Progetto extra-scolastico

La “nostra” casa domotica

ISTITUTO TECNICO INDUSTRIALE E. MEDI

V ELETTROTECNICA SERALE

ANNO SCOLASTICO: 2015/2016

Carrano Emanuele

Di Giacomo Antonio

Sannino Christian

Sannino Simone

Introduzione

*“Domotica”, dall'unione delle parole “**domus”** (che in latino significa casa) e “**robotica”**, è un neologismo che sta ad indicare la scienza che si occupa dello studio delle tecnologie atte a migliorare la qualità della vita nella casa e più in generale negli ambienti alterati in funzione delle esigenze dell'uomo.*

La Domotica svolge un ruolo importante nel rendere intelligenti apparecchiature, impianti e sistemi. Ad esempio un impianto elettrico intelligente può autoregolare l'accensione degli elettrodomestici per non superare la soglia che farebbe scattare il contatore.

Inoltre con "*casa intelligente*" si indica un ambiente domestico, il quale mette a disposizione dell'utente impianti che vanno oltre il "*tradizionale*", dove apparecchiature e sistemi sono in grado di svolgere funzioni parzialmente o completamente autonome, in modo da permettere la gestione coordinata, integrata e computerizzata degli impianti tecnologici, delle reti informatiche e delle reti di comunicazione, allo scopo di migliorare la flessibilità di gestione e il comfort che l'ambiente domestico può offrire.

Una casa domotica può essere controllata dall'utilizzatore tramite opportune **interfacce utenti** come touch-screen, tastiere, riconoscimento vocale, che realizzano la comunicazione con il sistema intelligente di controllo, basato su un'**unità computerizzata centrale** oppure su un **sistema a intelligenza distribuita**.

Un sistema domotico si completa, di solito, attraverso uno o più **sistemi di comunicazione con il mondo esterno** ad esempio rete locale, bluetooth o wifi per permetterne il controllo e la visualizzazione dello stato della casa sia all'interno che all'esterno della stessa per mezzo di monitor di controllo, luci di segnalazione o meglio, interfacce web.

Il plastico

La “nostra” casa domotica è un semplice esempio di casa intelligente, essa ha dimensioni 120x95. E rappresenta una villetta con tanto di garage e giardino. Il tutto è stato realizzato con materiali di fortuna riciclati.

L'intera struttura è composta da: una stazione di controllo centrale a sinistra, al centro la casa vera e

propria con 3 stanze e a destra il garage per l'auto con al di sopra del quale è montato l'inseguitore solare per il pannello fotovoltaico.

Nel giardino è presente una maestosa fontana e un sensore di vento sul lato sinistro, rispetto all'ingresso principale, mentre a destra abbiamo il cancello automatico che porta dritto al garage.

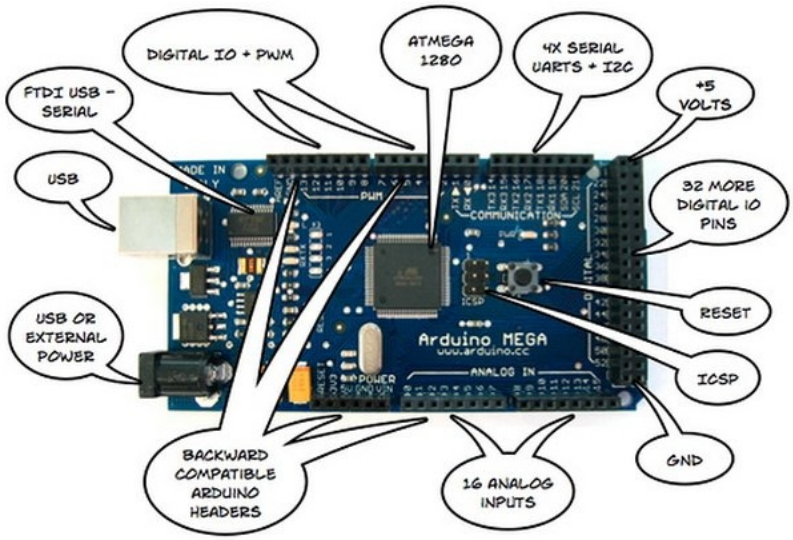
Alimentazione

Il primo problema da risolvere era quello di alimentare tutti gli apparati presenti nella casa, questo perchè ogni componente lavora a tensioni differenti. Quindi dovevamo procurarci vari trasformatori per le varie tensioni di cui avevamo bisogno, tenendo conto anche degli assorbimenti dei vari dispositivi.

