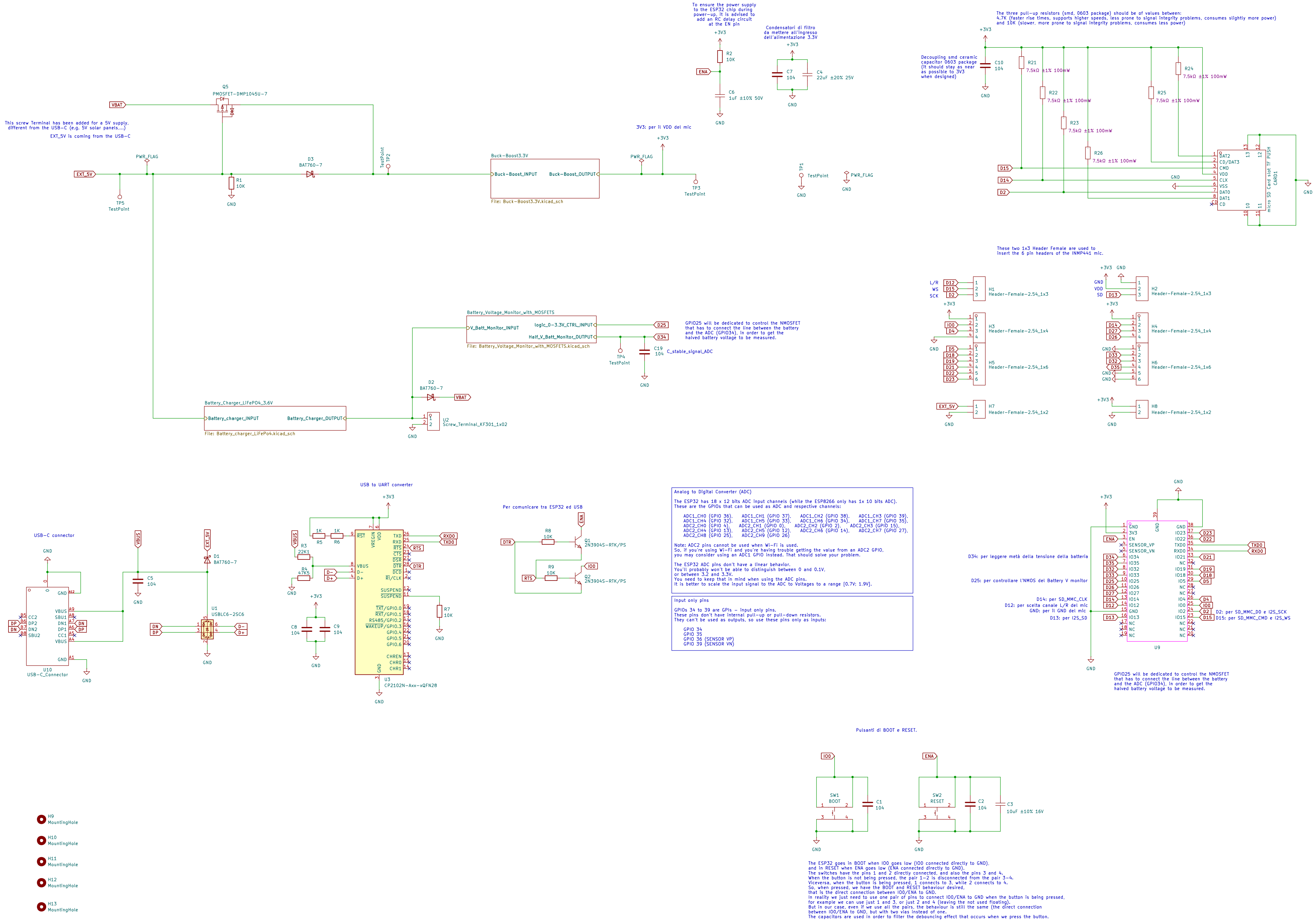


This screw Terminal has been added for a 5V supply,
different from the USB-C (e.g. 5V solar panels,...)

EXT_5V is coming from the USB-C

To ensure the power supply
to the ESP32 chip during
power-up, it is advised to
add an RC delay circuit
at the EN pin

Condensatori di filtro
da mettere all'ingresso
dell'alimentazione 3.3V

The three pull-up resistors (smd, 0603 package) should be of values between:
4.7K (faster rise times, supports higher speeds, less prone to signal integrity problems, consumes slightly more power)
and 10K (slower, more prone to signal integrity problems, consumes less power)

Decoupling smd ceramic
capacitor 0603 package
(it should stay as near
as possible to 3V3
when designed)

These two 1x3 Header Female are used to
insert the 6 pin headers of the INMP441 mic.

