

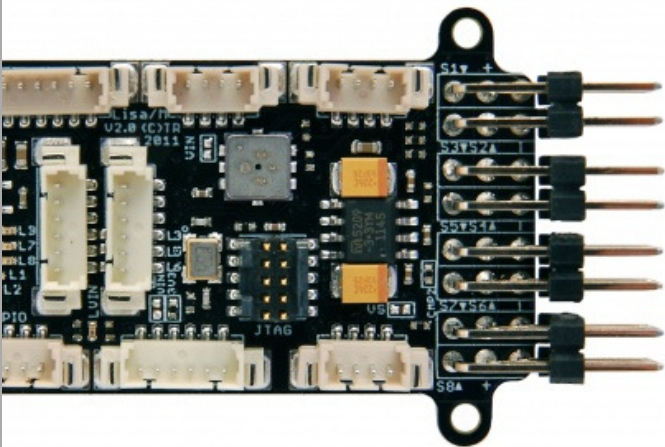
Lisa/M v20

From Paparazzi

(Redirected from Lisa/M)

Contents

- 1 Hardware Revision History
- 2 Features
- 3 Pinout
 - 3.1 Jumper Configuration
 - 3.2 Powering the Board
- 4 Schematic
- 5 Examples of Airborne Equipment Electrical Connections
 - 5.1 Small Aircraft Connection Diagram
 - 5.2 Large Aircraft Connection Diagram
 - 5.3 R/C Receivers
- 6 PCB
 - 6.1 Gerber & Drill Files
- 7 Assembly
 - 7.1 Components Layout
 - 7.2 Bill Of Material
- 8 PCB and assembled boards suppliers
- 9 Mechanical Dimensions
- 10 Downloads
- 11 Programming
 - 11.1 Using luftboot
 - 11.2 Uploading the Paparazzi USB Bootloader
 - 11.2.1 Required components
 - 11.2.2 Procedure
 - 11.2.3 Connection Diagram
 - 11.2.4 Boot Sequence
 - 11.3 JTAG
 - 11.3.1 Quick JTAG Upload Guide
 - 11.4 Serial Firmware Upload
- 12 Detailed Hardware Revision History
 - 12.1 Changes Between v1.1 and v2.0
 - 12.2 Changes Between v1.0 and v1.1
 - 12.3 Changes Between v0.1 and v1.0
 - 12.4 Hardware Change Requests



- [−] Autopilots
- [+] Booz
- [+] HB Autopilots
- [+] Lisa
- [+] Tiny
- [+] YAPA Autopilots
- Ap template*
- Booz*
- Classix*
- Future Autopilots*
- Previous Autopilots*
- TWOG/v1.0*
- Umarim v10*

Lisa/M is a small, general purpose autopilot designed with flexibility across multiple applications in mind. Small weight and size, with (optional) integrated Aspirin IMU and full size 0.1" servo headers make the Lisa/M suitable for both fixed-wing and rotorcraft vehicles. This autopilot is based on the STM32 for improved peripherals and faster processing.

A number of tutorials are being prepared for getting started with Lisa/M:

- Fixedwing tutorial
- Rotorcraft tutorial

Hardware Revision History

Version #	Release Date	Release Notes
-----------	--------------	---------------

v2.0(current)	03/2012	Updated Production Release
v1.1	MM/YYYY	Updated Prototype
v1.0	MM/YYYY	Initial Production Release
v0.1	MM/YYYY	Initial prototype of Lisa/M

For detailed hardware revision history, please see below.

Features

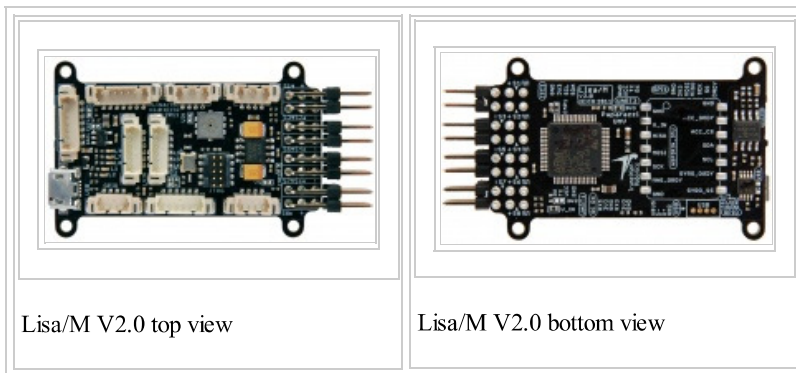
Lisa/M is based on the 64 pins STM32F105RCT6 connectivity line family (<http://www.st.com/internet/mcu/product/221023.jsp>) processor featuring 64k of RAM and 256k of FLASH. All the pins are exposed, providing access to the complete set of the STM32 peripherals. NOTE: This MCU is different from LISA/L. Lisa/L is based on the 64 pins STM32F103RE processor featuring 64k of RAM and 512k of FLASH, which is part of the high-density performance line family (<http://www.st.com/internet/mcu/product/164485.jsp>) .

