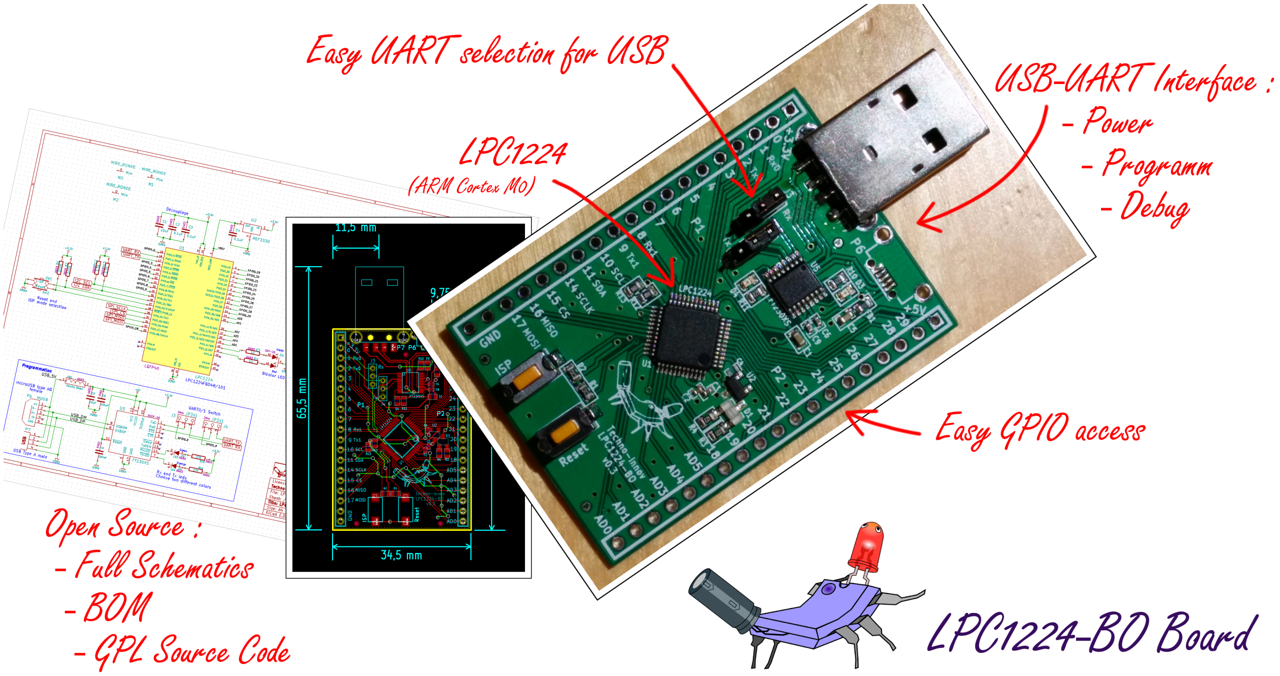


LPC1224 break-out board System Reference Manual

LPC1224_BO_SRM for Board v0.3

September 10, 2015 - Doc rev 0.3b

Author: Nathaël Pajani

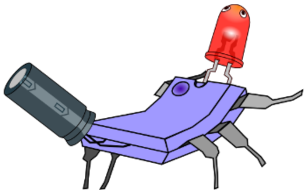


Table des matières

1 Introduction

You are reading the **System Reference Manual** for the LPC1224 break-out board. This manual covers the board use and design.

The LPC1224 break-out board is an electronics development and prototyping platform using the LPC1224 micro-controller from NXP [1].
The LPC1224 micro-controller has a Cortex-M0 ARM core, a minimum of 32KB of flash memory, 4KB of internal SRAM, and multiple interfaces.
The board also includes a bi-color user LED (Red / Green), a reset button, an ISP mode select / User button, an USB-to-UART bridge (used for programming and easy communication with the module), and 34 GPIO available on 2.54mm pins dispatched for easy use on prototyping boards.

Binaries for the LPC1224 break-out board can be generated using a gcc ARM toolchain and uploaded using the serial line (over USB thanks to the integrated USB-to-UART bridge) and our lpcprog tool (or similar tools).

The LPC1224 break-out board is designed for users interested in embedded ARM micro-controller development using free, libre and open source softwares only.
Every information about the design is available and all documentations are freely accessible. You can download the source files for the LPC1224 break-out board and modify them using KiCad EDA (GPL) according to the license terms found in the license section. You can create your own LPC1224 break-out board or a modified version.

In this document the LPC1224 break-out board will be referred as **the board**.

2 Licenses

2.1 Documentation

The present document is under Creative Commons CC BY-SA 3.0 License.
It is written in LaTeX and the PDF version is generated using pdflatex.

2.2 Hardware

The LPC1224 break-out board hardware and schematics are under Creative Commons CC BY-SA 3.0 License.
You can produce your own original or modified version of the LPC1224 break-out board, and use it however you like, even sell it for profit.

2.3 Software

All the software examples created for the LPC1224 break-out board are under GPLv3 License.
The lpcprog tool used to program the module is also under GPLv3 License.

