#WeAreConnectivity

SMART SOLUTIONS FOR A CHANGING WORLD

MTX-StarRoad II

BSP SOFTWARE

YOCTO 2.4

USER MANUAL

www.mtxm2m.com

INDEX

GENERAL NOTES

Product is deemed accepted by recipient and is provided without interface to recipient's products. The documentation and/or product are provided for testing, evaluation, integration and information purposes. The documentation and/or product are provided on an "as is" basis only and may contain deficiencies or inadequacies. The documentation and/or product are provided without warranty of any kind, express or implied. To the maximum extent permitted by applicable law, Matrix Electronica S.L.U. further disclaims all warranties; including without limitation any implied warranties of merchantability, completeness, fitness for a particular purpose and non-infringement of third-party rights. The entire risk arising out of the use or performance of the product and documentation remains with recipient. This product is not intended for use in life support appliances, devices or systems where a malfunction of the product can reasonably be expected to result in personal injury. Applications incorporating the described product must be designed to be in accordance with the technical specifications provided in these guidelines. Failure to comply with any of the required procedures can result in malfunctions or serious discrepancies in results.

Furthermore, all safety instructions regarding the use of mobile technical systems, including GSM products, which also apply to cellular phones, must be followed. Matrix Electronica S.L.U. or its suppliers shall, regardless of any legal theory upon which the claim is based, not be liable for any consequential, incidental, direct, indirect, punitive or other damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information or data, or other pecuniary loss) arising out the use of or inability to use the documentation and/or product, even if Matrix Electronica S.L.U. has been advised of the possibility of such damages. The foregoing limitations of liability shall not apply in case of mandatory liability, e.g. under the Spanish Product Liability Act, in case of intent, gross negligence, injury of life, body or health, or breach of a condition which goes to the root of the contract. However, claims for damages arising from a breach of a condition, which goes to the root of the contract, shall be limited to the foreseeable damage, which is intrinsic to the contract, unless caused by intent or gross negligence or based on liability for injury of life, body or health. The above provision does not imply a change on the burden of proof to the detriment of the recipient. Subject to change without notice at any time. The interpretation of this general note shall be governed and construed according to Spanish law without reference to any other substantive law.

IMPORTANT INFORMATION

This technical description contains important information for the startup and use of the MTX-StarRoad II device. Read it carefully before you start working with the MTX-StarRoad II device. The warranty will be void should damage occur due to non-compliance with these instructions for use. We cannot accept any responsibility for consequential loss.

SERVICE AND SUPPORT

To contact customer support please contact your local distributor/sales agent or use the details below:

Address: Alejandro Sánchez 109, 28019 Madrid (Spain)
Email: gsmsupport@matrix.es
Website: mtxm2m.com

REVISION INFORMATION

REVISION	DATE	AUTHOR	CHANGES
1.0	2020/05	PG	Initial release

INTRODUCTION

This document describes how to build an image for an MTX-STAROAD, based on Solidrun’s iMX6 module.

REQUIREMENTS

1. Ubuntu 16.04 64 bits Linux or Virtual Machine Host
2. Essential host packages must be installed:
    - gawk
    - wget
    - git-core
    - diffstat
    - unzip
    - texinfo
    - build-essential
    - chrpath
    - libsdl1.2-dev
    - xterm
    - curl
    - ncursesw-dev
    - gparted

Execute the following commands to install the packages on the host:

```
$ sudo apt-get update

$ sudo apt-get install gawk wget git-core diffstat unzip texinfo
build-essential chrpath libsdl1.2-dev xterm curl  libncursesw5-dev
gparted
```

ENVIRONMENTAL SETUP

It is recommended to create a directory to install the environment, for example, in this document we are going to use ~/BSP

```
$ mkdir ~/BSP
```

