



**Corso di
Tecnologie Elettriche per l'Informatica Industriale
a.a. 2017/2018**

Relazione del Progetto Finale

Autori	Coppola Carlo Antonio Ferrentino Alessandro
Gruppo n.	
Data della prova	16/07/2018
Data di consegna	16/07/2018
Versione del Documento	3.0

INDICE

1. OBIETTIVI DELLA PROVA	2
2. SCHEMI	2
3. ELENCO DEI COMPONENTI.....	4
4. STRUMENTI UTILIZZATI.....	7
5. DESCRIZIONE DELLE MISURE EFFETTUATE.....	8
6. RISULTATI	9
7. CONCLUSIONI.....	12

1. Obiettivi della prova

L'obiettivo della prova è realizzare un sistema di automazione per la gestione di un giardino indoor.

Il sistema si basa sull'irrigazione del giardino in base al periodo dell'anno e ai parametri di umidità e temperatura dell'aria, umidità del terreno e luminosità.

Per realizzare il sistema si è fatto uso di tutti i componenti e gli strumenti elencati nelle apposite sezioni.

2. Schemi

- In Figura 2.1 è rappresentato lo schema circuitale necessario al caricamento della logica di controllo sul microcontrollore ATtiny84.

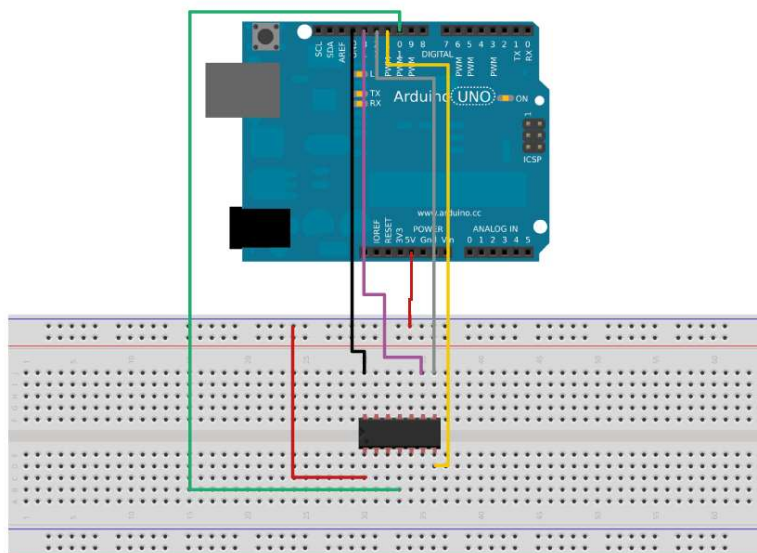


Figura 2.1

- In Figura 2.2 è rappresentato lo schema per il collegamento della pompa. E' sorta la necessità di collegare un amplificatore operazionale, in configurazione *inseguitore di tensione*, per aumentare la corrente in ingresso alla pompa così da fornirle maggiore potenza.

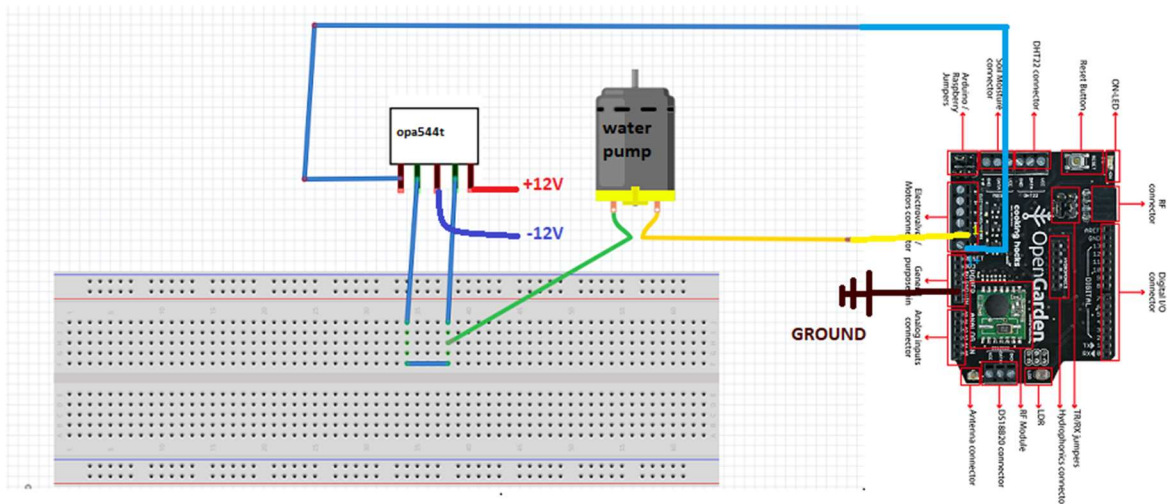


Figura 2.2

3. Elenco dei componenti

- 1) Microcontrollore ATtiny84
- 2) Arduino Genuino UNO (Figura 3.1)

