

# Automatismo Crepuscolare

Vittorio Polci

27 febbraio 2021

# 1 introduzione

Un automatismo crepuscolare è un circuito elettronico in grado di abilitare o disabilitare un relè secondo le ore del giorno.

In commercio sono presenti moltissimi apparecchi di questo genere, ma la maggior parte di essi non ha la possibilità di svolgere azioni troppo complesse o specifiche.

Questo automatismo crepuscolare ha infatti, oltre a un sensore che permette di rilevare l'intensità luminosa della luce solare, anche un sistema di ritardi e una serie di modalità di utilizzo che possono essere programmati facilmente dall'utente.

L'idea principale è quella di associare questa scheda elettrica ad un impianto fotovoltaico isolato (con delle batterie come accumulatori), in modo tale da attivare o disattivare carichi troppo pesanti (come ad esempio una pompa da 2KWH) a seconda delle condizioni atmosferiche e l'ora.

## 1.1 Principio di funzionamento

Il nostro interruttore in questione ha molti compiti da svolgere.  
Deve infatti:

- Saper rilevare la posizione del solare.
- Saper determinare se la quantità di luce solare è sufficiente per alimentare il carico in questione
- Capire se eventuali variazioni di luce solare possono essere ignorate
- Implementare eventuali ritardi di accensione o spegnimento

Capire se la quantità di luce solare è sufficiente e la posizione del sole è molto semplice. Basta infatti applicare al nostro circuito una semplice fotoresistenza e qualche altro componente in modo da poter verificare la tensione in uscita. La parte difficile è quella legata al sistema di ritardi e assorbimento.

*Perche è necessario un sistema di assorbimento?*

Prendiamo in considerazione il caso in cui una nuvola copra per qualche minuto i pannelli fotovoltaici. A causa dell'ombra, i pannelli non riusciranno a produrre energia sufficiente ad alimentare la pompa, quindi inizierà a prendere corrente da altre fonti (accumulatori o linea elettrica esterna).

Quindi se la nuvola rimane in quella posizione per molto tempo conviene spegnere direttamente il carico ed aspettare condizioni atmosferiche migliori, ma nel caso in cui la nuvola sorvolasse i pannelli solamente per pochi minuti lo spegnimento si potrebbe benissimo evitare (soprattutto perché il carico in questione potrebbe essere suscettibile ai troppi distacchi)

Il progetto naturalmente è mirato sia a minimizzare eventuali costi, sia a far in modo di preservare il funzionamento di altri componenti (come le batterie di un inverter).

*fotoresistenza:*  
componente elettronico capace di variare la sua resistenza interna secondo la luce ambientale

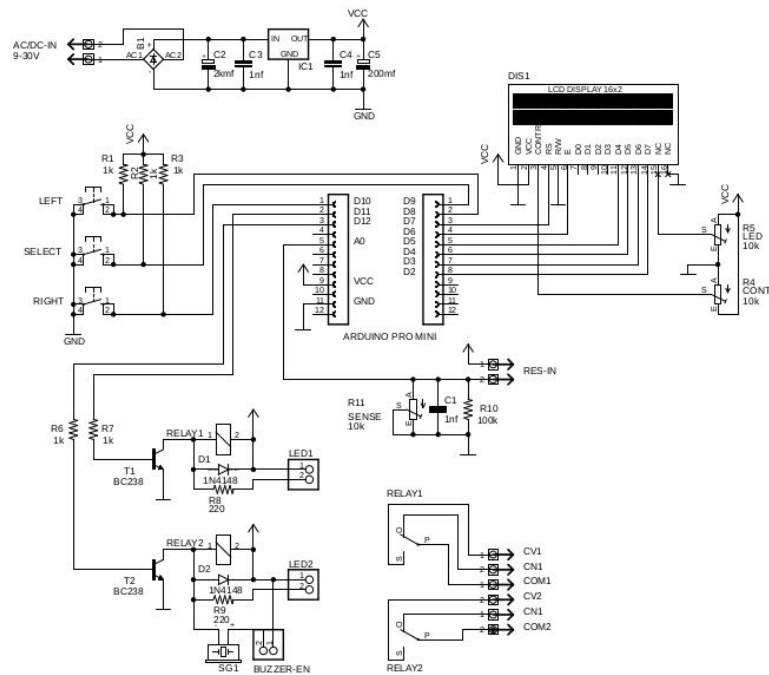


Figura 1: Lo schema elettrico dell'automatismo è stato costruito tramite il programma Autodesk Eagle

