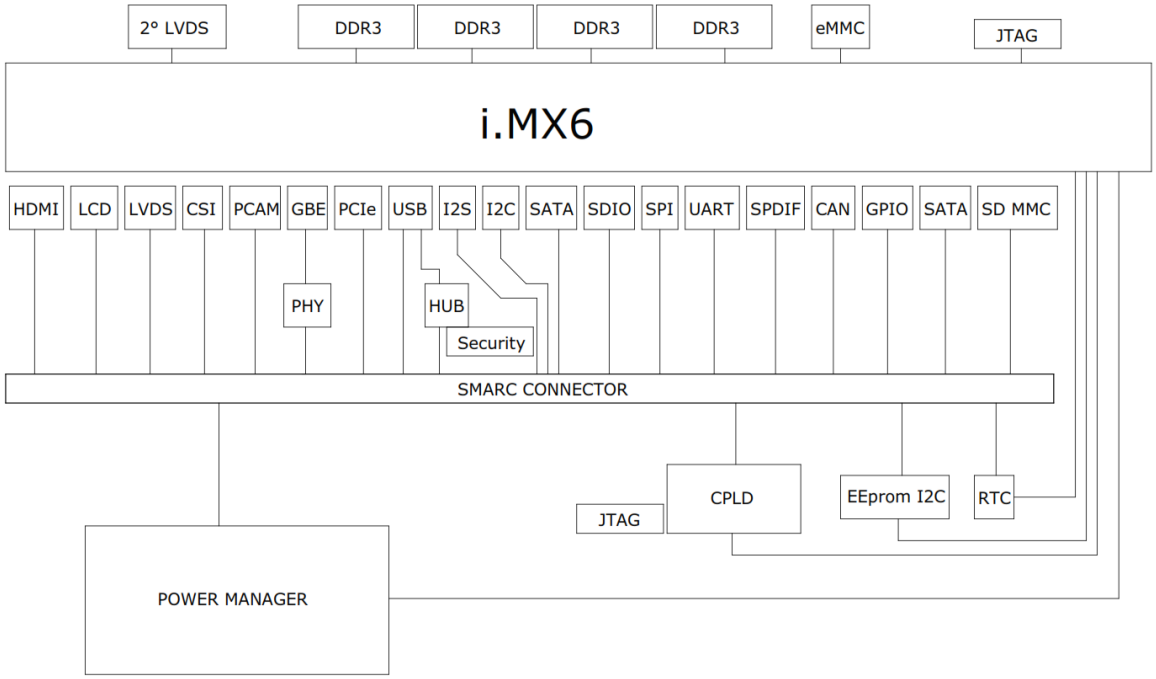
Our team is able to assist customers to review the placement/layout and schematics to ensure that the carrier board design meets their full requirements. ROJ's team can provide customers support to set up software build environment, development tools and evaluate the amount of time and resources needed for a project and, if necessary, ROJ can develop custom drivers in order to support specific Hardware. ROJ is also available to support third parties in case customer is using independent design houses for their application.

Contact Information

The contact information for ROJ customer service: info@roj.com

2
Hardware Reference

Block Diagram



Electrical specification

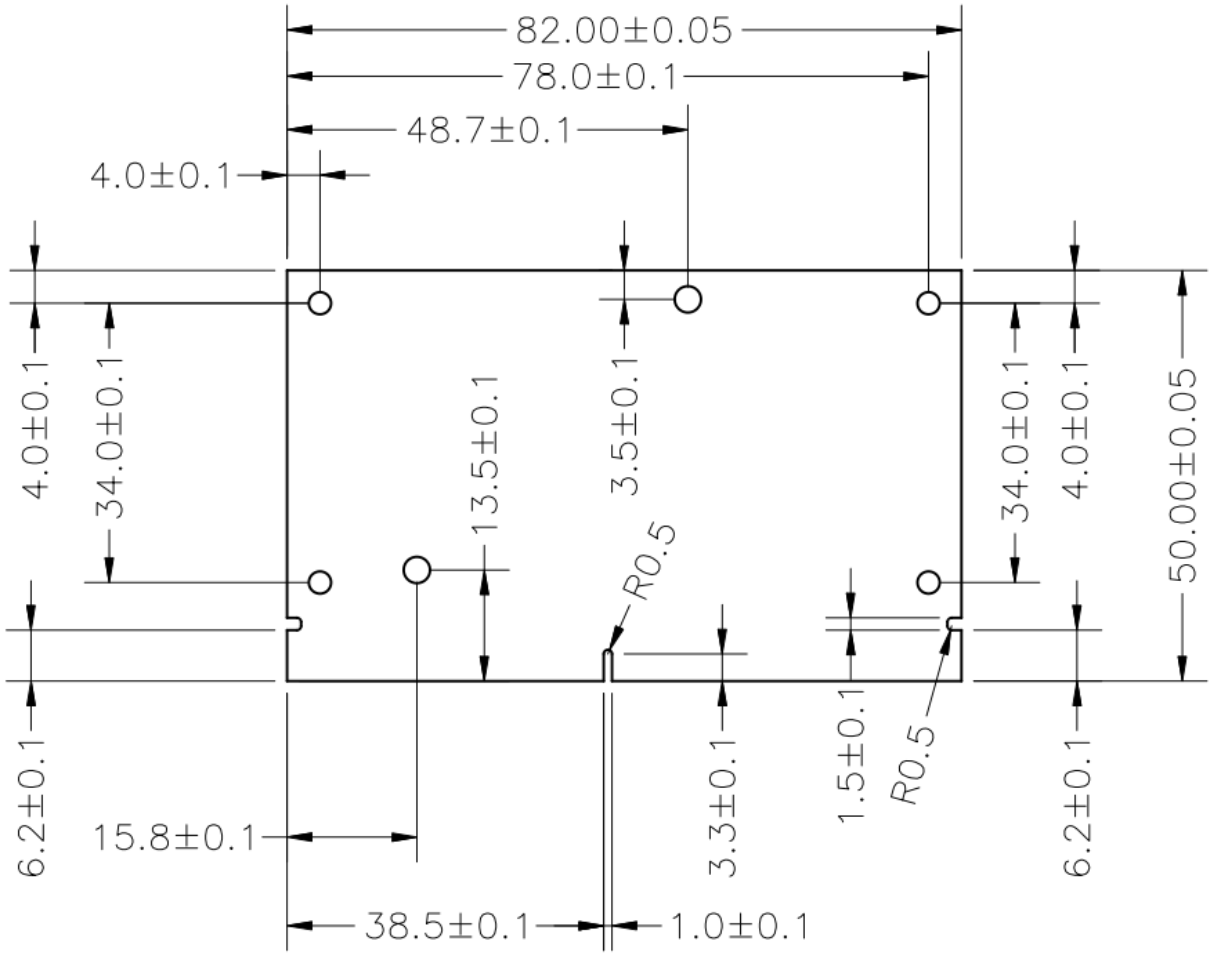
The following supply voltages are specified at the SMARC connector:

Rail	Min	Max	Current rating
VCC	3.0V	5.5V	Up to 2A
RTC	1.0V	5.5V	250nA@3V / 25°C

Input voltages shall rise monotonically from ≤10% of nominal to within the regulation ranges within 0.1ms to 20ms.
A maxium of 100mVpp of ripple in the band 0-20MHz is acceptable for reliable use.

