
LoRa Beacon

Manuale di Installazione

Vr. 1.6

1	Introduzione	4
2	Note di montaggio HW	7
2.1	Versione LoRa_Beacon_2020_vr4_1 (alias mini-Tracker)	7
2.2	Versione LoRa_Beacon_2020_vr3_pcs4 (alias iGate)	10
2.3	Carrier per moduli radio LoRa (LoRa carrier)	15
2.4	Note di montaggio relative ad entrambe le versioni di PCB carrier	20
3	Installazione SW Iniziale	23
3.1	Setup ambiente di caricamento SW e caricamento Immagine SW iniziale	24
3.2	Setup iniziale del dispositivo LoRa_Beacon (qualsiasi versione)	29
3.3	Setup sottosistema LoRa (qualsiasi versione)	33
3.4	Setup sottosistema APRS (qualsiasi versione)	34
3.4.1	Setup sottosistema APRS per modalità iGate	34
3.4.2	Setup sottosistema APRS per modalità Tracker	35
3.4.3	Setup sottosistema APRS per connessione ad un server di servizio	35
3.5	Setup Tipo di incapsulamento (qualsiasi versione)	36
3.6	HW Setup Configuration (qualsiasi versione)	36
3.7	Setup sottosistema IoT (qualsiasi versione)	37
4	Interfaccia di Debug Remoto	38
4.1	Accesso alla IF di Remote Debug tramite Telnet Client	38
4.1.1	Comando “gps_status”	39
4.1.2	Comando “temperature”	40
4.1.3	Comando “wifi_scan”	41
4.1.4	Comando “display_config”	41
4.1.5	Comando “show_stats”	41
4.1.6	Comando “show_events”	42
4.1.7	Comando “log_display”	42
4.2	Accesso alla IF di Remote Debug tramite Web App	44
5	Porting del SW LoRa Beacon su altre piattaforme HW	47
5.1	Installazione SW su dispositivo TTGO T-beam-V1-2019	47
5.2	Installazione SW su dispositivo Heltec_wifi_lora32	50
6	Installazione SW via OTA (Over-The-Air)	52
7	Salvataggio e Caricamento della configurazione via OTA	55
8	Esempi di dove acquistare la componentistica	60

Pagina lasciata intenzionalmente vuota

1 Introduzione

Il presente documento descrive le modalità di montaggio HW e di installazione del SW dei dispositivi basati sul progetto LoRa_Beacon.

L'obiettivo di questo progetto è quello di generare una piattaforma HW/SW utilizzabile per attività di **sperimentazione relativamente alla tecnologia LoRa applicata ad attività di tipo radioamatoriali** e quindi non necessariamente utilizzando le linee guida e le implementazioni inizialmente pensate per questa tecnologia in ambito IoT (Internet of Things).

Va la pena di ricordare che la tecnologia LoRa è nata principalmente per consentire la realizzazione di dispositivi di tipo “sensore” ovvero per delle applicazioni di tipo M2M (Machine-To-Machine communication) caratterizzate da una limitata esigenza di comunicazione in termini di quantità di dati trasmessi, ma con l'obiettivo di realizzare la massima indipendenza da fonti di alimentazione esterne, cercando di utilizzare delle batterie per l'alimentazione dei dispositivi con un obiettivo di autonomia dell'ordine degli anni.

Come risultato di questi requisiti funzionali il protocollo radio LoRa ha mirato a ridurre la potenza di trasmissione dei dispositivi a parità di **“link budget radio”** disponibile ovvero aumentando tale valore a parità di potenza trasmessa; la chiave per ottenere tale obiettivo è l'utilizzo di una metodica di trasmissione a spettro disperso che consentisse di ottenere un significativo “guadagno di processo” .

Quindi nell'ambito dell'IoT un tema che assume enorme importanza è proprio la riduzione del consumo energetico dei dispositivi, in previsione del loro utilizzo come sensori remoti non alimentati da fonti di energia esterne.

Nell'uso radioamatoriale ovviamente questo aspetto passa in secondo piano mentre assumono rilevanza soprattutto gli aspetti di massimizzazione delle prestazioni in termini di “link budget radio disponibile”.

E' stato quindi deciso nella nostra sperimentazione di trascurare gli aspetti di minimizzazione dei consumi energetici, inessenziale ai fini dei nostri impieghi.

Il target di utilizzatori è stato quindi pensato per essere il radioamatore medio desideroso di sperimentare con poca spesa le nuove tecnologie HW/SW collegate in particolare a questa particolare tipologia di trasmissione radio.

La soluzione scelta si basa sul riutilizzo di “blocchi funzionali” HW e SW già disponibili allo scopo di semplificare e rendere economica la creazione di nuove configurazioni da sperimentare, con un minimo di sviluppi HW e SW.

La piattaforma HW è basata sul concetto di “carrier board”, ovvero di un circuito stampato destinato ad assemblare una serie di altri blocchetti funzionali a loro volta realizzati tramite piccoli circuiti stampati acquistabili sulle classiche piattaforme di e-commerce a prezzi molto convenienti. Per agevolare l'intercambiabilità di alcuni moduli la soluzione è caratterizzata dalla possibilità di ospitare moduli HW anche fisicamente differenti a parità di funzione svolta: in questo caso la carrier board viene dotata di diverse “impronte” atte a collegare diverse varianti di un certo blocco in maniera opzionale (ad es. montare diversi moduli GPS disponibili sul mercato).

Dal punto di vista della tipologia di montaggio si è cercato di utilizzare la modalità “Pin in Hole” limitando l’uso di componentistica SMD solo a casi eccezionali. Anche questa scelta è stata ovviamente motivata dall’obiettivo di rendere il più agevole possibile il montaggio dei circuiti evitando di richiedere conoscenze specifiche o particolari abilità per il montaggio.

Dal punto di vista SW si è cercato di utilizzare una metodologia simile cercando di usare anche a questo livello “**blocchi funzionali SW**” già disponibili nell’ambito dell’open source, limitando i nuovi sviluppi allo stretto indispensabile.

L’aspetto SW è ovviamente strettamente legato alla scelta HW di una specifica tipologia di processore da utilizzare: la scelta è caduta sull’uso del **processore ESP32** che nell’ambito dei microcontrollori rappresenta una soluzione ottimale in quanto fornisce ad un costo estremamente contenuto una **piattaforma di processo molto potente** in grado di operare in modalità “**multiprocessing**” con caratteristiche di tipo “**real time**”, supportato da un **sistema operativo molto leggero ed efficiente (FreeRTOS)** e in grado di essere supportato da un **ambiente di sviluppo basato sulla piattaforma SW Arduino**.

Grazie a queste scelte la soluzione si presenta molto “user friendly” e consente di sviluppare agevolmente del nuovo SW senza richiedere una curva di apprendimento eccessivamente lunga a chi si voglia cimentare in tale direzione.

Un aspetto molto curato è stato quello della **gestione dei dispositivi**; si è cercato anche qui di evitare ai potenziali utilizzatori di dover necessariamente passare tramite l’installazione completa dell’ambiente di sviluppo SW (strada tipicamente richiesta dai progetti basati su microcontrollori) fornendo **una semplice interfaccia grafica**, agevolmente customizzabile, tramite la quale impostare molte delle funzionalità presenti, consentendo quindi di poter realizzare della sperimentazione, a livello delle funzionalità radio, senza dover necessariamente passare per la fase di sviluppo SW.

Nel seguito vengono presentate innanzitutto alcune note relative al montaggio di un prototipo delle due versioni HW attualmente disponibili, per poi passare alla descrizione del caricamento di una immagine SW sui prototipi stessi per finire con alcune note relative alla configurazione, tramite interfaccia grafica, dei principali parametri di funzionamento.

Un tema collaterale trattato alla fine del documento è il tema del “debug” HW/SW, delle funzionalità presenti a tale fine nel progetto e delle modalità di testing dell’insieme HW/SW in caso di sviluppo di nuovo SW.

A partire dalla versione SW Vr 1.0.9.2 sono state introdotta una serie di piccole modifiche per rendere possibile l’utilizzazione della componente SW del progetto non solo su piattaforme HW SARIMESH, ma anche sui classici “schedini cinesi”, attualmente molto utilizzati per applicazioni LoRa e che montano chips LoRa di prima generazione.

Le funzioni aggiunte hanno riguardato in particolare il supporto dei chips LoRa di prima generazione, introducendo una procedura per semplificare la fase di tuning della frequenza utilizzata in RX/TX in modo da consentire l’uso di questi dispositivi anche con valori di larghezza di banda inferiore a 62.5 Khz, e la possibilità di modificare da GUI una serie di parametri legati alla architettura HW dei circuiti, ovvero in particolare la possibilità di

modificare da GUI i pins utilizzati dei vari chips presenti, in modo da adeguarsi alle specificità dei vari schedini.

Una ulteriore funzionalità introdotta a partire dalla stessa release SW è il supporto, in aggiunta alla modalità di incapsulamento AX.25 per i pacchetti dati APRS, della modalità di incapsulamento utilizzata in altre implementazioni LoRa APRS attualmente esistenti e denominata OE_Style; in particolare i dispositivi che montano questa release di SW possono ricevere traffico APRS indifferentemente caratterizzati da incapsulamento AX.25 o OE_Style; il traffico LoRa uscente potrà, da interfaccia GUI, essere impostata per essere dotato dell'uno o dell'altro tipo di incapsulamento.

Questa feature consente di utilizzare in una stessa rete LoRa APRS sia dispositivi che utilizzano SW SARIMESH che dispositivi che utilizzano altre implementazioni LoRa APRS utilizzando l'incapsulamento OE_Style.

2 Note di montaggio HW

Allo stato esistono due varianti HW del circuito LoRa_Beacon: le due varianti si differenziano per il tipo di moduli fisici/funzionali supportati e per le dimensioni fisiche del prodotto finale, fermo restando il supporto delle funzioni essenziali per consentire l'utilizzo di entrambi i tipi di PCB per realizzare le principali funzioni (Tracker ed iGate APRS LoRa).

2.1 Versione LoRa_Beacon_2020_vr4_1 (alias mini-Tracker)

Le figure a seguire rappresentano lo schema elettrico e il PCB di questa versione, che si caratterizza per essere di **dimensioni molto contenute e destinata ad un uso mobile**.

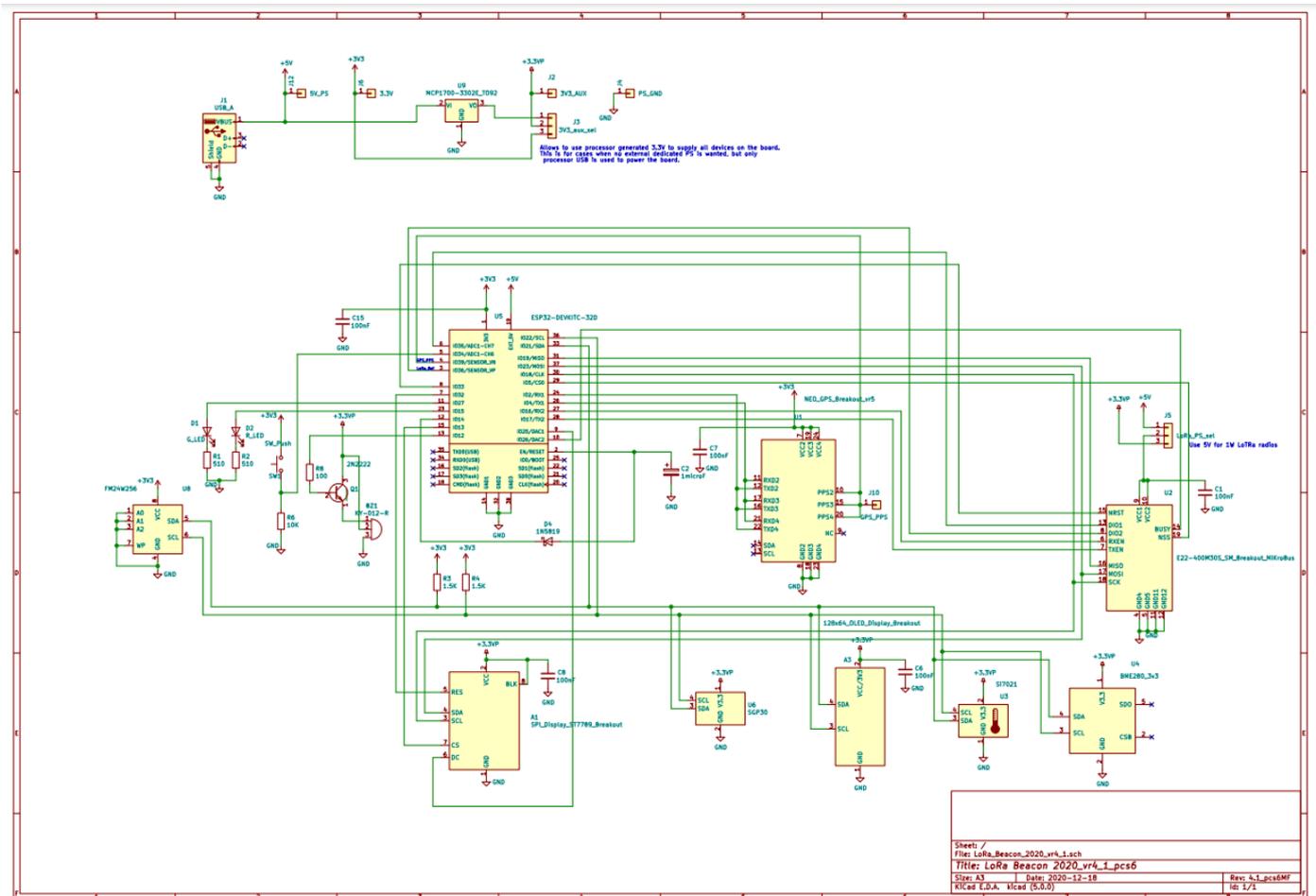


Figura 1 Schema elettrico Versione LoRa_Beacon_2020_vr4.1_pcs6

Questa versione è prevista per essere alimentata tramite un cavo USB (quindi a 5V) utilizzando una qualsiasi fonte tipo alimentatore di telefono cellulare, presa USB di auto o un qualsiasi powerbank previsto per alimentare un classico telefono cellulare.

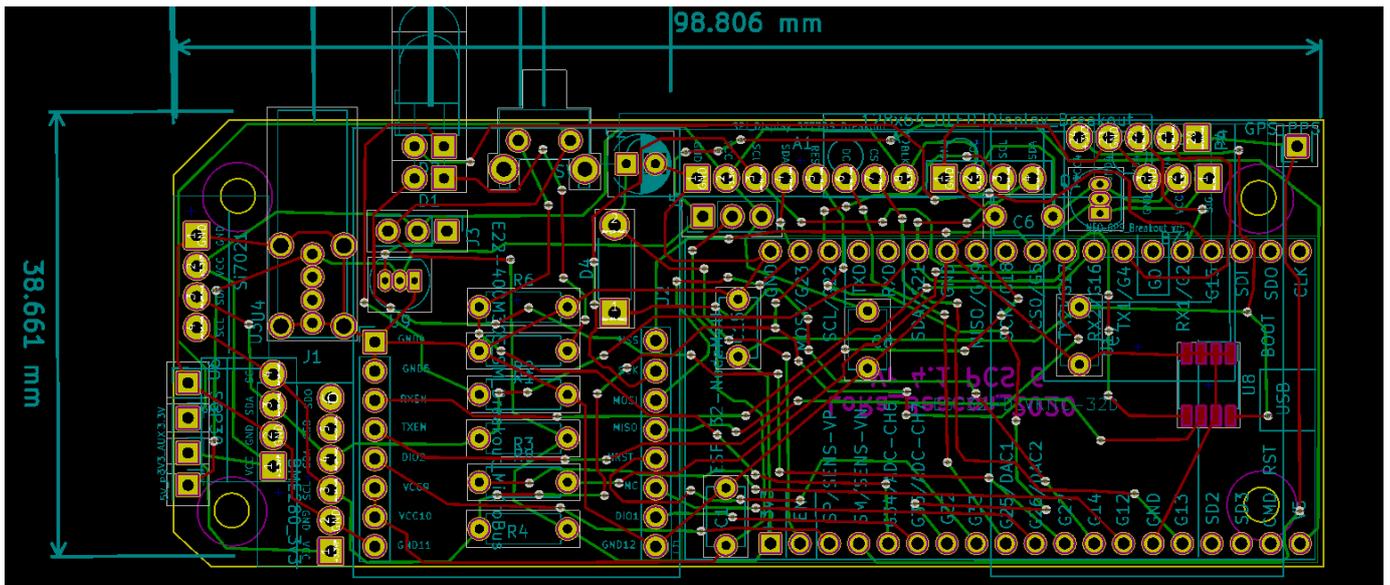


Figura 2 Layout LoRa_beacon_2020_Vr_4.1_pcs6

A seguire la part list del circuito

1	A1 - SPI_Display_ST7789_Breakout : APRS_Mini_Tracker:SPI_Display_ST7789_breakout
2	A3 - 128x64_OLED_Display_Breakout : APRS_Mini_Tracker:128x64_OLED_Vert_Display_breakout
3	B21 - KY-012-R : APRS_Mini_Tracker:KY-012-R_breakout
4	C1 - 100nF : Capacitors_THT:C_Disc_D5.1mm_W3.2mm_P5.00mm
5	C2 - 1microF : Capacitors_THT:CP_Radial_D5.0mm_P2.50mm
6	C6 - 100nF : Capacitors_THT:C_Disc_D5.1mm_W3.2mm_P5.00mm
7	C7 - 100nF : Capacitors_THT:C_Disc_D5.1mm_W3.2mm_P5.00mm
8	C8 - 100nF : Capacitors_THT:C_Disc_D5.1mm_W3.2mm_P5.00mm
9	C15 - 100nF : Capacitors_THT:C_Disc_D5.1mm_W3.2mm_P5.00mm
10	D1 - G_LED : APRS_Mini_Tracker:LED_D5.0mm_Horizontal_O3.81mm_Z5.0mm
11	D2 - R_LED : APRS_Mini_Tracker:LED_D5.0mm_Horizontal_O3.81mm_Z5.0mm
12	D4 - 1N5819 : Diodes_THT:D_DO-41_SOD81_P7.62mm_Horizontal
13	J1 - USB_A : USB_A:USB_A_Vertical
14	J2 - 3V3_AUX : APRS_Mini_Tracker:Pin_Header_Straight_1x01_Pitch2.54mm_local
15	J3 - 3V3_aux_sel : Pin_Headers:Pin_Header_Straight_1x03_Pitch2.54mm
16	J4 - PS_GND : APRS_Mini_Tracker:Pin_Header_Straight_1x01_Pitch2.54mm_local
17	J5 - LoRa_PS_sel : Pin_Headers:Pin_Header_Straight_1x03_Pitch2.54mm
18	J6 - 3.3V : APRS_Mini_Tracker:Pin_Header_Straight_1x01_Pitch2.54mm_local
19	J10 - GPS_PPS : APRS_Mini_Tracker:Pin_Header_Straight_1x01_Pitch2.54mm_local
20	J12 - 5V_PS : APRS_Mini_Tracker:Pin_Header_Straight_1x01_Pitch2.54mm_local
21	Q1 - 2N2222 : TO_SOT_Packages_THT:TO-92_Inline_Narrow_Oval
22	R1 - 510 : Resistors_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P7.62mm_Horizontal
23	R2 - 510 : Resistors_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P7.62mm_Horizontal
24	R3 - 1.5K : Resistors_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P7.62mm_Horizontal
25	R4 - 1.5K : Resistors_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P7.62mm_Horizontal
26	R6 - 10K : Resistors_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P7.62mm_Horizontal
27	R8 - 100 : Resistors_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P7.62mm_Horizontal
28	SW1 - SW_Push : APRS_Mini_Tracker:Button_Angled
29	U1 - NEO_GPS_Breakout_vr5 : APRS_Mini_Tracker:NEO_GPS_Breakout_vr7
30	U2 - E22-400M30S_SM_Breakout_MiKroBus : APRS_Mini_Tracker:E22-M40030S_SM3_MiKroBus_Breakout_STACKED
31	U3 - SI7021 : APRS_Mini_Tracker:SI-7021_breakout
32	U4 - BME280_3v3 : APRS_Mini_Tracker:BME-280_breakout
33	U5 - ESP32-DEVKITC-32D : APRS_Mini_Tracker:MODULE_ESP32-DEVKITC-32D_STACKED
34	U6 - SGP30 : APRS_Mini_Tracker:SGP30_breakout
35	U8 - FM24W256 : Housings_SOIC:SOIC-8_3.9x4.9mm_Pitch1.27mm
36	U9 - MCP1700-3302E_TO92 : TO_SOT_Packages_THT:TO-92_Inline_Narrow_Oval

Figura 3 Part list LoRa_Beacon_2020_vr_4.1_pcs6

La figura seguente fornisce una vista forse più leggibile del posizionamento dei vari componenti sul PCB.

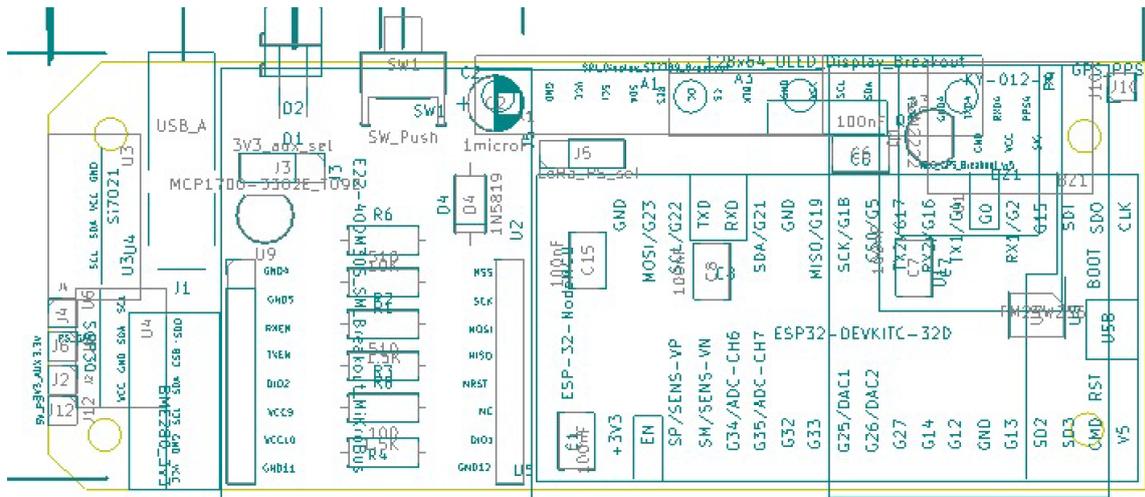


Figura 4 LoRa_Beacon Vr. 4.1-pcs6 : posizionamento componenti principali

La figura seguente fornisce un dettaglio del montaggio dell'unico chip SMD presente sul circuito; il pin 1 corrisponde ad un piccolo forellino presente sul corpo di plastica del dispositivo U8.

