



BENVENUTI!

**PROGETTO DI
SPERIMENTAZIONE
SULL'INTERNET DELLE COSE
(IoT)**



I PROSSIMI PASSI CON IoT

Imparare ad ottenere dati dalle API,
come estrarre questi dati dai pacchetti **JSON**
e visualizzarli sullo schermo del carrier

09|12|20

Con Francesco Piero Paolicelli

4 APPUNTAMENTI IMPERDIBILI

- Grazie a tutti gli esperti che hanno presentato negli appuntamenti precedenti



DOVE TROVARE L'EXPLORE IOT KIT



Arduino Explore IoT Kit Singolo

Codice: 333190

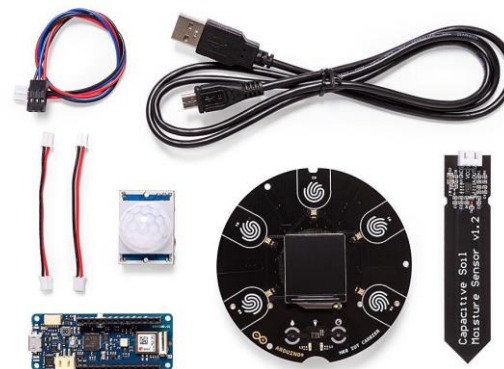
Codice MEPA: 333190CS



Arduino Explore IoT Kit Min. 10 pz

Codice: 334389

Codice MEPA: 334389CS



Arduino Explore IoT Kit Min. 20 pz

Codice: 334390

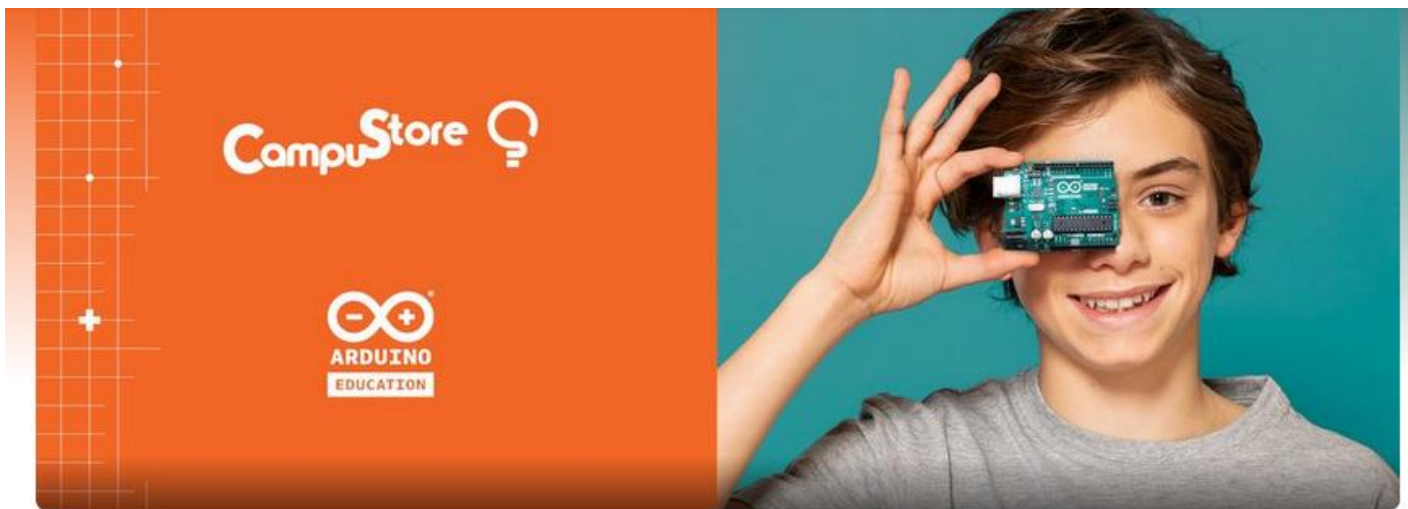
Codice MEPA: 334390CS

Arduino Explore IoT Kit Min. 100 pz

Codice: 334391

Codice MEPA: 334391CS

ARDUINO EDUCATION ITALIA - GRUPPO UFFICIALE FACEBOOK



Arduino Education Italia

 Gruppo Privato · 726 membri

Per raccogliere tutti gli educatori italiani, genitori e studenti interessati ad Arduino, CampuStore e Arduino Education hanno unito le forze e creato un gruppo **Facebook** chiamato “**Arduino Education Italia**”





A scuola con Arduino Create

<https://forms.gle/LR84ggcMkvdxP9uM9>

Il tuo contributo è importante e ci aiuterà a progettare strumenti ed esperienze che possano essere inseriti con successo nel programma formativo