[1] http://www.nxp.com/products/microcontrollers/cortex_m0_m0/LPC1224FBD48.html

3 Hardware

3.1 Dimensions

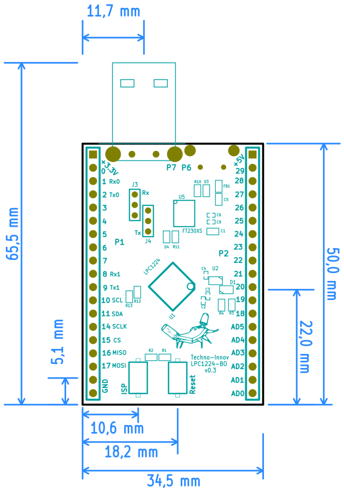


Fig 1 – USB A board type

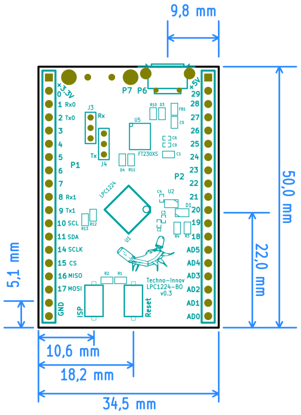


Fig 2 – micro USB board type

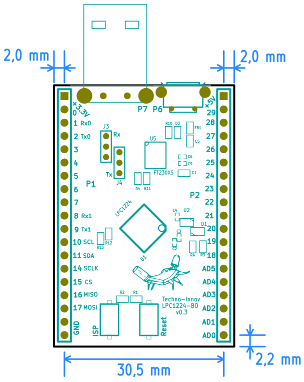


Fig 3 – Headers (2.54mm)

Figures 1, 2 and 3 give the different dimensions and the position of the main elements (connectors, buttons and user led) of the module.

The only difference between the "*USB A*" and the "*micro USB* board types is the USB connector.

Note : Not all components are
connector are the same on both b

3.2 Connectors

The module has two 2.54mm pitch headers numbered P1 and P2, and one USB connector, either P6 or P7 depending on the board type. Refer to figure 4 for connectors position and to table 1 for a short description. Detailed description of the signals found on each connector pin follow.

Name	Description
P1	19 pins, 2.54mm pitch header. Provides +3.3V, ground, UARTS, I2C, SPI, and GPIO from port 0.
P2	19 pins, 2.54mm pitch header. Provides +5V from USB, ADC, and GPIO from port 0 and 1.
P6	USB micro-AB female connector. Available only on micro USB board type.
P7	USB A male connector. Available only on USB A board type.

Table 1 – Module Connectors Description

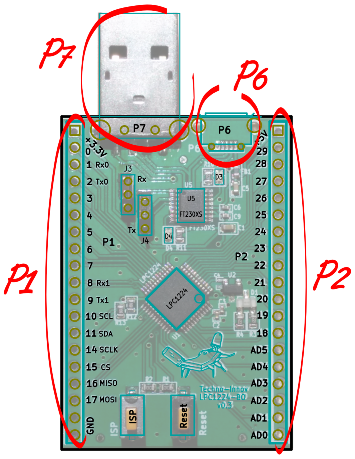


Fig 4 – Module Connectors

3.2.1 P1 Connector

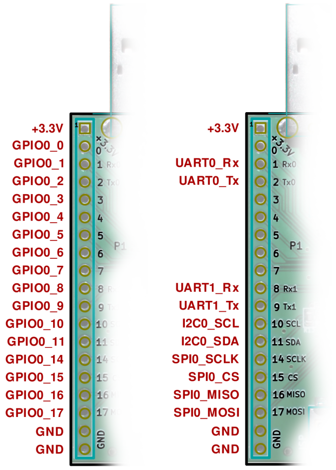


Fig 5 – P1 Connector

Pin #	Description	LPC Pin
1	+3.3V : +3.3 Volt	-
2	-	LPC pin 15 : PIO0_0
3	Rx0 : LPC UART_0 Receive Data	LPC pin 16 : PIO0_1
4	Tx0 : LPC UART_0 Transmit Data	LPC pin 17 : PIO0_2
5	-	LPC pin 18 : PIO0_3
6	-	LPC pin 19 : PIO0_4
7	-	LPC pin 20 : PIO0_5
8	-	LPC pin 21 : PIO0_6
9	-	LPC pin 22 : PIO0_7
10	Rx1 : LPC UART_1 Receive Data	LPC pin 23 : PIO0_8
11	Tx1 : LPC UART_1 Transmit Data	LPC pin 24 : PIO0_9
12	SCL : Clock for I²C bus	LPC pin 25 : PIO0_10
13	SDA : Bidirectional Serial Data for I²C bus	LPC pin 26 : PIO0_11
14	SCK : Clock for SPI bus	LPC pin 29 : PIO0_14
15	SSEL : Slave Select for SPI bus	LPC pin 30 : PIO0_15
16	MISO : Master In Slave Out for SPI bus	LPC pin 31 : PIO0_16
17	MOSI : Master Out Slave In for SPI bus	LPC pin 32 : PIO0_17
18	GND : Ground	-
19	GND : Ground	-

Table 2 – P1 Connector Pinout

P1 connector is a standard 2.54mm (0.1 inch) pitch header, with 1 row of 19 pins, and can be populated using either male or female header, and mounted either on top or on bottom of the board.
P1 connector provides access to +3.3V, Ground, UART0, UART1, I²C, SPI, and additional GPIO pins from port 0 of the LPC micro-controller.

3.2.2 P2 Connector

P2 connector is a standard 2.54mm (0.1 inch) pitch header, with 1 row of 10 pins, and can be populated using either male or female header, and mounted either on top or on bottom side of the board.
P2 connector provides access to USB +5V power supply, ADC, and GPIO pins from port 0 and 1 of the LPC micro-controller.

