

## Il programma per il sensore di CO<sub>2</sub>

In questo manuale vedremo tutte le operazioni per poter importare, comprendere e caricare sulla scheda il programma di gestione del kit di rilevazione della CO<sub>2</sub> in ppm costruito per questo laboratorio STEM.

Il programma che useremo utilizza delle particolari **Librerie**, ovvero elenchi di funzioni o di strutture dati che servono per far operare alcuni dei componenti inseriti nel progetto.

In particolare nel software saranno presenti le librerie:

- LCDIC2: permette la gestione dello schermo LCD
- WIRE: permette di comunicare con i componenti con standard I<sup>2</sup>C, come il nostro LCD
- SPARKFUN SCD30 ARDUINO LIBRARY: permette di gestire il sensore di CO2
- **ESP8266WIFI:** permette di utilizzare la connessione WiFi della NodeMCU
- THINGSPEAK: permette di inviare i dati a ThingSpeak™

Le librerie WIRE e ESP8266WIFI sono già presenti di default, mentre le altre necessitano di installazione. Questa operazione va fatta unicamente una volta: le librerie risulteranno già installate anche per progetti successivi. Vediamo i passaggi necessari.

Per installare qualsiasi libreria, cliccare sul menù **Strumenti** e poi su **Gestione librerie** (bisognerà essere connessi a internet per scaricare e installare i pacchetti)



Per installare la libreria LCDIC2 digitare nel campo di ricerca della scheda aperta la stringa *lcdic2.* Apparirà il tab della corrispondente libreria, nel quale si dovrà cliccare su Installa

sketch_mar07c   Arduino 1.8.13 (Windows Store 1.8.42.0) File Modifica Sketch Strumenti Aiuto		- 0	×
			P
sketch_mar07c			
<pre>void setup() {     // put your setup code here, to run once:</pre>			^
}			
<pre>void loop() {</pre>	© Gestore librerie X		
<pre>// put your main code nere, to run repeatedly:</pre>	Tipo Tutti V Argomento Tutti V kdic2		
}	LOIC2 by Helder Rodrigues Class Interface for Liquid Crystal More info Versione 2.3.3 Installs		
			~
1		Arduin	io Micro

Dopo aver atteso la fine dell'installazione, sarà possibile installare la libreria **SPARKFUN SCD30 ARDUINO LIBRARY**. Per fare ciò scrivere nel campo di ricerca *sparkfun scd30*. Apparirà il tab della corrispondente libreria, nel quale si dovrà cliccare su **Installa** 

sketch_mar07c   Arduino 1.8.13 (Windows Store 1.8.42.0) File Modifica Sketch Strumenti Aiuto		- (	•	×
			E	0
sketch mar07c			5	1
<pre>void setup() {     // put your setup code here, to run once:</pre>				^
3				
<pre>void loop() {</pre>	© Gestore librerie X			
<pre>// put your main code here, to run repeatedly:</pre>	Tipo Tutti V Argomento Tutti V sportdun scd30			
)	Sparfum SC030 Ardium Library         by Sparfum Electronics         Library for the Sensition SCD30 CO2 Sensor An Arduinoibrary for the SCD30 CO2 sensor from Sensition. The SCD30 is a high quality MDIR         based CO2 sensor capable of detecting 400 to 1000pm with an accuracy of 43(00pm-3%). In order to improve accuracy the SCD30 has         temperature and humidity sensing built-in, as well as commands to set the current albtude.         Get the SCD30 hare.         Morea info         Morea info         Versione 1.0.12 v         Installe			
				~
1		A	Arduino Mic	ro

Dopo aver atteso la fine dell'installazione, sarà possibile installare la libreria **THINGSPEAK**. Per fare ciò scrivere nel campo di ricerca *thingspeak*. Apparirà il tab della corrispondente libreria, nel quale si dovrà cliccare su **Installa** 

