Openframe R.TOUCH USR Manual 1.2.4

***** REV 1.2.4 *****

DATE	REVISION	CHANGE DESCRIPTION
12/04/12	1.0.0	Release
16/04/12	1.0.1	General enhancement
24/05/12	1.0.2	Added chapters about updated mode
26/07/12	1.0.3	Changed ftp address
17/09/12	1.0.4	General enhancement
19/10/12	1.0.5	Added bootargs setup using GEA6428
29/10/12	1.0.6	Added Ordering Code and added current consumption
21/12/12	1.0.7	Added information about GEAM6428 on Expansion connector
19/04/13	1.0.8	Added bootargs setup using i.Core6X and General enhancement
28/05/13	1.0.9	Modified the Expansion connector chapter
19/07/13	1.1.0	Added "How to start" chapter
09/10/13	1.1.1	Removed obsolete information about software patch and support address
15/11/13	1.1.2	Added chapter on setting modules' kernel options
30/01/14	1.1.3	Added bootargs update
04/02/14	1.1.4	Modified the bootargs update
25/02/14	1.1.5	Added patch information settings; added the environmental variable settings
04/04/14	1.1.6	Added bootargs for UBI fs
04/04/14	1.1.7	Added Temperature informations
29/08/14	1.1.8	Added Chimera module's informations
24/10/14	1.1.9	Added settings for Yocto system
16/04/15	1.2.0	Added Bootloader's settings for Yocto system
22/12/15	1.2.1	Updated ordering code
02/05/16	1.2.2	Added carrier mechanical specifications
18/07/16	1.2.3	General enhancements
03/05/17	1.2.4	Added "Product Compliance" chapter

Summary

1. Cable Map Overview....3
2. How to start....4
3. Carrier Board Specifications....5
   3.1 Mechanical informations....5
   3.2 Micro SD Connections....6
   3.3 RS 485 RS 232 & CAN Bus Connections....7
   3.4 USB Connections....9
   3.5 Ethernet Connections....10
   3.6 Power Supply Connections....11
   3.7 Current consumption....12
   3.8 Linux or Windows CE Console Debug Connections....13
   3.9 Expansion Connector....15
4. Compiling options of the Modules....17
   4.1 Using GEAM6425....17
   4.2 Using GEAM6428....17
   4.3 Using i.CoreM53....17
   4.4 Using i.CoreM6x....17
5. Bootargs Setup....18
   5.1 Using GEA6425 modules....18
      5.1.1 Using JFFS2 filesystem....18
      5.1.2 Using UBI filesystem....18
   5.2 Using GEA6428 modules....19
   5.3 Using i.CoreM53 modules....20
   5.4 Using i.CoreM6X modules....20
      5.4.1 U-boot setup using LTIB....20
      5.4.2 U-boot setup using YOCTO....21
      5.4.3 U-boot setup using LTIB version previous 1.09....22
   5.5 Using Chimera modules....24
6. Ordering Information....25
7. Technical support....26
   7.1 Ftp account structure....26
   7.2 How to upgrading your BSP using patch....26
      7.2.1 Structure of the patch folder....26
      7.2.2 Patch structure....27
      7.2.3 How to apply the patch....27
   7.3 Product Compliance....29
8. On-line Support....30
   8.1 Support....30
   8.2 Disclaimer....30

1. Cable Map Overview

This document is an overview about cable connecting map of Open-Frame system.
It describes the connector and the interface unit available to user, and it specifies the electrical characteristic of signals.

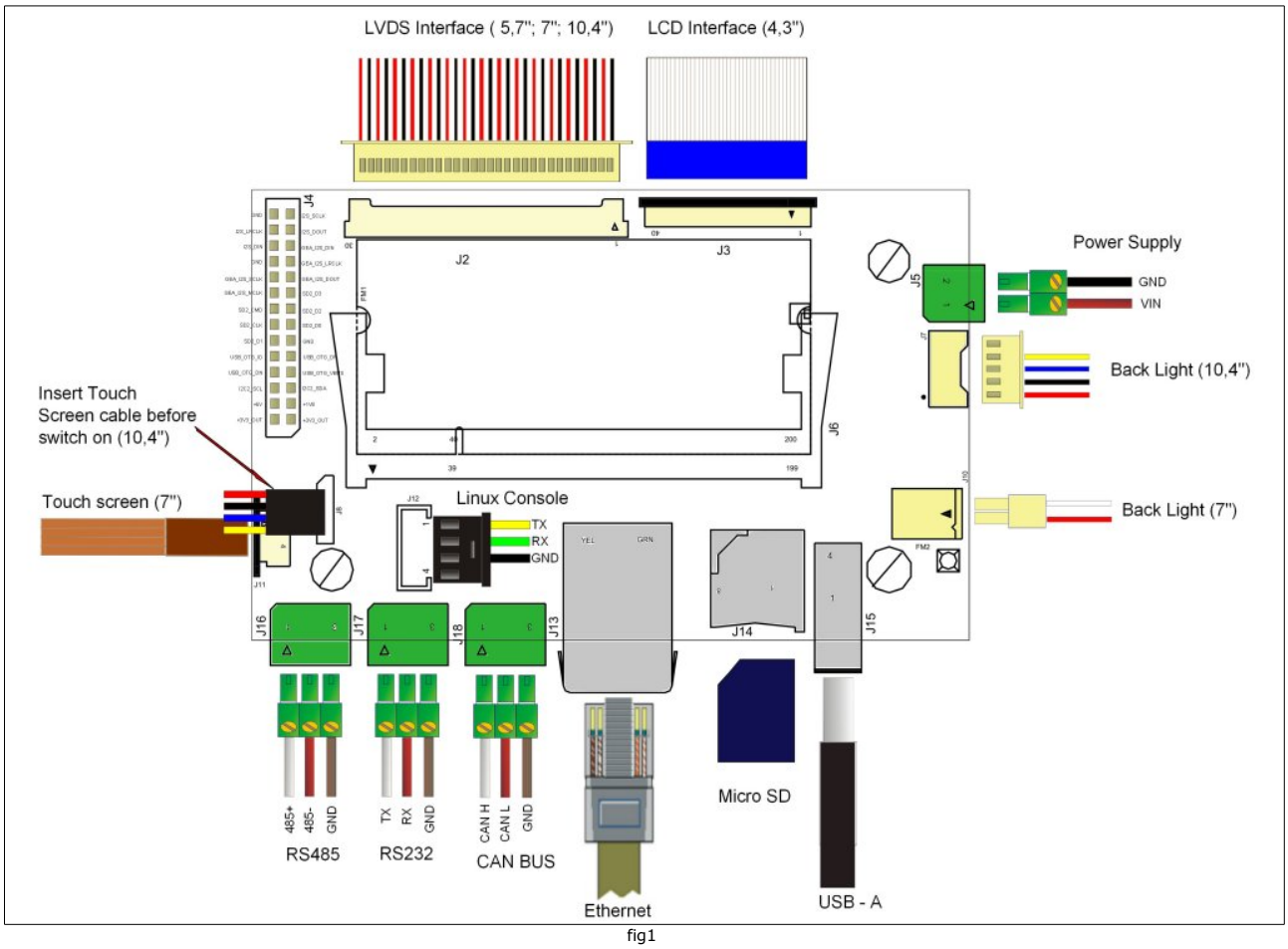


fig1

The figure1 shows the whole wiring map and its join connectors.
This document will try to analyse all type of used connections to help the the user's start up.