Noi abbiamo scelto una via più comoda, ma allo stesso tempo più efficiente, infatti abbiamo utilizzato un **alimentatore** per PC fisso (ATX) dal quale possiamo prelevare differenti tensioni: 5V, 3.3V, 12V, -5V, -12V. Inoltre l'alimentatore supporta fino a 500W di potenza quindi anche il problema dei carichi è risolto.

Sistema di controllo

Il sistema di controllo che abbiamo utilizzato è l'oramai famosa piattaforma **Arduino**. Abbiamo scelto la versione **mega2560** perchè questo progetto necessita di molte interconnessioni.



Arduino è una scheda di controllo per hobbisti che sta avendo successo anche nel mondo professionale. Esso è equipaggiato con un microcontrollore **Atmel** che varia da versione a versione, nel nostro caso abbiamo un Atmel256 che presenta 70 pin digitali di I/0, di cui 16 pin possono essere usati per l' I/0 analogici e 6 pin dedicati agli interrupt. Tuttavia per il progetto necessitavamo di altre interconnessioni e quindi Arduino mega è affiancata da un **Arduino uno** con altri 14 pin digitali e 6 pin analogici disponibili.

Arduino per svolgere le sue funzioni deve essere programmato con il **linguaggio C++**; il

programma viene scritto usando l'IDE messo a disposizione dagli stessi sviluppatori di Arduino e una volta compilato senza errori, il controllore viene flashato tramite la porta USB.

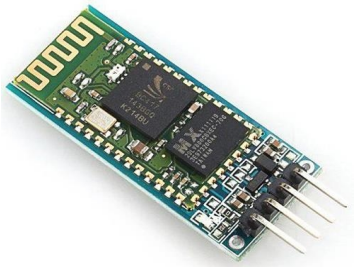
Cosa fa il nostro programma?

Abbiamo programmato i due Arduino in maniera tale che essi “*leggano*” i segnali esterni che provengono dai vari *sensori* con cui abbiamo equipaggiato la nostra casa, oltre ai segnali provenienti da uno smartphone tramite *bluetooth*. Una volta acquisiti questi segnali il programma effettua varie *operazioni* per inviare segnali opportuni ai vari *attuatori*.

Sistema di comunicazione

Come è stato accennato prima, l'interfaccia di comunicazione è stata realizzata tramite **bluetooth**, utilizzando un modulo compatibile con Arduino: l'HC-06.

Questo modulo presenta 4 pin: 2 per l'alimentazione (3.3V) e 2 per la **comunicazione seriale** con Arduino (TX, RX). Ovviamente per una corretta comunicazione i pin RX e TX devono essere collegati ad arduino in modo incrociato cioè: TXb-RXa e Rxb-Txa.