Mechanical specification

Module dimensions are 50mm x 82mm.
Max component height on top layer (excluding heat spreader) is 3mm.
Max component height on bottom layer is 1.3mm.
Module 3D model is available from ROJ on request



Power Control and boot modes

EZ-ROJ-1 power control pins are managed directly from the embedded CPLD, which is powered independently from the iMX6, hence it's possible to customize the behavior of the board at startup by changing its firmware.
CPLD also controls enable and sequencing of all the power rails and is connected via I2C to the iMX6 so it's possible to implement advanced power control mechanisms if required.

SMARC provides a number of pins to control power up and standby:

Pin name	Pin number	Description
POWER_BTN#	P128	Power-button input from Carrier board. Carrier to float the line in inactive state. Active low, level sensitive. de-bounced on the Module Pulled up on Module. Driven by OD part on Carrier.
VIN_PWR_BAD#	S150	Power bad indication from Carrier board. Module and Carrier power supplies (other than Module and Carrier power supervisory circuits) shall not be enabled while this signal is held low by the Carrier.

		Pulled up on Module. Driven by OD part on Carrier.
CARRIER_STBY#	S153	The Module drives this signal low when the system is in a standby power state
RESET_OUT#	P126	General purpose reset output to Carrier board.
RESET_IN#	P127	Reset input from Carrier board. Carrier drives low to force a Module reset, floats the line otherwise. Pulled up on Module. Driven by OD part on Carrier.
SLEEP #	S149	Sleep indicator from Carrier board. May be sourced from user Sleep button or Carrier logic. Carrier to float the line in in-active state. Active low, level sensitive. de-bounced on the Module. Pulled up on Module. Driven by OD part on Carrier.
LID#	S148	Lid open/close indication to Module. Low indicates lid closure (which system may use to initiate a sleep state). Carrier to float the line in in-active state. Active low, level sensitive. De-bounced on the Module Pulled up on Module. Driven by OD part on Carrier.
BATLOW#	S156	Battery low indication to Module. Carrier to float the line in in-active state. Pulled up on Module. Driven by OD part on Carrier.
CHARGING#	S151	Held low by Carrier during battery charging. Carrier to float the line when charge is complete. Pulled up on Module. Driven by OD part on Carrier.
CHARGER_PRSNT#	S152	Held low by Carrier if DC input for battery charger is present. Pulled up on Module. Driven by OD part on Carrier.

Boot selection is handled through the BOOT_SEL [2:0] pins as follows:

BOOT_SEL2#	BOOT_SEL1#	BOOT_SEL0#	Boot Source
GND	GND	GND	Carrier SATA
GND	GND	Float	Carrier SD Card (SDIO/SD1)
GND	Float	GND	Carrier eMMC Flash (MMC/SD4)
GND	Float	Float	Unsupported
Float	GND	GND	Unsupported
Float	GND	Float	Unsupported
Float	Float	GND	Module eMMC Flash
Float	Float	Float	Unsupported

Pin Assignments

Parallel Display

SMARC pin	SMARC name	Device	Pin	Pin name	SMARC pin	SMARC name	Device	Pin	Pin name
S118	LCD_D[23]	iMX6	W24	DISP0_DAT23	S100	LCD_D[7]	iMX6	R24	DISP0_DAT7
S117	LCD_D[22]	iMX6	V24	DISP0_DAT22	S99	LCD_D[6]	iMX6	R23	DISP0_DAT6
S116	LCD_D[21]	iMX6	T20	DISP0_DAT21	S98	LCD_D[5]	iMX6	R25	DISP0_DAT5
S115	LCD_D[20]	iMX6	U22	DISP0_DAT20	S97	LCD_D[4]	iMX6	P20	DISP0_DAT4
S114	LCD_D[19]	iMX6	U23	DISP0_DAT19	S96	LCD_D[3]	iMX6	P21	DISP0_DAT3
S113	LCD_D[18]	iMX6	V25	DISP0_DAT18	S95	LCD_D[2]	iMX6	P23	DISP0_DAT2
S112	LCD_D[17]	iMX6	U24	DISP0_DAT17	S94	LCD_D[1]	iMX6	P22	DISP0_DAT1
S111	LCD_D[16]	iMX6	T21	DISP0_DAT16	S93	LCD_D[0]	iMX6	P24	DISP0_DAT0
S109	LCD_D[15]	iMX6	T22	DISP0_DAT15	S123	LCD_PCK	iMX6	N19	DI0_DISP_CLK
S108	LCD_D[14]	iMX6	U25	DISP0_DAT14	S120	LCD_DE	iMX6	N21	DI0_PIN15
S107	LCD_D[13]	iMX6	R20	DISP0_DAT13	S122	LCD_HS	iMX6	N25	DI0_PIN2
S106	LCD_D[12]	iMX6	T24	DISP0_DAT12	S121	LCD_VS	iMX6	N20	DI0_PIN3
S105	LCD_D[11]	iMX6	T23	DISP0_DAT11	S142	LVDS_DUAL_PCK	iMX6	P25	DI0_PIN4
S104	LCD_D[10]	iMX6	R21	DISP0_DAT10	S133	LCD_VDD_EN	iMX6	T5	GPIO00
S103	LCD_D[9]	iMX6	T25	DISP0_DAT9	S127	LCD_BKLT_EN	iMX6	T1	GPIO01
S102	LCD_D[8]	iMX6	R22	DISP0_DAT8	S141	LCD_BKLT_PWM	CPLD	B14	PR2A

LVDS Display

Two channels are available. The first is on the SMARC connector and the second is on CN2 connector. CN2 is a JST SM15B-SRSS-TB which mates to SHR-15V-S-B

SMARC pin	SMARC name	Device	Pin	Pin name	CN2 Pin	CN2 Name	Device	Pin	Pin name
S137	LVDS[3]+	iMX6	U2	LVDS0_DATA[3]+	1	LVDS1_TX0_N	iMX6	Y1	LVDS1_DATA[0]-
S138	LVDS[3]-	iMX6	U1	LVDS0_DATA[3]-	2	LVDS1_TX0_P	iMX6	Y2	LVDS1_DATA[0]+
S131	LVDS[2]+	iMX6	U4	LVDS0_DATA[2]+	4	LVDS1_TX1_N	iMX6	AA2	LVDS1_DATA[1]-
S132	LVDS[2]-	iMX6	U3	LVDS0_DATA[2]-	5	LVDS1_TX1_P	iMX6	AA1	LVDS1_DATA[1]+
S128	LVDS[1]+	iMX6	V2	LVDS0_DATA[1]+	7	LVDS1_TX2_N	iMX6	AB1	LVDS1_DATA[2]-
S129	LVDS[1]-	iMX6	V1	LVDS0_DATA[1]-	8	LVDS1_TX2_P	iMX6	AB2	LVDS1_DATA[2]+
S125	LVDS[0]+	iMX6	W2	LVDS0_DATA[0]+	13	LVDS1_TX3_N	iMX6	AA3	LVDS1_DATA[3]-
S126	LVDS[0]-	iMX6	W1	LVDS0_DATA[0]-	14	LVDS1_TX3_P	iMX6	AA4	LVDS1_DATA[3]+
S134	LVDS_CK+	iMX6	V3	LVDS0_CK+	10	LVDS1_CLK_N	iMX6	Y3	LVDS1_CK-
S135	LVDS_CK-	iMX6	V4	LVDS0_CK-	11	LVDS1_CLK_P	iMX6	Y4	LVDS1_CK+
					3,6,9,12	GND			