COME CONNETTERSI CON IL MONDO → API DI TELEGRAM

Ovvero: come comandare il nostro Arduino IOT KIT tramite la chat di Telegram



```
{
  ok: true,
  - result: [
    - {
      update_id: 854060854,
      - message: {
        message_id: 176,
        - from: {
          id: 69668132,
          is_bot: false,
          first_name: "Piersoft",
          username: "piersoft",
          language_code: "it"
        },
        - chat: {
          id: 69668132,
          first_name: "Piersoft",
          username: "piersoft",
          type: "private"
        },
        date: 1607014048,
        text: "/start",
        - entities: [
          - {
            offset: 0,
            length: 6,
            type: "bot_command"
          }
        ]
      }
    ]
  ]
}
```

UN'INFARINATURA SULLE API TELEGRAM

Inizieremo prima di tutto configurando il nostro BOT (da Robot) e quindi le istruzioni che richiederanno dati da una **API**, poi useremo qualcosa chiamato **JSON parsing** per estrarre i dati, e infine, saremo in grado di leggere i comandi inviati in Chat Telegram utilizzando il **Serial Monitor**.

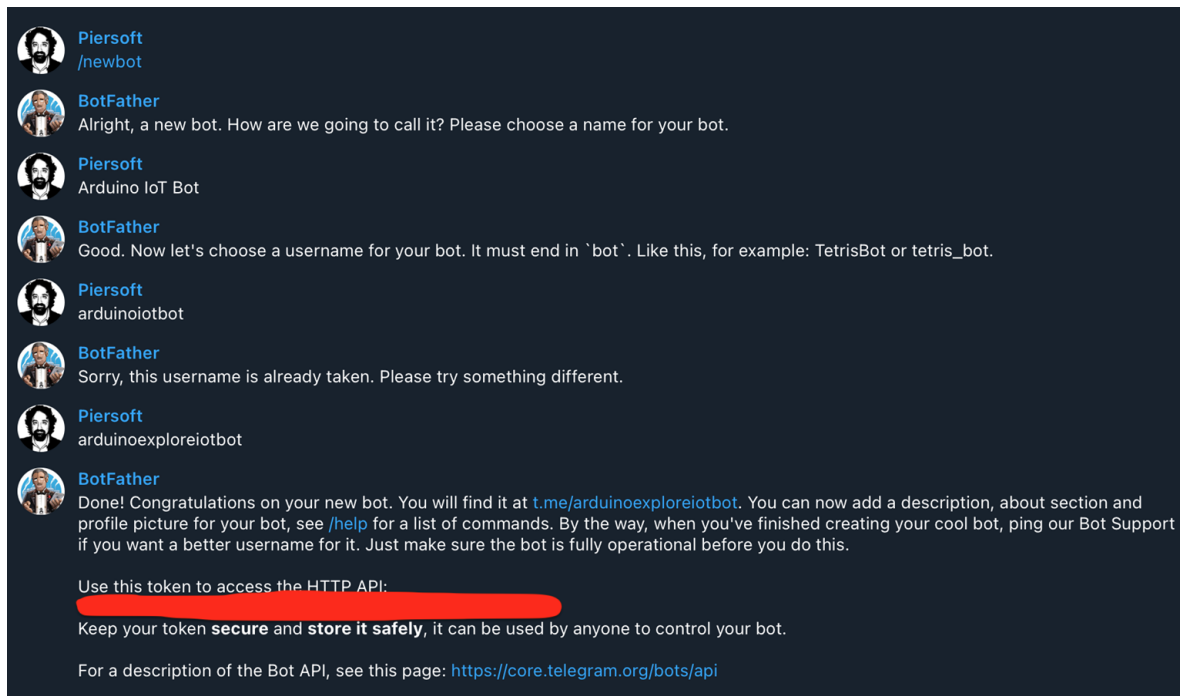
Esempio risposta JSON dei dati di una Chat Telegram al comando:

<https://api.telegram.org/bot<<TOKEN>>/getUpdates>

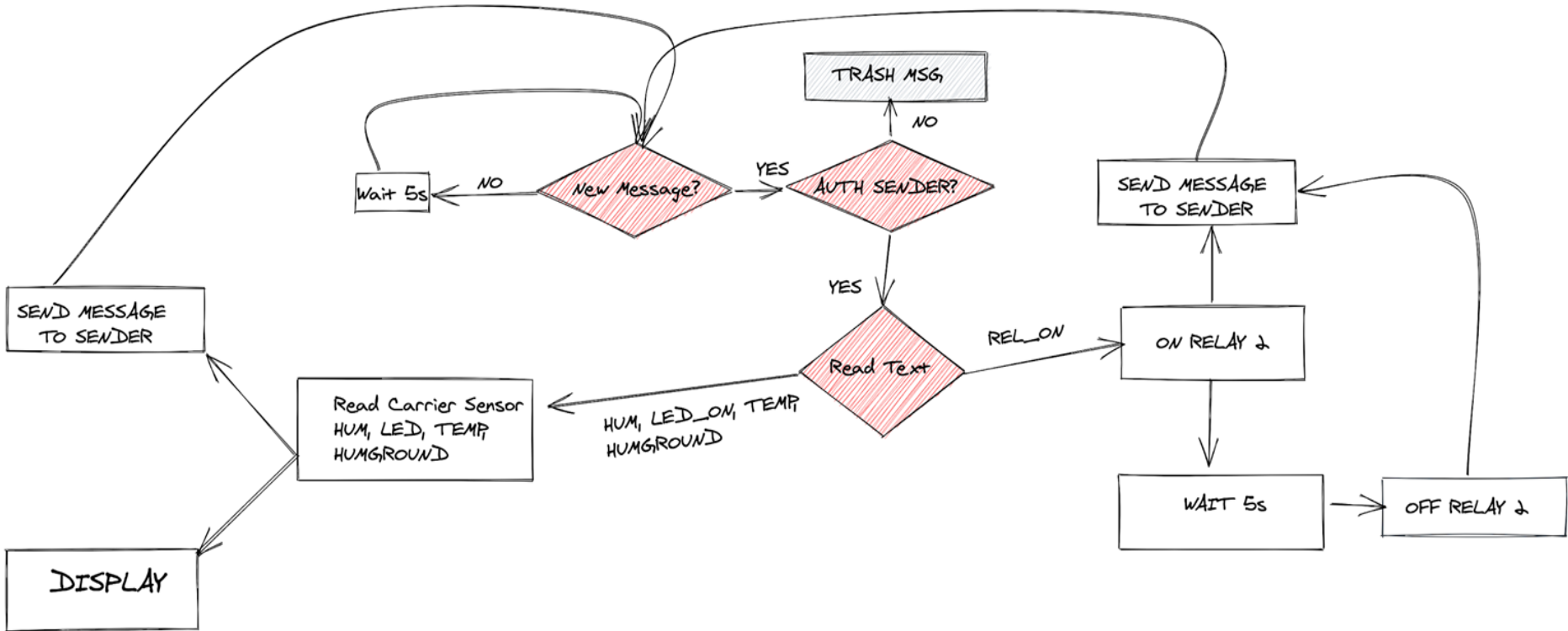


COSTRUIAMO IL PROCESSO GRADUALMENTE

1. Aprire Telegram e cercare l'utente **@Botfather**
2. Dopo averlo avviato digitare: **/newbot**
3. Creare un nome pubblico (es. **Mio Bot**) e uno username (es. **miobot**)
4. **@Botfather** mi assegnerà un **TOKEN**. Complimenti avete attivato un Bot personale :)
5. Attenzione!! Il **TOKEN** è strettamente personale!!