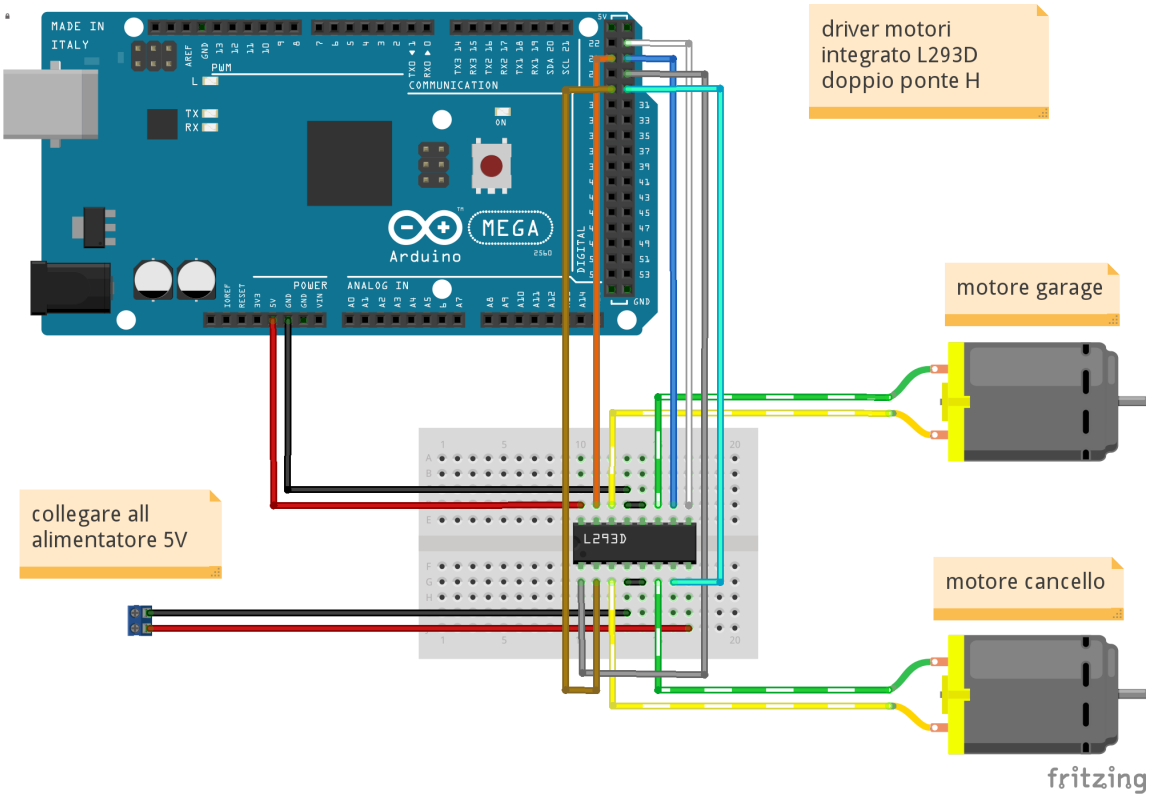
A questo punto necessitavamo di un'applicazione per smartphone e per questo abbiamo chiesto aiuto al nostro amico *Antonio Mentone*, il quale ha realizzato l'**app Android** che funge da telecomando. L'app presenta una semplice interfaccia utente con dei pulsanti ognuno dei quali consente l'invio di un particolare segnale (carattere), Arduino riceve questi segnali e in base al loro valore, svolge determinati compiti.

I motori DC

Da un punto di vista informatico è stato necessario implementare un **driver**, *sia hardware che software (libreria)*, per la gestione dei motori, che consente di accendere, fermare e cambiare direzione ai motori. In questo modo Arduino non fa altro che inviare i giusti segnali al driver hardware su cui è montato il motore.

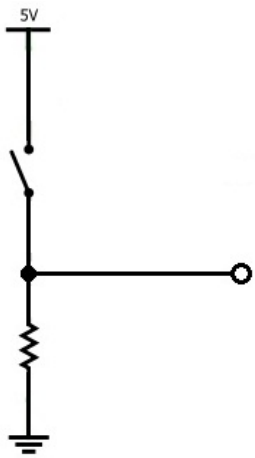
Nella casa abbiamo usato vari motori DC per: il garage, il cancello, la tenda, il sensore vento, l'inseguitore solare.

Per il garage, il cancello, la tenda e l'inseguitore solare si ha necessità di invertire il senso di marcia del motore e quindi abbiamo utilizzato 2 **integrati L293D** (Driver Motor) con 16 pin, ognuno composto da 2 **ponti H** che consentono di invertire il senso della corrente.