HDMI

SMARC pin	SMARC name	Device	Pin	Pin name
P92	HDMI_D2+	iMX6	K4	HDMI_D2+
P93	HDMI_D2-	iMX6	K3	HDMI_D2-
P95	HDMI_D1+	iMX6	J4	HDMI_D1+
P96	HDMI_D1-	iMX6	J3	HDMI_D1-
P98	HDMI_D0+	iMX6	K6	HDMI_D0+
P99	HDMI_D0-	iMX6	K5	HDMI_D0-
P101	HDMI_CK+	iMX6	J6	HDMI_CK+
P102	HDMI_CK-	iMX6	J5	HDMI_CK-
P104	HDMI_TX_HPD	iMX6	K1	HDMI_TX_HPD
P107	HDMI_CEC	iMX6	W4	KEY_ROW2

Camera Interfaces

Camera interfaces signals can be arranged in three different formats, depending on hardware and software configurations.

The possible configurations are as follows:

Configuration	HW option code	Notes
2 lane serial + 10 bit parallel	-S	
2x 8 bit parallel	-P	Pin multiplex selected through CPLD
1x16 bit parallel	-P	Pin multiplex selected through CPLD

Pins independent from configuration are the following:

SMARC pin	SMARC name	Device	Pin	Pin name	SMARC pin	SMARC name	Device	Pin	Pin name
S57	PCAM_ON_CSI0#	CPLD	F13	PR4C	S4	PCAM_PXL_CK0	iMX6	F21	EIM_DATA17
S58	PCAM_ON_CSI1#	CPLD	F14	PR4D	P1	PCAM_PXL_CK1	iMX6	P1	CSI0_PIXCLK
S2	PCAM_HSYNC	iMX6	M20	EIM_AD11	P6	PCAM_MCK	iMX6	P4	CSI0_HSYNC
S1	PCAM_VSYNC	iMX6	M24	EIM_AD12	S6	CAM_MCK	iMX6	P4	CSI0_HSYNC
P5	PCAM_DE	iMX6	M22	EIM_AD10					

Configuration for 2 lane serial + 10 bit parallel option is as follows:

SMARC pin	SMARC name	Device	Pin	Pin name	SMARC pin	SMARC name	Device	Pin	Pin name
S15	CSI0_D1-/ PCAM_D15	iMX6	D1	CSI_DATA1_N	P14	CSI1_D2-/ PCAM_D7	iMX6	F24	EIM_ADDR22
S14	CSI0_D1+ / PCAM_D14	iMX6	D2	CSI_DATA1_P	P13	CSI1_D2+ / PCAM_D6	iMX6	H23	EIM_ADDR21
S12	CSI0_D0-/ PCAM_D13	iMX6	E4	CSI_DATA0_N	P11	CSI1_D1-/ PCAM_D5	iMX6	H22	EIM_ADDR20
S11	CSI0_D0+ / PCAM_D12	iMX6	E3	CSI_DATA0_P	P10	CSI1_D1+ / PCAM_D4	iMX6	G25	EIM_ADDR19
S9	CSI0_CK-/ PCAM_D11	iMX6	F4	CSI_CLK0_N	P8	CSI1_D0-/ PCAM_D3	iMX6	J22	EIM_ADDR18
S8	CSI0_CK+ / PCAM_D10	iMX6	F3	CSI_CLK0_P	P7	CSI1_D0+ / PCAM_D2	iMX6	G24	EIM_ADDR17
P17	CSI1_D3-/ PCAM_D9	iMX6	F25	EIM_ADDR24	P4	CSI1_CK-/ PCAM_D1	iMX6	K21	EIM_EB0
P16	CSI1_D3+ / PCAM_D8	iMX6	J21	EIM_ADDR23	P3	CSI1_CK+ / PCAM_D0	iMX6	K23	EIM_EB1

Configuration for 2x 8 bit bit parallel option is as follows:

SMARC pin	SMARC name	Device	Pin	Pin name	SMARC pin	SMARC name	Device	Pin	Pin name
S15	CSI0_D1-/ PCAM_D15	iMX6	L6	CSI0_DATA19	P14	CSI1_D2-/ PCAM_D7	iMX6	F25	EIM_ADDR24
S14	CSI0_D1+ / PCAM_D14	iMX6	M6	CSI0_DATA18	P13	CSI1_D2+ / PCAM_D6	iMX6	J21	EIM_ADDR23
S12	CSI0_D0-/ PCAM_D13	iMX6	L3	CSI0_DATA17	P11	CSI1_D1-/ PCAM_D5	iMX6	F24	EIM_ADDR22
S11	CSI0_D0+ / PCAM_D12	iMX6	L4	CSI0_DATA16	P10	CSI1_D1+ / PCAM_D4	iMX6	H23	EIM_ADDR21
S9	CSI0_CK-/ PCAM_D11	iMX6	M5	CSI0_DATA15	P8	CSI1_D0-/ PCAM_D3	iMX6	H22	EIM_ADDR20
S8	CSI0_CK+ / PCAM_D10	iMX6	M4	CSI0_DATA14	P7	CSI1_D0+ / PCAM_D2	iMX6	G25	EIM_ADDR19
P17	CSI1_D3-/ PCAM_D9	iMX6	L1	CSI0_DATA13	P4	CSI1_CK-/ PCAM_D1	iMX6	J22	EIM_ADDR28
P16	CSI1_D3+ / PCAM_D8	iMX6	M2	CSI0_DATA12	P3	CSI1_CK+ / PCAM_D0	iMX6	G24	EIM_ADDR17

Configuration for 16 bit bit parallel option is as follows:

SMARC pin	SMARC name	Device	Pin	Pin name	SMARC pin	SMARC name	Device	Pin	Pin name
S15	CSI0_D1-/ PCAM_D15	iMX6	F25	EIM_ADDR24	P14	CSI1_D2-/ PCAM_D7	iMX6	K21	EIM_EB0
S14	CSI0_D1+ / PCAM_D14	iMX6	J21	EIM_ADDR23	P13	CSI1_D2+ / PCAM_D6	iMX6	K23	EIM_EB1
S12	CSI0_D0-/ PCAM_D13	iMX6	F24	EIM_ADDR22	P11	CSI1_D1-/ PCAM_D5	iMX6	L20	EIM_AD00
S11	CSI0_D0+ / PCAM_D12	iMX6	H23	EIM_ADDR21	P10	CSI1_D1+ / PCAM_D4	iMX6	LJ25	EIM_AD01
S9	CSI0_CK-/ PCAM_D11	iMX6	H22	EIM_ADDR20	P8	CSI1_D0-/ PCAM_D3	iMX6	L21	EIM_AD02
S8	CSI0_CK+ / PCAM_D10	iMX6	G25	EIM_ADDR19	P7	CSI1_D0+ / PCAM_D2	iMX6	K24	EIM_AD03

