

Rilevamento della quantità di acqua rimanente in un sistema di irrigazione improvvisato

Descrizione del progetto

Questo progetto riguarda il rilevamento della quantità di acqua rimanente in un sistema di irrigazione improvvisato che può essere attaccato a vasi da fiori e/o piante.

Durante l'estate, molti proprietari di piante e fiori in Grecia stanno cercando di continuare ad innaffiarli utilizzando metodi e soluzioni improvvisate e piuttosto ecologiche. Uno di questi è l'uso di comuni bottiglie di plastica, contenenti acqua, che si attaccano in diversi modi alle piante e/o ai vasi da fiori (Figura 1). Tuttavia, i proprietari devono controllare (anche quotidianamente) la quantità di acqua rimanente all'interno delle bottiglie e, se necessario, sostituirla.



Figura 1 Soluzioni improvvisate per l'irrigazione degli impianti

Pertanto, l'idea principale dell'attività attuale è la creazione di un dispositivo basato su Arduino, dotato di un sensore ad ultrasuoni o di un rilevatore metallico, che informi i proprietari sulla quantità di acqua rimanente all'interno di ogni bottiglia. Quando la bottiglia sta per svuotarsi, il sensore ad ultrasuoni viene attivato (Figura 2) emettendo un segnale ottico o audio al sistema basato su Arduino, che informerà istantaneamente il proprietario attraverso un allarme GSM Caller Alarm (Figura 4) o attraverso un'applicazione dati Bluetooth Arduino. In alternativa, può essere applicato un sensore metallico che rileva il livello dell'acqua (Figura 3), emettendo un segnale quando il livello è basso.