## 2 Hardware

Lo schema elettrico prevede l'utilizzo di un ATMEGA328 montato su una scheda Arduino Pro Mini, da programmare separatamente.

Ho scelto un microcontrollore arduino per la sua buona reperibilità sul mercato, ma soprattutto perché è molto facile da programmare.

La scheda comprende anche un display LCD da 16x2 (eventualmente modificabile) collegato con i primi piedini digitali D2-D7. Vicino ad esso sono presenti anche due resistenze variabili da 10kohm (R5 e R6) che regolano il contrasto e la luminosità del display.

Collegati ai piedini D9, D10 e D11 troviamo tre pulsanti (Left, Select e Right) e tre resistenze pull up (vengono utilizzate per evitare eventuali stati di incertezza). L'arduino capisce se uno dei pulsanti è stato premuto semplicemente vedendo se sul piedino *non arriva tensione*.

Sui piedini digitali D11 e D12 sono collegati due amplificatori di corrente formati da due transistor di piccola potenza, qualche resistenza e due diodi, collegati a due relè, che vengono utilizzati come uscite per gestire i due carichi.

Inoltre, per capire quali relè sono abilitati o disabilitati, sono stati montati due uscite led da due piedini ciascuno.

Insieme al led secondo relè è possibile inserire anche un buzzer piezoelettrico (che si può abilitare o disabilitare tramite un ponticello) per far sì di poter utilizzare la modalità *allarme* anche senza collegare una campanella esterna.

Sul piedino analogico A0 viene collegata un estremo di una fotoresistenza (l'altro

Nello schema elettrico l'arduino viene rappresentato con due Pin Headers

Il ponticello del buzzer va messo solamente nel caso in cui la modalità sia *allarme*

estremo viene collegato ai 5V) che viene portato a massa tramite la resistenza R10, un condensatore da 1nF, e il trimmer R11 (che serve per regolare la sensibilità).

Il tutto viene alimentato da un regolatore di tensione 78L05, dotato anche di condensatori e ponte di diodi, che fa in modo di stabilizzare l'ingresso (da 8 a 30 volt continua o alternata).

In tutto, tra uscite ed entrate, Sono presenti una morsettiera per l'ingresso dell'alimentazione, una per l'ingresso della fotoresistenza (che va piazzata in un punto in cui arriva bene la luce solare) e tre per i collegamenti di uscita dei due relè.

Le due uscite relè comprendono tre collegamenti:

- *C* -> polo comune
- *CN* -> collegamento non invertente
- *CV* -> collegamento invertente