2. How to start

WARNING !!!

The OpenFrame works with several modules, at the delivery all modules **have only the linux bootloader installed**, therefore to put the OpenFrame in working status you must do the following operations:

1) connect the linux console to AMP MODU II connector J12
2) connect the LAN cable UTP to RJ connector J13
3) run the TFTP application and configure the "serverip"
4) install and configure the kernel using SW manual procedure
5) install and configure the file system using SW manual procedure
6) choice the LCD display configuration

Note: the linux console is available on the Amp Modu II connector J12 in RS232 format. Another Rs232 is available on terminal block connector J17 but it can not be used as a linux console

Note: For further details about programming please refer to the SW manual

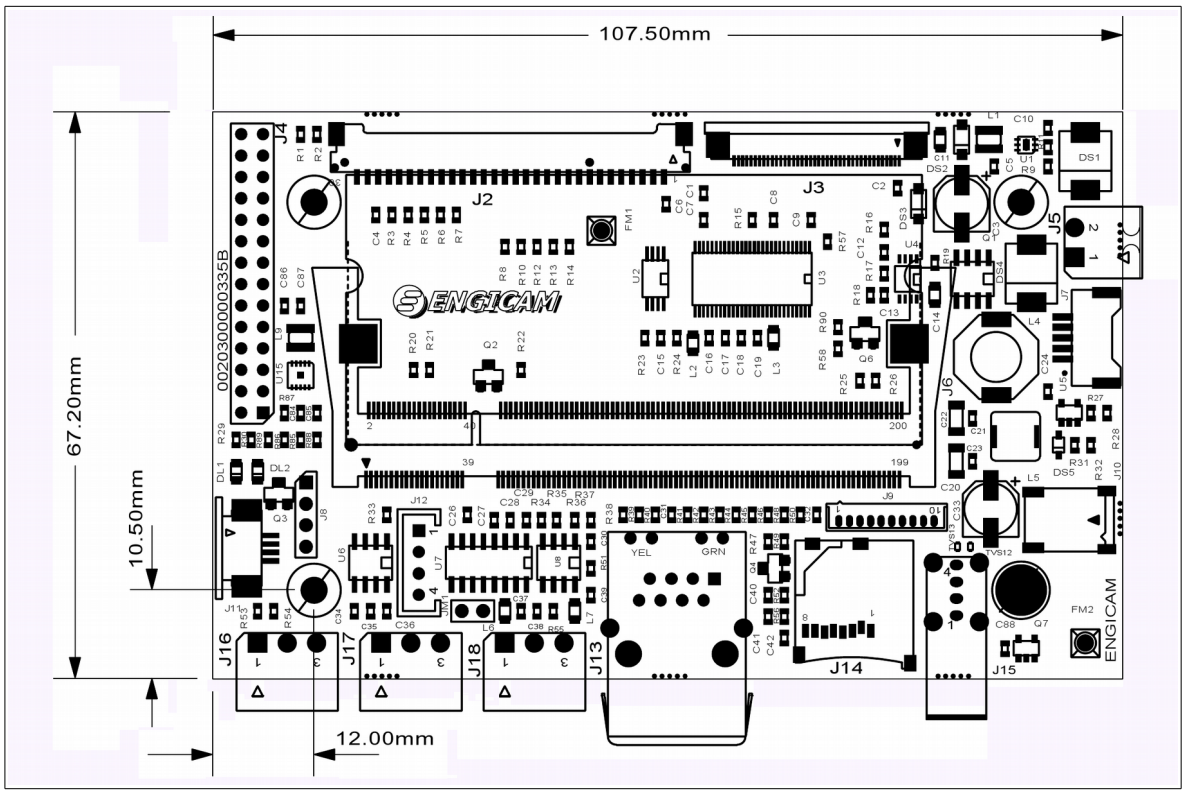
3. Carrier Board Specifications

In this chapter are described the informations about the carrier board 10.1” LOCO, these specifications include the descriptions of all the available peripheral assembled on the board and the code or the family of any mating connectors for each interface.

Concerning the useful requirements for the design of a complete customized system, are reported the informations about the mechanical dimensions of the carrier board, and the fixed points and the size of the holes

3.1 Mechanic

In the following picture is reported the drawing and the size of the carrier board. This can be useful for the customer for integrate and fix the carrier into a custom system.