- STM32 microcontroller STM32F105RCT6 datasheet (http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/DATASHEET/CD00220364.pdf) with 256kB flash and 64kB RAM
- Pressure sensor BMP085 (<http://www.bosch-sensortec.com/content/language1/html/3477.htm>)
- 7 x Analog input channels
- 3 x Generic digital in-/out-puts
- 2 x 3.3V TTL UART (5V tolerant)
- 8 x Servo PPM outputs (6 w/ second I2C bus in use)
- 1 x CAN bus
- 1 x SPI (http://en.wikipedia.org/wiki/Serial_Peripheral_Interface) bus
- 1 x I²C (<http://en.wikipedia.org/wiki/I2c>) bus (2 when using only 6 Servo PPM outputs)
- 1 x Micro USB
- 4 x status LEDs with attached test point
- 10.8 grams (0.4 oz) (with Aspirin IMU mounted)
- 9.9 grams (0.35 oz) (without Aspirin IMU mounted)
- ~34mm x ~60mm x ~10mm
- 4 layers PCB design

with mounted IMU has the following additional sensors on board:

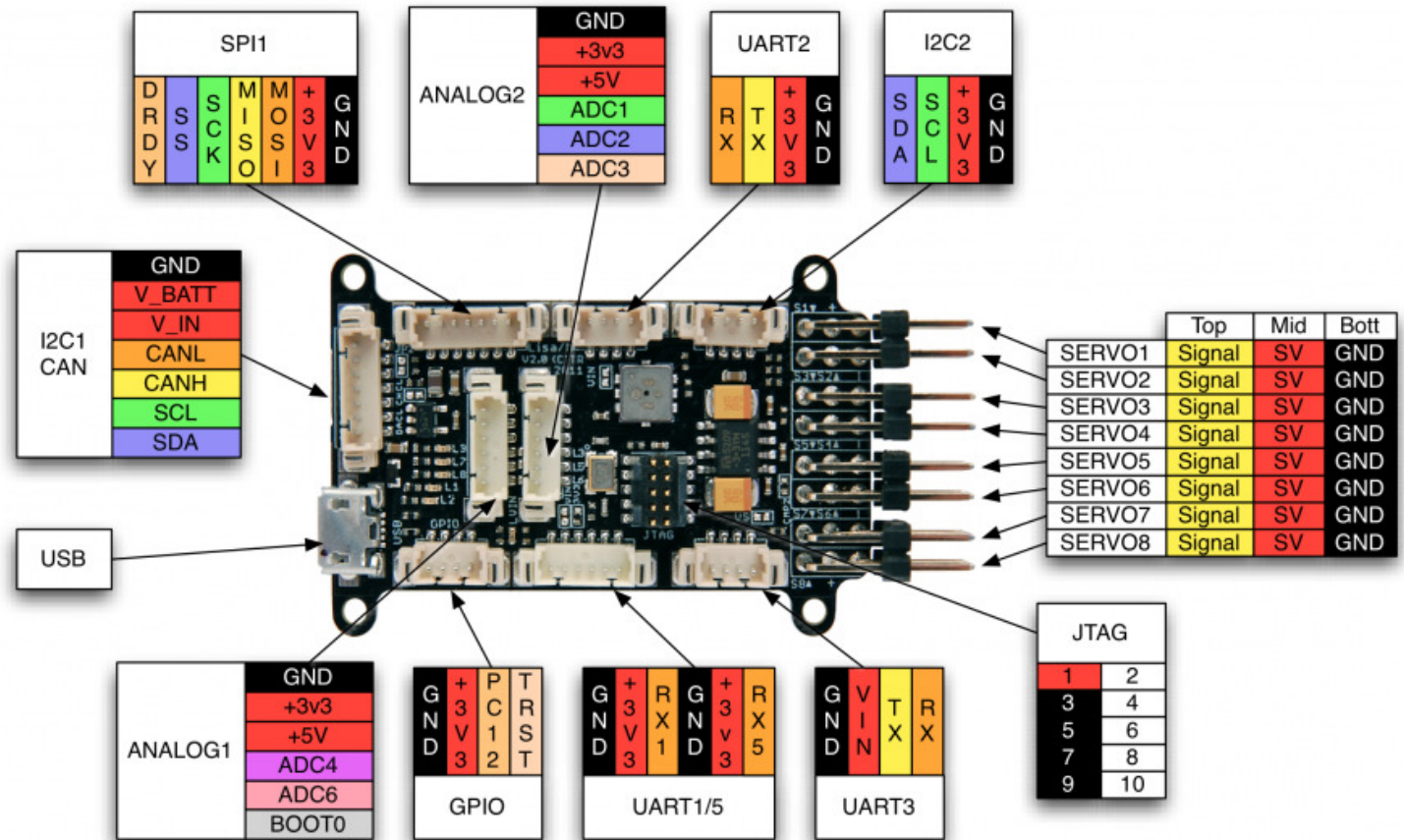
- 3 Axis Gyroscope
- 3 Axis Accelerometer
- 3 Axis Magnetometer

The pressure sensor is mounted directly on the board as this sensor is not provided by Aspirin (*v0.1* - *v1.1*). Except for a GPS unit you have all necessary sensors for full attitude and altitude stabilization in an extremely small package.