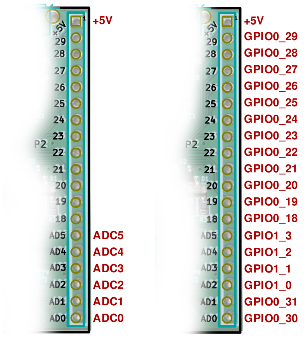


Fig 6 – P2 Connector

Pin #	Description	LPC Pin
1	+5V : +5 Volt from USB connector	-
2	-	LPC pin 14 : PIO0_29
3	-	LPC pin 13 : PIO0_28
4	-	LPC pin 12 : PIO0_27
5	-	LPC pin 11 : PIO0_26
6	-	LPC pin 10 : PIO0_25
7	-	LPC pin 9 : PIO0_24
8	-	LPC pin 8 : PIO0_23
9	-	LPC pin 7 : PIO0_22
10	-	LPC pin 6 : PIO0_21
11	-	LPC pin 5 : PIO0_20
12	-	LPC pin 4 : PIO0_19
13	-	LPC pin 33 : PIO0_18
14	ADC5	LPC pin 39 : PIO1_3
15	ADC4	LPC pin 38 : PIO1_2
16	ADC3	LPC pin 37 : PIO1_1
17	ADC2	LPC pin 36 : PIO1_0
18	ADC1	LPC pin 35 : PIO0_31
19	ADC0	LPC pin 34 : PIO0_30

TABLE 3 – P2 Connector Pinout

Note : Most P2 pins also provide alternate capture or match input functions for 32-bit timers. Refer to the LPC1224 documentation from NXP for full documentation of the alternate functions.

Note : When the board is not connected to a power source on the USB port the +5V is not present on pin 1 of P2 connector.

3.2.3 P6 and P7 Connectors

P6 and P7 are one-time choice options. Only one of them is present, depending on the board type. Both are standard USB connectors. P6 is a female micro-AB port, and P7 is a male USB-A port.
Refer to the Universal Serial Bus (USB) [2] page on Wikipedia for pinout and more information on the USB bus and connectors.

[2] http://fr.wikipedia.org/wiki/Universal_Serial_Bus

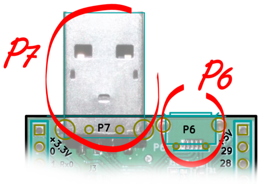


Fig 7 – P6 and P7 Connectors

3.3 Jumpers

The module has 2 configuration jumpers, numbered J3 and J4. Jumpers are common to all board types.

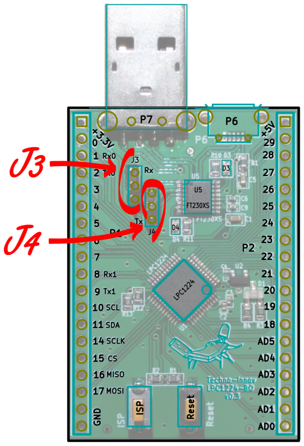


Fig 8 – Module Jumpers

Name	Description
J3 and J4	Selection between UART0 andUART1 for USB-to-UART bridge.

Table 4 – Module Jumpers Description

3.3.1 J3 and J4 jumpers

J3 and J4 jumpers are used to connect UART0 or UART1 pins to the FTDI FT230XS USB-to-UART bridge.

When the jumpers are on the top two pins for each jumper, then UART0 is connected to the USB-to-UART bridge. This position must be used for binary upload in ISP mode.

When the jumpers are on the bottom pins for each jumper, then UART1 is connected to the USB-to-UART bridge.

It is possible to operate the board without jumpers. Then the UAS-to-UART is not connected to the micro-controller.

Note : It is not mandatory for the jumpers to be on the same position on J3 and J4.

4 Electronics

The LPC1224 break-out board has been created using KiCad[3] EDA software suite for the creation of the schematics and printed circuit boards.

See page 19 in the annexes for the full schematics. The sources for the schematics are available for download from the module page[4] on Techno-Innov.fr.

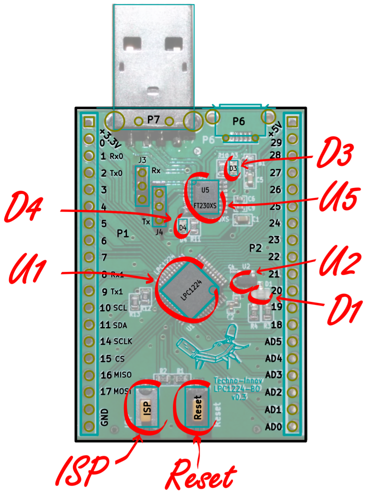


Fig 9 – Module Main Components

Name	Description
U1	LPC1224 ARM Cortex-M0 micro-controller.
U2	3.0V LDO reference voltage generator.
U5	FTDI FT230XS USB to UART bridge.
D1	User led, bicolore (red / green).
D3	Green led : FTDI Rx activity.
D4	Orange led : FTDI Tx activity..
Reset	Reset button for LPC1224 (SW2).
ISP	ISP mode select button for LPC1224 (SW1).

Table 5 – Module Main Components Description

4.1 Micro-controller LPC1224

The module's micro-controller is a LPC1224 from NXP[5]. The LPC1224 version used on the module is the LPC1224FBD48/101. All LPC1224 have an ARM Cortex-M0 core running at up to 45 MHz.

The module uses the internal 12 MHz RC Oscillator as main clock. Its 1% accuracy is suitable for most applications.

[3] http://www.kicad-pcb.org/display/KICAD/
[4] http://www.techno-innov.fr/technique-lpc1224-bo/
[5] http://www.nxp.com/products/microcontrollers/cortex_m0_m0/LPC1224FBD48.html

Note : Refer to the LPC1224 documentation from NXP for full list and documentation of the LPC1224 features. Here are only the descriptions of the features used on the module.

4.1.1 Internal RAM

The LPC1224FBD48/101 has 4kB of internal SRAM mapped in one block at address 0x1000 0000.

