

BENVENUTI!

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



SPERIMENTIAMO CON IL CARRIER

Utilizzando l'MKR IoT Carrier all'interno della custodia, collegheremo sensori esterni, utilizzando dispositivi ad alta tensione attraverso i relè, impareremo come collegare una batteria e le batterie compatibili con il kit

2|12|20 con Domenico Aprile

4 APPUNTAMENTI IMPERDIBILI

- **5 tra i massimi esperti italiani** di didattica, robotica, elettronica, coding e open data
- Anche se non potrai partecipare in diretta, iscrivendoti ti assicurerai di ricevere le **videoregistrazioni** e poterne **fruire in differita** in qualsiasi momento
- **Link di iscrizione** unico a tutti gli appuntamenti



DOVE TROVARE L'EXPLORE IOT KIT



Arduino Explore IoT Kit Singolo

Codice: 333190

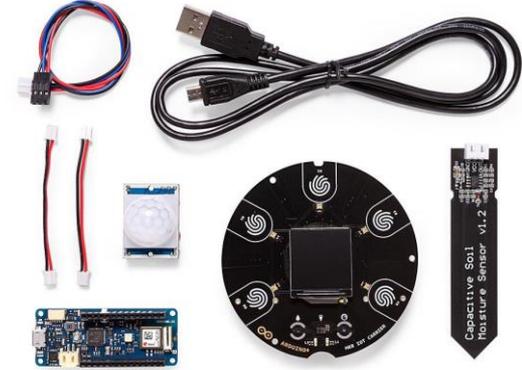
Codice MEPA: 333190CS



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

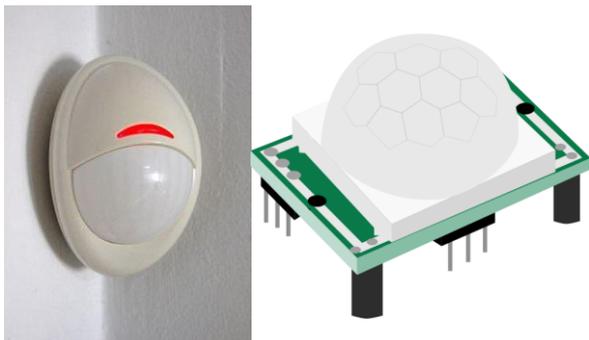
 Gruppo Privato · 726 membri

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**”



I sensori

- Trasduttori
- Input per la board
- Trasformano in segnale elettrico una diversa tipologia di segnale



ALLARME PER LA CASA

- USARE UN SENSORE ESTERNO
- PIN PER ALTA TENSIONE & RELE'

Didatticamente

IMPAREREMO DI PIÙ SU:

SENSORE DI MOVIMENTO

SISTEMI AUTOMATIZZATI

MONITORARE I SENSORI DA REMOTO USANDO IL CLOUD

CONSIDERAZIONI SULLA SICUREZZA DI UN NETWORK

ISTRUZIONI CONDIZIONALI

Obiettivi didattici

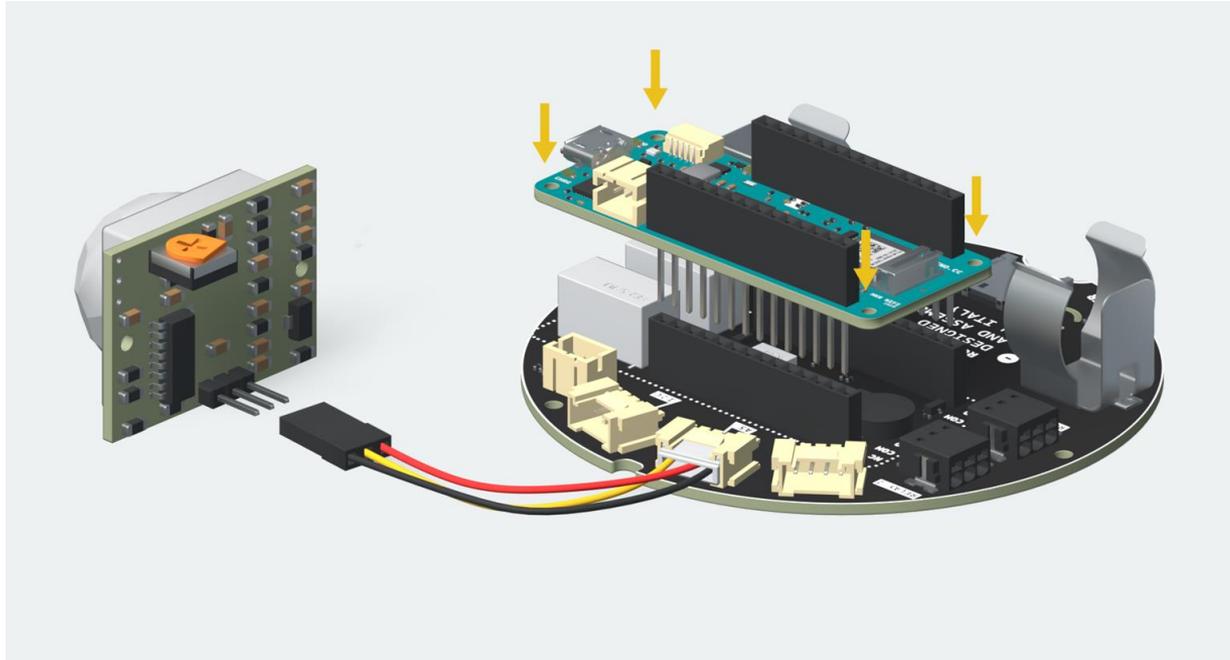
Gli obiettivi di questa attività sono:

- ◇ Imparare a conoscere i sensori di movimento PIR
- ◇ Esplorare le problematiche della sicurezza della rete
- ◇ Utilizzo della proprietà Boolean e String di Arduino IoT Cloud