LA LOGICA



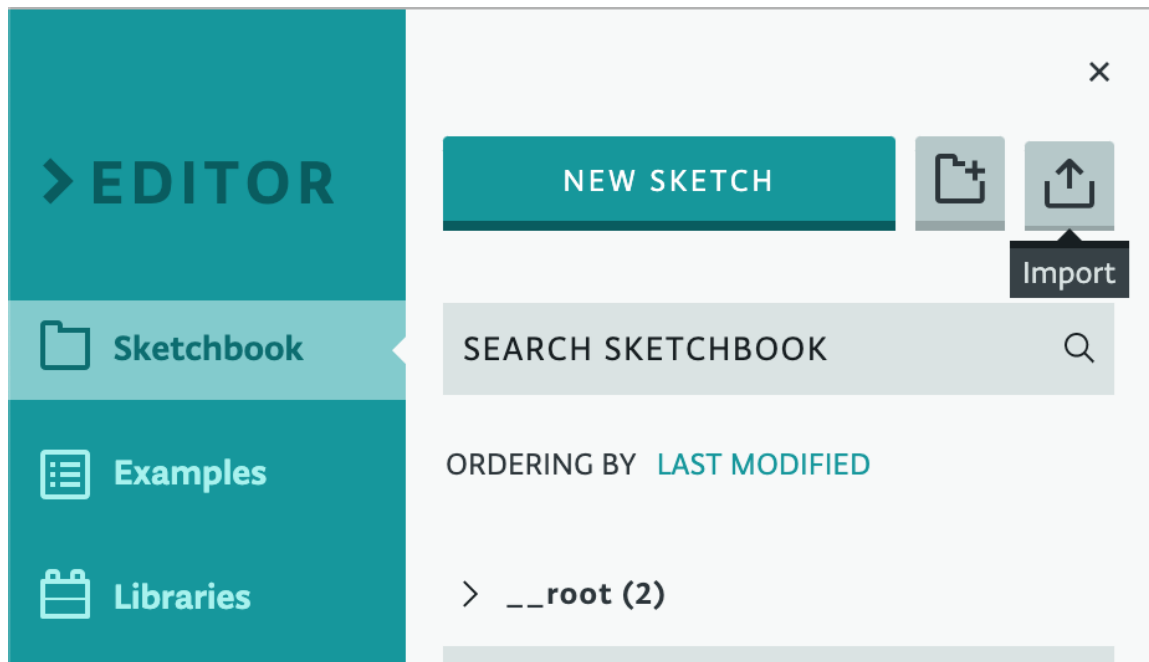
Cartella condivisa →



<https://drive.google.com/drive/folders/1w6BXQjiCxAYyjSrXLARE8yKoQRZCkbJK?usp=sharing>

COSTRUIAMO IL CODICE GRADUALMENTE

1. Colleghiamoci al WebEditor di Arduino
2. Creiamo un nuovo Sketch
3. **Importiamo lo sketch presente nella cartella condivisa sotto linkata**
(MKR1010Bot_V1_08_Piersoft_C ampustore.zip)

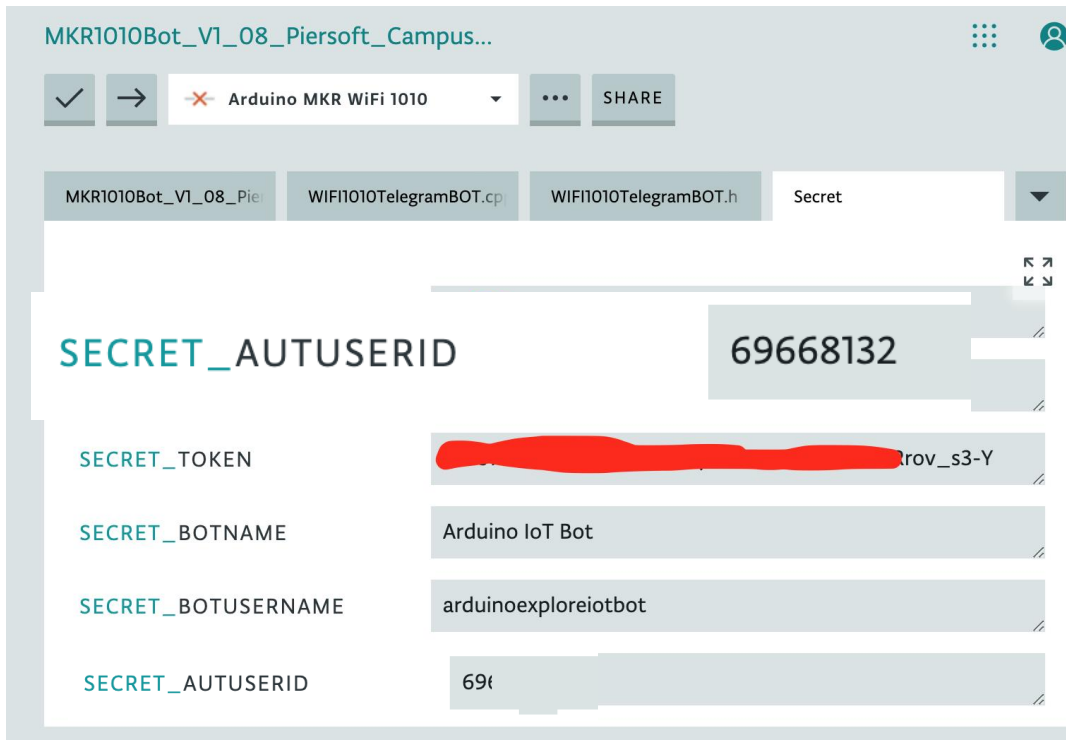


Cartella condivisa →


<https://drive.google.com/drive/folders/1w6BXQjiCxAYyjSrXLARE8yKoQRZCkbJK?usp=sharing>

COSTRUIAMO IL CODICE GRADUALMENTE

1. Compliamo la scheda “Secret” con i dati del WiFi della nostra connessione e con i dati del Bot che abbiamo creato (nome, username e TOKEN)
2. Inseriamo il nostro userid Telegram. (Per trovarlo cercare @userinfobot e lanciarlo)




MKR1010Bot_V1_08_Piersoft_Campus...