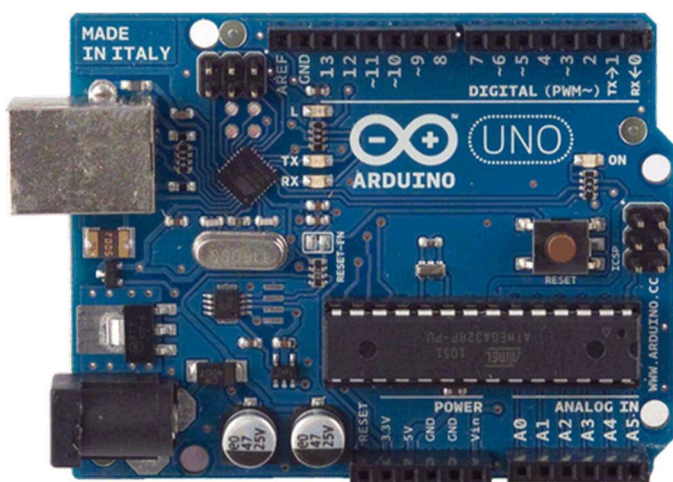


Figura 3.1

- 3) Open Garden Shield per Arduino (Figura 3.2) :
La piattaforma permette di controllare lo stato della pianta dal sensing dei seguenti parametri:

- Umidità del terreno
- Temperatura dell'aria
- Umidità dell'aria
- Luminosità ambientale
- Lettura del Ph
- Conducibilità

La piattaforma permette inoltre di attivare diversi tipi di attuatori per modificare lo stato della pianta:

- Pompa ad acqua
- Sgocciolatore
- Elettrovalvola
- Pompa per l'ossigeno

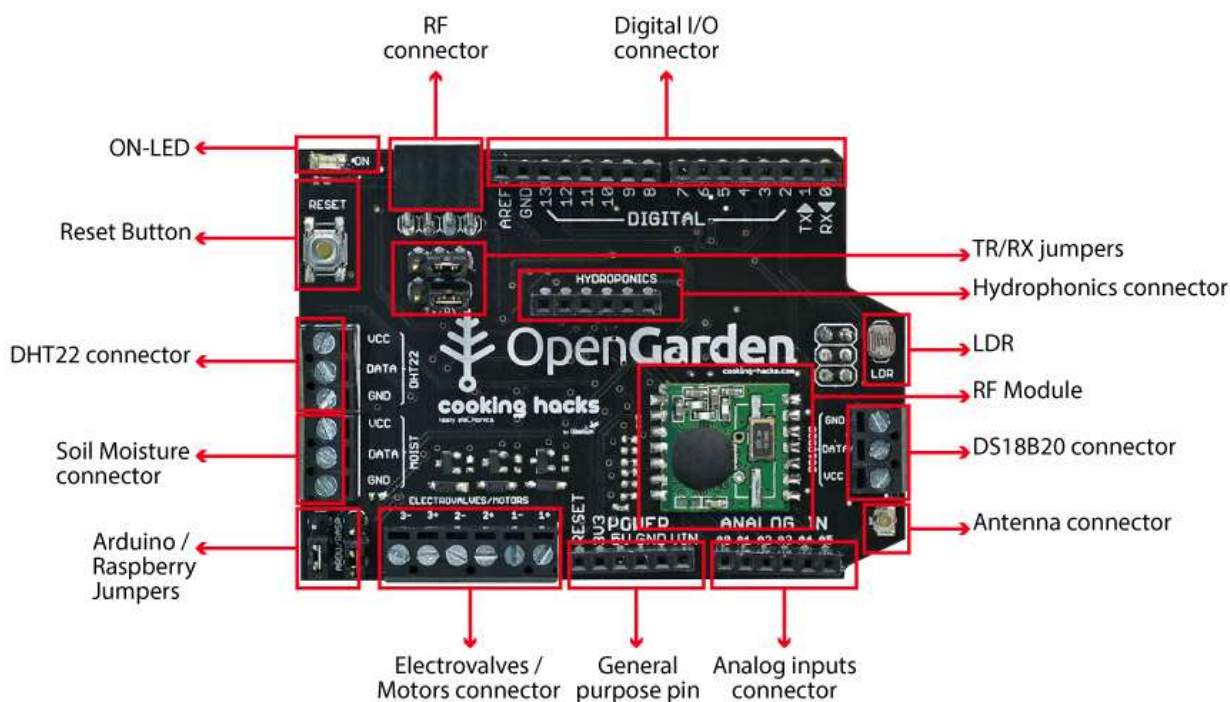


Figura 3.2

4) Open Garden Node (Figura 3.3) :

La piattaforma permette di controllare lo stato della pianta dal sensing dei seguenti parametri:

- Umidità del terreno
- Temperatura dell'aria
- Umidità dell'aria
- Luminosità ambientale
- Lettura del Ph
- Conducibilità

La piattaforma permette inoltre di attivare diversi tipi di attuatori per modificare lo stato della pianta:

- Pompa ad acqua
- Gocciolatore
- Elettrovalvola
- Pompa per l'ossigeno

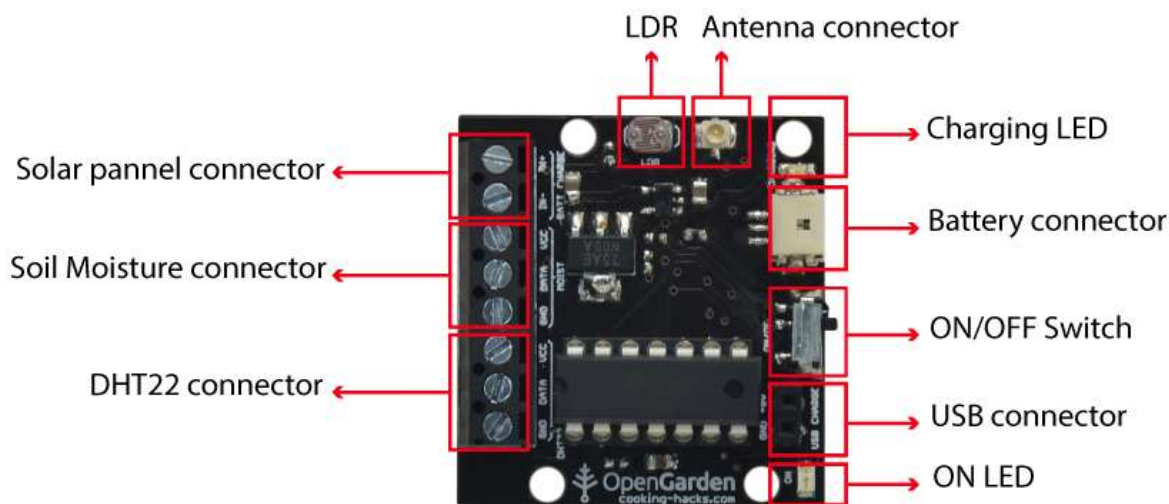


Figura 3.3

5) Amplificatore Operazionale OPA544T(Figura 3.4)

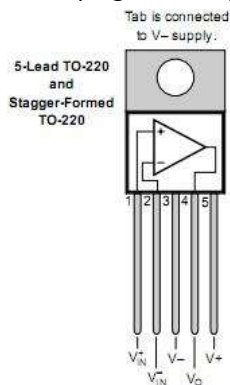


Figura 3.4

- 6) Batteria ricaricabile da 2300 mA•h
- 7) Pannello solare 70x50mm
- 8) SparkFun microB USB
- 9) Breadboard
- 10) Box per i nodi x2



4. Strumenti utilizzati

1) Alimentatore stabilizzato

- Marca: Tektronix
- Serie: PS280 & PS283
- Caratteristiche di targa:
 - I. Caratteristiche fisiche:

Dimensione	Misura
Larghezza	255mm
Altezza	145mm
Profondità	335mm
Peso	11.5 Kg per il modello PS280 9.0 Kg per il modello PS283