In questo modo i motori vanno montati sull'integrato che viene pilotato da Arduino tramite 3 segnali per motore: 1 per l'**abilitazione** (avviamento o arresto del motore) e la regolazione della velocità (tramite tecnica PWM) e gli altri 2 per il **senso di marcia**.

I restanti pin servono per l'alimentazione. I suddetti integrati presentano un'**alimentazione duale**: una (5V) per alimentare la circuiteria interna dell'integrato, l'altra è la tensione di riferimento dei motori (6V).



Per fermare i motori del cancello, del garage e della tenda abbiamo utilizzato dei **fine corsa** auto-costruiti.

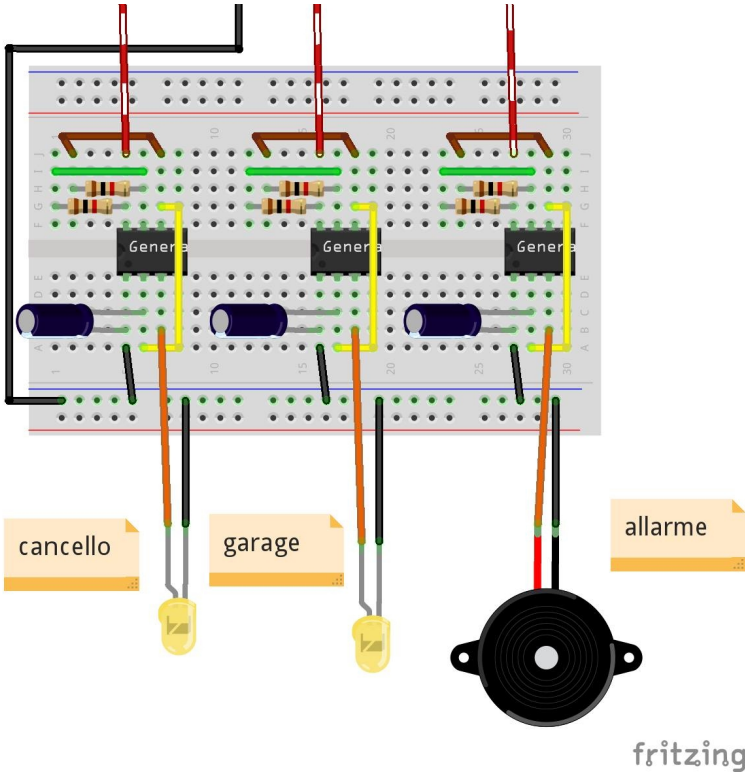
Il circuito dei fine corsa è molto semplice abbiamo solo utilizzato una resistenza di **pull-down** (10 kΩ) per evitare cortocircuiti e per inviare un segnale alto ad Arduino quando vengono eccitati.

Arduino riceve questo segnale in modo **asincrono** (in gergo tecnico è chiamato *interrupt*) in questo modo si interrompe appunto, la normale esecuzione del programma e Arduino provvede ad arrestare i motori istantaneamente.

Per il cancello abbiamo usato dei semplici pulsanti che vengono premuti dal cancello stesso, per il garage invece i pulsanti li abbiamo realizzati con delle semplici linguette di metallo.

Per la tenda abbiamo fatto un discorso a parte. In questo caso i fine corsa sono dei **sensori ottici** (digitali) formati da 2 led: uno a *infrarossi* che emette la luce e un altro che la riceve se si posiziona, a massimo 1 cm di distanza da esso, un corpo con una superficie che riflette la luce. Quindi il motore oltre a muovere la tenda stessa fa salire o scendere questo corpo che si posizionerà davanti ai sensori al termine della corsa della tenda.

Quando il cancello o i garage sono in movimento lampeggerà un led di segnalazione. Per consentire il lampeggiamento abbiamo realizzato un altro circuito usando l'**integrato NE555**. Questo integrato è un multivibratore che può essere configurato come monostabile, astabile e bistabile.