✓ →  Arduino MKR WiFi 1010 ... SHARE

MKR1010Bot_V1_08_Pie WIFI1010TelegramBOT.cp WIFI1010TelegramBOT.h Secret

SECRET_AUTUSERID 69668132

SECRET_TOKEN  rov_s3-Y

SECRET_BOTNAME Arduino IoT Bot

SECRET_BOTUSERNAME arduinoexploreiotbot

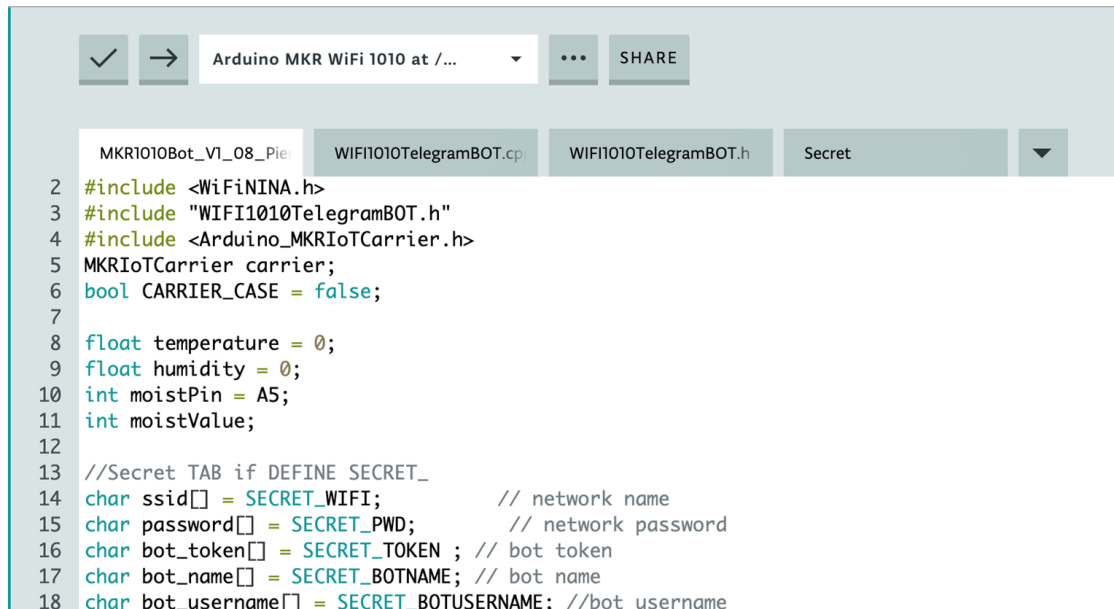
SECRET_AUTUSERID 696

Cartella condivisa →

<https://drive.google.com/drive/folders/1w6BXQjiCxAYyjSrXLARE8yKoQRZCkbJK?usp=sharing>

COSTRUIAMO IL CODICE GRADUALMENTE

1. Sono presenti delle librerie ottimizzate per l'Arduino MKR1010 in modo da usare comandi semplificati
2. Non dobbiamo cambiare nient'altro nello Sketch.



```
2 #include <WiFinINA.h>
3 #include "WIFI1010TelegramBOT.h"
4 #include <Arduino_MKRIOTCarrier.h>
5 MKRIOTCarrier carrier;
6 bool CARRIER_CASE = false;
7
8 float temperature = 0;
9 float humidity = 0;
10 int moistPin = A5;
11 int moistValue;
12
13 //Secret TAB if DEFINE SECRET_
14 char ssid[] = SECRET_WIFI;           // network name
15 char password[] = SECRET_PWD;       // network password
16 char bot_token[] = SECRET_TOKEN ; // bot token
17 char bot_name[] = SECRET_BOTNAME; // bot name
18 char bot_username[] = SECRET_BOTUSERNAME; //bot username
```

Cartella condivisa →

<https://drive.google.com/drive/folders/1w6BXQjiCxAYyjSrXLARE8yKoQRZCkbJK?usp=sharing>