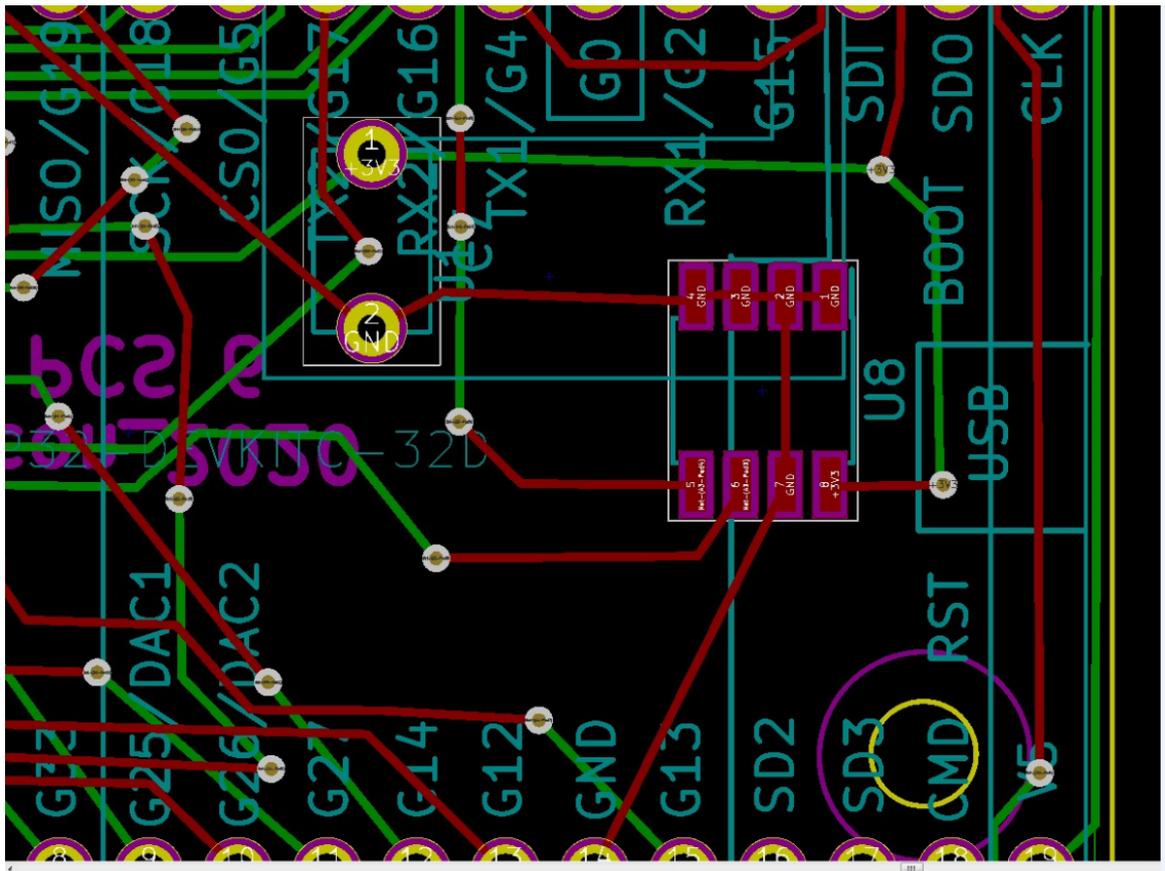


Figura 5 LoRa_Beacon Vr. 4.1_pcs6 : dettaglio montaggio FRAM U8 SMD

La foto seguente riporta una vista del lato componenti del PCB di questa versione

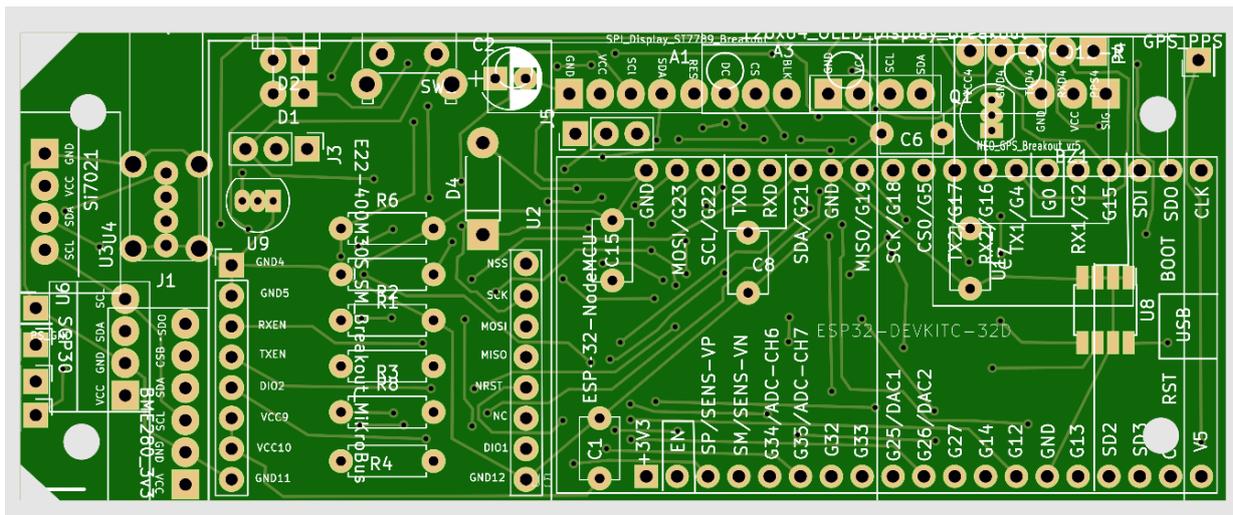


Figura 6 : LoRa_Beacon Vr 4.1_pcs6 foto lato superiore del circuito stampato

Le figure seguenti rappresentano delle foto di un esemplare parzialmente smontato in modo da evidenziare i componenti principali costituenti il circuito completo.

Da aggiungere

2.2 Versione LoRa_Beacon_2020_vr3_pcs4 (alias iGate)

Le figure a seguire rappresentano lo schema elettrico e il PCB di questa versione, che si caratterizza per essere di dimensioni maggiori della versione precedente e che presenta la possibilità di montare una serie di moduli aggiuntivi per consentire una sperimentazione più avanzata rispetto alla versione precedente.

In particolare questa versione si presta in maniera ottimale per essere usata in una installazione fissa ed in particolare per svolgere le funzionalità di iGate; infatti è prevista l'alimentazione a 12V, presenta la possibilità di connettersi ad internet sia in modalità WiFi che tramite interfaccia LAN, e monta un dispositivo RTC (Real Time Clock) per garantire la possibilità di tenuta del tempo locale anche in assenza di connessione ad un dispositivo GPS o alla presenza di connettività internet.

Le dimensioni del PCB sono maggiori della versione precedente in ragione dei moduli aggiuntivi installabili, ma per le parti presenti anche nella versione precedente esiste completa intercambiabilità dei moduli comuni; a livello di schema elettrico le similitudini sono ovviamente elevatissime anche se la numerazione e denominazione dei vari componenti sono diverse.

A seguire i dettagli di questa versione.

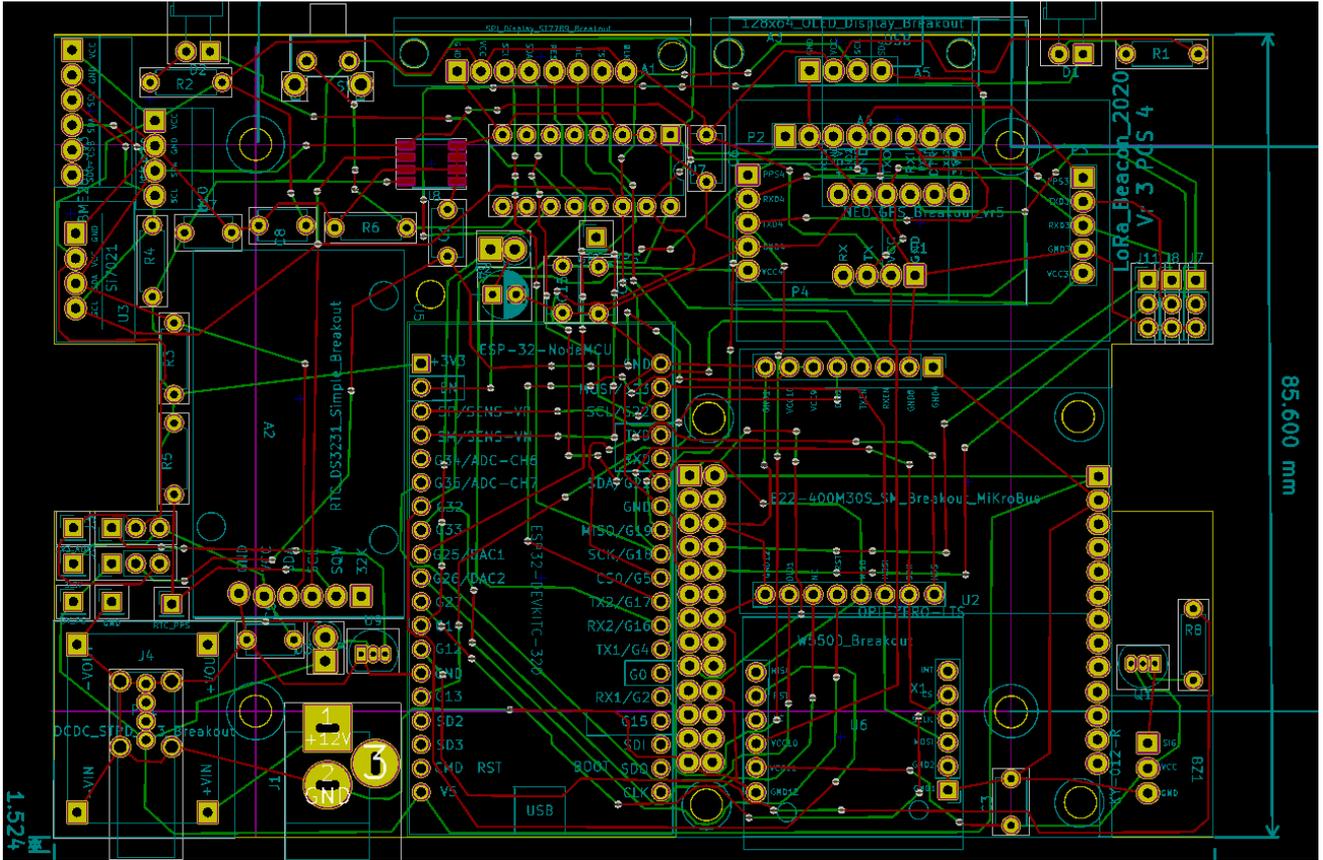


Figura 8 LoRa_Beacon_2020_vr3_pcs4 : Layout PCB carrier

A seguire la part list di questa versione:

1	A1 - SPI_Display_ST7789_Breakout : APRS_Mini_Tracker:SPI_Display_ST7789_breakout
2	A2 - RTC_DS3231_Simple_Breakout : APRS_Mini_Tracker:RTC_DS3231_breakout_simpl
3	A3 - 128x64_OLED_Display_Breakout : APRS_Mini_Tracker:128x64_OLED_Vert_Display_breakout
4	A4 - RS232_Adapter : APRS_Mini_Tracker:RS232_adapter
5	A5 - USB_USART_small : APRS_Mini_Tracker:USB_USART_small
6	BZ1 - KY-012-R : APRS_Mini_Tracker:KY-012-R_breakout
7	C1 - 100nF : Capacitors_THT:C_Disc_D5.1mm_W3.2mm_P5.00mm
8	C2 - 1microF : Capacitors_THT:CP_Radial_D5.0mm_P2.50mm
9	C3 - 100nF : Capacitors_THT:C_Disc_D5.1mm_W3.2mm_P5.00mm
10	C4 - 100nF : Capacitors_THT:C_Disc_D5.1mm_W3.2mm_P5.00mm
11	C5 - 100nF : Capacitors_THT:C_Disc_D5.1mm_W3.2mm_P5.00mm
12	C6 - 100nF : Capacitors_THT:C_Disc_D5.1mm_W3.2mm_P5.00mm
13	C7 - 100nF : Capacitors_THT:C_Disc_D5.1mm_W3.2mm_P5.00mm
14	C8 - 100nF : Capacitors_THT:C_Disc_D5.1mm_W3.2mm_P5.00mm
15	C15 - 100nF : Capacitors_THT:C_Disc_D5.1mm_W3.2mm_P5.00mm
16	D1 - G_LED : APRS_Mini_Tracker:LED_D5.0mm_Horizontal_O3.81mm_Z5.0mm
17	D2 - R_LED : APRS_Mini_Tracker:LED_D5.0mm_Horizontal_O3.81mm_Z5.0mm
18	D3 - 1N5819 : Diodes_THT:D_DO-41_SOD81_P2.54mm_Vertical_KathodeUp
19	D4 - 1N5819 : Diodes_THT:D_DO-41_SOD81_P2.54mm_Vertical_KathodeUp
20	J1 - PWR5_12V : Connectors:JACK_ALIM
21	J2 - 3V3_AUX : APRS_Mini_Tracker:Pin_Header_Straight_1x01_Pitch2.54mm_local
22	J3 - 3V3_aux_sel : Pin_Headers:Pin_Header_Straight_1x03_Pitch2.54mm
23	J4 - USB_A : USB_A:USB_A_Vertical
24	J5 - LoRa_PS_sel : Pin_Headers:Pin_Header_Straight_1x03_Pitch2.54mm
25	J6 - 3.3V : APRS_Mini_Tracker:Pin_Header_Straight_1x01_Pitch2.54mm_local
26	J7 - CON : Pin_Headers:Pin_Header_Straight_1x03_Pitch2.54mm
27	J8 - UART1 : Pin_Headers:Pin_Header_Straight_1x03_Pitch2.54mm
28	J9 - RTC_PPS : APRS_Mini_Tracker:Pin_Header_Straight_1x01_Pitch2.54mm_local
29	J10 - GPS_PPS : APRS_Mini_Tracker:Pin_Header_Straight_1x01_Pitch2.54mm_local
30	J11 - C_SER : Pin_Headers:Pin_Header_Straight_1x03_Pitch2.54mm
31	J12 - 5V_PS : APRS_Mini_Tracker:Pin_Header_Straight_1x01_Pitch2.54mm_local
32	J13 - GND : APRS_Mini_Tracker:Pin_Header_Straight_1x01_Pitch2.54mm_local
33	PS1 - DCDC_STPD_3_3_Breakout : APRS_Mini_Tracker:DCDC_STPD_3_3_TOP_Breakout
34	Q1 - 2N2222 : TO_SOT_Packages_THT:TO-92_Inline_Narrow_Oval
35	R1 - 510 : Resistors_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P7.62mm_Horizontal
36	R2 - 510 : Resistors_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P7.62mm_Horizontal
37	R3 - 1.5K : Resistors_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P7.62mm_Horizontal
38	R4 - 1.5K : Resistors_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P7.62mm_Horizontal
39	R5 - 2.4K : Resistors_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P7.62mm_Horizontal
40	R6 - 10K : Resistors_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P7.62mm_Horizontal
41	R8 - 100 : Resistors_THT:R_Axial_DIN0207_L6.3mm_D2.5mm_P7.62mm_Horizontal
42	SW1 - SW_Push : APRS_Mini_Tracker:Button_Angled
43	U1 - NEO_GPS_Breakout_vr5 : APRS_Mini_Tracker:NEO_GPS_Breakout_vr5
44	U2 - E22-400M30S_SM_Breakout_MiKroBus : APRS_Mini_Tracker:E22-M40030S_SM3_MiKroBus_Breakout
45	U3 - Si7021 : APRS_Mini_Tracker:SI-7021_breakout
46	U4 - EME280_3v3 : APRS_Mini_Tracker:EME-280_breakout
47	U5 - ESP32-DEVKITC-32D : APRS_Mini_Tracker:MODULE_ESP32-DEVKITC-32D
48	U6 - W5500_Breakout : APRS_Mini_Tracker:W5500_Breakout
49	U7 - PCF8574 : APRS_Mini_Tracker:PCF8574_DIP
50	U8 - FM24W256 : Housings_SOIC:SOIC-8_3.9x4.9mm_Pitch1.27mm
51	U9 - MCP1700-3302E_I092 : TO_SOT_Packages_THT:TO-92_Inline_Narrow_Oval
52	U10 - SGP30 : APRS_Mini_Tracker:SGP30_breakout
53	X1 - OPI-ZERO-LTS : APRS_Mini_Tracker:Controller_Vr1

Figura 9 LoRa_Beacon_2020_vr3_pcs4 : part list

A seguire una vista del posizionamento dei componenti:

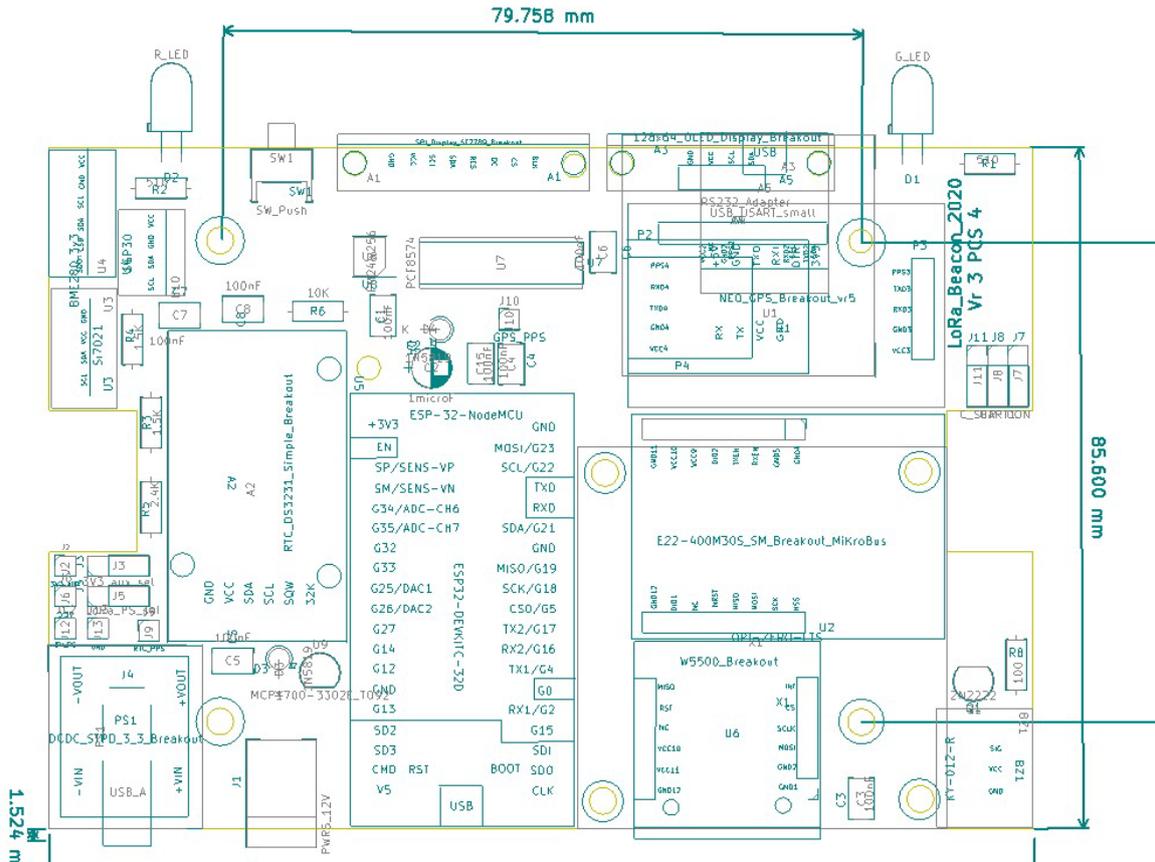


Figura 10 LoRa_Beacon_2020_vr3_pcs4 : posizionamento componenti principali

A seguire il dettaglio di montaggio dell'unico componente SMD U8

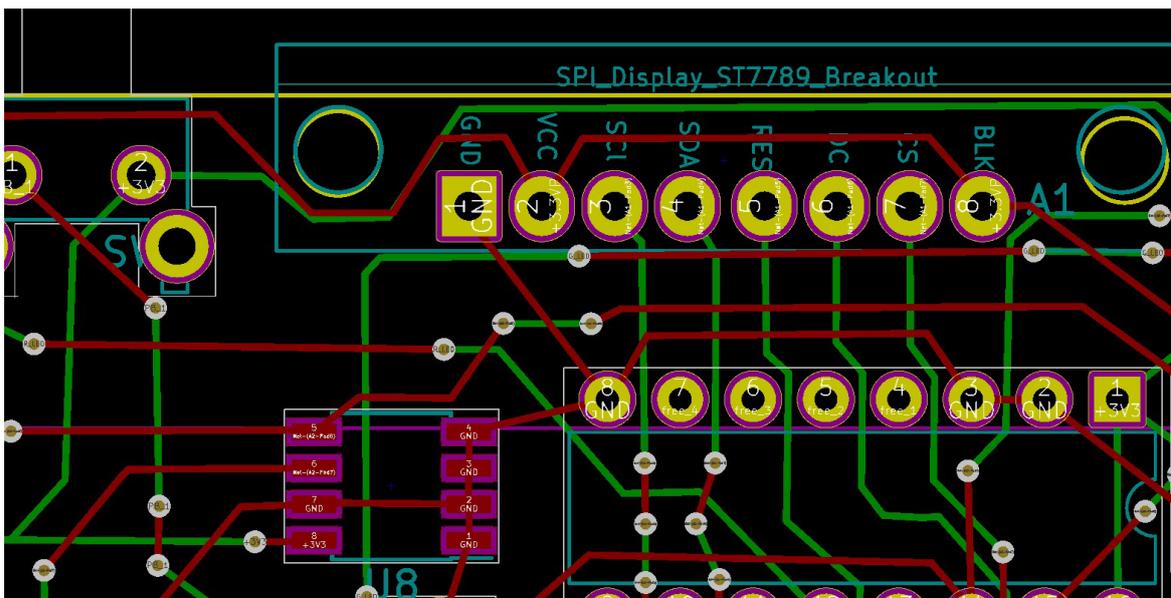


Figura 11 LoRa_Beacon_2020_vr3_pcs4 : dettaglio posizionamento FRAM U8 SMD

La figura seguente è una foto del lato componenti del PCB di questa versione

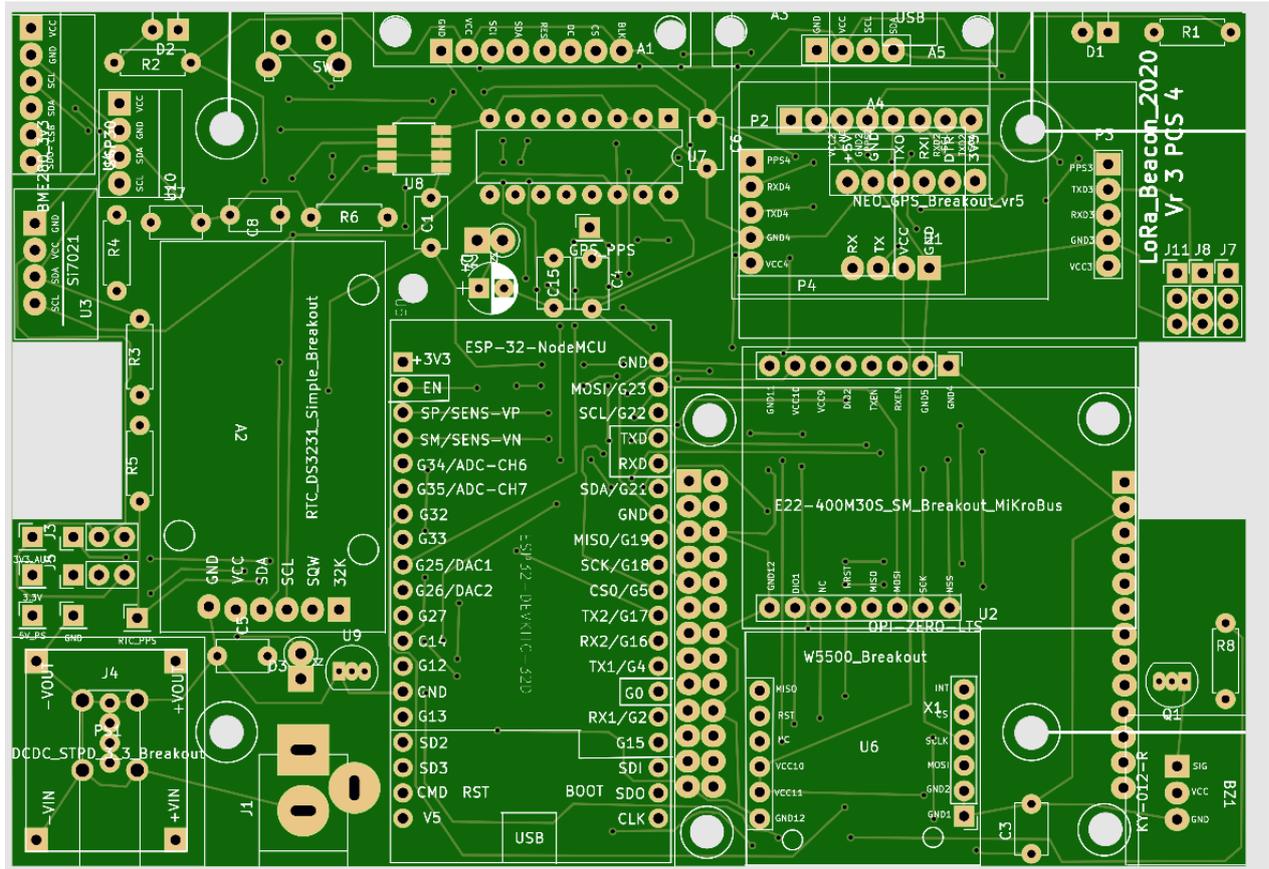


Figura 12 . LoRa_Beacon_Vr 3 pcs 4 : foto lato superiore PCB

2.3 Carrier per moduli radio LoRa (LoRa carrier)

I moduli radio LoRa disponibili sul mercato hanno purtroppo una notevole diversità di pin layout e formato fisico, in genere quasi sempre non compatibile con una pinnatura del tipo 2.54mm che caratterizza la maggioranza degli altri moduli richiesti per la nostra implementazione.

Da questa evidenza è nata l'esigenza di progettare un carrier ad hoc con cui adattare il pinout dei vari moduli LoRa presenti sul mercato ad un unico pinout a spaziatura 2.54mm da usare per montare tali moduli sul carrier principale.

Allo stato esistono diversi di questi carrier; a seguire si documenta quello che più verosimilmente potrà tornare utile almeno inizialmente.