Execute the following commands to download the git repositories.

```
$ cd ~/BSP

$ git clone -b rocko git://git.yoctoproject.org/poky.git

$ cd poky

$ git clone -b rocko https://github.com/Freescale/meta-freescale.git

$ git clone -b rocko https://github.com/Freescale/meta-freescale-3rdparty.git

$ git clone -b rocko https://github.com/Freescale/meta-freescale-distro.git

$ git clone -b rocko git://github.com/meta-qt5/meta-qt5.git

$ git clone -b rocko git://git.openembedded.org/meta-openembedded
```

Extract meta-mtx-arm-imx6.targz to complete git repositories:

```
$ tar -xvzf meta-mtx-arm-imx6.tar.gz -C ~/BSP/poky/
```

BUILD IMAGE

The first step to build a image is loading the enviorement. Execute the following commands:

```
$ cd ~/BSP/poky/

$ TEMPLATECONF=meta-mtx-arm-imx6/custom-bsp-files/mtxstaroad-imx6/conf
source oe-init-build-env <BUILD>
```

Replace <BUILD> by a custom build directory name, for instance "build".

NOTES:

The developer must execute the previous command in each new terminal.

The first time, it creates a <BUILD> directory and compilation files, and load the environment for the mtxstaroad-imx6 machine. More executions only load the environment. Check [8] for more information.

Execute the following command to start the build.

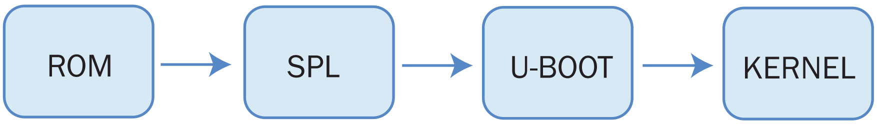
```
$ bitbake core-image-base
```

The compilation result will be in:

"<BSP_DIR>/poky/<BUILD>/tmp/deploy/images/mtxstaroad-imx6"

The main files are:

- core-image-base-mtxstaroad-imx6.wic.gz: This tarball file includes filesystem, kernel, devicetree and generated modules. The developer must to extract it to get the image core-image-base-mtxstaroad-imx6.wic (it is needed for flashing an SD or EMMC)
- modules- mtxstaroad-imx6.tgz: This tarball file includes the generated kernel modules
- zImage: kernel image
- u-boot.img: Uboot image (it is needed for flashing an SD or EMMC)
- SPL: Secondary Program Loader image (it is needed for flashing an SD or EMMC)



- imx6dl-mtxstaroad2-emmc-som-v15.dtb: Devicetree for Solo/DualLite CPU module
- imx6q-mtxstaroad3-emmc-som-v15.dtb: Devicetree for Dual/Quad CPU module

FLASHING SD CARD IMAGE

An SD card image provides the full system to boot with U-Boot and kernel.

To flash an SD card image, execute the following command:

dd if=<fichero.wic> of=<nodo_sd> bs=4M conv=fsync

To flash an SD card image according to this document. The developer needs to insert the sd card to Host and research the device file for it.   For this example, we are going to assume the device is /dev/sdx.

SD image: core-image-base-mtxstaroad-imx6.wic.gz

NOTE: A micro SD card of 1GB or larger is required.

Execute the following commands:

```
$ cd ~/BSP/poky/build/tmp/deploy/images/mtxstaroad-imx6/

$ gunzip -fk core-image-base-mtxstaroad-imx6.wic.gz

$ sudo dd if=./core-image-base-mtxstaroad-imx6.wic of=/dev/sdx bs=4M
conv=fsync
```