P17	CSI1_D3-/ PCAM_D9	iMX6	J22	EIM_ADDR18	P4	CSI1_CK-/ PCAM_D1	iMX6	L22	EIM_AD04
P16	CSI1_D3+ / PCAM_D8	iMX6	G24	EIM_ADDR17	P3	CSI1_CK+ / PCAM_D0	iMX6	L23	EIM_AD05

I2C

I2C channel Mapping is as follows:

SMARC Channel	iMX channel
PM	I2C3
CAM	I2C2
GP	I2C1
LCD	GPIO
HDMI	HDMI_TX_DDC

SMARC pin	SMARC name	Device	Pin	Pin name	SMARC pin	SMARC name	Device	Pin	Pin name
P122	I2C_PM_DAT	iMX6	R2	GPIO_16	S48	I2C_GP_CK	iMX6	N5	CSI_DATA09
P121	I2C_PM_CK	iMX6	R4	GPIO_05	S140	I2C_LCD_DAT	iMX6	P6	GPIO_18
S7	I2C_CAM_DAT	iMX6	T7	KEY_ROW3	S139	I2C_LCD_CK	iMX6	R1	GPIO_17
S5	I2C_CAM_CK	iMX6	E22	EIM_EB2	P106	HDMI_CTRL_DAT	iMX6	C25	EIM_DATA16
S49	I2C_GP_DAT	iMX6	N6	CSI_DATA08	P105	HDMI_CTRL_CK	iMX6	U5	KEY_COL3

SDIO

SDIO port is mapped on SD1 port on the iMX6

SMARC pin	SMARC name	Device	Pin	Pin name	SMARC pin	SMARC name	Device	Pin	Pin name
P39	SDIO_D[0]	iMX6	A21	SD1_DATA0	P36	SDIO_CK	iMX6	D20	SD1_CLK
P40	SDIO_D[1]	iMX6	C20	SD1_DATA1	P33	SDIO_WP	iMX6	A18	NAND_DATA00
P41	SDIO_D[2]	iMX6	E19	SD1_DATA2	P35	SDIO_CD#	iMX6	C17	NAND_DATA01
P42	SDIO_D[3]	iMX6	F18	SD1_DATA3	P37	SDIO_PWR_EN	iMX6	F16	NAND_DATA02
P34	SDIO_CMD	iMX6	B21	SD1_CMD					

MMC

MMC port is mapped on SD4 port of the iMX6.

SMARC pin	SMARC name	Device	Pin	Pin name	SMARC pin	SMARC name	Device	Pin	Pin name
S26	SDMMC_D[0]	iMX6	D18	SD4_DATA0	S32	SDMMC_D[6]	iMX6	B20	SD4_DATA6
S27	SDMMC_D[1]	iMX6	B19	SD4_DATA1	S33	SDMMC_D[7]	iMX6	D19	SD4_DATA7
S28	SDMMC_D[2]	iMX6	F17	SD4_DATA2	S36	SDMMC_CMD	iMX6	B17	SD4_CMD
S29	SDMMC_D[3]	iMX6	A20	SD4_DATA3	S35	SDMMC_CK	iMX6	E16	SD4_CLK
S30	SDMMC_D[4]	iMX6	E18	SD4_DATA4	S37	SDMMC_RST#	iMX6	A16	SD4_RESET
S31	SDMMC_D[5]	iMX6	C19	SD4_DATA5					

SPI

SMARC pin	SMARC name	Device	Pin	Pin name	SMARC pin	SMARC name	Device	Pin	Pin name
P43	SPI0_CS0#	iMX6	J19	EIM_D29	P54	SPI1_CS0#	iMX6	K20	EIM_RW
P31	SPI0_CS1#	iMX6	H19	EIM_ADDR25	P55	SPI1_CS1#	iMX6	K22	EIM_LBA
P44	SPI0_CK	iMX6	H20	EIM_D21	P56	SPI1_CK	iMX6	H24	EIM_CS0
P45	SPI0_DIN	iMX6	E23	EIM_D22	P57	SPI1_DIN	iMX6	J24	EIM_OE
P46	SPI0_DO	iMX6	G23	EIM_D28	P58	SPI1_DO	iMX6	J23	EIM_CS1

I2S

I2S channel Mapping is as follows:

SMARC Channel	iMX channel
I2S0	AUD3
I2S1	AUD4
I2S2	ESAI

SMARC pin	SMARC name	Device	Pin	Pin name	SMARC pin	SMARC name	Device	Pin	Pin name
S39	I2S0_LRCK	iMX6	N4	CSI0_DAT6	S45	I2S1_SDIN	iMX6	A22	SD2_DATA0
S40	I2S0_SDOUT	iMX6	P2	CSI0_DAT5	S46	I2S1_CK	iMX6	B22 F19	SD2_DATA3 SD2_CMD
S41	I2S0_SDIN	iMX6	N3	CSI0_DAT7	S50	I2S2_LRCK	iMX6	W22	ENET_RX_DATA1
S42	I2S0_CK	iMX6	N1	CSI0_DAT4	S51	I2S2_SDOUT	iMX6	V21	ENET_TX_EN
S38	AUDIO_MCK	iMX6	R7	GPIO03	S52	I2S2_SDIN	iMX6	U20	ENET_TX_DATA0
S43	I2S1_LRCK	iMX6	E20 C21	SD2_DATA1 SD2_CLK	S53	I2S2_CK	iMX6	U21	ENET_CRS_DV
S44	I2S1_SDOUT	iMX6	A23	SD2_DATA2					

Note

Highlighted pins are 1.8V volts only even when VIO is set to 3.3V

Note

I2S1 has a double connection for LRCK and CK so that it is possible, for this port, to be configured either as master or slave. Depending on which configuration is used the unused pins are to be configured as tri state inputs so that they don't interfere with the functionality of the interface

SPDIF

SMARC pin	SMARC name	Device	Pin	Pin name	SMARC pin	SMARC name	Device	Pin	Pin name
S60	SPDIF_IN	iMX6	W23	ENET_RX_ER	S59	SPDIF_OUT	iMX6	W21	ENET_RXD0

Note

SPDIF pins are 1.8V volts only even when VIO is set to 3.3V

UART

UART channel Mapping is as follows:

SMARC Channel	iMX channel
SER0	UART1
SER1	UART2
SER2	UART3
SER3	UART4

SMARC pin	SMARC name	Device	Pin	Pin name	SMARC pin	SMARC name	Device	Pin	Pin name
P132	SER0_CTS#	iMX6	G20	EIM_D20	P139	SER2_CTS#	iMX6	F23	EIM_EB3
P131	SER0_RTS#	iMX6	G21	EIM_D19	P138	SER2_RTS#	iMX6	D25	EIM_DATA23
P130	SER0_RX	iMX6	M3	CSI0_DAT11	P137	SER2_RX	iMX6	G22	EIM_DATA25
P129	SER0_TX	iMX6	M1	CSI0_DAT10	P136	SER2_TX	iMX6	F22	EIM_DATA24
P135	SER1_RX	iMX6	E25	EIM_D27	P141	SER3_RX	iMX6	V6	KEY_ROW0
P134	SER1_TX	iMX6	E24	EIM_D26	P140	SER3_TX	iMX6	W5	KEY_COL0