Le figure a seguire rappresentano lo schema elettrico e il PCB di questa versione di carrier LoRa:

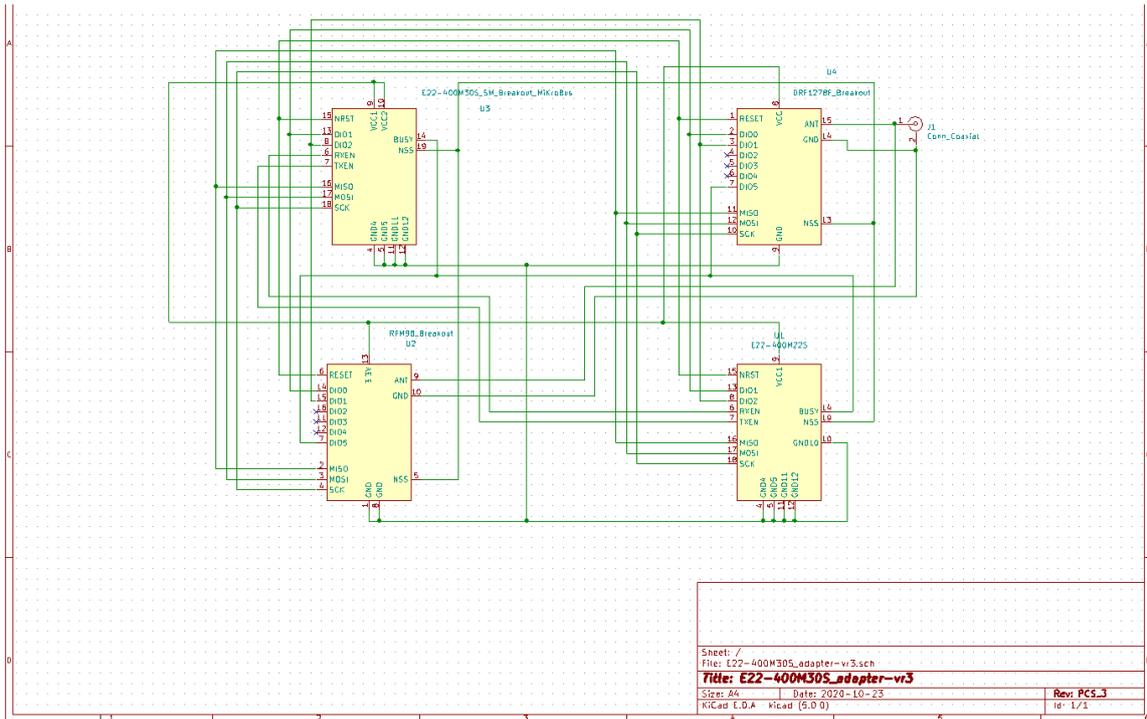


Figura 13 LoRa carrier : schema elettrico

Come si potrà notare la funzione di questo piccolo PCB è quella di consentire di adattare fisicamente i diversi moduli per essere montati sui PCB principali condividendo una piedinatura a 2.54 mm identica per i diversi tipi di moduli LoRa. Questo consentirà verosimilmente di intercambiare diversi moduli LoRa sugli stessi PCB principali per poter fare eventuali confronti.

Il PCB come si potrà notare presenta numerose impronte diverse tra loro: ovviamente una sola delle impronte andrà usata per volta; per montare chips LoRa diversi si costruiranno quindi diversi esemplari di carrier + chip LoRa.

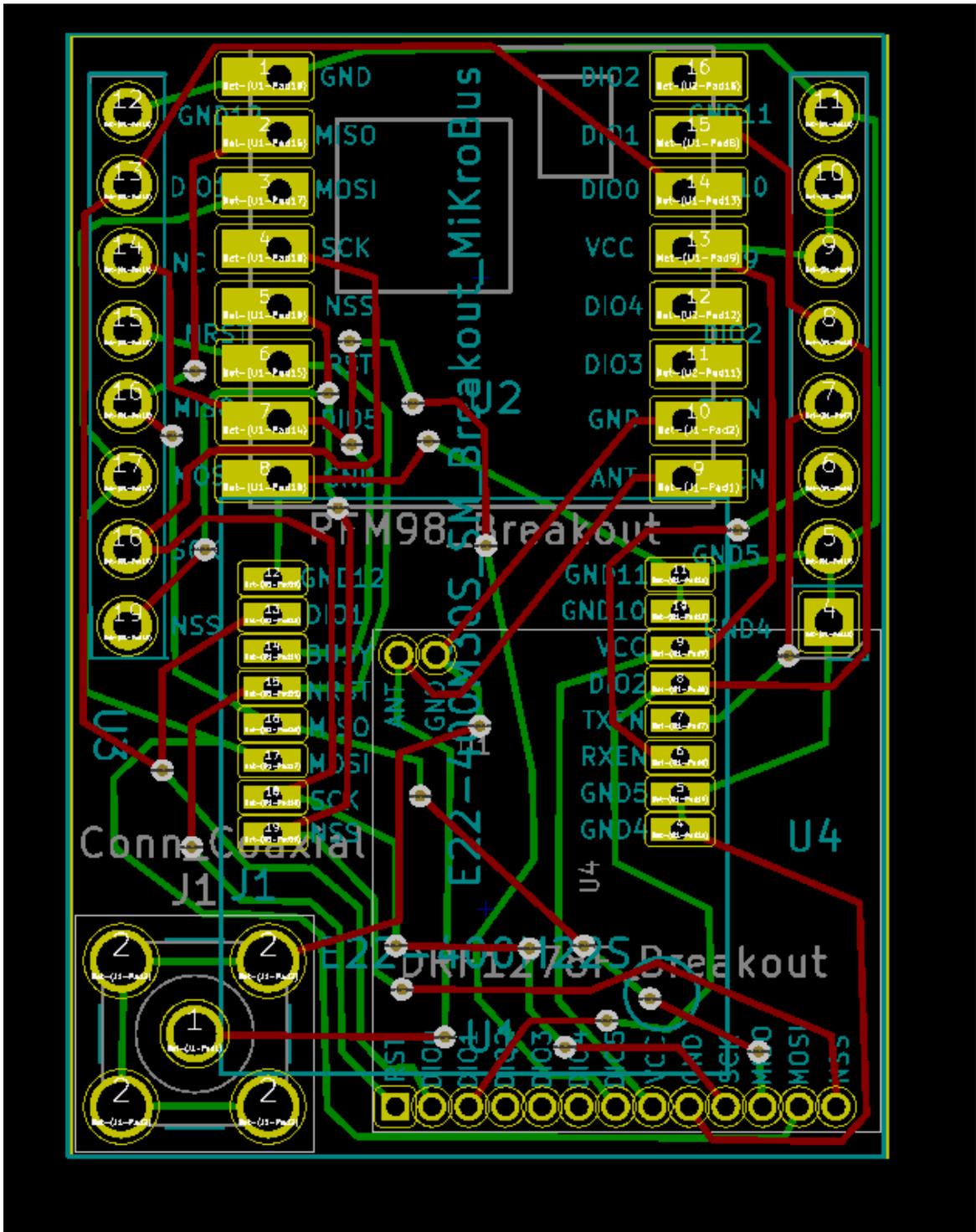


Figura 14 LoRa carrier : PCB layout

La figura seguente è una foto del lato superiore del PCB

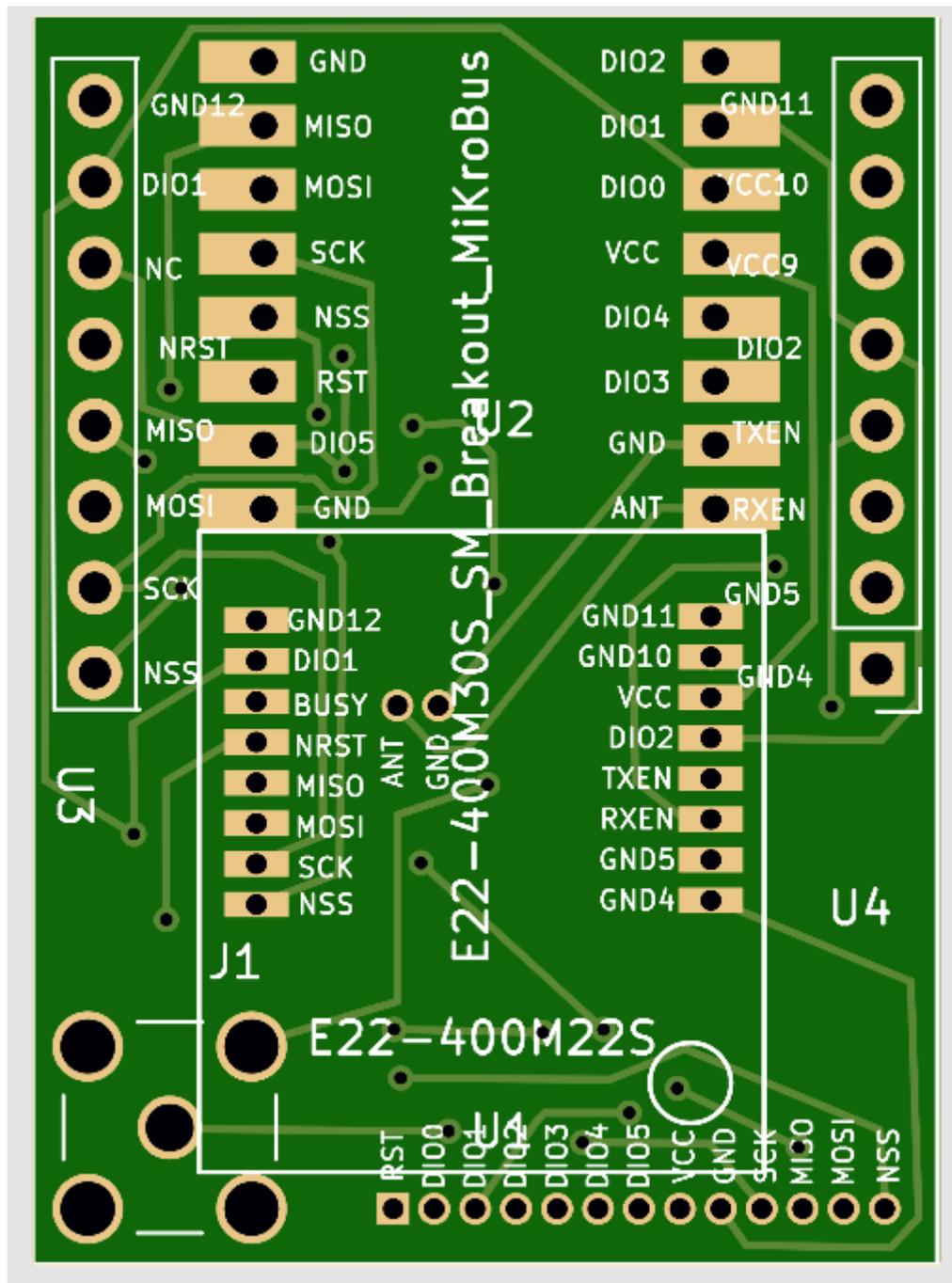


Figura 15 LoRa_Carrier : vista superiore del PCB

Il caso verosimilmente più frequente è quello in cui il modulo radio LoRa sia il tipo E22-400M30S che rappresenta il modulo radio attualmente a maggiori caratteristiche tecniche.

In questo specifico caso le dimensioni fisiche del modulo radio sono tali che già il modulo presenta una spaziatura dei pin di 2.54 mm anche se con un layout per montaggio SMD. In questo caso per il montaggio si sfrutterà il carrier indicato che consentirà di dotare il modulo di una piedinatura standard a 2.54 mm simile a tutti gli altri moduli usati nel progetto, consentendo di montare in maniera ottimale il modulo radio. Le figure seguenti illustrano la modalità di montaggio di questo modulo.



Figura 16 LoRa carrier : LoRa module montato su relativo PCB carrier



Figura 17 LoRa carrier : LoRa module montato su relativo PCB carrier

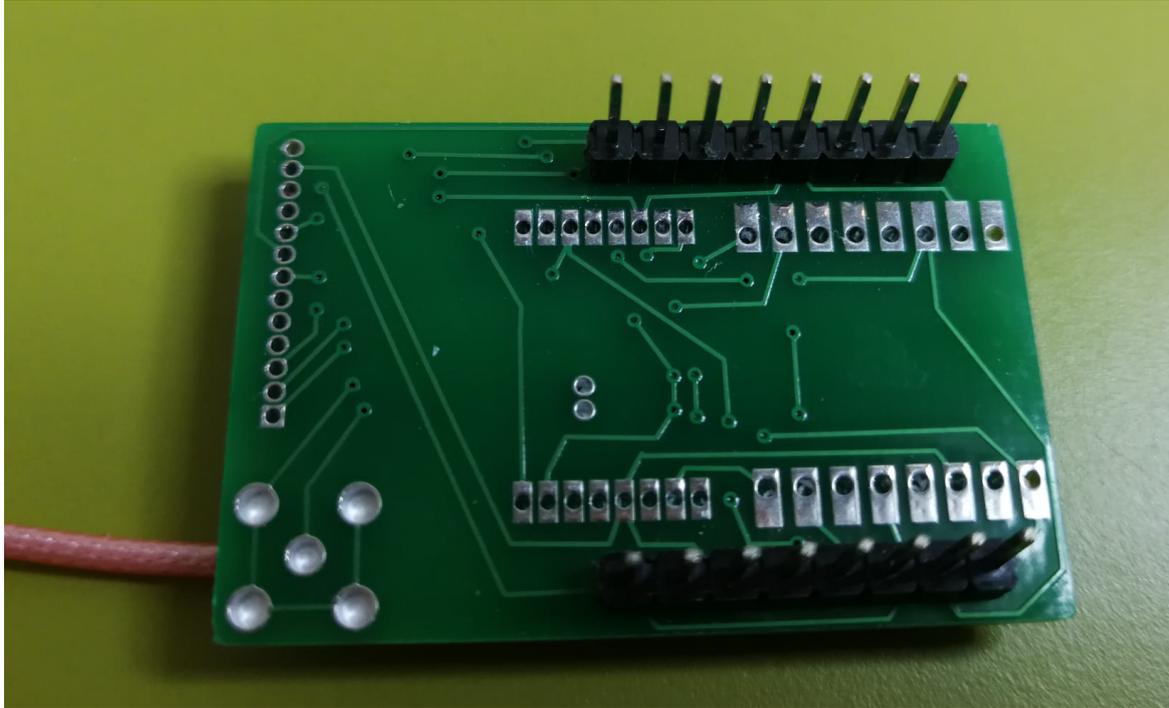


Figura 18 LoRa carrier : LoRa module montato su relativo PCB carrier lato inferiore

2.4 Note di montaggio relative ad entrambe le versioni di PCB carrier.

Per entrambe le versioni di PCB carrier principale esistono una serie di opzioni di montaggio inserite a solo scopo di flessibilità e per far fronte ad eventuali specifiche applicazioni.

La prima nota riguarda i moduli di display: entrambe le versioni presentano le impronte ed i collegamenti per poter montare sia un display a colori che un display monocromatico; i due display hanno interfacce diverse e possono essere utilizzati in alternativa o anche simultaneamente tramite una opportuna eventuale customizzazione del SW; nella versione standard del SW attualmente disponibile vengono supportati simultaneamente un modulo TFT a colori con interfaccia SPI ed un modulo OLED con interfaccia I2C, mostrando però lo stesso tipo di contenuti (questo consente di utilizzare alternativamente uno dei due moduli indifferentemente, senza nessun tipo di modifica SW).

Una ulteriore nota riguarda sempre il modulo display a colori che a seconda della versione acquistata può avere o meno i pin già saldati sul relativo PCB: per un montaggio ottimale si rende necessario usare dei pin con piegatura a 90°; qualora già sul modulo siano saldati dei pin diritti si rende necessario rimuoverli e sostituirli con dei pin piegati a 90°; per semplificare il lavoro di dissaldatura dei pin originali conviene con una tronchesina tagliare il supporto di plastica che lega i pin originali in modo che sia poi più agevole rimuoverli dissaldandoli e scuotendo gentilmente il modulo per farli cadere....

Per tutti i moduli da montare si consiglia di montare sul PCB carrier una fila di connettori a passo 2.54mm di tipo femmina in cui inserire poi il generico modulo dotato di pinnatura maschio sempre a 2.54mm di spaziatura.

Questo consente in futuro di poter agevolmente sostituire o intercambiare i moduli senza complicate operazioni di dissaldatura che inevitabilmente finirebbero per rovinare i circuiti.

Per entrambe le versioni di PCB i moduli sensore sono opzionali e attualmente non supportati nel SW base.

Per la versione iGate il modulo LAN è richiesto unicamente in caso si voglia utilizzare come modalità di connessione internet una connessione di tipo LAN anziché wifi; di default questa modalità non è abilitata.

Il modulo GPS è mandatorio per la versione tracker mentre è opzionale per la versione iGate.

Il modulo RTC per la versione iGate è supportato da SW ma è comunque opzionale.

Il modulo GPS è stato previsto su entrambi i PCB carrier con una impronta che si può adattare a diversi tipi di moduli reperibili sui soliti portali di acquisto internet; per ulteriore dettagli sui moduli compatibili contattare lo scrivente per e-mail

Il PCB iGate presenta numerose ulteriori possibilità di customizzazione in quanto a moduli equipaggiabili che in questa fase non vengono ancora documentati in quando non ancora sufficientemente testati a livello di supporto SW.

La versione iGate presenta due opzioni di alimentazione: o tramite connessione a 5V tramite cavo USB o tramite connessione a 12V e connettore standard 12V DC; la versione di default prevede di utilizzare l'alimentazione a 12V e quindi richiede il montaggio del modulo DC/DC PS1 ed il relativo connettore di alimentazione.

Esistono infine alcune predisposizioni effettuabili tramite opportuni jumpers con formato 2.54mm che si vanno di seguito a documentare:

Versione PCB mini-tracker: jumpers e loro significato

J3: 3V3_aux_sel default connettere pins 1-2

J5: LoRa_PS_sel utilizzare la posizione 1-2 per il modulo LoRa E22-400M30S che opera a 5V; posizione 2-3 per i moduli LoRa che lavorano a 3.3V

Versione PCB iGate: jumpers e loro significato

J3: 3V3_aux_sel default connettere pins 1-2

J5: LoRa_PS_sel utilizzare la posizione 1-2 per il modulo LoRa E22-400M30S che opera a 5V; posizione 2-3 per i moduli LoRa che lavorano a 3.3V

J7: per future espansioni nessun jumper da collegare

J8: per future espansioni nessun jumper da collegare

J11: per future espansioni nessun jumper da collegare

Come nota generale si consiglia di testare il circuito collegando gradualmente i vari moduli, iniziando dal processore ESP32 e continuando con il display, il GPS e infine il modulo LoRa.

Allo scopo di consentire eventualmente delle azioni di troubleshooting sono disponibili dei semplici “test segments” ovvero dei piccoli programmini che caricati opportunamente , tramite l’ambiente di sviluppo SW Arduino, sul processore ESP32 possono consentire di testare i vari blocchi funzionali singolarmente o a piccoli gruppi.

La descrizione delle possibili azioni di test atte a fissare eventuali malfunzionamenti in fase di HW setup viene rimandata ad una successiva appendice di questo documento.

3 Installazione SW Iniziale

Il microcontrollore utilizzato nel progetto LoRa_Beacon è l'ESP32: si tratta di un dispositivo ad elevatissima scala di integrazione che contiene al suo interno una significativa mole di funzionalità che qui si evita di descrivere per brevità; su internet è possibile trovare moltissima documentazione in merito.

Il processore è acquistabile sotto forma di moduli con piedinatura standard a 2.54mm che contengono oltre al microcontrollore anche una serie di altri componenti tra cui una interfaccia USB tramite la quale è possibile interagire direttamente con il dispositivo sia per funzioni di sviluppo SW che per funzioni di monitoraggio diretto del funzionamento del dispositivo e di caricamento iniziale del SW.

La figura seguente riporta una foto del modulo processore e delle sue interfacce fisiche. Vale la pena osservare che in commercio esistono diverse varianti di questo modulo che differiscono per il numero di piedini e per il chip processore utilizzato; il nostro progetto richiede di usare la versione di modulo a 38 pins con antenna WiFi on board (stampata), come chiaramente mostrato nelle figure a seguire.

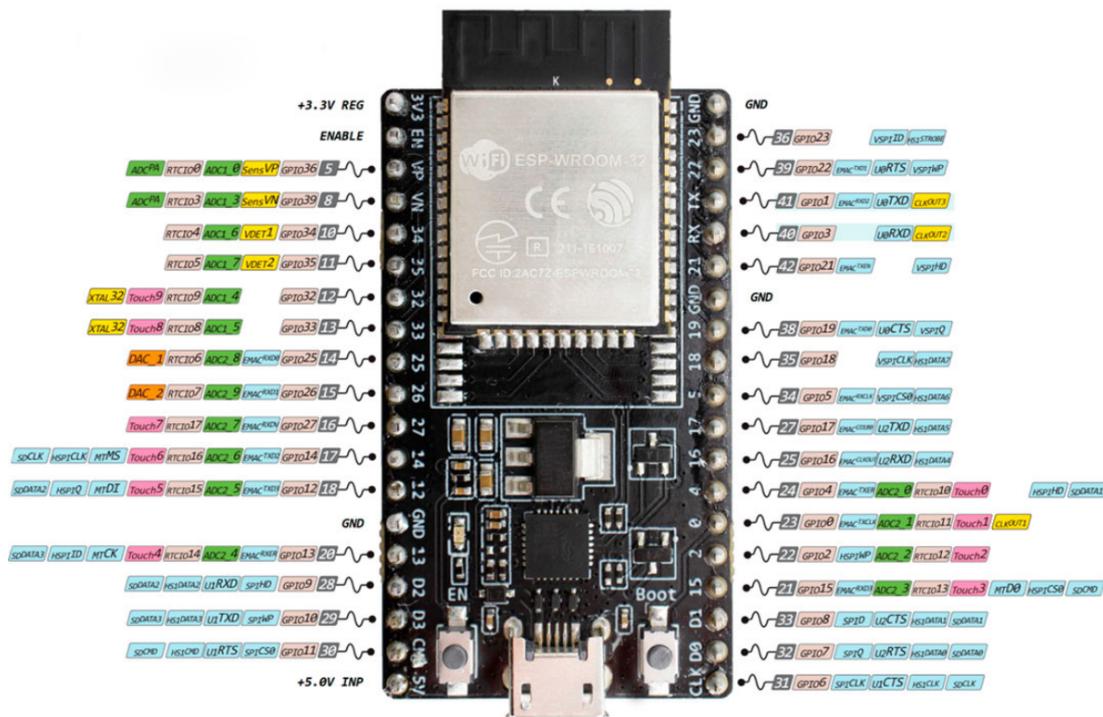


Figura 19 ESP32 modulo processore con relative interfacce

Il modulo utilizzato come processore contiene in sé tutto il necessario per poter funzionare autonomamente; in particolare contiene una memoria flash per contenere il programma SW, una memoria RAM da usare come memoria di lavoro, una interfaccia USB per il caricamento del SW e per il controllo del processore ed una interfaccia radio WiFi per poter interagire con il modulo via WiFi senza ulteriori componenti esterni richiesti.

Al momento dell'acquisto il processore arriva già dotato di un particolare SW che consente di caricare tramite l'interfaccia USB il SW utente necessario allo specifico progetto.

Questo SW consente sia di interagire con il processore direttamente tramite una semplice interfaccia grafica atta a caricare il SW richiesto, sia di interagire con il processore tramite un ambiente di sviluppo SW ad hoc.

Esistono vari ambienti di sviluppo possibili; quello utilizzato nel corso del progetto LoRa_Beacon è basato sulla piattaforma SW Arduino IDE.

Nel seguito viene illustrato il primo metodo di caricamento del SW, lasciando la descrizione della seconda modalità ad un altro documento.

3.1 Setup ambiente di caricamento SW e caricamento Immagine SW iniziale

L'operazione di caricamento del SW sul processore ESP32 richiede la predisposizione di un opportuno programma su un PC dotato di sistema operativo Windows o Linux.

E' richiesto di effettuare il collegamento tra il PC e il processore utilizzando un cavetto USB intestato con connettori appropriati.

All'atto del collegamento tramite il cavetto USB il processore verrà alimentato tramite il cavetto stesso e stimolerà il PC (Windows) ad installare automaticamente (in genere) i drivers che consentono di visualizzare il modulo come una interfaccia seriale per il computer.

Per scoprire l'identità della porta seriale con cui viene visto il modulo ESP32 è sufficiente esplorare la lista dei dispositivi del PC (tramite il pannello di controllo) ; su piattaforma Linux in genere il riconoscimento della nuova interfaccia è automatica.

In questa fase il modulo ESP32 può essere programmato anche non collegato al circuito su cui deve essere utilizzato.

Per scaricare da internet il tool di programmazione è possibile usare il seguente URL:
<https://www.espressif.com/en/support/download/other-tools>

La figura seguente mostra la pagina di download del tool e indica la versione da scaricare.