***Note**: all the unit reported measure are in mm*

3.2 Micro SD Connections

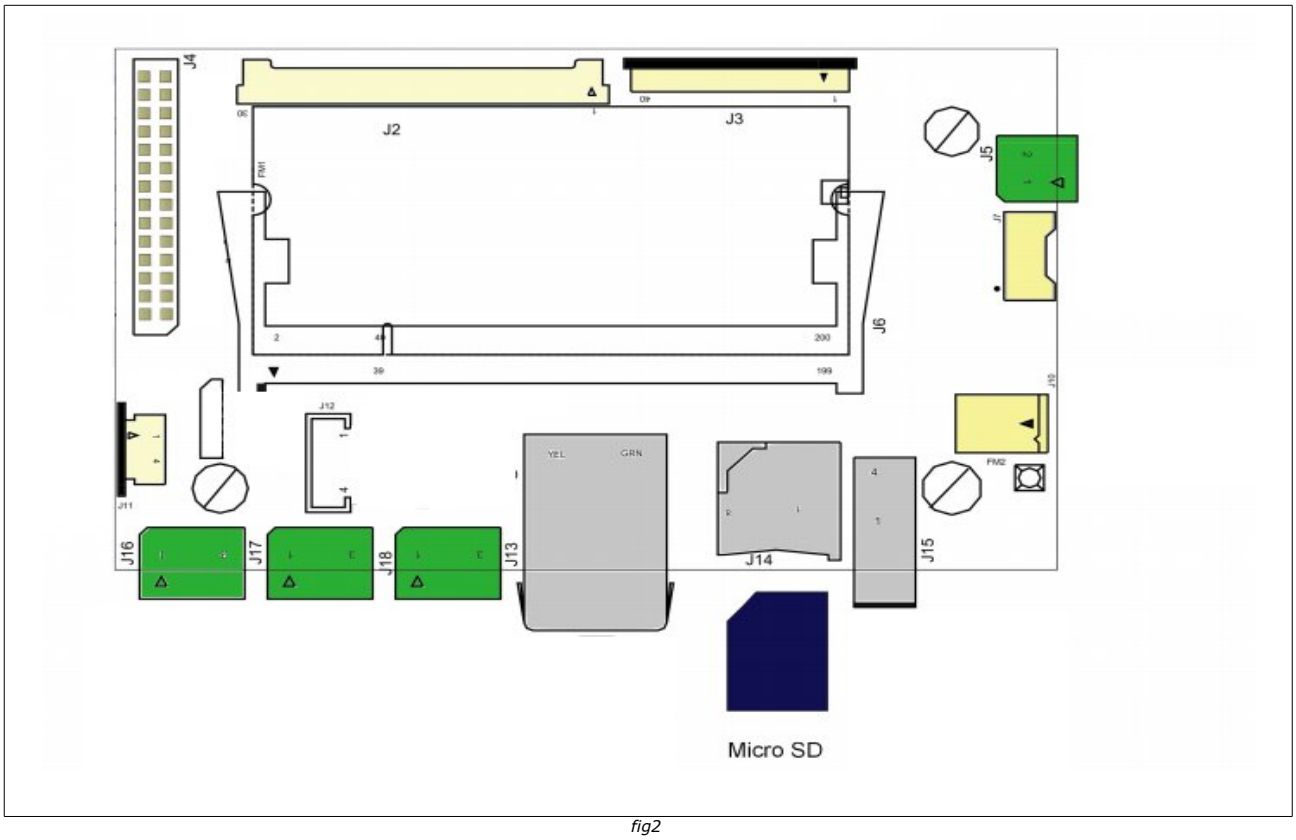


fig2

The Open-Frame board has a Micro SD switch detected connector (J14). It uses both SD standard type card and SD High Capacity type card

Micro SD card features are:

- Capacity: variable from 32 MB to 32 GB
- Length: 15 mm
- Depth:11 mm
- Height: 1 mm
- Voltage supply 2,7 V 3,6 V
- Slot : TransFlash
- MTBF: 1.000.000 h/e

3.3 RS 485 RS 232 & CAN Bus Connections

The following figure represents the connections of the RS485, RS232 and CAN bus's signals. Both RS485, RS232 and CAN bus are connected through a terminal male connector (Phoenix code MC 1.5/3-G-3.5 - 1844223 90° or compliant) mounted on PCB and referenced J16, J17, J18.

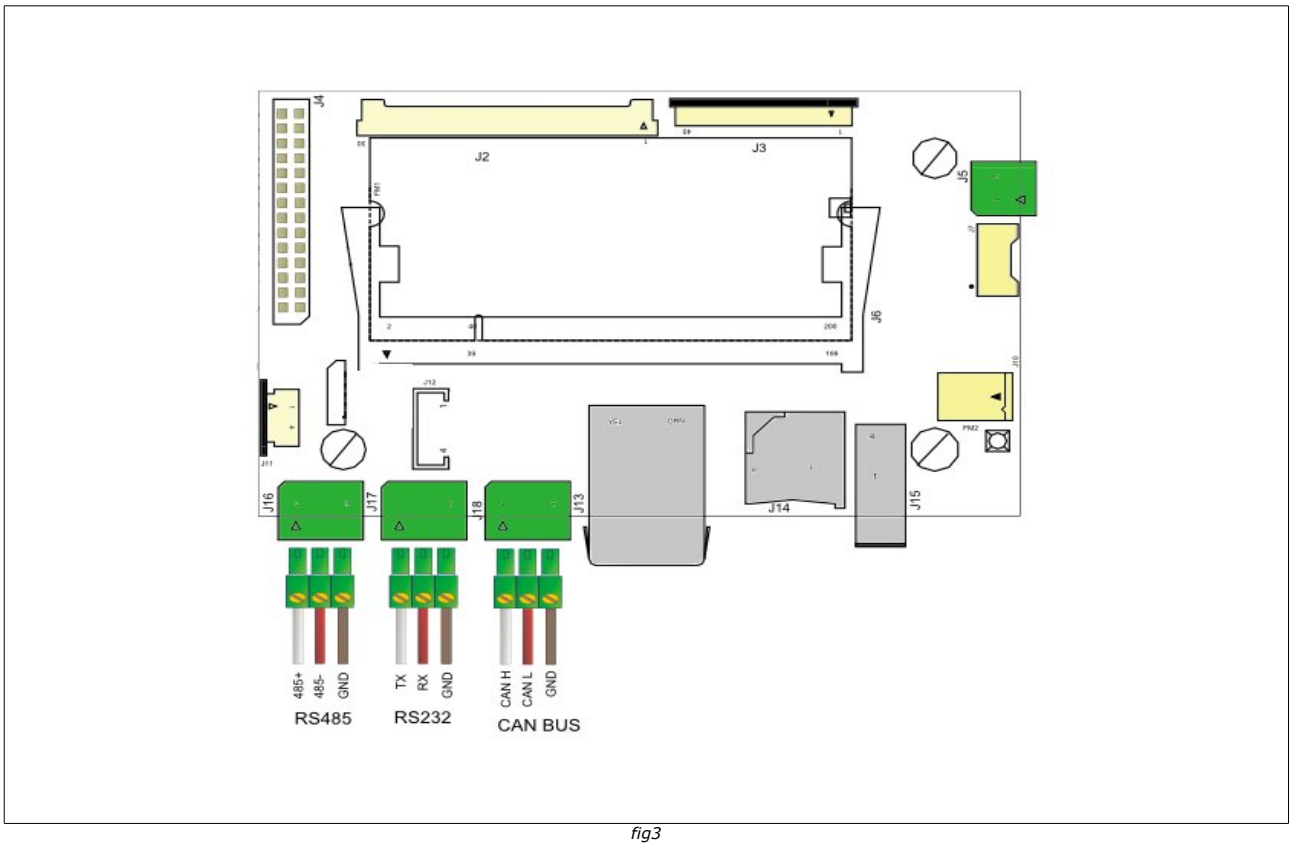


fig3

The wiring map of female connector (Phoenix code MC 1.5/3-ST-3.5 - 1840379) used for Uarts&Can is shown in the following table.

J18

Pin Number	Signal Name	Function Description	Voltage
1	CAN H	High level can bus line	Standard CAN Compliant
2	CAN L	Low level can bus line	Standard CAN Compliant
3	GND	Power Signal	-

Table 1

The CAN bus may be terminated by the using of jumper JM1 (120 Ohm)

Following are reported the cable maps for the UART interface.