CAN

SMARC pin	SMARC name	Device	Pin	Pin name	SMARC pin	SMARC name	Device	Pin	Pin name
P144	CAN0_RX	iMX6	R5	GPIO_08	P146	CAN1_RX	iMX6	V5	KEY_ROW4
P143	CAN0_TX	iMX6	R3	GPIO_07	P145	CAN1_TX	iMX6	T6	KEY_COL4

USB

USB channel mapping

SMARC Channel	iMX channel
USB0	USB_OTG
USB1/2	USB_H1

SMARC pin	SMARC name	Device	Pin	Pin name	SMARC pin	SMARC name	Device	Pin	Pin name
P60	USB0+	iMX6	A6	USB_OTG_DP	P65	USB1+	USB2512	2	USBDP_DN1
P61	USB0-	iMX6	B6	USB_OTG_DP	P66	USB1-	USB2512	1	USBDM_DN1
P62	USB0_EN_OC#	iMX6	P5 D16	GPIO_19 NANDF_CS3	P67	USB1_EN_OC#	USB2512	12	BC_EN1
P63	USB0_VBUS_DET	iMX6	E9	USB_OTG_VBUS	P69	USB2+	USB2512	4	USBDP_DN2
P64	USB0_OTG_ID	iMX6	T4	GPIO01	P70	USB2-	USB2512	3	USBDM_DN2
					P71	USB2_EN_OC#	USB2512	16	BC_EN2

Note

USB0_EN_OC# is connected to two pins because this way iMX can separate the enable output (connected on pin GPIO_19) from the overcurrent input (connected on pin NANDF_CS3). Board contains an open drain driver and for the enable pin and voltage translator for the overcurrent pin so no particular setup is required for the pins other than their direction.

Hub additional connections

Hub Pin	Hub pin Name	iMX pin	iMX pin Name	Notes
26	RESET_N	D15	SD3_RST	
24	SCL	R2	GPIO_16	I2C_PM bus with 3.3V translation
22	SDA	R4	GPIO_05	

Ethernet

SMARC pin	SMARC name	Device	Pin	Pin name	SMARC pin	SMARC name	Device	Pin	Pin name
P28	GBE_CTREF			To ground via 100nF cap	P26	GBE_MDI1-	KSZ9031	6	TXRXM_B
P25	GBE_LINK_ACT#	CPLD	D12	PR2D	P27	GBE_MDI1+	KSZ9031	5	TXRXP_B
P21	GBE_LINK100#	CPLD	C13	PR2B	P23	GBE_MDI2-	KSZ9031	8	TXRXM_C
P22	GBE_LINK1000#	CPLD	C14	PR2C	P24	GBE_MDI2+	KSZ9031	7	TXRXP_C
P29	GBE_MDI0-	KSZ9031	3	TXRXM_A	P19	GBE_MDI3-	KSZ9031	11	TXRXM_D
P30	GBE_MDI0+	KSZ9031	2	TXRXP_A	P20	GBE_MDI3+	KSZ9031	10	TXRXP_D

Phy additional connections

Phy Pin	Phy pin Name	iMX pin	iMX pin Name	Notes
36	MDC	V20	ENET_MDC	
37	MDIO	V23	ENET_MDIO	
32	RXD0/MODE0	C24	RGMII_RD0	Care must be taken when resetting chip as these pins set up the chip mode
31	RXD1/MODE1	B23	RGMII_RD1	
28	RXD2/MODE2	B24	RGMII_RD2	
27	RXD3/MODE3	D23	RGMII_RD3	
33	RX_DV	D22	RGMII_RX_CTL	
35	RX_CLK/PHYAD2	B25	RGMII_RXC	Care must be taken when resetting chip as this pin selects PHY address
19	TXD0	C22	RGMII_TD0	
20	TXD1	F20	RGMII_TD1	
21	TXD2	E21	RGMII_TD2	
22	TXD3	A24	RGMII_TD3	
25	TX_EN	C23	RGMII_TX_CTL	
24	GTX_CLK	D21	RGMII_TXC	
42	RESET	R6	GPIO04	
41	CLK125	V22	ENET_REF_CLK	

PCIe

SMARC pin	SMARC name	Device	Pin	Pin name	SMARC pin	SMARC name	Device	Pin	Pin name
P78	PCIE_A_CKREQ#	iMX6	D17	NAND_DATA03	P75	PCIE_A_RST#	iMX6	T3	GPIO06
P74	PCIE_A_PRSNT#	iMX6	A19	NAND_DATA04	P87	PCIE_A_RX-	iMX6	B1	PCIE_RX_N
S146	PCIE_WAKE #	iMX6	B18	NAND_DATA05	P86	PCIE_A_RX+	iMX6	B2	PCIE_RX_P
P84	PCIE_A_REFCK-	iMX6	C7	CLK1_N	P90	PCIE_A_TX-	iMX6	A3	PCIE_TX_N
P83	PCIE_A_REFCK+	iMX6	D7	CLK1_P	P89	PCIE_A_TX+	iMX6	B3	PCIE_TX_P
P78	PCIE_A_CKREQ#	iMX6	D17	NAND_DATA03	P75	PCIE_A_RST#	iMX6	T3	GPIO06

SATA

SMARC pin	SMARC name	Device	Pin	Pin name	SMARC pin	SMARC name	Device	Pin	Pin name
P52	SATA_RX-	iMX6	A14	SATA_PHY_RX_N	P49	SATA_TX-	iMX6	B12	SATA_PHY_TX_N
P51	SATA_RX+	iMX6	B14	SATA_PHY_RX_P	P48	SATA_TX+	iMX6	A12	SATA_PHY_TX_P

Note

Smarc PIN S54 (SATA_ACT#) is left unconnected

Watchdog

SMARC pin	SMARC name	Device	Pin	Pin name
S145	WDT_TIME_OUT#	iMX6	T2	WDOG1_B

GPIO

SMARC pin	SMARC name	Device	Pin	Pin name	SMARC pin	SMARC name	Device	Pin	Pin name
P108	GPIO0 / CAM0_PWR#	iMX6	L20	EIM_DA00	P114	GPIO6 / TACHIN	iMX6	L24	EIM_DA08
P109	GPIO1 / CAM1_PWR#	iMX6	J25	EIM_DA01	P115	GPIO7 / PCAM_FLD	iMX6	M21	EIM_DA09
P110	GPIO2 / CAM0_RST#	iMX6	L21	EIM_DA02	P116	GPIO8 / CAN0_ERR #	iMX6	M23	EIM_DA13
P111	GPIO3 / CAM1_RST#	iMX6	K24	EIM_DA03	P117	GPIO9 / CAN1_ERR #	iMX6	N23	EIM_DA14
P112	GPIO4 / HDA_RST#	iMX6	K25	EIM_DA06	P118	GPIO10	iMX6	N24	EIM_DA15
P113	GPIO5 / PWM_OUT	iMX6	L25	EIM_DA07	P119	GPIO11	iMX6	M25	EIM_WAIT

