CENTRALINA

DI RILEVAZIONE DEI DATI

SULLA QUALITA' DELL'ARIA



Note di progetto e manuale d'uso

Rev. 1 – 15/4/19

Autore: Luca Crotti

Sommario

1. PRESENTAZIONE del PROGETTO "MILANO SMART PARK" 3

1.1. Obiettivi 3

1.2. La centralina 3

1.3. Funzionalità della centralina 4

1.4. Il sito web 5

2. LO SVILUPPO DELLA CENTRALINA 6

2.1. AMBIENTE DI SVILUPPO SOFTWARE 7

2.2. Progettazione e schema elettrico 8

2.3. Realizzazione PCB scheda 9

2.4. Elenco dei componenti utilizzati (BOM) 10

2.5. Funzionalità del software 10

2.5.1. Tipologia di trasmissione 12

2.5.2. Funzionamento della scheda di memoria 13

3. Montaggio della centralina 15

3.1. Elenco degli attrezzi necessari 15

3.2. Preparazione dei collegamenti 17

3.3. Preparazione della scatola che conterrà la parte elettronica 22

3.4. Montaggio dei componenti sulla scheda 25

3.5. Collegamento dei vari sensori alla scheda 30

3.6. collegamento dell'alimentazione e primo avvio 35

4. USO DELLA CENTRALINA 36

4.1. Configurazione parametri 36

4.2. Guida alla lettura delle informazioni a schermo 37

4.3. Guida alla installazione di ESP32 sull'IDE Arduino (per programmare) 42

4.3.1. Installazione di Arduino IDE *42*

4.3.2. Installazione dell'Hardware e SW di sviluppo di ESP32 su IDE arduino *43*

4.3.3. Installazione delle librerie necessarie per il funzionamento dei sensori *45*

4.3.4. PROGRAMMAZIONE della scheda ESP32 ..... *46*

1. PRESENTAZIONE del PROGETTO “MILANO SMART PARK”

1.1. Obiettivi

Le 8 Azioni del progetto

Le riflessioni che hanno condotto alla scelta della modalità di intervento, suddivisa in 8 azioni, per la realizzazione del progetto #MilanoSmartPark sono nate dall’esperienza accumulata negli anni dall’Associazione Parco Segantini e dall’identificazione di buone pratiche che sollecitano la partecipazione attiva di cittadini di tutte le età, soggetti svantaggiati, istituzioni, scuole, associazioni, mondo scientifico e comunità e, in particolare, sensibilizzando e stimolando i diversi tipi di frequentatori:

- nell’udire i suoni della natura;
- nell’ascoltare e produrre la musica;
- nell’osservare flora e fauna dell’area naturalistica;
- nel seguire il percorso sensoriale che favorisce il riconoscimento del luogo come spazio, attraverso la stimolazione olfattiva (spazio snoezelen);
- nel gustare e nel toccare i prodotti della terra.

L’associazione Parco Segantini Onlus, all’interno del progetto delle “8 azioni”, ha deciso di lanciare il progetto di una centralina di rilevazione dei dati ambientali, allo scopo di diffondere conoscenze e competenze:

- sulla complessità del fenomeno dell’inquinamento atmosferico, nel corso delle giornate e nei diversi periodi stagionali, per i diversi parametri: temperatura, umidità, pressione, identificando, con i monitoraggi del DISAT della Università Milano-Bicocca i più importanti agenti inquinanti come la anidride carbonica, l’ozono o il particolato;
- sull’interpretazione dei dati e sui comportamenti quotidiani più adatti a proteggerci dall’ inquinamento atmosferico;
- sulle nuove tecnologie che sono di semplice utilizzo ed economiche, la cui padronanza da un lato migliora la vita, dall’altro può costituire un trampolino di lancio per i giovani che vogliono inserirsi nel mondo del lavoro con iniziative imprenditoriali e avanzate.

1.2. La centralina

Il progetto, in particolare, consiste nella progettazione e realizzazione di una centralina per la rilevazione e misurazione di parametri caratterizzanti la qualità dell'aria, utilizzando metodi e strumenti tecnologici innovativi (IoT), che ne permettano la autoproduzione parziale da parte dei cittadini.

Si sono dovuti quindi individuare sul mercato i singoli componenti ed acquistarli nelle quantità previste sia per le fasi di test che per la fase di ingegnerizzazione finale.

E’ altresì stato definito e messo in opera il sistema di rilevazione, comunicazione, elaborazione e diffusione dei dati raccolti via web e definiti i sistemi di diffusione in rete.

Il progetto è stato portato avanti con la collaborazione attiva, altre che di APSo, anche del Comune di Milano, di Cooperativa Sociale OPERA IN FIORE, DISAT Bicocca Milano e Istituto Feltrinelli, ed ha coinvolto un discreto numero di persone tra appassionati, residenti, docenti, studenti oltre ad altri cittadini interessati.

Le finalità e i risultati attesi da APSo sono:

- diffusione di conoscenza e sensibilità ambientali, diffusione della conoscenza scientifica sul tema della qualità dell'aria
- creazione di una rete di centraline nel territorio lombardo, al fine di coordinare azioni di tutela ambientale condivisa;
- formazione teorica e pratica dei fruitori del Parco sul tema della qualità dell'aria creazione di un pacchetto di formazione relativo all'implementazione e all'uso delle Centraline e diffusione tramite municipalità, comuni, replicabile in altre realtà sul territorio;

verifica dell'opportunità per la creazione di nuovo lavoro, nell'ambito del settore ""green economy".

1.3. Funzionalità della centralina

La centralina è equipaggiata con i seguenti dispositivi:

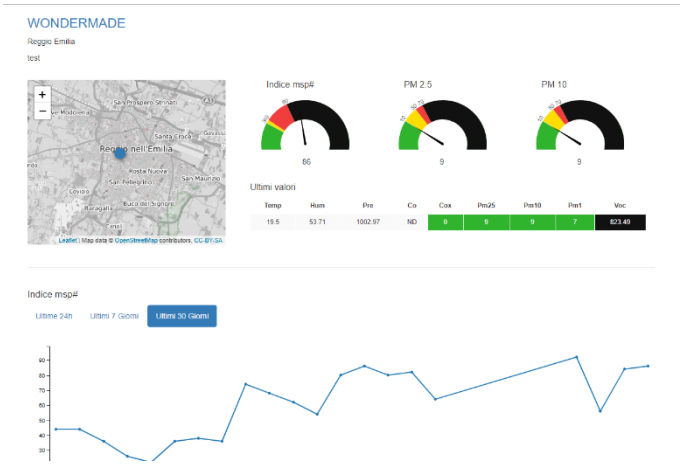
- ✓ Controllore integrato ESP32-Wrover
- ✓ Antenna amplificata wi-fi 6dBi
- ✓ Sensore BME680
- ✓ Sensore MQ7
- ✓ Sensore PMS5003
- ✓ Display OLED 126x64
- ✓ Lettore micro SD card
- ✓ Memoria interna da 8GB

Grazie ai sensori di cui è composto è in grado di svolgere le seguenti funzionalità:

- ✓ lettura temperatura ambientale (°C)
- ✓ lettura umidità ambientale (%HR)
- ✓ lettura pressione barometrica (mbar)
- ✓ lettura Cox (ppm)
- ✓ stima e-VOC (ppm)
- ✓ lettura concentrazione PM1
- ✓ lettura concentrazione PM2.5
- ✓ lettura concentrazione PM10
- ✓ conta particellare PM1, 2.5, 5, 10
- ✓ gestione automatica della trasmissione in rete dei dati
- ✓ salvataggio in locale dei dati in un opportuno file di log
- ✓ Predisposizione per interfacciamento con sensore di Ozono (O3)
- ✓ Predisposizione per interfacciamento con sensore NOx (tipo MICS-4541)
- ✓ Trasmissione dei dati in rete ogni 30 minuti
- ✓ Basso consumo grazie alla gestione efficiente del risparmio energetico in fase di attesa tra successive trasmissioni

1.4. Il sito web

I dati raccolti dalla centralina sono inviati in modo automatico sul sito dell'associazione, dove sono raccolti e memorizzati. Attraverso opportune interfacce web è possibile visualizzare i dati di ciascuna grandezza monitorata, analizzarne l'andamento ed i dati storici.



Anche se non precisi come i sensori utilizzati all'interno delle centraline professionali, i sensori installati nella centralina realizzata consentono di verificare in modo molto immediato l'andamento della qualità dell'aria ed è stato anche creato un indice complessivo, chiamato INDICE SEGANTINI, allo scopo di dare un giudizio generale sulla qualità dell'aria al momento della lettura.

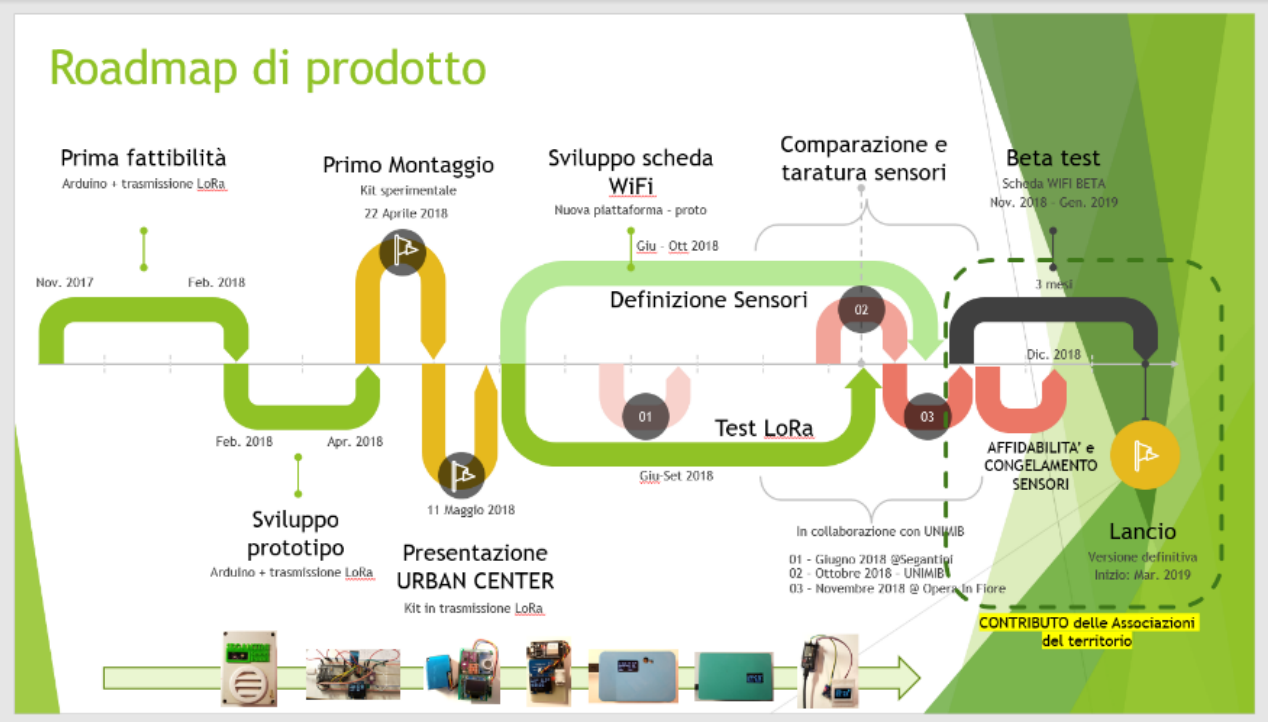
Attraverso le funzionalità offerte dal sito web è possibile quindi:

- ✓ Registrare una nuova centralina
- ✓ Indicare precisamente il punto di installazione della centralina (e quindi visualizzare su una mappa l'esatta collocazione)
- ✓ Gestire le diverse centraline attraverso un codice univoco
- ✓ Verificare l'andamento delle letture per diverse centraline sparse sul territorio
- ✓ Farsi un'idea della qualità dell'aria in una ampia zona osservando la qualità media di un gruppo omogeneo di centraline

2. LO SVILUPPO DELLA CENTRALINA

Lo sviluppo della centralina ha richiesto circa 14 mesi tra definizione delle specifiche, sviluppo e realizzazione. Sono stati realizzati un totale di 8 prototipi prima di arrivare alla versione definitiva attuale.

Di seguito si riporta un breve grafico che illustra la ROAD MAP seguita:



Per poter centrare l'obiettivo della DIVULGAZIONE, sono state effettuate precise scelte progettuali, che consentono a persone interessate di comprendere il funzionamento della centralina, di imparare a montare nuove centraline, di poter, nel tempo, anche modificare l'attuale configurazione hardware e software.

In particolare qui di seguito si dettagliano le scelte progettuali utilizzate, specificandone anche il razionale che ne ha comportato la scelta:

A. OPEN SOURCE:

la centralina è totalmente OPEN SOURCE. Questo significa che gli schemi, il software ed i componenti sono noti, a disposizione di tutti per future modifiche o semplicemente a fini didattici. Oltre a questo, il sistema di sviluppo, l'hardware utilizzato ed i software di progettazione sono anch'essi TUTTI OPEN SOURCE. Senza entrare nel dettaglio (si rimanda a: https://it.wikipedia.org/wiki/Open_source), si sottolinea qui che open-source non necessariamente significa FREE, anche se spesso (ed anche nel nostro caso) quasi tutte le risorse utilizzate sono state messe a disposizione a titolo gratuito dalla comunità di appassionati. L'open source è molto più che FREE.

B. BASSO COSTO:

per poter essere costruita da tutti, la centralina deve costare poco. I sensori soprattutto, sono stati scelti per l'ottimo compromesso tra precisione, affidabilità e costo, soprattutto considerando il prezzo

RETAIL (ossia al dettaglio, ossia per chi, come noi, ne acquista poche quantità). Per poter essere utilizzato nelle scuole e da appassionati di ogni tipo, la centralina ha avuto un target di costo limite di circa 150€ (ossia 10 volte meno il più economico dispositivo commerciale di prestazioni paragonabili).

L'obiettivo è stato raggiunto, pur con qualche compromesso.

C. FACILITA' di REPERIMENTO, DI COSTRUZIONE, DI UTILIZZO

Un altro aspetto molto importante è relativo alla facilità di reperimento dei materiali, soprattutto in un formato gestibile da appassionati senza apparecchiatura specializzata.

Si è fatto largo uso di soluzioni commerciali, la disponibilità dei materiali è stata verificata su siti quali Amazon e altri negozi online, e si sono preferiti, dove possibile, soluzioni per i sensori adatte alla sperimentazione (in gergo "breakout boards") in cui cioè lo spazio non è ottimizzato a favore della facilità di saldatura, oppure utilizzando apposite schede parte di kit di sviluppo (SDK).

Dove possibile, poi, si è scelto di utilizzare componenti il più possibile prodotti con tecnologia PTH e non SMD (PTH – in inglese Pin Through Hole, riferito a componenti elettronici con terminali (pin) che per essere montati sul circuito stampato richiedono uno o più fori (hole). Il termine si contrappone alla tecnologia più recente detta Surface mount technology (SMT) e i dispositivi elettronici che l'adottano si chiamano Surface mounting device (SMD) cioè dispositivi a montaggio superficiale), per ovvie ragioni di facilità di utilizzo data dalla non eccessiva miniaturizzazione dei componenti.