Pinout

Pins Name and Type are specified with respect to the Autopilot Board.



Never exceed 5V
power supply limit!

UART3 GPS connector
is connected to V_IN check your GPS
input voltage before connecting!

SERVO1/2/3/4/5/6/7/8

Pin #	Name	Type	Description	Color
1	SERVOx	OUT	Servo signal (PWM)(See Note 1 below)	Yellow
2	SV	PWR	Servo Bus Voltage Rail (conf w/ JP1)	Red
3	GND	PWR	common ground	Black

JTAG

Pin #	Name	Type	Description	Color
1	N/A	N/A	JTAG Debug Header (Pin 1 is +3V3)	None

UART3

Pin #	Name	Type	Description	Color
1	GND	PWR	common ground	Black
2	V_IN	PWR	UART Voltage (conf w/ JP6 and JP7)	Red
3	TX	OUT	USART3 Serial Output (3.3V level)	Yellow
			USART3 Serial Input (3.3V level)(Pullup to Pin 2 voltage)(5V	

4	RX	IN	tolerant)	Orange
---	----	----	-----------	--------

UART1/5

<i>Pin #</i>	<i>Name</i>	<i>Type</i>	<i>Description</i>	<i>Color</i>
1	GND	PWR	common ground	Black
2	+3V3	PWR	3.3V Rail from autopilot (conf w/ JP8 and JP9)	Red
3	RX1	IN	USART1 Serial Input (3.3V level)(Pullup to Pin 2 voltage)(5V tolerant)	Orange
4	GND	PWR	common ground	Black
5	+3V3	PWR	3.3V Rail from autopilot (conf w/ JP8 and JP9)	Red
6	RX5	IN	UART5 Serial Input (3.3V level)(Pullup to Pin 5 voltage)(5V tolerant)	Orange

GPIO

<i>Pin #</i>	<i>Name</i>	<i>Type</i>	<i>Description</i>	<i>Color</i>
1	GND	PWR	common ground	Black
2	+3V3	PWR	3.3V Rail from autopilot	Red
4	PC12	I/O	GPIO, connected to PC12 (5V tolerant)	Dark Tan
5	TRST	I/O	JTAG_TRST (also connected to LED1 cathode)	Light Tan

ANALOG1

<i>Pin #</i>	<i>Name</i>	<i>Type</i>	<i>Description</i>	<i>Color</i>
1	GND	PWR	common ground	Black
2	+3V3	PWR	3.3V Rail from autopilot	Red
3	+5V	PWR	5V Rail from autopilot	Red
4	ADC4	I/O	ADC4, by default connected to LED7 cathode (Remove LED/resistor to use as ADC)	Magenta
5	ADC6	I/O	ADC6, by default connected to LED8 cathode (Remove LED/resistor to use as ADC)	Pink
6	BOOT0	I/O	BOOT0	Grey

USB

<i>Pin #</i>	<i>Name</i>	<i>Type</i>	<i>Description</i>	<i>Color</i>
1	N/A	N/A	USB (The USB connections are also available as 0.05" (1.27mm) through hole pads underneath the GPIO header)	None

I2C1 CAN

<i>Pin #</i>	<i>Name</i>	<i>Type</i>	<i>Description</i>	<i>Color</i>
1	GND	PWR	common ground	Black
2	V_BATT	PWR	V_BAT Bus on autopilot, voltage divider for V_BAT_MEAS, (conf w/ JP2 to connect to V_IN)	Red
			Connected to autopilot voltage regulator inputs (conf w/ JP1, JP2)	

3	V_IN	PWR	and JP3)	Red
4	CANL	I/O	CANL (5V level)	Orange
5	CANH	I/O	CANH (5V level)	Yellow
6	SCL	I/O	SCL (5V level)(See Note 1 below)	Green
7	SDA	I/O	SDA (5V level)(See Note 1 below)	Blue

SPI1

<i>Pin #</i>	<i>Name</i>	<i>Type</i>	<i>Description</i>	<i>Color</i>
1	GND	PWR	common ground	Black
2	+3V3	PWR	3.3V Rail from autopilot	Red
3	MOSI	Out	MOSI	Orange
4	MISO	In	MISO	Yellow
5	SCK	Out	SCK	Green
6	SS	Out	SS	Blue
7	DRDY	I/O	DRDY	Dark Tan