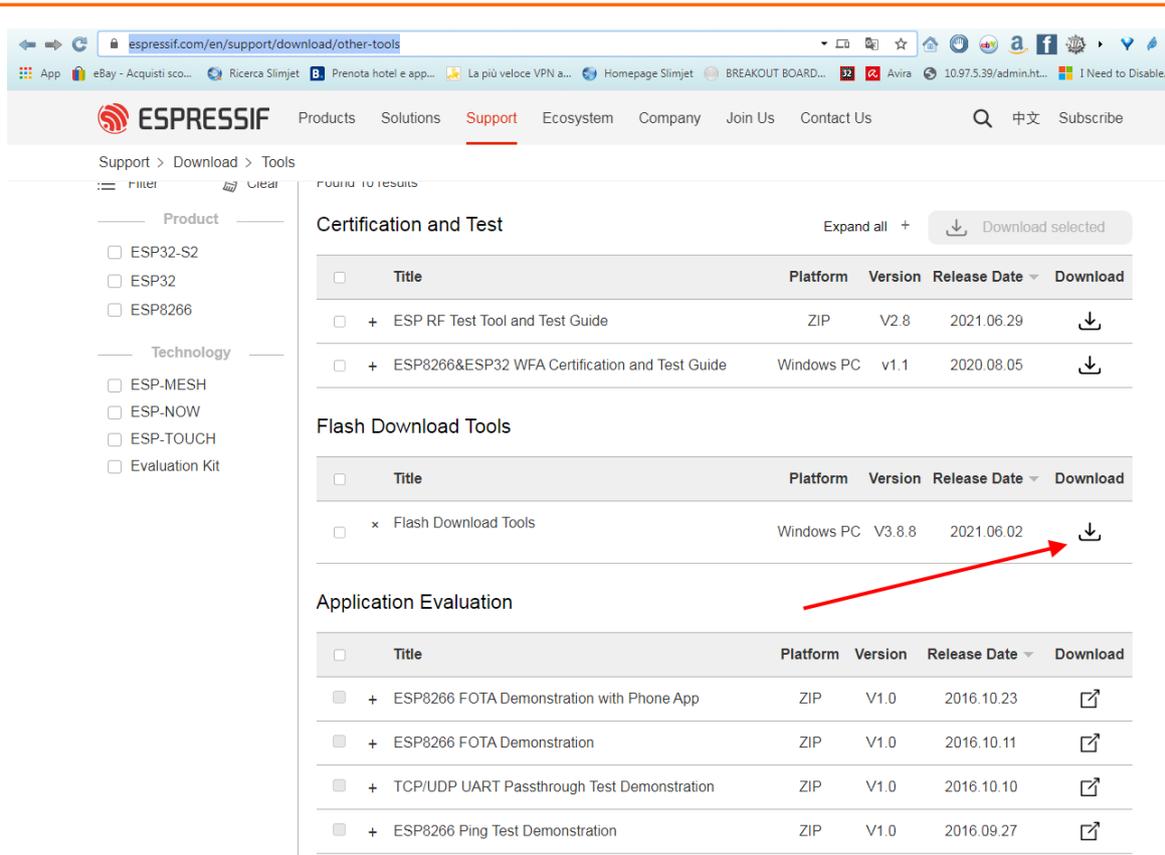


Figura 20 Pagina di download del tool di programmazione per il processore ESP32

Sul sito <http://iot-bits.com/esp32/esp32-flash-download-tool-tutorial/> è possibile anche trovare un piccolo tutorial sull'uso del tool; nel seguito descriviamo il solo caso di utilizzo del tool su ambiente Windows.

Una volta scaricato il tool “Flash Download Tool” sul PC espandere il relativo archivio e lanciare il file `flash_download_tool_3.8.8.exe`; quindi selezionare dal pannello che appare “DOWNLOAD TOOL MODE” i valori “`chip_Type = ESP32`” e “`WorkMode=develop`”; apparirà un pannello da settare come alla figura seguente:

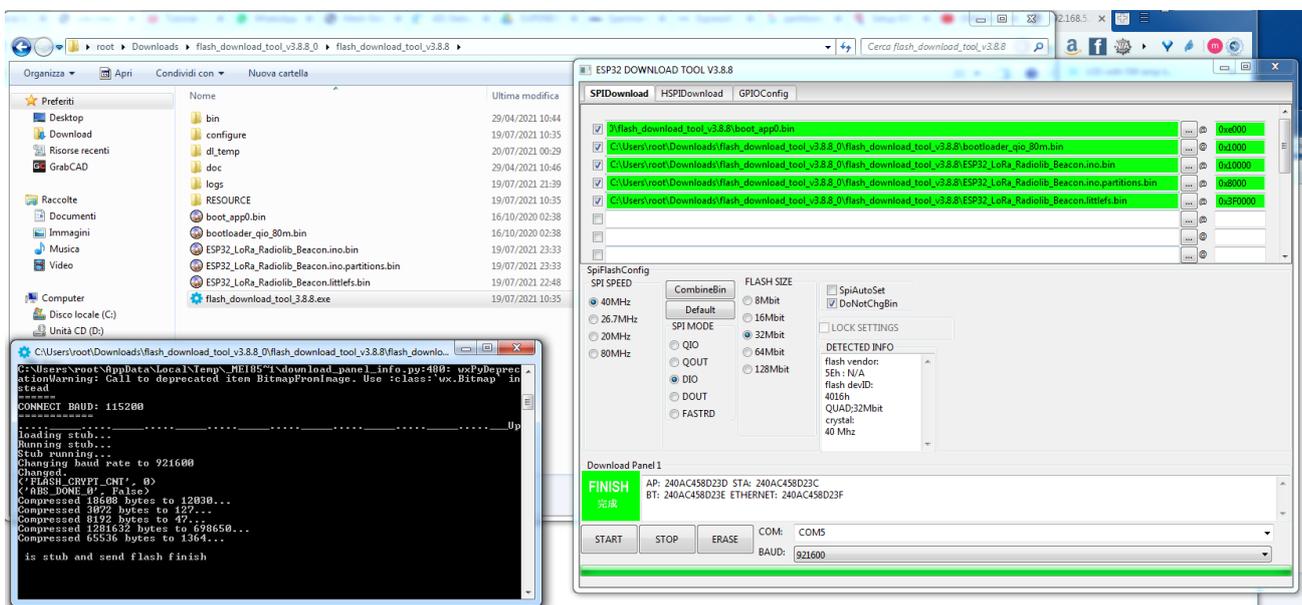


Figura 21 Utilizzo del tool di SW Download su processore ESP32

Le righe in verde sono i vari moduli SW che costituiscono l'immagine completa del SW: si tratta di 5 load modules che corrispondono ad altrettante sezioni della flash.

Il significato dei moduli SW che costituiscono l'immagine completa è il seguente:

- Boot_app di startup del processore
- Boot_loader per il caricamento del SW operativo
- Immagine SW applicativo
- Partition_table della flash
- Partizione LittleFS

La figura seguente è un ingrandimento ; per semplicità l'immagine viene fornita come un file .zip da espandere nel direttorio dove è stato espanso il tool di download del SW ; **per ogni sezione bisogna impostare manualmente nella parte destra della riga corrispondente l'indirizzo esadecimale dove quel pezzettino di SW deve essere caricato.** Questi indirizzi dipendono da come è stata configurata la flash del dispositivo in fase di sviluppo del SW e sono contenuti in un opportuno file readme.txt presente dell'archivio contenente i vari moduli. **Prestare molta attenzione al setup di questi valori, pena la non ripartenza corretta del processore.** In caso di errore è sufficiente ripetere l'operazione di flashing introducendo i valori corretti.

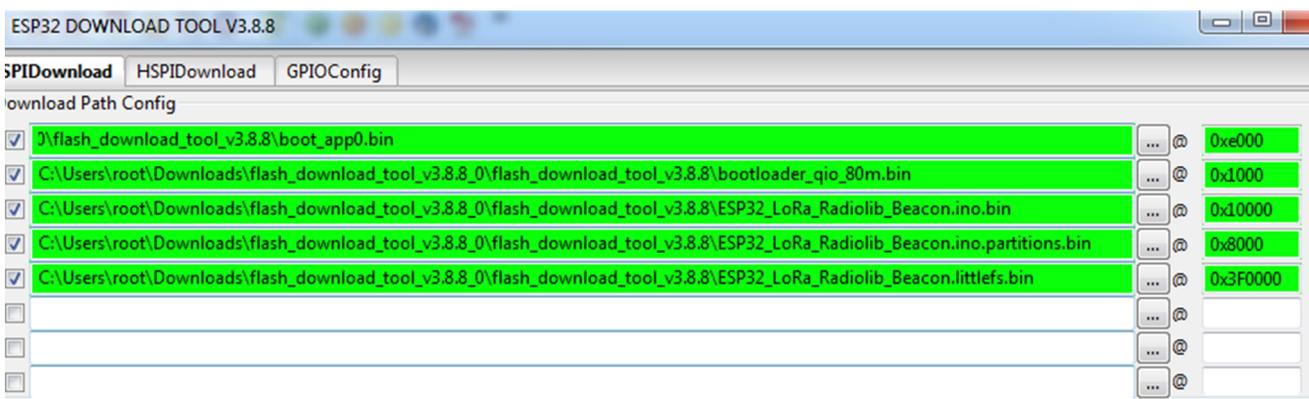


Figura 22 Impostazione load modules e relativi indirizzi in flash

Nella finestrella COM va inserita la porta seriale su cui il computer ha allocato l'interfaccia USB del modulo processore che si intende caricare; per gli altri parametri settarli come in figura. Settare il valore di BAUD a 921600 ; se non va, provare con 115200.

Verificare che in questa fase non sia aperta nessuna altra finestra relativa per es. al sistema di sviluppo Arduino o altra applicazione che utilizzi la seriale su cui è attestato il processore.

Premendo il tasto START il tool provvederà a caricare sul processore l'immagine SW indicata: nella finestra command che accompagna il tool è possibile seguire l'andamento del caricamento del SW; qualora non si noti la partenza della riga verde sul margine inferiore della finestra principale verificare i valori inseriti ed eventualmente provare a modificare il valore del BAUD impostato.

Una volta caricata l'immagine SW montare il modulo processore sul PCB principale e verificare che sul display si abbiano dei segni di vita :).

Provare quindi a verificare con il PC che sia comparsa una nuova rete WiFi con identità del tipo ESP32-<mac address> : selezionare questa rete e avviare il collegamento ad essa (il PC deve essere impostato per acquisire automaticamente l'indirizzo IP). Verificando nelle connessioni di rete dovrà apparire una situazione simile alla figura seguente:

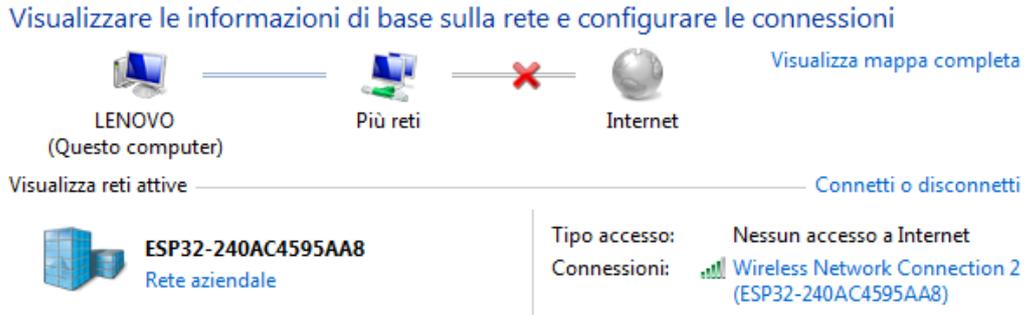


Figura 23 Visualizzazione rete WiFi di gestione del dispositivo

Accedere con un browser tipo chrome o firefox al seguente URL: <http://192.168.5.1> dovrà apparire una schermata come in figura sotto:



Figura 24 Pagina Iniziale GUI

I Valori presenti nelle finestrelle ovviamente dipenderanno dal SW caricato e dal dispositivo specifico che si sta usando. Premendo il tasto “continue” apparirà la schermata principale della GUI, come in figura seguente:

LoRa Tracker Admin

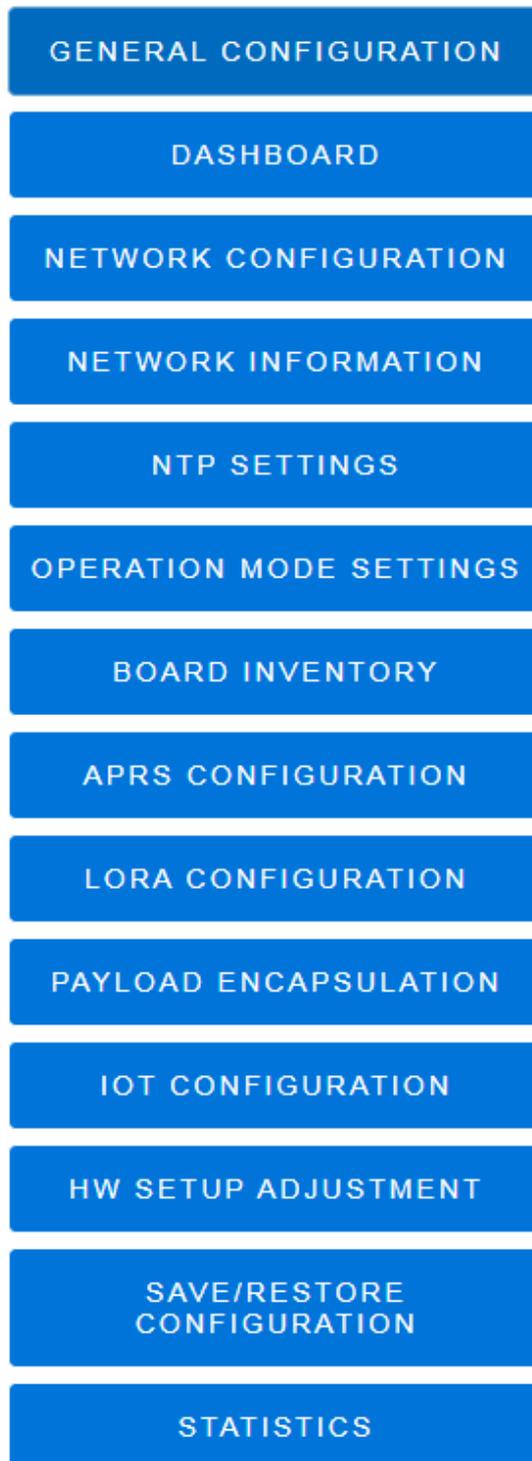


Figura 25 Pagina Principale interfaccia GUI

3.2 Setup iniziale del dispositivo LoRa_Beacon (qualsiasi versione)

L'operazione di caricamento del SW sul processore ESP32 come illustrata al paragrafo precedente si rende necessaria in linea di principio unicamete all'inizio dei tempi: la sua funzione è predisporre un ambiente SW in grado di essere agevolmente configurato tramite la sua interfaccia grafica senza dover ricorrere ad un collegamento diretto tra il PC utilizzato per la gestione ed il dispositivo fisico da gestire, e senza richiedere la predisposizione di un ambiente di sviluppo Arduino necessariamente.

Nella sua configurazione di default il SW non è in grado di operare in maniera appropriata in quanto è **necessario specificare una serie di elementi strettamente dipendenti dallo specifico esemplare** di dispositivo che si sta gestendo, per cui è richiesta una fase di “configurazione del SW installato”.

Per illustrare la configurazione conviene innanzitutto illustrare grossolanamente come è impostata l'interfaccia di gestione e introdurre il concetto di “**modalità operative**” del dispositivo.

Il SW è stato sviluppato per consentire di configurare il dispositivo fisico per poter comportarsi in diversi modi; questi diversi modi di funzionamento sono in generale tali che non è possibile passare da un modo ad un altro senza effettuare una ripartenza da fermo (reboot) del SW.

Esistono inoltre delle attività, quali ad es. quelle di sostituzione del SW e di salvataggio o restore della configurazione operativa, che richiedono parimenti che il processore venga posto in una opportuna modalità di funzionamento caratterizzata dal congelamento di alcune funzioni (ad es. la parte LoRa e la parte GPS) che definiamo “**Admin Mode**”.

Esiste un pannellino nella interfaccia grafica , accessibile tramite la voce “**OPERATION MODE SETTINGS**” che consente di gestire queste diverse modalità operative e che quindi rappresenta la **prima sezione da configurare** dopo il caricamento iniziale del SW.

La figura 26 illustra il contenuto della pagina di cui sopra.

Una prima sezione della schermata consente di settare una serie di modalità di debug utilizzabili per tracciare e risolvere eventuali anomalie di funzionamento o di settare specifiche condizioni di equipaggiamento del dispositivo. Tale sezione inizialmente può essere lasciata nella sua condizione di default.

La seconda sezione della schermata consente di impostare la modalità operativa per il dispositivo; sono possibili varie modalità operative che nel seguito verranno individualmente illustrate.

La terza sezione consente di effettuare il reboot manuale del dispositivo senza bisogno di togliere e rimettere l'alimentazione elettrica al dispositivo.

Particolarmente importante è la modalità “Admin_Mode” come sopra descritto. Tale modalità andrà impostata ogni qualvolta sia richiesta a livello di interfaccia grafica.

La modalità “iGate_Mode” predispose il dispositivo per operare in modalità iGate APRS LoRa e richiede la configurazione di una serie di parametri nella schermata “APRS CONFIGURATION”.

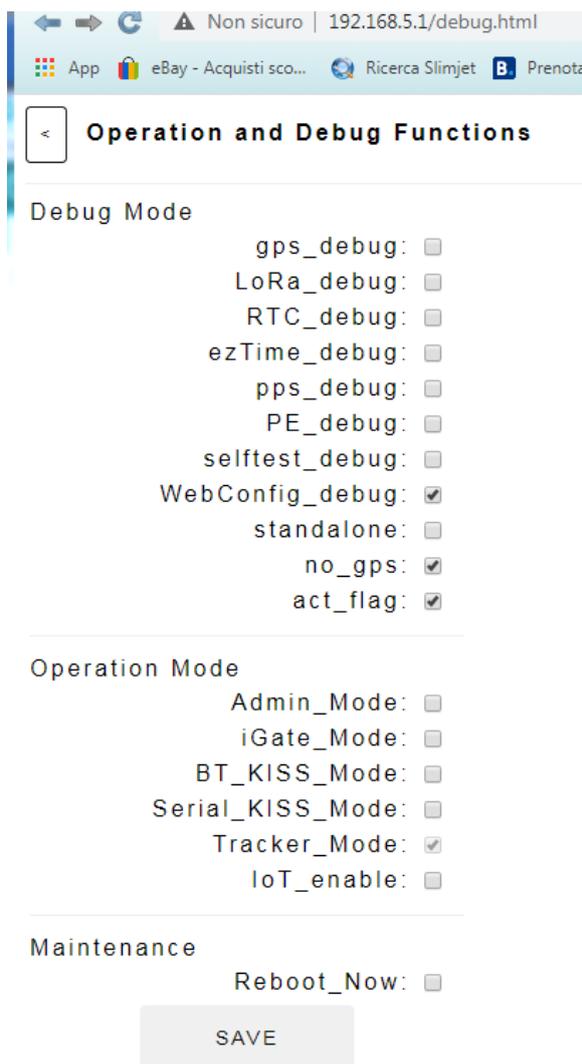


Figura 26 Operation Mode Settings

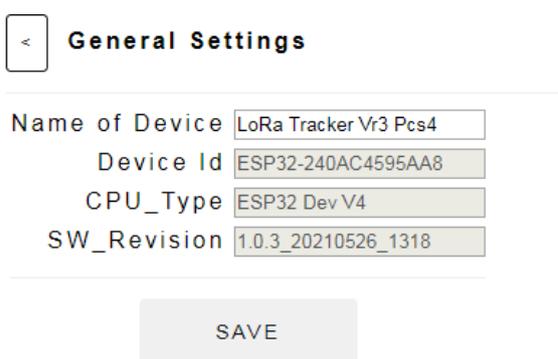


Figura 27 General settings

Qualora questa voce non venga selezionata, viene automaticamente selezionata la modalità “Tracker APRS”.

Le modalità “BT_KISS_Mode”, “Serial_KISS_Mode” sono per il momento non documentate.

La voce “IoT_enable” serve ad abilitare una serie di funzionalità di tipo IoT collegate all’utilizzo della sensoristica presente sul dispositivo ed è allo stato ancora da documentare.

Una ulteriore sezione da configurare inizialmente è rappresentata dalla voce “**GENERAL CONFIGURATION**” della schermata principale della GUI.

Questa sezione riporta una serie di dati relativi alla versione SW, al tipo di processore e alla identità del dispositivo, il tipo di CPU di cui è dotato, e la revisione SW installata e consente di settare manualmente alla prima riga un nome di fantasia per il dispositivo.

Questo nome verrà unicamente utilizzato per individuare il dispositivo ed è quindi liberamente settabile dall’utente.

Una ulteriore sezione da settare inizialmente è la “**NETWORK CONFIGURATION**”: questa sezione consente di settare la modalità di connessione a livello WiFi che si intende utilizzare per il dispositivo.

A tale proposito bisogna illustrare brevemente quali funzionalità WiFi sono implementate sul dispositivo ed il loro utilizzo.

Il dispositivo implementa due tipi di funzionalità WiFi, utilizzabili simultaneamente e per esigenze diverse:

- Modalità di AP Access Point WiFi
- Modalità di tipo “Station WiFi”

La prima modalità consente di accedere al dispositivo da un qualsiasi client WiFi come se fosse un normale punto di accesso WiFi: effettuando con il PC per es. una scansione delle reti WiFi disponibili si noterà la presenza di una rete WiFi con SSID del tipo “ESP32-**<mac_address>**”: collegandosi a tale rete WiFi, avendo impostato il PC per acquisizione dell’indirizzo IP tramite

Network Configuration

Upstream WiFi Network Configuration:

SSID:

Password:

DHCP:

IP:

Netmask:

Gateway:

Dns:

Connection State:
CONNECTED

Networks:
Found 4 Networks

Name	Quality	Enc
RFC2019-24	100%	
Wind3_HUB-3E4E81	54%	
Vodafone-WiFi	38%	
Vodafone-35188680	36%	

Figura 28 Network Configuration

WiFi Upstream Network Status

SSID : RFC2019-24

IP : 192.168.2.232

Netmask : 255.255.255.0

Gateway : 192.168.2.1

Dns : 8.8.4.4

Mac : 24:0A:C4:59:5A:A8

Figura 29 Network Informations

dispositivo. Finito il setup delle varie caselle sarà necessario salvare la configurazione con il tasto “SAVE”.

Il tasto “REFRESH” serve a riacquisire la lista delle reti WiFi disponibili.

Per conoscere lo stato della connessione di rete WiFi si può consultare la sezione “**NETWORK INFORMATION**” della pagina principale della GUI: tale pagina fornirà i valori attualmente in uso da parte del dispositivo per la sua connessione alla rete locale e ad internet.

DHCP, il dispositivo fornirà al PC una configurazione IP tale che sarà possibile collegarsi al dispositivo utilizzando l’indirizzo standard 192.168.5.1.

Il sistema è parimenti utilizzabile usando un qualsiasi dispositivo mobile dotato di Browser tipo Chrome o Firefox. Le schermate della GUI sono ottimizzate per l’uso con dispositivi mobili.

In questo modo si potrà configurare il dispositivo anche senza la presenza di un access point esterno o di una qualsiasi copertura WiFi di accesso ad internet.

Nella modalità “Station WiFi” invece il dispositivo effettuerà autonomamente una scansione delle reti WiFi presenti in zona e presenterà una lista delle SSID ascoltate consentendo di impostare il collegamento del dispositivo ad una di queste reti; la funzionalità riguarda solo le reti a 2.4 Ghz.

La schermata di cui si accennava sopra serve proprio a selezionare la rete WiFi a cui collegare il dispositivo in modalità Station, e in pratica consente di collegare il dispositivo alla propria rete locale e ad internet.

E’ possibile sia specificare per questa connessione l’acquisizione dell’indirizzo IP da parte del dispositivo tramite protocollo DHCP, sia specificare un indirizzo statico manualmente.

Se si accede a questa schermata la prima volta la sezione “Network” potrà apparire vuota... aspettando alcuni secondi verrà automaticamente popolata con la lista delle reti WiFi a 2.4Ghz disponibili: selezionando una delle reti indicate si potrà impostare nella prima sezione della schermata l’eventuale password.

Selezionando la casella DHCP il dispositivo acquisirà automaticamente la configurazione IP dalla rete selezionata.

In assenza della selezione della casella DHCP si potrà specificare manualmente un indirizzamento per il

Figura 30 NTP Settings

Per completare la configurazione dei parametri di rete del dispositivo è necessario specificare un “Network Time Server” da eventualmente utilizzare per la sincronizzazione dell’orario locale del dispositivo in modo da consentire di marcare temporalmente una serie di dati acquisiti durante il funzionamento, quali ad es. gli spot LoRa o i messaggi di evento/errore prodotti dal dispositivo.

Allo scopo si utilizzerà la sezione “NTP SETTINGS” della pagina principale riportata a fianco.

Il valore del server NTP dovrà essere impostato con il nome DNS del server che si intende utilizzare mentre la casella Timezone va selezionata in base al fuso geografico.

Figura 31 Dashboard: modalità di sincronizzazione

La casella Daylight Stting serve ad indicare se è in uso l’orario legale.

Il settaggio di un server NTP è importante in quanto consente di avere correttamente impostato l’orario del dispositivo in particolare in assenza di un modulo GPS come tipicamente avverrà nell’utilizzo come iGate.

Figura 32 Inventory configuration

Infatti il dispositivo è progettato per asservire il suo orologio locale ad un modulo GPS, se presente, o in sua assenza ad un Server NTP raggiungibile via internet.

Per conoscere lo stato di connessione del dispositivo e le sue principali caratteristiche operative è possibile utilizzare la schermata “**DASHBOARD**” presente sulla pagina principale della GUI.

Questa schermata, che verrà descritta nel prosieguo in dettaglio, presenta alle prime righe proprio lo stato di funzionamento del dispositivo dal punto di vista del settaggio del tempo locale.

Una sezione particolarmente importante da consultare è la pagina “**BOARD INVENTORY**” accessibile dalla pagina principale della GUI.

Come conseguenza della sua modularità il dispositivo potrà avere equipaggiati una serie di **moduli opzionali: la maggior parte di questi moduli verrà riconosciuta automaticamente alla partenza del dispositivo** nella sua fase di

configurazione dinamica: il risultato di questa fase viene sinteticamente riportata nella schermata di cui sopra.

In particolare consultando questa pagina si potrà scoprire se tutti i moduli teoricamente inseriti sul PCB principale siano stati riconosciuti.

