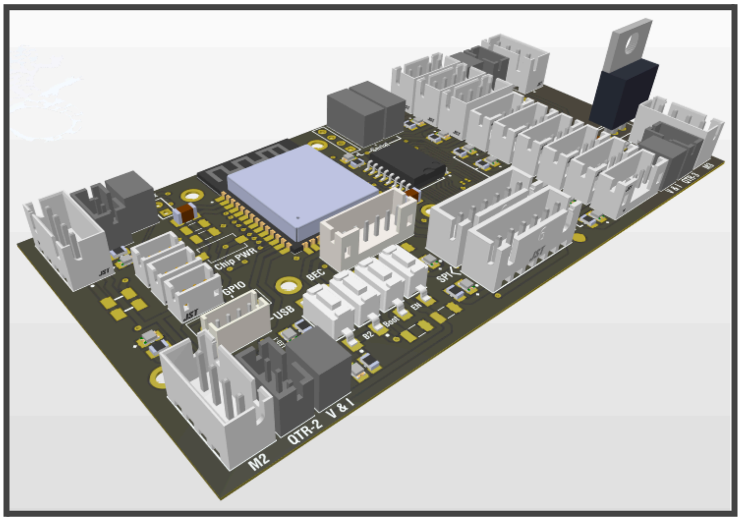
Leguizard PCB
Controlador Motores Brushless
y sensores

Features:

- ESP32 Wroom 1 N8R8 CPU
- 4 ESC Outputs
- 4 Line detecting sensor Inputs
- 9 I²C connectors
- 6 I²C channels design for LASER module
- 2 SPI outputs
- SD card reader
- 4 GPIO connectors
- 2 UART connectors
- USB programable



Description:

This board is designed to drive a fully autonomous brushless powered robot. The board was designed to fulfill most needs that any autonomous robot might need. It has the usual ports that any robot might need, I²C, SPI, UART and GPIO. The board can be powered by two means, by a dedicated BEC or by the integrated regulator that the ESC has. To regulate the power the board counts with two options, a NCP1117DT33T5G or a LD1117, both can be disabled with a soldered jumper. Uploading code to the ESP32 can be done by three different ways,OTA, USB and UART.

Pin description:

The chart indicates every pin function and, in the most right column, the use we assign them.

Pin N	Name	Type	Function	Use
1	GND	P	GND	GND
2	3V3	P	Power supply	3.3v
3	EN	I	High: on, enables the chip. Low: off, the chip powers off. Note: Do not leave the EN pin floating.	EN
4	IO4	I/O/T	RTC_GPIO4, GPIO4, TOUCH4, ADC1_CH3	Q1
5	IO5	I/O/T	RTC_GPIO5, GPIO5, TOUCH5, ADC1_CH4	Q3
6	IO6	I/O/T	RTC_GPIO6, GPIO6, TOUCH6, ADC1_CH5	GPIOP1
7	IO7	I/O/T	RTC_GPIO7, GPIO7, TOUCH7, ADC1_CH6	GPIOP2
8	IO15	I/O/T	RTC_GPIO15, GPIO15, U0RTS, ADC2_CH4, XTAL_32K_P	GPIOP3
9	IO16	I/O/T	RTC_GPIO16, GPIO16, U0CTS, ADC2_CH5, XTAL_32K_N	GPIOP4
10	IO17	I/O/T	RTC_GPIO17, GPIO17, U1TXD, ADC2_CH6	GPIOP5
11	IO18	I/O/T	RTC_GPIO18, GPIO18, U1RXD, ADC2_CH7, CLK_OUT3	GPIOP6
12	IO8	I/O/T	RTC_GPIO8, GPIO8, TOUCH8, ADC1_CH7, SUBSPICS1	GPIOP7
13	IO19	I/O/T	RTC_GPIO19, GPIO19, U1RTS, ADC2_CH8, CLK_OUT2, USB_D-	USB -
14	IO20	I/O/T	RTC_GPIO20, GPIO20, U1CTS, ADC2_CH9, CLK_OUT1, USB_D+	USB +
15	IO3	I/O/T	RTC_GPIO3, GPIO3, TOUCH3, ADC1_CH2	
16	IO46	I/O/T	GPIO46	GND
17	IO9	I/O/T	RTC_GPIO9, GPIO9, TOUCH9, ADC1_CH8, FSPIHD, SUBSPIHD	M1-PWM
18	IO10	I/O/T	RTC_GPIO10, GPIO10, TOUCH10, ADC1_CH9, FSPICS0, FSPIIO4, SUBSPICS0	M1-C
19	IO11	I/O/T	RTC_GPIO11, GPIO11, TOUCH11, ADC2_CH0, FSPID, FSPIIO5, SUBSPID	B1
20	IO12	I/O/T	RTC_GPIO12, GPIO12, TOUCH12, ADC2_CH1, FSPICLK, FSPIIO6, SUBSPICLK	B2
21	IO13	I/O/T	RTC_GPIO13, GPIO13, TOUCH13, ADC2_CH2, FSPIQ, FSPIIO7, SUBSPIQ	POWER-I