GPIO25 will be dedicated to control the NMOSFET
that has to connect the line between the battery
and the ADC (GPIO34), in order to get the
halved battery voltage to be measured.

Analog to Digital Converter (ADC)

The ESP32 has 18 x 12 bits ADC input channels (while the ESP8266 only has 1x 10 bits ADC).
These are the GPIOs that can be used as ADC and respective channels:

```
    ADC1_CH0 (GPIO 36),   ADC1_CH1 (GPIO 37),   ADC1_CH2 (GPIO 38),   ADC1_CH3 (GPIO 39),
    ADC1_CH4 (GPIO 32),   ADC1_CH5 (GPIO 33),   ADC1_CH6 (GPIO 34),   ADC1_CH7 (GPIO 35),
    ADC2_CH0 (GPIO 4),   ADC2_CH1 (GPIO 0),   ADC2_CH2 (GPIO 2),   ADC2_CH3 (GPIO 15),
    ADC2_CH4 (GPIO 13),   ADC2_CH5 (GPIO 12),   ADC2_CH6 (GPIO 14),   ADC2_CH7 (GPIO 27),
    ADC2_CH8 (GPIO 25),   ADC2_CH9 (GPIO 26)
```

Note: ADC2 pins cannot be used when Wi-Fi is used.
So, if you're using Wi-Fi and you're having trouble getting the value from an ADC2 GPIO,
you may consider using an ADC1 GPIO instead. That should solve your problem.

The ESP32 ADC pins don't have a linear behavior.
You'll probably won't be able to distinguish between 0 and 0.1V,
or between 3.2 and 3.3V.
You need to keep that in mind when using the ADC pins.
It is better to scale the input signal to the ADC to Voltages to a range [0.7V; 1.9V].

Input only pins

GPIOs 34 to 39 are GPIs – input only pins.
These pins don't have internal pull-up or pull-down resistors.
They can't be used as outputs, so use these pins only as inputs:

- GPIO 34
- GPIO 35
- GPIO 36 (SENSOR VP)
- GPIO 39 (SENSOR VN)

D34: per leggere metà della tensione della batteria

D25: per controllare l'NMOS del Battery V monitor

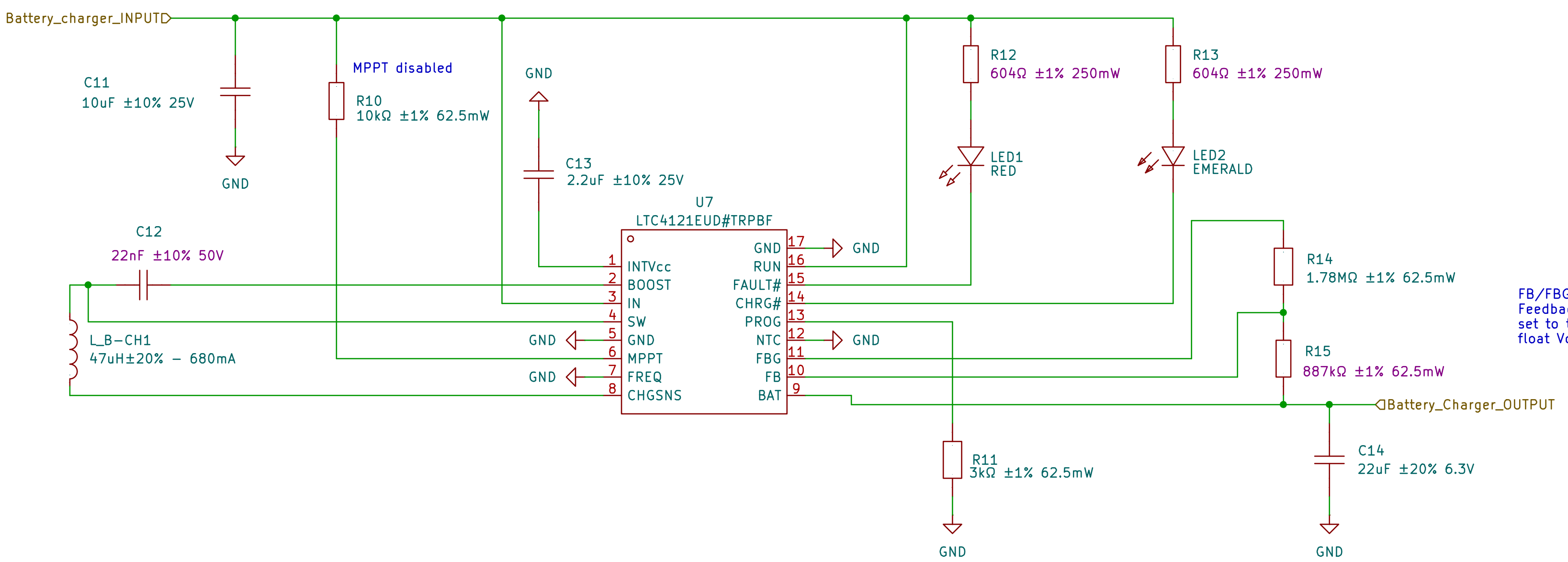
D14: per SD_MMC_CLK
D12: per scelta canale L/R del mic
GND: per il GND del mic
D13: per I2S_SD