4.1.2 Internal Flash

The LPC1224FBD48/101 has 32kB of internal FLASH memory, mapped at address 0x0000 0000. The flash memory programming requires no additional hardware thanks to the In-System Programming (ISP) and In-Application Programming (IAP) on-chip bootloader software.

See section 4.1.6 (Reset and ISP mode) or sections 5.2 (Code Compilation) and 5.3 (Uploading binary on target) for more information on internal FLASH memory.

4.1.3 Communication interfaces

The module makes use of the following communication interfaces found on the LPC1224 :

- **Two UARTs** : UART0 and UART1 are connected to P1 header and to the USB to UART bridge through J3 and J4 jumpers.
  UART0 is used for In-System Programming of the LPC1224.
- **One I²C bus interface** supporting full I²C-bus specification and Fast-mode Plus with a data rate of 1 Mbit/s. I²C is connected to P1 header. See section 4.2 for more information.
- **One SSP/SPI controller** with FIFO and multi-protocol capabilities. The SPI bus is also connected to P1 header.

4.1.4 GPIO

The module gives access to 34 GPIO pins dispatched on P1 and P2 connectors.

Refer to tables 2 and 3 for details of the signals available on these GPIO and to the LPC1224 documentation from NXP for full list of features for each GPIO.

4.1.5 ADC

The last six GPIO pins on P2 connector are inputs channels 0 to 5 for the 10-bit ADC of the LPC1224 microcontroller.

The internal ADC uses the voltage on the Vref pin as reference voltage for the conversion. The LPC1224 break-out board integrates a 3.0V reference voltage LDO to provide a stable and accurate Vref to the LPC1224.

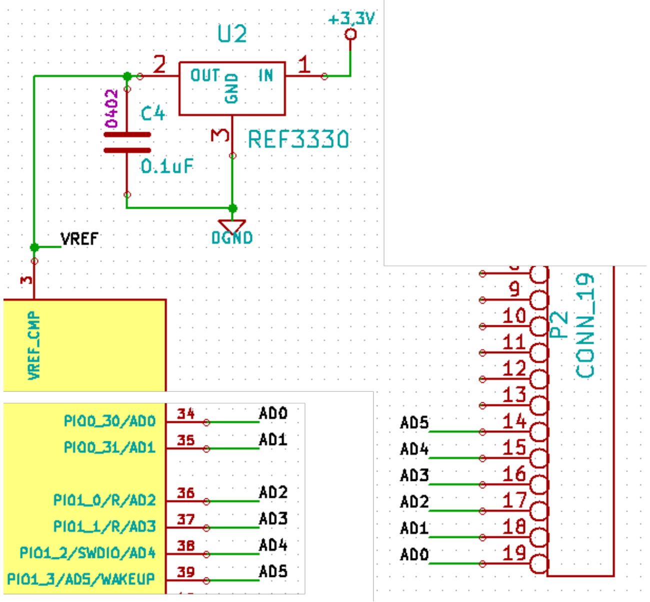


Fig 10 – ADC Input Pins and Vref

4.1.6 Reset and ISP mode

Reseting the LPC1224 without removing the power can be done with the Reset button (SW2).

To enter In-System Programming (ISP) mode after reset you must hold the ISP button (SW1) when you release the reset button. The LPC1224 bootloader considers a LOW level on the PIO0_12 pin as an external hardware request to enter ISP mode and start the ISP command handler. The sampling of the GPIO0_12 pin may take up to 3ms.

Refer to section 5.3 or to the LPC1224 user manual for more information on ISP mode.

If the ISP button is not held when th Reset button is released (and a valid user code is found in Flash memory) then the execution is transfered to the user program.

4.2 I^{2}C

The LPC1224 break-out board has no components on the I^{2}C bus, leaving the whole address space available for the user. The LPC1224 break-out board provides two 1.5kOhms pull-up resistors on both SDA and SCL lines.

4.3 User Led and Button

The module has three leds and two buttons. The two leds connected to the USB to UART bridge (D3 and D4) and the Reset button have dedicated functions and cannot be assigned other functions.
The remaining led (D1) and button (ISP) can be used as the user wishes.

The D1 led is a bi-color red / green led connected to PIO1_4 (pin 40) and PIO1_5 (pin 41). Both can be turned on at the same time, providing a third color (orange).

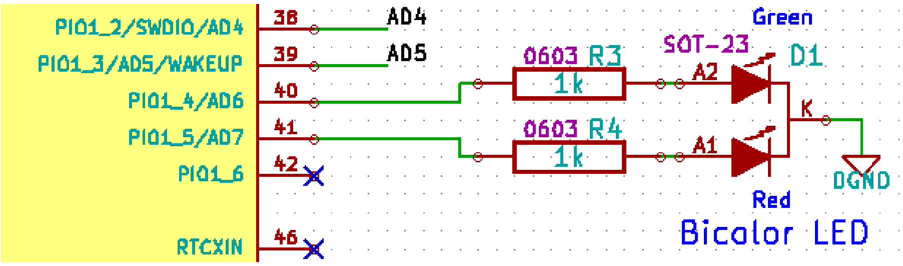


Fig 11 – User Led

Note : The PIO1_4 and PIO1_5 pins are not PWM capable so it's not possible to create shades between red and green without using a lot of processing power.

After reset the ISP button can be used by the user to any purpose. It's state can be read on pin PIO0_12 (pin 27).