In addition to iMX6, GPIO pins are also connected to CPLD:

SMARC pin	SMARC name	Device	Pin	Pin name	SMARC pin	SMARC name	Device	Pin	Pin name
P108	GPIO0 / CAM0_PWR#	CPLD	J2	PL8B	P114	GPIO6 / TACHIN	CPLD	M8	PB15B
P109	GPIO1 / CAM1_PWR#	CPLD	M7	PB11A	P115	GPIO7 / PCAM_FLD	CPLD	J1	PL8A
P110	GPIO2 / CAM0_RST#	CPLD	L3	PL10B	P116	GPIO8 / CAN0_ERR #	CPLD	P2	PB4A
P111	GPIO3 / CAM1_RST#	CPLD	P4	PB6B	P117	GPIO9 / CAN1_ERR #	CPLD	N3	PB6A
P112	GPIO4 / HDA_RST#	CPLD	M9	PB18A	P118	GPIO10	CPLD	G3	PL5A
P113	GPIO5 / PWM_OUT	CPLD	P3	PB4C	P119	GPIO11	CPLD	N4	PB6D

Management

SMARC pin	SMARC name	Device	Pin	Pin name	SMARC pin	SMARC name	Device	Pin	Pin name
S150	VIN_PWR_BAD#	CPLD	A2	PT9A	S149	SLEEP#	iMX6	U6	KEY_ROW1
S154	CARRIER_PWR_ON	CPLD	C11	PT16B	S148	LID#	iMX6	W6	KEY_COL2
S153	CARRIER_STBY#	iMX6	U7	KEY_COL1	S156	BATLOW#	iMX6	P3	CSI0_DATA_EN
P126	RESET_OUT#	CPLD	B5	PT11A	S151	CHARGING#	iMX6	N2	CSI0_VSYNC
P127	RESET_IN#	CPLD	A3	PT10A	S152	CHARGER_PRSNT#	iMX6	E15	NAND_WP_B
P128	POWER_BTN#	CPLD	B13	PT17C	S157	TEST#	iMX6	C16	NAND_CS1_B
-	INT_RTC	iMX6	A17	NANDF_CS2	S155	FORCE_RECOV#	CPLD	A9	PT15B
P123	BOOT_SEL0	CPLD	B9	PT15C	P124	BOOT_SEL1	CPLD	C12	PT17A
P125	BOOT_SEL2	CPLD	A10	PT16A					

Note

Highlighted pins are 1.8V volts only even when VIO is set to 3.3V

AFB

SMARC pin	SMARC name	Device	Pin	Pin name	SMARC pin	SMARC name	Device	Pin	Pin name
S17	AFB0_OUT	iMX6	N22	EIM_BCLK	S22	AFB5_IN	iMX6	E17	NAND_DATA06
S18	AFB1_OUT	iMX6	H25	EIM_ADDR16	S23	AFB6_PTIO	iMX6	C18	NAND_DATA07
S19	AFB2_OUT	iMX6	D24	EIM_DATA18	S24	AFB7_PTIO	iMX6	C15	NAND_CLE
S20	AFB3_IN	iMX6	J20	EIM_DATA30	S55	AFB8_PTIO	iMX6	B16	NAND_READY
S21	AFB4_IN	iMX6	H21	EIM_DATA31	S56	AFB9_PTIO	iMX6	F15	NAND_CS0_B

The possible hardware configurations for the differential pin part are as follows:

Configuration	HW option code	Notes
MLB+MIPI CSI	-M	MLB+MIPI CSI
MIPI DSI	-P	MIPI DSI + MIPI CSI

Differential AFB configuration for MLB+MIPI CSI

SMARC pin	SMARC name	Device	Pin	Pin name	SMARC pin	SMARC name	Device	Pin	Pin name
S63	AFB_DIFF0-	iMX6	E1	CSI_DATA2_N	S68	AFB_DIFF2+	iMX6	B11	MLB_CP
S62	AFB_DIFF0+	iMX6	E2	CSI_DATA2_P	S72	AFB_DIFF3-	iMX6	B10	MLB_DN
S66	AFB_DIFF1-	iMX6	F2	CSI_DATA3_N	S71	AFB_DIFF3+	iMX6	A10	MLB_DP
S65	AFB_DIFF1+	iMX6	F1	CSI_DATA3_P	S75	AFB_DIFF4-	iMX6	A9	MLB_SN
S69	AFB_DIFF2-	iMX6	A11	MLB_CN	S74	AFB_DIFF4+	iMX6	B9	MLB_SP

Differential AFB configuration for MIPI DSI + MIPI CSI

SMARC pin	SMARC name	Device	Pin	Pin name	SMARC pin	SMARC name	Device	Pin	Pin name
S63	AFB_DIFF0-	iMX6	E1	CSI_DATA2_N	S68	AFB_DIFF2+	iMX6	H4	DSI_CLK0_P
S62	AFB_DIFF0+	iMX6	E2	CSI_DATA2_P	S72	AFB_DIFF3-	iMX6	G2	DSI_DATA0_N
S66	AFB_DIFF1-	iMX6	F2	CSI_DATA3_N	S71	AFB_DIFF3+	iMX6	G1	DSI_DATA0_P
S65	AFB_DIFF1+	iMX6	F1	CSI_DATA3_P	S75	AFB_DIFF4-	iMX6	H2	DSI_DATA1_N
S69	AFB_DIFF2-	iMX6	H3	DSI_CLK0_N	S74	AFB_DIFF4+	iMX6	H1	DSI_DATA1_P

IMX6 JTAG

Connector on board is a JST SM10B-SRSS-TB which mates to SHR-10V-S-B connector housing.

CN3 pin	Name	Device	Pin	Pin name
1	VDD_IO			
2	TRST	iMX6	C2	JTAG_TRST
3	TMS	iMX6	C3	JTAG_TMS
4	TDO	iMX6	G6	JTAG_TDO
5	TDI	iMX6	G5	JTAG_TDI
6	TCK	iMX6	H5	JTAG_TCK
7	10K to GND			
8,10	GND			
9	JTAG_MOD	iMX6	H6	JTAG_MOD

CPLD JTAG

Connector on board is a JST SM10B-SRSS-TB which mates to SHR-10V-S-B connector housing.