22	IO14	I/O/T	RTC_GPIO14, GPIO14, TOUCH14, ADC2_CH3, FSPIWP, FSPIDQS, SUBSPIWP	POWER-V
23	IO21	I/O/T	RTC_GPIO21, GPIO21	M2-C
24	IO47	I/O/T	SPICLK_P_DIFF,GPIO47, SUBSPICLK_P_DIFF	M2-PWM
25	IO48	I/O/T	SPICLK_N_DIFF,GPIO48, SUBSPICLK_N_DIFF	M3-C
26	IO45	I/O/T	GPIO45	M3-PWM
27	IO0	I/O/T	RTC_GPIO0, GPIO0	
28	IO35	I/O/T	SPIIO6, GPIO35, FSPID, SUBSPID	SCL [I2C]
29	IO36	I/O/T	SPIIO7, GPIO36, FSPICLK, SUBSPICLK	SDA [I2C]
30	IO37	I/O/T	SPIDQS, GPIO37, FSPIQ, SUBSPIQ	SS1
31	IO38	I/O/T	GPIO38, FSPIWP, SUBSPIWP	M4-C
32	IO39	I/O/T	MTCK, GPIO39, CLK_OUT3, SUBSPICS1	M4-PWM
33	IO40	I/O/T	MTDO, GPIO40, CLK_OUT2	SCK
34	IO41	I/O/T	MTDI, GPIO41, CLK_OUT1	MOSI
35	IO42	I/O/T	MTMS, GPIO42	MISO
36	RXD0	I/O/T	U0RXD, GPIO44, CLK_OUT2	RDX
37	TXD0	I/O/T	U0TXD, GPIO43, CLK_OUT1	TDX
38	IO2	I/O/T	RTC_GPIO2, GPIO2, TOUCH2, ADC1_CH1	Q2
39	IO1	I/O/T	RTC_GPIO1, GPIO1, TOUCH1, ADC1_CH0	Q4
40	GND	P	GND	GND
41	EPAD	P	GND	GND

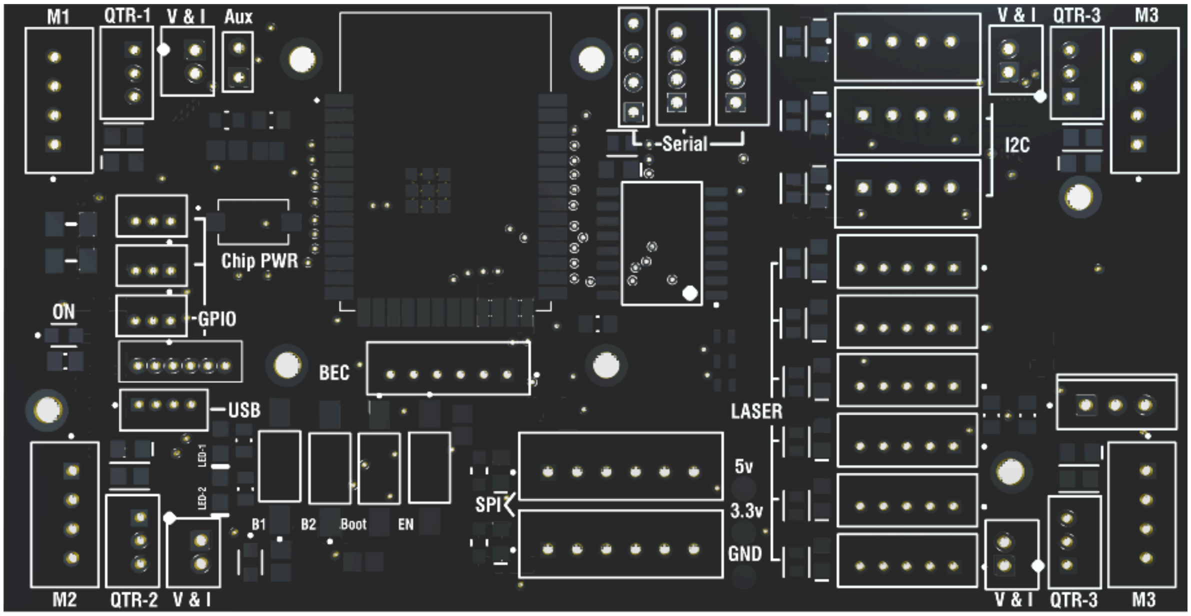
In yellow are pins that have specific functions during the boot and reboot process. Pin 3, Enable, Reboots the ESP, while the pin 27, IO0, Is used for booting a new program into the chip.