COSTRUIAMO IL CODICE GRADUALMENTE

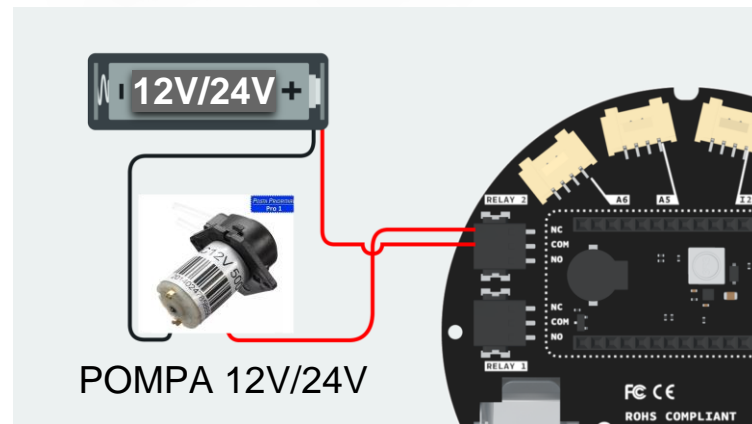
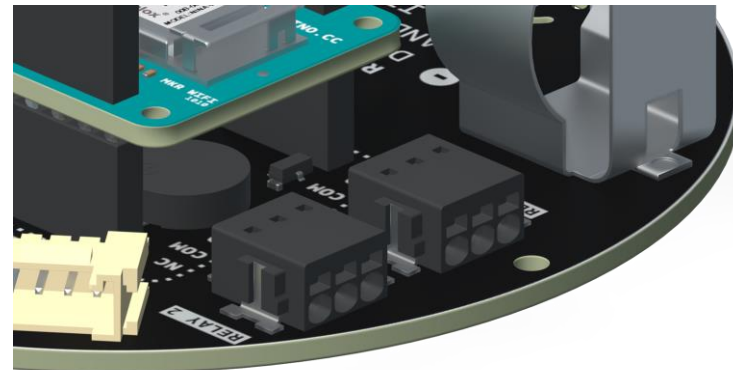
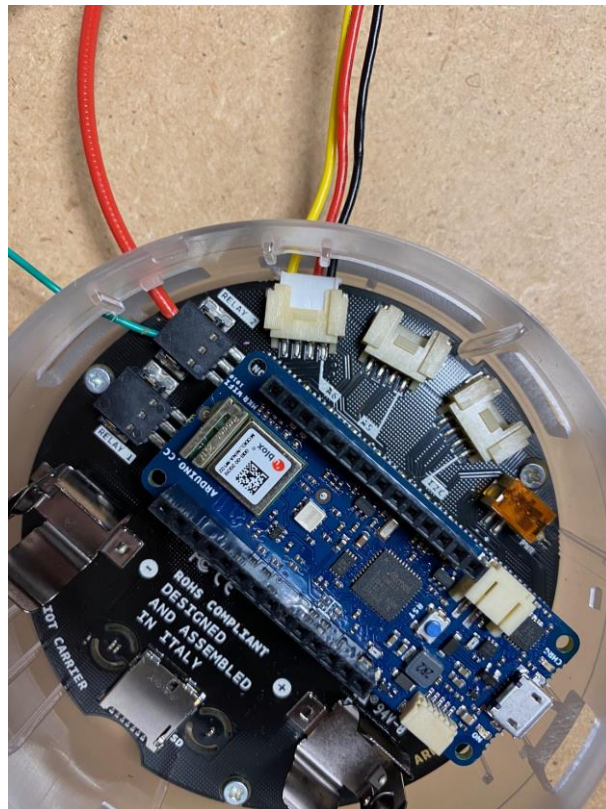


Cartella condivisa →

<https://drive.google.com/drive/folders/1w6BXQjiCxAYyjSrXLARE8yKoQRZCkbJK?usp=sharing>

COLLEGAMENTO SENSORI:

- 1) Usiamo il Relay2 colleghiamo i PIN per NO (primi due dal basso come da foto)
- 2) Colleghiamo il sensore di umidità del terreno alla prima porta in alto a sinistra come da foto
- 3) Colleghiamo una Pompa da 12/24V con un alimentatore. Il cavo positivo entra ed esce dl Relay2



Cartella condivisa →

<https://drive.google.com/drive/folders/1w6BXQjiCxAYyjSrXLARE8yKoQRZCkbJK?usp=sharing>

COSTRUIAMO IL CODICE GRADUALMENTE

1. Grazie al Carrier, la lettura della Temperatura, Umidità e del sensore umidità del terreno sono semplificati al massimo
2. Usiamo la funzione “map” per trasformare i valori grezzi in ingresso del sensore umidità del terreno (rawMoistValue) in valori da 0 a 100 simulando un’umidità relativa percentuale attraverso la variabile “moistValue”

```
void loop() {  
  // read the sensor values  
  temperature = carrier.Env.readTemperature();  
  humidity = carrier.Env.readHumidity();  
  int rawMoistValue = analogRead(moistPin);  
  moistValue = map(rawMoistValue, 0, 1023, 100, 0);  
  
  // print each of the sensor values  
  Serial.print("Temperature = ");  
  Serial.print(temperature);  
  Serial.println(" °C");  
  
  Serial.print("Humidity = ");  
  Serial.print(humidity);  
  Serial.println(" %");  
  
  Serial.print("Humidity Ground = ");  
  Serial.print(moistValue);  
  Serial.println(" %");  
}
```



Cartella condivisa →

<https://drive.google.com/drive/folders/1w6BXQjiCxAYyjSrXLARE8yKoQRZCkbJK?usp=sharing>

COSTRUIAMO IL CODICE GRADUALMENTE

1. Nella funzione loop controlliamo se c'è qualche comando inserito nella Chat di Telegram. Se dovesse esserci un messaggio richiameremo la funzione `Bot_EchoMessages()` per processarli

```
if (bot.getUpdates(bot.message[0][1]) == true) {  
    //Telegram is responding!  
    Serial.println("Bot connected!");  
    Bot_EchoMessages();           // reply to message(s) with Echo  
} else {  
    //Telegram is NOT responding!  
    Serial.println("Bot not connected...");  
    //Serial.println("Bot not connected, restarting the Bot...");  
    //NVIC_SystemReset();        // processor software reset, just in case...  
}  
  
delay(500);
```