J17

Pin Number	Signal Name	Function Description	Voltage
4	TX	Transmit Signal Output	RS232 Standard
5	RX	Receive Signal Input	RS232 Standard
6	GND	Power Signal	-

Table 2

J16

Pin Number	Signal Name	Function Description	Voltage
4	RS485 P	Non Inverting Receiver In/Driver Out	RS485 Standard
5	RS485 N	Inverting Receiver In/Driver Out	RS485 Standard
6	GND	Power Signal	-

Table 3

Note: Do NOT use the UART o c J17 as Linux c

3.4 USB Connections

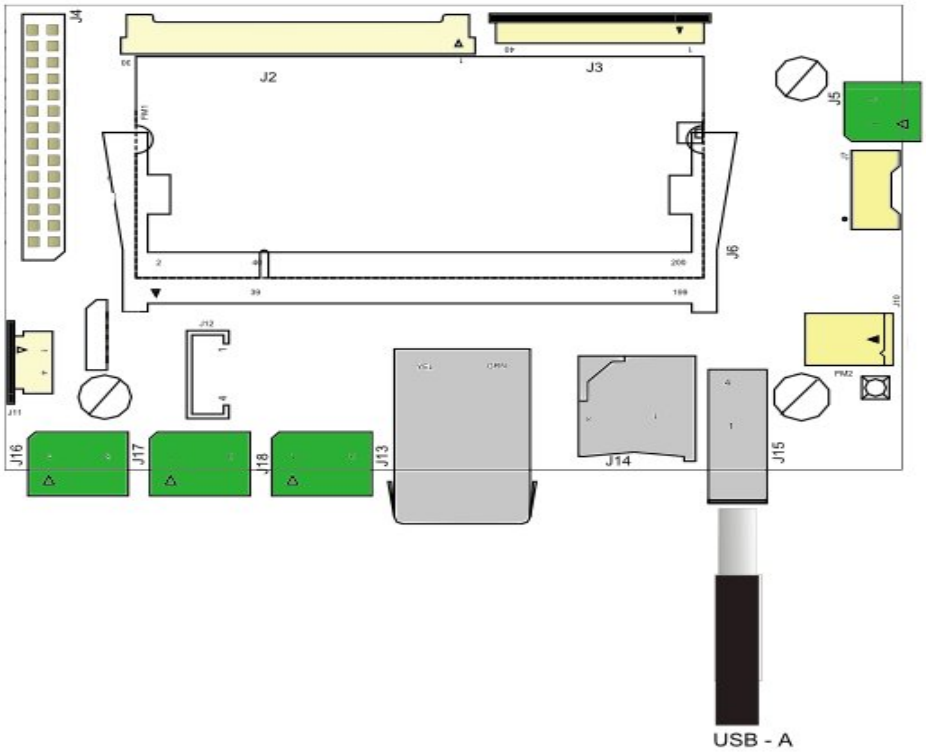


fig4

In the following tables are represented the electrical connection of the USB standard interface connections (USB type A plug)

Pin Number	Signal Name	Function Description	Voltage
1	VBUS	Power Signal	Standard USB
2	DM	Data N	Standard USB
3	DP	Data P	Standard USB
4	GND	Power Signal	Standard USB

Table 4

3.5 Ethernet Connections

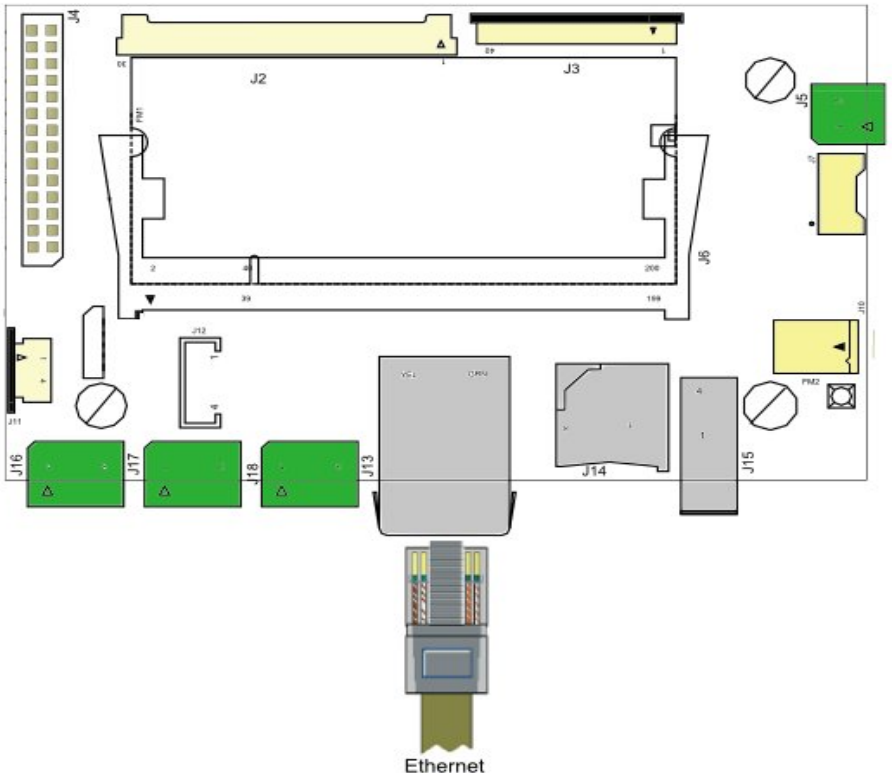


fig5

The figure5 represents the Ethernet 10/100 connection. This connection uses a RJ45 standard plug (8 wires) and the following table shows the wiring map.

Pin Number	Signal Name	Function Description	Voltage
1	TX-	Transmit positive signal	Standard Ethernet
2	TX+	Transmit negative signal	Standard Ethernet
3	RX+	Receive positive signal	Standard Ethernet
4	NC	-	-
5	NC	-	-
6	RX-	Receive negative signal	Standard Ethernet
7	NC	-	-
8	NC	-	-

Table 5

3.6 Power Supply Connections

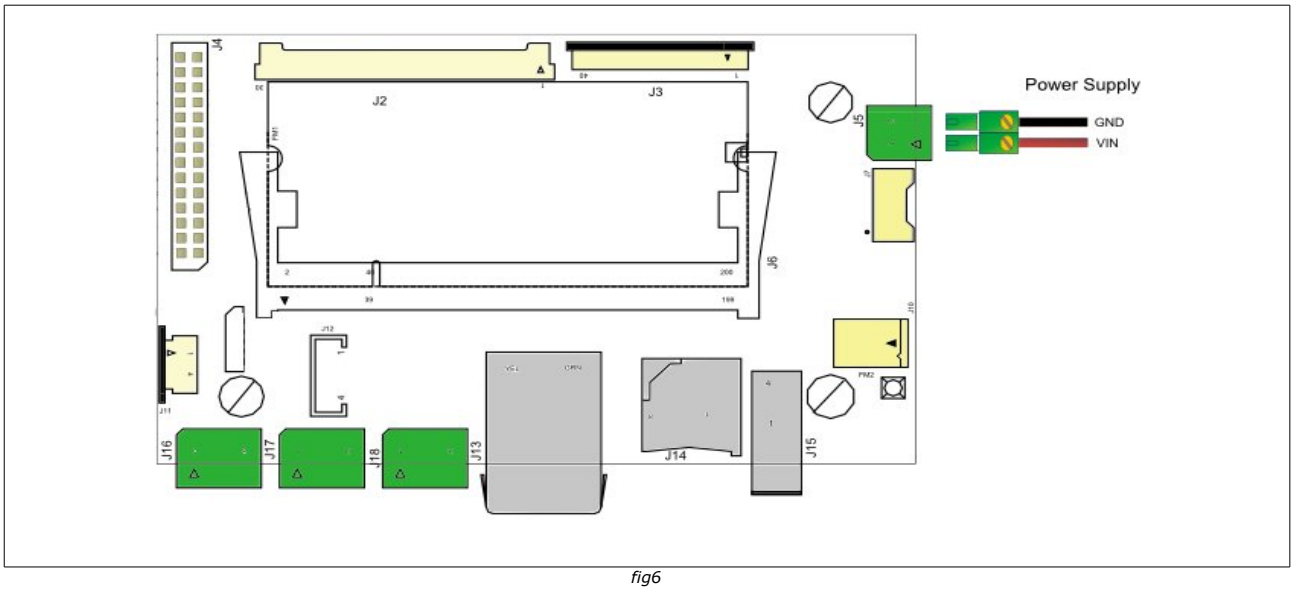


fig6

The figure shows the power supply connection. The Open-Frame receives an input DC voltage, which ranging from +10V to +30V. J5 is Phoenix MC 1.5/2-G-3.5 1844210 90° positions p.3.5mm male connector, linked as follows:

Pin Number	Signal Name	Function Description	Voltage
1	+VIN	Power Signal	Up 10 to 30 VDC
2	GND	Power Signal	-

Table 6

3.7 Current consumption

The following table shows the system's current consumption measured at 12 V and at 24 V

Gea6425 on Board	Current @ 12 V	Current @ 24V
Open-Frame 4,3”	200 mA	110 mA
Open-Frame 5,7”	370 mA	210 mA
Open-Frame 7”	330 mA	190 mA
Open-Frame 10,4”	620 mA	330 mA

Table 7

Gea6428 on Board	Current @ 12V	Current @ 24V
Open-Frame 4,3”	270 mA	130 mA
Open-Frame 5,7”	450 mA	230 mA
Open-Frame 7”	400 mA	210 mA
Open-Frame 10,4”	640 mA	340 mA

Table 8

i.CoreM53 on Board	Current @ 12V	Current @ 24V
Open-Frame 4,3”	330 mA	180 mA
Open-Frame 5,7”	390 mA	210 mA
Open-Frame 7”	380 mA	210 mA
Open-Frame 10,4”	540 mA	290 mA

Table 9

The measure is done during the standard operating mode, the LCD switched on and Ethernet cable connected

3.8 Linux or Windows CE Console Debug Connections

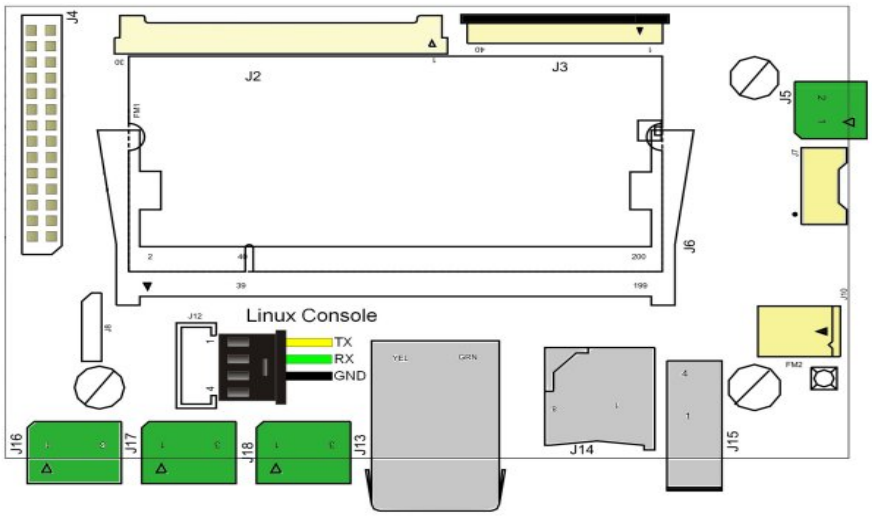


fig7

When Linux or Windows CE OS is installed on the Open-Frame module, UART1 is used like console debug. The connector used is Modu II type; in the following table is shown the electrical features:

Pin Number	Signal Name	Function Description	Voltage
1	TX	Transmit Signal	Standard RS232
2	RX	Receive Signal	Standard RS232
3	GND	Power Signal	-
4	NC	-	-

Table 10

The default communications settings is shown in following table

console default settings	
Baud rate	115200
Data length	8 bit
Parity	none
Stop	1bit

Table 11

3.9 Expansion Connector

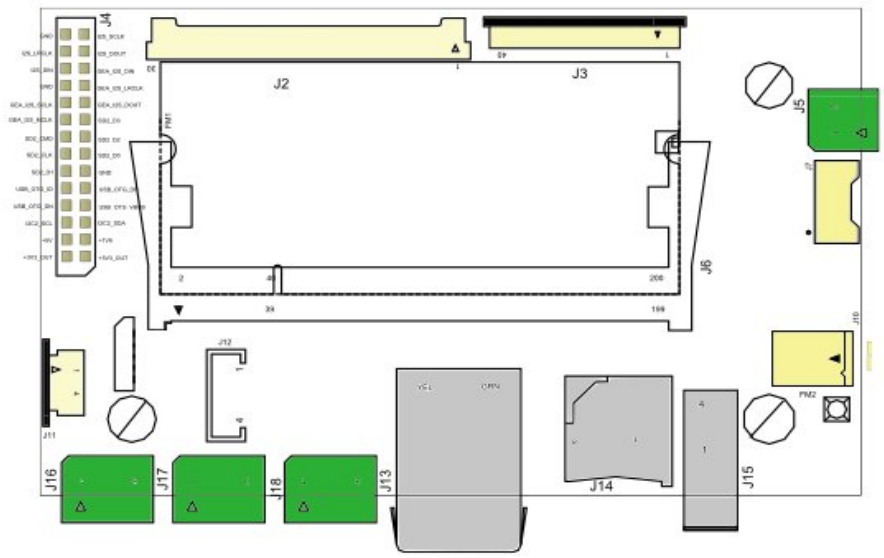


fig8

The Open-Frame is provided of Expansion connector (STRIP 2x14 Pole, 2.54mm step referenced J4) which allows to connect the following module's pins:

Pin Number on Expansion Connector	Pin on Module	Function Description	Voltage Reference
1	134	Power Signal	+3,3V_OUT
2	135	Power Signal	+3,3V_OUT
3	1, 2	Power Signal	+1.8V
4	197, 198, 199, 200	Power Signal	+5V
5	24	*	+3,3V
6	23	*	+3,3V
7	195	*	USB OTG Comp
8	193	*	USB OTG Comp
9	192	*	USB OTG Comp
10	191	*	USB OTG Comp
11	GND	Power Signal	-
12	171	*	+3,3V
13	170	*	+3,3V
14	169	*	+3,3V

Pin Number on Expansion Connector	Pin on Module	Function Description	Voltage Reference
15	168	*	+3,3V
16	167	*	+3,3V
17	166	*	+3,3V
18	12	*	+3,3V
19	14	*	+3,3V
20	15	*	+3,3V
21	16	*	+3,3V
22	34	*	+3,3V
23 **	17	*	+3,3V
24 ***	114	*	+3,3V
25	122	*	+3,3V
26	115	*	+3,3V
27	124	*	+3,3V
28	GND	Power Signal	-

Table 12

*** Note:** for the signal function please refer to modules' Hardware manual. To customise the use of those signals with alternative pin's functions please consult the modules' reference manual.