ANALOG2

<i>Pin #</i>	<i>Name</i>	<i>Type</i>	<i>Description</i>	<i>Color</i>
1	GND	PWR	common ground	Black
2	+3V3	PWR	3.3V Rail from autopilot	Red
3	+5V	PWR	5V Rail from autopilot	Red
4	ADC1	In	ADC1	Green
5	ADC2	In	ADC2	Blue
6	ADC3	In	ADC3	Light Tan

UART2

<i>Pin #</i>	<i>Name</i>	<i>Type</i>	<i>Description</i>	<i>Color</i>
1	GND	PWR	common ground	Black
2	+3V3	PWR	UART Voltage (conf w/ JP4 and JP5)	Red
3	TX	OUT	USART2 Serial Output (3.3V level)	Yellow
4	RX	IN	USART2 Serial Input (3.3V level)(NOT 5V TOLERANT) (Pullup to Pin 2 voltage)	Orange

I2C2

<i>Pin #</i>	<i>Name</i>	<i>Type</i>	<i>Description</i>	<i>Color</i>
1	GND	PWR	common ground	Black
2	+3V3	PWR	3.3V Rail from autopilot	Red
3	SCL	I/O	SCL (3.3V level)	Green
4	SDA	I/O	SDA (3.3V level)	Blue

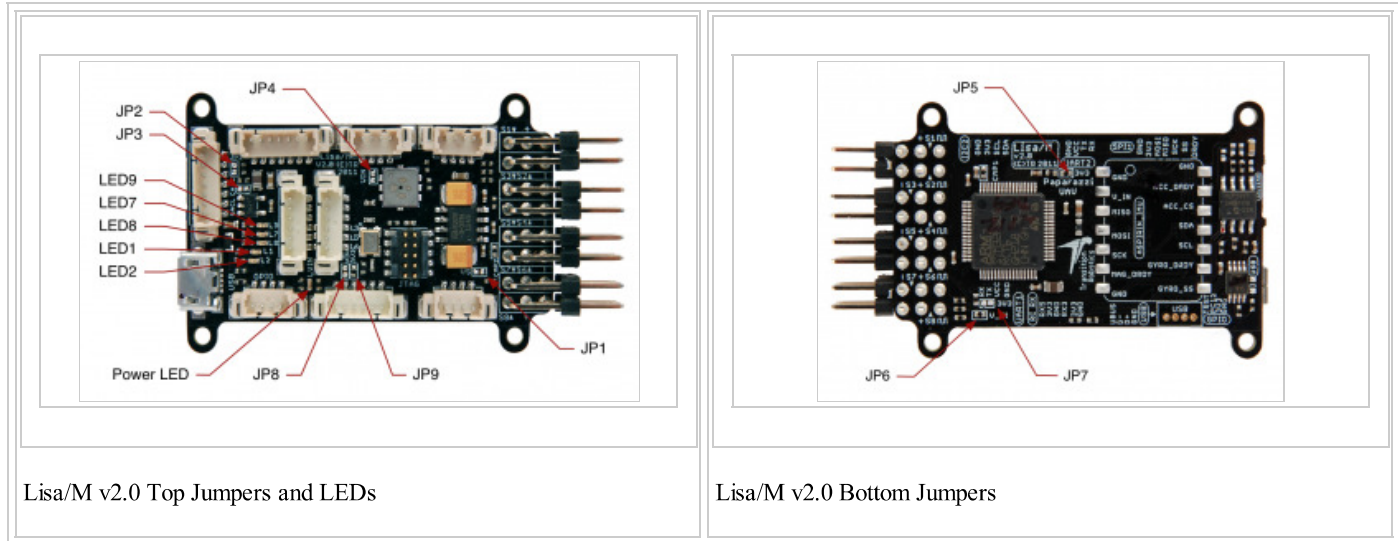
NOTE 1: SERVO7 and SERVO8 are directly connected to I2C1_SCL and I2C1_SDA. As such, SERVO7 and SERVO8 **CAN NOT** be used

while I2C1 is being used.

Jumper Configuration

There are a number of jumpers on Lisa/M used to configure voltage levels and power input.

The default configuration is UART3 VCC at V_IN, UART1/2/5 VCC at +3V3, with the V_SERVO servo voltage rail NOT connected to the autopilot V_IN rail, allowing one to power the autopilot and servos separately. The +5V regulator is NOT bypassed, allowing a regulated +5V on listed headers and for the CAN transceiver and I2C level shifter. The V_BAT connector is NOT connected to V_IN, so one can attach a battery voltage to the V_BAT pin to measure the battery voltage, if so desired.