CN3 pin	Name	Device	Pin	Pin name
1	VCC_BANK_0	CPLD	A8, B10, C5	VCCIO0
2	TP4			
3	TMS	CPLD	A6	PT11D_TMS
4	TDO	CPLD	A4	PT10C_TDO
5	TDI	CPLD	B4	PT10D_TDI
6	TCK	CPLD	B6	PT11C_TCK
7	10K to GND			
8,10	GND			
9	TP5			

CPLD pinout

CPLD pin	Signal name	Pin name	Description	SMARC pin	SMARC name	Device	Description
M4	bEIM_A16	PB6C	EIM pins used for bootstrapping iMX6.	J12	iPCAM_DATA[2]	PR8A	Parallel camera data pins from SMARC
M13	bEIM_A17	PR9D		L14	iPCAM_DATA[3]	PR9C	
P12	bEIM_A18	PB20B		K13	iPCAM_DATA[4]	PR9A	
M5	bEIM_A19	PB9D		H12	iPCAM_DATA[5]	PR5D	
K2	bEIM_A20	PL8D		J14	iPCAM_DATA[6]	PR8B	
P11	bEIM_A21	PB18D		J13	iPCAM_DATA[7]	PR8C	
N14	bEIM_A22	PR10D		K14	iPCAM_DATA[8]	PR9B	
K1	bEIM_A23	PL9A		G14	iPCAM_DATA[9]	PR5B	
N13	bEIM_A24	PR10C		F12	iPCAM_DE	PR4B	Parallel Camera DE from SMARC
J2	bEIM_DA0	PL8B		N8	iPCAM_HSYNC	PB11B	Parallel Camera HSYNC from SMARC
M7	bEIM_DA1	PB11A		G13	iPCAM_PXL_CLK0	PR5C	Parallel clocks from SMARC
M3	bEIM_DA10	PB4D		N6	iPCAM_PXL_CLK1	PB9A	
H2	bEIM_DA11	PL5B		N10	iPCAM_VSYNC	PB18B	Parallel Camera VSYNC from SMARC
P9	bEIM_DA12	PB15C		B12	iPGOOD	PT16D	Power good from PMIC
P2	bEIM_DA13	PB4A		A7	iPM_SCL	PT12A	Power Management I2C
N3	bEIM_DA14	PB6A		A3	iRESET_IN_	PT10A	Reset input from SMARC
G3	bEIM_DA15	PL5A		B9	iSMARC_BOOT0	PT15C	Boot mode selection from SMARC
L3	bEIM_DA2	PL10B		C12	iSMARC_BOOT1	PT17A	
P4	bEIM_DA3	PB6B		A10	iSMARC_BOOT2	PT16A	
H1	bEIM_DA4	PL5C		A2	iVIN_POWER_BAD_	PT9A	VIN_POWER_BAD from SMARC
N2	bEIM_DA5	PB4B		B3	oBOOT_MODE0	PT9B	Boot mode output to iMX
M9	bEIM_DA6	PB18A		C4	oBOOT_MODE1	PT10B	
P3	bEIM_DA7	PB4C		G12	oCAM_MCK	PR5A	Parallel camera MCK output to SMARC
M8	bEIM_DA8	PB15B		M2	oEIM_DATA17	PL10D	EIM pins used for bootstrapping iMX6.
J1	bEIM_DA9	PL8A		A11	oEN_B1	PT16C	Enable pins for PMIC regulators
J3	bEIM_EB0	PL8C		C6	oEN_B3	PT11B	
P13	bEIM_EB1	PB20D		C9	oEN_B4	PT15A	
M10	bEIM_EB2	PB18C		A13	oEN_L2	PT17D	
K3	bEIM_EB3	PL9B		E3	oEN_L4	PL4A	
N12	bEIM_LBA	PB20C		C10	oEN_VIO	PT15D	
M1	bEIM_RW	PL10C		C14	oGBE_LINK1000_	PR2C	Ethernet LED outputs to SMARC
N4	bEIM_WAIT	PB6D		C13	oGBE_LINK100_	PR2B	
B7	bPM_SDA	PT12B	Power Management I2C	D12	oGBE_LINK_ACT_	PR2D	
B2	bPOR	PL2B	Power on reset	C3	oGPIO_5	PL2D	CSI pixel clock 1 to iMX6
C8	bSCL	PT12C	I2C Reprogramming interface	C1	oIPU1_CSI0_DATA[12]	PL2C	Parallel camera pins to iMX
B8	bSDA	PT12D		B1	oIPU1_CSI0_DATA[13]	PL2A	
C2	iCSI_HSYNC	PL3A	Parallel Camera MCK from iMX	D1	oIPU1_CSI0_DATA[14]	PL3B	
A9	iFORCE_RECOV_	PT15B	From SMARC	H3	oIPU1_CSI0_DATA[15]	PL5D	
M12	iLED1	PR10A	LED inputs from ETH phy	E1	oIPU1_CSI0_DATA[16]	PL3C	
M14	iLED2	PR10B		E2	oIPU1_CSI0_DATA[17]	PL3D	
E14	iPCAM_DATA[0]	PR3B	Parallel camera data pins from SMARC	F2	oIPU1_CSI0_DATA[18]	PL4B	
P7	iPCAM_DATA[10]	PB11C		F1	oIPU1_CSI0_DATA[19]	PL4C	
N9	iPCAM_DATA[11]	PB15D		M11	oPCAM_MCK	PB20A	MCK to SMARC
P6	iPCAM_DATA[12]	PB9B		F13	oPCAM_ON_CSI0n	PR4C	Parallel camera present flags to SMARC
N5	iPCAM_DATA[13]	PB9C		F14	oPCAM_ON_CSI1n	PR4D	
N7	iPCAM_DATA[14]	PB11D		B14	oPWM	PR2A	PWM output for LCD backlight
P8	iPCAM_DATA[15]	PB15A		C11	oPWR_ON	PT16B	CARRIER_PWR_ON to SMARC
E13	iPCAM_DATA[1]	PR4A		B5	oRESET_OUT_	PT11A	Reset output to SMARC

3 Software reference

Introduction

EZ-ROJ-1 SMARC BSP is provided through an open source Git repository located on GitHub. Since the repository is constantly updated, references to specific versions of u-boot, kernel and compiled packages may change, so please always check on line to verify the last supported version for each deliverable.

Yocto

For those not familiar with Yocto, this is a build environment used to automate with very simple commands the build process of bootloader, kernel and file system image.
Yocto, derived from the OpenEmbedded project, provides a large set of *recipes* which basically are build and deployment scripts for each pagkage. Recipes are arranged in a layered structure which allows a given recipe to be modified by an higher priority layer without impacting the original recipe, thus allowing separate archival of the “base recipe” and the customizations for a given board or combination of packages.
In order to set up the build environment, the **repo** download script is used to download the various layer source files from different repositories into a common work directory.
All the ROJ customizations and BSP for the EZ-ROJ-1 SMARC board are located under a custom layer called meta-roj, which is maintained by ROJ and is based on the Freescale Community BSP.
For more information on Yocto please check (3)

Build environment Setup

The first step to set up the build environment is to set up a linux (virtual) machine with a supported OS. As of writing the preferred OS is Ubuntu 12.04.5 LTS, however migration to more recent versions should not generally create many issues. Please check (4) to verify the list of supported OS and the required steps to ensure the required packages are installed prior to moving to the next steps.