II. Caratteristiche ambientali:

Caratteristica	Temperatura	Umidità Relativa
Conservazione	-10°C fino a +70°C	70%
Operativo	0°C fino a 40°C	80%

III. Caratteristiche operazionali:

Caratteristica	Misura
Uscita	Due da 0 a 30 VDC, una d 5 VDC
Voltaggio (5V)	5.0±0.25 VDC a 3.0 A
Voltaggio (0-30 V)	0-30 VDC costante a 2.0 A costante (Max per PS280) o a 1.0 A costante (Max per PS283)
Regolazione lineare (5 V)	≤5 mV
Regolazione lineare (CV)	≤0.01% +3 mV PS280 ≤0.01% +5mV PS283
Regolazione lineare (CC)	≤0.2% +3 mA
Regolazione del carico (5 V)	≤0.2%
Ondulazione/Rumore (5 V)	≤2 mV rms
Ondulazione/Rumore (CV)	≤1 mV rms, 5 Hz–1 Mhz
Coefficiente di Temperatura (CV)	≤300ppm/°C



2) Sonda DHT22

Caratteristiche:

- 3.3–6V Input
- 1–1.5mA corrente misurata
- 40–50 uA corrente in standby
- Umidità da 0–100% RH
- -40 – 80 °C temperatura
- +-2% RH precisione
- +-0.5 gradi C

3) Sonda per l'umidità del terreno

Caratteristiche:

- 3.3–5V Input
- 0–35mA corrente misurata
- Output analogico
- Umidità da 0–1000

5. Descrizione delle misure effettuate

Le misure vengono automaticamente effettuate dai sensori ed inviate tramite un'antenna dal nodo al gateway, il quale, collegato ad un computer, stampa i valori raccolti sul terminale dell'IDE di Arduino.

Non avendo a disposizione un vero sistema fisico di applicazione, il sistema di controllo così come la variazione dei parametri fisici sono stati testati ricorrendo ai metodi successivamente descritti.

Per il sensore di luce presente sullo shield è stata utilizzata una torcia, in modo da aumentare il valore di luminosità letto.

Per il sensore di umidità e temperatura dell'aria, è stato necessario soffiare all'interno della sonda, così da aumentare l'umidità dell'aria e, anche se in modo lieve, la temperatura.

Infine, il sensore di umidità del terreno è stato inserito all'interno di una vaschetta d'acqua, così da aumentare l'umidità letta.

6. Risultati

Dalle prove effettuate si è rilevato un ritardo di risposta, da parte dei sensori, di un minimo di 55 secondi ad un massimo di 60 secondi misurati grazie ad un contatore inserito nel programma.

In figura 6.1 è mostrato il diagramma di flusso rappresentante la logica di programmazione