Noi necessitavamo di un *oscillatore* e quindi lo abbiamo configurato come **astabile**. Il periodo di oscillazione è determinata da 2 resistenze (1 kΩ) ed un condensatore elettrolitico (220 µf), sfruttandone la carica e scarica determiniamo i tempi dell'oscillazione e quindi anche la frequenza. Una resistenza determina la durata dell'impulso alto e l'altra dell'impulso basso.

Per quanto riguarda l'**anemometro** (sensore di vento) il discorso è completamente diverso in quanto abbiamo sfruttato la reversibilità del motore. Infatti il motore non è in nessuno modo alimentato. Esso è stato collegato nel seguente modo: un filo va a massa e un altro va in ingresso ad un pin analogico di Arduino consentendone la lettura del segnale.

Il principio di funzionamento è semplice, l'energia del vento fa ruotare l'elica montata sul motore che genererà una differenza di potenziale ai suoi capi.

Arduino legge questa **differenza di potenziale** (nell'ordine dei mV), se supera la soglia determinata dal programma accende il led di segnalazione posto all'interno dell'anemometro e se la tenda fosse aperta la chiude per evitare che il vento forte possa danneggiarla.

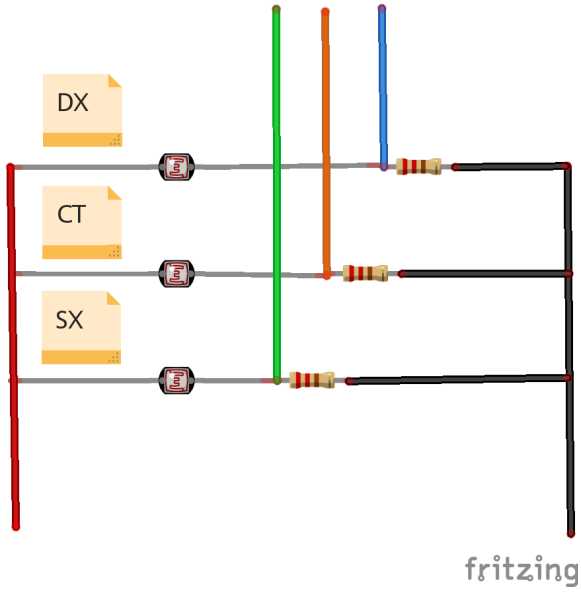
Inoltre il cancello automatico è equipaggiato da una “*fotocellula*” realizzata tramite il **sensore**

ad ultrasuoni HC-06. Questo sensore presenta 4 pin: 2 per l'alimentazione (5V),uno di uscita chiamatoTRIGG e un di ingresso chiamato ECHO.

Il funzionamento è molto semplice, tramite Arduino si invia un breve impulso (segnale) sul TRIGG, poi si legge la durata dell'impulso sul pin ECHO, la quale determina la distanza da esso di un eventuale oggetto. Quindi un impulso di breve durata implica che vi è un oggetto in prossimità della “fotocellula”e quindi il cancello se si sta chiudendo si ferma per poi riaprirsi automaticamente.

Inseguitore solare

La nostra casa è dotata di un **pannello fotovoltaico** da 9,5V (0.5A). Per far sì che il pannello abbia un rendimento migliore abbiamo realizzato l'**inseguitore solare**, in modo che il pannello si trovi “sempre” perpendicolare ai raggi solari.



L'inseguitore è dotato di 3 fotoresistenze: una centrale, una destra e una sinistra.

L'**algoritmo** è molto semplice, dopo aver effettuato le letture delle fotoresistenze viene calcolato il valore massimo, in questo modo è possibile stabilire come deve orientarsi il pannello. Se il valore massimo è della fotoresistenza centrale allora è orientato correttamente e il motore si ferma, se il massimo è della sinistra il motore si muoverà verso sinistra e viceversa.