A livello di progettazione dell'hardware non si è cercata l'integrazione spinta, a scapito della facilità di montaggio, bensì è stata realizzata una scheda di raccolta di tutto l'hardware che avesse funzioni un po' da grosso "connettore" con chiare indicazioni di montaggio. I connettori utilizzati hanno un verso unico di inserimento e la serigrafia sulla basetta riporta i nomi dei componenti in modo che le possibilità di errore fossero ridotte veramente al minimo.

2.1. AMBIENTE DI SVILUPPO SOFTWARE

Un aspetto sicuramente centrale per garantire la riproducibilità del lavoro è la scelta del SW e più in dettaglio dell'ambiente di sviluppo.

In origine si è scelto il sistema di sviluppo basato su ARDUINO, per via della sua enorme diffusione e alta versatilità. Già utilizzato in molte scuole, la rete è piena di progetti, forum, video tutorial e altro materiale per qualsiasi applicazione alla quale si può pensare.

Dopo i primi test ci si è resi conto che l'ambiente di sviluppo di Arduino presentava purtroppo grossi limiti in termini di performances e di espandibilità: all'aumentare della complessità il sistema Arduino ha subito mostrato limiti nella capacità di accogliere un alto numero di sensori e sistemi da gestire, come pure una generale mancanza di memoria di programma. Oltre a questo, la gestione dell'ambiente di rete (ed in particolare del protocollo wifi) ha fin da subito mostrato i limiti della piattaforma.

Per l'evoluzione della scheda si è scelto tuttavia di restare il più possibile legati all'ambiente di sviluppo di arduino (IDE) in quando di assoluta facilità per tutti i non addetti ai lavori.

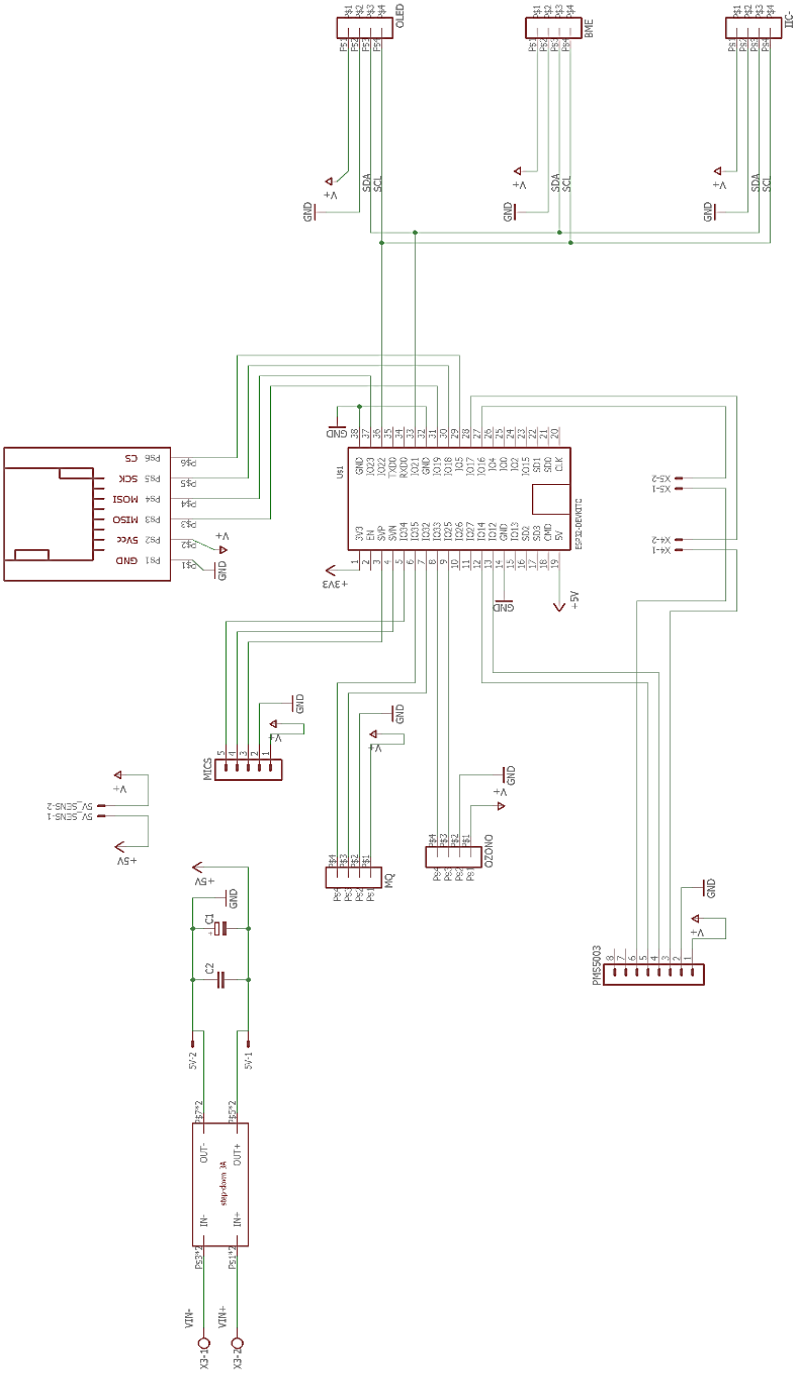
Sul mercato già da qualche tempo era presente il processore ESP32 (chi usa arduino ha sicuramente sentito parlare del sistema ESP8266, fratellino minore dell'ESP32). Il sistema è stato portato su IDE

Arduino, il che significa che è possibile (a meno di qualche livello di astrazione) utilizzare lo stesso IDE utilizzato per Arduino anche per programmare e flashare (trasferire il firmware) utilizzando hardware differenti da Arduino e ATMEL.

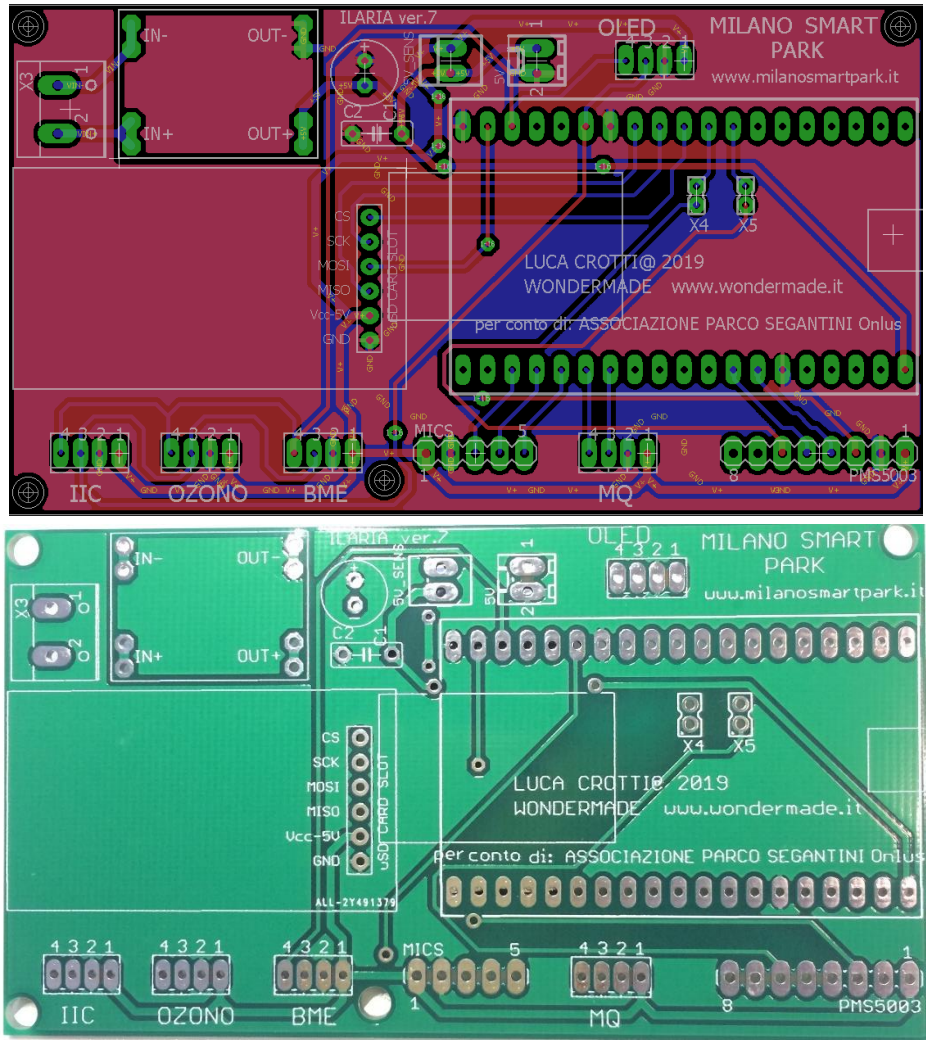
ESPRESSIF ha fatto un grosso lavoro con il suo processore, che gestisce in modo nativo la rete WIFI e che ha sufficiente potenza di calcolo per soddisfare le nostre esigenze grazie alla sua architettura dual core da 240MHz e 4MB di memoria integrati. Il fattore di forma poi è un altro punto che ne ha decretato la scelta (19+19 pin in linea, 100% compatibile con breadboard e altri sistemi di sviluppo).

2.2. Progettazione e schema elettrico

Lo schema elettrico è stato progettato con il SW EAGLE (ora Autodesk, le versioni precedenti prodotte da Cadsoft):



2.3. Realizzazione PCB scheda



La scheda è stata completata con Eagle e sono stati generati i file GERBER ed EXCELLON necessari.

I file sono stati controllati con un viewer di file gerber per verificare la correttezza della generazione.

Dopo di che i file sono stati inviati service chiamato ALLPCB, per la realizzazione, che ha richiesto circa 3 settimane (spedizione inclusa).

2.4. Elenco dei componenti utilizzati (BOM)

Si elencano di seguito i componenti utilizzati:

ID	Part	Device	Package	CATEGORIA	Description	FORNITORE	NOTE
1	5V	22-27-2021-02	6410-02	CONNECTOR	CONNECTOR	AMAZON	
2	5V_SENS	22-23-2021	22-23-2021	CONNECTOR	.100 (2.54mm) Center Header - 2 Pin"	MOLEX	22-23-2021
3	AN-1	JST_4PIN_20	JST_4PIN_20	CONNECTOR	CONNECTOR	AMAZON	
4	C1	CPOL-EUE2.5-7	E2,5-7	COMPONENT	POLARIZED CAPACITOR, European symbol	AMAZON	
5	C2	C-EU050-025X075	C050-025X075	COMPONENT	CAPACITOR, European symbol	AMAZON	
6	MICS	MA05-1	MA05-1	CONNECTOR	PIN HEADER	AMAZON	
7	OLED	JST_4PIN_20	JST_4PIN_20	CONNECTOR	CONNECTOR	AMAZON	
8	PMS5003	MA08-1	MA08-1	CONNECTOR	PIN HEADER	AMAZON	
9	U$1	ESP32DEVKITC	ESP32-DEVKITC	MICROCONTROLLER	Espressif Systems ESP32-DevKitC-VIB	MOUSER	356-ESP32-DEVKITCVIB
10	U$2	USD_CARD	USDCARD_BOARD	BOARD	uSD BOARD	AMAZON	
11	U$3	STEP_DOWN	STEP_DOWN_MODULE	BOARD	MP1584EN Ultra Small DC-DC 3A	AMAZON	
12	U$4	JST_4PIN_20	JST_4PIN_20	CONNECTOR	Cavo 4 poli + connettore femmina	AMAZON	
13	U$5	JST_4PIN_20	JST_4PIN_20	CONNECTOR	Cavo 4 poli + connettore femmina	AMAZON	
14	U$6	JST_4PIN_20	JST_4PIN_20	CONNECTOR	Cavo 4 poli + connettore femmina	AMAZON	
15	X3	W237-102	W237-102	CONNECTOR	WAGO SCREW CLAMP	NEWARK	70K9898
16	X4	87758-0216	87758-0216	CONNECTOR	2 Pin - 2mm Dual Row Single Wafer, Vertical T/H HDR	NEWARK	94M1513
17	X5	87758-0216	87758-0216	CONNECTOR	2 Pin - 2mm Dual Row Single Wafer, Vertical T/H HDR	NEWARK	94M1513
18	S1	PMS 5003	SENSORE	SENSOR	sensore §PM PMS5003	MOUSER	485-3686
19	S2	BME680	SENSORE	SENSOR	Adafruit BME680 - T, H, P & G Sensor	MOUSER	
20	S3	MQ7	SENSORE	SENSOR	sens. MQ7 Cox	MOUSER	485-3660
21	PP1	OLED 1.3" 128x64	OLED IIC 4 pin	DISPLAY	display on board	MOUSER	
22	SV1	CON	OLED	CONNECTOR	Cavo 4 poli + connettore femmina	AMAZON	
23	SV2	CON	BME	CONNECTOR	Cavo 4 poli + connettore femmina	AMAZON	
24	SV3	CON	SensPM	CONNECTOR	CONNETTORE 8 pin	AMAZON	
25	SV4	CON	4514	CONNECTOR	Cavo 5 poli + connettore femmina	AMAZON	
26	SV6	CON	DAC	CONNECTOR	Cavo 4 poli + connettore femmina	AMAZON	
27	CON1	19+19 strip pin	socket for ESP 32	CONNECTOR	2.54mm pin header female	AMAZON	
28	SD1	CARD	uSD card	DEVICE	SDHC scheda per LOG e per settaggio	AMAZON	
29	D1	POWER	alimentatore 9V 2A	DEVICE	SoulBay 9V 2A AC/DC Adattatore Caricatore	AMAZON	
30	CON2	CON_PWR	connettori	CONNECTOR	Presa jack di alimentazione DC 5,5 x 2,1 mm	AMAZON	
31	B1	BOX	scatola	BOX	scatola 190x100 IP65 BRICOMAN	BRICO	
32	B2	BOX_CONN	scatola	BOX	tappi per sensori per scatola D30mm	BRICO	
33	D2	ANT	ANT	DEVICE	antenna 8dB IPEX	AMAZON	
34	PCB1	PCB1	PCB		PCB	SERVICE	
35	*R1*	*R1*	*10k*	*non usato*	*RESISTOR, European symbol*	AMAZON	
36	*R2*	*R2*	*10k*	*non usato*	*RESISTOR, European symbol*	AMAZON	
37	*R3*	*R3*	*10k*	*non usato*	*RESISTOR, European symbol*	AMAZON	
38	*R4*	*R4*	*10k*	*non usato*	*RESISTOR, European symbol*	AMAZON	
39	*SW2*	*DIP SWITCH*	*MODE*	*non usato*	*DIP SWITCH*	AMAZON	

2.5. Funzionalità del software

Il software utilizzato è stato sviluppato in ambiente IDE Arduino ed utilizza alcune librerie specifiche per i sensori. L'ambiente di arduino non è configurato all'origine con tutti i device che la centralina utilizza, è quindi necessario settare vari parametri che saranno di seguito dettagliati.

Il sw utilizza una struttura tipica per l'ambiente Arduino, basata sui costrutti SETUP() e LOOP() che hanno 2 utilizzi differenti.