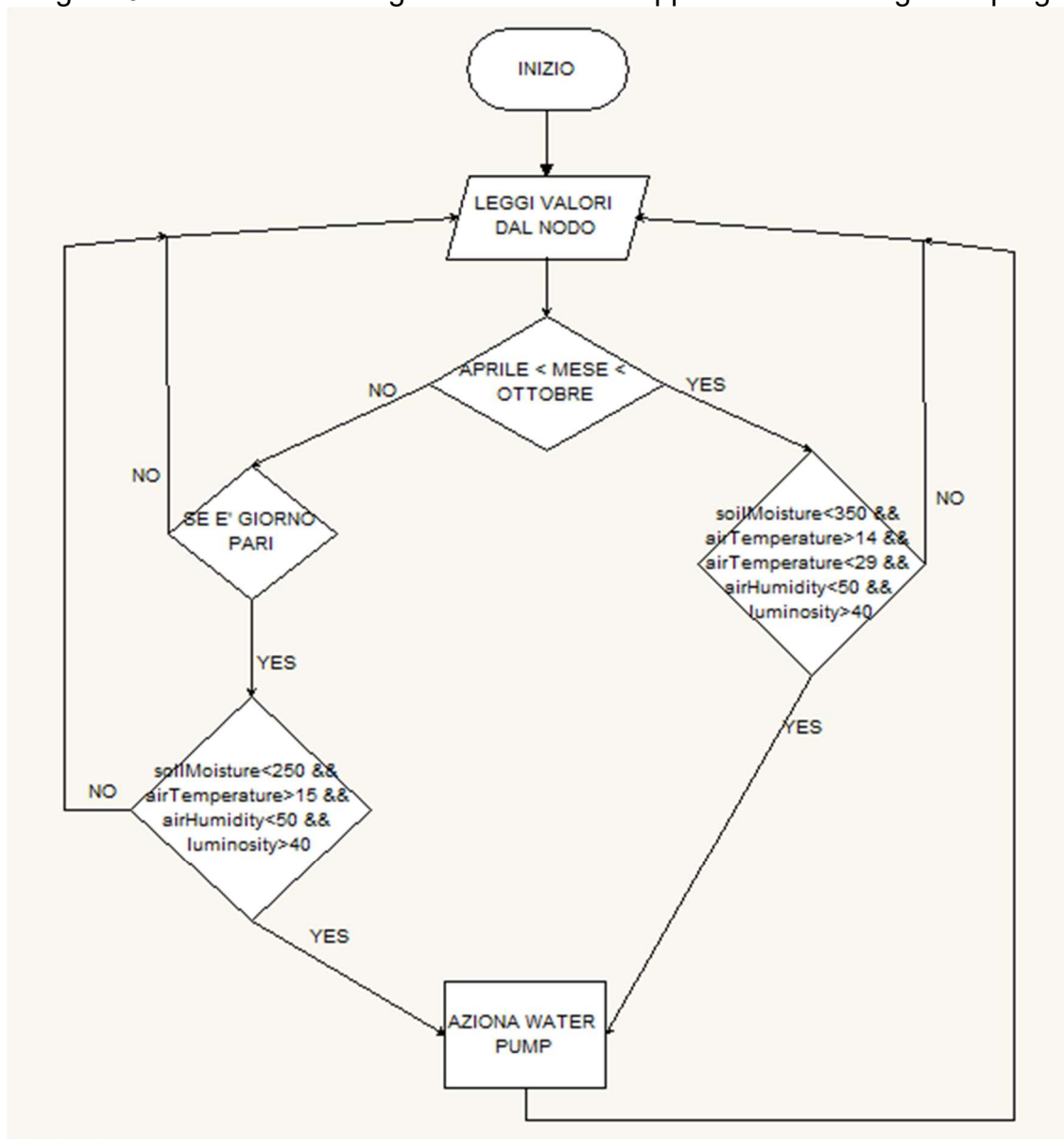


Figura 6.1



- **Funzionamento normale.**

```
06/11/2018 MON 10:23:31*****
Node 1:
Temperature: 22°C
Humidity: 34%RH
Soil Moisture: 0
Luminosity: 37%
Battery Voltage: 3254mV
*****
Node 2:
No node connected
*****
Node 3:
No node connected
*****
```

Il primo risultato è stato ottenuto valutando le condizioni ambientali correnti. Poiché i parametri non soddisfano i requisiti minimi per l'accensione della pompa (la luminosità non è superiore al 40%), questa non viene attiva, come è possibile osservare dai risultati.

- **Forzamento della luminosità.**

```
06/11/2018 MON 10:24:51*****
Node 1:
Temperature: 22°C
Humidity: 34%RH
Soil Moisture: 0
Luminosity: 87%
Battery Voltage: 3254mV
*****
Node 2:
No node connected
*****
Node 3:
No node connected
*****
Irrigation 1: ON
Irrigation 1: OFF
```

Dopo aver forzato la luminosità del sensore, questa soddisfa la condizione necessaria. I risultati, infatti, dimostrano che la pompa è stata attivata.



- **Risultati successivi.**

```
06/11/2018 MON 10:27:54*****
Node 1:
Temperature: 22°C
Humidity: 34%RH
Soil Moisture: 973
Luminosity: 37%
Battery Voltage: 3254mV
*****
Node 2:
No node connected
*****
Node 3:
No node connected
*****
```

Dopo aver attivato la pompa, è stato posto il sensore di umidità del terreno nella vaschetta d'acqua per aumentare il valore letto, simulando dunque un'ipotetica irrigazione. Osserviamo che poiché il terreno risulta umido, non è soddisfatta la condizione d'irrigazione, dunque la pompa non viene avviata.

- **Dicembre dispari.**

```
12/11/2018 TUE 10:33:27*****
Node 1:
Temperature: 22°C
Humidity: 34%RH
Soil Moisture: 0
Luminosity: 81%
Battery Voltage: 3254mV
*****
Node 2:
No node connected
*****
Node 3:
No node connected
*****
```

Cambiando la data all'undici dicembre osserviamo che nonostante siano soddisfatte le condizioni sui parametri fisici, la pompa non è in funzione poiché il giorno non è pari.