D2: per SD_MMC_D0 e I2S_SCK
D15: per SD_MMC_CMD e I2S_WS

GPIO25 will be dedicated to control the NMOSFET
that has to connect the line between the battery
and the ADC (GPIO34), in order to get the
halved battery voltage to be measured.

The ESP32 goes in BOOT when IO0 goes low (IO0 connected directly to GND),
and in RESET when ENA goes low (ENA connected directly to GND).
The switches have the pins 1 and 2 directly connected, and also the pins 3 and 4.
When the button is not being pressed, the pair 1-2 is disconnected from the pair 3-4.
Viceversa, when the button is being pressed, 1 connects to 3, while 2 connects to 4.
So, when pressed, we have the BOOT and RESET behaviour desired,
that is the direct connection between IO0/ENA to GND.
In reality we just need to use one pair of pins to connect IO0/ENA to GND when the button is being pressed,
for example we can use just 1 and 3, or just 2 and 4 (leaving the not used floating).
But in our case, even if we use all the pairs, the behaviour is still the same (the direct connection
between IO0/ENA to GND, but with two vias instead of one.
The capacitors are used in order to filter the debouncing effect that occurs when we press the button.

For this schematic, see LTC4121EUD datasheet,
at pag.26, Figure 10. Design Example 3, SLA Charging with LTC4121