FLASHING EMMC IMAGE

It is possible to use the image for SD to flash it onto eMMC device. Therefore

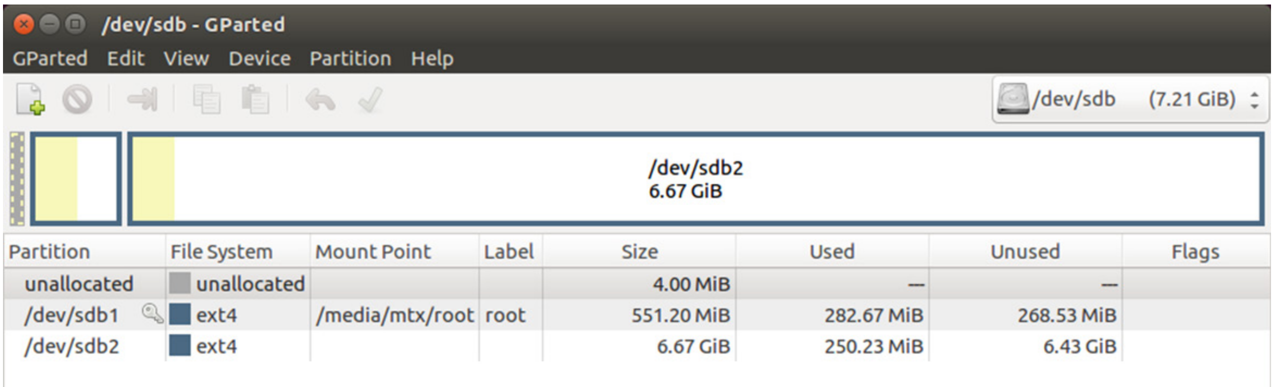
eMMC image: core-image-base-mtxstaroad-imx6.wic.gz (the same image for SD)

This procedure needs to create an SD (point 5) and create a second ext4 partition in SD to store the files to install on eMMC.

We'll use GParted, but it is possible to use another program to create the second partition.

The following figure shows the partitions for an SD in GParted program. The "root" partition is the system to load on the target and the second one is the partition storage for the files to flash onto eMMC.

NOTE: A micro SD card of 2GB or larger is required.



Partition	File System	Mount Point	Label	Size	Used	Unused	Flags
unallocated	unallocated			4.00 MiB	—	—	
/dev/sdb1	ext4	/media/mtx/root	root	551.20 MiB	282.67 MiB	268.53 MiB	
/dev/sdb2	ext4			6.67 GiB	250.23 MiB	6.43 GiB	

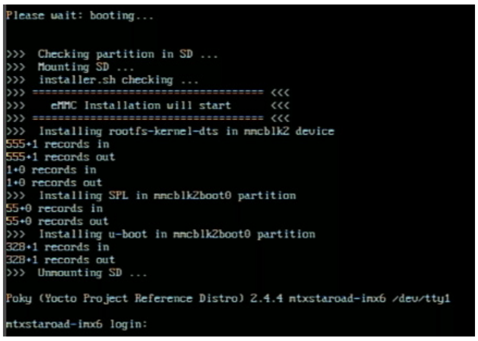
Copy the following files to the second partition:

- core-image-base-mtxstaroad-imx6.wic
- u-boot.img
- SPL
- installer.sh

```
$ cd ~/BSP/poky/build/tmp/deploy/images/mtxstaroad-imx6/

$ sudo cp core-image-base-mtxstaroad-imx6.wic <SD_SECOND_PARTITION_
PATH>

$ sudo cp u-boot.img <SD_SECOND_PARTITION_PATH>

$ sudo cp SPL <SD_SECOND_PARTITION_PATH>

$ sudo cp ~/BSP/poky/meta-mtx-arm-imx6/custom-bsp-files/mtxstaroad/
emmc-installer/installer.sh <SD_SECOND_PARTITION_PATH>

$ sudo chmod +x <SD_SECOND_PARTITION_PATH>/installer.sh

$ sync
```

Once the above steps are done, flashing eMMc image is possible.

