***Sistema automatico per irrigazione a pioggia***

**Preventivo**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Oggetto*** | ***Marca*** | ***Modello*** | ***Prezzo [€]*** | ***Quantità*** |
| Batteria al piombo AGM | ELECTRON |  | 5.60 cad. | 1 |
| Igrometro[[1]](#footnote-1) | SODIAL(R) | 014844 | 1.79 cad. | 1 |
| Pompa ad immersione | SODIAL(R) | SPAGMA 47566 | 5.68 cad. | 1 |
| Serbatoio | Stefanplast | 22070 | 4.54 | 1 |
| Sottovaso mediterraneo | Stefanplast | 83300  | 1.43 cad. | 1 |
| Starter kit[[2]](#footnote-2) | Arduino | Uno | 88.10 cad. | 1 |
| Tubo flessibile | Mebra Plastik | PU 98 MB-LONGLIFE | 0.65 al metro | 1 metro |
| Vaso mediterraneo | Stefanplast | 82160 | 1.89 cad. | 1 |

Spesa complessiva: 109.08 € (senza pianta e terriccio)

**Specifiche tecniche**

*Batteria al piombo AGM*

* Corrente erogata: 2.2 Ah
* Tensione erogata: 12 V

*Igrometro*

* Sensibilità: regolabile tramite potenziometro manuale
* Tensione di alimentazione: 3.3 V / 5 V
* Tipo di misura: resistività

*Pompa ad immersione*

* Corrente di avviamento: 70 mA
* Corrente di spegnimento: 50 mA
* Corrente nominale: 400 mA
* Impermeabilità: IP68
* Portata: 240 L/h
* Potenza assorbita: 5 W
* Resistenza a vuoto: 210 kΩ
* Tensione di avviamento: 6.8 V
* Tensione di spegnimento: 4 V
* Tensione nominale: 12 V

*Starter kit: diodo 1N4007*

* Capacità di giunzione: 8 pF
* Tensione di break-down: 1 kV

*Starter kit: mosfet IRF5200*

* Corrente continua di Drain: 6.5 A / 9.2 A
* Massima tensione Drain-Source: 100 V
* Massima tensione Gate-Source: ± 20 V

**Schema circuitale**



Fig.A

**Funzionamento**

Il progetto propone l’automazione di un sistema di irrigazione.
Per fare ciò abbiamo utilizzato gli strumenti sopra-elencati.
Facendo riferimento al circuito in Fig.A, analizziamo il funzionamento del progetto: le parti principali sono costituite da un sensore, una pompa e un transistor.

Attraverso l’uso del sensore (opportunamente alimentato con Arduino) riusciamo a ricavare, in ingresso per Arduino, un valore analogico che va da un minimo di 0 ad un massimo di 210-1. Questo valore rappresenta, indirettamente, il valore di umidità del terreno. Infatti il sensore misura valori di resistività: tanto più il terreno è secco tanto più sarà resistivo e quindi il sensore restituirà un valore prossimo a 1023, tanto più il terreno sarà umido tanto meno sarà resistivo, il sensore restituirà quindi, in questo caso, un valore prossimo allo 0.
Il sensore di umidità, collegato ad Arduino, è composto da due estremità separate e rivestite da una lega metallica conduttrice; all'interno di quest'ultime viene fatta circolare una corrente elettrica e i valori che il sensore restituisce sono strettamente collegati alla resistenza posta tra le due estremità, una volta chiuso il circuito, come spiegato sopra.

D’altra parte abbiamo l’output di Arduino: una pompa (necessaria per la “prelevazione” di acqua da un serbatoio per poi essere convogliata in un tubo per l’irrigazione) che è alimentata da un generatore. Ora, non volendo che la pompa funzioni costantemente, onde evitare di annegare la pianta, dobbiamo creare, con un qualunque artificio, un’interruttore posto in serie alla pompa. Abbiamo infatti inserito un MOSFET utilizzandolo, appunto, come interruttore. Solo quando il MOSFET è aperto il generatore riesce ad alimentare la pompa, che quindi si aziona, altrimenti la pompa rimane spenta.

Il pilotaggio del MOSFET è assegnato interamente ad Arduino: un pin digitale è collegato al Gate del MOSFET: quando questo pin applica un potenziale pari a 5V permette al MOSFET di aprirsi e, a sua volta, permette al generatore di alimentare correttamente la pompa che quindi inizia a prelevare acqua dal serbatoio.
Il pin digitale è controllato dal programma caricato su Arduino: dopo diverse prove abbiamo trovato due valori di umidità che, una volta raggiunti e registrati, permettono al pin digitale di attivarsi ed erogare 5V.

NOTE:

1) I valori analogici in ingresso (range 0-1023) sono stati mappati e limitati tra 0-1000

2) Il “delay” (tempistiche di accensione e spegnimento) sono controllate da Arduino:

* la pompa sta accesa fino a quando il terreno non è “abbastanza umido”, in caso di accensione della pompa Arduino registra 1 valore ogni secondo.
* la pompa rimane spenta fino a quando il terreno non è “abbastanza secco”, in questo caso Arduino registra 10 valori ogni secondo.

3) Sul monitor seriale di Arduino sono riportati i valori di umidità in percentuale e lo stato del terreno.

**Vantaggi e Svantaggi**

*Vantaggi*

* Economico
* Permette di automatizzare il processo di irrigazione
* Semplice da realizzare
* Versatile per ogni tipo di terreno: parametri che determinano irrigazione regolabili

*Svantaggi*

* Alta probabilità che il sensore si ossidi: in continuo contatto con l'acqua
* Irrigazione a pioggia non adatto per piante a foglie larghe: l'acqua non arriverebbe al terreno
* Sensore fornisce un valore di umidità per il terreno nelle sue vicinanze
1. Include: 1 modulo rivelatore di umidità del suolo, 1 sonda e 5 cavi [↑](#footnote-ref-1)
2. Include: 1 Arduino Projects Book, 1 Scheda Arduino UNO rev.3, 1 cavo USB, 1 breadboard, 1 basetta di supporto in legno, 1 adattatore per batteria 9 V, 70 cavi plastificati rigidi, 2 cavi plastificati flessibili, 6 fotoresistori, 3 potenziometri da 10 kΩ, 10 tasti a pressione, 1 sensore di temperatura, 1 sensore tilt, 1 LCD alfanumerico, 29 LED di diversi colori, 1 Motore CC 6/9V, 1 servo motore, 1 piezo, 1 ponte H, 2 fotoaccoppiatori, 5 transistor, 2 transistor mosfet [IRF520], 15 condensatori di diversa capacità, 5 diodi [1N4007], 3 gelatine trasparenti didifferenti colori, 1 strip di connettori maschio e 55 resistenze di varie impedenze. [↑](#footnote-ref-2)