<ul> <li>sketch_mar07d   Arduino 1.8.13 (Windows Store 1.8.42.0)</li> <li>File Modifica Sketch Strumenti Aiuto</li> </ul>		- 0	$\times$
			<b>O</b>
skatch mar07d			
<pre>void setup() {     // put your setup code here, to run once:</pre>			^
3			
<pre>void loop() {</pre>	© Gestore librerie X		
<pre>// put your main code here, to run repeatedly:</pre>	Tipo Tutti V Argomento Tutti V thingspeak		
3	ThingSpeak         by MatiWorks         ThingSpeak Communication Library for Arduino, ESP8266 & EPS32 ThingSpeak ( https://www.thingspeak.com ) is an analytic toT platform         service that allows you to aggregate, visualize and analyze live data streams in the doud.         More info         Versione 2.0.1 v         Installe         by Acakil Kono         An API manager for ThingSpeak It writes field values for ThinkgSpeak.         More info         Versione 2.0.1 v         Installe		
			v
1		Ardui	ino Micro

Ora siamo pronti per lavorare sul nostro progetto. Selezioniamo la nostra scheda fra l'elenco di quelle presenti. Per fare ciò apriamo il menù **Strumenti – Scheda – ESP8266 Boards** e selezioniamo la scheda **NodeMCU (ESP-12E Module)** 



Successivamente verifichiamo che la velocità di caricamento del programma sulla scheda sia impostata sul valore di 115200. Per fare ciò apriamo il menù **Strumenti – Upload Speed** e verifichiamo che sia selezionato il valore **115200**.



Ora siamo proprio pronti per caricare il nostro programma e analizzarlo. Il programma, nel linguaggio di Arduino<sup>®</sup> si chiama **sketch**. Dopo aver scaricato il file **sensore\_CO2.ino** dalla pagina web del progetto seguiamo i seguenti passaggi per caricarlo. Aprire il menù **File - Apri** 



Selezionare il file che si desidera aprire, e cliccare su **Apri**. Lo sketch verrà quindi caricato in Arduino IDE

sketch_mar07d   Ardui File Modifica Sketch Str	ino 1.8.13 (Windov umenti Aiuto	vs Store 1.8.42.0)					- 0	×
								<b>1</b>
sketch_mar07d								
<pre>void setup() {     // put your setup</pre>	o code here, t	o run once:						^
roid loop() (	🍰 Apri uno skete	h di Arduino			×			
// put your main	Cerca in:	sensore_CO2	2	v 🗿 🎓 🖡	▼			
}	-	Nome	^	Stato	Ultima modifica			
	Accesso rapido	Sensore_CO2	2.110	$\odot$	06/03/2021 13:48			
	Desktop							
	-							
	Raccolte							
	_							
	Questo PC					/		
	1	<		_				
	Rete	Nome aggetto:	sensore_CO2.ino		~ Apri			
		Tipo oggetto:	Tutti i file (*.*)		~ Annulla			
l								
								~
						10-dati		Indulo)

Se il file non è contenuto in una cartella che riporti il medesimo nome, il software chiederà di poterla creare. A questa richiesta rispondiamo affermativamente.

## Analizziamo lo sketch

Il programma è stato compilato con molte sezioni di spiegazione. Qui commentiamo solo i passaggi generali e le attenzioni da avere nelle modifiche necessarie