Una feature importantissima è quella indicata come “has_FM24W256”: trattasi della memoria FRAM destinata a contenere i dati di configurazione del dispositivo; l’assenza del riconoscimento di tale dispositivo impedisce il salvataggio dei dati di configurazione in FRAM (comunque in questo caso verrà utilizzata la flash per salvare i dati di configurazione).

A conclusione di questa sezione vale la pena ritornare un attimo sulla pagina “OPERATION MODE SETTINGS” per accennare alle funzioni di debug settabili in tale pagina.

Il dispositivo essendo previsto per fare della sperimentazione potrà richiedere eventualmente di avere accesso ad una serie di funzioni di test e debug non necessariamente richieste durante la normale operatività del dispositivo.

La pagina indicata consente di settare una serie di “flags” che ativeranno selettivamente le funzioni di debug presenti per “sezioni” del progetto.

Questo consente di avere accesso tramite le vie di monitoraggio che verranno descritte in seguito, ad una notevole mole di messaggi di debug atti ad individuare eventuali problemi o caratteristiche operative in tempo reale, senza dover necessariamente accedere ad un ambiente di debug specifico aggiuntivo.

Un cenno particolare al flag “act_flag” , ovvero activity flag: è un flag da lasciare sempre attivato.

3.3 Setup sottosistema LoRa (qualsiasi versione)

Selezionando la sezione “**LORA CONFIGURATION**” dalla pagina principale della GUI si accede alla configurazione del sottosistema LoRa.

Come noto sul dispositivo possono essere equipaggiati diversi tipi di moduli radio LoRa: per conoscere il modulo LoRa attuale configurato e riconosciuto è possibile consultare la pagina di “Inventory configuration”: le righe presenti nella pagina di configurazione LoRa dipenderanno dal tipo di modulo equipaggiato/configurato.

LoRa Configuration

LoRa Configuration:

LoraFreq: 433.725

LoraBw: 31.25KHz ▾

LoraSf: 7 ▾

LoraCodingRate: 4/7 ▾

LoraPreambleLen: 15 ▾

LoraSync: 0x34 ▾

LoraPower: +17 dbm ▾

LoraFreqCorr(ppm): 0

LoRa_FreqJitter(ppm)now: -4

SAVE

Nella schermata sono presenti solo un sottinsieme di parametri configurabili su un generico modulo LoRa; i parametri non presenti sono volutamente non riportati in quanto il loro settaggio è tipicamente legato a particolari discorsi di natura HW/SW che richiedono modifiche a livello di codice sorgente del SW.

Molti dei parametri sono intuitivi e il loro settaggio dipende proprio dal tipo di sperimentazione che si intende fare.

In linea di principio affinché due dispositivi possano comunicare tra loro i parametri LoRa dei due dispositivi devono coincidere.

Figura 33 LoRa Configuration

Gli unici parametri che possono differire sono il valore di potenza in trasmissione e il valore di correzione di frequenza.

Quest'ultimo parametro potrà essere tipicamente lasciato a zero per i dispositivi che montano moduli LoRa ad alta precisione di frequenza (ad es. il modello E22_400M30S o altri moduli di seconda generazione dotati di TCXO) mentre andrà sperimentalmente impostato per i moduli LoRa di prima generazione che non hanno una buona precisione di frequenza o per quelli di seconda generazione sprovvisti di TCXO. **Allo scopo si potrà utilizzare come suggerimento il valore di FreqJitter misurato in tempo reale per i pacchetti ricevuti e riportato nell'ultima riga della schermata.**

Come verifica del corretto allineamento di frequenza si potrà osservare su un RX SDR il segnale trasmesso per giudicare il valore di offset in parti-per-milione da impostare.

3.4 Setup sottosistema APRS (qualsiasi versione)

Selezionando la sezione “APRS CONFIGURATION” dalla pagina principale della GUI si accede alla pagina che consente di configurare i parametri per il servizio APRS.

Come ormai è chiaro il dispositivo può operare in due modalità APRS diverse:

- come iGate/repeater
- come tracker

< Aprs Configuration

Location
Coordinates
Latitude: 4038.67N
Longitude: 01424.55E

APRS/IS iGate
APRS_Host: rotate.aprs2.net
APRS_Port: 14580
APRS_Login: I8FUC-10
APRS_Pass:
APRS_Filter: max 2 km
iGate_Beacon: I8FUC-10>APZMDM,WIDE1
iGate_Beacon_Int: 5 min.

APRS Tracker
Tracker_Beacon: I8FUC-8>APZMDM,WIDE1
Tracker_Beacon_Int: 30 secs.

APRS Logger
APRS Logger Host: 192.168.2.150
APRS Logger Port: 44445

SAVE

Figura 34 APRS Configuration

I parametri da configurare per le due modalità dono diversi anche se simili. la figura a seguire copre entrambe le modalità.

3.4.1 Setup sottosistema APRS per modalità iGate
Affinchè la modalità iGate possa svolgersi è necessario innanzitutto che il dispositivo sia in grado di collegarsi ad internet in modo da potersi collegare ad un opportuno server della rete ARPS-IS che dovrà risultare raggiungibile a livello IP.

E' poi necessario che il gateway internet in uso sia configurato in modo da consentire il traffico uscente sulla porta IP utilizzata dal server APRS-IS e che dipenderà dal sever APRS-IS scelto.

Infine è necessario che sia disponibile un account sulla rete APRS-IS con cui consentire l'autenticazione del dispositivo da parte del server APRS-IS scelto; in particolare serviranno una Login identity ed una Password.

Tutti i parametri ora indicati vanno popolati nelle rispettive caselle presenti nella sezione “APRS/IS iGate” della schermata indicata.

Per quanto riguarda la posizione che l'iGate riporterà verso la rete come propria posizione è possibile dedurla automaticamente, se è presente un modulo GPS in grado di acquisire un fix 3D; in caso contrario è possibile inserire manualmente la posizione da riportare .

Per la determinazione del valore di coordinate da riportare bisognerà utilizzare il formato nativo APRS; ovvero:

- latitudine: 2 digit per i gradi + 4 digit con punto dopo i primi due digit per i primi e secondi come previsto da APRS seguiti dalle lettere N/S per nord o sud
- longitudine: 3 digit per i gradi + 4 digit con punto dopo i primi due digit per i primi e secondi come previsto da APRS seguiti dalle lettere E/W per nord o sud

Sempre per la funzionalità iGate è necessario inserire un parametro che consenta di filtrare le posizioni da replicare sulla rete LoRa e provenienti da APRS-IS: l'unica modalità supportata dalla GUI è la modalità “distanza massima dalla posizione del iGate” ovvero la distanza in Km entro cui replicare sul lato LoRa gli spots APRS ricevuti da APRS/IS.

E' poi possibile inserire la stringa da utilizzare come beacon dell'iGate verso la rete LoRa e verso la rete APRS/IS. Tale stringa andrà composta secondo l'esempio di default.

Ad es.:

18FUC-10>APLS01,WIDE1-1:!GPS_LAT/GPS_LON#iGate-LoRa, 31.25/SF7

La stringa “GPS_LAT/GPS_LON” è una chiave per indicare i parametri latitudine e longitudine raccolti dal modulo GPS locale. La parte successiva al carattere “#” riporta in maniera sintetica le condizioni di emissione LoRa e andrebbe settata manualmente ai parametri inseriti nella schermata LoRa.

La primissima parte della stringa è il nominativo con cui far apparire il dispositivo su APRS.

La chiave “APLS01” indica il tipo di SW in uso sul dispositivo (SW SARIMESH).

Il carattere # presente dopo le coordinate nell'esempio di cui sopra può essere sostituito con un altro carattere allo scopo di modificare il simbolo con cui lo spot APRS apparirà sui portali relativi; er una descrizione in merito è possibile consultare il seguente URL: <https://www.iz3mez.it/aprs-server/simboli-aprs/> (grazie a Giovanni IZ0CZW per la segnalazione).

E' infine possibile specificare il tempo da far intercorrere tra due invii del beacon dell'iGate.

3.4.2 Setup sottosistema APRS per modalità Tracker

La modalità tracker è molto più snella come configurazione; gli unici parametri richiesti sono la stringa da inviare come beacon (secondo il formato spiegato per la sezione iGate) e il periodo del beacon in sec.

Si assume che nell'applicazione sia sempre disponibile un modulo GPS da cui vengono dedotti sia la posizione che il tempo reale per il dispositivo.

3.4.3 Setup sottosistema APRS per connessione ad un server di servizio

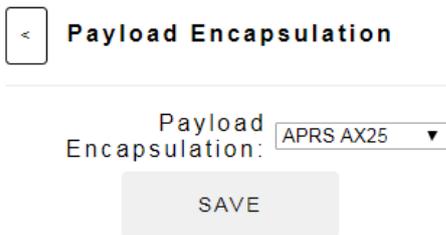
Il dispositivo è in grado di collegarsi ad un server specializzato a raccogliere una serie di dati relativi agli spots APRS ricevuti dalla rete LoRa; tale interfaccia è basata su un semplice protocollo UDP e consente di raccogliere da parte di questo server ad hoc, in genere da locare sulla rete locale o

sulla rete mesh a cui eventualmente il dispositivo, funzionante come iGate, risulta collegato, gli spots LoRa ricevuti in modo da poterli rappresentare su di una mappa geolocalizzata.

La documentazione di questa interfaccia è lasciata a successive attività di documentazione.

3.5 Setup Tipo di incapsulamento (qualsiasi versione)

Nella applicazione APRS i pacchetti trasmessi devono essere “incapsulati” come in una busta per essere riconoscibili come delle entità correttamente trasmesse e ricevute; allo scopo sono possibili diverse modalità.

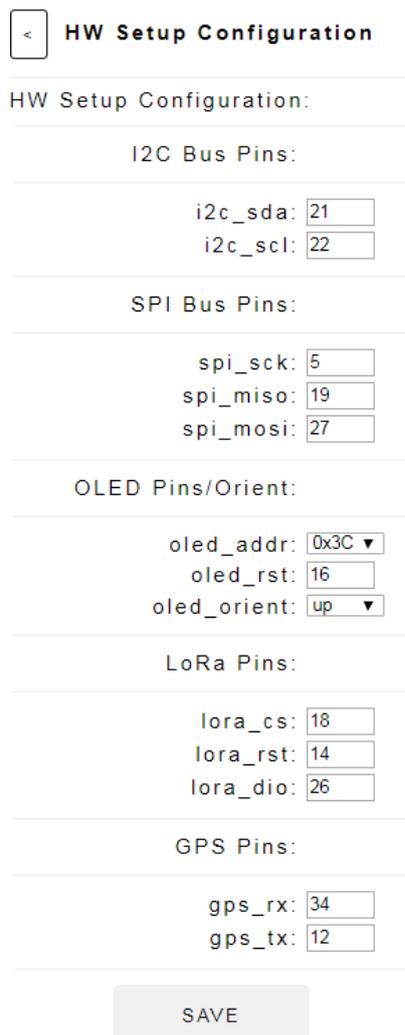


< Payload Encapsulation

Payload Encapsulation: APRS AX25 ▼

SAVE

Figura 35 - Modalità di TX encapsulation



< HW Setup Configuration

HW Setup Configuration:

I2C Bus Pins:

i2c_sda: 21

i2c_scl: 22

SPI Bus Pins:

spi_sck: 5

spi_miso: 19

spi_mosi: 27

OLED Pins/Orient:

oled_addr: 0x3C ▼

oled_rst: 16

oled_orient: up ▼

LoRa Pins:

lora_cs: 18

lora_rst: 14

lora_dio: 26

GPS Pins:

gps_rx: 34

gps_tx: 12

SAVE

Figura 36 HW Setup Configuration

Nella implementazione SARIMESH l’incapsulamento preferito è quello cosiddetto AX.25 che trae la sua origine proprio da tale protocollo e presenta i migliori attributi di compatibilità con altre applicazioni di trasmissione dati in uso (es. modalità TNC KISS).

Esistono altre implementazioni di LoRa APRS in cui si utilizza un diverso tipo di incapsulamento: in particolare viene supportato anche la modalità diffusa nelle implementazioni austriaca e tedesca che indichiamo sinteticamente come modalità “OE_Style.

Il SW in ricezione consente di ricevere pacchetti che utilizzano entrambe le modalità di incapsulamento, realizzando le eventuali azioni necessarie alla compatibilità dei dati trasportati; in trasmissione è necessario invece specificare quale tipo di incapsulamento utilizzare; a questo provvede la schermata a fianco che quindi **consente di impostare il tipo di incapsulamento da usare in trasmissione.**

3.6 HW Setup Configuration

Qualora il SW venga utilizzato su piattaforme HW diverse da quelle SARIMESH, in generale si può rendere necessario settare in maniera appropriata alcune features del SW in modo da adattarsi alle diverse caratteristiche dell’HW; in particolare quello che può differire, nell’ambito di una certa tipologia di schedino HW, è il valore di alcuni pins del processore a cui si collegano le periferiche presenti sullo schedino.

Questa schermata consente di settare i pins del processore ESP32 da utilizzare per le seguenti funzioni:

- Bus I2C
- Bus SPI
- OLED pins, addr and orientation
- LoRa specific pins
- GPS specific pins

I valori da impostare vanno dedotti dalla documentazione HW dei dispositivi usati.

Per l'HW SARIMESH questa pagina riporta i valori utilizzati ma non modificabili in quanto invariati.

3.7 Setup sottosistema IoT (qualsiasi versione)

Da completare

4 Interfaccia di Debug Remoto

I dispositivi della famiglia LoRa Beacon sono dotati di una interfaccia di debug remoto che consente di accedere ad una serie di funzionalità pensate per un uso avanzato dei dispositivi ovvero per accedere a delle funzioni di monitoraggio del funzionamento in tempo reale senza richiedere il collegamento ad un computer via USB o seriale e quindi sfruttabile anche da remoto per dispositivi per es. locati in un punto collegato via rete internet.

L'accesso alla interfaccia di Debug Remoto utilizza il protocollo telnet dotato di un semplice sistema di sicurezza basato su password.

L'accesso a tale funzionalità può avvenire sia utilizzando un classico client Telnet disponibile per tutte le piattaforme di computer esistenti, sia via una Applicazione web sfruttando un qualsiasi browser internet.

Indipendentemente dalla modalità di accesso alla interfaccia di Debug Remoto, sono disponibili una serie di comandi che corrispondono ad altrettante funzioni di monitoraggio o di debug.

Di seguito vengono descritte le due modalità di accesso e successivamente i comandi di debug disponibili.

4.1 Accesso alla IF di Remote Debug tramite Telnet Client

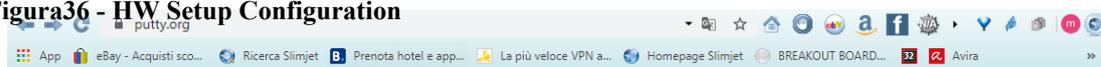
Per utilizzare questa modalità di accesso è sufficiente procurarsi un qualsiasi client Telnet quale ad esempio il classico “putty” scaricabile dal seguente URL: <http://putty.org>

Scaricato il file putty.exe corrispondente alla propria piattaforma PC che si intende utilizzare è sufficiente avviare il file senza necessità di installazione.

Selezionare dalla schermata che viene proposta le seguenti opzioni in aggiunta a quelle di default:

- Terminal: Implicit CR in every LF

Figura36 - HW Setup Configuration



Download PuTTY

PuTTY is an SSH and telnet client, developed originally by Simon Tatham for the Windows platform. PuTTY is open source software that is available with source code and is developed and supported by a group of volunteers.

You can download PuTTY [here](#).

Figura 37 Putty Download

- Session: Other --> Telnet
- Host Name: indirizzo IP in rete LAN del dispositivo

Apparirà una schermata che richiede l'immissione di una password: default è esp32

Immettendo la password apparirà l'interfaccia a comandi della console di debug del dispositivo.

Tale interfaccia è basata sulla modalità “comando” ovvero sono resi disponibili una serie di “comandi” che permettono di accedere alle funzioni disponibili.

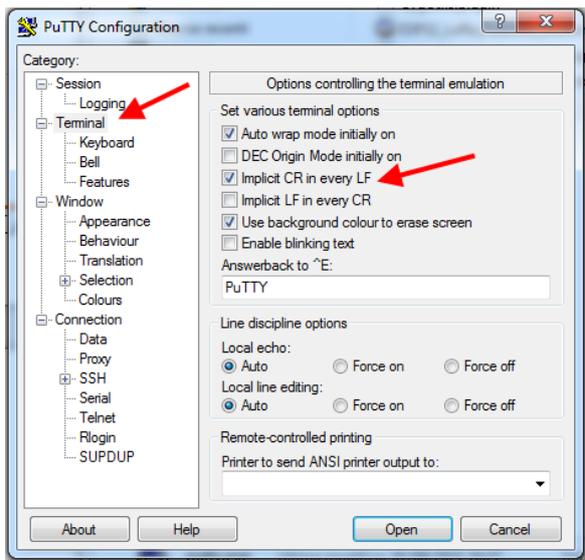


Figura 38 Putty setup - 1

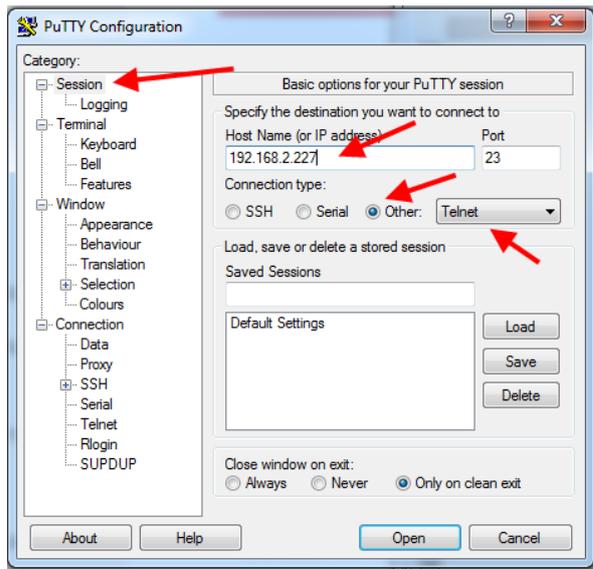


Figura 39 Putty setup - 2

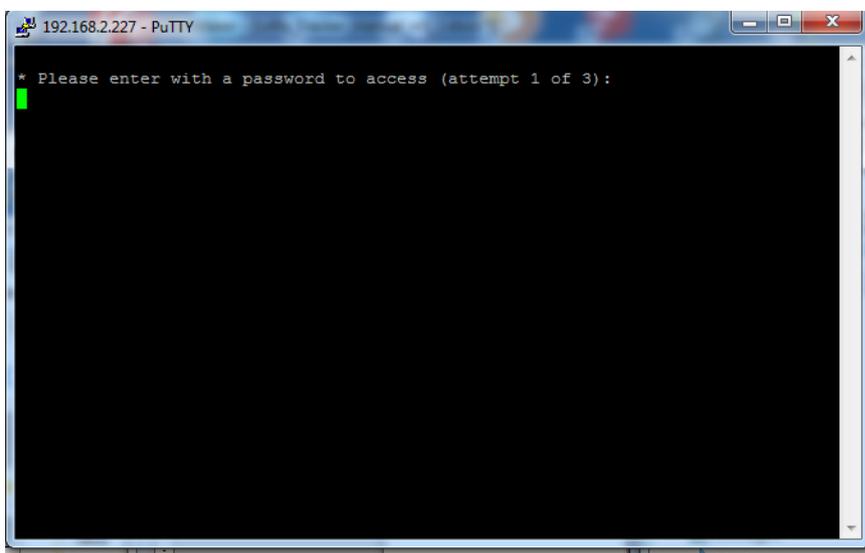


Figura 40 Schermata richiesta password console di debug

Inoltre di default la console riporta una serie di informazioni diagnostiche prodotte dal SW del dispositivo, sulla base delle funzioni di debug abilitate allo stato.

La schermata iniziale, dopo il login, elenca i comandi disponibili ed il loro significato sinteticamente; alcuni comandi sono definiti “Application Commands” e servono ad attivare funzioni specifiche del SW.

Alcuni dei comandi disponibili rappresentano l’esatto equivalente di quanto visualizzabile sulla console seriale via USB accessibile tipicamente dall’interno dell’ambiente arduino.

Altri comandi rappresentano funzioni aggiuntive disponibili esclusivamente tramite questa interfaccia di Remote Debug e vengono descritti di seguito in dettaglio.

4.1.1 Comando “gps_status”

```

192.168.2.227 - PuTTY
* Please enter with a password to access (attempt 1 of 3):
esp32
* Password ok, allowing access now...
*** Remote debug - over telnet - for ESP32 - version 3.0.5
* Host name: ESP32-10521C691A18 IP:192.168.2.227 Mac address:10:52:1C:69:1A:18
* Free Heap RAM: 149844
* ESP SDK version: v3.3.1-61-g367c3c09c
*****
* Commands:
? or help -> display these help of commands
q -> quit (close this connection)
m -> display memory available
v -> set debug level to verbose
d -> set debug level to debug
i -> set debug level to info
w -> set debug level to warning
e -> set debug level to errors
s -> set debug silence on/off
l -> show debug level
t -> show time (millis)
profiler:
p -> show time between actual and last message (in millis)
p min -> show only if time is this minimal
P time -> set debug level to profiler
c -> show colors
filter:
filter <string> -> show only debugs with this
nofilter -> disable the filter

* Application commands:

reboot
gps_status
temperature
wifi_scan
display_config
i2c_scan
selftest_start
selftest_stop
show_stats
show_events
log_display
fram_dump
fram_log_set
fram_log_reset

* Please type the command and press enter to execute.(? or h for this help)
***

```

Figura 41 Schermata iniziale console di Debug Remoto

Il comando “gps_status” consente di avere una serie di dettagliate informazioni sullo stato del GPS: in particolare fornisce i seguenti parametri:

- Elenco dei satelliti attualmente utilizzati per il fix: numero di satelliti utilizzati, valori di PRN, elevazione, di azimut e di rapporto SNR per ogni satellite utilizzato per il fix
- parametri relativi al fix : latitudine, longitudine, età del fix, data, tempo, altezza, velocità, parametri di qualità del fix

```

gps_status
(D) (gps_status_header) (C1)
(D) (gps_status_header) (C1)
(D) (gps_status_header) (C1)
(D) (gps_status) (C1) Sats=12 Nums=13 14 20 30 Elevation=18 19 57 77 Azimuth=281 167 291 227 SNR=31 0 30 18
(D) (loop_extended) (C1) GPS status -> 12 1.2 40.644827 14.409652 273 08/01/2021 19:39:37 385 79.80 23.41 0.26 NNE 1642 322.20 NW 62376 179 4

```

GPS Satellites used
 GPS Fix data (according to gps_status_header)

Figura 42 Uscita el comando “gps_status”

4.1.2 Comando “temperature”

Fornisce una stima del valore della temperatura in °C a livello del chip ESP32 interno al dispositivo; questo valore va utilizzato come una indicazione di massima del valore di temperatura essendo la sua determinazione basata su di una feature non documentata del chip ESP32.

4.1.3 Comando “wifi_scan”

Effetua una scansione della banda WiFi dei 2.4Ghz e riporta la lista delle reti trovate e dei relativi livelli di segnale e presenza o meno di crittografia (*).

```
wifi_scan
(do_wifi_scan)(C1) wifi_scan start...scan done: 11 networks found
1: RFC2019-24 (-55)*
2: MikroTik-66BABF-24 (-59)*
3: SARIMESH (-59)*
4: ESP32-240AC4595AA8 (-73)
5: TIM-24138604 (-83)*
6: casa Izzo (-89)*
7: Vodafone-35188680 (-90)*
8: Vodafone-WiFi (-91)
9: FASTWEB-SK5JE6 (-91)*
10: FASTWEB-VJX4Y7 (-92)*
11: WOW FI - FASTWEB (-93)*
```

Figura 43 Output comando “wifi_scan”

a fianco riporta un piccolo sottosieme.

```
display_config
(Display_Config)(C1) Display active Configuration data
ESP_config.DeviceName = LoRa Tracker Vr4.1
ESP_config.DeviceId = Not Available
ESP_config.cpu_type = ESP32 Dev V4
ESP_config.dhcp = 1
ESP_config.daylight = 1
ESP_config.Update_Time = 0
ESP_config.timezone = 10
ESP_config.IP = 192.168.2.60
ESP_config.Netmask = 255.255.255.0
ESP_config.GatewayIP = 192.168.2.1
ESP_config.DnsIP = 8.8.8.8
ESP_config.ssid = RFC2019-24
ESP_config.password = ██████████
ESP_config.ntpServer = ntp1.inrim.it
ESP_config.gps_debug = 0
ESP_config.LoRa_debug = 0
ESP_config.RTC_debug = 0
ESP_config.ezTime_debug = 0
ESP_config.pps_debug = 0
ESP_config.PE_debug = 0
ESP_config.BT_KISS_Mode = 0
ESP_config.Serial_KISS_Mode = 0
ESP_config.Tracker_Mode = 1
```

Figura 44 Output del comando “display_config”

4.1.5 Comando “show_stats”

Riporta una lista di parametri sintetici relativi al funzionamento del dispositivo dettagliatamente per la parte LoRa.