4.4 USB to UART bridge

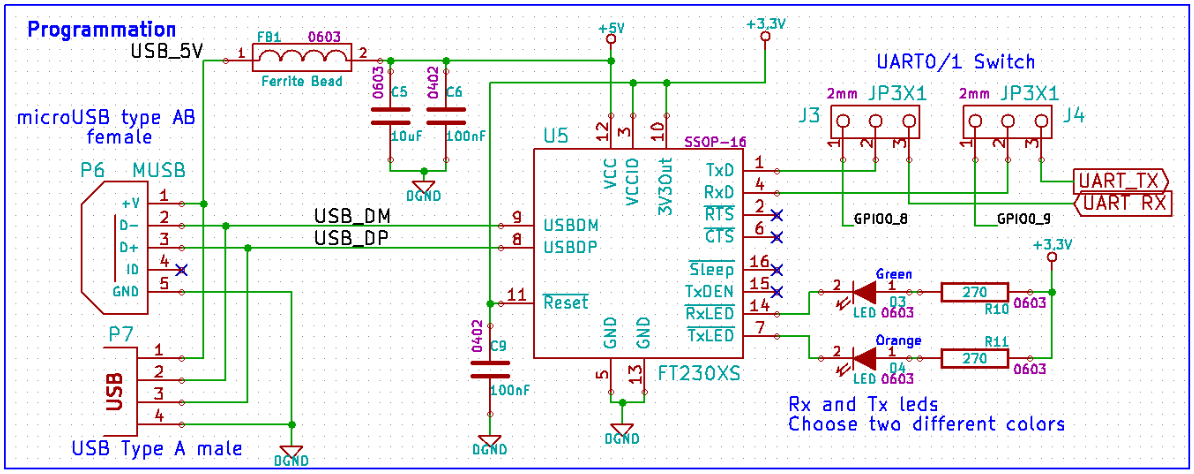


Fig 12 – USB to UART bridge

In order to ease the development process and the use of the module we added a USB to UART bridge on-board. This bridge is made by a FTDI FT230XS chip. It provides a 3.3V regulated voltage for the module and is well supported on most operating systems so there is usually no configuration required to use it as a serial line on the host development system, removing the need of any additional power source or of specific hardware to program the LPC1224 micro-controller and communicate with the module.

The FTDI chip controls two "activity" leds for Rx (D3, the green one) and Tx (D4, orange one) data over the serial link.

Removing jumpers J3 and J4 disconnects the USB to UART bridge from the LPC1224 micro-controller.

5 Software

The LPC micro-controller family uses ARM cores, which make them very easy to use. Apart from a few wrappers, all the code can be written in C and compiled using gcc.

ARM, NXP and other vendors provide sample code, but published under many different licences. The code we provide for the LPC1224 break-out board is published under the well known GPLv3 licence.

5.1 Sample Source Code

5.1.1 Grab the sources

An example application code can be downloaded from our git repository [6] using the following clone command :

```
user@host:~/sw$ git clone http://gitclone.techno-innov.fr/modules
```

The LPC1224 break-out board can use the code from the `apps/base` sub-directory as there are very few differences between the LPC1224 break-out board and the GPIO Demo module. The only differences are the lack of EEPROM memory and temperature sensor on the I^2C bus and the SPI CS pin which is not used for I^2C clock activation for the EEPROM and thus fully available to the user.

5.1.2 Sample code content

This code provides the micro-controller definitions (Cortex-M0 specific definitions, registers, interrupts ...) and the routines required to start the micro-controller (bootstrap, vector table, power state, flash, clocks).

At the time of writing it also provides a basic set of library functions and the drivers for the interfaces found on the module. The list of supported features and interfaces is updated as the development goes on, so read the README file for the full list of supported features and interfaces.

The code has been split in five parts : `core/`, `drivers/`, `extdrv/`, `apps/` and `lib/` (with the associated directories under `include/` for the headers) :

- `core` : Contains all the required parts and system initialisations. Many functions in there are defined as weak aliases of dummy functions, so the code compiles even if no drivers are used. When these functions are redefined in the driver code they override the weak definition.
- `lib` : Contains the implemented parts of the small C library for our code. The micro-controller does not run a full Linux system, so the gnu libc must not be used, and even a µClibc is much more than what's required. Most of the code in these files come from the kernel implementations of libc parts.
- `drivers` : Contains the drivers for the different interfaces found on the module.
- `extdrv` : Provides drivers for external components, either on the module (EEPROM, status led, tmp101 temperature sensor), or to be purshased separately and connected to the module.
  The number of external parts supported will grow with time. Note that it may not be possible to use all of them at the same time.
- `apps` : Provides sample applications for the different modules made by Techno-Innov which demonstrate either LPC1224 interfaces or external drivers, which can be used as base for your own developments.
  Most modules will have their own directory under `apps/`. The LPC1224 break-out board uses the `base/`

[6] http://git.techno-innov.fr/?p=modules;a=summary

subdirectory, which is common with the GPIO Demo module. Creating a new app is as easy as creating a new sub-directory under apps/base (with no spaces or special caracters in the name), copying the Makefile from one of the other apps in your new app directory, and creating your own C source file(s) (maybe starting with a copy of an existing example).
If you created a new module, you should consider creating a new "module" directory under apps/ with it's own sub-directories for specific apps.

5.1.3 Sample code entry point

The main loop is in main.c in function main(), as with any C program, though main() is called by the bootstrap code (Reset_Handler() in core/bootstrap.c) and could have any name.
The calls to the system initialisation routines have been put together in the system_init() function.
SELECTED_FREQ must be set to one of the FREQ_SEL_**MHz defined in include/core/system.h :

- FREQ_SEL_12MHz
- FREQ_SEL_24MHz
- FREQ_SEL_36MHz
- FREQ_SEL_48MHz
- FREQ_SEL_60MHz

Note : The frequency can go up to 60MHz despite what is said in the documentation, but the micro-controller needs much more power at higher frequencies.