💿 sensore_CO2   Arduino 1.8.13 (Windows Store 1.8.42.0)				
File Modifica Sketch Strumenti Aiuto				
/*				
In questa sezione importeremo le librerie necessarie al fun LCDIC2 serve a comunicare e controllare il display LCD Wire è una libreria base di Arduino per comunicare coi disp Sparkfun SCD30 è la libreria che ci permette di comunicare ESP8266WiFi è la libreria che ci permetterà di usare la sch ThingSpeak è la libreria che ci permette l'invio dei dati a	nzionamento dei componenti positivi I2C (come lo schermo LCD) col sensore SCD30 meda WiFi a bordo del micro controllore per inviare i dati a ThingSpeak a ThingSpeak			
Adamata Propred by				
<pre>#include "LCDIC2.h" #include "Wire.h"</pre>				
<pre>#include "SparkFun_SCD30_Arduino_Library.h" #include <esp8266wifi.h> #include <thingspeak.h></thingspeak.h></esp8266wifi.h></pre>	Questa è la sezione dove si dichiarano le librerie da utilizzare			
In questa sezione gestiremo la connessione WiFi e i paramet Dopo char ssid[], al posto di "nomewifi", andrà scritto il Dopo char pass[], al posto di "passwordwifi", andrà scritta Dopo channelID andrà indicato al posto di xxxxxxx l'ID del Dopo APIkey andrà indicato al posto di xxxxxxxxxxxxx l'A WiFiClient client e SCD30 airSensor definiscono la presenza	cri necessari a far funzionare ThingSpeak nome della rete WiFi (sempre fra virgolette)a cui ci si vorrà connettere a la password della rete WiFi (sempre fra virgolette) a cui ci si vorrà connettere canale che avremo creato su ThingSpeak API key del canale che avremo creato su ThingSpeak a della WiFi e del sensore, a cui viene dato il nome "airSensor"			
<pre>char ssid[] = "nomewifi"; // nome char pass[] = "passwordwifi"; // pass unsigned long channelID = xxxxxxx; // ID d char * APIkey = "xxxxxxxxxx; // API WiFiClient client; SCD30 airSensor; /*</pre>	e della rete wifi SSID (RICORDARSI DI SOSTITUIRLO) sword della rete WiFi (RICORDARSI DI SOSTITUIRLO) lel proprio canale Thingspeak (RICORDARSI DI SOSTITUIRLO) key del proprio canale ThingSpeak (RICORDARSI DI SOSTITUIRLO)			
C	connettere la scheda e il ChannellD e l'API Key di ThingSpeak. <b>Ricordarsi di</b> <b>sostituire le diciture corrette</b> al posto dei campi corrispondenti, avendo l'accortezza di mantenere le virgolette laddove presenti			
1	NodemCU 1.0 (ESP-1 2E Module), 80 MHZ, Flash, Legacy (new can return huliptr), All SSL cipne			
<ul> <li>sensore_CO2   Arduino 1.8.13 (Windows Store 1.8.42.0)</li> <li>File Modifica Sketch Strumenti Aiuto</li> <li>The sensore CO2</li> <li>In questa sezione definiamo tutte le variabili che ci serviranno r</li> </ul>	nel programma			
*/				
<pre>float co2_lettura = 0.0; // questa variabile conterrà la lettura float co2_somma = 0; // questa variabile sarà utile per fare float co2_media = 0; // questa variabile sarà utile per fare int cicli_di_lettura = 10; // questa variabile controlla il tempo- int tempo_di_attesa = 1000; // questa variabile definisce il valo int valore_attenzione = 700; // questa variabile definisce il valo int valore_attenzione = 700; // questa variabile definisce il valo int valore_attenzione = 700; // questa variabile definisce che il 1 int LEDorso = D3; // questa variabile definisce che il 1 int LEDorso = D5; // questa variabile definisce che il 1 int buzzer = D6; // questa variabile definisce che il 1 String co2_lettura_testo; // questa variabile In questa sezion //In questa riga definiamo come è fatto lo se //In questa reale che realeration //In questa realeration secion //In questa realeration //In questa realeration secion //In questa realeration //In</pre>	di CO2 data dal sensore (valore iniziale 0.0) la somma di diverse letture di CO2 (valore iniziale 0) la media di diverse letture di CO2 (valore iniziale 0) co di letture su cui fare la media (valore preimpostato 10) o fra una lettura e la successiva; è espressa in millisecondi (1000 ms = 1 s) ce di CO2 sopra il quale l'aria diventa satura (MODIFICARE IL VALORE COME SI DESIDERA) lore di CO2 sopra il quale prestare attenzione (MODIFICARE IL VALORE COME SI DESIDERA) lore di cO2 sopra il quale prestare attenzione (MODIFICARE IL VALORE COME SI DESIDERA) led rosso si connette al pin D3 lei giallo si connette al pin D5 more si connette al pin D6 more definiamo tutte le variabili del programma e come è fatto lo schermo LCD. I critico e di attenzione sono da modificare secondo i propri desideri			