In particolare riporta la versione SW in uso sul dispositivo, il numero di pacchetti LoRa ricevuti e trasmessi, il numero di pacchetti LoRa persi (ovvero soppressi in trasmissione) a causa della congestione del canale radio, il numero di pacchetti LoRa ricevuti con valore di CRC a livello radio

4.1.4 Comando “display_config”

Effetua un display in forma testuale della attuale configurazione del dispositivo in uso.

La configurazione è costituita da una lista di linee del tipo “chiave = valore” di cui l’immagine

Il comando consente quindi di esplorare in dettaglio tutti i parametri di configurazione in uso anche senza disporre di una interfaccia grafica.

La nomenclatura usata per le chiavi dovrebbe rendere agevole individuare il significato dei vari parametri. elencati

errato (in genere a causa di errori di trasmissione sul canale radio o di conflitto di accesso al canale radio), il tempo di attesa per la trasmissione dei pacchetti LoRa, la temperatura della CPU ESP32 ed il valore di tempo “uptime” della CPU dall’ultimo reboot.

```
show_stats
(D) (show_stats)(C1) ===== Stats start =====
(D) (show_stats)(C1) SW_version = 1.0.5_20210730_1611
(D)
(D) (show_stats)(C1) LoRa_rx_packets = 200
(D) (show_stats)(C1) LoRa_tx_packets = 66
(D) (show_stats)(C1) LoRa_lost_packets = 0
(D) (show_stats)(C1) LoRa_CRC_errored_packets = 0
(D)
(D) (show_stats)(C1) LoRa_OnAirTime(msec) = 602
(D) (show_stats)(C1) CPU_Temperature(C°) = 57.42
(D) (show_stats)(C1) Processor_UpTime(secs) = 2083
(D) (show_stats)(C1) ===== Stats end =====
```

Figura 45 Output comando “show_stats”

Il dispositivo, nella versione basata su HW LoRa Beacon , presenta on board una memoria FRAM utilizzata per la tenuta oltre che dei dati di configurazione del dispositivo, anche di una serie di dati raccolti in tempo reale dal dispositivo e relativi sia ad eventi particolari che alla ricezione di spots a livello del sottosistema radio LoRa.

Il comando “show_events” elenca la sola componente di “eventi” presenti nel log tenuto nella memoria FRAM; gli eventi riportati sono tipicamente legati ad azioni di management che implicano cambi di configurazione e ripartenze del processore.

Gli eventi sono marcati con il tempo reale nel quale sono stati registrati e quindi sopravvivono ai restart del dispositivo, per cui rappresentano uno strumento di “post mortem data analysys” per eventualmente diagnosticare situazioni anomale di funzionamento.

```
show_events
(D) (show_events)(C1) FRAM Log parameters: log_head=340 log_len=400 log_size=400 fram_base_log=2048
(D) (show_events)(C1) ===== Events Log Start =====
(D) ==> 20210801 09:33:41.000 cptr=308 [pnt=4352]====> [1627810421|EVENT|===== system reboot completed =====]
(D) ==> 20210801 08:34:28.000 cptr=303 [pnt=4712]====> [1627806868|EVENT|===== system reboot by GUI =====]
(D) ==> 20210801 21:38:01.000 cptr=297 [pnt=5144]====> [1627853881|EVENT|===== system reboot completed =====]
(D) (show_events)(C1) ===== Events Log End =====
```

Figura 46 Output comando “show_events”

4.1.7 Comando “log_display”

Il dispositivo, nella versione basata su HW LoRa Beacon , presenta on board una memoria FRAM utilizzata per la tenuta oltre che dei dati di configurazione del dispositivo, anche di una serie di dati raccolti in tempo reale dal dispositivo e relativi sia ad eventi particolari che alla ricezione di spots a livello del sottosistema radio LoRa.

Il comando “log_display” riporta la lista completa della “coda circolare” in cui sono riportati tutti gli eventi e gli spots registrati in tempo reale nel corso del funzionamento del dispositivo.

L’utilizzo di una “coda circolare” per conservare questi dati consente di tenere traccia degli ultimi 400 eventi o spots registrati, scartando per ogni nuovo evento da registrare il più vecchio di quelli già registrati.

4.1.6 Comando “show_events”

```
(D) (log_display)(C1) FRAM Log parameters: log_head=395 log_len=400 log_size=400 fram_base_log=2048
(D) (log_display)(C1) ===== Log Start =====
(D) ==> 19700101 00:15:10.093  cntr=399 [pntr=30560]====> [910|40.6435013|14.4095001|433.725|-88.00|10.75|0.00] d=3341105
(D) ==> 19700101 00:15:29.093  cntr=398 [pntr=30632]====> [929|40.6435013|14.4095001|433.725|-102.00|4.75|0.00] d=3341105
(D) ==> 19700101 00:15:30.093  cntr=397 [pntr=30704]====> [930|40.6435013|14.4095001|433.725|-41.00|11.50|0.00] d=3341105
(D) ==> 19700101 00:15:33.093  cntr=396 [pntr=30776]====> [923|40.6435013|14.4095001|433.725|-101.00|13.25|0.00] d=3341105
(D) ==> 19700101 00:15:41.093  cntr=395 [pntr=2048]====> [941|40.6445007|14.4090004|433.725|-96.00|8.00|0.00] d=3882434
(D) ==> 19700101 00:15:42.093  cntr=394 [pntr=2120]====> [942|40.6445007|14.4090004|433.725|-32.00|12.00|0.00] d=3882434
(D) ==> 19700101 00:15:43.093  cntr=393 [pntr=2192]====> [943|40.6445007|14.4090004|433.725|-96.00|8.50|0.00] d=3882434
(D) ==> 19700101 00:16:06.093  cntr=392 [pntr=2264]====> [966|40.6445007|14.4090004|433.725|-86.00|11.25|0.00] d=3882434
(D) ==> 19700101 00:16:11.093  cntr=391 [pntr=2336]====> [971|40.6446686|14.4095001|433.725|-39.00|10.50|0.00] d=8415512
(D) ==> 19700101 00:16:12.093  cntr=390 [pntr=2408]====> [972|40.6446686|14.4095001|433.725|-79.00|12.00|0.00] d=8415512
(D) ==> 19700101 00:16:13.093  cntr=389 [pntr=2480]====> [973|40.6446686|14.4095001|433.725|-42.00|12.00|0.00] d=8415512
(D) ==> 19700101 00:00:30.093  cntr=388 [pntr=2552]====> [30|EVENT]===== system reboot completed ===== d=4749550
(D) ==> 19700101 00:00:39.093  cntr=387 [pntr=2624]====> [39|0.0000000|0.0000000|433.725|-49.00|11.75|0.00] d=4749550
(D) ==> 19700101 00:00:39.093  cntr=386 [pntr=2696]====> [39|0.0000000|0.0000000|433.725|-86.00|11.75|0.00] d=4749550
(D) ==> 19700101 00:00:44.093  cntr=385 [pntr=2768]====> [44|0.0000000|0.0000000|433.725|-88.00|11.25|0.00] d=4749550
(D) ==> 19700101 00:01:10.093  cntr=384 [pntr=2840]====> [70|40.6448326|14.4096670|433.725|-49.00|11.25|0.00] d=5323454
(D) ==> 19700101 00:01:10.093  cntr=383 [pntr=2912]====> [70|40.6448326|14.4096670|433.725|-87.00|11.75|0.00] d=5323454
(D) ==> 19700101 00:01:41.093  cntr=382 [pntr=2984]====> [101|40.6448326|14.4096670|433.725|-47.00|11.25|0.00] d=10600898
(D) ==> 19700101 00:01:41.093  cntr=381 [pntr=3056]====> [101|40.6448326|14.4096670|433.725|-88.00|11.25|0.00] d=10600898
(D) ==> 19700101 00:01:43.093  cntr=380 [pntr=3128]====> [103|40.6448326|14.4096670|433.725|-49.00|12.00|0.00] d=10600898
```

Figura 47 Output comando “log_display”: dati più vecchi

```
(D) ==> 20210801 20:34:19.003  cntr=16 [pntr=29336]====> [1627850059|40.6448326|14.4096670|433.725|-89.00|11.25|0.00] d=23
(D) ==> 20210801 20:34:49.003  cntr=15 [pntr=29408]====> [1627850089|40.6448326|14.4096670|433.725|-48.00|11.25|0.00] d=23
(D) ==> 20210801 20:34:50.003  cntr=14 [pntr=29480]====> [1627850090|40.6448326|14.4096670|433.725|-88.00|11.25|0.00] d=23
(D) ==> 20210801 20:35:20.003  cntr=13 [pntr=29552]====> [1627850120|40.6448326|14.4095001|433.725|-48.00|11.25|0.00] d=18
(D) ==> 20210801 20:35:21.003  cntr=12 [pntr=29624]====> [1627850121|40.6448326|14.4095001|433.725|-87.00|11.00|0.00] d=18
(D) ==> 20210801 20:35:52.003  cntr=11 [pntr=29696]====> [1627850152|40.6448326|14.4096670|433.725|-48.00|11.00|0.00] d=23
(D) ==> 20210801 20:35:52.003  cntr=10 [pntr=29768]====> [1627850152|40.6448326|14.4096670|433.725|-87.00|11.25|0.00] d=23
(D) ==> 20210801 20:36:23.003  cntr=9 [pntr=29840]====> [1627850183|40.6448326|14.4096670|433.725|-48.00|11.25|0.00] d=23
(D) ==> 20210801 20:36:23.003  cntr=8 [pntr=29912]====> [1627850183|40.6448326|14.4096670|433.725|-88.00|11.25|0.00] d=23
(D) ==> 20210801 20:36:54.003  cntr=7 [pntr=29984]====> [1627850214|40.6448326|14.4096670|433.725|-50.00|11.25|0.00] d=23
(D) ==> 20210801 20:36:54.003  cntr=6 [pntr=30056]====> [1627850214|40.6448326|14.4096670|433.725|-90.00|10.75|0.00] d=23
(D) ==> 20210801 20:37:21.003  cntr=5 [pntr=30128]====> [1627850241|40.6448326|14.4096670|433.725|-48.00|11.25|0.00] d=23
(D) ==> 20210801 20:37:22.003  cntr=4 [pntr=30200]====> [1627850242|40.6448326|14.4096670|433.725|-88.00|10.75|0.00] d=23
(D) ==> 20210801 20:37:25.003  cntr=3 [pntr=30272]====> [1627850245|40.6448326|14.4096670|433.725|-48.00|11.50|0.00] d=23
(D) ==> 20210801 20:37:26.003  cntr=2 [pntr=30344]====> [1627850246|40.6448326|14.4096670|433.725|-87.00|11.00|0.00] d=23
(D) ==> 20210801 20:37:41.003  cntr=1 [pntr=30416]====> [1627850261|40.6448326|14.4096670|433.725|-87.00|11.25|0.00] d=23
(D) (log_display)(C1) ===== Log End =====
```

Figura 48 Output comando “log_display”: dati più recenti

Il formato degli “spots” registrati è diverso per il caso di utilizzo come iGate o come Tracker:

- nel primo caso (uso come iGate) ogni spot riporta il nominativo della stazione sorgente come contenuto nello spot ricevuto via radio dall’iGate, la posizione contenuta nello spot ricevuto ed il livello di segnale e SNR con cui lo spot è stato ricevuto
- nel secondo caso (uso come tracker) lo spot riporta la posizione fisica in cui si trovava il tracker al momento della ricezione di uno spot, la frequenza cui cui è stato ricevuto lo spot ed i valori di livello del segnale e del SNR con cui lo spot è stato ricevuto

La figura precedente riporta un esempio di log per un iGate: il motivo del diverso contenuto informativo deriva dalla impossibilità nel caso tracker di identificare la sorgente del messaggio LoRa.

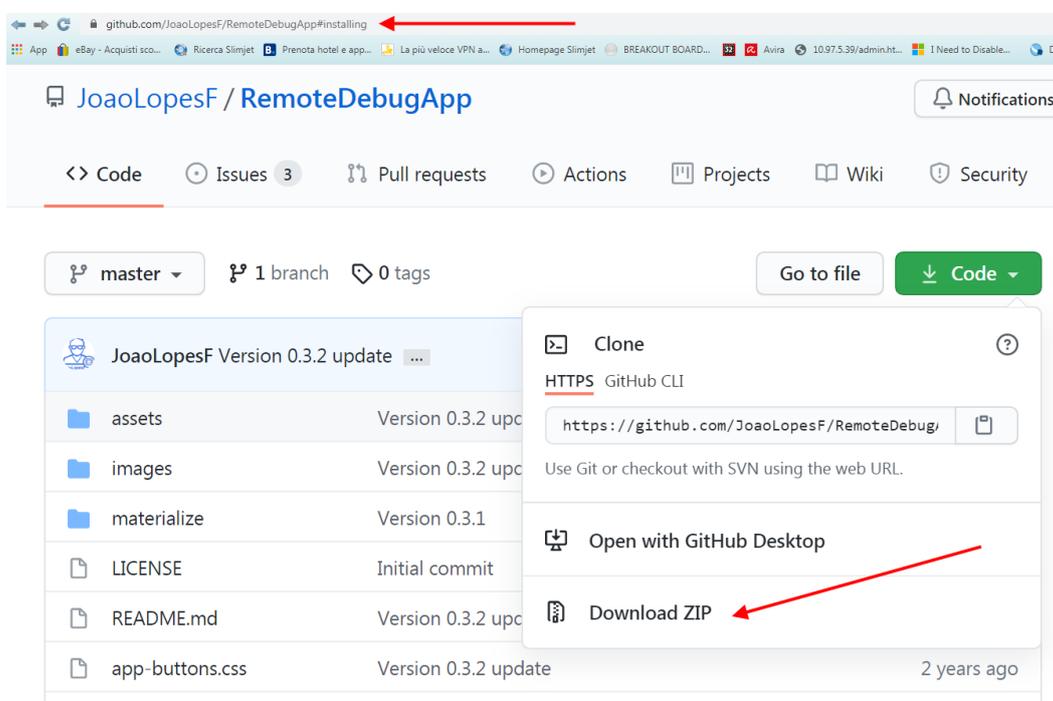
In ogni spot è anche contenuto un campo d (distanza) il cui significato è lasciato ad una futura fase di documentazione; allo stato è da considerare un campo non significativo.

```
(D) ==> 20210731 16:29:30.690  cntr=52 [pntr=9080]====> [1627750770|41.2881660|13.2609997|120CZW-9|-118.00|0.00] d=121756
(D) ==> 20210731 17:00:01.690  cntr=31 [pntr=9752]====> [1627750801|41.2905006|13.2644997|120CZW-9|-114.50|-6.50|0.00] d=121684
(D) ==> 20210731 17:01:03.690  cntr=30 [pntr=9824]====> [1627750863|40.6445007|14.4091663|18FUC-10|-62.00|11.50|0.00] d=3682
(D) ==> 20210731 17:01:32.690  cntr=29 [pntr=9896]====> [1627750892|40.6119995|14.4008331|IQ8SO-10|-62.00|11.75|0.00] d=0
(D) ==> 20210731 17:03:03.690  cntr=28 [pntr=9968]====> [1627750983|40.6445007|14.4091663|18FUC-10|-62.00|11.75|0.00] d=3682
(D) ==> 20210731 17:03:08.690  cntr=27 [pntr=10040]====> [1627750988|41.3068352|13.2873335|120CZW-9|-110.00|-4.00|0.00] d=121325
(D) ==> 20210731 17:05:12.690  cntr=26 [pntr=10112]====> [1627751112|41.3193321|13.3013334|120CZW-9|-98.00|3.25|0.00] d=121311
(D) ==> 20210731 17:05:43.690  cntr=25 [pntr=10184]====> [1627751143|41.3224983|13.3056669|120CZW-9|-101.00|1.50|0.00] d=121262
(D) ==> 20210731 17:06:14.690  cntr=24 [pntr=10256]====> [1627751174|41.3268318|13.3078337|120CZW-9|-105.75|-1.75|0.00] d=121437
(D) ==> 20210731 17:06:34.690  cntr=23 [pntr=10328]====> [1627751194|40.6119995|14.4008331|IQ8SO-10|-62.00|11.75|0.00] d=0
(D) ==> 20210731 17:07:05.690  cntr=22 [pntr=10400]====> [1627751225|40.6445007|14.4091663|18FUC-10|-62.00|11.75|0.00] d=3682
(D) ==> 20210731 17:09:06.690  cntr=21 [pntr=10472]====> [1627751346|40.6445007|14.4091663|18FUC-10|-62.00|11.75|0.00] d=3682
(D) ==> 20210731 17:11:07.690  cntr=20 [pntr=10544]====> [1627751467|40.6445007|14.4091663|18FUC-10|-62.00|11.50|0.00] d=3682
(D) ==> 20210731 17:11:35.690  cntr=19 [pntr=10616]====> [1627751495|40.6119995|14.4008331|IQ8SO-10|-64.00|11.00|0.00] d=0
(D) ==> 20210731 17:13:09.690  cntr=18 [pntr=10688]====> [1627751589|40.6445007|14.4091663|18FUC-10|-62.00|11.50|0.00] d=3682
(D) ==> 20210731 17:15:10.690  cntr=17 [pntr=10760]====> [1627751710|40.6445007|14.4091663|18FUC-10|-62.00|11.50|0.00] d=3682
(D) ==> 20210731 17:16:36.690  cntr=16 [pntr=10832]====> [1627751796|40.6119995|14.4008331|IQ8SO-10|-62.00|12.00|0.00] d=0
(D) ==> 20210731 17:19:12.690  cntr=15 [pntr=10904]====> [1627751952|40.6445007|14.4091663|18FUC-10|-62.00|11.75|0.00] d=3682
(D) ==> 20210731 17:21:13.690  cntr=14 [pntr=10976]====> [1627752073|40.6445007|14.4091663|18FUC-10|-62.00|12.00|0.00] d=3682
(D) ==> 20210731 17:21:17.690  cntr=13 [pntr=11048]====> [1627752077|41.3328323|13.3161669|120CZW-9|-121.50|-9.50|0.00] d=121345
```

Figura 49 Esempio record di log relativi a spot ricevuti e riportati da un tracker

4.2 Accesso alla IF di Remote Debug tramite Web App

Per utilizzare questa modalità di accesso è necessario installare sul proprio PC un particolare applicativo di nome Remote Debug WEB App scaricabile dal seguente URL:
<https://github.com/JoaoLopesF/RemoteDebugApp#installing>



La figura sopra riporta una immagine della pagina di download; per scaricare la App è necessario selezionare la casella “Code” e dalla successiva finestra di selezione che viene presentata scegliere l’opzione “Download”.

Figura 50 Pagina di download Remote Debug App

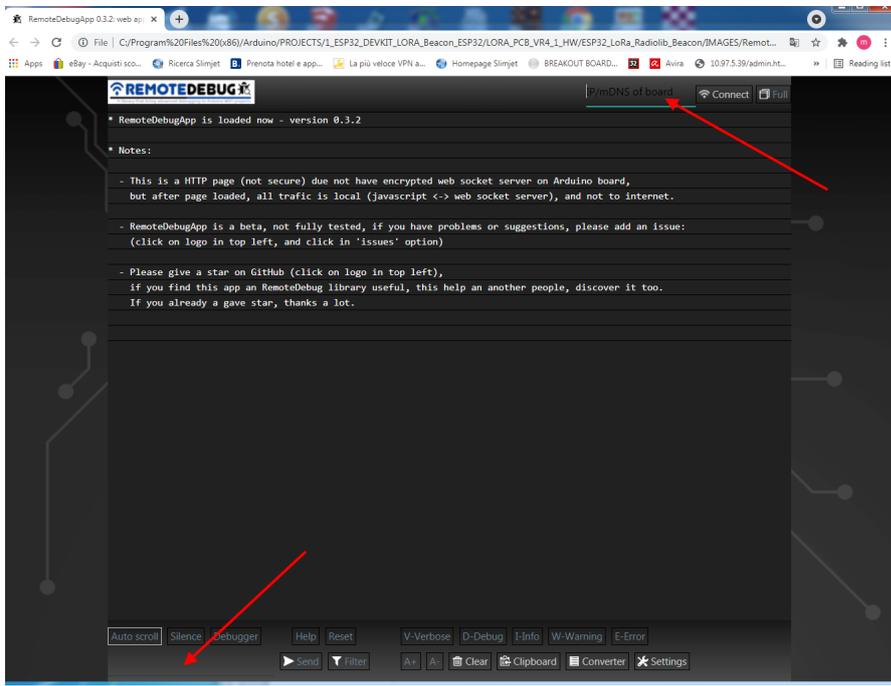
L’archivio .zip così ottenuto va espanso sul PC che si sta utilizzando in una directory a piacere; quindi con il tool locale “esplora risorse” selezionare all’interno della directory in cui è stato espanso l’archivio il file index.html ed aprirlo con un browser tipo chrome o firefox.

La figura successiva riporta l’apparenza della finestra del browser chrome che si ottiene.

Come si potrà notare è una classica interfaccia web che richiede inizialmente di introdurre l’indirizzo IP del dispositivo da monitorare.

E' poi necessario inserire nella finestra di input la password di accesso (di default esp32).

A valle dell'introduzione della password ci si troverà una finestra del tutto simile a quella che si sarebbe ottenuta utilizzando l'applicazione putty di cui al paragrafo precedente.



Valgono a questo punto esattamente tutte le funzionalità e considerazioni svolte al paragrafo precedente.

Utilizzando questo tipo di interfaccia sono disponibili alcune funzioni aggiuntive tra cui forse la più interessante è quella di poter introdurre un "Filtro" sul contenuto delle informazioni presentate nella finestra di lavoro in modo per es. da displayare solo dei particolari messaggi; questa funzione quindi si rende molto utile in caso di attivazione delle

Figura 51 Schermata iniziale Remote Debug App

funzionalità di debug presenti sul dispositivo.

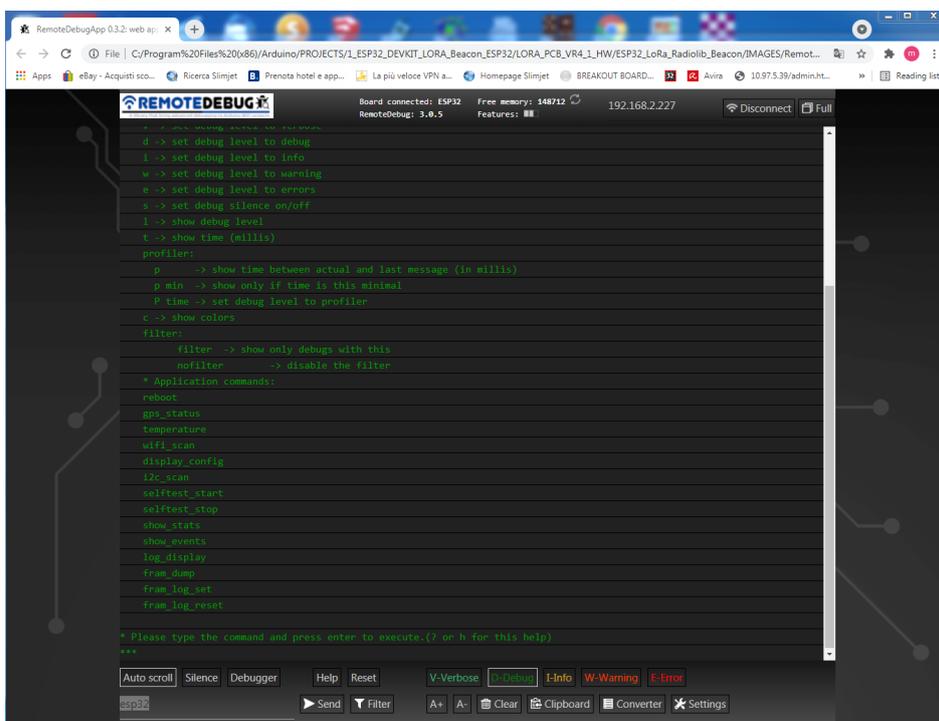


Figura 52 schermata iniziale console di Remote Debug App

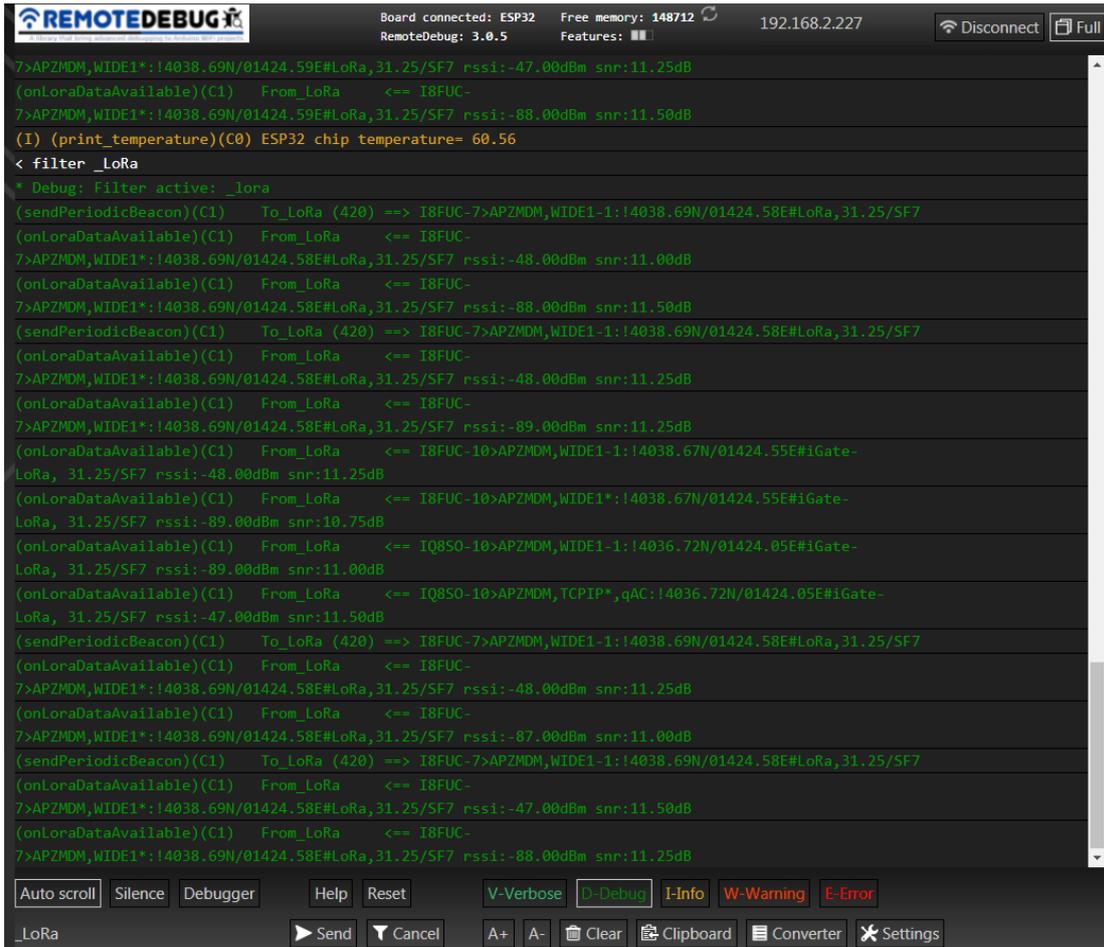


Figura 53 Remote Debug App Filter function

5 Porting del SW LoRa Beacon su altre piattaforme HW

Il SW del progetto LoRa Beacon è stato pensato per poter essere utilizzato non solo sulla piattaforma HW LoRa Beacon descritta precedentemente, ma anche su dispositivi HW simili, che cioè utilizzino lo stesso tipo di microcontrollore (ESP32) e che comprendano un insieme di funzioni HW simili, quali ad es. un modulo GPS e/o un modulo LoRa.