La funzione "setup()" è utilizzata solo 1 volta all'avvio del codice (dopo reboot oppure quando si da alimentazione) e generalmente è utilizzata per le fasi di configurazione. Nel codice della centralina all'interno della funzione SETUP si trovano:

- ✓ Inizializzazione delle variabili
- ✓ Avvio comunicazione con sensori, display oled e seriale (di fatto si ISTANZIANO le strutture dati)
- ✓ Gestione della parte di DEEP SLEEP

- ✓ Avvio display e splash screen
- ✓ Fase preliminare di lettura della scheda SD
- ✓ Fase preliminare di scansione della rete wifi

La funzione “LOOP()” invece è la funzione che cicla per tutta la vita, intendendo come vita in modo indefinito il periodo che intercorre tra 2 accensioni o tra 2 reboot. Nel nostro caso, tuttavia, anche il ciclo all’interno di LOOP() è interrotto ad un certo punto per l’avvio della procedura di DEEP SLEEP.

Tutta la struttura del codice si articola all’interno di una grossa macchina a stati (FSM, vedere teoria qui: https://it.wikipedia.org/wiki/Automa_a_stati_finiti) con le seguenti fasi:

- ✓ FASE 5: aggiornamento data e ora
- ✓ FASE 10: aggiornamento sensore MQ7, se presente
- ✓ FASE 20: aggiornamento sensore MICS, se presente
- ✓ FASE 30: aggiornamento sensore SDS021 (sensore PM prima versione) se presente
- ✓ FASE 35: aggiornamento sensore PMS5003 (sensore PM), se presente
- ✓ FASE 40: aggiornamento sensore BME280, se presente
- ✓ FASE 45: aggiornamento sensore BME680, se presente
- ✓ FASE 50: aggiornamento sensori di LUCE e UV, se presente
- ✓ FASE 60: scrittura dati su seriale
- ✓ FASE 70: generazione file di LOG e scrittura dati LOG su SD, se SD presente
- ✓ FASE 80: aggiornamento dati su display OLED e gestione visualizzazione icone
- ✓ FASE 90: verifica presenza WIFI e tentativo di riconnessione, se necessario
- ✓ FASE 100: aggiornamento server (Host: api.milanosmartpark.net)

Alla fine del ciclo LOOP è presente la funzione che manda in DEEP SLEEP il processore, dopo aver visualizzato il codice MAC a schermo.

La gestione a fasi consente di avere una chiara visuale del programma in fase di debug, ed ha anche una forte valenza didattica.

Oltre a questo, le fasi si prestano bene ad essere “attivate” in base all’hardware presente, e gestite con semplici istanze di configurazione.

Nel codice è presente una variabile chiamata DEBBUG (si, scritto con 2 “B”) che ha lo scopo di gestire il dettaglio dei messaggi scritti su seriale. È una variabile booleana e, quando “true”, consente di avere molti messaggi di dettaglio inviati su seriale, in modo da facilitare il debug o la risoluzione di problemi di funzionamento. Alcuni messaggi di debug sono anche scritti sul file di LOG, in modo da tracciare particolari tempi o doppi passaggi nelle fasi.

All’inizio del codice, dopo i commenti che mostrano i collegamenti elettrici ed i segnali di base che sono gestiti dalla logica, sono presenti le chiamate alle librerie, che si riportano qui per completezza:

```
#include "FS.h"
#include "SD.h"
#include "SPI.h"

#include <Arduino.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BME280.h>
#include <Adafruit_BME680.h>
```

```
#include <Wire.h>

// librerie per gestione RETE WIFI
#include <WiFi.h>
#include <NTPClient.h>
#include <WiFiUdp.h>

// librerie per gestione display OLED
#include <U8g2lib.h>

// strutture e istanze BME280
#define SEALEVELPRESSURE_HPA (1013.25)
#define BME280_ADD 0x76
Adafruit_BME280 bme; // I2C///////Adafruit_BME280 bme(I2C_SDA, I2C_SCL);

//strutture e istanze per BME680
//#define SEALEVELPRESSURE_HPA (1013.25)
Adafruit_BME680 bme680; // I2C

// Istanza monitor OLED 1.3" controller SH1106... monitor 0.96" invece con SSD1306
U8G2_SH1106_128X64_NONAME_F_HW_I2C u8g2(U8G2_R0, U8X8_PIN_NONE, 22, 21);   // ESP32
Thing, HW I2C with pin remapping

// seriale per SDS021 o per PMS5003
HardwareSerial PMSerial(2);  // seriale1  (UART0=0; UART1=1; UART2=2)
```

Di seguito è presente la parte di gestione del DEEP SLEEP e l'istanza delle strutture per la gestione della rete (CLIENT, NTPUDP e SERVER):

```
//$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
// definizioni per deep speep
#define uS_TO_S_FACTOR 1000000  /* Conversion factor for micro seconds to seconds */
#define TIME_TO_SLEEP  1800     /* Time ESP32 will go to sleep (in seconds) 30 minuti =
1800 sec */
RTC_DATA_ATTR int bootCount = 0;
//$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

File root;
WiFiClient client;


// Define NTP Client to get time
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP);

//++++++++++++ CREDENZIALI PER SITO WEB   ++++++++++++++++++++
//sito web: https://api.milanosmartpark.net
const char* server = "api.milanosmartpark.net";
//++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

2.5.1. Tipologia di trasmissione

La trasmissione dei dati avviene tramite comandi di GET e POST, le api utilizzate sono state prese da versioni precedenti del software, in cui la trasmissione avveniva verso il sito "thingspeak" (una piattaforma di servizi IOT che offre un piano gratuito di utilizzo fino a 3 canali e fino a 8 campi/canale) e modificate per essere adattate alla struttura del DB presente sul server del sito Segantini.

La parte di trasmissione avviane, come detto, in fase 100, e questo è il pezzo di codice interessato:

```
client.print("POST /api/channels/writelog HTTP/1.1\r\n");
client.print("Host: api.milanosmartpark.net\r\n");
client.print("Connection: close\r\n");
client.print("User-Agent: Wondermade\r\n");
```

```
client.print("Content-Type: application/x-www-form-urlencoded\r\n");
client.print("Content-Length: ");
client.print(postStr.length());
client.print("\r\n\r\n");
client.print(postStr);
```

La trasmissione che arriva sul server ha l'aspetto seguente (dopo formattazione da parte della logica del sito):