Lisa/M v2.0 Top Jumpers and LEDs

Lisa/M v2.0 Bottom Jumpers

Power Jumper Configuration

<i>Jumper</i>	<i>Bus Connection</i>	<i>Default</i>	<i>Description</i>
JP1	SERVO_BUS to V_IN	OPEN	Connects servo header voltage rail SERVO_BUS to autopilot input voltage V_IN rail
JP2	V_BAT to V_IN	OPEN	Connects I2C1/CAN rail V_BAT to autopilot input voltage V_IN rail
JP3	V_IN to +5V	OPEN	Connects autopilot input voltage V_IN rail to autopilot +5V rail, bypassing onboard 5V supply

UART3 VCC Configuration

<i>Jumper</i>	<i>Bus Connection</i>	<i>Default</i>	<i>Description</i>
JP6	UART3_VCC to V_IN	CLOSED	Connects UART3 connector VCC to autopilot input voltage V_IN rail
JP7	UART3_VCC to +3V3	OPEN	Connects UART3 connector VCC to autopilot +3V3 rail

WARNING: UART3 GPS Connector is connected to V_IN, check your GPS input voltage before connecting!!!

WARNING: DO NOT CLOSE BOTH JP6 AND JP7 SIMULTANEOUSLY!!!

UART2 VCC Configuration

<i>Jumper</i>	<i>Bus Connection</i>	<i>Default</i>	<i>Description</i>
	UART2_VCC to		Connects UART2 connector VCC to autopilot input voltage

JP4	V_IN	OPEN	V_IN rail SEE WARNING BELOW
JP5	UART2_VCC to +3V3	CLOSED	Connects UART2 connector VCC to autopilot +3V3 rail

WARNING: UART2 RX is NOT 5V TOLERANT. Thus, while possible to connect UART2_VCC to V_IN, DO NOT ATTEMPT THIS. Only use JP5 (the default).

WARNING: DO NOT CLOSE BOTH JP4 AND JP5 SIMULTANEOUSLY!!!

UART1 and UART5 VCC Configuration

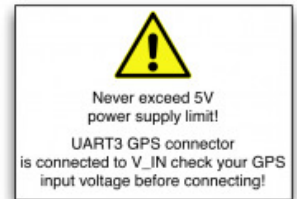
Jumper	Bus Connection	Default	Description
JP8	UART1&5_VCC to V_IN	OPEN	Connects UART1 and UART5 connector VCC to autopilot input voltage V_IN rail
JP9	UART1&5_VCC to +3V3	CLOSED	Connects UART1 and UART5 connector VCC to autopilot +3V3 rail

WARNING: DO NOT CLOSE BOTH JP8 AND JP9 SIMULTANEOUSLY!!!

There are additional jumpers on the board for expert or developer configurations, please see schematic and layout for more information.

Powering the Board

The 3.3V regulator on Lisa/M is a MIC5209-3.3YM (<http://www.micrel.com/page.do?page=/product-info/products/mic5209.shtml>) capable of delivering up to 500mA. While it is possible to power this regulator with up to 16V, **DO NOT** do this. By default, the UART3 RX pin is pulled up to the input voltage V_IN. For this reason, 5V is the maximum input voltage. Note that the UART3 GPS Connector is connected to V_IN, check your GPS input voltage before connecting. If one desires to have V_IN at a higher voltage, the jumpers should be adjusted accordingly. As noted, this regulator can handle up to 16V, though experience has shown that this regulator can become very hot in operation with high input voltages, resulting in potential thermal shutdown while in flight. Depending on the expected current draw, it is best to power this regulator with a lower voltage. 5V is the perfect choice.



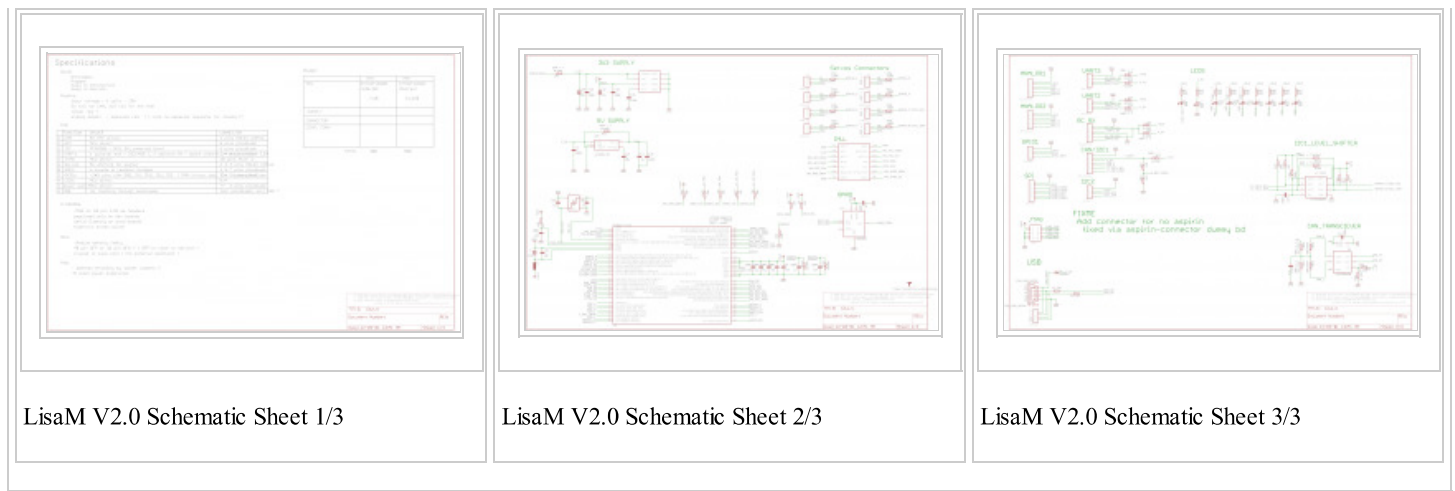
The onboard 5V regulator on Lisa/M is a LP2992 (<http://www.national.com/pf/LP/LP2992.html>), a low-noise, low-dropout linear regulator capable of delivering up to 250mA. This regulator can be bypassed with JP3, connecting the autopilot V_IN bus directly to the autopilot 5V bus if, for example, one is using an external 5V regulated supply, and a higher current is needed. Unless external use of 5V is required on the ANALOG1 and ANALOG2 headers, the only 5V usage onboard is for the CAN transceiver and the I2C1 level shifter.