Allo scopo di poter utilizzare le funzionalità SW presenti ovviamente è necessario che l'HW sia in grado di fornire le funzionalità HW sottese da tali funzionalità SW.

A seconda dello “schedino” che si vuole utilizzare è quindi necessario verificare questo aspetto e settare opportunamente alcuni parametri a livello della configurazione di “build” del SW in modo da attivare o disattivare dei pezzi di SW relativi.

Allo stato sono stati effettuati i porting del SW su due schedini particolari che rappresentano degli esempi significativi di dispositivi “all-in-one” reperibili sui soliti portali cinesi.

L'operazione di “porting” consiste nelle seguenti operazioni:

- configurare opportunamente l'ambiente Arduino di sviluppo del SW selezionando opportunamente la piattaforma HW da supportare
- analizzare la struttura HW del dispositivo e settare opportunamente il file “master_config.ino” di configurazione del SW
- effettuare una “build” del SW collegandosi direttamente al target tramite USB
- effettuare i necessari test di “non regressione” per verificare il corretto funzionamento dell'insieme HW-SW, per le sole funzioni supportabili dall'HW in uso
- verificare che funzionalmente il dispositivo si comporti come atteso.

Una volta completato con successo il lavoro di porting è possibile generare una immagine completa del SW caricabile sullo schedino tramite il classico tool di download del microcontrollore ESP32, senza quindi richiedere necessariamente la presenza dell'ambiente di sviluppo SW Arduino per coloro che intendono utilizzare semplicemente il SW senza necessariamente apportare modifiche allo stesso, ma sfruttando l'interfaccia di configurazione fornita per l'impostazione del dispositivo e per effettuare eventuali sperimentazioni con lo stesso.

Nei paragrafi seguenti viene riportata la documentazione delle impostazioni da utilizzare per il tool di download in relazione ai diversi tipi di schedini da supportare.

5.1 Installazione SW su dispositivo TTGO T-beam-V1-2019

Questo schedino è la prima versione di una lunga serie di dispositivi resi disponibili dallo stesso Vendor e che pur condividendo lo stesso nome commerciale T-Beam hanno mostrato un equipaggiamento HW diverso anche parecchio dalla versione originale. Facendo una ricerca su internet è possibile approfondire questo argomento (<https://github.com/Xinyuan-LilyGo/TTGO-LoRa-Series>)

La figura seguente riporta i dettagli del dispositivo utilizzato per il test.

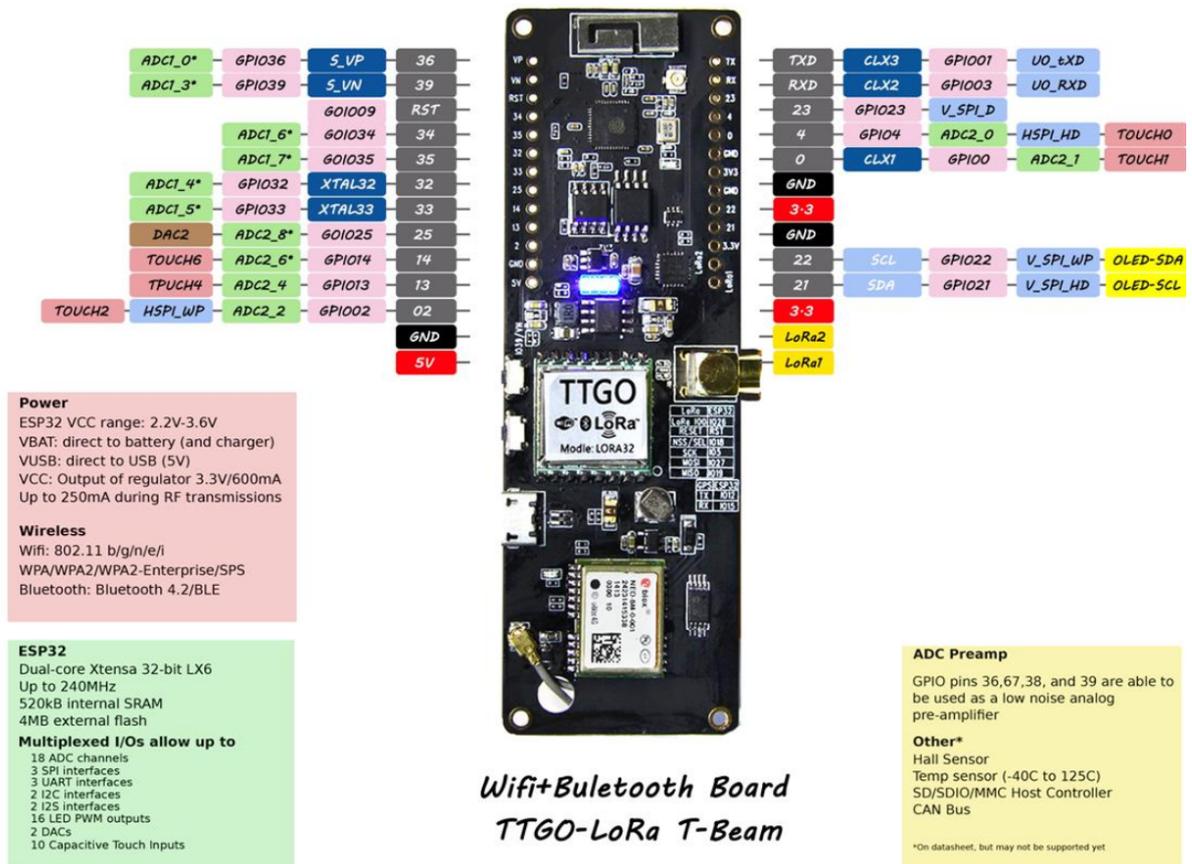


Figura 54 Dispositivo TTGO T-Beam V1 utilizzato per il test

La procedura seguente fa riferimento ovviamente allo specifico dispositivo indicato per ovvi motivi di disponibilità dell’HW su cui effettuare il test.

Questo schedino non è in grado di supportare la memoria FRAM presente nel progetto LoRa Beacon, per cui non sono supportate tutte le funzioni SW che richiedono tale componente, con esclusione della sola parte di configurazione del dispositivo.

Per conservare la configurazione del dispositivo viene sfruttata, al posto delle FRAM, una porzione della memoria Flash; dal punto di vista funzionale ci sono solo minime differenze rispetto al caso di utilizzo della FRAM.

Come chipset LoRa questo schedino supporta solo (nella versione testata) chips di prima

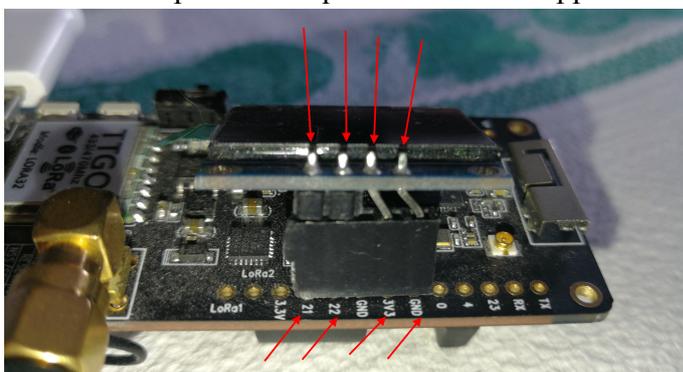


Figura 55 Collegamento Display I2C

generazione LoRa quindi con potenza max di uscita di circa 100miliWatt, con sensibilità della parte ricevente radio non migliorata e senza supporto di TCXO per la stabilizzazione della frequenza emessa.

Come display l’unico display supportato è quello monocromatico da 0.96” con interfaccia I2C.

La figura a fianco illustra i collegamenti del display allo schedino.

Non sono supportate le funzionalità dei leds.

A livello di funzionalità di debug non sono supportate tutte le funzioni legate alla FRAM, ad es. la tenuta dei log e quindi degli eventi di riconfigurazione e reboot e la registrazione degli spots.

Per l'installazione del SW tramite il download tool presentato nei capitoli precedenti si applicano esattamente le stesse considerazioni con l'unica differenza che è necessario impostare differentemente l'interfaccia seriale su cui si presenta il dispositivo una volta collegato al PC, e gli indirizzi a cui caricare i diversi moduli che costituiscono l'immagine SW.

La figura riporta le impostazioni da utilizzare:

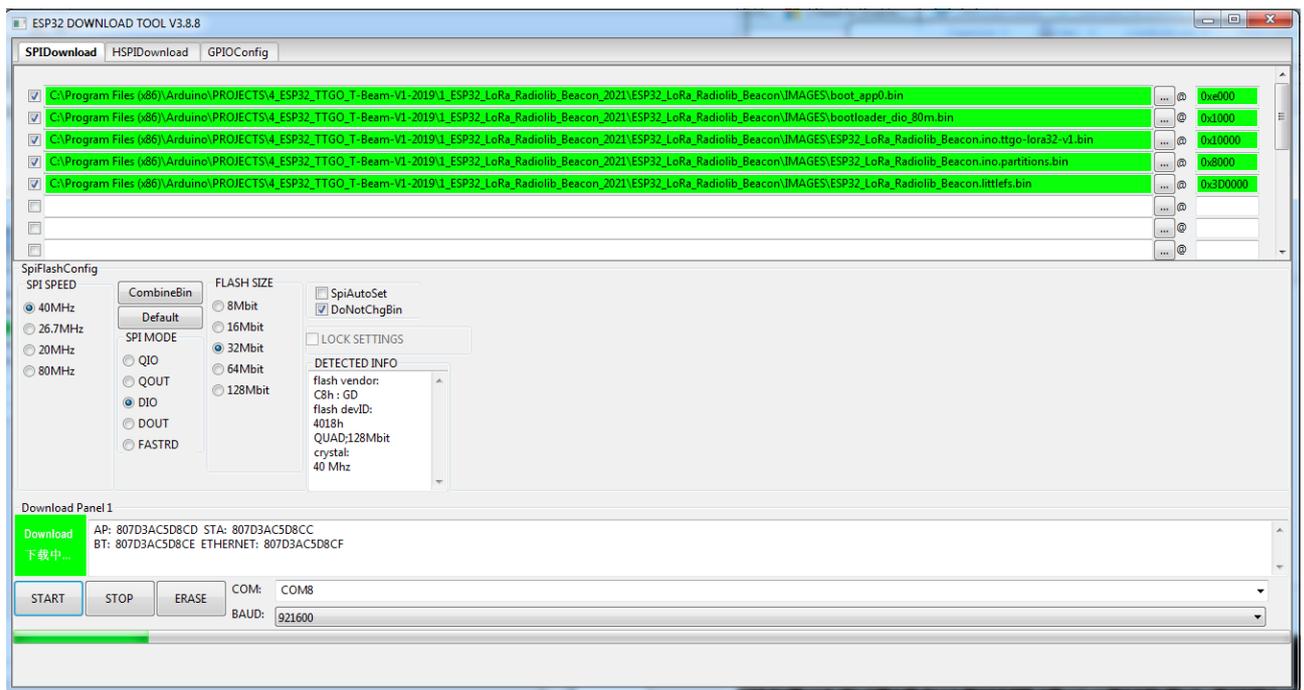


Figura 56 Settaggi da usare per il download del SW su TTGO T-Beam V1

L'immagine SW è distribuita sotto forma di archivio .zip che contiene i 5 moduli SW da caricare sul dispositivo. Tali moduli vanno caricati seguendo l'esempio della figura precedente.

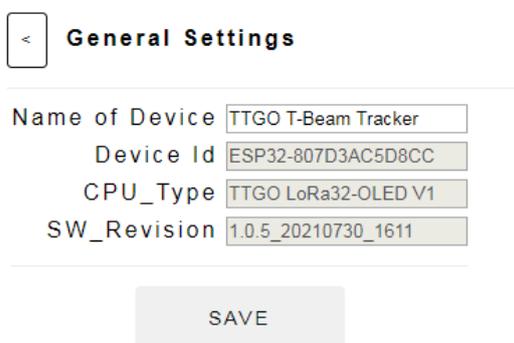


Figura 57 General settings TTGO T-Beam

La figura seguente riporta la schermata delle caratteristiche generali di questa versione come appaiono collegandosi alla interfaccia grafica del dispositivo a valle dell'installazione del SW LoRa Beacon.

La figura a seguire è un esempio del contenuto del display.

Si noti la presenza del campo “ERR” che riporta l'errore di allineamento di frequenza tra RT/TX dovuto alla mancanza di TCXO su questo tipo di HW; tale offset può essere corretto tramite il campo “ppm” presente nella schermata di setup della sezione LoRa.



Figura 58 Esempio Display TTGO T-Beam

Questa procedura di caricamento è richiesta esclusivamente per il primo caricamento del SW sullo schedino; per i successivi aggiornamenti SW è possibile utilizzare in alternativa il caricamento tramite OTA , quindi via radio, senza richiedere il collegamento USB con il dispositivo fisico.

5.2 Installazione SW su dispositivo Heltec_wifi_lora32

Questo schedino è molto simile al dispositivo di cui al paragrafo precedente, ma manca del modulino GPS; un possibile utilizzo è come iGate non essendo necessario in questa applicazione disporre della funzionalità di GPS.

Per il resto condivide quasi tutte le limitazioni di cui al paragrafo precedente.

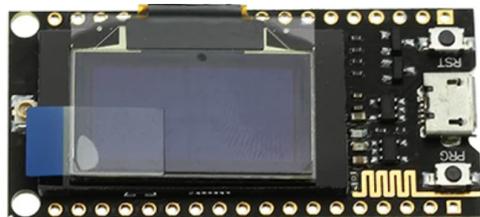


Figura 59 Heltec WiFi LoRa 32

La figura a seguire ritrae lo schedino in questione; esiste uno schedino molto simile reperibile con il nome commerciale TTGO WiFi Lora 32 che è praticamente identico a quello mostrato

La figura successiva rappresenta il pinout del dispositivo utilizzato per il test.

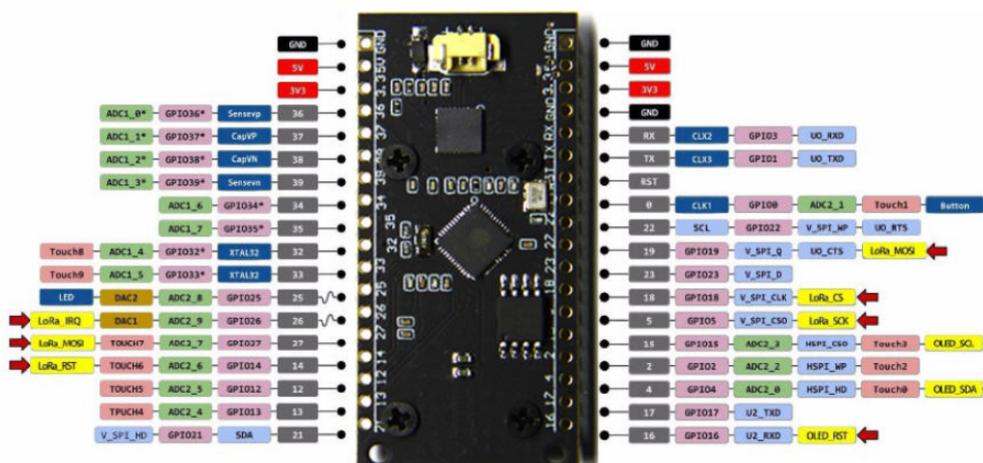


Figura 60 Pin Layout Heltec WiFi LoRa 32

La figura seguente riporta le impostazioni da utilizzare per il tool di caricamento del Sw:

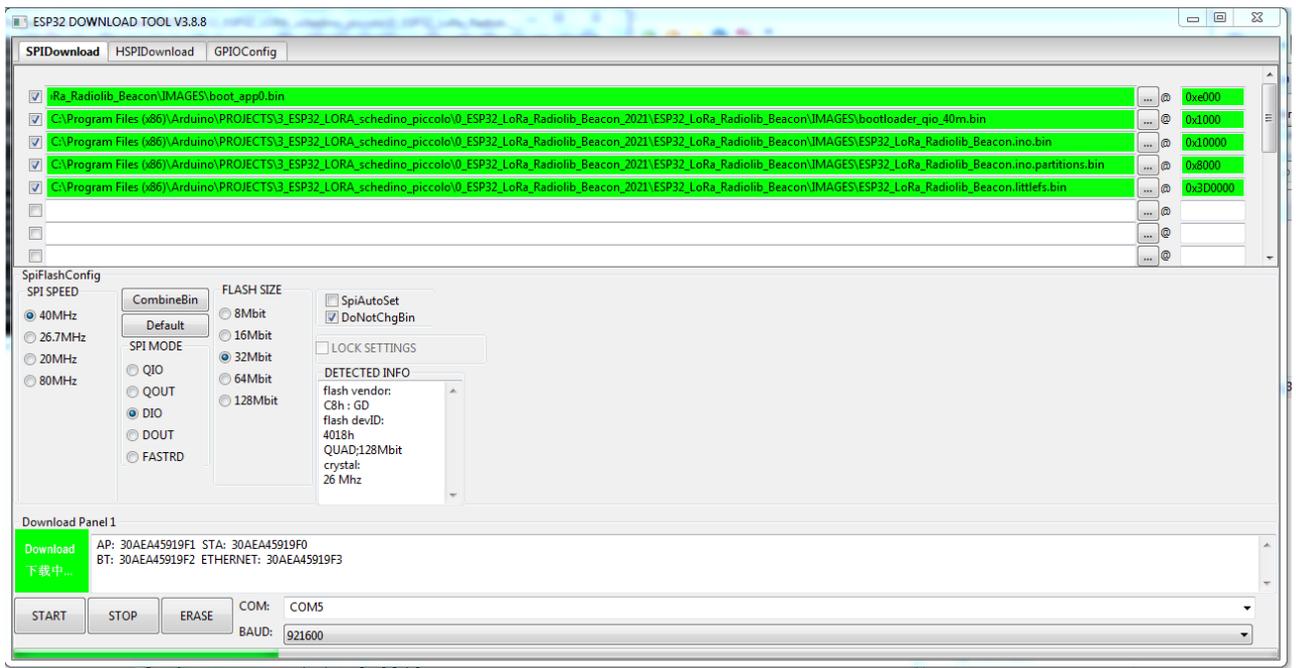


Figura 61 Settings Tool di download Sw per schedino Heltec WiFi LoRa 32

La figura a fianco mostra la schermata delle proprietà generali di questa versione.

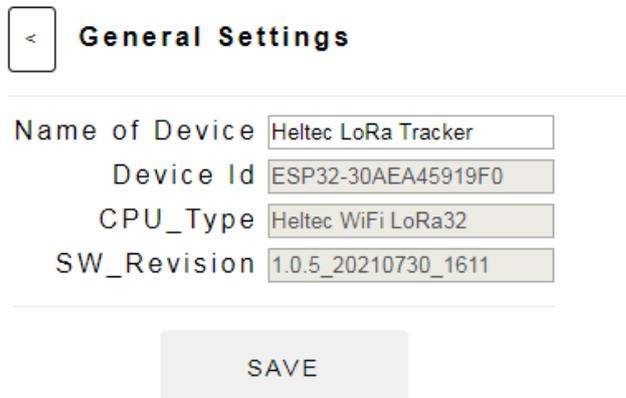


Figura 62 General Settings Heltec WiFi LoRa 32

6 Installazione SW via OTA (Over-The-Air)

Il SW LoRa Beacon fornisce una modalità per l'aggiornamento del SW senza richiedere alcun collegamento fisico al dispositivo target.

Questa modalità, chiamata OTA (Over-The-Air), sfrutta la funzionalità di WiFi Station di cui è dotato il SW per consentire, a meno del primissimo caricamento di una immagine sul dispositivo HW di destinazione, di aggiornare la versione SW presente sul dispositivo.

La disponibilità di questa funzione consente per esempio di aggiornare il SW di un dispositivo remoto senza bisogno di essere on-site.

L'aggiornamento del SW non impatta i dati di configurazione che quindi vengono preservati a cavallo di un aggiornamento SW.

Per poter realizzare quindi l'aggiornamento del SW è necessario disporre di un unico modulo SW da scaricare su un computer dotato di un browser della famiglia chrome o firefox .

Il computer che si intende utilizzare per l'aggiornamento SW deve essere in grado di raggiungere il dispositivo da aggiornare tramite WiFi (ed eventualmente tramite una connessione internet): quindi il dispositivo da aggiornare deve essere collegato a sua volta ad un access point da cui sia possibile raggiungere (anche tramite internet) il PC che si intende utilizzare per l'aggiornamento SW.

Quindi il primo passo per poter realizzare l'aggiornamento SW è verificare la connettività tra PC e dispositivo da aggiornare sfruttando per es. il classico strumento “ping” disponibile su qualsiasi PC.

Una volta verificata la connettività, è necessario porre il dispositivo da aggiornare in una condizione operativa particolare detta “Admin Mode” che è richiesta allo scopo di disabilitare temporaneamente alcune funzioni non necessarie nella fase di aggiornamento SW.

A tale scopo è possibile accedere alla schermata “OPERATION MODE SETTINGS” e selezionare la casella Admin_Mode salvando quindi la pagina.

Il dispositivo remoto effettuerà a questo punto un reboot e si ripresenterà con la modalità Admin_Mode selezionata.

Va osservato che il reboot automatico è disponibile esclusivamente utilizzando dispositivi basati sulla piattaforma HW LoRa Beacon; in caso di piattaforme diverse sarà necessario far ripartire manualmente il dispositivo remoto (e quindi sarà necessaria una persona vicina al dispositivo o un altro meccanismo per far ripartire il dispositivo in maniera remota ad es. cliccando l'alimentazione).

In queste condizioni il dispositivo apparirà temporaneamente non funzionante a livello LoRa in quanto tale funzionalità sarà, come spiegato , temporaneamente non disponibile.

L'attivazione della modalità Admin_Mode farà apparire una nuova interfaccia grafica dedicata all'aggiornamento SW: per accedere a tale interfaccia bisognerà puntare con una altra finestra del browser all'indirizzo del dispositivo remoto, usando un numero di porta pari a 88 come nell'esempio della figura a fianco.

A questo punto andranno inserite come credenziali (di default) i seguenti valori:

- User Id: admin
- Password: adminota

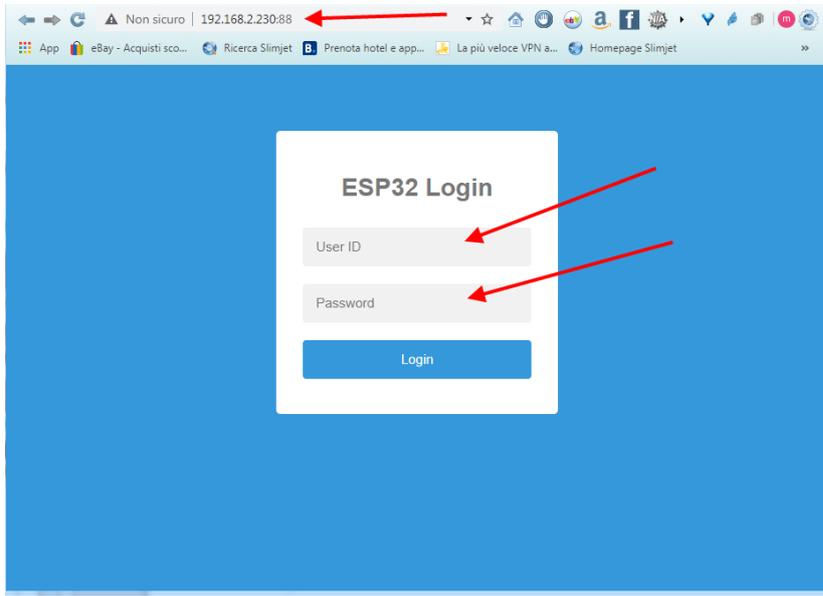


Figura 63 Interfaccia per caricamento SW via OTA

Apparirà una nuova schermata che inviterà a selezionare dal proprio PC la nuova immagine SW di aggiornamento: selezionare il file ricevuto come aggiornamento, il cui nome terminerà tipicamente in .bin.