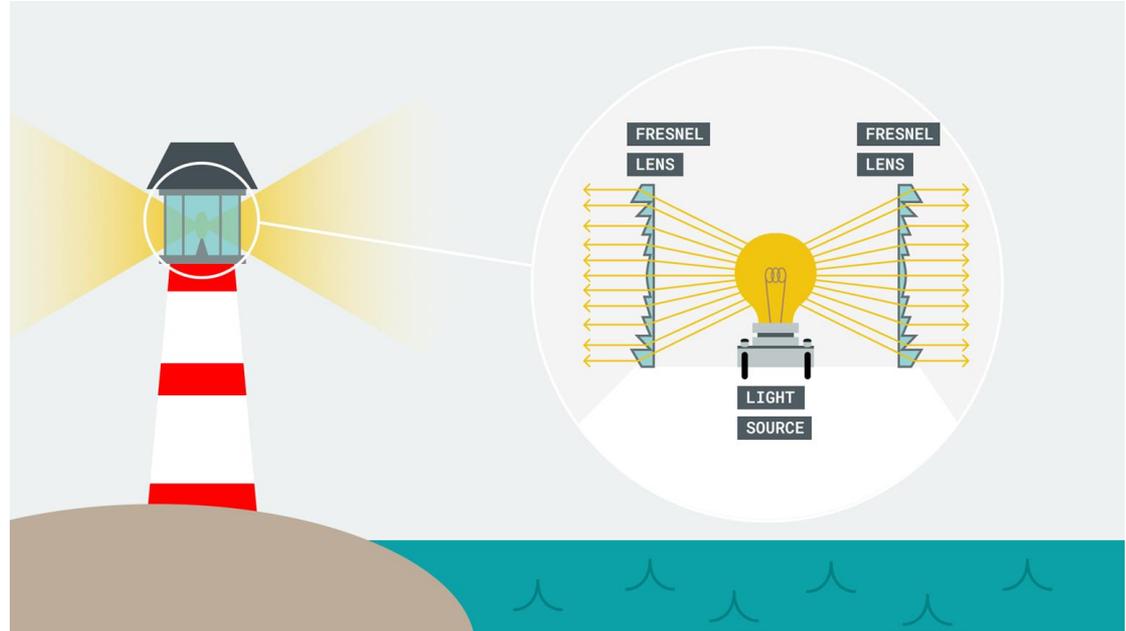
PIR → PASSIVE INFRARED SENSOR

I sensori a infrarossi passivi (PIR) sono componenti elettronici utilizzati per rilevare il movimento. Misurano la luce infrarossa che irradia qualsiasi oggetto che si trova di fronte. Questo modifica la tensione di uscita, che il sensore è in grado di misurare.



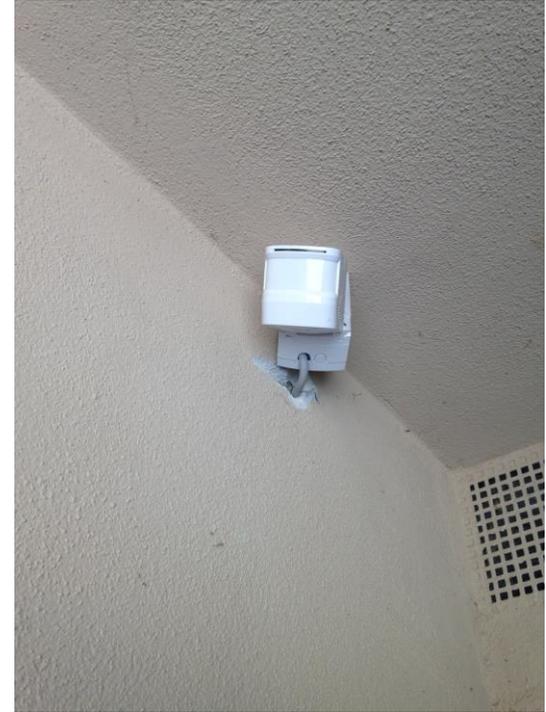
PIR → SENSORE A INFRAROSSI

Questo si ottiene attraverso l'uso di una cosa chiamata lente di **Fresnel**, che permette al sensore di rilevare un raggio d'azione molto più ampio di un semplice sensore. Tuttavia, i sensori di movimento PIR non possono dirci la distanza di una persona o quante persone si trovano nel suo campo visivo.



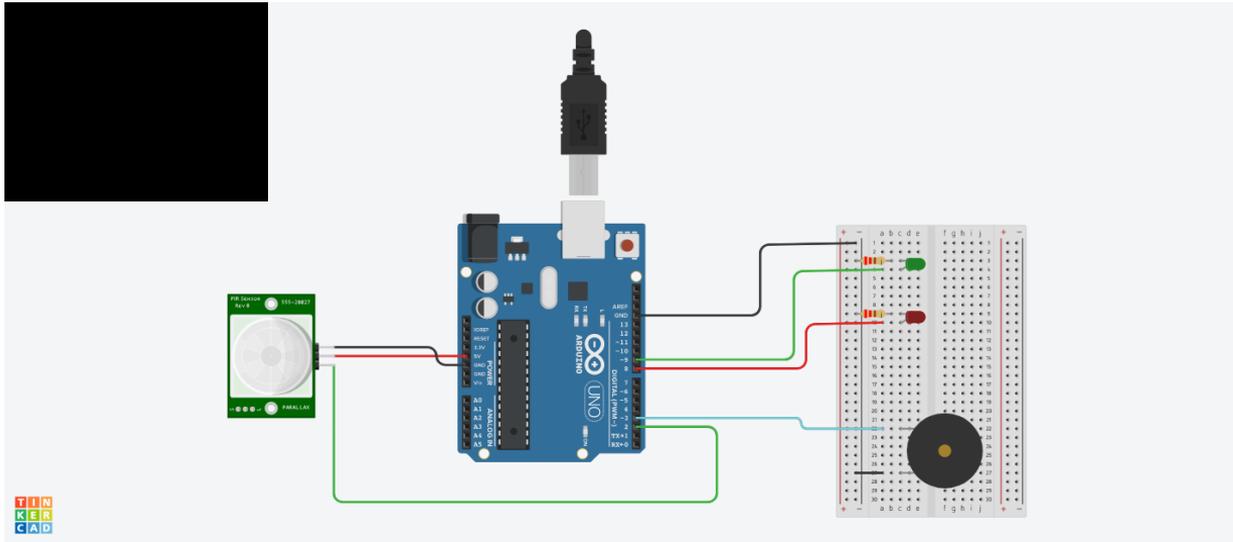
Applicazioni PIR

- Allarmi volumetrici domestici
- Accensione temporizzata luci ambienti (es: bagni locali pubblici)
- Animali domestici (chiusura temporizzata vassoio cibo)
- Intrusione animali selvatici



Potremmo partire da qui...