- **Dicembre pari.**

```
12/12/2018 WED 10:38:29*****
Node 1:
Temperature: 22°C
Humidity: 34%RH
Soil Moisture: 0
Luminosity: 83%
Battery Voltage: 3254mV
*****
Node 2:
No node connected
*****
Node 3:
No node connected
*****
Irrigation 1: ON
Irrigation 1: OFF
```

Infine, cambiando la data in un giorno pari di dicembre, i risultati verificano il corretto funzionamento del programma.

7. Conclusioni

Il sistema è stato progettato in modo generico per poter gestire la maggiore varietà di piante da casa, tuttavia si ribadisce che non è stato effettuato alcun test in una vera applicazione, dunque potrebbe essere necessario un riadattamento della logica di controllo, effettuabile dal semplice cambiamento dei valori di soglia per l'accensione della pompa all'interno del codice (allegato in seguito).

Il sistema può essere ottimizzato impiegando un sensore di livello per il serbatoio dell'acqua ed un buzzer, in modo da inviare un allarme e spegnere la pompa, nel momento in cui l'acqua scenda al di sotto di un livello minimo.

E' possibile inoltre aggiungere un sensore di durezza dell'acqua, in quanto il calcare è dannoso tanto per le piante quanto per la pompa, rischiando di ostruirla.

Infine, si rammenta che la presenza della porta microUSB permette di alimentare esternamente il sistema, così da non dipendere dalla batteria.



Codice.

- Inizializzazione.

```
#include <OpenGarden.h>
#include <Wire.h>

DateTime myTime;
int flag = 0; // auxiliar variable

void setup() {
  Serial.begin(9600);
  OpenGarden.initSensors(); //Initialize sensors power
  OpenGarden.sensorPowerON(); //Turn On the sensors
  delay(500);
  OpenGarden.initRF();
  OpenGarden.initRTC();
  OpenGarden.initIrrigation(1); //Initialize irrigation 1
  OpenGarden.initIrrigation(2); //Initialize irrigation 2
  OpenGarden.initIrrigation(3); //Initialize irrigation 3
}
```

Vengono inizializzati ed avviati i sensori, l'antenna ed un orologio per avere informazioni sulla data.

- Logica di controllo.

```
void loop() {
  myTime = OpenGarden.getTime();
  OpenGarden.receiveFromNode();

  int luminosity0 = OpenGarden.readLuminosity();

  //Get Node Sensors
  Payload node1Packet = OpenGarden.getNodeData(node1);
  Payload node2Packet = OpenGarden.getNodeData(node2);
  Payload node3Packet = OpenGarden.getNodeData(node3);
```

Viene effettuata la lettura della luminosità dallo shield di Arduino e ricevuta una struttura contenente i dati raccolti dai nodi (OpenGarden.getNodeData).



```
int soilMoisture1 = node1Packet.moisture;
float airTemperature1 = node1Packet.temperature;
float airHumidity1 = node1Packet.humidity;
int luminosity1 = node1Packet.light;
int battery1 = node1Packet.supplyV;

int soilMoisture2 = node2Packet.moisture;
float airTemperature2 = node2Packet.temperature;
float airHumidity2 = node2Packet.humidity;
int luminosity2 = node2Packet.light;
int battery2 = node2Packet.supplyV;

int soilMoisture3 = node3Packet.moisture;
float airTemperature3 = node3Packet.temperature;
float airHumidity3 = node3Packet.humidity;
int luminosity3 = node3Packet.light;
int battery3 = node3Packet.supplyV;

OpenGarden.printTime(myTime);
Serial.println("*****");
Serial.println("Node 1:");
OpenGarden.printNode(node1Packet);
Serial.println("*****");
Serial.println("Node 2:");
OpenGarden.printNode(node2Packet);
Serial.println("*****");
Serial.println("Node 3:");
OpenGarden.printNode(node3Packet);
Serial.println("*****");
```