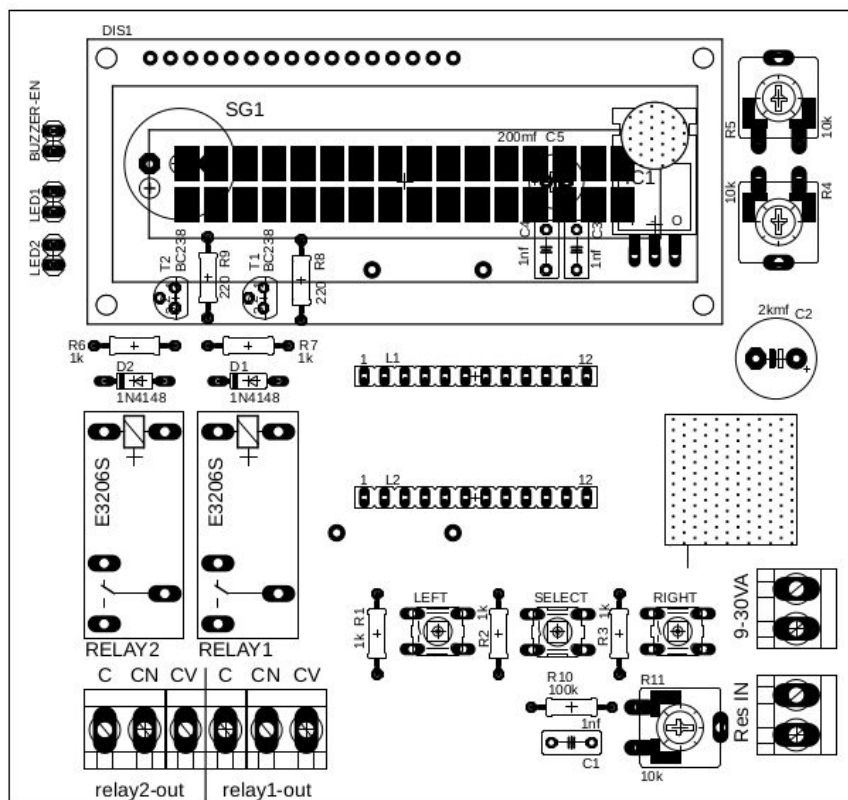


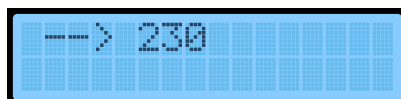
Figura 2: disposizione dei componenti

Sul circuito stampato alcuni componenti danno l'impressione di essere sovrapposti, ma in verità il display LCD si trova in una posizione rialzata rispetto agli altri componenti.

Lo schema elettrico può essere eventualmente modificato o ricostruito (i files della progettazione sono presenti su [GitHub](#) ) a seconda delle varie esigenze.

### 3 Avvio del sistema

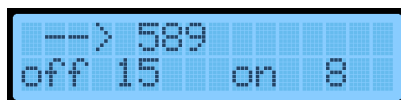
Una volta che si collega il circuito in modo adeguato all'alimentazione, il display si accenderà mostrando una scritta simile a questa



Il valore rappresentato dopo la freccia sta ad indicare la quantità di luce che il sensore crepuscolare rileva. Questo numero può andare da 0 (totalmente buio) fino a 1024 (totalmente luminoso).

Questo range può variare secondo la sensibilità, che si può regolare tramite l'apposito trimmer. Quando il valore va al di sotto della la soglia specificata nelle impostazioni (modificabili premendo il tasto *select*) allora sotto al valore appariranno anche delle scritte.

Nel caso in cui la modalità di funzionamento è impostato su *indipendente* il display mostrerà delle scritte simili a queste



Nella seconda riga possiamo notare due colonne. La prima è riferita al primo relè, mentre l'altra al secondo relè.

Se appare *off*, significa che il valore del sensore crepuscolare è sceso sotto a quello di soglia e, se il valore non torna al di sopra di essa, il relè si spegnerà alla fine del conto alla rovescia segnato a destra.

Al contrario, se appare *on*, significa che il valore del sensore crepuscolare è salito sopra a quello di soglia e, se il valore non torna al di sotto di essa, il relè di accenderà alla fine del conto alla rovescia a destra.

Quindi la scritta *off* appare solamente nel caso in cui il relè specificato è in fase di spegnimento, mentre la scritta *on* appare solamente se il relè è in fase di accensione.

Nella modalità *indipendente* un relè può anche entrare in fase di spegnimento mentre il secondo è in fase di accensione e viceversa.

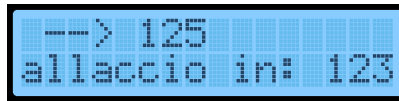
Nel caso in cui la modalità non sia *indipendente*, il display, in fase di spegnimento mostrerà



mentre in fase di accensione mostrerà

In partenza, i due relè rimangono spenti fino a quando non arriva la fase di accensione. Quindi molto probabilmente se il valore della del sensore segna che

I valori di soglia e i ritardi si possono modificare nelle impostazioni



è presente un a luce abbondante, all'avvio partirà automaticamente la fase di accensione.