<https://screencast-o-matic.com/watch/cYXtjELLvV>



```
#define PIR_PIN 2
#define LEDRED_PIN 8
#define LEDGREEN_PIN 9
#define buzzer_pin 3

bool state = false;

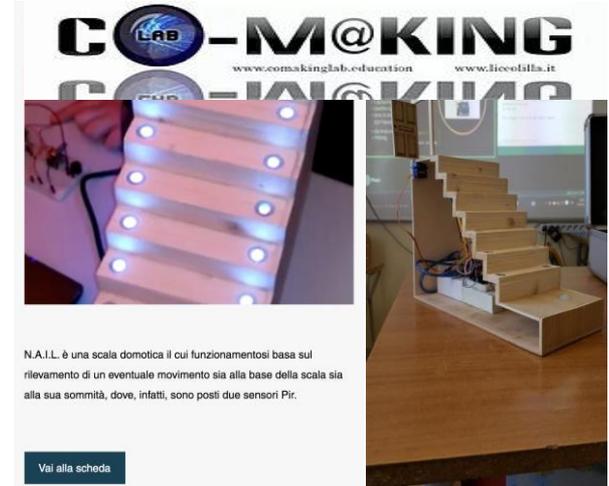
void setup(){
  pinMode(LEDRED_PIN, OUTPUT);
  pinMode(LEDGREEN_PIN, OUTPUT);
  pinMode(PIR_PIN, INPUT);
  pinMode(buzzer_pin, OUTPUT);
  digitalWrite(LEDGREEN_PIN, HIGH);
}

void loop(){
  if(digitalRead(PIR_PIN) == HIGH){
    if(!state){
      digitalWrite(LEDRED_PIN, HIGH);
      digitalWrite(LEDGREEN_PIN, LOW);
      tone(buzzer_pin, 1000);
      state = true;
    }
  }else{
    if(state){
      digitalWrite(LEDRED_PIN, LOW);
      digitalWrite(LEDGREEN_PIN, HIGH);
      noTone(buzzer_pin);
      state = false;
    }
  }
  delay(10);
}
```



Un progetto basato sul PIR

- **N.A.I.L. - New Auto Illuminated Ladder**
 - scala domotica basata sul rilevamento di un eventuale movimento
 - due sensori Pir (uno in cima, l'altro alla base)
 - In base al movimento rilevato, si accendono in modo sequenziale consentendo una salita/discesa in sicurezza
 - [LINK AL PROGETTO](#)



TESTIAMO IL SENSORE PIR

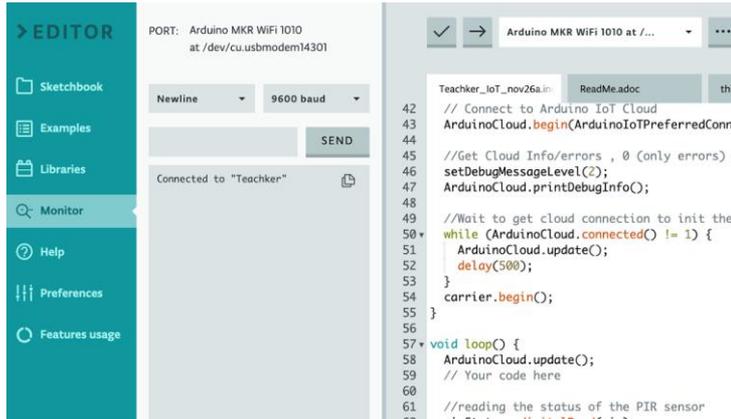


```
int pir = A5;  
bool pirState = HIGH;
```

```
setup()  
  pinMode(pir, INPUT);
```

```
loop()  
  pirState = digitalRead(pir);
```

Allarme per la casa



Home security Alarm

 boolean_ala...

ON

Teachker_IoT

 string_message

Mon 30 Nov

Oh oh...mi è sembrato di vedele un gatto!

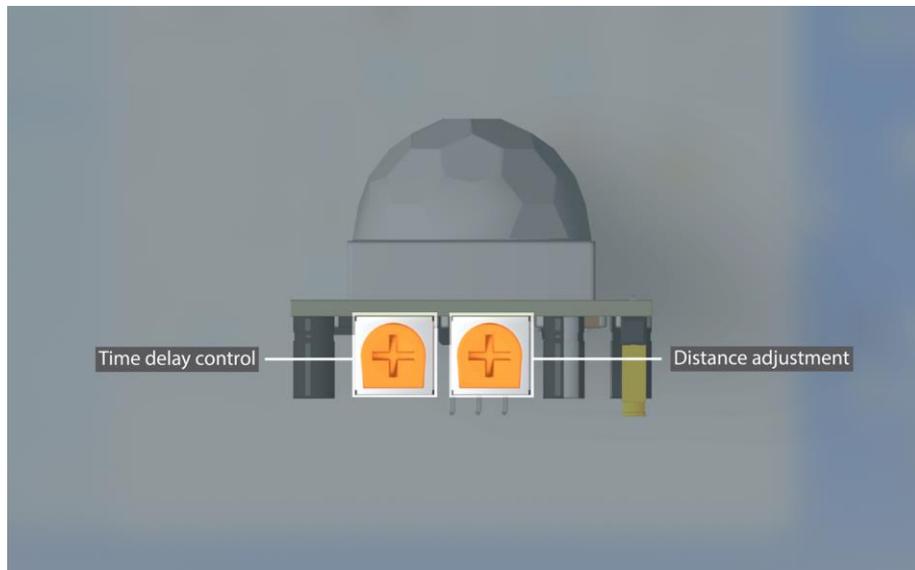
20:06

Teachker_IoT



CONFIGURIAMO IL SENSORE PIR

Possiamo configurare il comportamento della risposta del sensore ruotando i due potenziometri sul retro. Il potenziometro a destra nell'immagine sottostante imposta la distanza in cui il sensore va a rilevare il movimento. Il potenziometro a sinistra cambia il tempo di risposta quando rileva i movimenti e imposta il pin di uscita su HIGH.



AGRICOLTURA URBANA

101

- SENSORE DI UMIDITÀ DEL TERRENO
- AUTOMATIZZARE IL SISTEMA

Didatticamente

IMPAREREMO DI PIÙ SU:

RELÈ

SENSORE DI UMIDITÀ

MISURAZIONI SULL'ECOSISTEMA

MAPPARE VALORI

PROGETTARE ECOSISTEMI AUTOMATIZZATI

Obiettivi didattici

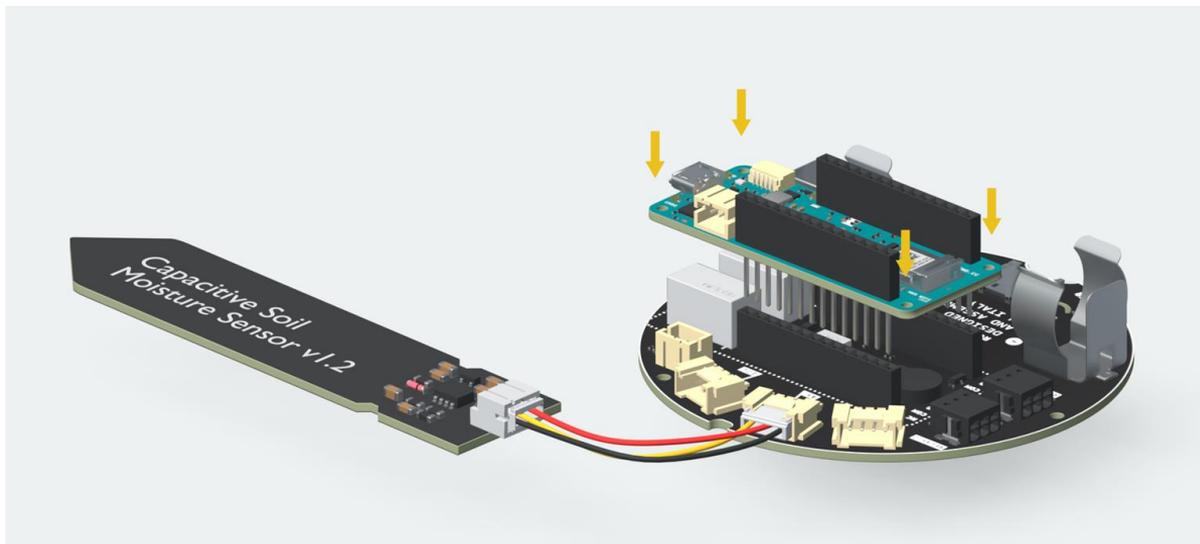
Gli obiettivi di questa attività sono:

- ◇ Creare un ambiente agricolo urbano
- ◇ Capire come funzionano i relè
- ◇ Capire come funziona il sensore di umidità
- ◇ Creare un ambiente ideale per una pianta



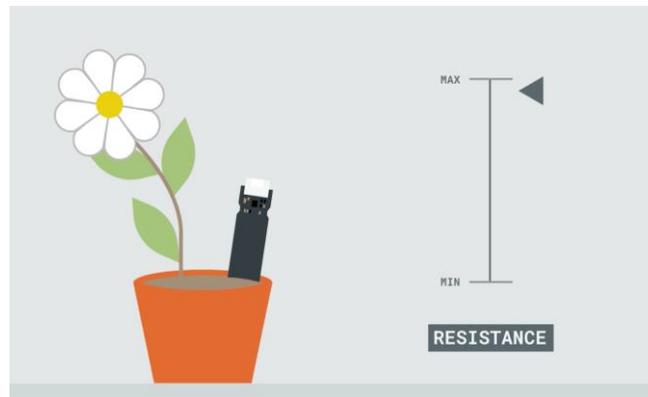
SENSORE DI UMIDITÀ

Il sensore di umidità è uno dei più elementari, ma potenti sensori in circolazione. Ha due conduttori esposti ed è fondamentalmente una resistenza variabile. Quando è esposto all'acqua, la resistenza diminuisce, poiché la conducibilità aumenta, mentre meno acqua significa maggiore resistenza.



SENSORE DI UMIDITÀ

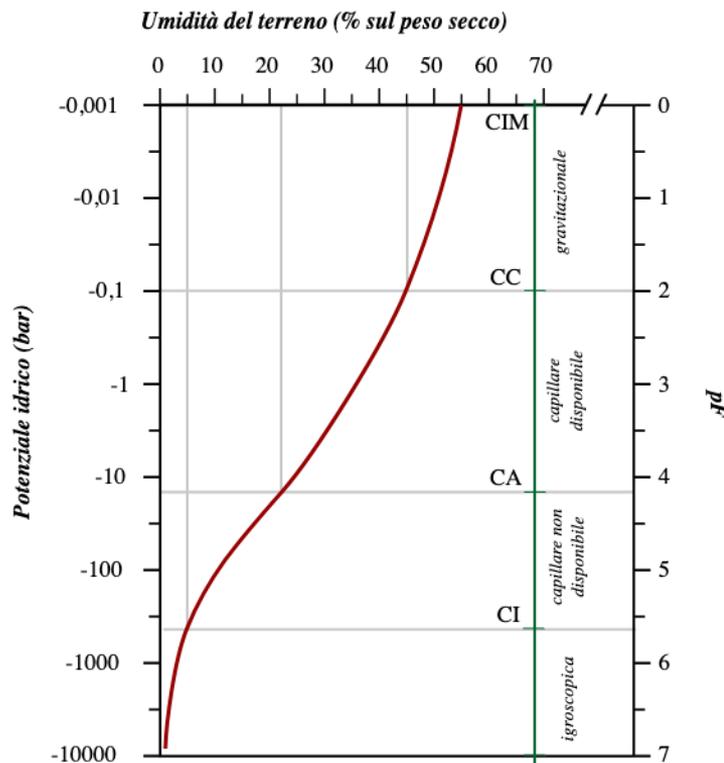
Se mettiamo questo sensore in un vaso per piante che è stato innaffiato di recente, la resistenza è inferiore, e se effettuiamo una lettura, potremo misurare quanto è umida la pianta. Ma se mettiamo il sensore in un vaso per piante che non è stato annaffiato da giorni, molto probabilmente mostrerà 1023 (completamente asciutto). Questo ci dà un feedback immediato: dobbiamo annaffiare le nostre piante, altrimenti non sopravvivranno!



Applicazioni sensore umidità terreno

Water retention curve

Fonte img: [Wikimedia CC-BY-SA 3.0](https://commons.wikimedia.org/wiki/File:Water_retention_curve.png)



Un progetto di serra automatizzata

- **S.A. - La Serra del Domani**
 - sistema di ventilazione, illuminazione, sensori di umidità e temperatura, sensori di umidità del terreno e sistema d'irrigazione.
 - display lcd 128*64 con implementazione di menù
 - sistema di controllo manuale (attraverso il menù) o automatizzata
 - i sensori rilevano i valori monitorati ed attivano gli attuatori della serra.
 - [LINK AL PROGETTO](#)