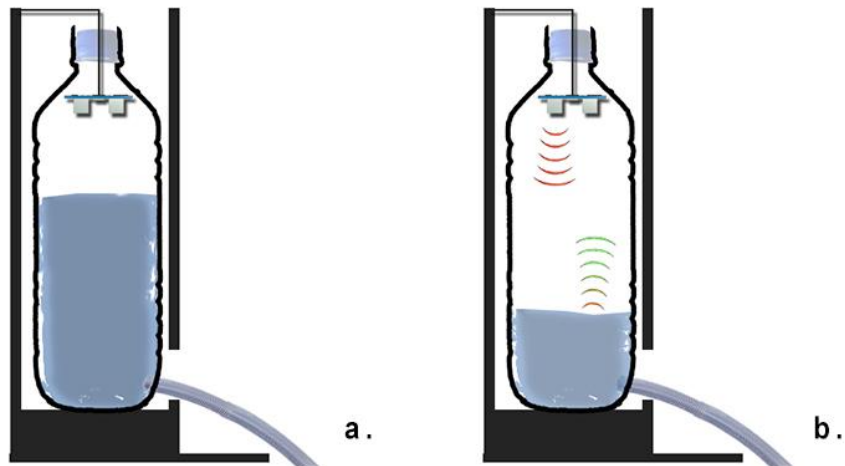


Figura 2 Esempio schematico del meccanismo con un sensore ad ultrasuoni

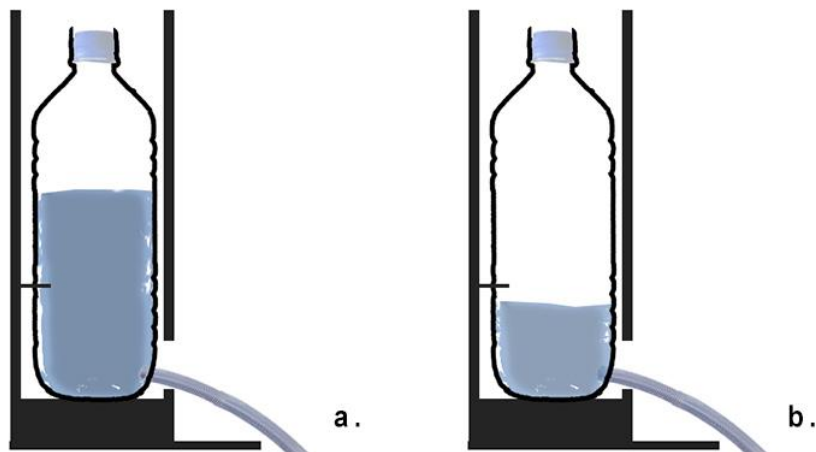
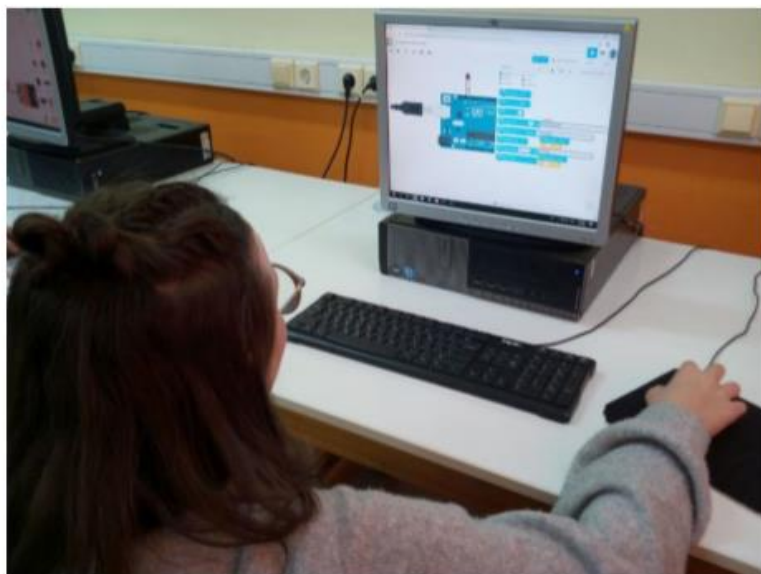


Figura 3 Esempio schematico del meccanismo con un rilevatore metallico



Figura 4: Informazioni per questo modulo si trovano qui: <http://tiny.cc/zp871y>

Entrambi i componenti (Arduino e GSM) potrebbero essere alimentati da banchi solari e/o batterie ricaricabili.



```

-- Select Board or Port --
SHARE

half_baked.ino  README.adoc
1  #include<SoftwareSerial.h>
2
3  int sensorPin = A;
4
5  int sensorValue = 0; // variable to store the value coming from the sensor
6
7  int led = ; // Output pin for LED ??
8
9  const int buzzer = ; // Output pin for Buzzer ??
10
11 void setup() {
12     // declare the ledPin and buzzer as an OUTPUT:
13
14     pinMode(led, OUTPUT);
15
16     pinMode(buzzer, OUTPUT);
17
18     Serial.begin(9600);
19
20 }
21
22 void loop()
23
24
CREATE

```

```

-- Select Board or Port --
SHARE

water_detector.ino  README.adoc
12 // initialize the LED pin as an output:
13 pinMode(ledPin, OUTPUT);
14 // initialize the pushbutton pin as an input:
15 pinMode(buttonPin, INPUT);
16 pinMode(buzzer, OUTPUT); //The Speaker
17 Serial.begin(9600);
18 }
19
20 void loop() {
21     // read the state of the pushbutton value:
22     buttonState = digitalRead(buttonPin);
23
24     // check if the pushbutton is pressed. If it is, the buttonState is HIGH:
25     if (buttonState == HIGH) {
26         // turn LED on:
27         digitalWrite(ledPin, HIGH);
28         // turn buzzer on:
29         tone(buzzer, 1000); // Send 1KHz sound signal...
30         delay(1000); // ...for 1 sec
31         noTone(buzzer); // Stop sound...
32         delay(1000);
33     } else {
34         // turn LED & buzzer off:
35         digitalWrite(ledPin, LOW);
36         digitalWrite(buzzer, LOW);
37     }
38 }
CREATE

```