**** Note:** pin on PMIC DA9053-3JHA2-A from Dialog (only for i.CoreM53).

***** Note:** from the PCB revision "A", added GPIO pin used as Master Clock (pin 22) for the I2S implemented on pin 24-27.

4. Compiling options of the Modules

When you compile the kernel to use it on an Open-Frame you have to remember to add the following compiling options

4.1 Using GEAM6425

system type --->
Freescale MXC implementation --->
MX25 Option --->
[*] Support GEAM6425 resistive Openframe

4.2 Using GEAM6428

system type -->
Freescale i.MXA implementation -->
[*] Support GEAM6428 Openframe

4.3 Using i.CoreM53

system type --->
Freescale MXC implementation --->
Support MX53 LOCO platform --->
[*] Support iCore53 resistive Openframe

4.4 Using i.CoreM6x

This option is required only if you have a previous version of the BSP to 2.3

system type -->
Freescale i.MXA implementation -->
[*] Support i.CoreM6 resistive Openframe

Using these options you also manage in the correct way the turn on/off of the display.
These option fix also all the problems with synchronism of the signal on LCD, that may be the cause of wrong colours and noise's effects.

5. Bootargs Setup

Following is shown how to setup the bootargs to enable one of the possible TFT display used in the Open-Frame configurations.

5.1 Using GEA6425 modules

Power on the Open-Frame after have connected the serial port and have ran the hyperterminal or similar application. To enter in the shell console pressing any keys on the keyboard before the end of the countdown.

5.1.1 Using JFFS2 filesystem

Write one of the following options in the shell using the strings:

if you are using **DISPLAY 4.3"**:
set bootargs 'console=ttymxc0,115200 root=/dev/mtdblock2 rootfstype=jffs2 video=mxcfb:480x272,16bpp,URT-8484MD'

if you are using **DISPLAY 5.7"**:
set bootargs 'console=ttymxc0,115200 root=/dev/mtdblock2 rootfstype=jffs2 video=mxcfb:640x480,16bpp,URT-VGA'

if you are using **DISPLAY 7"**:
set bootargs 'console=ttymxc0,115200 root=/dev/mtdblock2 rootfstype=jffs2 video=mxcfb:800x480,16bpp,URT-LVDS'

if you are using **DISPLAY 10.4"**:
set bootargs 'console=ttymxc0,115200 root=/dev/mtdblock2 rootfstype=jffs2 video=mxcfb:800x600,16bpp,HIT-LVDS'

5.1.2 Using UBI filesystem

Write one of the following options in the shell using the strings for the UBI filesystem:

if you are using **DISPLAY 4.3"**:
set bootargs 'console=ttymxc0,115200 ubi.mtd=2 root=ubi0:rootfs rootfstype=ubifs video=mxcfb:480x272,16bpp,URT-8484MD'

if you are using **DISPLAY 5.7"**:
set bootargs 'console=ttymxc0,115200 ubi.mtd=2 root=ubi0:rootfs rootfstype=ubifs video=mxcfb:640x480,16bpp,URT-VGA'

*if you are using **DISPLAY 7"**:*
set bootargs 'console=ttymxc0,115200 ubi.mtd=2 root=ubi0:rootfs rootfstype=ubifs video=mxcfb:800x480,16bpp,URT-LVDS'

*if you are using **DISPLAY 10.4"**:*
set bootargs 'console=ttymxc0,115200 ubi.mtd=2 root=ubi0:rootfs rootfstype=ubifs video=mxcfb:800x600,16bpp,HIT-LVDS'

*Use "**print**" command to check the setup of **bootargs** and the type of filesystem (value **rootfstype=ubifs** or **rootfstype=jffs2**).*

***Note**: After the set up remember to "**save**" the configuration.*

***WARNING**:*
After the change of the string of the TFT a calibration of the touchscreen may be necessary. To do this type the command in the shell:

ts_calibrate

and then press the center of the crosses with a pen or finger. After this operation reboot the Open-Frame.

5.2 Using GEA6428 modules

Power on the Open-Frame after have connected the serial port and have ran the hyperterminal or similar application. To enter in the shell console pressing any keys on the keyboard before the end of the countdown

Write one of the following options in the shell using the strings:

For version 1.03 or latest is enough to set the following parameter to any type of file system is used because and it will be attached to the bootargs used:

if you are using **DISPLAY 4.3**":
set lcd_panel 'URT-8484MD'

if you are using **DISPLAY 5.7**":
set lcd_panel 'URT-VGA'

if you are using **DISPLAY 7**":
set lcd_panel 'URT-LVDS'

if you are using **DISPLAY 10.4**":
set lcd_panel 'HIT-LVDS'

Check what is the U-Boot version you are using (Version: Engicam U-Boot 1.xx) for version previous the 1.03 with the filesystem type jffs2 use the following bootargs:

if you are using **DISPLAY 4.3"**:
set bootargs 'console=ttyAM0,115200 root=/dev/mtdblock3 rootfstype=jffs2 gpmi lcd_panel=URT-8484MD mtdparts=gpmi-nfc-general-use:8m(kernel),220m(rootfs),-(aux)'

if you are using **DISPLAY 5.7"**:
set bootargs 'console=ttyAM0,115200 root=/dev/mtdblock3 rootfstype=jffs2 gpmi lcd_panel=URT-VGA mtdparts=gpmi-nfc-general-use:8m(kernel),220m(rootfs),-(aux)'

if you are using **DISPLAY 7"**:
set bootargs 'console=ttyAM0,115200 root=/dev/mtdblock3 rootfstype=jffs2 gpmi lcd_panel=URT-LVDS mtdparts=gpmi-nfc-general-use:8m(kernel),220m(rootfs),-(aux)'

if you are using **DISPLAY 10.4"**:
set bootargs 'console=ttyAM0,115200 root=/dev/mtdblock3 rootfstype=jffs2 gpmi lcd_panel=HIT-LVDS mtdparts=gpmi-nfc-general-use:8m(kernel),220m(rootfs),-(aux)'

Use "**print**" command to check the setup of **bootargs**.

*Note: After the set up remember to **save** the configuration.*

WARNING:
After the change of the string of the TFT a calibration of the touchscreen may be necessary.
To do this type the command in the shell:

ts_calibrate

and then press the center of the crosses with a pen or finger. After this operation reboot the Open-Frame.

5.3 Using i.CoreM53 modules

Power on the Open-Frame after have connected the serial port and have ran the hyperterminal or similar application. To enter in the shell console pressing any keys on the keyboard before the end of the countdown.
Write one of the following options in the shell using the strings:

if you are using **DISPLAY 4.3"**:
set bootargs_nand 'setenv bootargs ${bootargs} console=ttymxc0,115200 root=/dev/mtdblock2 rootfstype=jffs2 rw debug noinitrd video=mxcdi0fb:RGB666,URT-8484MD di0_primary'

if you are using **DISPLAY 5.7"**:
set bootargs_nand 'setenv bootargs ${bootargs} console=ttymxc0,115200 root=/dev/mtdblock2 rootfstype=jffs2 rw debug noinitrd video=mxcdi0fb:RGB666,URT-VGA di0_primary'

if you are using **DISPLAY 7"**:
set bootargs_nand 'setenv bootargs ${bootargs} console=ttymxc0,115200 root=/dev/mtdblock2 rootfstype=jffs2 rw debug noinitrd video=mxcdi0fb:RGB666,URT-LVDS di0_primary'

if you are using **DISPLAY 10.4"**:
set bootargs_nand 'setenv bootargs ${bootargs} console=ttymxc0,115200 root=/dev/mtdblock2 rootfstype=jffs2 rw debug noinitrd video=mxcdi0fb:RGB666,HIT-LVDS di0_primary'

Use "**print**" command to check the setup of **bootargs**.