Once the machine is set up download the **repo** script and add it to the search path:

```
$: mkdir ~/bin
$: curl http://commondatastorage.googleapis.com/git-repo-
downloads/repo > ~/bin/repo
$: chmod a+x ~/bin/repo
$: PATH=${PATH}:~/bin
```

Now download the BSP source:

```
$: mkdir fsl-community-bsp
$: cd fsl-community-bsp
$: repo init -u https://github.com/ROJ-ITALY/roj-community-bsp-
platform -b dizzy
$: repo sync
```

This will create the *source* directory containing all the metadata you need to start working.

To set up a working environment for the EZ-ROJ-1 board enter the following command:

```
$: MACHINE=mx6qroj source setup-environment build
```

This will create a directory called build with the necessary configuration files for the mx6qroj machine (EZ-ROJ-1).
In case you're building for the solo/duallite version of the board, please replace mx6qroj with mx6sroj.
Note that this will also be required each time you start a new console session (eg. After a reboot) when the build directory has already been created, in order to set up the necessary environment variables; running again the command won't overwrite any existing setting.

Now to create a file system (and in turn the bootloader and the kernel) type the following:

```
$: bitbake core-image-minimal
```

Burning an image on the eMMC

Assuming the board has booted from a different source and you're not using MfgTools, you can manually write the files to eMMC with the following commands:

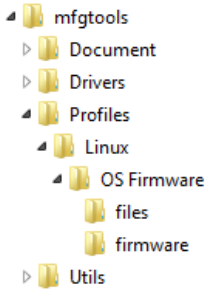
```
$: echo 0 > /sys/block/mmcblk1boot0/force_ro
$: dd if=u-boot.imx of=/dev/mmcblk1boot0 bs=512 seek=2
$: dd if=u-boot.imx of=/dev/mmcblk1 bs=512 seek=2
$: echo 1 > /sys/block/mmcblk1boot0/force_ro
$:
$: mount -t vfat /dev/block/mmcblk1p1 /mnt
$: cp zImage /mnt
$: cp zImage-imx6q-roj.dtb /mnt
$: umount /mnt
$:
$: mkfs.ext2 -E nodiscard /dev/block/mmcblk1p2
$: mount /dev/block/mmcblk1p2
$: tar xvjf core-image-minimal-mx6qroj.tar.bz2 /mnt
$: umount /mnt
$: sync
```

u-boot customizations

u-boot has been customized to automatically select the proper source for kernel and rootfs according to boot modes. At the moment of writing, booting from eMMC, carrier SD and carrier eMMC is supported and no additional change is required other than selecting SMARC boot select pins as required.

Using MFGTool

Although the board comes preloaded with a bootable image, it is possible to use Freescale MFGTool to update the whole system with a single, easy to use, automatic procedure.
In order to use MFGTool It's necessary to download the latest MFGTool software from ROJ's repositories.



As shown in the picture on the side, under the "Profiles/Linux/OS Firmware" path there are two directories: files and firmware.
The firmware directory contains the image files used by mfgtools to initialize the board and are used to perform board programming. These files can be generated from yocto using the u-boot-imx-mfgtool, linux-imx-mfgtool and fsl-image-mfgtool-initramfs respectively to build u-boot, linux, dtb files and root file system images.
Note that MFGTool automatically selects the u-boot and dtb files to load based on vid/pid detected so it's important to make sure the version for the target being updated has been updated.
In any case, rebuilding from yocto the MFGTool images is not usually required so it is recommended to use the ones provided in the ROJ MFGTool release package.

The files directory contains the files being copied to the board eMMC, hence these are the files that need to be replaced with the custom ones built with yocto with the "normal" recipes.
Note that here as well MFGTool will select u-boot and dtb files matching the version of the board so pay attention to updating the right files.

File naming and update procedure are set up in the ucl2.xml file present under the "Profiles/Linux/OS Firmware" path and can be easily customized; please refer to MFGTool documentation, present in the "Document" path for more information.

Note that in order to use MFGTools you have to make sure the FORCE_RECOV# pin (S155) is tied low and USB0 pins (including ID pin) are connected to a PC.

4 CPLD reference

CPLD functions

CPLD is used to handle power management, bootstrapping and multiplexing of camera interface.
In addition to several pins being shared with the CPLD, iMX6 also can control CPLD via the PM I^2C bus through which registers can be implemented to perform custom functions.

Reset management

CPLD implements a simple state machine along with a 1 msec timer used for power sequencing.
At power up the CPLD waits for VIN_POWER_BAD# pin to go high and then starts the power up sequence enabling progressively all the power rails with the proper sequence. When all rails are enabled CARRIER_PWR_ON# is asserted and CPU is taken out of reset.
It is optionally possible to hold the startup until the POWER_BTN# signal is asserted in cases where the board is not supposed to startup autonomously.
CPLD drives/receives directly the iMX POR, RESET_IN# and RESET_OUT# so it is possible to handle CPU and carrier resets separately and any custom configuration.

Bootstrap pins configuration

When CPLD resets or detects iMX has been reset, the bootstrap pins will be driven according to the configuration specified by the SMARC BOOT_SEL pins.
Although the standard SMARC boot modes have been implemented it's possible to select any of the iMX6 boot modes

Ethernet LEDs

Since KSZ9031 has LED configuration that differs from the SMARC standard, a small logic has been developed to convert between what the PHY outputs and what SMARC requires.

I2C slave

The CPLD has 2 I2C slave devices. The first one is implemented in hardware and can be used to reprogram the CPLD. This function is actually not supported in normal usage as when the CPLD is being reprogrammed its pins are floated, thus removing power to the module.
The second one is used for register access through an ip core that exports address/data bus and read/write signals.
The I2C slave is connected to several peripherals implemented in the CPLD as follows:

Register Address	Register Name	Bits	Description
0	Control	1:0	Select IPU configuration as follows: 00 – 10 bit interface on IPU2-CSI1 01 – 2x 8 bit interfaces on IPU1 & IPU2 10 – 1x 16 bit interface on IPU2-CSI1 11 - INVALID
		3:2	Direct control of SMARC PCAM_ON_CSI[1:0] pins
		4	Writing 1 to this bit forces RESET_OUT# to go low
1	PWM1	7:0	Selects the duty cycle of the PWM1 pin (used to drive LCD backlight signal)
7	STATUS	0	Reads back the status of VIN_POWER_BAD# pin

5 Warranty Policy

Warranty Period

ROJ products are entitled to a 2 years complete and prompt global warranty service. Product defect in materials and workmanship, are covered from the date of shipment.

Exclusions from Warranty:

The product is excluded from warranty if

– The product has been found to be defective after expiry of the warranty period.
– The product has been misused, abused, or subjected to unauthorized disassembly/modification; placed in an unsuitable physical or operating environment; improperly maintained by the customer; or failure caused which ROJ is not responsible whether by accident or other cause. Such conditions will be determined by ROJ at its sole unfettered discretion.
– The product is damaged beyond repair due to a natural disaster such as a lighting strike, flood, earthquake, etc.
– Product updates/upgrades and tests upon the request of customers who are without warranty.

Obtaining an RMA Number:

All returns from customers must be authorized with an ROJ RMA (Return Merchandise Authorization) number. Any returns of defective units or parts without valid RMA numbers will not be accepted; they will be returned to the customer at the customer's cost without prior notice.

To obtain the RMA number contact your sales or write to info@roj.com