In red are pins that configure the Chip Boot Mode Control.[1]

[1] To know more about these pins read section 3.3.1 of the ESP32 datasheet.

Connectors diagram:

The board counts with 33 connectors, each designed with a purpose in mind. The terminals can be subdivided in 8 big groups, between those groups the connector pitch changes making it impossible to mix up the wiring. Anding to that, each connector has a clear silkscreen name indicating the function that it complies with.



Motors:

This group is formed by four JST XH B4B connectors in each corner of the board. The identifiers are M1, M2, M3 and M4. Every connector has in common three signals, GND, Motor PWM and Motor Direction. Additionally each terminal has one more pin, the pin N1, that is of no use in three out of four connectors, only in M1 it serves the function of Auxiliary BEC. This is because most Electronic Speed Controllers(ESCs) have one in them supplying 5v. These functions can be toggled on and off[2].

Line detecting sensors:

This group is formed by four JST XH B3B connectors in each corner of the board. The identifiers are QTR-1, QTR-2, QTR-3 and QTR-4. Every connector has two signals in common, Vcc and GND. The third pin corresponds to an analog input.

[2] see Jumpers section.

Current and Voltage measurements:

This group is formed by four JST XH B2B connectors in each corner of the board. The identifiers are V & I. Every connector has two analog inputs that measure the voltage and current that flows to each ESC.

Inter-Integrated Circuit(I^{2}C):

This group is formed by three JST XH B4B and six JST PH B5B connectors placed in a row. The identifiers are LASER and I2C. Every terminal shares three signals, GND, SDA and SCL. In the one hand the JST XH B4B (I2C), have in the fourth pin a Vcc connection, which can be 3.3v and 5v according to the jumpers configuration[3]. On the other hand the JST PH B5B has a 5v non configurable source and a fifth pin which goes to a digital IO pin. This pin is for the XShut pin on most LASER modules.

Serial Peripheral Interface(SPI):

This group is formed by two JST XH B6B placed side by side. The identifier is SPI. Each connector has five pins in common, GND, configurable Vcc[4], MOSI, MISO and SCK. The sixth pin corresponds to the Slave Select(SS) that goes to a digital IO pin.

Universal Asynchronous Receiver Transmitter(UART):

This group is formed by two JST PH B4B and one four pin Male Header placed in a row. The identifier for the group is Serial. Every terminal has the same connection, GND, 5v, Rx and Tx. The header is meant only for special cases where OTA and USB methodes for programming the chip didn’t work. In those cases there is the option to solder the header and upload code via an ST-Link stick.

General Inputs & Outputs(GPIO):

This group is formed by three JST ZR B3B and one JST ZR B6B In a row. The group identifier is GPIO. In the one hand the JST ZR B3B connectors have two pins in common, GND and a configurable Vcc[5], the third pin is connected to an analog Input. On the other hand the JST ZR B6B connector has GND, a configurable Vcc and four analog inputs.

[3] See Jumper section for more info.

[4] See Jumper section for more info.

[5] See jumper section.

Miscellaneous connectors:

This group is formed by one JST ZR B4B and one two pin male header. The JST ZR B4B connector is identified as USB and the male header is identified as Aux. The JST ZR B4B is connected to the D+ and D- of the Esp32 while the Aux header is connected to the 3.3v net via an NC switch. This allows you to toggle on only the esp32 without turning on the other circuitry.

Power:

This group is formed by one JST XH B6B. The JST XH B6B connector is identified as BEC. It has two GND pins, two togable Vcc pins, a current and a voltage sensing pin. This BEC is intended to power the board and measure the battery voltage and current.