*Note: After the set up remember to **save** the configuration.*

WARNING:
After the change of the string of the TFT a calibration of the touchscreen may be necessary.
To do this type the command in the shell:

ts_calibrate

and then press the center of the crosses with a pen or finger. After this operation reboot the Open-Frame.

5.4 Using i.CoreM6X modules

5.4.1 U-boot setup using LTIB

Power on the Open-Frame after have connected the serial port and have ran the Hyperterminal or similar application. To enter in the shell console pressing any keys on the keyboard before the end of the countdown.

if you are using **DISPLAY 4.3"**:

```
set video_type 'mxcfb0:dev=lcd'
set board 'OF.RES'
set lcd_panel 'URT-8484MD'
```

if you are using **DISPLAY 5.7"**:

```
set video_type 'mxcfb0:dev=lcd'
set board 'OF.RES'
set lcd_panel 'URT-VGA'
```

if you are using **DISPLAY 7"**:

```
set video_type 'mxcfb0:dev=lcd'
set board 'OF.RES'
set lcd_panel 'URT-LVDS'
```

if you are using **DISPLAY 10.4"**:

```
set video_type 'mxcfb0:dev=lcd'
set board 'OF.RES'
set lcd_panel 'HIT-LVDS'
```

5.4.2 U-boot setup using YOCTO

To use the open frame, or related, is required to run the configuration parameters to enable the several devices via bootloader (see previous chapter for settings) then load the related open-frame fdt_file instead of the iCore starterkit .dtb file.

Bootloader's settings

if you are using **DISPLAY 4.3"**:

```
set video_type 'mxcfb0:dev=lcd'
set board 'OF.RES'
set lcd_panel 'URT-8484MD'
```

if you are using **DISPLAY 5.7"**:

```
set video_type 'mxcfb0:dev=lcd'
set board 'OF.RES'
set lcd_panel 'URT-VGA'
```

if you are using **DISPLAY 7"**:

```
set video_type 'mxcfb0:dev=lcd'
set board 'OF.RES'
set lcd_panel 'URT-LVDS'
```

if you are using **DISPLAY 10.4"**:

```
set video_type 'mxcfb0:dev=lcd'
set board 'OF.RES'
set lcd_panel 'HIT-LVDS'
```

Device tree setup

Now it is required to load the right device tree at the bootloader. In the current state all the compiling, performed by Yocto, produce a different dtb for each Engicam board. All the dtb labelled "**ofres**" are referred to the open-frame board all the dtb labelled "**icore**" are referred to the starterkit.

<module_type>-ofres.dtb
<module_type>-icore.dtb

Boot from SD

If the bootloader starts from SD card, the bootloader partition must be included both the file dtb. To load the right dtb file, stop in the boot mode and edit the following command:

set ftd_file imx6dl-ofres.dtb
set ftd_file imx6q-ofres.dtb

Boot from NAND

When the system is started from NAND the file uImage.dtb must be copied with the correct dtb source with respect to the open-frame. See the following example:

cp imx6dl-ofres.dtb uImage.dtb

5.4.3 U-boot setup using LTIB version previous 1.09

Using LTIB and a versions of u-boot previus to 1.09, proceed as follows:

This bootargs are required for previous version

if you are using **DISPLAY 4.3"**:

set bootargs 'console=ttymxc3,115200n8 arm_freq=800 engi_board=OF.RES ubi.mtd=2 root=ubi0:rootfs rootfstype=ubifs rootwait rw video=mxcfb0:dev=lcd,URT-8484MD'

if you are using **DISPLAY 5.7"**:

set bootargs 'console=ttymxc3,115200n8 arm_freq=800 engi_board=OF.RES ubi.mtd=2 root=ubi0:rootfs rootfstype=ubifs rootwait rw video=mxcfb0:dev=lcd,URT-VGA'

if you are using **DISPLAY 7"**:

set bootargs 'console=ttymxc3,115200n8 arm_freq=800 engi_board=OF.RES ubi.mtd=2 root=ubi0:rootfs rootfstype=ubifs rootwait rw video=mxcfb0:dev=lcd,URT-LVDS'

if you are using **DISPLAY 10.4"**:

set bootargs 'console=ttymxc3,115200n8 arm_freq=800 engi_board=OF.RES ubi.mtd=2 root=ubi0:rootfs rootfstype=ubifs rootwait rw video=mxcfb0:dev=lcd,HIT-LVDS'

Use "**print**" command to check the setup of **bootargs**.

*Note: After the set up remember to **save** the configuration.*

WARNING:
After the change of the string of the TFT a calibration of the touchscreen may be necessary.
To do this type the command in the shell:

ts_calibrate

and then press the center of the crosses with a pen or finger. After this operation reboot the Open-Frame.

For further information please refer to "i.CoreM6x_Yocto_sw_manual"

5.5 Using Chimera modules

Power on the Open-Frame after have connected the serial port and have ran the hyperterminal or similar application. To enter in the shell console pressing any keys on the keyboard before the end of the countdown.

if you are using **DISPLAY 4.3"**:

```
set video_type 'mxcfb0:dev=lcd'
set board 'OF.RES'
set lcd_panel 'URT-8484MD'
```

if you are using **DISPLAY 5.7"**:

```
set video_type 'mxcfb0:dev=lcd'
set board 'OF.RES'
set lcd_panel 'URT-VGA'
```

if you are using **DISPLAY 7"**:

```
set video_type 'mxcfb0:dev=lcd'
set board 'OF.RES'
set lcd_panel 'URT-LVDS'
```

if you are using **DISPLAY 10.4"**:

```
set video_type 'mxcfb0:dev=lcd'
set board 'OF.RES'
set lcd_panel 'HIT-LVDS'
```

Use "**print**" command to check the setup of **bootargs**.

*Note: After the set up remember to **save** the configuration.*

WARNING:
After the change of the string of the TFT a calibration of the touchscreen may be necessary.
To do this type the command in the shell:

ts_calibrate

and then press the center of the crosses with a pen or finger. After this operation reboot the Open-Frame.

For further details please refer to

6. Ordering Information

Following we provide the ordering informations and the description for the Basic technical specifications:

Ordering Code	MPQ	Description	Operating temperature range °C
00257000003077	1	Open-Frame 4,3”	-20 to +85 *
00257000003076	1	Open-Frame 5,7”	-20 to +85 *
00257000003114	1	Open-Frame 7”	-20 to +85 *
00257000003078	1	Open-Frame 10,4”	-20 to +85 *

Table 13

** LCD and Module excluded!*

For each LCD is declared the following operating temperature range: ***-20 to +70****.*
For each Open-Frame code a sample was tested in the operative temperature range (-20 to +85).