When measuring the supply voltage of a battery with the V_BAT pin (could be connected to V_IN through JP2), it is important to note the maximum voltage limit. The voltage divider on the board for measuring with a 3.3V ADC is --V_BAT--//10k//--V_BAT_MEAS--//2k2//--GND--. This means that the maximum allowable voltage on V_BAT is

$$V_{BAT_{max}} = 3.3V * \frac{10k}{2.2k} = 15V$$

If a higher voltage measurement is desired, another voltage divider is required off-board. Alternatively, one could modify the existing voltage divider (e.g. change 10k resistor to 22k to get 33V maximum). When checking if voltage exceeds the maximum, make sure to consider maximum battery voltage, not nominal voltage (e.g. 4.22V or so for a single lithium cell, not 3.7V nominal, so the maximum number of cells in series is 3, like a 3S LiPo pack).

Schematic



Examples of Airborne Equipment Electrical Connections

Small Aircraft Connection Diagram

Need an Umarim_v1.0-style (here) small aircraft airborne equipment electrical connections here.

Large Aircraft Connection Diagram

Need an Umarim V1.0-style (here) large aircraft airborne equipment electrical connections diagram here.

R/C Receivers

There is Spektrum parser available already, enabling the direct use of 1 or 2 Spektrum satellite receivers.

Also UART pins can be used as general purpose I/O to be used for PPM input. Additionally PPM encoder can be used to avoid receiver hardware modification.

PCB

Gerber & Drill Files

Download Lisa/M v2.0 gerber & drill files (zip) NOT YET AVAILABLE BUT SEE Downloads Need some generated gerbers and drill files here.

Assembly

Components Layout

NOT YET AVAILABLE BUT SEE Downloads Need some top and bottom of board images and line drawings here.

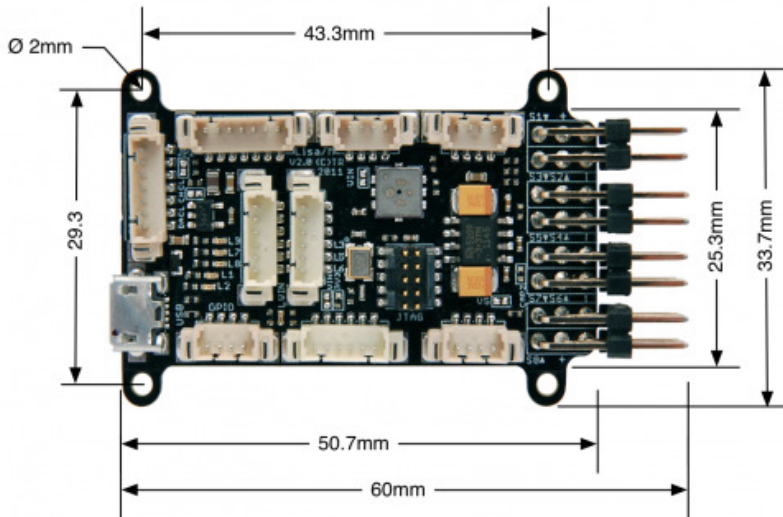
Bill Of Material

Download Lisa/M v2.0 Bill Of Material (zipped .xls file) NOT YET AVAILABLE BUT SEE Downloads

PCB and assembled boards suppliers

Available on Get Hardware page, hopefully :)

Mechanical Dimensions



The overall height of the board including the servo connectors is 10mm. Note that the overall length includes the USB connector. The mounting holes are nominal 2mm diameter (with a bit of clearance).