1. Make sure the equipment is off
2. Insert the SD (point 6)
3. Make sure the microswitches position are in the SD BOOT ON/OFF/ON/OFF
4. Power on the equipment
5. The installation will start automatically. The messages will show on the display
6. When the installation is done, power off the equipment and extracts the SD card



```
Please wait: booting...


>>>  Checking partition in SD ...
>>>  Mounting SD ...
>>>  installer.sh checking ...
>>> ==================================== <<<
>>>    eMMC Installation will start      <<<
>>> ==================================== <<<
>>>  Installing rootfs-kernel-dts in mmcblk2 device
555+1 records in
555+1 records out
1+0 records in
1+0 records out
>>>  Installing SPL in mmcblk2boot0 partition
55+0 records in
55+0 records out
>>>  Installing u-boot in mmcblk2boot0 partition
328+1 records in
328+1 records out
>>>  Unmounting SD ...

Poky (Yocto Project Reference Distro) 2.4.4 mtxstaroad-imx6 /dev/tty1

mtxstaroad-imx6 login:
```

7. Change the microswitches position to eMMC BOOT ON/OFF/OFF/ON

NOTE: If you already have an installer SD and need to reflash an image onto eMMC, you only need to replace the following files in the second partition:

- core-image-base-mtxstaroad-imx6.wic
- u-boot.img
- SPL

LOAD THE SYSTEM

Once a bootable SD was created or the Flash image onto eMMC is done. You can load the system from SD or eMMC.

Uboot has the first boot option from SD and the second one from eMMC.

Make sure you have an inserted/extracted SD, and the properly configured microswitches, according to the chosen boot option.

	SW2			
SD	ON	OFF	ON	OFF
eMMC	ON	OFF	OFF	ON

ENTER THE SYSTEM

The system has two options:

SERIAL	ETHERNET
115200-8-N-1 User: root Password: root	Protocol: ssh User: root Password: root Target IP: 192.168.1.2

NOTE: Check [9] to customize users and passwords.

TESTS

To check the equipment is working properly, you could execute the following tests through serial or ssh session.

1. Native GPIOS

The GPIOs can be accessed from the sysfs.

The following formula is needed to get the gpio number in userspace:

Num = (Port - 1) *32 + Pin

For example:

GPIO4_30 is gpio 126 in the sysfs.

Num = (4 - 1) *32 + 30 = 126

Then you could export and use it.

```
$ echo 126 > /sys/class/gpio/export
$ cat /sys/class/gpio/gpio126/value
$ cat /sys/class/gpio/gpio126/direction
```

Check [10] for more information.

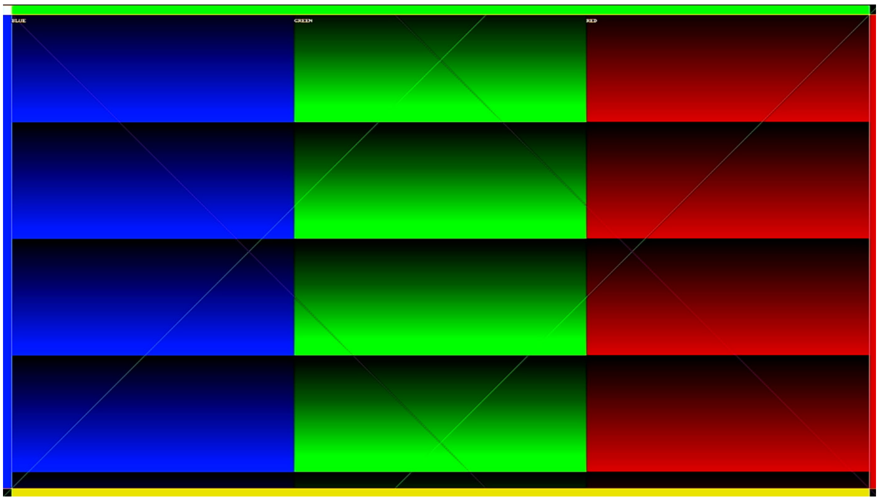
2. I2C Devices

Execute the following command to get the i2c devices list on bus 0:

```
$ i2cdetect 0

WARNING! This program can confuse your I2C bus, cause data loss and
worse!

I will probe file /dev/i2c-0.

I will probe address range 0x03-0x77.

Continue? [Y/n]

     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f

00:          -- -- -- -- -- -- -- -- -- -- -- -- --

10: -- -- -- -- -- -- -- -- UU -- -- -- -- -- -- --

20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --

30: 30 -- -- -- -- -- -- -- -- -- -- -- -- -- -- --

40: -- -- -- -- -- -- -- -- UU -- -- -- -- -- -- --

50: 50 UU -- -- 54 -- -- -- 58 -- -- -- -- -- -- --

60: -- -- -- -- -- -- -- -- -- UU UU -- -- -- -- --

70: -- -- -- -- UU -- -- --
```