Cartella condivisa →

<https://drive.google.com/drive/folders/1w6BXQjiCxAYyjSrXLARE8yKoQRZCkbJK?usp=sharing>

COSTRUIAMO IL CODICE GRADUALMENTE

```
216 void printDisplay(String text, uint16_t textcolor, uint16_t backgroundcolor) {  
217     //configuring display, setting background color, text size and text color  
218     carrier.display.fillScreen(backgroundcolor); //black background  
219     carrier.display.setTextColor(textcolor); //white text  
220     carrier.display.setTextSize(2); //medium sized text  
221     carrier.display.setCursor(20, 110); //sets position for printing (x and y)  
222     carrier.display.print(text);  
223     // ST77XX_BLACK, ST77XX_WHITE, ST77XX_RED, ST77XX_GREEN  
224 }  
225
```

Cartella condivisa →



<https://drive.google.com/drive/folders/1w6BXQjiCxAYyjSrXLARE8yKoQRZCkbJK?usp=sharing>

COSTRUIAMO IL CODICE GRADUALMENTE

```
void Bot_EchoMessages() {  
    for (int i = 1; i < bot.message[0][0].toInt() + 1; i++) {  
        if (bot.message[i][4] == AutUserID) {  
            // Serial.println(bot.message[i][5]);  
  
            Sender = bot.message[i][4];  
            // Serial.println(bot.message[i][4]);  
            //save the user message  
            String userMsg = bot.message[i][5];  
            userMsg.toLowerCase();  
            String userName = bot.message[i][2];  
            // Serial.println(bot.message[i][2]);  
  
            //check the user's command and take actions  
            if (userMsg == "temp") {  
                //do something for this command  
                printDisplay("Temp:"+String(temperature)+" C",ST77XX_RED,ST77XX_BLACK );  
                bot.sendMessage(Sender, String(temperature)+" C", "");  
            }  
        }  
    }  
}
```

IL BOT RISPONDE SOLO ALLO **userid** AUTORIZZATO

IL BOT RISPONDE SOLO ALL'UTENTE CHE HA SCRITTO

LEGGO IL TESTO SCRITTO DALL'UTENTE NEL BOT

PER NORMALIZZAZIONE TRASFORMO IL TESTO IN MINUSCOLO

SE L'UTENTE HA SCRITTO TEMP ALLORA ->

Cartella condivisa →

<https://drive.google.com/drive/folders/1w6BXQjiCxAYyjSrXLARE8yKoQRZCkbJK?usp=sharing>

COSTRUIAMO IL CODICE GRADUALMENTE

MKR1010Bot_V1_08_Pier

WIFI1010TelegramBOT.cp

WIFI1010TelegramBOT.h

Secret

```
else if (userMsg == "hum") {  
    //do something for this command  
    printDisplay("Hum:"+String(humidity)+"%",ST77XX_WHITE,ST77XX_BLUE );  
    bot.sendMessage(Sender, String(humidity)+"%", "");  
}  
else if (userMsg == "led_on") {  
    //do something for this command  
    //first pixel, green  
    carrier.leds.setPixelColor(0, 255, 0, 0);  
    //second pixel, red  
    carrier.leds.setPixelColor(1, 0, 255, 0);  
    //third pixel, blue  
    carrier.leds.setPixelColor(2, 0, 0, 255);  
    //fourth pixel, purple  
    carrier.leds.setPixelColor(3, 255, 0, 255);  
    //fifth pixel, no color  
    carrier.leds.setPixelColor(4, 0, 0, 0);  
  
    //update the pixels  
    carrier.leds.show();  
    bot.sendMessage(Sender, "Comando eseguito!", "");
```

HUM → DISPLAY + BOT

ACCENDO LED CARRIER

MANDO MSG AL BOT




Cartella condivisa →

<https://drive.google.com/drive/folders/1w6BXQjiCxAYyjSrXLARE8yKoQRZCkbJK?usp=sharing>