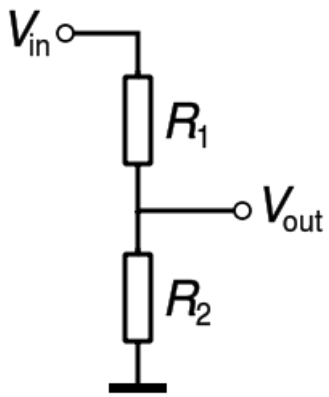
La parte più complessa è stata la messa in opera dei cavi, infatti, dato che il pannello è in movimento, sembrava impossibile non far attorcigliare i cavi. Ma, grazie all'aiuto del nostro amico e ingegnere *Enrico Navarra,* abbiamo realizzato dei **contatti striscianti** sull'albero del motore.

Il pannello solare di giorno carica una **batteria** da 8,4V cosicchè la sera quando non c'è il sole le luci vengono alimentate dalla batteria. Tuttavia dovevamo realizzare un meccanismo che controllasse la carica della batteria per evitare danneggiamenti da **sovraccarico.**

Il meccanismo è molto semplice, grazie ad un pin analogico di Arduino uno possiamo leggere

il livello di tensione della nostra batteria, ma dato che la batteria è da 8,4 volt abbiamo realizzato un **partitore di tensione** per avere in uscita 5V utilizzando la formula:

$$V_{out} = a * V_{ing} \quad , \quad a = \frac{R2}{R1 + R2} \quad \text{con} \quad a < 1$$



Avendo Vout pari a 5V e Vin pari a 8,4V abbiamo scelto R2 pari a 1kΩ, tramite il procedimento inverso ci siamo trovati che R1 deve essere pari a 680Ω. Ora che Arduino è in grado di “sapere” il **livello della batteria** il secondo passo è stato quello di collegare il pannello ad un relè in questo modo se il livello della batteria supera la soglia Arduino fa sganciare il relè o viceversa.

Modulo relè

Arduino non è in grado di pilotare autonomamente **grossi carichi** dunque per quei dispositivi che necessitano di molta potenza abbiamo utilizzato i relè.

Come funziona un relè?

Il **relè** è un deviatore che viene azionato da un **elettromagnete** che presenta 3 contatti uno mobile e 2 fissi. Quando si trova in stato di riposo il contatto mobile è connesso con uno dei contatti fissi mentre quando viene eccitato il contatto mobile si connette all'altro contatto fisso, in questo modo è possibile utilizzare il relè in configurazione *normalmente aperto* o *normalmente chiuso*: i nostri dispositivi sono tutti connessi in configurazione normalmente aperto, quindi I nostri relè fungono da interruttori quando vengono eccitati.

Nella casa abbiamo inserito un modulo composto da 4 relè per pilotare I seguenti carichi:

- fontana (220 VAC),
- impianto luci esterne (batteria),
- pannello fotovoltaico (9,5 VDC),
- ventola (12 VDC).

Per eccitarli Arduino invia un **segnale basso** poichè il modulo relè è costruito per essere **attivo basso**.

Uno dei relè è stato collegato alla pompa sommersa della fontana; una volta stimolato il relè si chiuderà e permetterà il passaggio della corrente. Alla fontana si affiancano altri due circuiti: uno per le **luci di segnalazione** e l'altro per il **sensore d'acqua**.

Le luci di segnalazione sono formate da tre led con tre rispettive resistenze di protezione (per evitatre che il led si danneggi): uno verde che si accenderà se la fontana è in funzione, uno rossa che Arduino accenderà se la fontana è ferma ed infine uno gialla che si accenderà per la segnalazione di eventuali guasti.

Per prevenire guasti abbiamo inserito un **sensore del livello dell'acqua**. Tale sensore invia un **segnale analogico** direttamente proporzionale al livello d'acqua nella fontana.