```
Tipo: Post
Array
(
    [codice] => 123456
    [temp] => 21.57
    [hum] => 35.00
    [pre] => 992.13
    [voc] => 0.00
    [cox] => 0.00
    [pm1] => 21.00
    [pm25] => 31.00
    [pm10] => 33.00
    [mac] => 30:AE:A4:CC:4F:74
    [data] => 2019-01-29
    [ora] => 17:28:33
)
```

In particolare, per i campi CODICE e MAC:

CODICE: è un campo di 6 cifre, univoco, che è associato a ciascuna centralina. Viene rilasciato dall'Associazione ed è legata alla centralina. Il codice NON è scelto dall'utente ma è inserito nella scheda SD di configurazione. E' necessario per far visualizzare i dati della centralina sul sito web.

MAC: è un campo di 17 caratteri alfanumerici, che corrisponde al MAC ADDRESS della centralina ed è anch'esso univoco. Non è settabile da nessuno, ma è visualizzato dalla centralina dopo la prima trasmissione. Nel sito web viene utilizzato questo campo per verificare che i dati trasmessi siano corretti e sempre in arrivo dalla centralina corretta. Nell'ipotesi che dovesse essere necessario sostituire il processore nella centralina (per un difetto ad esempio) questo campo cambierebbe e sarebbe da modificare anche nella configurazione del sito web. E' necessario per far visualizzare i dati della centralina sul sito web.

2.5.2. Funzionamento della scheda di memoria

All'interno della centralina è presente, come già accennato, una scheda di memoria micro SD. Al suo interno sono presenti 2 file necessari al funzionamento della centralina stessa.

L'assenza della scheda di memoria, oppure qualche problema di lettura della stessa, oppure ancora la mancanza dei file indicati, impedisce alla centralina di trasmettere.

Ecco i file:

- ✓ LOG_ILARIA.txt
- ✓ RETE.txt

Il file **LOG_ILARIA.txt** è un semplice file di LOG dei dati .trasmessi. E' compatibile con excel, e si apre con un convertitore CSV (comma separated values, vedere qui: https://it.wikipedia.org/wiki/Comma-separated_values).

Se non è presente, il sistema lo crea al primo avvio e continua a scrivere i dati trasmessi ad ogni invio.

Con 1 trasmissione ogni 30 minuti, si stima che una card di 8GB possa contenere decine di anni di registrazioni. In pratica diventerà obsoleta la scheda prima di riempirsi completamente.

Il file **RETE.txt** è un file di testo che deve essere editato dall'utente della centralina, che deve inserire le coordinate di accesso della rete WIFI su cui la centralina si appoggerà per il normale funzionamento.

In pratica, il file è così composto:

#ssid=nomedellarete#pwd=passworddellarete#codice=123456#splash=Wondermade*

o, più leggibile:

#ssid=nomedellarete
#pwd=passworddellarete
#codice=123456
#splash=Wondermade
*

Dove:

- ✓ Nomedellarete – inserire qui il nome della rete WIFI (il famoso SSID)
- ✓ Passworddellarete – inserire qui la password della rete WIFI
- ✓ 123456 – inserire qui il codice univoco associato alla vostra centralina
- ✓ Wondermade – un testo (max 12 caratteri) che si vuole venga visualizzato a schermo

E' importante non modificare il costrutto del file: i caratteri "#" vanno lasciati dove sono, non cancellare il carattere "*" alla fine del file e non modificare in alcun modo altre caratteristiche del file, oppure la vostra centralina potrebbe non funzionare più.

ATTENZIONE!

Le schede micro SD sono delicate e spesso risultano NON FORMATTATE. Prima di buttarle perché difettose è possibile provare una FORMATTAZIONE COMPLETA (**NON** FORMATTAZIONE VELOCE).
Le schede necessitano di essere formattate secondo il **formato FAT 32.**

Un SW gratuito per la formattazione completa delle schede è disponibile, in rete, qui:

https://www.sdcard.org/downloads/formatter/

(sito ufficiale della SD Association Inc.)

3. Montaggio della centralina

Il montaggio della centralina si articola in 4 fasi distinte:

- ✓ Preparazione dei collegamenti esterni con sensori e display
- ✓ Preparazione della scatola contenitore
- ✓ Montaggio dei componenti e connettori sul PCB
- ✓ Collegamento finale e messa in funzione

Alcune fasi del montaggio sono delicate, e richiedono una certa manualità (soprattutto la fase di saldatura). Chi è poco pratico potrebbe trovare delle iniziali difficoltà, oppure causare errati funzionamenti a causa di errate saldature. Il problema più comune è infatti quello di saldare male e creare ponti con stagno oppure corto-circuiti a causa di saldature "sporche" che toccano parti che dovrebbero essere isolate. Un altro problema da non sottovalutare è relativo alle "saldature fredde", ossia saldature che sembrano corrette ma che sono state eseguite senza che lo stagno si sia amalgamato completamente, causato giunzioni non corrette o falsi contatti, entrambe cose difficili da trovare in fase di debug successivo e che portano a funzionamento irregolare della centralina anche aleatorio.

Suggeriamo, a chi è alle prime armi, qualche sessione di pratica per imparare ad utilizzare gli strumenti necessari, per evitare problemi, velocizzare i tempi di montaggio e rendere l'operazione meno "noiosa e stressante".

3.1. Elenco degli attrezzi necessari

Per completare il montaggio della centralina servono pochi attrezzi, e soprattutto ogni fase di lavoro richiede alcuni attrezzi dedicati, che vengono elencati di seguito. Chi ha qualche esperienza di montaggi elettronici troverà sommaria questa sezione, forse incompleta, oppure preferirà utilizzare altri attrezzi in base alle sue capacità ed esperienza specifica.

Elenchiamo di seguito gli attrezzi necessari per ciascuna fase di lavoro:

- ✓ <u>Preparazione dei collegamenti esterni con sensori e display</u>
  - Saldatore 30w pinta media
  - Stagno 0.6-0.8mm con anima di flussante
  - Supporto saldatore
  - Spugnetta per pulire punta saldatore
  - Spelafili
  - Terza mano (non strettamente necessaria, utile per principianti)

- ✓ <u>Preparazione della scatola contenitore</u>
  - Punta da trapano D6mm
  - Punta da trapano D8mm
  - Punta da trapano D30-32mm (oppure svasatore) per foro tappi
  - Pinzetta per facilitare i serraggi

- ✓ <u>Montaggio dei componenti e connettori sul PCB</u>
  - Saldatore 30w pinta media
  - Stagno 0.6-0.8mm con anima di flussante
  - Supporto saldatore
  - Spugnetta per pulire punta saldatore
  - Spelafili

- Terza mano (non strettamente necessaria, utile per principianti)
- Tester con funzione “prova continuità” per verificare l’assenza di ponticelli e falsi contatti
- Cacciavite piccolo piatto per settaggio modulo step-down
- Tester con funzione “misura tensione” per verifiche tensione in uscita da step-down