## 4 Funzionalità principali

Le funzionalità del programma permettono all'utente di modificare le varie impostazioni con facilità e leggere tutti i dati campionati dal microprocessore tramite il display LCD e i tre tasti principali.

I valori possono essere facilmente inseriti tramite il menu *impostazioni*, che dispone di quattro modalità diverse, secondo le esigenze dell'utente:

- *indipendente*, permette di sfruttare i due relè in modo indipendente (con ritardi e valori di soglia indipendenti)
- *unita*, in questa modalità i due relè funzionano nella stessa maniera (quando si accende uno si accende anche l'altro e viceversa)
- *invertita*, in questa modalità i due relè funzionano in modo opposto (se si abilita uno s disabilita l'altro e viceversa)
- *allarme* utilizza il secondo relè per inviare un impulso sonoro

La modalità *allarme* utilizza il secondo relè per inviare un impulso sonoro nel momento dell'abilitazione o disabilitazione del primo. In questo modo, il primo relè verrà abilitato o disabilitato secondo l'intensità luminosa, mentre il secondo si abiliterà solamente per un periodo di tempo, modificabile, nel momento in cui il primo cambia di stato.

La modalità *indipendente* utilizza invece i due relè in modo indipendente. Questo richiede l'inserimento di due serie di valori diverse per ogni relè.

La modalità di configurazione si può abilitare premendo solamente il tasto *Select*

Da notare che la modalità di configurazione richiederà impostazioni diverse a seconda delle modalità inserita.

Una volta avviato il pannello di configurazione, l'utente dovrà inserire la modalità desiderata, utilizzando *Left* e *right* per modificare la scelta e *Select* per selezionare e salvare l'impostazione.



Figura 3: output su LCD

Una volta selezionata la modalità desiderata con il tasto select, l'utente deve inserire il valore di soglia della fotoresistenza (il valore minimo) relativo al primo



Figura 4: output su LCD

relè, sempre utilizzando i tre tasti come nel menu precedente. Nel caso in cui la modalità viene impostata in *indipendente*, verrà richiesto anche l'inserimento del valore di soglia del relè 2.

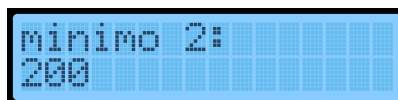


Figura 5: output su LCD

Una volta inseriti i valori desiderati, l'utente dovrà inserire il ritardo di spegnimento relativo al relè 1

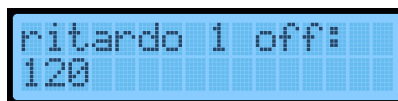


Figura 6: output su LCD

il ritardo di spegnimento fa in modo di spegnere il relè dedicato solamente se il valore del sensore scende sotto quello di soglia per un tempo maggiore di quello inserito in questa impostazione.

Naturalmente con la modalità *indipendente* va inserito anche il ritardo di spegnimento del secondo relè.

Tutti i valori di ritardi e durata sono in secondi e vanno da un minimo di 0 ad un massimo di 999

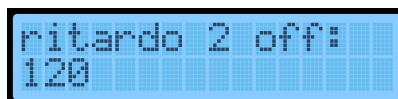


Figura 7: output su LCD

Una volta inserito il ritardo di spegnimento, va inserito anche il ritardo di accensione, che ha un principio di funzionamento molto simile al precedente. il ritardo di accensione fa in modo di accendere il relè dedicato solamente se il valore del sensore sale sopra quello di soglia per un tempo maggiore di quello inserito in questa impostazione.

Nella modalità *indipendente* va inserito anche il valore relativo al secondo relè

Nel caso in cui sia inserita la modalità *allarme*, al posto dei ritardi e dei valori di soglia del secondo relè, bisognerà impostare anche una durata dell'avviso.