Avviare quindi il caricamento del pacchetto SW ed attendere che l'operazioni si completi....

Terminata la fase di caricamento della nuova immagine di aggiornamento il dispositivo remoto ripartirà e si ripresenterà con la sua interfaccia grafica, in accordo al nuovo SW caricato; accedendo alla pagina "General Configuration" si potrà verificare che la versione SW del dispositivo sia quella attesa come effetto del caricamento effettuato.

A questo punto il dispositivo è aggiornato per cui è possibile uscire dal "Admin_Mode" tramite la pagina di "Operation_Mode_settings" ; il dispositivo effettuerà un nuovo reboot e si presenterà nella modalità di funzionamento selezionata.

A valle di ogni aggiornamento SW è bene controllare che la configurazione del dispositivo sia

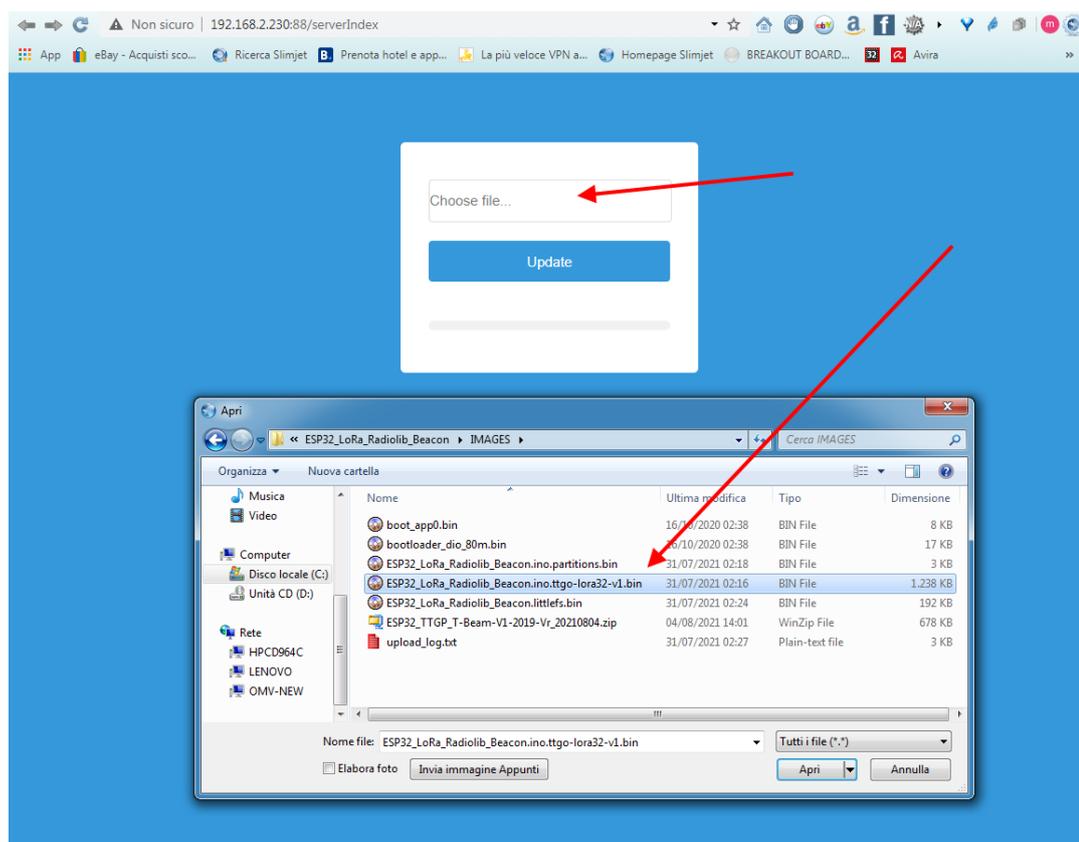


Figura 64 selezione immagine SW da caricare dal proprio PC

corretta in quanto a seconda dei cambi introdotti nella nuova versione SW potrebbero essere necessari dei piccoli aggiustamenti della configurazione; tali aggiustamenti dovrebbero essere eventualmente documentati nella “Release Note” che dovrebbe accompagnare ogni nuovo rilascio SW.

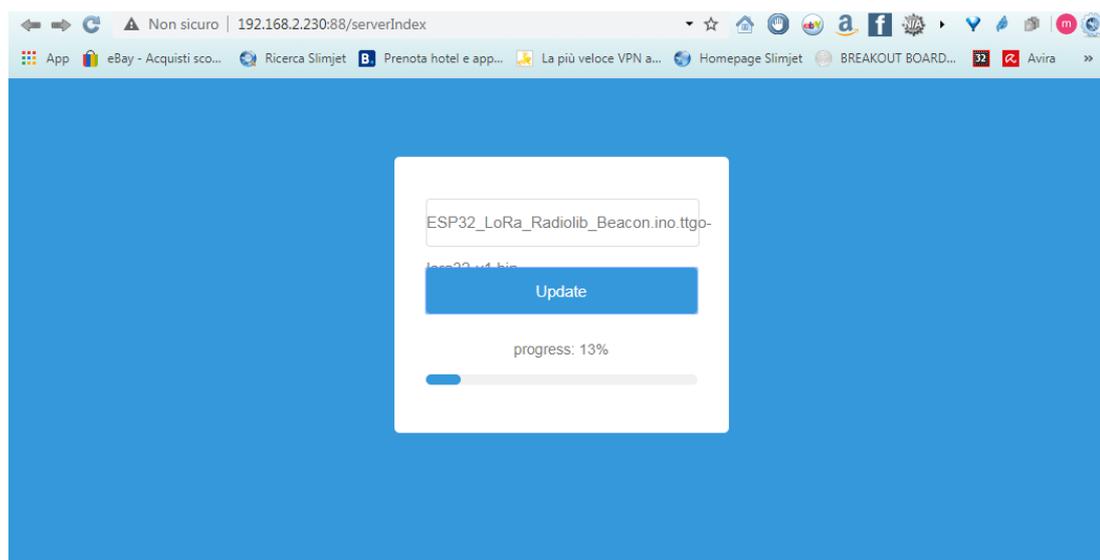


Figura 65 Display durante la fase di caricamento del nuovo SW

7 Salvataggio e Caricamento della configurazione via OTA

Il SW LoRa Beacon è stato concepito per essere utilizzato agevolmente per fare della sperimentazione basata sul protocollo radio LoRa per cui più che mirare ad un esercizio di sviluppo SW è mirato ad un utilizzo strumentale ai fini della valutazione della tecnologia LoRa in impegni non standard (secondo gli obiettivi per cui è nata), ovvero per servizi tipici del mondo dei radioamatori.



Figura 66 Schermata di Save/Restore della configurazione di un dispositivo LoRa Beacon

Quindi l'obiettivo è stato quello di fornire la possibilità di modificare agevolmente una serie di parametri di funzionamento senza dover necessariamente avere esperienza di sviluppo SW.

Il modo scelto è stato quello quindi di parametrizzare tutta una serie di elementi funzionali rendendoli modificabili tramite una opportuna interfaccia grafica; l'insieme dei parametri che caratterizzano una data impostazione di ogni dispositivo viene qui indicata con il termine "configurazione" ed è visualizzabile come un insieme di attributi e valori corrispondenti.

La configurazione di un certo dispositivo è mantenuta internamente al dispositivo in una opportuna memoria non volatile che può essere una FRAM per i dispositivi che chiamiamo LoRa Beacon HW oppure una frazione della memoria flash per quei dispositivi privi di FRAM.

Il modo canonico per gestire la configurazione di un dispositivo è tramite l'interfaccia grafica dello stesso, salvo a dare la possibilità di visualizzare, salvare su un PC esterno o caricare da un PC esterno una certa configurazione.

Il formato scelto per il file di configurazione è lo standard JSON che è agevolmente leggibile e gestibile sia con un semplice editor, che con uno dei tanti tools esistenti allo scopo.

La possibilità di salvare e ricaricare una certa configurazione consente di tenere agevolmente traccia delle condizioni di prova in cui un certo insieme di test viene effettuato; in mancanza di un tale strumento sarebbe certamente molto più complesso tenere traccia ordinata delle condizioni di prova allo scopo di poter fare dei confronti.

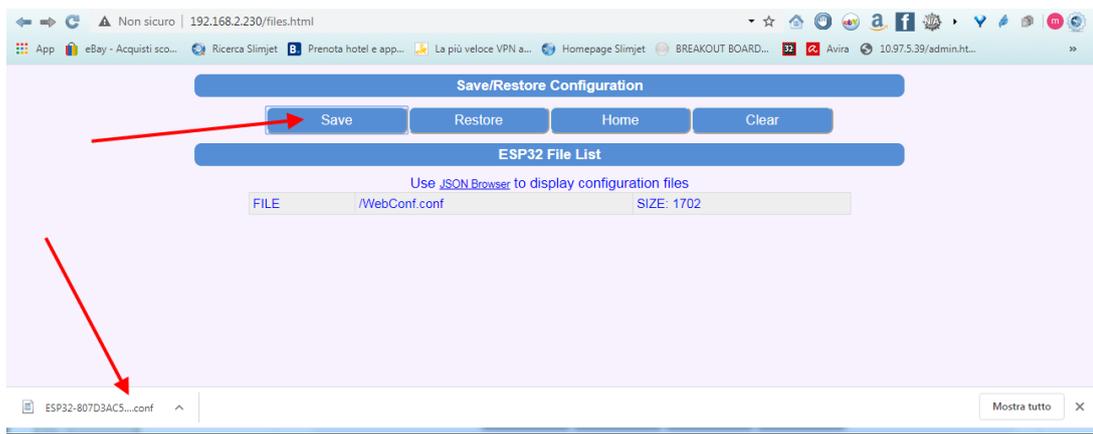
La scelta di un formato testuale per il file di configurazione consente di poter realizzare agevolmente il confronto di più files di configurazine per evidenziare eventuali differenze tra gli stessi.

Tutte le operazioni di save e restore della configurazione richiedono che il dispositivo sia messo nella modalità “Admin_Mode” come per le operazioni di aggiornamento SW.

Per accedere alle funzioni di save/restore della configurazione è disponibile la pagina “**SAVE/RESTORE CONFIGURATION**”.

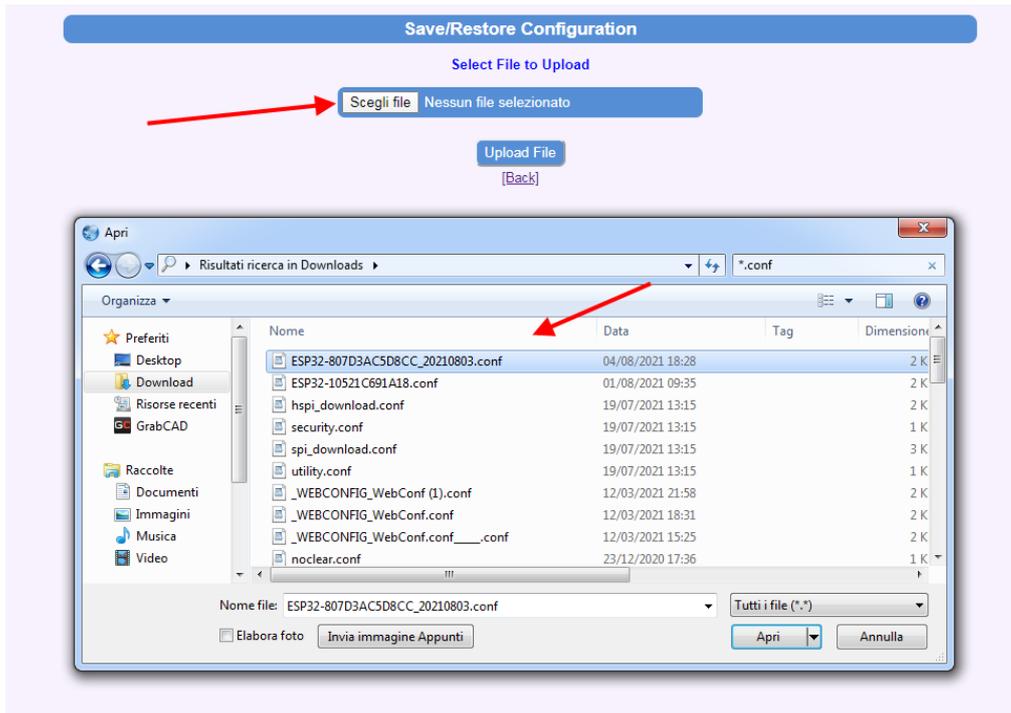
La figura seguente riporta il contenuto della schermata che appare: come si può notare è presente una lista di files che consente di conoscere i files di configurazione presenti nel dispositivo ed eventualmente caricati manualmente. Il File /WebConf.conf rappresenta sempre il file di configurazione attivo al momento.

Per salvare la configurazione corrente del dispositivo è sufficiente selezionare il tasto Save: verrà creato e scaricato un file direttamente sul PC, come dalla figura seguente:

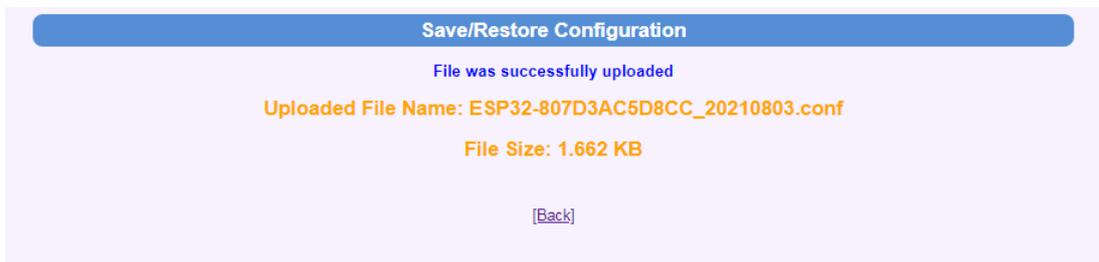


Il file scaricato avrà un nome del tipo “ESP32-<MAC address>.conf” e verrà salvato in una location dipendente dalla configurazione del browser in uso. La figura sopra è un esempio.

Per effettuare il Restore della configurazione da un certo file è sufficiente selezionare il tasto “Restore” e cercare e selezionare il file che interessa caricare; quindi selezionare il tasto Upload File.



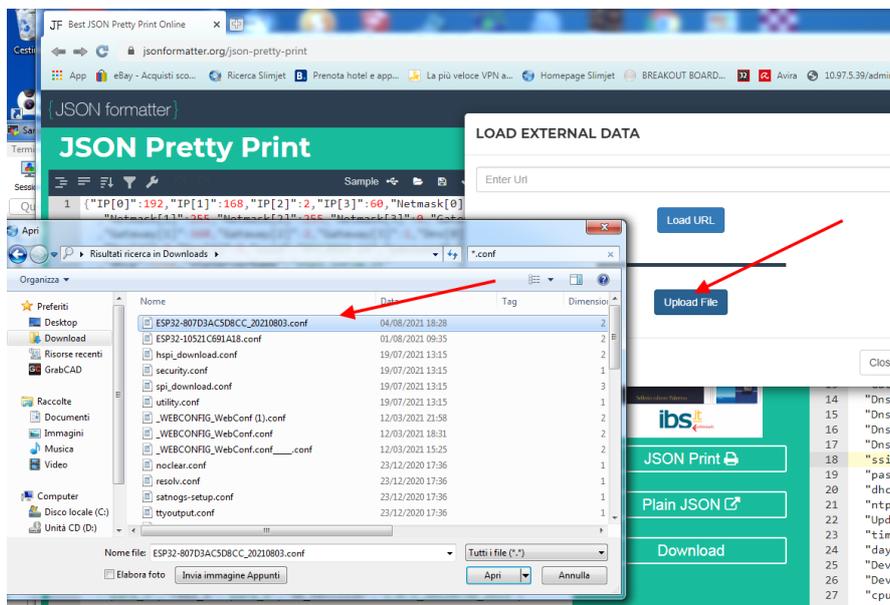
Se il caricamento avviene con successo si avrà una schermata del tipo:



Selezionando il tasto “Back” apparirà la schermata seguente dalla quale si evidenzia la presenza del file caricato tra i files presenti sul dispositivo.



Per rendere effettiva la nuova configurazione si rende necessario uscire dalla modalità “admin_Mode” e riavviare il dispositivo.



Volendo ispezionare e/o eventualmente modificare la configurazione di un dispositivo sulla base di un file di configurazione è possibile utilizzare per esempio il tool <https://jsonformatter.org/json-pretty-print> accessibile liberamente su internet tramite un qualsiasi browser tipo chrome o firefox.

La figura seguente è un esempio di utilizzo di questo tool per visualizzare in chiaro il contenuto di un file di configurazione di un certo dispositivo:

Una volta caricato il file da analizzare è possibile utilizzare il tool per modificare ed eventualmente salvare il file modificato.

La figura seguente riporta la schermata principale del tool di cui sopra.

Best JSON Pretty Print Online

JSON formatter

JSON Pretty Print

```

1 [{"IP[0]":192,"IP[1]":168,"IP[2]":2,"IP[3]":60,"Netmask[0]":255
  ,"Netmask[1]":255,"Netmask[2]":255,"Netmask[3]":0,"Gateway[0]":192
  ,"Gateway[1]":168,"Gateway[2]":2,"Gateway[3]":1,"Dns[0]":8,"Dns[1]":8
  ,"Dns[2]":8,"Dns[3]":8,"ssid":"RFC2019-24","password":"rollyepenny"
  ,"dhcp":true,"ntpServerName":"ntp1.inrim.it","Update_Time_Via_NTP_Every"
  :0,"timezone":10,"daylight":true,"DeviceName":"TTGO T-Beam Tracker"
  ,"DeviceId":"Not Available","cpu_type":"TTGO LoRa32-OLED V1","gps_debug"
  :false,"LoRa_debug":false,"RTC_debug":false,"ezTime_debug":false
  ,"pps_debug":false,"PE_debug":false,"BT_KISS_Mode":false
  ,"Serial_KISS_Mode":false,"Tracker_Mode":true,"IoT_enable":false
  ,"selftest_debug":false,"WebConfig_debug":false,"standalone":false
  ,"no_gps":false,"iGate_Mode":false,"Admin_Mode":true,"Reboot_Now":false
  ,"EnableRfToIs":false,"EnableIsToRf":false,"EnableRepeater":false
  ,"AprsHost":"rotate.aprs2.net","AprsPort":14580,"AprsLogin":"I0C...
  ,"AprsPass":"11111","AprsFilter":"m/40","AprsRawBeaconIGATE":"IC...
  >APZMDM,WIDE1-1:IGPS_LAT/GPS_LON#iGate-LoRa, 15.6/SF9
  ,"AprsRawBeaconTRACK":"I0C...>APZMDM,WIDE1-1:IGPS_LAT/GPS_LON#LoRa,15.6
  /SF9","AprsRawBeaconPeriodSecondsIGATE":301
  ,"AprsRawBeaconPeriodSecondsTRACK":31,"LoraFreq":433.725,"LoraBw":31
  .25,"LoraSF":7,"LoraCodingRate":7,"LoraSync":52,"LoraPower":17
  ,"LoraFreqCorr":10,"LoraPreambleLen":15,"Loc_Lat":4038.13N,"Loc_Lon"
  :01425.45E,"AprsLoggerAddr":"192.168.2.150","AprsLoggerPort":44444
  ,"IoT_Host":"io.adafruit.com","IoT_Port":1883,"IoT_Login":"mi...
  ,"IoT_Key":"aijC1...eIXUgyC9JbMdn","feed_1":"para_1","feed_2"
  : "para_2","feed_3":"para_3","feed_4":"para_4","feed_5":"para_5","feed_6"
  : "para_6","SW_Revision":"1.0.5_20210730_1611"}]

```

Load Data

2 Tab Space

Make Pretty

JSON Print

Plain JSON

Download

```

37 "IoT_enable": false,
38 "selftest_debug": false,
39 "WebConfig_debug": false,
40 "standalone": false,
41 "no_gps": false,
42 "iGate_Mode": false,
43 "Admin_Mode": true,
44 "Reboot_Now": false,
45 "EnableRfToIs": false,
46 "EnableIsToRf": false,
47 "EnableRepeater": false,
48 "AprsHost": "rotate.aprs2.net",
49 "AprsPort": 14580,
50 "AprsLogin": "I0C...",
51 "AprsPass": "11111",
52 "AprsFilter": "m/40",
53 "AprsRawBeaconIGATE": "IC...",
54 "AprsRawBeaconTRACK": "I0C...>APZMDM,WIDE1-1:IGPS_LAT/GPS_LON#iGate
  -LoRa, 15.6/SF9",
55 "AprsRawBeaconPeriodSecondsIGATE": 301,
56 "AprsRawBeaconPeriodSecondsTRACK": 31,
57 "LoraFreq": 433.725,
58 "LoraBw": 31.25,
59 "LoraSF": 7,
60 "LoraCodingRate": 7,
61 "LoraSync": 52,
62 "LoraPower": 17,
63 "LoraFreqCorr": 10,

```

8 Esempi di dove acquistare la componentistica

Tutta la componentistica necessaria per la costruzione di entrambe le versioni di LoRa Beacon è facilmente acquistabile sui classici portali asiatici; è importante tenere conto che per quanto generalmente è molto agevole trovare quanto necessario per le proprie esigenze, **non bisogna essere superficiali nell'acquistare componenti apparentemente identici a quelli consigliati, ma decisamente più economici**: è purtroppo una classica trappola in quanto come ben noto non esiste in pratica nessun reale tipo di garanzia che i componenti acquistati siano conformi a quanto dichiarato o anche mostrato in foto.

Il consiglio è di cercare si l'affare, ma badando molto bene a che sia realmente tale !

Come detto la totalità della componentistica è facilmente reperibile; gli unici componenti ovviamente non reperibili su tali portali sono i circuiti stampati che possono essere richiesti al seguente indirizzo info@sarimesh.net: il prezzo di tali PCB è comunque molto contenuto in quanto non gravato da particolari markup ma funzione delle sole spese di fabbricazione e spedizione.

A seguire si fornisce una lista puramente indicativa, alla data, dei componenti richiesti per il montaggio delle due versioni di LoRa Beacon: la lista è unica per le due versioni per cui sulla base del tipo di versione che si vuole montare sarà necessario eventualmente acquistare solo un sottinsieme di componenti. **Non si assume ovviamente nessuna responsabilità per eventuali errori contenuti in questa lista, né per l'affidabilità dei venditori indicati.**

I tempi di spedizione dalla Cina sono ultimamente calati moltissimo grazie all'entrata in vigore della nuova regolamentazione sugli acquisti effettuati sui portali online; è però da considerare che tutti i prezzi che si vedono sui portali vanno poi maggiorati dal valore dell'IVA (22%) all'atto della conclusione degli ordini.

Lista di esempio per acquisto parti aggiornata al 20210805

- [Microcontrollore ESP32-WROOM-32D versione 38 pin con antenna stampata](#)
- [Display I2C 0,96"](#)
- [Display SPI IPS 1.14" 135x240 LCD Module](#)
- [FRAM FM24W256](#)
- [Moduli GPS NEO-6M NEO-7M NEO-8M](#)
- [Port Expander PCF8574](#)
- [Modulo LoRa 1 Watt SX1268 Ebyte E22-400M30S](#)
- [Modulo LoRa 100milliWatt SX1268 E22-400M22S](#)
- [Modulo LoRa 100 milliWatt SX1278 RFM98W](#)
- [Buzzer KY-012](#)
- [Connettore USB Verticale per PCB](#)
- [Assortimento di diodi](#)
- [transistor 2N2222](#)
- [Assortimento LEDs 5 mm rosso e verde](#)
- [MCP1700](#)

-
- [Tastino 6x6x12 verticale](#)
 - [Strisce contatti 2.54 mm altezza 7,1 mm femmina](#)
 - [strisce pin 2.54 mm maschio angolo retto tipo R1](#)
 - [DS3231 RTC](#)
 - [Cavetto IPX-SMA 15 cm](#)
 - [Strisce pin 2.54 mm maschio diritto](#)
 - [Assortimento condensatori ceramici](#)
 - [Assortimento resistenze](#)
 - [Zoccoletto DIP 16 pin](#)
 - [Antennina con calamita e attacco SMA 433 Mhz](#)
 - [Cavo prolunga USB](#)
 - [Connettore alimentazione 12V](#)
 - [Antenna GPS esterna attiva](#)
 - [DC/DC Step down conv.](#)
 - [Assortimento condensatori elettrolitici](#)