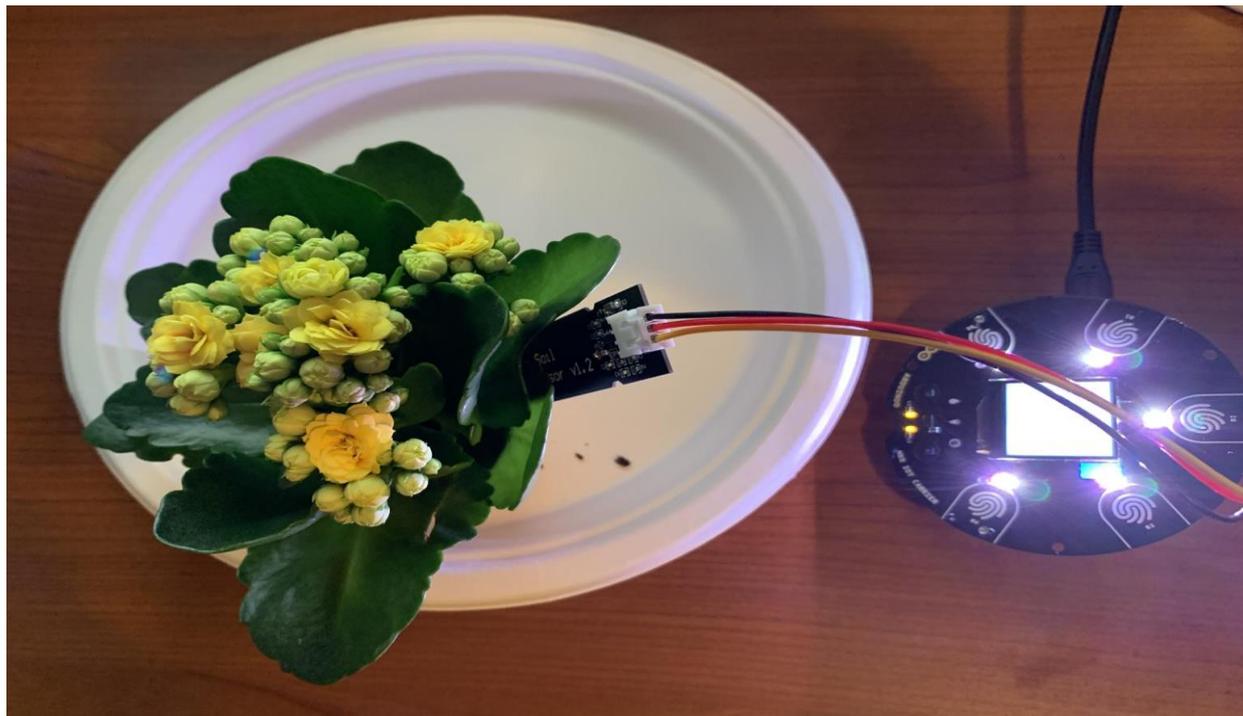


Il progetto consiste in una serra automatizzata, munita di sistema di ventilazione, illuminazione, sensori di umidità e temperatura, sensori di umidità del terreno e sistema d'irrigazione.

Vai alla scheda



Il circuito



L'IoT cloud di Arduino

YOUR THINGS

Arduino IoT Cloud allows you to connect devices to the internet and to other devices. This tool makes the creation of connected objects quick, simple and secure. Read more or check out our step-by-step tutorial. You might also look at our FAQ.

ADD NEW THING

Teacher IoT

Properties: 2
Created: 26 Nov 2020, 11:02
Modified: 26 Nov 2020, 11:02

Urban_Farming

Properties: 8
Created: 30 Nov 2020, 17:23
Modified: 30 Nov 2020, 17:23

Urban_Farming

Last synced a few seconds ago

[EDIT SKETCH](#)

Properties
Webhooks
Associated Device

NAME	TYPE	UPDATE	PERMISSION
humidity	Float	On change	RO
light	Int	On change	RO
moistValue	Int	On change	RO
relay1	ON/OFF (Boolean)	On change	R&W
relay2	ON/OFF (Boolean)	On change	R&W
rgbColor	Colored light	On change	R&W
temperature	Float	On change	RO
updateDisplay	ON/OFF (Boolean)	On change	R&W

Thing ID: 888c24e7-5ed1-414e-a930-20a366d47E85

[Report Bug](#)

Devices

[ADD DEVICE](#)

All device types All device status

Name	Status	Linked Thing
Teacher Arduino MKR WiFi 1010	● Online	Urban_Farming



L'editor IoT

PORT: Arduino MKR WiFi 1010
at /dev/cu.usbmodem14301

Newline 9600 baud

SEND

```
***** Arduino IoT Cloud -
configuration info *****
Device ID: 9fad9e0e-9b9d-431f-8e2e-
b988398a1308
Thing ID: 888c24e7-5ed1-414e-a930-
20a366d47f85
MQTT Broker: mqtt-
sa.iot.arduino.cc:8883
WiFi.status(): 0
Current WiFi Firmware: 1.4.1
Connected to "Teacher"
Sd card not detected
Connected to Arduino IoT Cloud
ArduinoIoTCLOUDTCP::handle_Connected
MQTT client connection lost
Connected to Arduino IoT Cloud
```

EDITOR

PORT: Arduino MKR WiFi 1010
at /dev/cu.usbmodem14301

Newline 9600 baud

SEND

```
***** Arduino IoT Cloud -
configuration info *****
Device ID: 9fad9e0e-9b9d-431f-8e2e-
b988398a1308
Thing ID: 888c24e7-5ed1-414e-a930-
20a366d47f85
MQTT Broker: mqtt-
sa.iot.arduino.cc:8883
WiFi.status(): 0
Current WiFi Firmware: 1.4.1
Connected to "Teacher"
Sd card not detected
Connected to Arduino IoT Cloud
ArduinoIoTCLOUDTCP::handle_Connected
MQTT client connection lost
Connected to Arduino IoT Cloud
```