Oltre alla modalità di configurazione, esistono anche altre due funzioni:



Figura 8: output su LCD

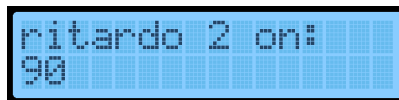


Figura 9: output su LCD

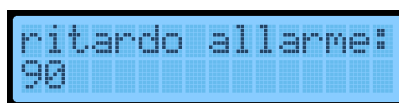


Figura 10: output su LCD

- Reset impostazioni, si avvia premendo i due tasti *Select* e *left* contemporaneamente
- Test relè, si avvia premendo i due tasti *Select* e *Right* contemporaneamente

Il reset delle impostazioni va ad azzerare tutti i valori della EEPROM interna del microcontrollore, impostando quindi dei valori predefiniti:

- modalità: indipendente
- ritardo spegnimento e accensione relè1: 0
- ritardo spegnimento e accensione relè2: 0
- ritardo allarme (non visibile dopo il reset): 0

Una volta effettuato il reset apparirà questa scritta sul display LCD per 3 secondi.



Il test invece fa in modo di attivare e disattivare 5 volte i due relè con una pausa di 100ms tra cambio di stato. Si raccomanda, per evitare il danneggiamento dell'apparecchiatura, di scollegare il carico presente all'uscita dei relè prima di fare questo test.

L'ultima funzione, che è interna al programma, fa in modo di riavviare automaticamente la scheda dopo un certo periodo di tempo, che varia da 1 secondo a 52 giorni (impostabile modificando una costante nella programmazione). In questo modo si possono evitare eventuali malfunzionamenti legati ad eventuali agenti esterni, come ritorni di corrente inaspettati, che possono compromettere il funzionamento del sistema.



## 5 Software

Come già spiegato prima, questa scheda utilizza un arduino pro mini. La programmazione in C++ scritta direttamente dal suo IDE si può trovare nella pagina [GitHub](#) di questo progetto.

### 5.1 Struttura

Il programma è strutturato in diverse sezioni:

- Dichiarazione delle costanti
- Definizione delle funzioni
- Esecuzione del primo avvio
- Esecuzione dei cicli del programma principale
- Implementazione delle funzioni

### 5.2 Dichiarazione delle costanti

```
//definizione dei pin
#define pinSelect 8 //ingresso digitale del tasto select (impulso negativo alla pressione)
#define pinRight 9 //ingresso digitale del tasto right (impulso negativo alla pressione)
#define pinLeft 10 //ingresso digitale del tasto left (impulso negativo alla pressione)
#define resIn 0 //ingresso analogico per la fotoresistenza
#define relay1 11 //uscita digitale del relè 1
#define relay2 13 //uscita digitale del relè 2
#define refresh 100 //tempo di attesa per nuovo campionamento
#define ResetSec 86400 //secondi dopo la quale il microcontrollore si riavvia automaticamente
LiquidCrystal lcd(7,6,5,4,3,2); //definizione dei piedini del display

//definizione debug mode
#define DebugMode true //abilita la modalità di debug (esclude tutto l'hardware input esterno e prende in considerazione gli input solamente la porta seriale)
#define SerialOut true //abilita o disabilita l'output anche su porta seriale (da abilitare se si vuole eseguire il debug esclusivamente su porta seriale)
#define baudrate 9600 //velocità della porta seriale (utilizzata per il debug)

//definizione delle configurazioni
#define ConfigDelay 200 //ritardo nella fase di configurazione
#define eepromMode 0 //locazione contenente la modalità di configurazione
#define eepromMinVal 1 //locazione contenente il primo valore minimo (comprende 2 bytes)
#define eepromMinVal2 9 //locazione contenente il secondo valore minimo (comprende 2 bytes)
#define eepromDelay1 3 //locazione contenente il ritardo dello spegnimento del primo relè (comprende 2 bytes)
#define eepromDelay2 5 //locazione contenente il ritardo di accensione del primo relè (comprende 2 bytes)
#define eepromDelay3 7 //locazione contenente il ritardo dell'allarme (comprende 2 bytes)
#define eepromDelay4 11 //locazione contenente il ritardo dello spegnimento del primo relè (comprende 2 bytes)
#define eepromDelay5 13 //locazione contenente il ritardo di accensione del primo relè (comprende 2 bytes)
```