3. HDMI

```
$ fb-test
```



4. Audio

```
$ aplay /opt/testing-tools/test_utils/HeyJude_orig.wav
```

5. GPU

Execute gstreamer pipeline to test the GPU:

```
$ gst-launch-1.0   \

imxg2dcompositor name=c background-color=0x223344 \

sink_0::xpos=0 sink_0::ypos=90 sink_0::width=160 sink_0::height=110
sink_0::zorder=55 sink_0::fill_color=0xff00ff00 sink_0::alpha=0.39
sink_0::rotation=0 \

sink_1::xpos=0 sink_1::ypos=20 sink_1::width=620 sink_1::height=380
sink_1::fill_color=0x44441133 ! \

queue2 ! “video/x-raw, width=800, height=600” ! imxg2dvideosink \

videotestsrc pattern=0 ! “video/x-raw, framerate=30/1” ! c.sink_0 \

videotestsrc pattern=18 ! “video/x-raw, framerate=30/1” ! c.sink_1
```

6. VPU

Execute gstreamer pipeline to test the VPU:

- Play video:

```
$ gst-launch-1.0 filesrc location=/opt/testing-tools/test_
utils/sintel_trailer-1080p.mp4 \

! qtdemux ! queue ! h264parse ! imxvpudec ! imxg2dvideosink
```

- Play video and audio:

```
$ gst-launch-1.0 filesrc location=/opt/testing-tools/test_
utils/sintel_trailer-1080p.mp4 ! \

qtdemux name=d \

d. ! queue ! h264parse ! imxvpudec ! imxg2dvideosink \

d. ! queue ! avdec_aac ! audioconvert ! alsasink
```

NOTE: These pipelines are a few examples of GStreamer using. More information about Gstreamer [5] [6] y [7].

The video for this test is available to download: https://download.blender.org/durian/trailer/sintel_trailer-1080p.mp4.

7. Temperature Sensor - LM73

Execute the following command to get the temperature of the sensor. This value needs to divide by 1000 to convert to Celsius degrees.

```
$ cat /sys/class/hwmon/hwmon1/temp1_input

32000
```

8. Gyroscope - Accelerometer

Execute the following command to get raw data from the device.

Gyroscope:

```
$ ls /sys/bus/iio/devices/iio\:device1/
buffer                         in_anglvel_y_raw              power
current_timestamp_clock         in_anglvel_y_scale              sampling_
frequency
dev                             in_anglvel_z_raw                sampling_
frequency_available
in_anglvel_scale_available      in_anglvel_z_scale              scan_
elements
in_anglvel_x_raw                name                            subsystem
in_anglvel_x_scale              of_node                         uevent


$ cat /sys/bus/iio/devices/iio\:device1/in_anglvel_x_raw
-2725
$ cat /sys/bus/iio/devices/iio\:device1/in_anglvel_y_raw
492
$ cat /sys/bus/iio/devices/iio\:device1/in_anglvel_z_raw
476
$ cat /sys/bus/iio/devices/iio\:device1/in_anglvel_x_scale
0.000153
```

Then you could get the real measurement through this formula:

Input angl.vel.<x,y,z>(rad) = in_anglvel_<x,y,z>_raw × in_anglvel_<x,y,z>_scale

For example:

Input angl.vel. x (rad)=-2725×0.000153=-0,416925

Accelerometer:

```
$ ls /sys/bus/iio/devices/iio\:device0/
buffer                         in_accel_y_raw               power
current_timestamp_clock         in_accel_y_scale              sampling_
frequency
dev                             in_accel_z_raw                sampling_
frequency_available
in_accel_scale_available        in_accel_z_scale              scan_
elements
in_accel_x_raw                  name                          subsystem
in_accel_x_scale                of_node                       uevent


$ cat /sys/bus/iio/devices/iio\:device0/in_accel_x_raw
421
$ cat /sys/bus/iio/devices/iio\:device0/in_accel_y_raw
-501
$ cat /sys/bus/iio/devices/iio\:device0/in_accel_z_raw
16626
$ cat /sys/bus/iio/devices/iio\:device0/in_accel_x_scale
0.000598
```

Then you could get the real measurement through this formula:

Input accel<x,y,z> (m/s^2)= in_accel_<x,y,z>_raw × in_accel_<x,y,z>_scale

For example:

Input accel x (m/s^2)= 421 × 0.000598=0,251758

9. USB

```
$ /opt/testing-tools/mtxstaroad2/test_usb.py
```

10. USB OTG

Connect a microusb cable to Host and open a session 115200-8-N-1 on the serial port.

```
$ /opt/testing-tools/mtxstaroad2/test_otg.py
```

11. IO Expander - TCA9539

First, you must know the sysfs gpio number for the SIGNALs.

J1 – MAIN			
PIN	**SIGNAL**	**TYPE**	**SYSFS GPIO**
3	OUT1	Output	505
4	OUT2	Output	506
5	OUT3	Output	507
6	OUT4	Output	508
13	IN1	Input	496
14	IN2	Input	497
15	IN3	Input	498
16	IN4	Input	499

AUX			
PIN	**SIGNAL**	**TYPE**	**SYSFS GPIO**
7	IN5	Input	500
8	IN6	Input	501
2	IN7	Input	502
3	IN8	Input	503

Then you could test the gpios according to the type. For example: OUT1 is the gpio 505.

```
$ echo 505 > /sys/class/gpio/export            #export gpio
$ echo out > /sys/class/gpio/gpio505/direction #set gpio output
direction
$ echo 1 > /sys/class/gpio/gpio505/value       #set gpio output value
$ cat /sys/class/gpio/gpio505/value            #get gpio output value
```

12. Analog to Digital Converter - ADC081C

Execute the following command to read ADC voltage raw code and scale. Then you could get the voltage measurement through this formula:

[Input voltage (mV)] = [in_voltage_raw] × [in_voltage_scale]

For example:

[Input voltage (mV)] = 164 × 12.5 = 2050

```
$ cat /sys/bus/iio/devices/iio\:device0/in_voltage_raw
164
$ cat /sys/bus/iio/devices/iio\:device0/in_voltage_scale
12.500000000
```

13. CAN Bus

Conect can0 and can1 buses, and execute the following commands in 2 terminals:

Terminal 1: This script will waiting for a frame from terminal 2

```
$ /opt/testing-tools/mtxstaroad2/test_can.py
```

Terminal 2: Send a frame

```
$ cansend can0 031#01020304  #can0 sender   and can1 receiver
$ cansend can1 030#01020304  #can0 receiver and can1 sender
```

14. HW ID – IO Expander

```
$ /opt/testing-tools/mtxstaroad2/test_hwid.py -id 17
```

15. Modem

```
$ /opt/testing-tools/mtxstaroad2/test_modem.py -p /dev/ttyACM0 -cmd
ati
```

16. GPS

The UART for GPS is the same UART for debug. Therefore, this test cannot executed in SERIAL session. And Getty is configurated to use the same console. Comment the following line in /etc/inittab before starting the test:

```
$ cat /etc/inittab | grep mxc0

#mxc0:12345:respawn:/bin/start_getty 115200 ttymxc0 vt102

$ /opt/testing-tools/mtxstaroad2/test_gps.py
```

NOTE: Make sure the jumper for debugging is not connected.