Urban_Farming_nov30a

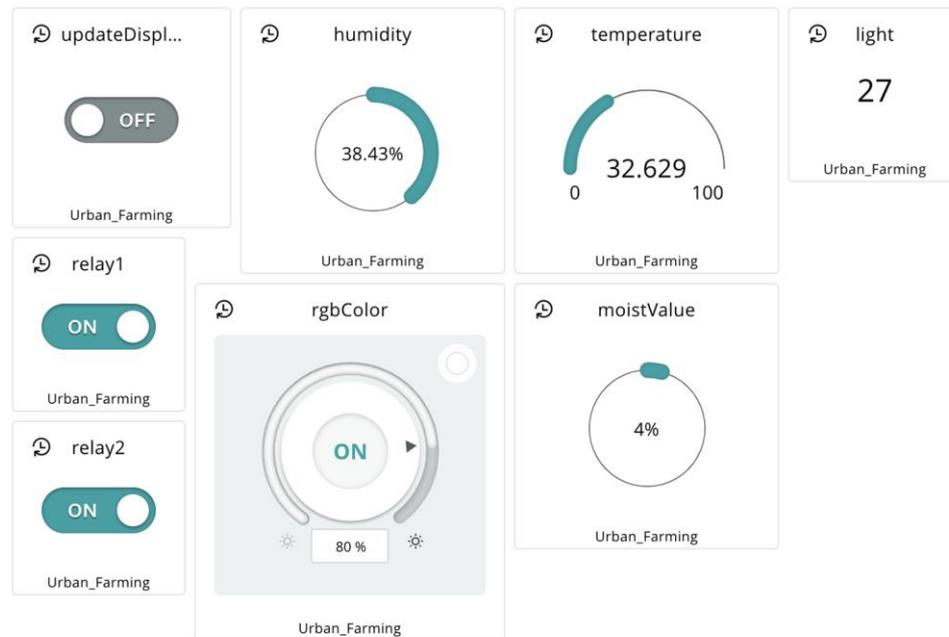
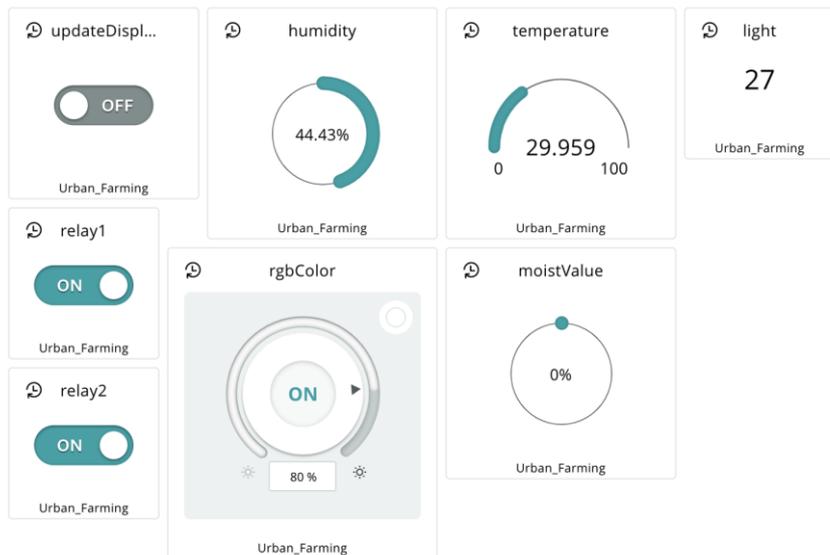
Urban_Farming_nov30a: ReadMe.adoc thingProperties.h Secret

```
1 //
2 Sketch generated by the Arduino IoT Cloud Thing "Urban_Farming"
3 https://create.arduino.cc/cloud/things/888c24e7-5ed1-414e-a930-20a366d47f85
4
5 Arduino IoT Cloud Properties description
6
7 The following variables are automatically generated and updated when changes are made to the Thing properties
8
9 Float humidity;
10 Float temperature;
11 CloudColoredLight rgbColor;
12 int light;
13 int moisture;
14 bool relay1;
15 bool relay2;
16 bool updatedisplay;
17
18 Properties which are marked as READ/WRITE in the Cloud Thing will also have functions
19 which are called when their values are changed from the Dashboard.
20 *These functions are generated with the Thing and added at the end of this sketch.
21 */
22
23 #include "thingProperties.h"
24 #include <Arduino_MKR101Carrier.h>
25 MKR101Carrier carrier;
26 bool CARRIER_CASE = false;
27
28 int moistPin = A5;
29
30 String relayState1 = "";
31 String relayState2 = "";
32
33 void setup() {
34   // Initialize serial and wait for port to open:
35   Serial.begin(9600);
36   // This delay gives the chance to wait for a Serial Monitor without blocking if none is found
37   delay(1500);
38   while (!Serial);
39
40   // Defined in thingProperties.h
41   initProperties();
42   // Connect to Arduino IoT Cloud
43   ArduinoCloud.begin(ArduinoIoTPreferredConnection);
44
45   //Get Cloud Info/errors . 0 (Only errors) up to 4
46   setDebugMessageLevel(0);
47   ArduinoCloud.printDebugInfo();
48
49   //Wait to get cloud connection to init the carrier
50   while (ArduinoCloud.connected() != 1) {
51     ArduinoCloud.update();
52     delay(500);
53   }
54 }
55
56 Success: Done uploading Urban_Farming_nov30a
57
58 checksum: (start_addr=0x26000, size=0x1000) = 484e
59 checksum: (start_addr=0x27000, size=0x1000) = 745d
60 checksum: (start_addr=0x28000, size=0x1000) = ecf
```



La Dashboard

Urban Farming



Ma ricordiamoci che...

<<Per costruire una serra, bisogna conoscere le piante!>>

Luca Scalzullo

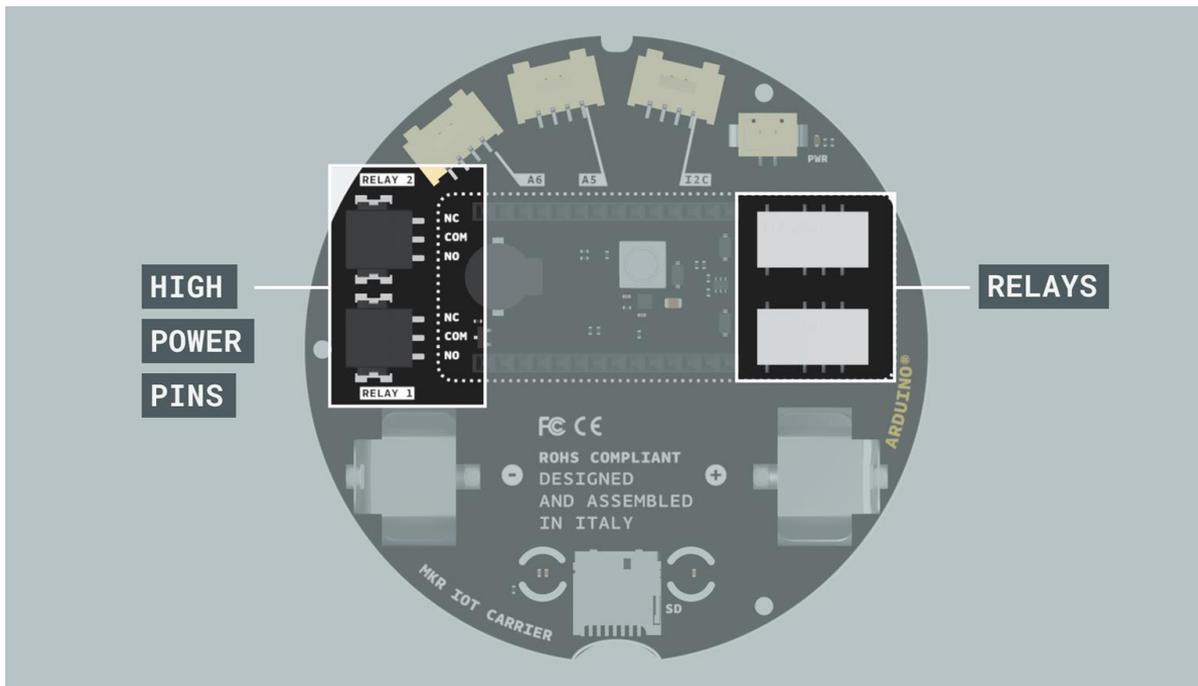
Insegnante di Tecnologia, Ingegnere chimico, (edu)maker.