Downloads

Source files

- download available on GitHub: *Lisa/M v2.0 Cadsoft Eagle 6 Design* (https://github.com/paparazzi/paparazzi-hardware/tree/master/controller/lisa_m/v2.0)

Gerber & Drill files

- download *NOT YET AVAILABLE* Need generated gerbers and drill files

Assembly files

- download *NOT YET AVAILABLE* Need Lisa/M v2.0 Components layouts (pdf)
- download *NOT YET AVAILABLE* Need Lisa/M v2.0 Bill Of Material

Programming

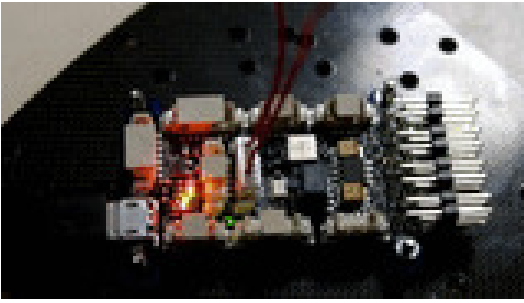
Lisa/M v2.0 can be programmed via its micro-USB port using the luftboot bootloader or with a JTAG via the 10-pin Samtec connector. All Lisa/M 2.0 from Transition Robotics Inc. come with luftboot preloaded.

Using luftboot

Paparazzi git dev branch already has support for the bootloader and will by default use the associated DFU loader to program the firmware. (Don't forget to check that your airframe file is set to use Lisa/M 2.0 as its target board)

Currently Paparazzi firmware does not contain the needed USB stack stub to switch into bootloader mode using software. To force the bootloader to run, bridge ADC2 and GND by making up a connector. This won't be necessary in the future once the USB stack stub is added to Paparazzi.

Once the bridge connector is attached, attach the Lisa/M to the PC via a micro-usb cable and it should start in bootloader mode, and the status LEDs will cycle up and down:



You can and should disconnect the bridge connector after enforcing the bootloader. Assuming your build target uses the Lisa/M board, pressing upload should now load code onto the board.

If you overwrite/remove luftboot you can recover it by following the instructions below:

Uploading the Paparazzi USB Bootloader

This section describes the process on how to upload/recover the luftboot bootloader on your Lisa/M 2.0. It is not needed if you got your Lisa/M 2.0 from Transition Robotics Inc. as the boards come with luftboot preloaded.

Luftboot (<https://github.com/paparazzi/luftboot>) is a Paparazzi-compatible bootloader for STM32-based autopilots. Depending on your vendor, your Lisa/M may already come with a bootloader, in which case you should skip to [Lisa/M#Programming](#)

Required components

- Floss-JTAG debugger
- Lisa/M
- PC

Procedure

1. Checkout Luftboot from Github (<https://github.com/paparazzi/luftboot>)

```
git clone https://github.com/paparazzi/luftboot.git
```
2. Change directory into the luftboot/src folder

```
cd ./luftboot/src
```
3. Build luftboot

```
make clean && make
```
4. Flash the Lisa/M Attach the floss-jtag unit to the PC and connect it to the Lisa/M via the black connector. Power the Lisa/M (easiest way is to connect to the PC via a micro-USB cable).

```
make flash DEV_SERIAL="LM2-ser"
```

where "ser" stands for the serial number of your Lisa/M. So for example if you have lisa/m with the serial number 020 this would be:

```
make clean && make flash DEV_SERIAL="LM2-020"
```

Connection Diagram

Boot Sequence

- Luftboot
 - Check if ADC2 is configured as output pull down indicating software bootloader request
 - **If ADC2 output pull down:** initialize usb and stay in bootloader mode
 - Setting the ADC2 pin to input pull up
 - Checking if the ADC2 pin is low
 - **If ADC2 low:** initialize USB and stay in bootloader mode
 - **If ADC2 high:** check if there is a payload at 0x8002000
 - **If payload detected:** set vector table pointer to be at 0x8002000 and jump to the reset handler of the payload
 - **If payload not detected:** initialize USB and stay in bootloader mode

JTAG

If you are using the Lisa/M 2.0 target board you still can use JTAG for programming and debugging your paparazzi firmware. Using JTAG will not

paparazzi.enac.fr/wiki/Lisa/M

overwrite the bootloader by default. You need to set NO_LUFTBOOT config in your airframe file to force overwrite.

- JTAG description;
- General debugging information;
- JTAG usage, includes Eclipse uplink tutorial.