17. WiFi

```
$ ifconfig wlan0         # show the wifi interface

$ iwlist wlan0 scanning  # scan access points
```

18. RGB LED

Turn on only red LED:

```
$ /usr/bin/testing-tools/test_rgb.py -l r -p on
```

Turn on only green LED:

```
$ /usr/bin/testing-tools/test_rgb.py -l g -p on
```

Turn on only blue LED:

```
$ /usr/bin/testing-tools/test_rgb.py -l b -p on
```

Turn on red, green and blue LEDs:

```
$ /usr/bin/testing-tools/test_rgb.py -l w -p on
```

BUILD SDK

The Yocto environment allows to build SDK image. It makes ease the creation of applications by having compiler and debugger, and independent platform development.

To get the SDK image, execute the following command:

```
$ bitbake core-image-base -c populate_sdk
```

The compilation creates an executable file:

~/BSP/poky/build/tmp/deploy/sdk/poky-glibc-x86_64-core-image-base-cortexa9hf-neon-toolchain-2.4.3.sh

To install the SDK, copy this file into a development host and execute the following commands.

```
$ chmod +x ~/BSP/poky/build/tmp/deploy/sdk/poky-glibc-x86_64-core-image-base-cortexa9hf-neon-toolchain-2.4.3.sh

$ sh ~/BSP/poky/build/tmp/deploy/sdk/poky-glibc-x86_64-core-image-base-cortexa9hf-neon-toolchain-2.4.3.sh
```

NOTE: The default directory is /opt/poky/2.4.3/

REFERENCES

[1] http://wiki.solid-run.com/doku.php

[2] https://community.nxp.com/docs/DOC-94953

[3] https://www.yoctoproject.org/docs/2.4/bitbake-user-manual/bitbake-user-manual.html

[4] http://www.yoctoproject.org/docs/2.4/dev-manual/dev-manual.html#adt-eclipse

[5] http://trac.gateworks.com/wiki/Yocto/gstreamer/multimedia

[6] https://github.com/Freescale/gstreamer-imx

[7] https://gstreamer.freedesktop.org/

[8] https://yoctoproject.org/docs/2.4/ref-manual/ref-manual.html#structure-build-conf-local.conf

[9] https://www.yoctoproject.org/docs/2.4/ref-manual/ref-manual.html#ref-classes-extrausers

[10] https://www.kernel.org/doc/Documentation/gpio/sysfs.txt

SALES CONTACT

MATRIX MADRID

Matrix Electrónica S.L.
C/ Alejandro Sánchez 109
28019 Madrid, Spain

Phone 1: 902.19.81.46
Phone 2: +34-91.560.27.37
Fax 1: 902.99.54.14
Fax 2: +34-91.565.28.65

MATRIX BARCELONA

Matrix Electrónica S.L.
@SantCugat Business Park
Av. Via Augusta 15-25, B2- 8, oficinas 9-10
08174 Sant Cugat del Vallés, Barcelona, Spain

Phone: 902.19.81.46
Fax: 902.99.54.14

MATRIX DURANGO

Matrix Electrónica S.L.
Pol. Aliendalde 11, oficina 2G
48200 Durango, Vizcaya, Spain

Phone: 902.19.81.46
Fax: 902.99.54.14

MATRIX VALENCIA

Matrix Electrónica S.L.
Valencia, Spain

Phone: 902.19.81.46
Fax: 902.99.54.14

MATRIX SEVILLA

Matrix Electrónica S.L.
Sevilla, Spain

Phone: 902.19.81.46
Fax: 902.99.54.14

MATRIX LISBOA

LusoMatrix Lda.
Av. Coronel Eduardo Galhardo 7-1ºC
1170-105 Lisboa, Portugal

Phone: +351-218.162.625
Fax: +351-218.149.482

MATRIX EAU

Phone: +34-91.560.27.37
Fax: +34-91.565.28.65
Dubai

MATRIX APAC

9F, No. 156, Sec. 3, Minsheng E. Rd.,
Songshan Dist., Taipei City 10596, Taiwan
Phone: +886-965-333-367
Taiwan

MATRIX EEUU

Matrix USA LLD
1112 W. Boughton Rd, Suite 346
Bolingbrook, IL 60440
USA

mtxm2m.com