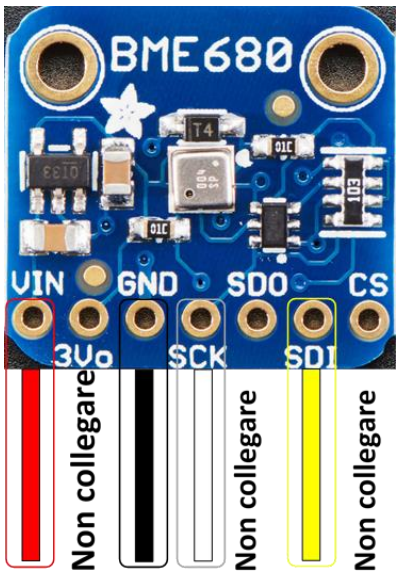
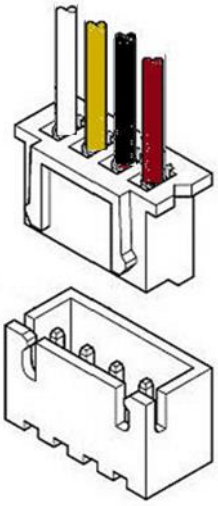
✓ <u>Collegamento finale e messa in funzione</u>

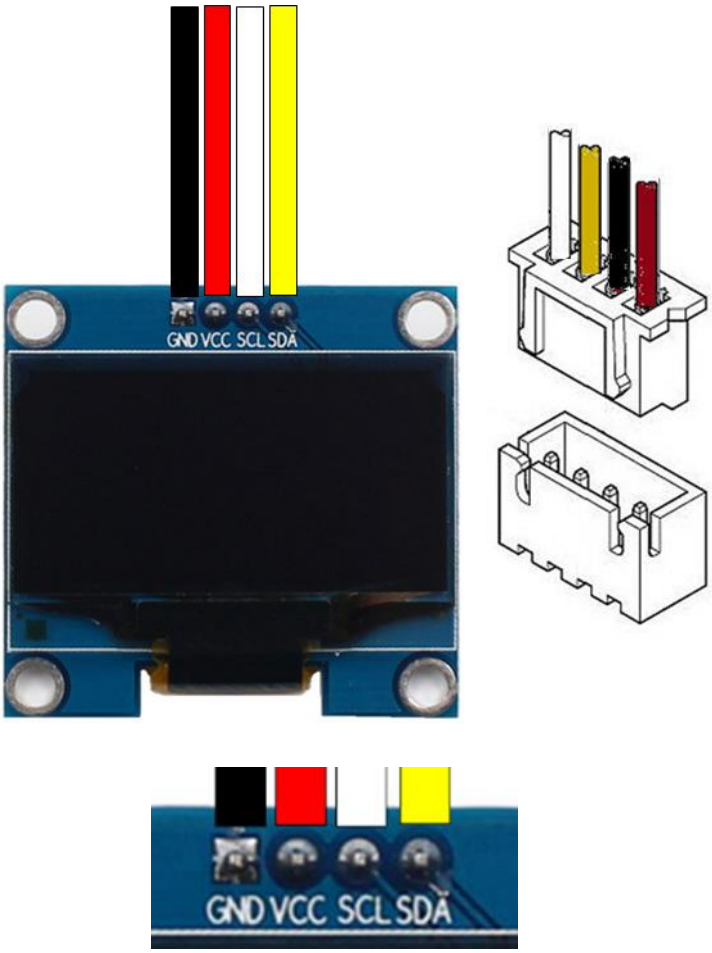
- Pinza per inserimento connettori
- Colla a caldo per fissaggi
- Cacciavite medio a croce
- Tester con funzione “prova continuità” per verificare l’assenza di ponticelli e falsi contatti
- Tester con funzione “misura tensione” e “misura resistenza” per verifiche
- Cavo micro-USB per prima messa in funzione

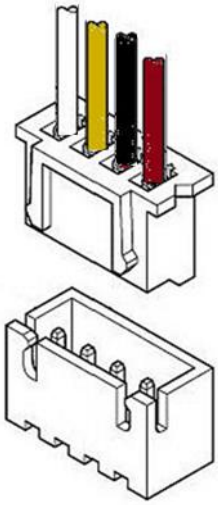
3.2. Preparazione dei collegamenti

Come prima fase è necessario collegare tutti i moduli esterni al relativo cavetto, in modo da poter essere in seguito connessi e disconnessi con facilità dalla scheda.

Di seguito si riporta, in formato tabellare, la spiegazione dei collegamenti necessari.

1	**BME680** Collegare V+ all'alimentazione (ROSSO) Collegare GND alla massa (NERO) Collega il pin SCK al filo SCL (clock I2C) (BIANCO) Collega il pin SDI al filo SDA (data I2C) (GIALLO)	  

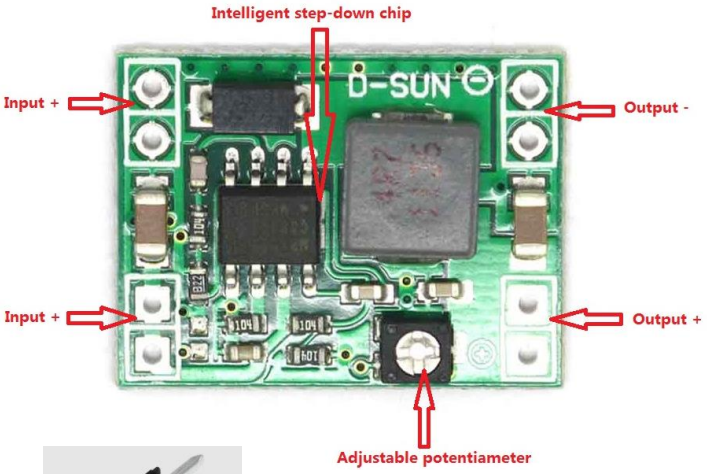
2	**OLED display** Collegare V+ all'alimentazione (ROSSO) Collegare GND alla massa (NERO) Collega il pin SCL al filo SCL (clock I2C) (BIANCO) Collega il pin SDA al filo SDA (data I2C) (GIALLO) ATTENZIONE: il display in vostro possesso potrebbe avere una diversa disposizione dei piedini. Seguire l'abbinamento tra colore e nome pin!	

3	**MQ (Sensore COx)** Collegare **V+** all'alimentazione VDD(o VCC) (**ROSSO**) Collegare **GND** alla massa (**NERO**) Collega il pin **AO** al filo **GIALLO** Collega il pin **DO** al filo **BIANCO**	  

4 STEP-UP MP1584

Collegare 4 connettori da 2, strip maschio, a ciascuno dei 4 lati, come mostrato in foto.

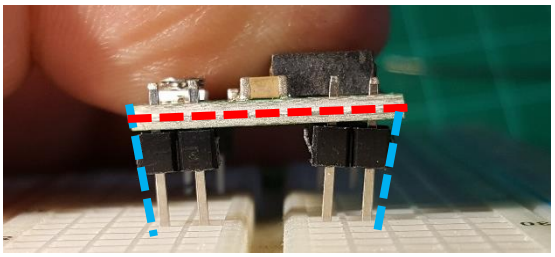
Ciascuna delle 4 coppie è relativo ad un terminale unico, quindi i 2 pin possono essere anche cortocircuitati tra loro.





4x

ATTENZIONE!!: i componenti arrivati dalla Cina in questa spedizione non fittano perfettamente sulla basetta Potrebbe capitare ancora…



Consiglio: è necessario piegare leggermente in dentro i piedini relativi e poi procedere a saldatura. aiutarsi con la basetta della centralina per ottenere la posizione, e poi saldare!

Vedere foto per chiarimenti!!

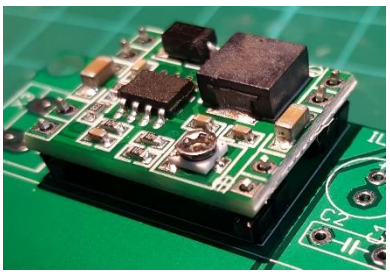
1) Inserire i 4 connettori nella basetta

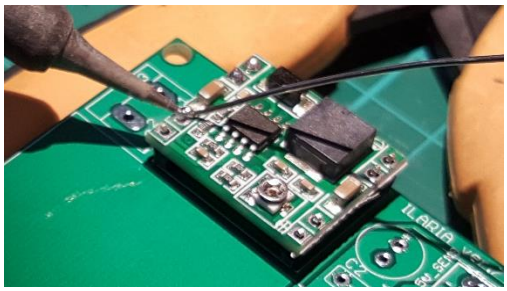
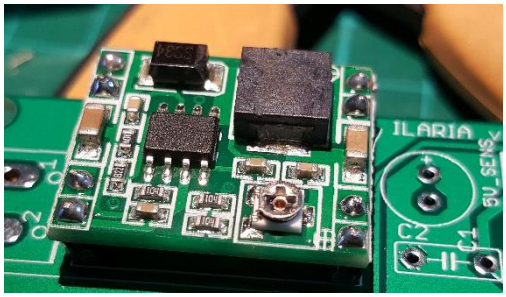
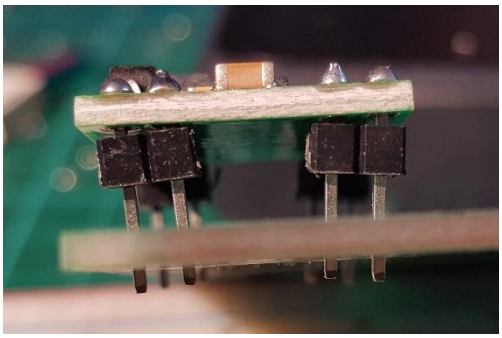
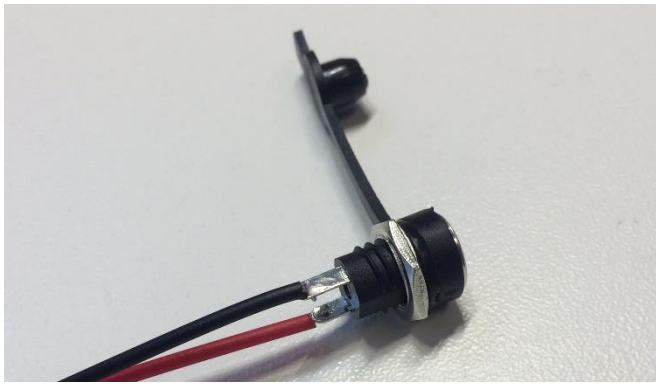
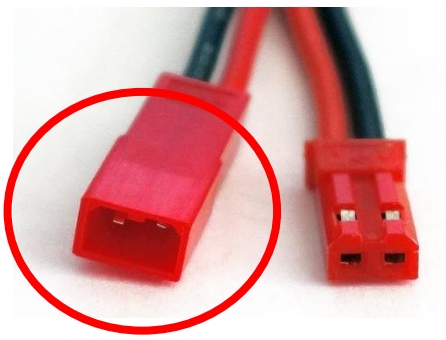


2) Calzare la scheda MP1584 facendo combaciare i pin sottostanti con i fori superiori



3) foto evidenzia un corretto inserimento della scheda STEP UP MP1584



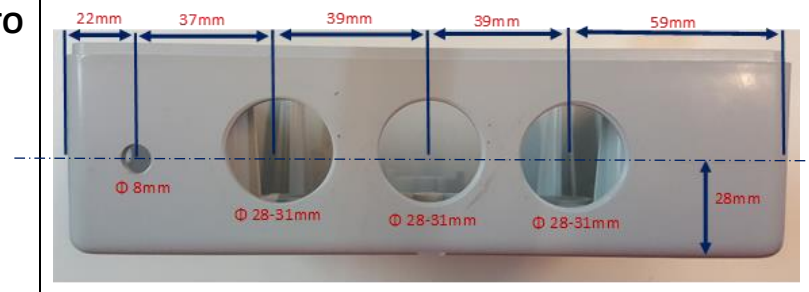
		4) procedere alla saldatura  5) come appare l'elemento a saldatura ultimata  6) evidenza della lieve deformazione dei piedinii necessaria per consentire un corretto inserimento 
5	**CONNETTORE ALIMENTAZIONE** Collegare **il filo rosso al piedino CORTO** Collegare **il filo nero al piedino LUNGO** Fare attenzione a non creare cortocircuiti tra i 2 pin. Se si preferisce, è possibile isolare i 2 terminali con del nastro adesivo nero oppure con della colla a caldo. **ATTENZIONE: USARE IL FILO con connettore "MASCHIO" per COLLEGARE il connettore di alimentazione!**	 

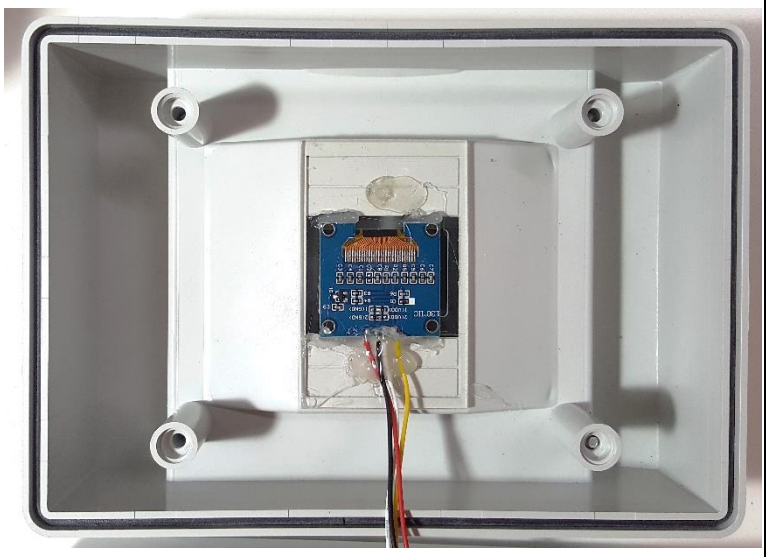
3.3. Preparazione della scatola che conterrà la parte elettronica

La scatola, nuova, si presenta in questo modo:



È necessario preparare la scatola, effettuando i seguenti fori:

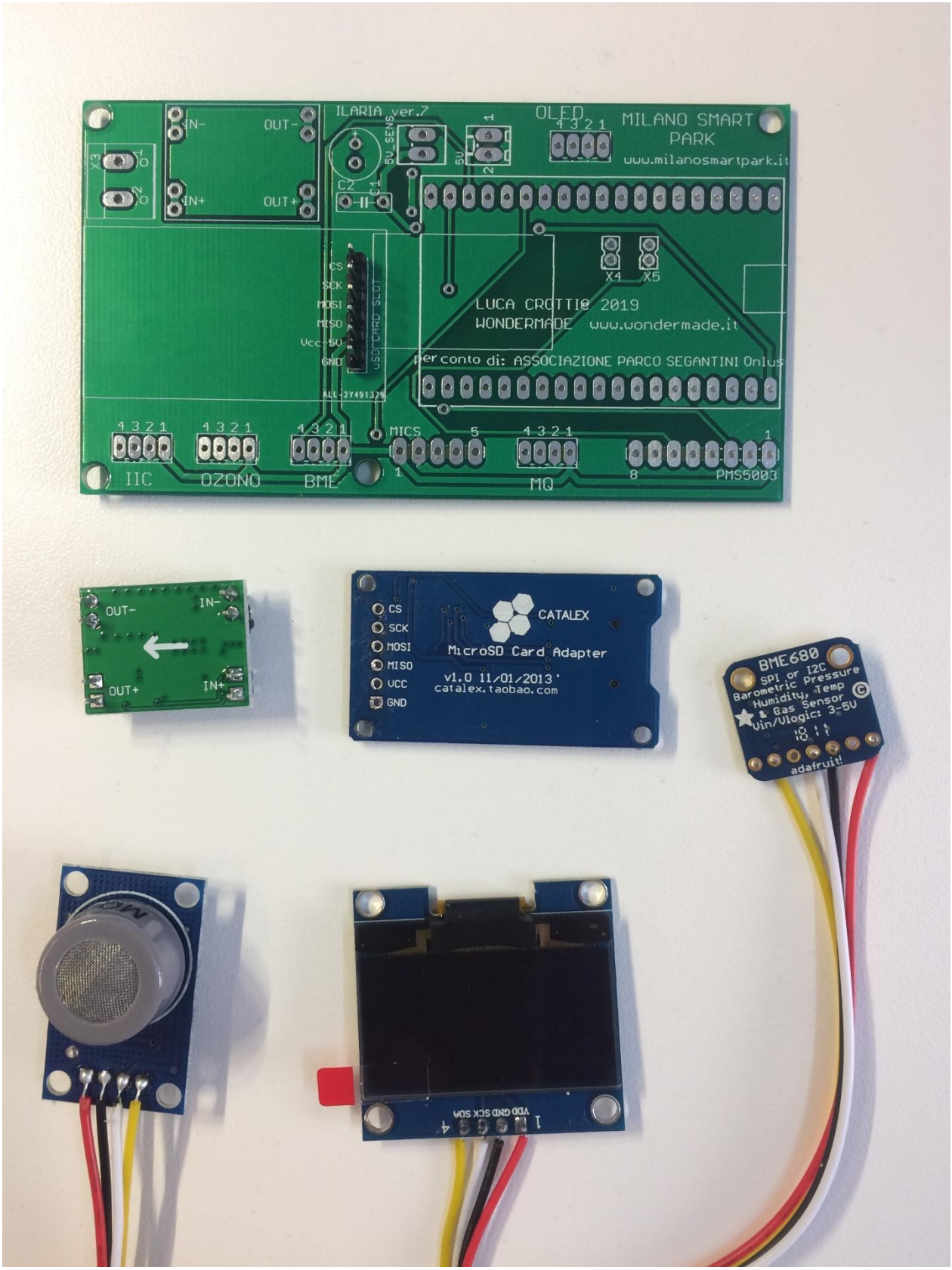
1	**PREPARAZIONE FORI SU LATO LUNGO** Questi fori servono per il connettore di alimentazione (8mm) e per i tappi che conterranno i sensori (D28 oppure 30mm in funzione del diametro dei tappi). Per i tappi, considerate un foro di mezzo millimetro più largo per consentire un facile inserimento.	 

2	**PREPARAZIONE FORO per ANTENNA** Questo foro serve per l'antenna. Guardando la centralina con i tappi rivolti verso il basso, il foro (diametro 6mm) va fatto in alto a sinistra del lato corto. Viene anche indicato come montare il cavo d'antenna, la barra grigia verticale rappresenta la plastica.	 
3	**Montaggio del display** Il display deve essere montato nella parte superiore con la parte delle saldature dei fili rivolte verso l'alto, ossia dal lato OPPOSTO al lato di apertura. È necessario realizzare un piccolo supporto per adattare l'apertura al formato del display. Qui un esempio, in cui il fissaggio è facilmente realizzabile con qualche punto di colla a caldo.	 

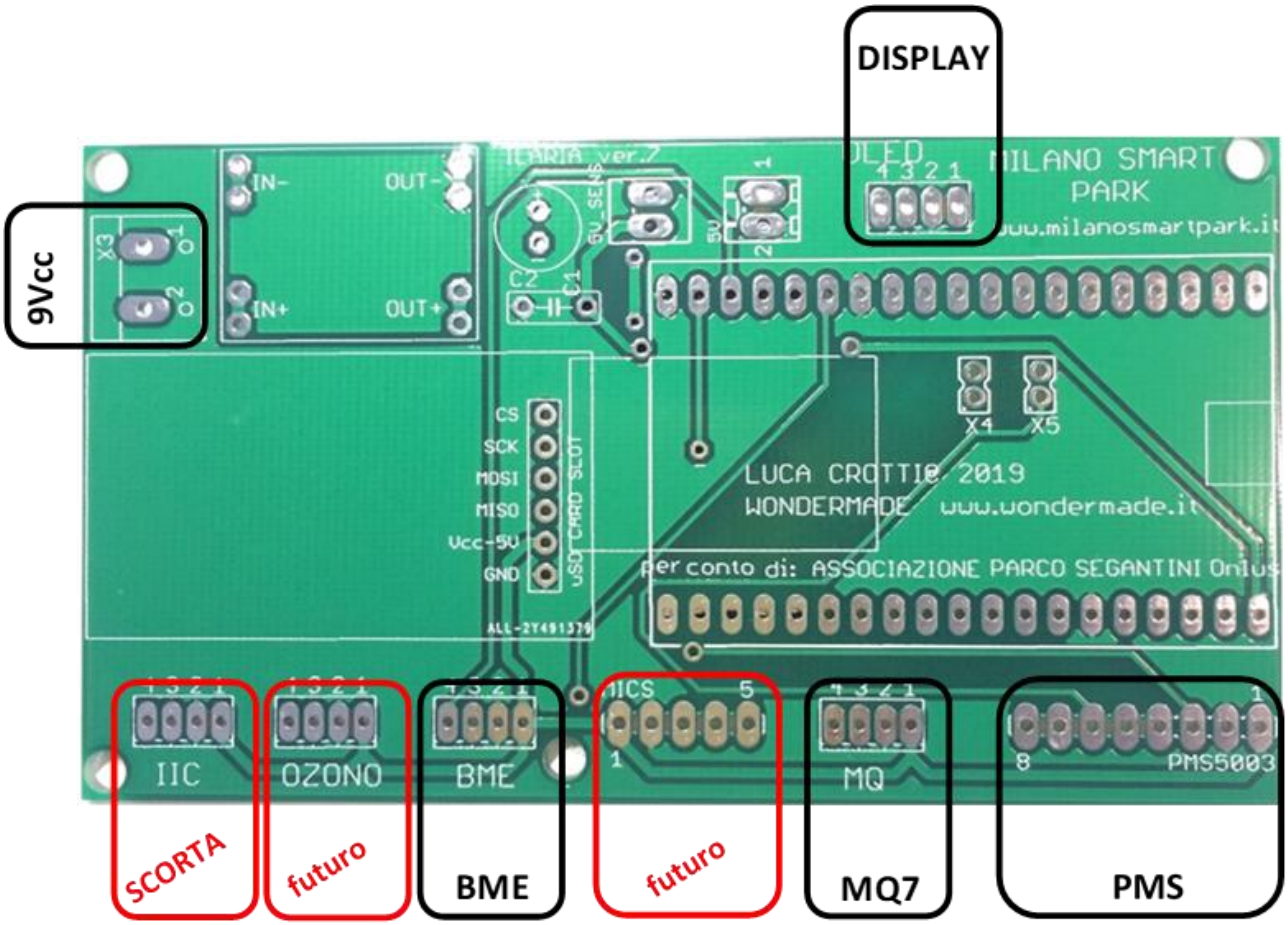
4	**SCATOLA COMPLETA**	

3.4. Montaggio dei componenti sulla scheda

FOTO DEI COMPONENTI PREPARATI E DELLA BASETTA



DISPOSIZIONE DEI COLLEGAMENTI SULLA BASETTA:

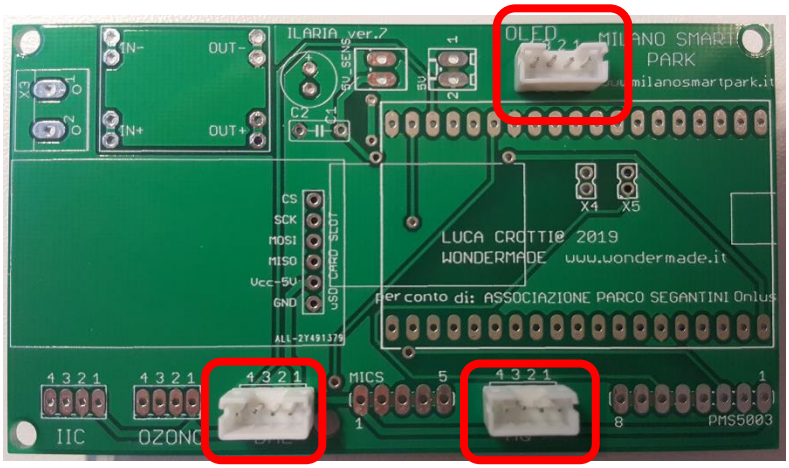
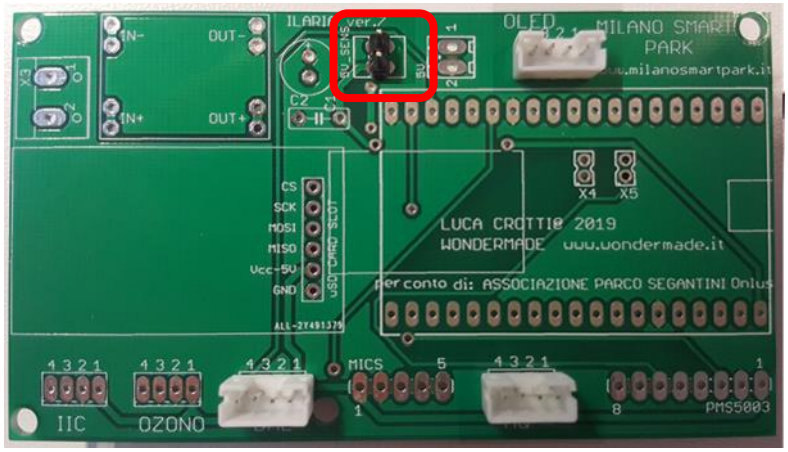
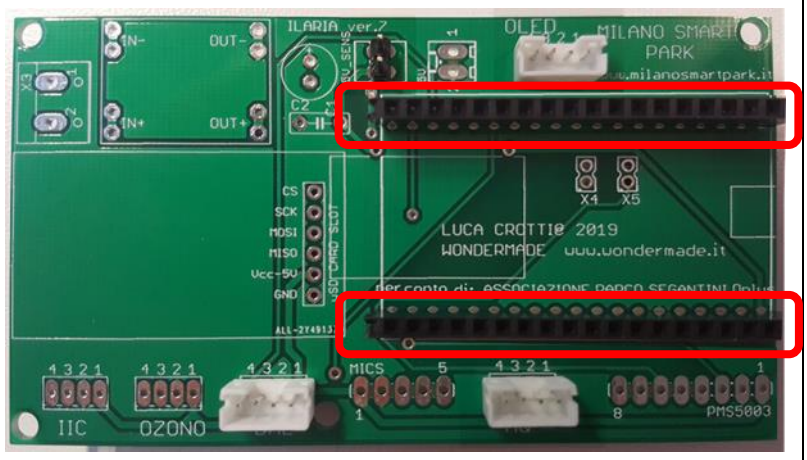


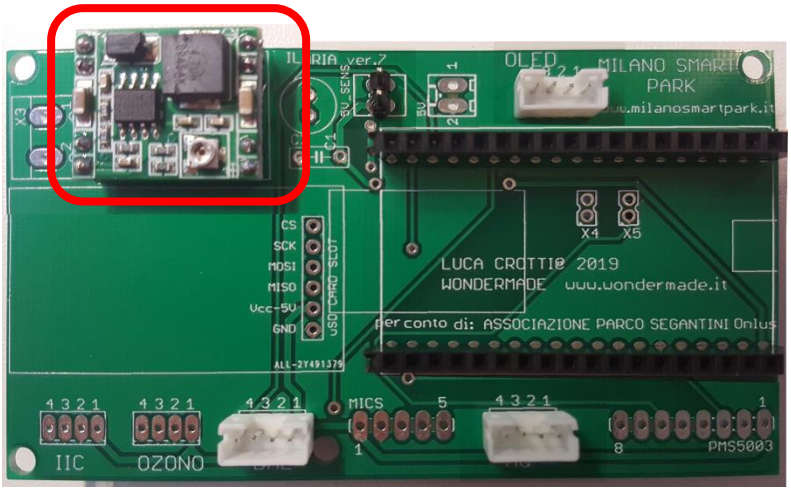
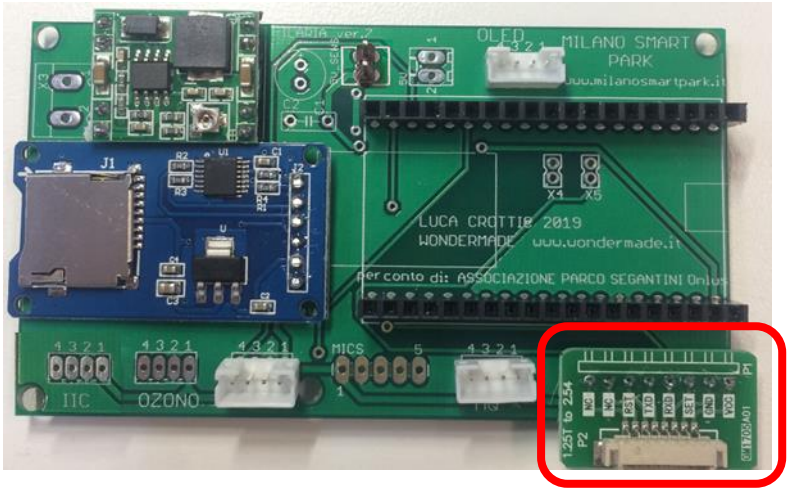
Strumenti necessari:

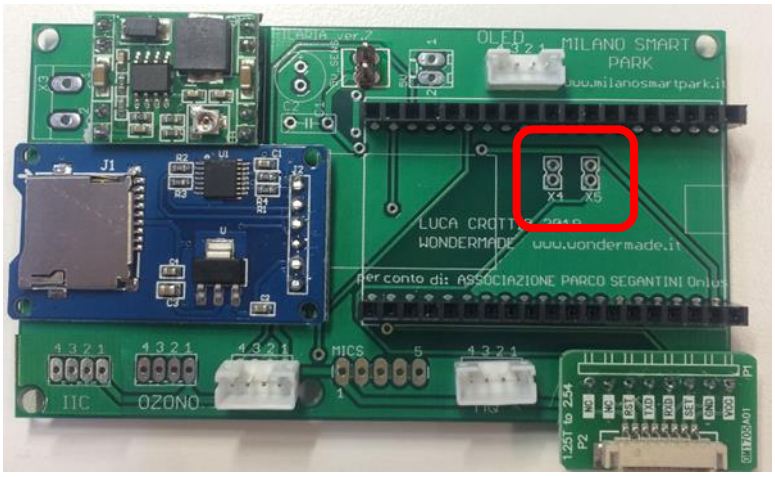
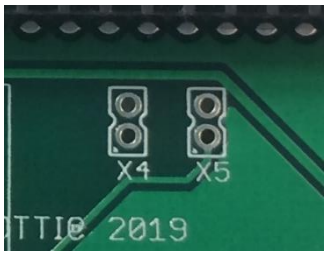
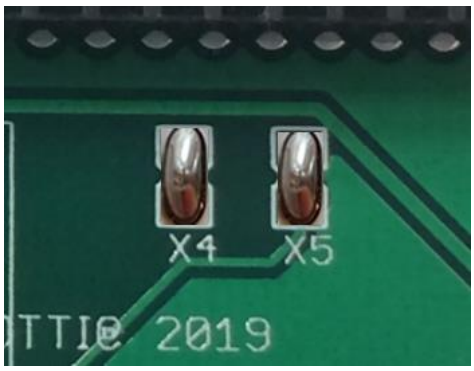
- ✓ Saldatore
- ✓ Pinzette
- ✓ Stagno
- ✓ Spugnetta
- ✓ Porta-saldatore

-

MONTAGGIO COMPONENTI:

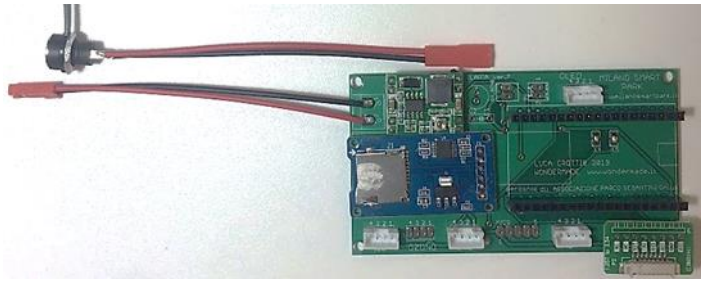
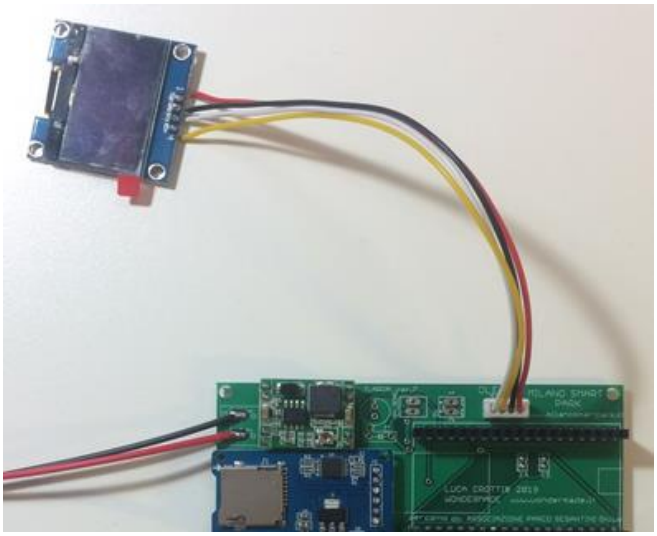
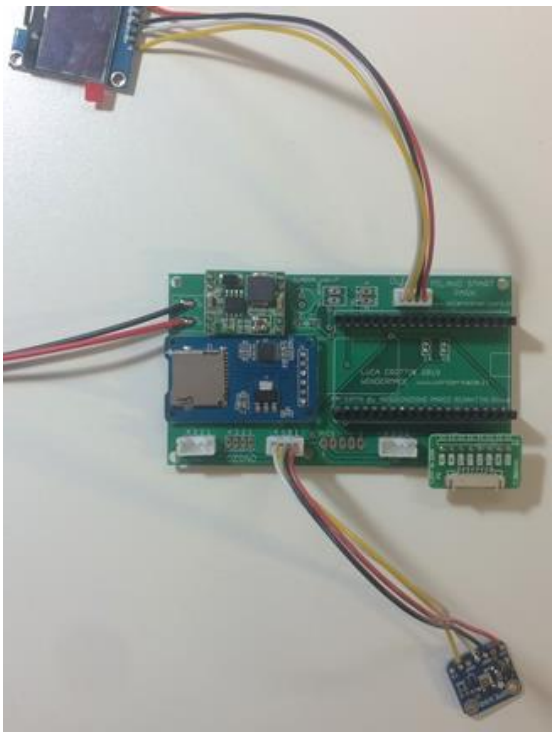
1	**connettori 4 poli JST per sensori** saldare i connettori come in figura, con lo spacco rivolto verso l'alto	
3	**Connettore 5V_SENS** Preparare un connettore 2 pin da strip maschio e saldarlo su scheda come in figura	
3	**Connettore 19+19 pin** Preparare n. 2 connettori 19 pin da strip FEMMINA e saldarli su scheda come in figura. <u>Questo connettore va a formare lo zoccolo di montaggio del microcontrollore ESP32</u>	

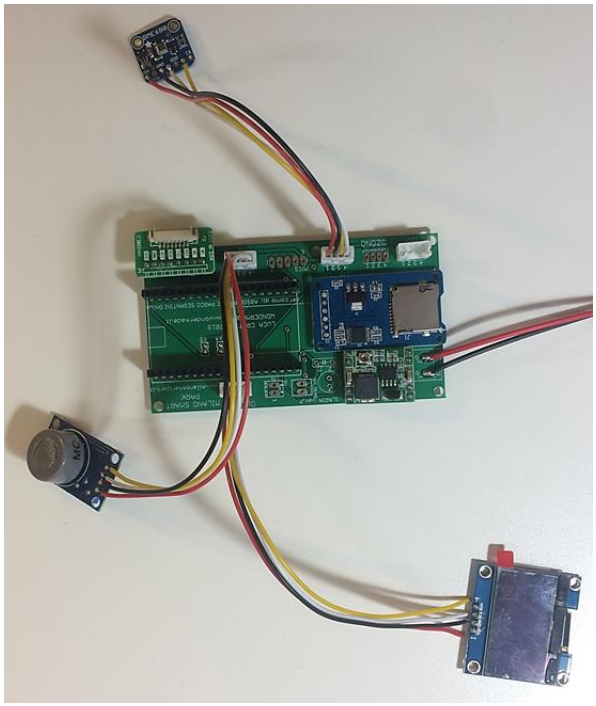
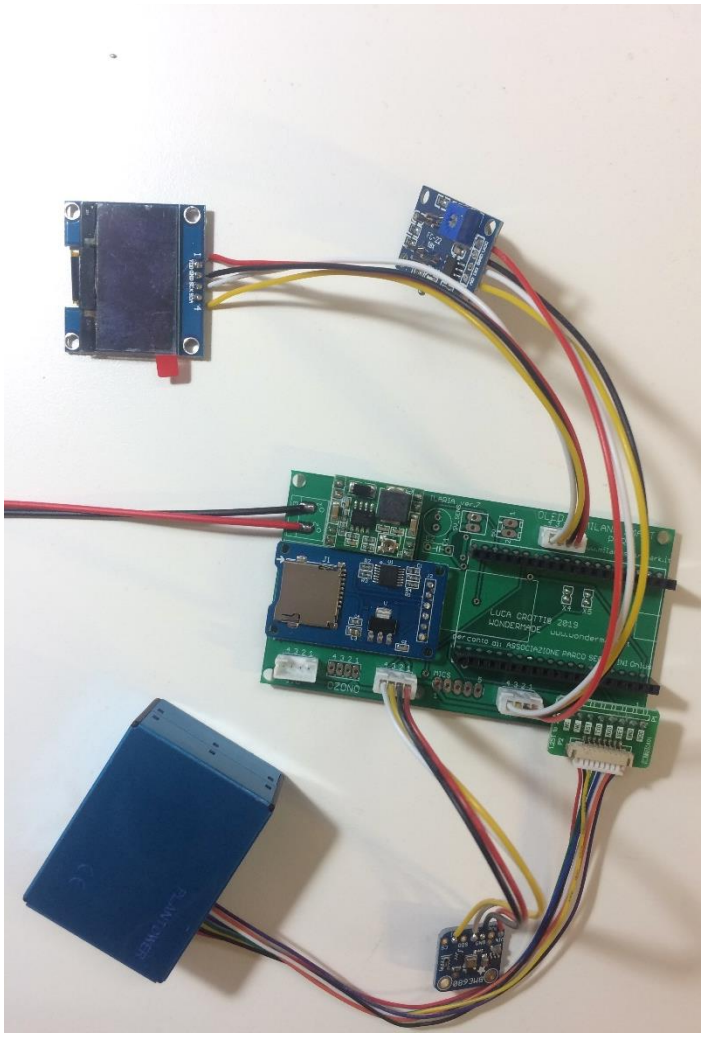
4	**MODULO STEP UP** Saldare il modulo step-up precedentemente preparato. Attenzione a rispettare la direzione (come in figura). La parte IN è sulla sinistra, la parte OUT è sulla destra.	
5	**MODULO SD-CARD** Saldare il modulo SDCARD come in foto. Nel caso in cui i piedini del modulo arrivino non saldati, è necessario saldare un connettore strip maschio da 6 poli; se il connettore saldato è con PIN a 90°, è necessario fare qualche piccolo aggiustamento raddrizzando i pin (uno alla volta, è facile e veloce).	