sensore\_CO2 | Arduino 1.8.13 (Windows Store 1.8.42.0)
 File Modifica Sketch Strumenti Aiuto

sensore_CO2	
<pre>void setup() {     WiFi.begin(ssid,pass);     while (WiFi.status() != WL_CONNECTED) {     delay(500);     Serial.print(".");     Serial.println("");     Serial.println("WiFi connected"); </pre>	<pre>// con questa istruzione ci si connette alla WiFi, tramite nome e password forniti in precedenza // in questa sezione si visualizza su monitor seriale lo status di connessione // alla WiFi, dando informazioni sull'avvenuta connessione o meno Questa sezione è il SETUP dello Sketch, ovvero la parte di codice che viene eseguita una sola volta all'accensione. Contiene la verifica dell'avvenuta connessione WiFi, la definizione come</pre>
<pre>Serial.println("IP address: "); Serial.println(WiFi.localIP());</pre>	output dei pin dei led e dei buzzer, e una routine che avvisa quando il sensore non risponde, illuminando a intermittenza il led giallo
<pre>ThingSpeak.begin(client); Wire.begin();</pre>	// con questa istruzione ci si connette a Thingspeak // con questa istruzione si inizializza la libreria Wire che gestisce i cablaggi
<pre>pinMode (LEDrosso, OUTPUT); pinMode (LEDverde, OUTPUT); pinMode (LEDgiallo, OUTPUT); pinMode (buzzer, OUTPUT);</pre>	// con queste quattro istruzioni si definisce che i pin della scheda che gestiscono // i led e il buzzer sono dei pin che generano un OUTPUT (attuatori)
<pre>if (airSensor.begin() == false) {     while (1) {         digitalWrite(LEDgiallo, HIGH);         delay(500);         digitalWrite(LEDgiallo, LOW);         delay(500); </pre>	// questa istruzione controlla se il sensore SCD30 è funzionante // questa sezione fa lampeggiare il led giallo ogni 0.5 s // se il sensore non è funzionante
<pre>lcd.begin(); lcd.setCursor(0, 0); lcd.print("CO2 Errore Sensore"); lcd.setCursor(0, 1); lcd.print("Controlla Circuito"); } };</pre>	// questa sezione mostra sul display LCD un messaggio di errore // nel caso il sensore SCD30 non sia funzionante // nella prima riga, prima colonna fa apparire ERRORE SENSORE // nella seconda riga, prima colonna fa apparire CONTROLLA CIRCUITO
<pre>lcd.begin(); }</pre>	// questa istruzione inizializza il display LCD

sensore\_CO2 | Arduino 1.8.13 (Windows Store 1.8.42.0)