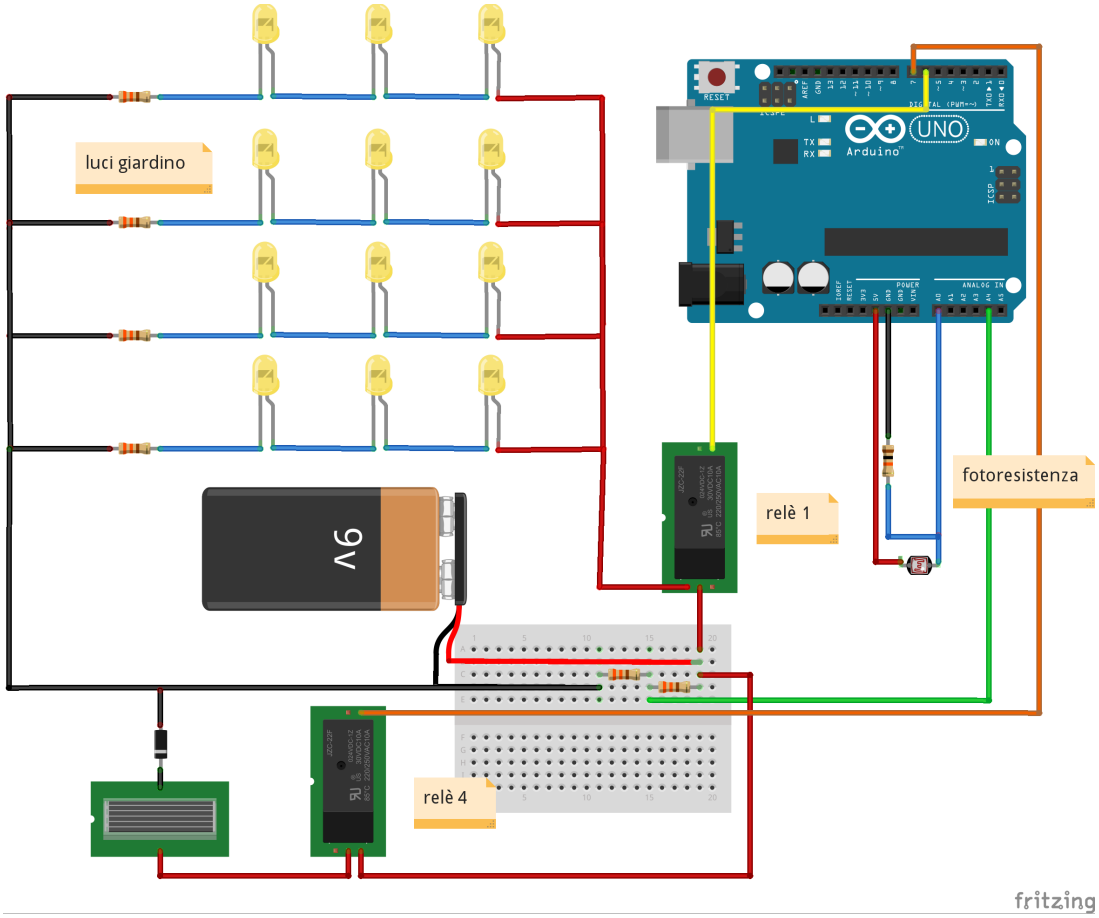
Arduino leggerà tale valore se è inferiore a una certa soglia, impostata nel programma in caso di mancanza d'acqua, esso accenderà la luce gialla per segnalare del problema (guasto) e se la fontana è accesa provvede a spegnerla.

Se in queste circostanze proviamo ad accendere la fontana Arduino farà lampeggiare il led rosso che ci indicherà che tale azione non è possibile.

Tale sistema è installato per evitare che il motore della pompa sommersa lavori a vuoto inutilmente.

Impianto luci giardino

l'**impianto luci** del giardino è composto da 9 lampioncini (quindi 9 led) ed è alimentato a tramite la batteria sopra citata. I lampioncini sono collegati in questo modo: 4 rami di 3 led in serie, collegati in parallelo tra loro.