6	**Connettore per PMS5003** Il connettore in foto serve da riduttore, per rendere compatibile il microconnettore presente sul sensore di particolato con basette i cui fori sono spaziati a passo 2.54mm. E' semplicemente necessario saldare gli 8 pin come in foto, facendo attenzione al corretto verso.	

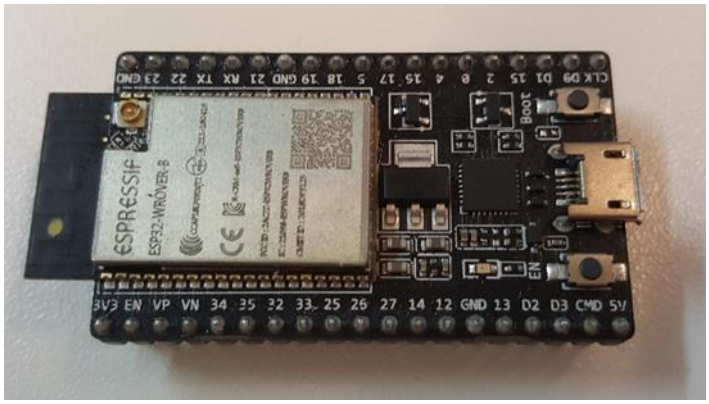
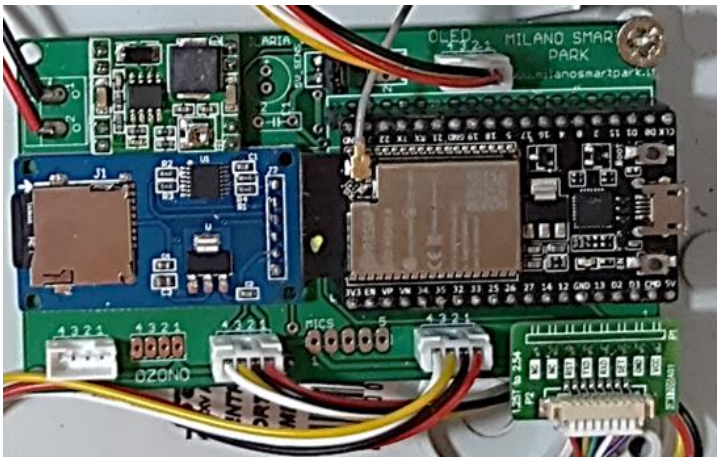
7	**COLLEGARE PONTICELLI X4 e X5** I 2 ponticelli sono da collegare con una goccia di stagno	  

3.5. Collegamento dei vari sensori alla scheda

Il collegamento dei sensori alla scheda è a questo punto semplicissimo.

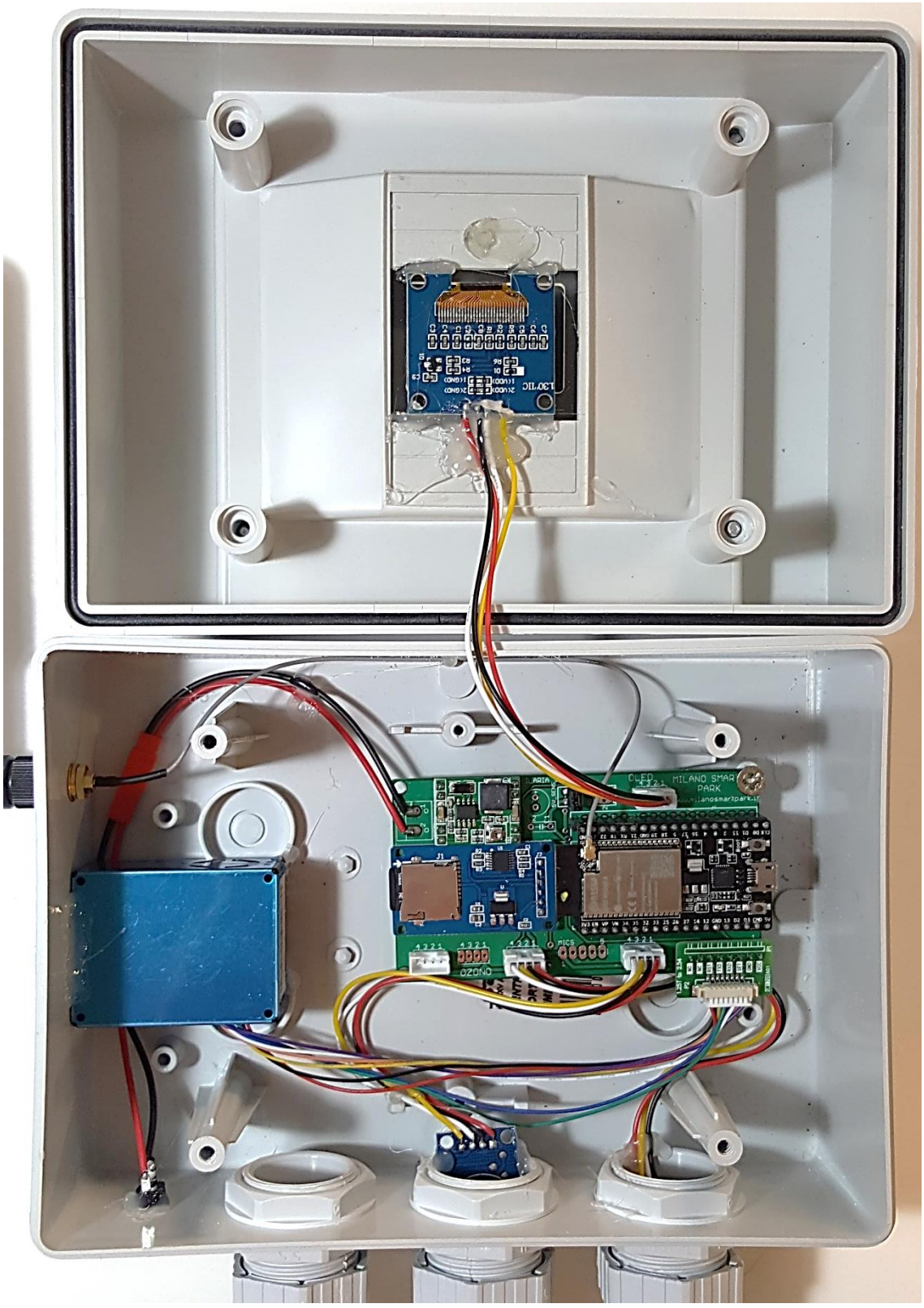
1	**CONNETTORE DI ALIMENTAZIONE**	
2	**DISPLAY OLED**	
3	**SENSORE BME680**	

4	**SENSORE COx**	
5	**SENSORE PMS5003** Il connettore è delicato, fare attenzione durante l'inserimento. Per aiutarvi, il colore NERO è all'estrema destra della scheda, in basso.	

6	**ESP32** L'ultima operazione da eseguire è quella di inserire il microcontrollore, ESP32 WROVER, e collegare il piccolo connettore d'antenna nell'apposito connettore presente sulla scheda. Attenzione, il connettore è delicato.	   

Mano a mano che i componenti sono collegati, è anche opportuno posizionarli nella scatola, in modo da realizzare un cablaggio pulito e ordinato ed evitare la remota possibilità che i connettori si distacchino.

In foto si mostra il risultato finale:



Si notino alcuni particolari:

- La scheda (PCB) della centralina è fissata con una vite in uno dei fori, in alto a destra
- Il cavo di alimentazione, una volta collegato, passa SOTTO l'involucro del sensore PMS
- Il sensore PMS risulta appoggiato al bordo sinistro della scatola, e fissato con un punto di colla a caldo
- Il sensore di Cox (sensore MQ7) ed il sensore BME si incrociano: il connettore del sensore BME è sulla sinistra, ma il sensore viene posizionato nel tappo di destra; allo stesso modo, il sensore MQ7 è inserito nel tappo centrale, mentre il suo connettore è più a destra. Questo accorgimento consente di distendere meglio i cavi e garantire ordine all'interno della scatola
- il tappo di sinistra è lasciato vuoto per garantire un corretto flusso d'aria verso il PMS ed anche per sviluppi futuri
- il cavo di antenna scorre lungo la parete superiore ed è fermato in posizione da un punto di colla a caldo
- i sensori all'interno dei tappi sono fermati con della colla a caldo, per evitare movimenti accidentali durante le operazioni di installazione.

ATTENZIONE!

Ricordarsi di chiudere il ponticello presente sul jumper "5V_SENS" (perché in questo modo si alimentano i sensori) SOLO DOPO AVER CONTROLLATO, A MONTAGGIO ULTIMATO, CHE LE TENSIONI PRESENTI SUI CAVI DEI SENSORI SIANO EFFETTIVAMENTE DI 5V (in realtà si dovrebbe leggere un valore di 5,1-5,2V, e questo è normale).

3.6. collegamento dell'alimentazione e primo avvio

Ecco le operazioni da effettuare prima del primo avvio, per COLLAUDARE il lavoro fatto:

1. scollegare tutti i sensori e staccare il display, e non connettere il microcontrollore ESP32
2. scollegare il ponticello 5V_SENS in modo da non connettere l'alimentazione con alcun sensore
3. collegare l'alimentatore 9V al connettore.
4. Verificare che sul connettore X3 sia presente una tensione di 9Vcc.
5. Verificare che sul modulo STEP UP lato INPUT (lato sinistro) sia presente una tensione di 9Vcc: come mostrato più in alto nel manuale, in alto a sinistra del modulo è presente IN negativo, in basso a sinistra è presente IN positivo
6. Controllare la tensione in uscita dal modulo STEP UP (anche qui, in alto a DESTRA del modulo è presente OUT negativo, in basso a DESTRA è presente OUT positivo). Se la tensione letta è 5,1 – 5,2V tutto è ok e passare al punto successivo. In caso contrario è necessario regolare la tensione agendo sulla vitina presente sul modulo. Rotazione antioraria per DIMINUIRE la tensione (e viceversa)
7. Una volta che la tensione risulta corretta, è possibile collegare il ponticello 5V_SENS (chiudere il ponticello con apposito morsetto in plastica, oppure con una saldatura). A questo punto è necessario verificare che su tutti i connettori dei sensori e della scheda sia presente una tensione di 5V tra il pin 1 (corrispondente al filo ROSSO) ed il pin 2 (corrispondente al filo nero)
8. Sul connettore del PMS5003 i fili sono invertiti, all'estrema destra non è presente il positivo ma GND. Invertire i puntalini (oppure si leggerà una tensione negativa…)
9. Se tutto è OK, è possibile togliere l'alimentazione e RICOLLEGARE TUTTO NUOVAMENTE
10. A questo punto è necessario formattare la scheda MICRO SD (formattazione FAT32) e copiare all'interno il file RETE.TXT (la cui spiegazione è stata fatta all'inizio di questo manuale). Non è necessario creare alcuna cartella, semplicemente copiare in ROOT (SD:\).
11. A questo punto è possibile ricollegare l'alimentazione e inserire il jack di alimentazione all'interno dell'apposito connettore.
12. La scheda dovrebbe accendersi e sul display dovrebbe comparire la seguente schermata:



4. USO DELLA CENTRALINA

4.1. Configurazione parametri

All'interno della centralina è presente, come già accennato, una scheda di memoria micro SD. Al suo interno sono presenti 2 file necessari al funzionamento della centralina stessa.

L'assenza della scheda di memoria, oppure qualche problema di lettura della stessa, oppure ancora la mancanza dei file indicati, impedisce alla centralina di trasmettere.

Ecco i file:

- ✓ LOG_ILARIA.txt
- ✓ RETE.txt

Il file **LOG_ILARIA.txt** è un semplice file di LOG dei dati .trasmessi. E' compatibile con excel, e si apre con un convertitore CSV (comma separated values, vedere qui: https://it.wikipedia.org/wiki/Comma-separated_values).

Se non è presente, il sistema lo crea al primo avvio e continua a scrivere i dati trasmessi ad ogni invio.

Con 1 trasmissione ogni 30 minuti, si stima che una card di 8GB possa contenere decine di anni di registrazioni. In pratica diventerà obsoleta la scheda prima di riempirsi completamente.

Il file **RETE.txt** è un file di testo che deve essere editato dall'utente della centralina, che deve inserire le coordinate di accesso della rete WIFI su cui la centralina si appoggerà per il normale funzionamento.

In pratica, il file è così composto:

#ssid=nomedellarete#pwd=passworddellarete#codice=123456#splash=Wondermade*

o, più leggibile:

#ssid=nomedellarete
#pwd=passworddellarete
#codice=123456
#splash=Wondermade
*

Dove:

- ✓ Nomedellarete – inserire qui il nome della rete WIFI (il famoso SSID)
- ✓ Passworddellarete – inserire qui la password della rete WIFI
- ✓ 123456 – inserire qui il codice univoco associato alla vostra centralina