File Modifica Sketch Strumenti Aiuto	
💎 📀 🗈 🔛 Carica	
sensore_CO2	
void loop() {	
co2_somma = 0;	// all'inizio di ogni ciclo si dichiara che la somma dei valori di CO2 è uguale a O
// viene creato un ciclo all'interno del LOOP	con lo scopo di effettuare 10 letture di CO2 e sommarle nella variabile cO2_somma
<pre>for (int i = l; i &lt; cicli_di_lettura; i++) {     if (airSensor.dataAvailable())     {         co2_lettura = int(airSensor.getCO2());     }     lcd.setCursor(l2, l);</pre>	Questa sezione è la prima parte del <b>LOOP</b> dello Sketch, ovvero la parte di codice che viene eseguita per sempre fino a che non si spegne la scheda. Contiene un ciclo che realizza 10 letture di CO <sub>2</sub> , il calcolo della media delle letture, e la scrittura di alcune informazioni sul display LCD
<pre>lcd.print(String(cicli_di_lettura-i)); delay(tempo_di_attesa); co2_somma += co2_lettura;</pre>	<pre>// nella posizione (12,1) scrive sul display LCD il valore corrispondente al ciclo in esecuzione // attende un tempo per effettuare una nuova lettura (l s per come definito sopra) // somma il valore della lettura corrente ai valori delle precedenti letture</pre>
<pre>lcd.setCursor(0, 0); lcd.print("Livello di CO2");</pre>	// si posiziona il cursore del display LCD per la scrittura alla posizione (colonna l, riga l) // nella posizione (l,l) scrive sul display LCD il testo Livello di CO2
<pre>lcd.setCursor(7, 1); lcd.print("PPM"); }</pre>	// si posiziona il cursore del display LCD per la scrittura alla posizione (colonna 8, riga 2) // nella posizione (7,1) scrive sul display LCD il testo PPM
<pre>// termina il ciclo precedentemente definito; // e sul display compaiono alcuni testi come</pre>	in questo momento la variabile co2_somma contiene la somma di 10 letture di CO2 fatte Livello di CO2, PPM e il conteggio progressivo
<pre>co2_media = co2_somma / (cicli_di_lettura-l); co2_media = int(co2_media); co2_lettura_testo = String(co2_media);</pre>	// questa istruzione calcola la media delle 10 letture (somma/numero di letture) // questa istruzione converte la media in un numero intero // questa istruzione converte il valore della media in un testo, per visionarlo sul display LCD
<pre>lcd.setCursor(0, 1); lcd.print(co2_lettura_testo);</pre>	// si posiziona il cursore del display LCD per la scrittura alla posizione (colonna 1, riga 2) // nella posizione (0,1) scrive sul display LCD il valore della media di 10 letture di CO2 in PPM

sensore\_CO2 | Arduino 1.8.13 (Windows Store 1.8.42.0)

File Modifica Sketch Strumenti Aiuto

🕑 🔶 🛅 🔝 🔛 Carica	
sensore_CO2	
<pre>/* si definisce un'istruzione condizionale (IF) pe precedentemente definito si accende il led ross se la media di letture è invece compreso tra il si spengono i led rosso e verde, e si spegne il se la media di letture è uguale a 0 (3° caso), si accendono tutti e tre i led, e il huzzer fa in tutti gli altri casi (4° caso), ovvero in cu si spengono i led giallo e rosso e si spegne il */</pre>	la quale, nel l° caso, se la media di letture della CO2 supera il valore critico (in FFM) si apengono i led giallo e verde, e il buzzer si mette a suonare alore di attenzione e il valore critico precedentemente definiti (2° caso), si accende il led giallo, izzer solitamente significa che il sensore è in calibrazione o è in errore, suono più frequente La media delle letture è maggiore di 0 ma sotto il valore di attenzione, si accende il led verde, izzer
<pre>if (co2_media &gt; valore_critico) {     digitalWrite(LEDverde,LOW);     digitalWrite(LEDgiallo,LOW);     digitalWrite(LEDrosso,HTGH);     tone(buzzer,440,5000);     }     else if (co2_media &gt; valore_attenzione) {         digitalWrite(LEDverde,LOW);         digitalWrite(LEDrosso,LOW);         digitalWrite(buzzer,LOW);     } }</pre>	// 1° caso // questa istruzione fa emettere un suono a una frequenza di 440 Hz (LÀ) per una durata di 5 s (5000 ms) // 2° caso
<pre>else if (co2_media == 0) {     digitalWite(LEDverde, HIGH);     digitalWite(LEDvrade, HIGH);     digitalWite(LEDvade, HIGH);     tone(buzzer, 440, 1000);     }     else{         digitalWite(LEDverde, HIGH);         digitalWite(LEDverde, HIGH);         digitalWite(LEDverde, LOW);         digitalWite(buzzer, LOW);     } }</pre>	<sup>7/ 3° caso</sup> Questa è la seconda parte del LOOP dello Sketch. Contiene il comportamento dei led a seconda che la media dei valori rilevato in ogni ciclo di lettura sia sopra o sotto determinati valori. Inoltre invia di dati a ThingSpeak™
//in quest'ultima parte, si inviano i dati a Th	jSpeak
<pre>ThingSpeak.setField(1,co2_media); ThingSpeak.writeFields(channelID, APIkey); }</pre>	// questa istruzione comunica su quale campo (1) del canale di ThingSpeak scrivere il dato della media (co2_media) // questa istruzione comunica su quale canale di ThingSpeak effettuare la scrittura, tramite i parametri precedentemente definiti