It's possible to find the module's operating temperature using ENGICAM's Hw manual and NXP's reference documentations.

7. Technical support

Engicam provides to its customers the latest versions of its BSP of development. In the time between one version and the other a few changes or improvements can be published and shared via patch or textual documents.
Every customer that bought a BSP has at its disposal a virtual account in which find all this informations.
In case you do not have a ftp account, contact your own dealer to obtain the access.
We recommend to always refer to the latest files of your FTP account.
Every new release of a BSP will contain all the patches and improvements described and released during the time between the release of a BSP and the other. For this reason, older versions will gradually be removed from the account. We recommend to keep aligned the own material with respect to the official releases of a BSP.

7.1 Ftp account structure

- **Virtual Machine**
  It can be in the form of archive or Windows installer
- **PCN Folder**
  Containing all the hardware and software PCN (Product Change Notification) of the Module
- **Doc Folder**
  Containing the documents of both hardware and software up to date with all the manuals of the module.
- **Patch Folder**
  Divided between boot and kernel patch contains all the patches for some bug fixes or improvements introduced. The patch.description file contains the description of each patch.
  In #14.2. How to upgrading your BSP using patch, however, the added descriptions are listed.

7.2 How to upgrading your BSP using patch

If you have already started the development and customization of the kernel, you can keep up to date through the use of patches that are on your account. This chapter gives some information on how to manage and update your BSP. If you have already achieved a stable version of the system it is advisable to only apply patches that may affect your application.

7.2.1 Structure of the patch folder

This folder contains any patches of the downloadable version. The patches have the following order:

PATCH_*KERNEL/BOOT*_**BSP**_*X.X*

In the name it's specified "KERNEL or BOOT", where the patch must be applied and the BSP revision "X.X"
Inside the folder you find the main patches that have the following method:

*00X*_**main_patch**_*YYMMDD*.**patch**

main patch at the date YYMMDD that aligns the whole kernel with our mainline.

Then you can find the singular patch with the following method:

*00X.0Y_argument_name***.patch**

where "00X" specified the membership to the main patch, "0Y" is an incremental number, the "argument_name" is the name of the fixed problem by the patch.
These Patches solves the individual problem and are usually used by the customer who has already customized the kernel so that is not necessary having to apply the main patch.

7.2.2 Patch structure

A patch consists of several sections of code to add or remove, these sections are localized reporting the previous and subsequent lines of code where to edit the changes.

```
1 diff --git a/arch/arm/mach-mx6/Kconfig b/arch/arm/mach-mx6/Kconfig
2 index 0c6e89e..db1a58b 100644
3 --- a/arch/arm/mach-mx6/Kconfig
4 +++ b/arch/arm/mach-mx6/Kconfig
5 @@ -222,6 +222,14 @@ config MACH_MX6Q_MINIMUM_FREQ400
6           This features set the minimum CPU clock frequency to 400 Mhz instead of 200 Mhz.
7           Recommended option for the use of video codecs.
8
9 +config MACH_MX6Q_ICORE_OPENFRAME_RESISTIVE
0 +       bool "Support i.CoreM6 resistive OpenFrame"
1 +       depends on MACH_MX6Q_ICORE
2 +       help
3 +         Include the support for Engicam openframe. This features enabled
4 +         the correct power up and power down of the on-board LVDS controller.
5 +          This features is mandatory.
6 +
7  config MACH_MX6Q_ICORE_STARTERKIT_CAP_EDT
8         bool "Support i.Core capacitive starterkit"
9         depends on MACH_MX6Q_ICORE
```

fig9

Diff –- indicates the file and the location you want to edit.
@ indicates the code lines where edit the modifies.

Following are shown the lines of code that should not be changed.

With **+** or **–** are shown the lines to add or remove to edit the file. The remaining code is used to identify where to apply the patch

7.2.3 How to apply the patch

A patch is applied if the command patch can correctly identify where to insert the code parts. Otherwise, it generates an error file and the patch must be changed manually.
In case you have customized the code, you can follow the structure of the patch and apply it manually to avoid errors in editing. Refer to the previous chapter for a description of the structure of the patch. To apply a patch follow the below procedures. Enter the folder you want to edit (U-Boot or kernel). Copy the patch inside the folder:

cd linux

with the dry-run command you can try to apply a patch and see if it returns an error to

evaluate whether it is possible to apply the patch.

patch -p1 --dry-run < 002.02_iCoreM6_openframe_lvds.patch

Once you have tested the application, if there are not too many mistakes, you may apply the patch using the command:

patch -p1 < 002.02_iCoreM6_openframe_lvds.patch

If the application is successful, rebuild the kernel and update it on the device. In the case of errors you will have an output like this:

patching file arch/arm/mach-mx6/Kconfig
Hunk #1 FAILED at 222.
1 out of 1 hunk FAILED -- saving rejects to file arch/arm/mach-mx6/Kconfig.rej
patching file arch/arm/mach-mx6/board-mx6q_icore.c
Hunk #1 FAILED at 107.
Hunk #2 succeeded at 264 (offset -6 lines).
Hunk #3 succeeded at 417 (offset -8 lines).
Hunk #4 succeeded at 1427 with fuzz 2 (offset -19 lines).
Hunk #5 succeeded at 1629 (offset -33 lines).
1 out of 5 hunks FAILED -- saving rejects to file arch/arm/mach-mx6/board-mx6q_icore.c.rej

In the above example the patch failed the application of the Kconfig file and board-mx6q_icore.c.

The failed parts are available in the file .rej that are generated by the patch command. Then open the file and manually enter the parts not included.

7.3 Product Compliance

In order to respect own internal policy regarding the environmental regulations and safety laws, Engicam in this chapter confirms the compliant, when applicable, of its own products to the normatives ROHS and REACH and to the recognized hazards.

Warning!
The current product board mounts a VL-1220/HFN Rechargeable Battery, that has the following elements included into the SVHC list:

- ***1,2-dimethoxyethane, ethylene glycol dimethyl ether (EGDME)***

8. On-line Support

We offer an on-line support to allow the customer to stay updated on the development of software release and on the enhancement of the documentation.

Following is shown the references for ENGICAM on-line support.

8.1 Support

ENGICAM Product Experts are available to answer questions via email:

support@engicam.com

8.2 Disclaimer

Information in this document is provided solely to enable system and software implementers to use Engicam products. Engicam does not guarantee that the information in this manual is up-to-date, correct, complete or of good quality. Nor does Engicam assume guarantee for further usage of the information.

Liability claims against Engicam, referring to material or non-material related damages caused, due to usage or non-usage of the information given in the manual, or due to usage of erroneous or incomplete information, are exempted.

Engicam explicitly reserves the rights to change or add to the contents of this manual or parts of it without special notification. All operating parameters must be validated for each customer application by customer's technical experts.