Jumpers:

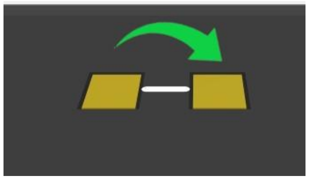
The board counts with a jumper system to give more freedom while choosing sensor options. This consists of three main systems, the address system, the voltage system and the enable system. The address system allows you to change the address of the IO expander, the voltage system allows you to connect 5v sensors or 3.3v sensors, offering a wider range of options, and the enable system allows you to choose the power source, BEC or ESC BEC. To set either option you have to solder a jumper in the designated area shown in the table below.

Connector	3.3v Jumper designator	5v Jumper designator
J22	W27	W24
J23	W28	W25
J24	W29	W26
CON1	W23	W14
J20	W11	W9
J19	W10	W2
J6	W7	W8
J4	W5	W6
J1	W3	W4

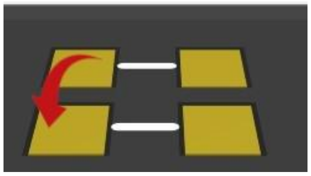
Designator	Enable pins
J11	W30 & W31
J10	W22 & W42
U10	W13

MPC23017		
Pin Number	GND	5v
1	W20	W17
2	W19	W16
3	W18	W15

The solder jumper must be done over the line that joins the footprint, as shown:



Correct way



Incorrect way

IT IS IMPORTANT NOT TO SOLDER MORE THAN ONE JUMPER PER CONNECTION, that will cause a short circuit. It is also mandatory to solder all jumpers on the address system, if not it could cause problems while assigning devices.

SD slot:

The board counts with an SD slot for data logging. To write to the Sd card the SPI protocol is used, the SS(Slave Select) pin is located on pin 30 of the ESP32.

Bill Of Materials:

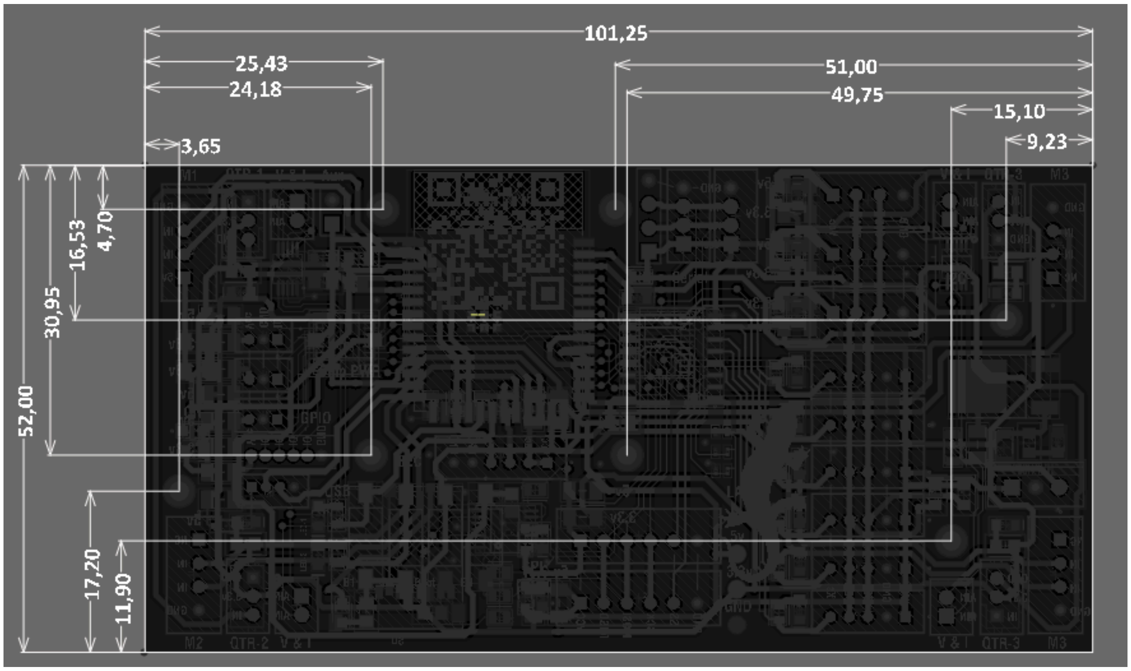
Name	Designator	Quantity	Manufacturer
APHCM2012SYCK-F01	D1	1	Kingbright
B3B-ZR(LF)(SN)	J7, J8, J9	3	JST
B4B-XH-A(LF)(SN)	J10 → J12, J21 → J24	7	JST
B4B-ZR(LF)(SN)	J6	1	JST
B6B-PH-K-K(LF)(SN)	J1	1	JST
B6B-XH-A(LF)(SN)	J19, J20	2	JST
CRCW080510K0FKEA	R1, R6, R8, R9, R12	5	Vishay
CRCW080547K0FKEA	R3, R5	2	Vishay
CRCW0805390RFKEA	R10, R11	2	Vishay
ERJ-P06F1001V	R13 → R21, R27, R28, R33	12	Panasonic
GRM31CR71A226KE15L	C6, C7	2	Murata
LTST-C170KFKT	D13	1	Vishay Lite-On
NCP1117DT33T5G	U7	1	ON Semiconductor
RG2012N-332-C-T5	R22, R23, R24, R25	4	Susumu
SN74LVC125APWR	U9	1	Texas Instruments
TMK212BBJ106KG-T	C1	1	Taiyo Yuden
TXS0108EQPWRQ1	U5, U6	2	Texas Instruments
B6B-ZR	CON1	1	JST
ESP32-S3-WROOM-1 N8R8	U1	1	Expressif systems
Header 2	P1	1	-