Apportate le modifiche desiderate nelle parti di codice coinvolte, è possibile ora caricare lo sketch. Per fare ciò colleghiamo la scheda al computer tramite il cavo USB – microUSB



Torniamo sullo sketch, e assicuriamoci che sia selezionata la porta di comunicazione corretta, individuata dal computer dopo il collegamento. Per fare ciò cliccare su **Strumenti – Porta** e verificare la porta presente (potrebbe differire dalla figura). A questo punto inseriamo la spunta sulla porta evidenziata.

sensore_CO2   Arduino 1.8.13 (Windows Store 1.8.42.0)	- 0	×
File Modifica Sketch Strumenti Amo		
Formattazione automatica Ctrl+T		0
Archivia sketch		
sensore_CO2 Correggi codifica e ricarica		
/* Gestione librerie Ctrl+Maiusc+I		^
si definisce un Monitor seriale Ctrl+Maiusc+M media di letture della CO2 supera il valore critico (in PPM)		
precedentemente Plotter seriale Ctrl+Maiusc+L verde, e la buzzer si mette a suonare		
se sa media di		
si accendono tut Scheda: "NodeMCII 10 (FSD-12E Module)" >>		
in tutti gli al glide de la contectione de la co		
as spengono 1.0 butter course in the sector of the sector		
if (co2_media CFOTREQUERY, OUNTZ /		
digitalWrite Tlobu Jack AWD (TSZMD OTA~1019KD) /		
digitalirité Deug por Disabled /		
tone (buzzer, bellugi tever: wessuno tever: wess		
invite variant: V2 Lower memory 2		
digitalWrite		
digitalWrite Exceptions: tegacy (new can teturn numpu)		
digitalWiir EldserHalt Uniy Settion /		
SSL support. An SSL comparison of the second s		
else if (co2 voite international de stands voite stands voite stands)		
digitalkrite Acquisici intornazioni sulla scieda		
digitalite Programmatore		
tone (Mazzer, Scrivi il bootloader		
else( // 4° caso digitalWrite(LEDverde,HIGH); digitalWrite(LEDgallo,LON);		ł
<pre>digitalWrite(buzzer,LOW); }</pre>		
//in quest'ultima parte, si inviano i dati a ThingSpeak		
ThingSpeak.setField(1,co2_media); // questa istruzione comunica su quale campo (1) del canale di ThingSpeak scrivere il dato della media (co2_media)		
ThingSpeak.writeFields(channelID, AFIkey); // questa istruzione comunica su quale canale di ThingSpeak effettuare la scrittura, tramite i parametri precedentemente definiti		
		_
	NodeMCU 1.0 (ESP-12E Module) su C	сомз

Tutto è pronto. Possiamo cliccare sul pulsante **Carica** per far iniziare prima di tutto il processo di **compilazione**, ovvero il controllo da parte del software che il codice sia scritto correttamente. In questo processo ci saranno segnalati eventuali errori e potremmo in caso porre rimedio. Se il processo di compilazione va a buon fine, in automatico partirà il processo di **caricamento**, al termine del quale, se comparirà la scritta **Caricamento completato**, significherà che il processo è andato a buon fine e il kit inizierà a funzionare.