Vengono stampati e salvati, all'interno delle variabili, i corrispettivi dati incapsulati nella struttura.



```
//NODO 1
//MAG-GIU-LUG-AGO-SET
if(myTime.month>4 && myTime.month<10){
  if(soilMoisture1<350 && airTemperature1>14 && airTemperature1<29 && airHumidity1<50
  && luminosity0>40) {
    OpenGarden.irrigationON(1); //Turn ON the irrigation number 1 (insert 1,2 or 3)
    Serial.println("Irrigation 1: ON");
    delay(5000)
    OpenGarden.irrigationOFF(1); //Turn OFF the irrigation number 1 (insert 1,2 or 3)
    Serial.println("Irrigation 1: OFF");
  }
}
//GEN-FEB-MAR-APR-OTT-NOV-DIC
if(myTime.month>9 && myTime.month<5){
  if(myTime.day%2==0){

    if(soilMoisture1<250 && airTemperature1>15 && airHumidity1<50 && luminosity0>40) {
      OpenGarden.irrigationON(1); //Turn ON the irrigation number 1 (insert 1,2 or 3)
      Serial.println("Irrigation 1: ON");
      delay(5000)
      OpenGarden.irrigationOFF(1); //Turn OFF the irrigation number 1 (insert 1,2 or 3)
      Serial.println("Irrigation 1: OFF");
    }
  }
}
```

Viene effettuato il controllo delle condizioni necessarie per l'accensione della pompa.

In particolare, la pompa viene accesa per 5 secondi e poi spenta se sussistono le seguenti condizioni:

1. Il mese corrente è compreso tra maggio e settembre.
2. L'umidità del terreno è inferiore a 350.
3. La temperatura dell'aria è compresa tra 14 e 29 °C.
4. L'umidità dell'aria è minore di 50.
5. La luminosità è superiore del 40%.

In alternativa:

1. Il mese corrente è compreso tra ottobre ed aprile.
2. Il giorno corrente è pari.
3. L'umidità del terreno è inferiore a 250.
4. La temperatura dell'aria è superiore a 15 °C.
5. L'umidità dell'aria è minore di 50.
6. La luminosità è maggiore del 40%.



```
//NODO 2
if(myTime.month>4 && myTime.month<10){
  if(soilMoisture2<350 && airTemperature2>14 && airTemperature2<29 && airHumidity2<50 && luminosity0>40)
  {
    OpenGarden.irrigationON(2); //Turn ON the irrigation number 1 (insert 1,2 or 3)
    Serial.println("Irrigation 1: ON");
    delay(5000)
    OpenGarden.irrigationOFF(2); //Turn OFF the irrigation number 1 (insert 1,2 or 3)
    Serial.println("Irrigation 1: OFF");
  }
}

if(myTime.month>9 && myTime.month<5){
  if(myTime.day%2==0){

    if(soilMoisture2<250 && airTemperature2>15 && airHumidity2<50 && luminosity0>40) {
      OpenGarden.irrigationON(2); //Turn ON the irrigation number 1 (insert 1,2 or 3)
      Serial.println("Irrigation 1: ON");
      delay(5000)
      OpenGarden.irrigationOFF(2); //Turn OFF the irrigation number 1 (insert 1,2 or 3)
      Serial.println("Irrigation 1: OFF");
    }
  }
}

//NODO 3
if(myTime.month>4 && myTime.month<10){
  if(soilMoisture3<350 && airTemperature3>14 && airTemperature3<29 && airHumidity3<50 && luminosity0>40)
  {
    OpenGarden.irrigationON(3); //Turn ON the irrigation number 1 (insert 1,2 or 3)
    Serial.println("Irrigation 1: ON");
    delay(5000)
    OpenGarden.irrigationOFF(3); //Turn OFF the irrigation number 1 (insert 1,2 or 3)
    Serial.println("Irrigation 1: OFF");
  }
}

if(myTime.month>9 && myTime.month<5){
  if(myTime.day%2==0){

    if(soilMoisture3<250 && airTemperature3>15 && airHumidity3<50 && luminosity0>40) {
      OpenGarden.irrigationON(3); //Turn ON the irrigation number 1 (insert 1,2 or 3)
      Serial.println("Irrigation 1: ON");
      delay(5000)
      OpenGarden.irrigationOFF(3); //Turn OFF the irrigation number 1 (insert 1,2 or 3)
      Serial.println("Irrigation 1: OFF");
    }
  }
}

delay(60000);
}
```

Vengono effettuati gli stessi controlli precedentemente descritti per il secondo e terzo Nodo del sistema. Infine, si attende un minuto così da fornire ai sensori il tempo necessario per lettura delle variabili e affinché la terra riesca ad assorbire l'acqua.