Note : system.h provides two sleep functions (msleep() and usleep()). These will activate the systick with a 1ms tick if it has not been done yet.

Note : usleep() ma

The pins used by your application should be configured using either the set_pins() function and pio_config structures (see common_pins[] in most examples) or the config_gpio() function for each used pin.

5.2 Building the binary

5.2.1 Get a toolchain

Build has been tested using gcc, and only gcc, in the version provided by the Debian project[7], but any ARM gcc toolchain should do.

In order to get the Debian ARM gcc cross-toolchain you must install package gcc-arm-none-eabi.
There's no need for the related libc package here, the libc does not fit in our micro-controller memory. Instead have a look at the content of the lib/ directory, and add stuff there.

Alternatively you can download pre-compiled gcc toolchains (many different projects provide their own), or build your own one using crosstool-ng[8] or similar projects. For more information on what is a (cross-)toolchain, have a look at this information page on elinux.org[9].

[7] http://www.debian.org/
[8] http://crosstool-ng.org/
[9] http://elinux.org/Toolchains

5.2.2 Build command and options

Once done with the toolchain installation (or if you already have one) you should build using the provided `Makefile` by running the simple "`make`" command from any of the `apps/base/` sub-directory.
You can also build all apps by running "`make`" from repository root directory or "`make base/my_app`" to compile "`my_app`" from the `apps/base/` sub-directory only. (`base` may be replaced by any other module name). NoteYou may want to change the `CROSS_COMPILE` variable from the main `Makefile` (in the repository root directory) and set it to the prefix of your toolchain.

5.2.3 Build process

The specific information about the target (LPC1224 micro-controller) memory (Flash and RAM) used by the linker is in the `lpc_link_lpc1224.ld` linker script.

The vector table is defined in the `core/bootstrap.c` file, but the checksum of the first seven entries in the vector table is left unmodified. This checksum must be computed and placed in the eighth vector entry as the bootloader needs to find a valid checksum in the eighth entry to consider the user code as valid and transfer execution to the reset handler (first vector table entry).
This is done by the lpcprog tool before sending the binary to the target.

5.3 Uploading binary on target

5.3.1 Tools

To flash the binary (the one with .bin) to the LPC Flash you can use our lpctools package, packaged for Debian as of 2014-09-10 and available in jessie or newer versions, or available in the lpctools git repository [10].
Lpctools is released under GPLv3 licence.
Clone the repository using :

```
user@host:~/sw$ git clone http://gitclone.techno-innov.fr/lpctools
```

Then build (`make`) the tools.

Note : Other tools may be used but have not been tested. No tools were found to be open source when we looked for tools to upload the binaries to the micro-controller. You must check that the tool you chose to use can take care of the checksum computation.

5.3.2 Connection with target and upload

Usual command line to upload a binary to the micro-controller :

[10] http://git.techno-innov.fr/lpctools

```
user@host:~/sw$ lpcprog -d /dev/ttyUSB0 -c flash mod_gpio.bin
Part ID 0x3640c02b found on line 18
Flash now all blank.
Checksum check OK
Flash size : 32768, trying to flash 8 blocks of 1024 bytes : 8192
Writing started, 8 blocks of 1024 bytes ...
user@host:~/sw$
```

If you want to get information on the connected device use the **id** command of **lpcprog** :

```
user@host:~/sw$ lpcprog -d /dev/ttyUSB0 -c id
Part ID 0x3640c02b found on line 18
Part ID is 0x3640c02b
UID: 0x1228f5f5 - 0x4b324307 - 0x08333834 - 0x4d7b2c1a
Boot code version is 1.6
user@host:~/sw$
```

Note : The part information definition for each supported micro-controller is in the lpctools package.
See lpctools readme and lpcprog or lpcisp help (-h option) or manpages for more information.

6 Board revisions history

6.1 v01

This board revision has not been sold.
First prototype version, with Rx and Tx crossed for UART1, making it impossible to connect them to USB to UART bridge.

6.2 v02

This board revision has not been sold.
Added 3.0V LDO voltage regulator for Vref.

6.3 v03

Actual version sold as of writting of this documentation.
Added a 100nF cap to Vref pin.