APPROFONDIAMO

- PIR ALTA TENSIONE E RELE'
- AUTOMATIZZARE IL SISTEMA
- CUSTODIA DEL CARRIER

PIN ALTA TENSIONE & RELÉ'



PIN ALTA TENSIONE & RELE'

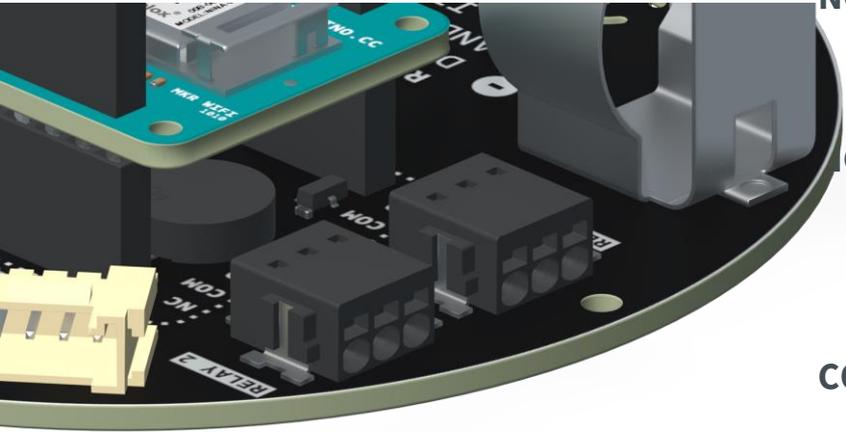
I relè consentono di utilizzare una scheda Arduino per controllare circuiti di potenza superiori a quelli che la scheda Arduino è in grado di controllare. Sono tipicamente utilizzati in applicazioni industriali per controllare circuiti ad alta potenza, ma sono anche utilizzati in auto, case e altre applicazioni elettriche. Un esempio molto comune di impiego dei relè è l'indicatore di direzione (freccia) sulle automobili. Quando il conducente dell'auto attiva la spia luminosa, possiamo sentire un suono di "ticchettio", quel suono è prodotto da un relè che accende e spegne la luce.

ECCO PERCHÉ SENTIAMO "CLICK----CLACK" OGNI VOLTA CHE ACCENDIAMO IL CARRIER.



SETTING UP HIGH POWER PINS & RELAYS

Ora che sappiamo come funzionano i relè e come effettuare le connessioni su di essi, diamo un'occhiata alle definizioni dei pin ad alta potenza:



NO - Sta per *normalmente aperto* (open). Ciò significa che quando abbiamo uno stato HIGH al relè, il pin NO è collegato con COM.

C - Sta per *normalmente chiuso*. Ciò significa che quando scriviamo uno stato LOW al relè, quando il relè non è alimentato, il pin NC è collegato con COM.

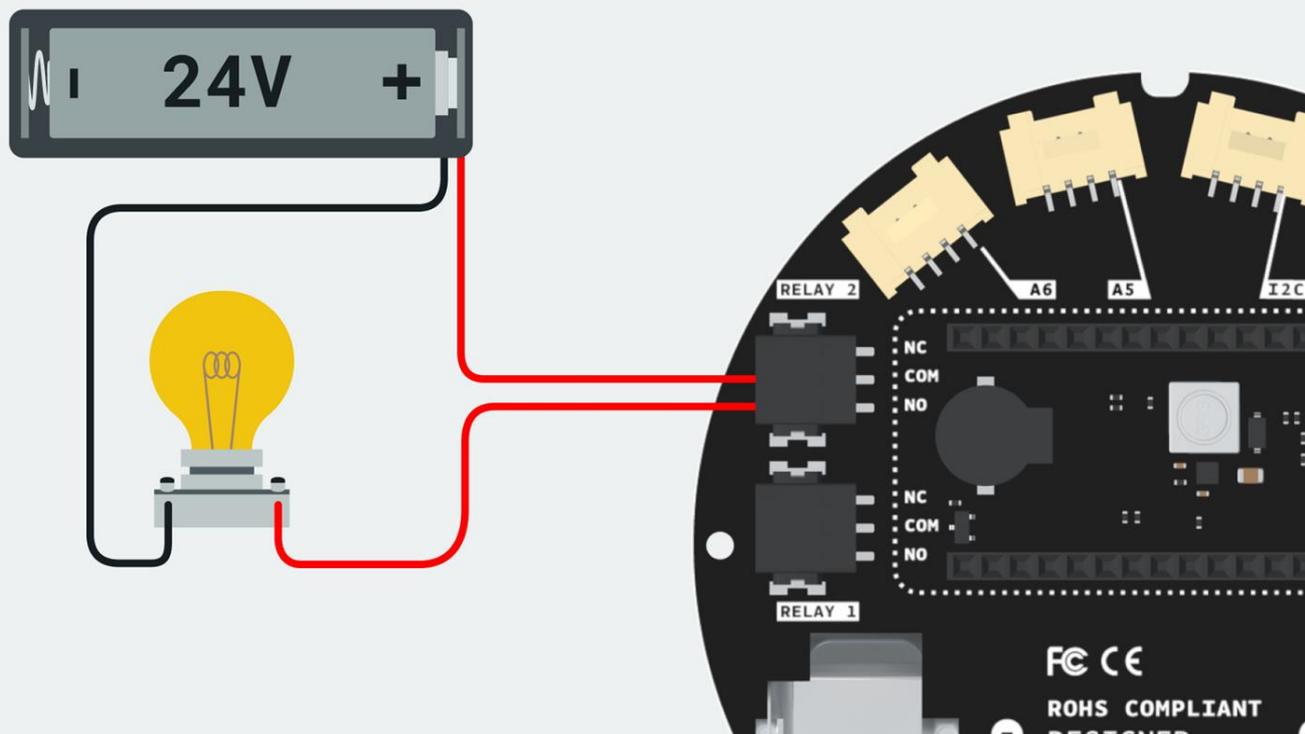
COM - Sta per *comune* e viene utilizzato come interruttore in un relè. Quando un segnale HIGH o LOW viene scritto sul relè, il pin comune sarà o fra NO o NC a seconda della configurazione che stiamo usando.



PIN ALTA TENSIONE & RELE'

IMPORTANTE!

I relè sullo schermo supportano solo fino a 24V, il che significa che non sono destinati a funzionare con tensioni superiori.



CUSTODIA DEL CARRIER

È un "extra" e il suo scopo principale è quello di proteggere l'HW quando viene installato in un progetto che deve durare a lungo.





APPROFONDIAMO

AUTOMATIZZARE IL SISTEMA

AUTOMATIZZARE IL SISTEMA

A questo punto vediamo il codice completo per gestire i sensori di temperatura, umidità e per il suolo; controllare i relè e far funzionare ventilatori e lampade in una piccola serra fai da te.