- ✓ Wondermade – un testo (max 12 caratteri) che si vuole venga visualizzato a schermo

E' importante non modificare il costrutto del file: i caratteri "#" vanno lasciati dove sono, non cancellare il carattere "*" alla fine del file e non modificare in alcun modo altre caratteristiche del file, oppure la vostra centralina potrebbe non funzionare più.

4.2. Guida alla lettura delle informazioni a schermo

Di seguito si riportano le informazioni a schermo, che rappresentano anche le fasi di funzionamento della centralina. Una descrizione e spiegazione passo-passo di cosa avviene consentirà di capire esattamente cosa succede, quali sono le fasi di misura etc.

Oltre alla conoscenza di quello che avviene, è importante analizzare ciò che viene scritto sul display anche in caso di anomalie e relativa diagnosi di primo livello (ossia per capire quale sia l'elemento che ha un funzionamento errato per provare a sostituirlo).

Con ordine:

all'accensione viene visualizzata la schermata di avvio (cosiddetta SPLASH SCREEN) in cui sono riportare alcune informazioni:

MILANO SMART PARK	titolo	non modificabile
SW VER. 21	versione SW	modificabile su agg.to SW
Luca Crotti – 2019	autore	non modificabile
casalegno	etichetta personalizzabile	modificabile su SD card (splash)

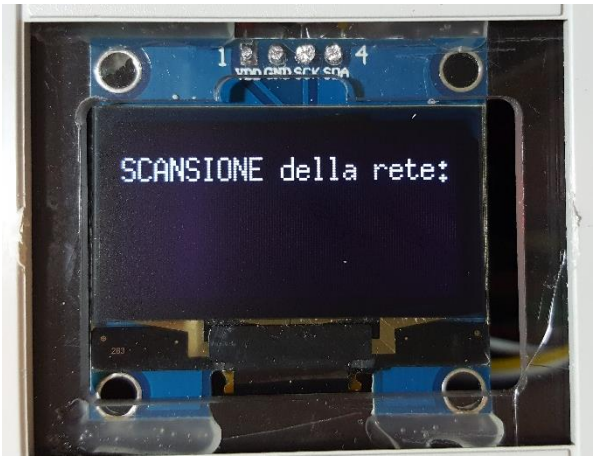


➔ in caso di mancata lettura della SD CARD (oppure mancanza scheda, scheda difettosa, oppure FILE RETE.txt corrotto, mancante o non leggibile) la centralina avvisa che non ci si potrà connettere al WEB (poiché la scheda SD contiene, tra gli altri, i dati di connessione alla rete wifi SSID e PWD)

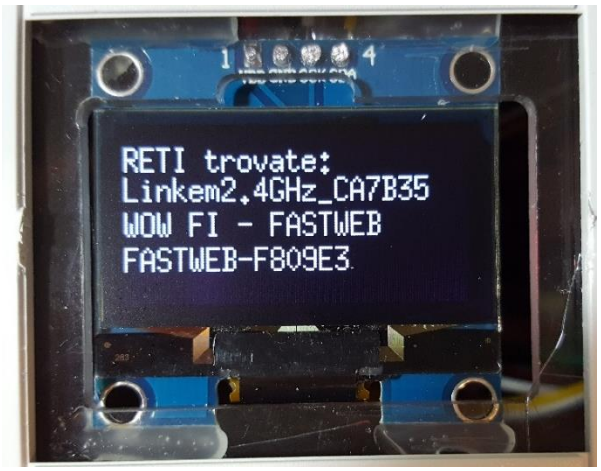


se invece tutto procede bene si ottiene la seguente sequenza di schermate, che di fatto:

- Analizza le reti WI-FI che sono raggiungibili
- Elenca le prime reti, per potenza del segnale
- Conferma la connessione alla rete indicata











➔ Fase importante perché se la rete a cui ci si vuole collegare non compare tra le prime 3-4 reti, è verosimile che la centralina farà fatica a collegarsi. In questo caso è necessario AVVICINARSI in una zona con maggiore copertura di rete

In caso di difficoltà a connettersi, lo schermo visualizza per qualche momento dei PUNTINI… in attesa della riconnessione. ogni 5 “PUNTINI” la scheda è in attesa, poi fa cadere la connessione e prova da capo la procedura.

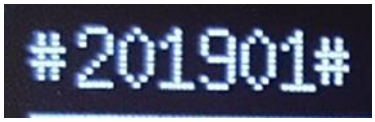
In caso di connessione alla rete WIFI indicata, dopo qualche istante (4-5 secondi) dalla connessione avvenuta viene presentato lo schermo con la lettura di tutti i sensori.

Nelle 2 immagini che seguono si vede come una icona (sulla sinistra) a forma vagamente di freccia appuntita sia presente per poi scomparire. Quella freccia indica il momento in cui la centralina sta inviando in rete i dati dei sensori.



Ecco di seguito la spiegazione del significato delle 4 icone presenti e del codice numerico:





codice della centralina, presente su SD, importante perché collegato al sito WEB



simbolo indicante la connessione avvenuta con un server NTP, in grado di trasmettere alla centralina data e ora precise (servizio presente sul WEB e gratuito)



indicante la corretta scrittura dei paramenti letti su SD card (generazione file LOG avvenuta con successo)



freccia indicante il momento in cui la centralina sta inviando in rete i dati dei sensori.



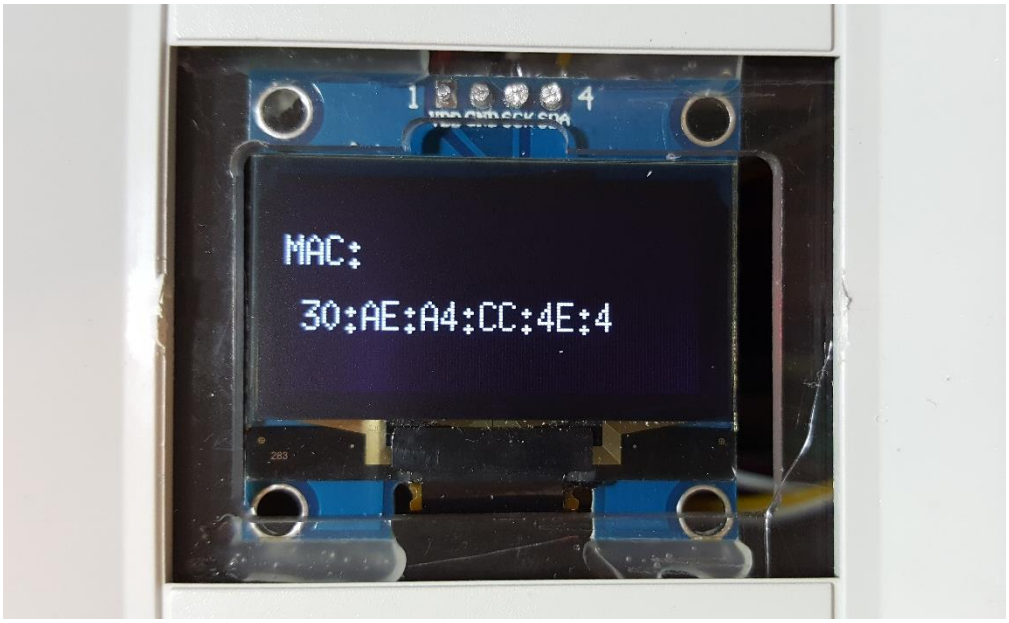
indicante la connessione avvenuta con successo verso il server su cui è necessario trasmettere i dati (sito milanosmartpark)

alla fine della trasmissione, la freccia scompare ma la schermata resta visibile ancora qualche istante, per dare modo di leggere i valori in caso di particolari necessità.



L'ultima fase è quella di spegnimento della trasmissione e di tutte le funzionalità non essenziali (fase di DEEP SLEEP, oppure POWER SAVE).

In questa fase i sensori restano attivi (è necessario, come da datasheet) e sullo schermo viene visualizzato l'indirizzo MAC (MAC ADDRESS) del noto WIFI. Questa stringa di testo deve essere riportata nel sito all'atto della registrazione, ed insieme al CODICE rappresenta l'univoco per poter gestire correttamente i dati della centralina senza confusione. Questa stringa particolare viene sempre trasmessa insieme ai dati e consente al codice SW presente sul sito WEB di riconoscere in modo univoco (appunto) la provenienza dei dati.



Il codice MAC è una stringa alfanumerica che è costituita da una serie di 6 numeri in base ESADECIMALE separati dal carattere " : ".

Per i non addetti ai lavori, qui una breve descrizione: https://it.wikipedia.org/wiki/Indirizzo_MAC

In foto è rappresentato un numero che termina con la cifra "4": questo numero è in realtà "04" e come tale deve essere inserito sul sito WEB. L'esempio è proprio scelto per spiegare che in generale, in caso di carattere singolo (0-9 e A-F) sarà necessario aggiungere uno "0" alla sinistra del valore letto durante l'inserimento nella pagina WEB.

Esempi:

- 30:AE:A4:CC:4E:4 diventa 30:AE:A4:CC:4E:04
- 17:2B:8:FG:A1:45 diventa 17:2B:08:FG:A1:45
- 30:A:80:12:3D:2D diventa 30:0A:80:12:3D:2D

Il sistema resta in stato di power save per 30 minuti, e lavora per un ciclo di meno di 1 minuto, per poi ritornare in SLEEP.

Il risparmio di energia è viziato non tanto dal display che resta acceso, quanto dal fatto che alcuni sensori, per funzionare correttamente, non possono essere spenti, e il loro consumo è di gran lunga maggiore del consumo dello schermo.

Si è deciso di mantenere acceso il display per discriminare lo stato di "riposo" da una centralina spenta o, peggio, rotta.

Il display montato sulla scheda è un OLED (Organic LED: https://it.wikipedia.org/wiki/OLED) che non ha retroilluminazione e il cui consumo dipende solo dal numero di pixel accesi.

Nel nostro caso, il display accende un centinaio di pixel per la scritta, su un totale di 8.192 punti a disposizione, accendendosi quindi per circa l'1,2% del totale possibile.

4.3. Guida alla installazione di ESP32 sull'IDE Arduino (per programmare)

Le istruzioni che seguono consentono di installare sul proprio computer il sistema di sviluppo con cui è possibile programmare la centralina.

PREMESSE:

- è fortemente raccomandato avere qualche esperienza di programmazione di Arduino o simili, in quanto i passi seguenti, anche se semplici, sono "tecnici" e forse risultano un po' ostici ai non addetti ai lavori e ai principianti assoluti.