Quick JTAG Upload Guide

1. Connect floss-jtag to Lisa via the cortex cable (little black socket)
2. Attach the UART port on the bottom of the floss-jtag to UART2 on the Lisa.
3. Plug in USB port of the floss jtag
4. Plug in USB port of the Lisa
5. Make sure your airframe uses the <target name="ap" board="lisa_m_2.0"> definition
6. Click Build, wait until complete, then click Upload. You should see the following towards the end:

```

{{{ ... Info : device id = 0x10016418 Info : flash size = 256kbytes stm32x mass erase complete Info : Padding image section 1 with 7972 bytes
wrote 152576 bytes from file src/paparazzi/var/Hexa_LisaL/ap/ap.elf in 7.498179s (19.871 KiB/s) Info : JTAG tap: stm32.cpu tap/device found:
0x3ba00477 (mfg: 0x23b, part: 0xba00, ver: 0x3) Info : JTAG tap: stm32.bs tap/device found: 0x06418041 (mfg: 0x020, part: 0x6418, ver: 0x0)
shutdown command invoked }}}

```

1. Run Flight USB-serial at the baud rate you need (default 57600 for rotorcraft)
2. You may need to change the device to /dev/ttyUSB1, and 'Redo' the Data Link

Serial Firmware Upload

Firmware upload using the factory integrated bootloader can be useful e.g. if you have overwritten Luftboot accidentally and don't have access to JTAG.

Due to hardware constraints, the board has to be modified to make use of the bootloader, which is only accessible on UART1:

1. Diode D3 has to be removed (the bigger black brick next to the USB connector). Attention, no more powering via USB after that.
2. BOOT1 has to be set to GND by connecting ACC_DRDY(used) to GND at the Aspirin pads

Now a boot sequence works as follows:

1. BOOT1 has to be set to 3.3V by use of a jumper cable
2. Connect a 3,3V serial cable (FTDI, MAX232...) to UART1, the TX pin is USB_VBUS
3. Power the board and activate the bootloader program

The according bootloader script can be found at : stm32loader from Github (<https://github.com/jsnyder/stm32loader>)

```
git clone https://github.com/jsnyder/stm32loader.git
```

To reload Luftboot, upload luftboot.bin

Serial upload can also be used directly from paparazzi Center by adapting the right path in Makefile.stm32 (<https://github.com/paparazzi/paparazzi/blob/dev/conf/Makefile.stm32>) for the LOADER argument and setting

```
<define name="FLASH_MODE" value="SERIAL"/>
```

in the target section of the airframe configuration.

Detailed Hardware Revision History

Changes Between v1.1 and v2.0

- Lots of silkscreen improvements
- Added attributes to all parts to make the usage of bom-ex ulp possible.
- Improved routing to allow teardropping
- Fixed stm32f1, f2 and f4 compatibility circuit. (has to jump to ground not to 3v3)
- Connected existing UART RX pullups to the respective connector power pins instead of 3v3. To prevent connecting 5V over IO pin to the 3v3 power rail.

- Added pullups on all UART RX lines to prevent undesired floatation.
- LED's are connected to 3v3 now. To make sure we don't have an issue with voltage tolerance on the gpio pins.
- ...

Changes Between v1.0 and v1.1

- Removed pull-ups on the USB gpio
- Removed pull-ups on the CAN gpio
- Connected usb_vbus to pa9 (needed by the USBotg)
- Removed USB pullup transistor as usbotg has a built in pullup
- Swapped UART1 with UART3 (uart1 was used for gps and pa9 was it's tx line, to be able to talk to the gps unit uart3 is a better choice, as uart1 only has an rx line now it is a better choice for spektrum RX modules)
- Removed USART3 TX gpio from the GPIO connector and moved to the GPS connector
- Added voltage selector jumpers to the RC RX connector; to enable powering of 3v3 or an 5v receivers
- Replaced vertical board solution with through hole servo pin headers (easier assembly)
- Servo connectors are in groups of two; for easier assembly
- Servo VBUS is connected together on all four layers; for lower resistance
- Moved LED's from under the analog2 connector; to be able to populate LED's and the connector
- Moved the RC RX connector a bit; to prevent crashing with the jtag plug
- Added one additional servo connector; now we have all 8 accessible through the standard servo connectors
- Fixed servo channel labeling to start at **S0** as it is the case on TWOOG and Tiny autopilot boards
- Added secondary through hole picoblade USB connector for easier routing of USB inside an airframe
- ...

Changes Between v0.1 and v1.0

- Switched to stm32f105 to be able to use usb and can at the same time
- Added alternative use of the adc lines as led output
- ...

Hardware Change Requests

- REQ: Replace BMP085 with MS5611 (the MS5611 seems to be better in performance then the BMP but it is more expensive and seems to be more difficult to obtain.
 - A: This upgrade will be available through Aspirin v2.0 --Esden 22:54, 5 January 2012 (CET)
- REQ: Replace 7 Pin CAN with molex with something less risky to be inserted in 7 Pin SPI in relation to powering the board via CAN molex.
- REQ: Separate spot for external power if powered via separate battery. Realizing we can via Servo ports by Bridge J1 but still like to measure board voltage then and have a way to add power without mistakenly inject CAN Molex into SPI.

Retrieved from "http://paparazzi.enac.fr/wiki/Lisa/M_v20"

Categories: Lisa | User Documentation

- This page was last modified on 21 April 2012, at 18:32.
- Content is available under GNU Free Documentation License 1.2.