Header 4	P6	1	-
Jumper	W1 → W29	29	-
Test point	T1, T2, T3	3	-
TS10-63-26-BE-250-SMT-TR	S1	1	CUI Devices
150080BS75000	D4 → D20	16	Wurth Electronics
150080SS75000	D3	1	Wurth Electronics
473521001	SD1	1	Molex
ADS1115IDGST	U3, U4	2	Texas Instruments
B2B-PH-K-S(LF)(SN)	J2, J3, J4, J5	4	JST
B3B-PH-K-S(LF)(SN)	P2, P3, P4, P5	4	JST
B4B-PH-K-S(LF)(SN)	P7, P8	2	JST
B5B-PH-K-S(LF)(SN)	J13, J14, J15, J16, J17, J18	6	JST
BAS3010A03WE6327HTSA1	D2	1	Infineon
C0805C105K4PACTU	C3	1	KEMET
CRCW0805680RFKEA	R7	1	Vishay
ERJ6ENF6800V	R26	1	Panasonic
ERJ-6ENF1500V	R29, R30, R31, R32	4	Panasonic
FSMSMTR	SW1, SW2, SW3, SW4	4	TE Connectivity
GRM21BR71H104KA01L	C2, C4, C5	3	Murata
HSMG-C170	DS1, DS2	2	Broadcom Avago
LM7805CT	U8	1	ON Semiconductor / Fairchild
PCF8574DWR	U2	1	Texas Instruments
RC0805FR-07100KL	R2, R4	2	Yageo

Any → indicates that the sequence goes from the first designator adding one to the last designator.
Eg: R1 → R4 = R1,R2,R3,R4

The complete BOM is attached to the project.

Physical dimensions:



The Mounting holes present are M2 HEX.

Programing the ESP32:

Programing the ESP32

1. Install Arduino IDE[6]
2. Download & install the ESP family boards[7]
3. Download & install the libraries
4. Upload the code to the board

There are several ways you can upload the code to the ESP32.

- Via OTA programing[8]
- Via USB port
- Via TTL protocol (use only when the other two methods didn't work)

[6] See resources section for more info about Arduino
[7] See resources section for more info about the ESP family
[8] See resources section for more info about OTA programing

To have access to Tx & Rx pins, for the TTL programing, there is an unsoldered header in the Serial section of the board. To Upload the code you must have an ST-LINK.

BE AWARE THAT THE CODE WON’T BE UPLOADED IF YOU DON'T PRESS THE BOOT BUTTON. If the code still doesn't uploads correctly you can try to supply the board with 3.3v ONLY externally and disconnect the rest of the circuit via the Chip PWR button. The connector to supply the board is identified as Aux.

Considerations:

Knowing the environment that the board will probably be used, here we list a few recommendations:

- Installing the board near brushless motors will inevitably cause electrical noise. We recommend shielding as best as possible the motors, the wires, and if possible the board.
- Installing the board in a mechanical stressful place will cause internal stresses in the board, in the solder joints and in the internal connections of the components. We recommend that if there are any problems with stability you change the board or at least reflow the PCB.
- Installing the board in an enclosed area can cause problems with the OTA, WiFi and BlueTooth wireless communications. This holds especially true if the casing is conductive.
- The GPIO pins, available on the board, can be used to source up to 28mA MAX without irreversibly burning the ESP32.
- We do not recommend installing the board in a environment that exceeds the 50◦ C.

Resources:

- ESP32 wroom 1U N8R8 datasheet
- ESP32 family
- Arduino
- Over The Air programing
- GitHub Repository