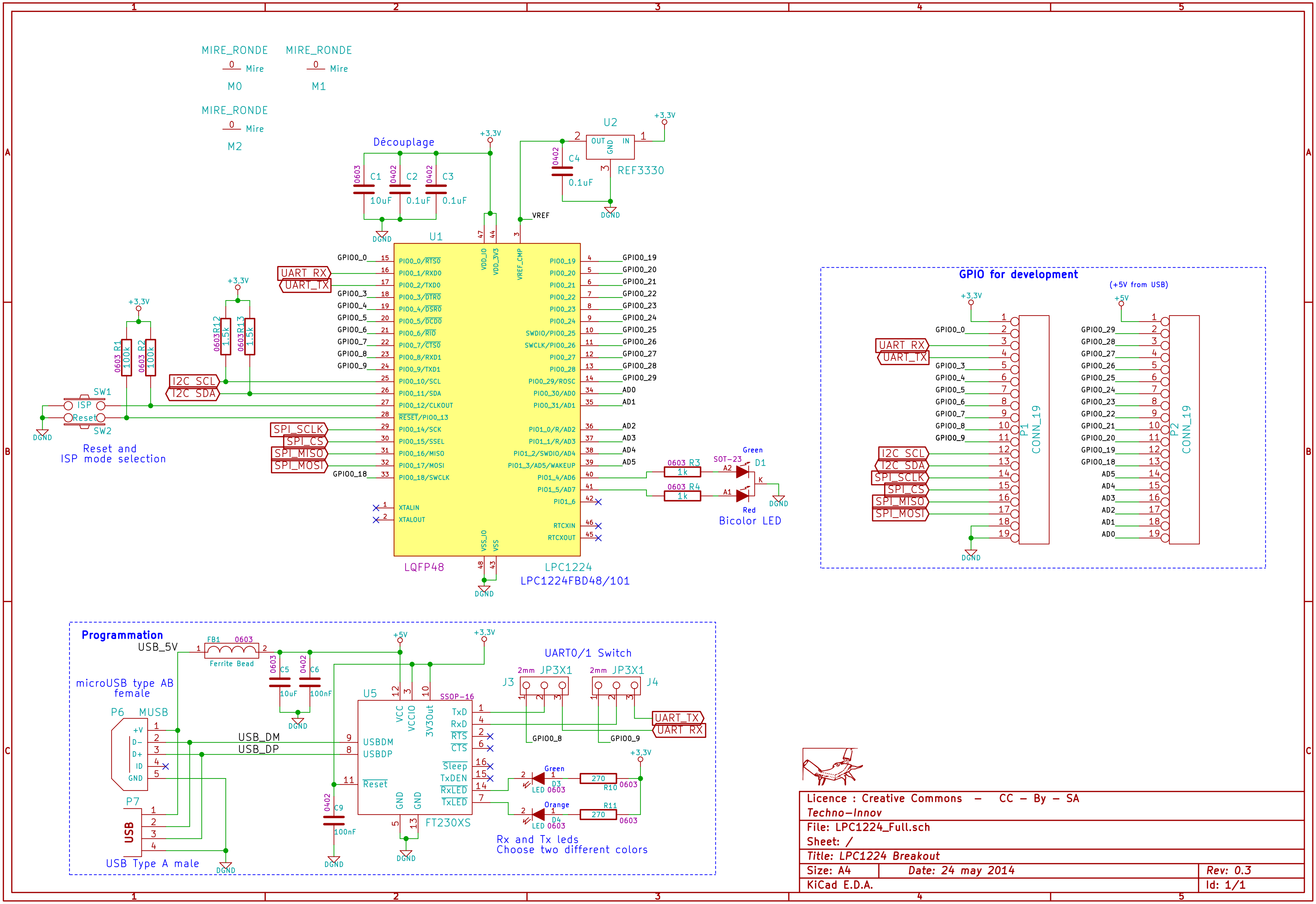
7 Annexes

7.1 Schematics

The board schematics and PCB layout have been created using KiCad [11] EDA software suite. You can download the sources on the module page [12] on Techno-Innov.fr.

(See on next pages)

[11] http://www.kicad-pcb.org/display/KICAD/
[12] http://www.techno-innov.fr/technique-lpc1224-bo/



Licence : Creative Commons – CC – By – SA

Techno–Innov

File: LPC1224_Full.sch

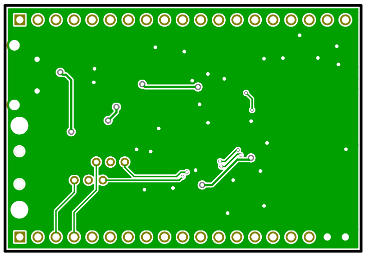
Sheet: /

Title: LPC1224 Breakout

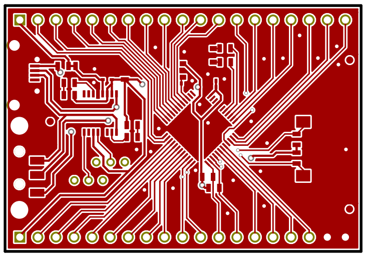
Size: A4 | *Date: 24 may 2014* | *Rev: 0.3*

KiCad E.D.A. | Id: 1/1

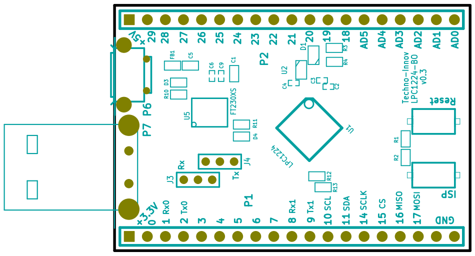
Bottom copper layer



Top copper layer



Top silk screen



Techno-Innov

File: LPC1224_Full.kicad_pcb

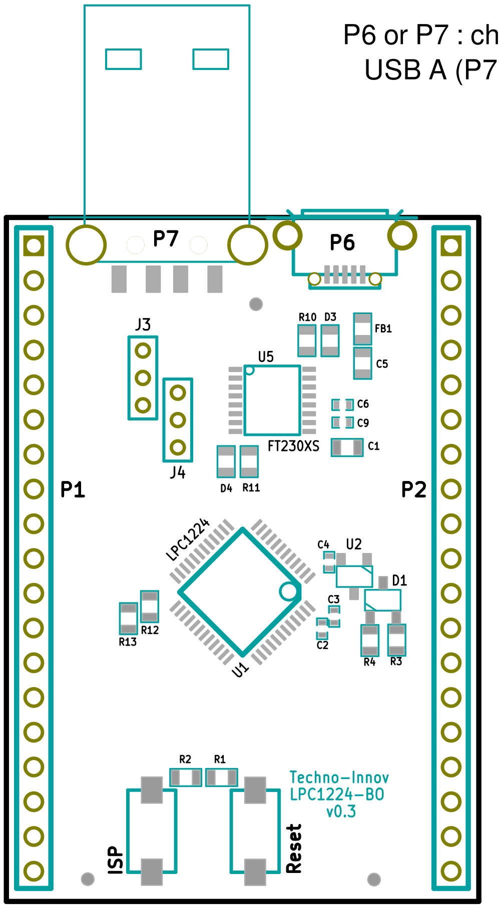
Sheet:

Title: LPC1224 Breakout

Size: A4	Date:	Rev: 0.3
KiCad E.D.A.		Id: 1/1

Licence : Creative Commons – CC – By – SA – NC
Techno-Innov
File: LPC1224_Full.kicad_pcb
Sheet:
Title: LPC1224 Breakout
Size: A4 Date: Rev: 0.3
KiCad E.D.A. Id: 1/1

P6 or P7 : chose between
USB A (P7) or USB microB (P6)



D3 and D4 :
The cathode (-) is on the U5 side
D3 : Green
D4 : Orange

J3 and J4 : top mount

P1 and P2 : top or bottom mount

7.2 BOM

7.2.1 Block version

Part Description	Ref	Module	Nb	Vendor	Vendor ref	Farnell ref
Micro-controller						
LPC1224	U1	LQFP48	1	NXP	LPC1224FBD48/101	1862465
Decoupling capacitors 100nF	C2, C3, C4	0402	3	Multicomp	MCCA000050	1758896
Filter capacitor 10µF	C1	0603	1	TDK	C1608X5R0J106M	2112705
Pull-Up resistors 100k Ohms	R1, R2	0603	2	Multicomp	MC0063W06031%100K	9330402
I^2C Pull-Up resistors 1,5k Ohms	R12, R13	0603	2	Multicomp	MC0063W06031%1K5	9330607
Bi-color Led resistors 270 Ohms	R3, R4	0603	2	Multicomp	MC0063W06031%270R	9330917
SMD Led Red / Green	D1	SOT-23	1	Kingbright	KM-23ESGW	1142614
SMD switchs	SW1, SW2		2	Multicomp	DTSM-32S-B	9471898
3.0V Vref	U2	SOT-23	1	Texas Instrument	REF3330AIDBZT	1755084
USB Bridge						
Led resistors 270 Ohms	R10, R11	0603	2	Multicomp	MC0063W06031%270R	9330917
Decoupling capacitors 100nF	C6, C9	0402	2	Multicomp	MCCA000050	1758896
Filter capacitor 10µF	C5	0603	1	TDK	C1608X5R0J106M	2112705
SMD chip bead	FB1	0603	1	TDK	MMZ1608R601A	1669700
Rx Led - Green	D3	0603	1	Vishay	VLMG1300-GS08	2251461
Tx Led - Orange	D4	0603	1	Vishay	VLMO1300-GS08	2251473
FT230XS USB to UART	U5	16SSOP	1	FTDI	FT230XS	2081321
Micro-USB type A-B female	P6	SMD	1	Molex	47590-0001	1568022
USB Type A male	P7	SMD	1	Multicomp	MC32605	1696546
Male headers 1x3	J3, J4	2mm	2	Fisher	SLY1.085.50G	9729135
Jumpers 2mm black			2	Harwin	M22-1900005	510932
GPIO Connectors						
Male headers GPIO (2 x 19 pins)	P1, P2	2,54mm	2	Fischer	SL1.025.36Z	9729038

TABLE 6 – BOM by functional block

Note : Components used on Board may change for functionnally equivalent references without prior notice

7.2.2 Easy order version

Part Description	Ref	Module	Nb	Vendor	Vendor ref	Farnell ref
LPC1224	U1	LQFP48	1	NXP	LPC1224FBD48/101	1862465
FT230XS USB to UART	U5	16SSOP	1	FTDI	FT230XS	2081321
SMD chip bead	FB1	0603	1	TDK	MMZ1608R601A	1669700
Decoupling capacitors 100nF	C2, C3, C4, C6, C9	0402	5	Multicomp	MCCA000050	1758896
Filter capacitor 10µF	C1, C5	0603	2	TDK	C1608X5R0J106M	2112705
Pull-Up resistors 100k Ohms	R1, R2	0603	2	Multicomp	MC0063W06031%100K	9330402
I^{2}C Pull-Up resistors 1,5k Ohms	R12, R13	0603	2	Multicomp	MC0063W06031%1K5	9330607
Led resistors 270 Ohms	R3, R4, R10, R11	0603	4	Multicomp	MC0063W06031%270R	9330917
SMD Led Red / Green	D1	SOT-23	1	Kingbright	KM-23ESGW	1142614
Rx Led - Green	D3	0603	1	Vishay	VLMG1300-GS08	2251461
Tx Led - Orange	D4	0603	1	Vishay	VLMO1300-GS08	2251473
SMD switchs	SW1, SW2		2	Multicomp	DTSM-32S-B	9471898
Micro-USB type A-B female	P6	SMD	1	Molex	47590-0001	1568022
USB Type A male	P7	SMD	1	Multicomp	MC32605	1696546
Male headers 1x3	J3, J4	2mm	2	Fisher	SLY1.085.50G	9729135
Jumpers 2mm black			2	Harwin	M22-1900005	510932
Male headers GPIO (2 x 19 pins)	P1, P2	2,54mm	2	Fischer	SL1.025.36Z	9729038

Table 7 – BOM by reference

Note : Components used on Board may change for functionnally equivalent references without prior notice

7.3 Document revision History

Version	Date	Author	Information
0.3a	April 12, 2015	Nathaël Pajani	Initial revision
0.3b	September 10, 2015	Nathaël Pajani	Changes according to code organisation modifications

7.4 Disclaimer

The LPC1224 break-out board is provided "as is" without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the LPC1224 break-out board is with you. Should the LPC1224 break-out board prove defective, you assume the cost of all necessary servicing, repair or correction.