COSTRUIAMO IL CODICE GRADUALMENTE

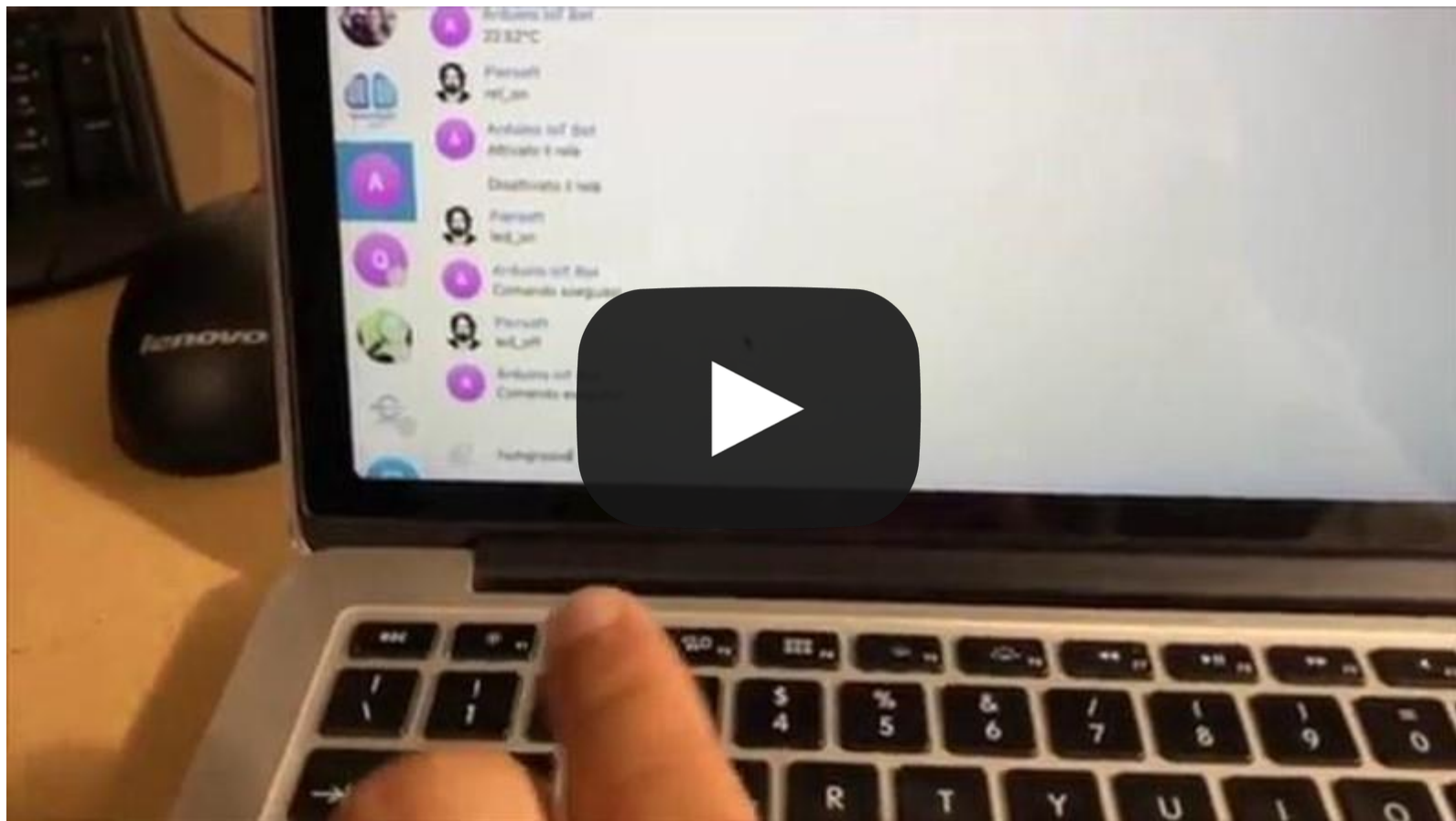
```
}  
else if (userMsg == "rel_on") {  
    //do something for this command  
    carrier.Relay2.open(); ECCITO IL RELE'  
    bot.sendMessage(Sender, "Attivato il relè", "");  
    delay(5000); // dopo 5 secondi spengo se devo innaffiare piantina :)  
    carrier.Relay2.close(); ATTENDO 5 SECONDI PRIMA DI SPEGNERLO  
    bot.sendMessage(Sender, "Disattivato il relè", "");  
}
```



Cartella condivisa →



<https://drive.google.com/drive/folders/1w6BXQjiCxAYyjSrXLARE8yKoQRZCkbJK?usp=sharing>



CONCLUSIONI



Ora possiamo inviare comandi (ma solo noi!!) al nostro Arduino IOT Kit dovunque siamo nel mondo.

I comandi utilizzabili sono:

**/start, led_on, led_off, rel_on,
rel_off, led_status, rel_status, temp,
hum, humground**





**Mai memorizzare quello
che puoi comodamente
trovare in un libro.
Albert Einstein (1879-1955)**

ARDUINO EXPLORE IoT KIT

SPERIMENTAZIONE DIDATTICA



GRAZIE MILLE!
@piersoft

ARDUINO EXPLORE IoT KIT

SPERIMENTAZIONE DIDATTICA