Ad ogni **ramo** è collegato una **resistenza in serie** per protezione dei led, il suo valore è dato dalla formula:

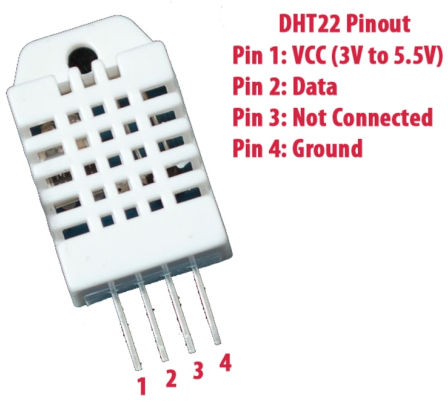
$$R = \frac{(V - N * V_L)}{I_L}$$

- V è la tensione di alimentazione 8,4V (dalla batteria)
- N è il numero dei led in serie quindi 4
- V_L è la caduta di tensione sul singolo led (2,2V)
- I_L è la corrente che assorbe il singolo led (20mA)

L'intero impianto di illuminazione esterna è abilitato da un **sensore LDR** (fotoresistenza), quindi grazie ad esso le luci si accendono automaticamente quando cala il sole.

Infatti la fotoresistenza è un sensore che varia la sua resistenza al variare della luminosità e quindi, in presenza di luce, emette un segnale alto mentre al buio emette un segnale basso.

Sensore di temperatura e umidità DHT22



Questo sensore utilizza una comunicazione di tipo **digitale** e grazie a questa cosa interfacciarlo con Arduino è molto semplice. Il sensore **DHT22** ha 4 piedini, ma solamente 3 di questi devono essere collegati ad Arduino (VCC, Data, GND). Lo svantaggio di questo sensore è la sua lentezza, infatti si parla di circa 2s tra una lettura e la successiva.

Per fare in modo che il sensore DHT22 venga riconosciuto dalla scheda Arduino bisogna aggiungere una libreria di terze parti. Inoltre viene consigliato l’utilizzo di un resistore da 10K fra il pin Data e VCC come resistore di pull-up.

Sensore di gas MQ5

L'interfaccia del **sensore MQ5** è la seguente:

- VCC a cui va collegata la tensione 5V
- GND a cui va collegata la massa comune
- AOUT su cui viene fornito un segnale analogico in tensione compreso tra 0 e 5 volt proporzionale alla lettura del sensore
- DOUT su cui viene fornito un segnale digitale che vale 0 o 5 volt a seconda che si superi o meno una soglia regolabile mediante un trimmer

Dei due led collocati nella parte posteriore, uno ricalca perfettamente il comportamento del PIN di uscita digitale. L’altro led, invece, indica semplicemente che il circuito è alimentato.

Quando il sensore rileva una perdita di gas, Arduino stampa il flag corrispondente sul display e innesca un **allarme sonoro** tramite un **buzzer**.

Display LCD

Il display LCD viene utilizzato come **monitor di controllo** sul quale vengono visualizzate: la temperatura e l'umidità interna della casa, nonchè una serie di *flag* che segnalano i vari eventi che si verificano. Gli eventi segnalati sono:

1. indicatori fontana
2. vento forte
3. cancello aperto
4. tenda aperta
5. fuga di gas

Il display in uso è un comune 16x2 pilotato tramite il **driver HITACHI HD44780** in questo modo è possibile programmarlo tramite la libreria standard di Arduino.

Per la comunicazione con Arduino abbiamo utlizzato un altro modulo ausiliario per il display (riducendo le connessioni da 8 a 2). Questo modulo implementa il protocollo di comunicazione **I2C** (*Inter Integrated Circuit*).

I2C è un *sistema di comunicazione seriale bifilare* utilizzato tra circuiti integrati. Il classico bus I2C è composto da almeno un *master* ed uno o più *slave*.

Ogni slave possiede un **indirizzo univoco** per poter essere riconosciuto dal master, in questo modo si evitano conflitti nella comunicazione.

Il **protocollo hardware** dell'I2C richiede due linee seriali di comunicazione:

- SDA (Serial DAta line) per i dati
- SCL (Serial Clock Line) per il clock (per questo l'I2C è un bus sincrono)