sensore_CO2   Arduino 1.8.12 File Modifica Sketch Strumenti Aiuto		- 0 ×
		<u>a</u>
sensore_CO2		
<pre>/* ai definince un'istruzione condizionale (IP) precedentamente definito ai accende il led ros se la media di letture è invece compreso tra i i apengono il led rosco e verde, e ai apegne i se la media di letture è uguale a 0 (3° caso), si accendono tutti e tra i led, eil butzer fa in tutti gli altri casi (4° camo), ovvero in c si spengono i led giallo e rosso e si spegne i */</pre>	er la quale, mel 1° camo, me la media di letture della CO2 supera il valore critico (in FFM) mo, ni apengono i led giallo e verde, e il buzzer ni mette a muonare l valore di attenzione e il valore critico precedentemente definiti (2° camo), mi accende il led giallo, l buzzer che solitamente significa che il sensore è in calibrazione o è in errore, un nuono piò frequente ui la media delle letture è maggiore di 0 ma sotto il valore di attenzione, mi accende il led verde, l buzzer	~
<pre>if (co2_media &gt; valore_critico) {     digitalWrite(LEDverdm.LOW) }     digitalWrite(LEDverdm.LOW) ;     digitalWrite(LEDverdm.NOW) ;     tome(buzzer.440.3000);     tome(buzzer.440.3000);     digitalWrite(LEDverdm.LOW) ;     di</pre>	// 1° caso // questa istruzione fa emettere un suono a una frequenza di 440 Hz (LA) per una durata di 5 s (3000 ms) // 2° caso	
<pre>else if (co2_media 0) {     digitalWrite(LEDverde,HIGH);     digitalWrite(LEDverde,HIGH);     digitalWrite(LEDgiallo, HIGH);     tone(Duzzer,440,1000); }</pre>	// 3° caso	
<pre>elne(     digitalWrite(LEDvorde, HIGH);     digitalWrite(LEDgiallo,LOW);     digitalWrite(LEDrosso,LOW);     digitalWrite(buzzer,LOW); }</pre>	// 4° сано	
//in quest'ultima parte, si inviano i dati a T	hingSpeak	
<pre>ThingSpeak.setField(1,co2_media;; ThingSpeak.writeFields(charmelID, APIkey); }</pre>	// questa istruzione comunica su quale campo (1) del canale di ThingSpeak scrivere il dato della media (co2_media) // questa istruzione comunica su quale canale di ThingSpeak effettuare la scrittura, tramite i parametri precedentemente definiti	•
Caricamento completato		
Leaving Rard resetting via RTS pin		^ 
45		NodeMCU 1.0 (ESP-12E Module) su COM3

N.B. lo sketch non funzionerà se caricato sulla scheda così come scaricato dalla pagina del progetto, senza apportare le modifiche necessarie variando il nome della WiFi (ssid), la password della WiFi, il channelID e l'API Key di ThingSpeak™.

Il kit si accenderà, e inizierà a fare le prime letture. Sarà possibile a questo punto anche staccarlo dal computer e alimentarlo con l'alimentatore 5V di un normalissimo smartphone



Sarà anche possibile fissure tutti i componenti ad esempio su una basetta di legno, e posizionarlo in un qualche punto di un locale chiuso per poter rilevare il tasso di CO<sub>2</sub> presenti nel locale.

A questo link

https://www.sensirion.com/en/environmental-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors/carbon-dioxide-se

sono presenti maggiori informazioni sul sensore, compresi i datasheet e le indicazioni di posizionamento.

Se il kit genera un qualche errore, sarà sufficiente premere sul piccolo tasto **RST** presente sulla scheda per riavviare il programma, o in casi più estremi staccare e riattaccare l'alimentazione.

Con l'accensione anche il canale creato su ThingSpeak™ inizierà a funzionare. Sarà sufficiente collegarci al sito web del nostro canale per visualizzare i dati sul monitor e scaricarli per la rielaborazione.



## Enjoy!!!