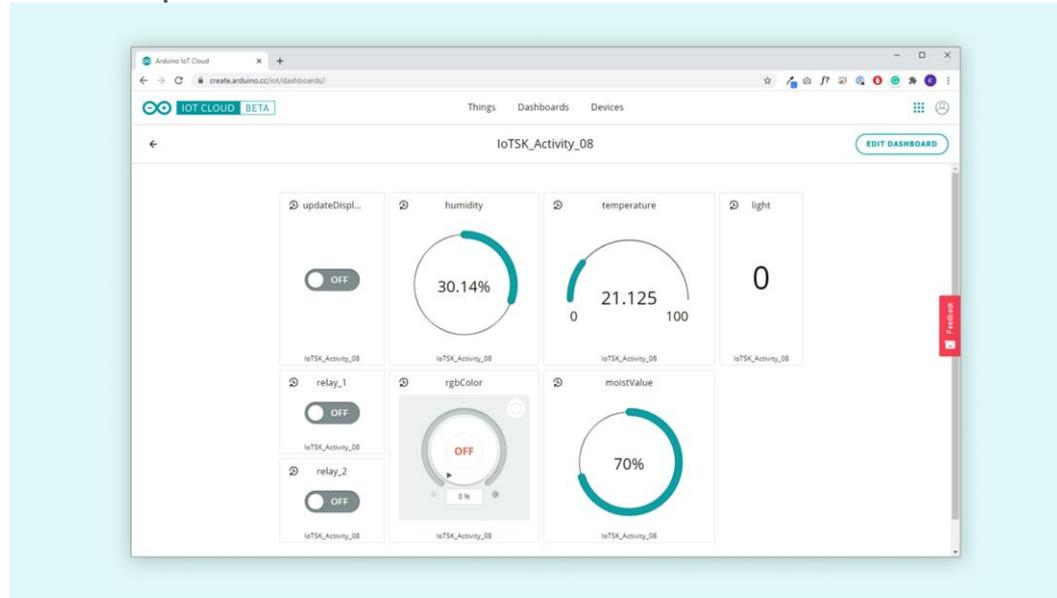
```
/*  
Sketch generated by the Arduino IoT Cloud Thing "IoTSK_Activity_08"  
https://create.arduino.cc/cloud/things/4c178f1c-a581-49bc-8123-4f6d8f21d4b2  
  
Arduino IoT Cloud Properties description  
  
The following variables are automatically generated and updated when changes are made to the Thing properties  
  
String plant_feeling;  
float humidity;  
float light;  
float temperature;  
CloudColoredLight rgbColor;  
int moistValue;  
bool relay_1;  
bool relay_2;  
bool updateDisplay;  
  
Properties which are marked as READ/WRITE in the Cloud Thing will also have functions  
which are called when their values are changed from the Dashboard.  
These functions are generated with the Thing and added at the end of this sketch.  
*/
```

<https://create.arduino.cc/editor/ArduinoEdu/cca176d7-5a1c-45fa-803c-2351ec8a304b/preview>



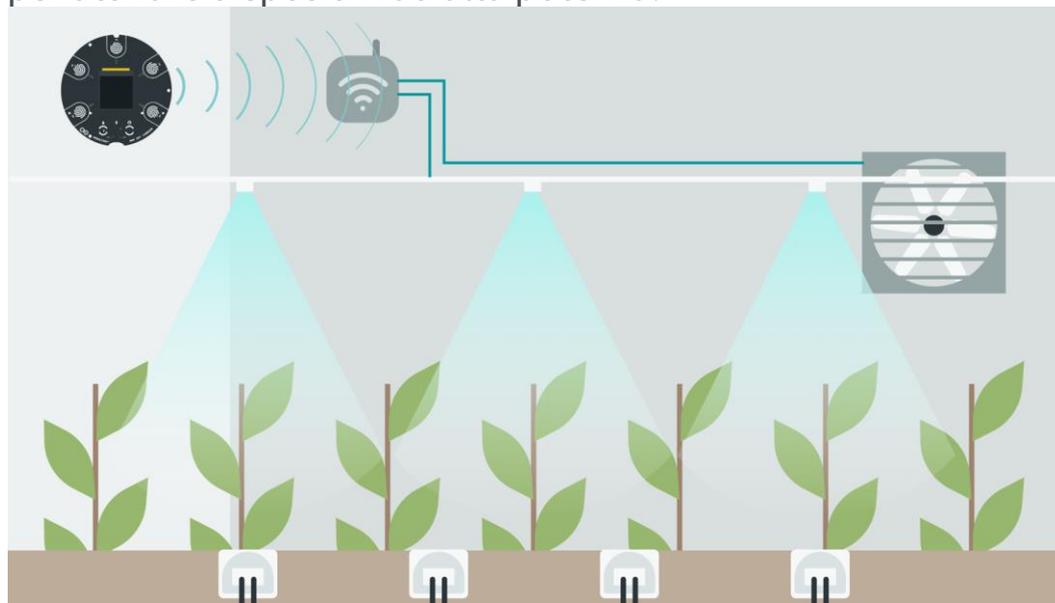
TESTIAMO L'AUTOMATIZZAZIONE IL SISTEMA

In questo dashboard, abbiamo *quattro widget* utilizzati per monitorare i sensori, e quattro proprietà utilizzate per controllare diversi attuatori. Ora possiamo posizionare l'MKR IoT Carrier nell'involucro di plastica, e collocarlo vicino ad un impianto che vogliamo monitorare! In questa attività, useremo la pianta di aloe vera come esempio.



TESTIAMO L'AUTOMATIZZAZIONE IL SISTEMA

Nell'attività 8, abbiamo trasformato il nostro MKR IoT Carrier in un dispositivo per l'agricoltura urbana! Abbiamo utilizzato i sensori del carrier e un sensore di umidità per analizzare le condizioni ambientali di una pianta, abbiamo utilizzato l'illuminazione artificiale e introdotto dei relè - un componente elettronico utilizzato per attivare dispositivi ad alta potenza.



LE SFIDE



PIR CHALLENGE

Un allarme è un sistema complesso.

Può essere ad uso domestico o industriale.

Quindi, a mero titolo di esempio:

- Aggiungere un allarme sonoro (buzzer), magari modulando il tono con la funzione **tone()**
- Aggiungere dei led che identifichino lo stato dell'allarme
- Inserire più sensori pir per sezionare le aree dell'appartamento
- Inserire un servomotore per simulare la chiusura a tenuta stagna di una zona ad alto rischio potenziale
- Usare una app per device mobile per costruire un sistema di controllo remoto



SENSORE DI UMIDITA' CHALLENGE

Si potrebbe aggiungere alla serra un po' di...umanità!

Per esempio:

- creare una proprietà che definisce lo “stato d’animo” della pianta
- codificare l’espressione delle “emozioni” della pianta, in base alla sua necessità di ricevere acqua



CONCLUSIONI

- **DOMANDE E RISPOSTE**
- **PROSSIMI APPUNTAMENTI**

I PROSSIMI APPUNTAMENTI

- **I primi passi con IoT** con Francesco Piersoft Paolicelli - Mercoledì **9 dicembre 2020** - 16:00-17:00





GRAZIE MILLE!

ARDUINO EXPLORE IoT KIT

SPERIMENTAZIONE DIDATTICA