- Pur se la guida è fatta in modo da indirizzare passo passo l'utente, le versioni di software, i sistemi operativi, le librerie e altri pezzi di programmi possono variare, e con queste variazioni ci sono inevitabili modifiche che sono difficilmente predicibili. Una schermata qui riprodotta potrebbe cambiare, un passaggio potrebbe non essere più necessario o potrebbero non corrispondere i tasti da premere…
- Insomma è molto consigliato capire quello che si sta facendo, in modo da poter sopperire ad eventuali diversità che dovessero accadere durante la fase di installazione.

L'installazione si compone di 3 fasi, distinte:

a. Installazione dell'ambiente ARDUINO IDE
b. Installazione del sistema HARDWARE ESP32
c. Installazione delle librerie necessarie alla centralina

4.3.1. Installazione di Arduino IDE

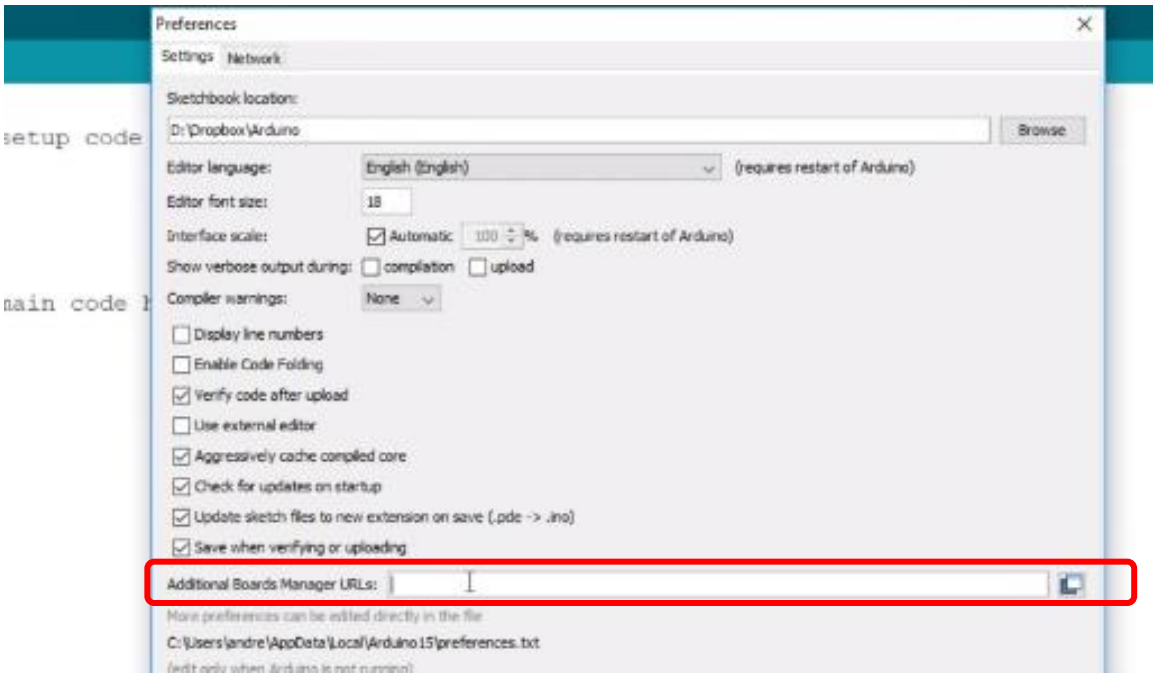
1. Collegarsi al sito www.arduino.cc
2. Dal menu "download" scaricare il file WINDOWS INSTALLER di installazione per il proprio S.O. (qui si farà riferimento a S.O. WIN10)



3. Lanciare il file ".EXE" al termine del download
4. Alla selezione dei componenti da installare, lasciare tutte le selezioni per installare:
    a. Arduino software
    b. USB driver
    c. Start Menu shortcut
    d. Desktop shortcut
    e. Associazione con files .INO
5. Il sistema fa scegliere ora la cartella di installazione. Si può lasciare la cartella di default suggerita e premere su INSTALL.
6. Dopo qualche istante il SW si ferma per chiedere se si vogliono installare i driver USB: ACCETTARE e proseguire con l'installazione
7. Al termine dell'installazione compare una ICONA sul desktop. Doppio click per lanciare il primo avvio di Arduino.
8. Installazione completata!

4.3.2. Installazione dell'Hardware e SW di sviluppo di ESP32 su IDE arduino

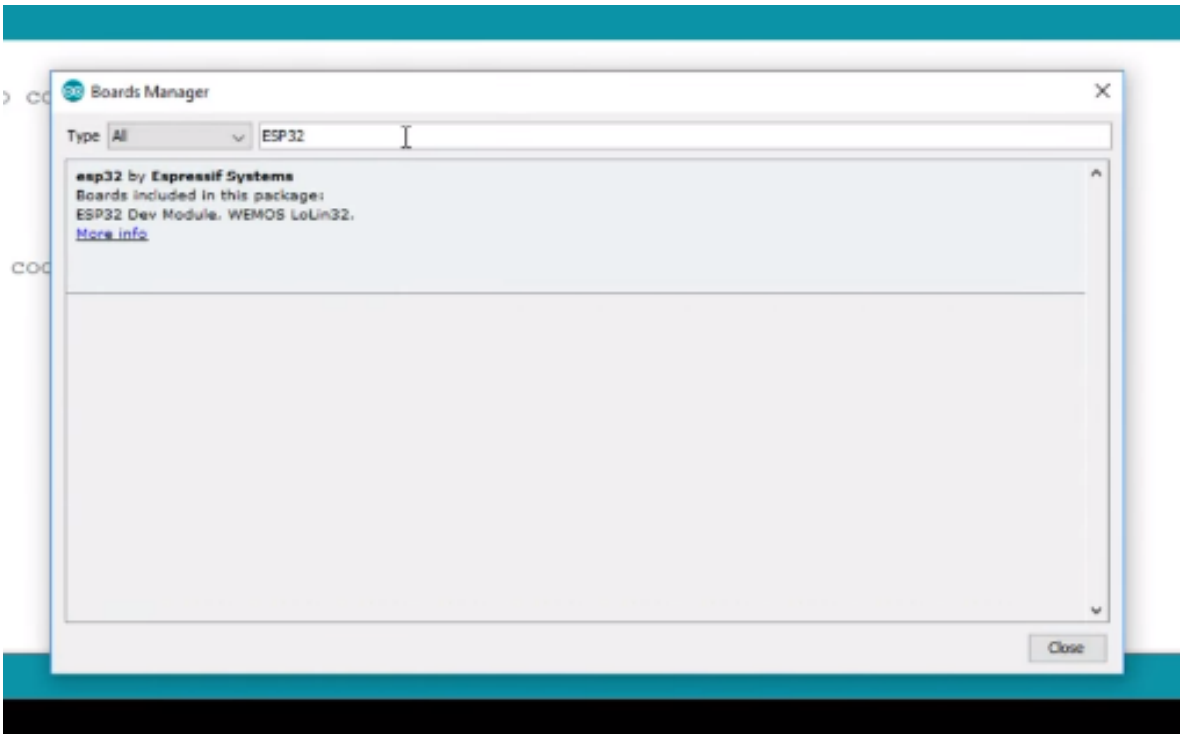
1. Aprire Arduino, selezionare il menu "File" → "Preferences" e si aprirà la finestra seguente:



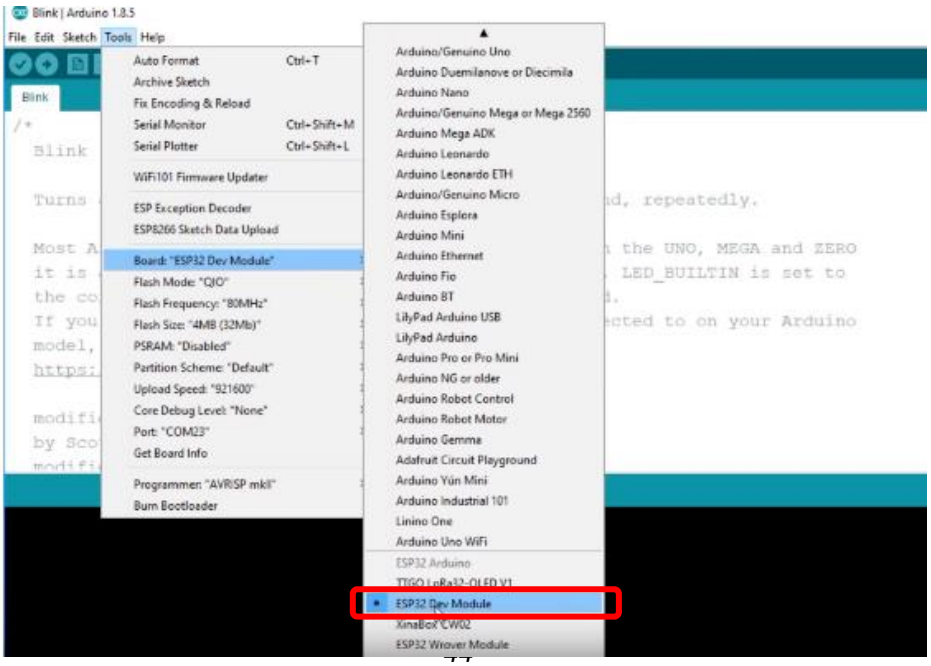
2. Nello spazio evidenziato (BOARD MANAGER) inserire il seguente link (se necessario, separare da altre stringhe già presenti utilizzando una VIRGOLA):

   https://dl.espressif.com/dl/package_esp32_index.json

3. Ora andare nel menu “Tools→ Board: →Board Manager… “ e si aprirà la seguente finestra:



4. Effettuare la ricerca per stringa “ESP32” e comparirà il “package” come in figura, di Espressif: selezionare il package e premere su INSTALL
5. Attualmente la versione stabile (che sarà installata) è la versione 1.0.0; non si esclude che in futuro la versione cambierà. Per il momento installare SOLO la versione indicata per evitare problemi (con la 1.0.2 ad esempio, su MAC non funziona).
6. In questo modo si è conclusa l’installazione del pacchetto di programmazione delle schede ESP32, presenti nella centralina.
7. L’ultimo passo da effettuare è quello di selezionare la scheda HW corretta. Dal menu “Tools→Board:” scorrere tutte le diverse schede fino ad incontrare la scheda **ESP32 Dev Module, e selezionarla**

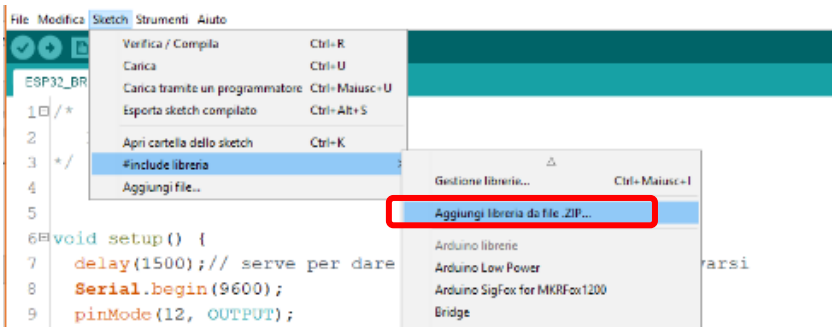


4.3.3. Installazione delle librerie necessarie per il funzionamento dei sensori

Le librerie necessarie sono fornite (come file ZIP) insieme al sorgente.

Per installare una libreria ZIP da arduino è necessario completare i seguenti passi:

1. Aprire Arduino, dal menu “Sketch →#include libreria” selezionare AGGIUNGI LIBRERIA DA FILE ZIP….:



2. Selezionare quindi la libreria che si desidera installare, il sistema installerà tutto in automatico. E’ necessario installare una libreria alla volta, nel seguente ordine:
    a. Arduino-esp32 master
    b. Adafruit Sensor Master
    c. Adafruit BME680 master
    d. Adafruit BME280 master
    e. Pms-5003
    f. PMS Master
    g. NTPClient master
    h. U8G2Lib (https://www.arduinolibraries.info/libraries/u8g2)

La procedura di installazione delle librerie su MAC si rivela diversa, ma dopo qualche tentativo (e qualche guida online se necessario) il tutto si dovrebbe risolvere.

Quando la compilazione va a buon fine avete completato la procedura di installazione.



In caso di problemi, Arduino segnalerà l’errore indicando anche quale libreria non è stata trovata.

Se tutto compila senza errori, siamo pronti a trasferire il codice nella scheda ESP32

4.3.4. PROGRAMMAZIONE della scheda ESP32

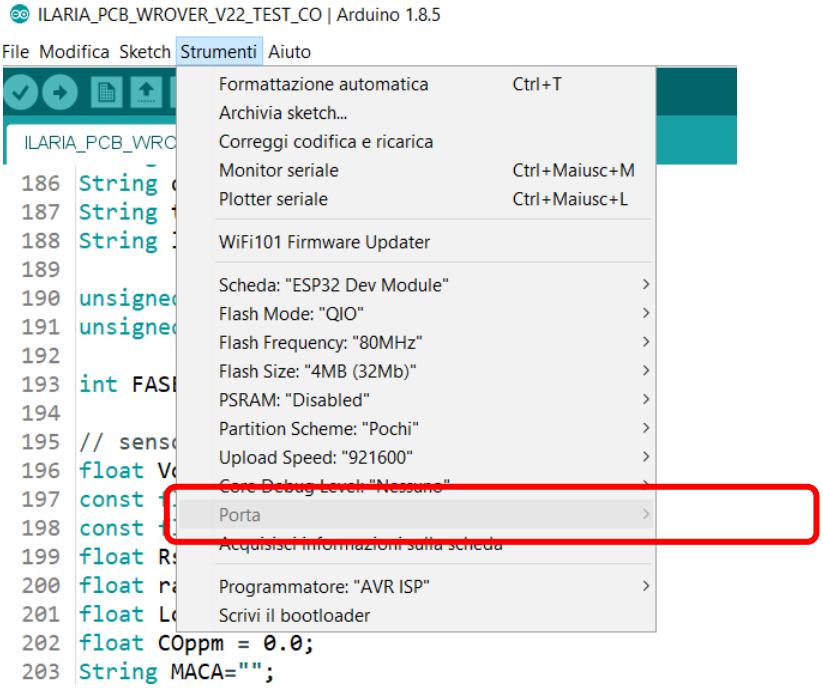
Aprire Arduino e caricare la logica sorgente (file: ILARIA_PCB_WROVER_V21… è la release aggiornata al momento della scrittura del seguente manuale, ma potrebbe cambiare in futuro) e provare a compilare il codice

1. Collegare il cavo al PC alla porta USB ed alla porta microUSB del dispositivo:



2. Il PC dovrebbe emettere il classico suono di conferma del riconoscimento di una nuova periferica, a questo punto è necessario selezionare la giusta porta seriale per consentire al PC di programmare il giusto hardware.

In Arduino è necessario andare su



E selezionare la COMxx corretta (xx sarà il numero della seriale trovata).

In caso di problemi, e' necessario installare i driver corretti: seguire ad esempio la guida qui:

https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers

scaricando quello relativo al proprio sistema operativo e lanciando l'eseguibile al suo interno.

premere il pulsante di programmazione:



3. Se tutto va per il verso giusto, la programmazione dovrebbe partire dopo qualche istante (la prima volta che si compila potrebbe volerci qualche MINUTO di attesa per la compilazione, fino a 15 secondi invece per la programmazione)
4. SE NON SI RIESCE A PROGRAMMARE, e a schermo viene generata una striscia simile a questa:

   “ . . . . .------- . . . . .------- . . . . .------ . . . . ------. . . . . ”

   Senza che la programmazione si avvii (fino ad un errore di timeout di connessione…) allora significa che il vostro PC non è in grado in autonomia di sincronizzarsi con l’hardware ESP32 in vostro possesso. In questo caso è necessario un “aiuto” da parte vostra (come indicato nel manuale).
   In questo caso dovete lanciare la programmazione nuovamente e, non appena comincia la scrittura dei punti e linee, effettuare un ciclo di BOOT nel processore.
   La cosa è molto semplice: ci sono 2 pulsanti sull’hardware: BOOT ed EN (Enable):



   Il ciclo di BOOT si esegue in questo modo:

   - Premere il pulsante BOOT
   - Tenendo premuto BOOT, premere e rilasciare il pulsante EN
   - Rilasciare il pulsante BOOT

   A questo punto la programmazione partirà con sicurezza, e l’hardware ESP32 verrà programmato con il codice che avete aperto.