Figura 11: Parte del codice che prevede la dichiarazione delle costanti

Le costanti sono molto utili quando si programma codici abbastanza lunghi. Nella prima parte si trovano le dichiarazioni dei pin digitali e analogici, nella seconda quelle relative alla porta seriale e nell'ultima le posizioni dei dati nell'eeprom interna del microcontrollore.

Nel caso in cui vengono, ad esempio modificate le posizioni dei pin, per far adattare il programma basta cambiare queste costanti.

Nella seconda parte notiamo che, oltre alla velocità della porta seriale anche la costante *SerialOut* (se abilitata, permette di inviare sulla porta seriale principale tutti gli output che vengono inviati sul displayLCD) e *DebugMode* (se abilitato, permette di escludere l'hardware esterno di input e di emulare l'inserimento dei dati direttamente tramite porta seriale)

Se si abilitano entrambe le dichiarazioni, è possibile testare il programma senza bisogno di creare un circuito stampato.

Se *DebugMode* è abilitato, una volta collegato l'arduino alla sua alimentazione, verrà automaticamente abilitata la modalità di configurazione. A questo

```

modalità:
allarme
minimo 1
7
minimo 1
7
ritardo 1 off
5
ritardo 1 off
5

```

punto l'utente può emulare la pressione dei tasti ("+" per Right, "-" per Left e "c" per Select) semplicemente inviando i caratteri specifici nella porta seriale. Una volta completata la modalità di configurazione, si avvia il ciclo di ricezione dei dati analogici.

Nella modalità debug arduino attende l'inserimento dei dati nella porta seriale per continuare l'esecuzione del programma

```

-->
12
: -->
90
allaccio in
3

: -->
0
allaccio in
2
:

```

A questo punto l'utente deve semplicemente inserire i valori luminosi quando viene stampato un ":" all'estremo sinistro della pagina. Eventuali condizioni di allaccio o distacco dei relè vengono segnalati subito dopo l'inserimento del dato. Tuttavia la modalità di Debug non simula l'abilitazione e la disabilitazione fisica dei relè. Per verificare quindi se vengono abilitati o disabilitati correttamente basta modificare la dichiarazione dei pin di output, ad esempio modificando l'output del relè 1 a 13 (in questo modo si può visualizzare sul led principale se quella porta viene abilitata o disabilitata correttamente)

### 5.3 Funzioni modificabili

Per inviare in output i dati o per riceverli in input, vengono utilizzate delle funzioni generali. Questo perché, nel caso qualcuno volesse sostituire il display o ad-

dirittura il sensore analogico, è possibile modificare il programma semplicemente andando a riscrivere queste funzioni.

Le funzioni in questione sono:

- *photoIn()*, ritorna il valore preso dal piedino analogico del sensore crepuscolare.
- *buttonIn()*, ritorna un numero intero da 0 a 5 secondo le varie combinazioni che si eseguono premendo i tasti.
- *printLcd(char\*/int val,row,col)*, invia al display LCD una stringa o un valore intero nella posizione determinata dai due numeri interi *row* e *col*
- *printLcdN(char\*/int val)*, invia al display LCD una stringa o un valore intero senza specificare la posizione

la funzione *buttonIn()* restituisce un valore da 0 a 5 secondo le seguenti combinazioni:

- 0 -> nessun tasto premuto
- 1 -> tasto *Left* premuto
- 2 -> tasto *Select* premuto
- 3 -> tasto *Right* premuto
- 4 -> tasti *Left* e *Select* premuti simultaneamente
- 5 -> tasto *Right* e *Select* premuti simultaneamente