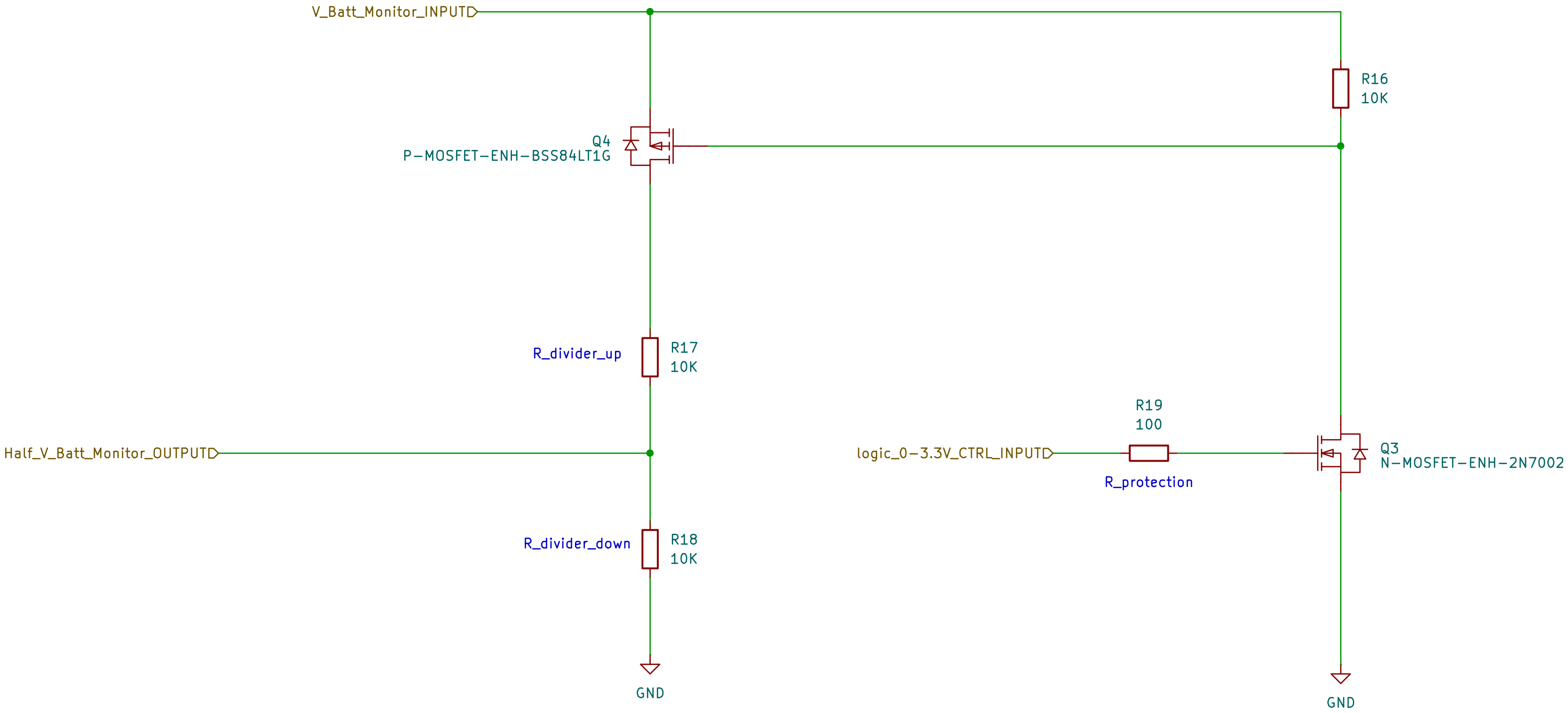


5mA led current

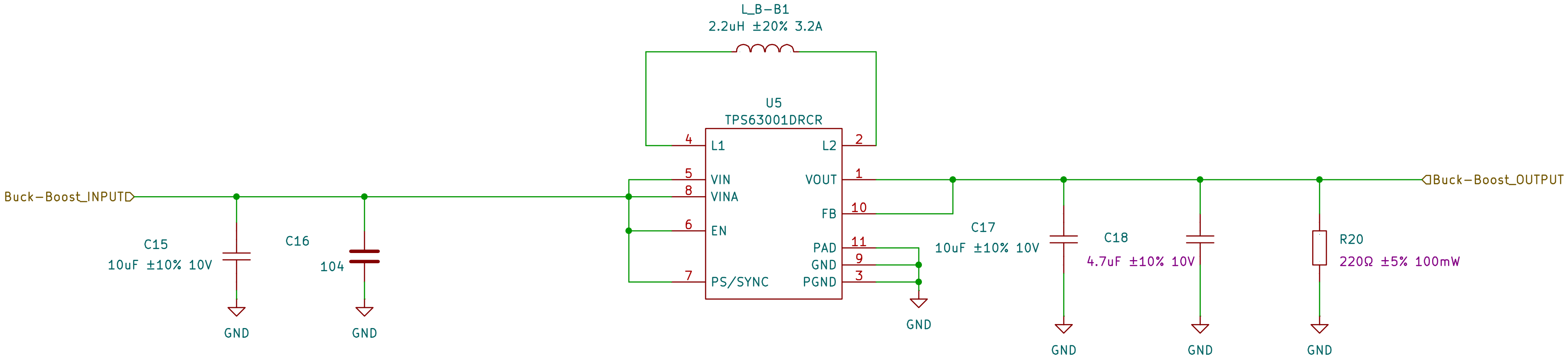
MPPT disabled

FB/FBG:
Feedback resistors
set to target 3.6V
float Voltage

PROG: 400mA charge current.
NTC: safety disabled.
FREQ: 750KHz switching freq.


V_Batt_Monitor_INPUT
R16
10K
Q4
P–MOSFET–ENH–BSS84LT1G
R_divider_up
R17
10K
R19
100
Half_V_Batt_Monitor_OUTPUT
logic_0–3.3V_CTRL_INPUT
Q3
N–MOSFET–ENH–2N7002
R_protection
R_divider_down
R18
10K
GND
GND

This Buck–Boost (TPS63001DRCR) is an efficient converter (up to 96%),
and can convert either the 5V coming from the USB,
or eventually the 3–3.6V coming from the batteries,
to fixed 3.3V necessary for the esp32.


L_B–B1
2.2uH ±20% 3.2A
U5
TPS63001DRCR
4
L1
L2
2
5
VIN
8
VINA
VOUT
1
FB
10
6
EN
PAD
11
GND
9
7
PS/SYNC
PGND
3
Buck–Boost_INPUT
Buck–Boost_OUTPUT
C15
10uF ±10% 10V
C16
104
C17
10uF ±10% 10V
C18
4.7uF ±10% 10V
R20
220Ω ±5% 100mW
GND
GND
GND
GND
GND
GND