Dispositivi di potenza con Arduino

I pin di Arduino possono fornire solo una certa corrente in uscita, se il carico, ovvero il dispositivo che voglio alimentare, richiede più corrente di quella erogabile, il circuito relativo al pin in questione si danneggia. Inoltre, sempre parlando di correnti, la somma delle correnti in uscita dai pin di Arduino non deve superare un determinato valore pena, anche qui, il danneggiamento del microcontrollore che costituisce il cuore della nostra scheda preferita.

Ultimo ma altrettanto degno di nota è il fatto che le uscite hanno una tensione fissa di 5V che non sempre corrisponde alla tensione nominale di funzionamento del dispositivi che vogliamo controllare (es motore in c.c. a 12 volt o una elettrovalvola a 24 v).

Resistenze, tensioni, correnti e potenze

Prima di poter continuare ho dovuto fare i conti con i parametri fondamentali che descrivono elettricamente un dispositivo e alcune regole dell'elettrotecnica che mettono in relazione questi parametri. Non voglio entrare nei dettagli ma giusto annotare un paio di importanti concetti di base che devono sempre essere tenuti presenti quando si progetta un sistema elettrico/elettronico. (Non me ne vogliano gli esperti se per semplicità ho trascurato qualche particolare/dettaglio in favore della comprensibilità)

Legge di Ohm

E' una legge fondamentale che nella sua semplicità racchiude tutto ciò che accade in un qualsiasi bipolo passivo lineare ideale in corrente continua (Lo so, a questo bipolo sono riferiti un sacco di aggettivi strani, un giorno, se a qualcuno interesserà ne parleremo). Si enuncia così:

```
V = R * I  [volt]
```

Dove V è la tensione ai capi del bipolo, I la corrente che lo attraversa e R la sua resistenza.

Solitamente, nella pratica, R è data e dipende dalle caratteristiche costruttive del bipolo che stiamo alimentando (nel nostro caso il dispositivo che vogliamo controllare, anche detto carico). Altrettanto solitamente la tensione nominale di alimentazione V è data. Quindi la corrente I che scorrerà nel nostro dispositivo quando alimentato alla sua tensione nominale di alimentazione sarà:

```
I = V / R  [amper]
```

Da questa seconda equazione si deduce anche un'altra importantissima cosa: la tensione (o differenza di potenziale) è la causa, la corrente è l'effetto. Per i nostri ragionamenti vale quindi

questa regola: imposta un differenza di potenziale tra due punti tra i quali esiste una certa resistenza, scorrerà una certa corrente.

Potenza in corrente continua

La seconda legge che serve per capire cosa succede quando si collegano i carichi a delle sorgenti di alimentazione è quella che consente di calcolare le potenze. Si enuncia così:

```
P = V * I   [watt]
```

Cioè la potenza P dissipata/erogata da un bipolo (es. resistenza) è pari al prodotto della tensione ai suoi capi per la corrente che lo attraversa.

Sempre semplificando un po', a seconda del tipo di dispositivo, parte della potenza può essere trasformata per compiere il compito per cui il dispositivo è stato concepito e parte verrà dispersa in calore (secondo un parametro detto rendimento).
La parte che viene dissipata in calore è quella che, se non correttamente gestita, determina in vario modo e misura guasti ai dispositivi. Se si supera la potenza MAX. gestibile da un bipolo si rischia di bruciarlo.

Superare i limiti operativi di Arduino

La prima cosa da capire è quindi quali siano le correnti massime prelevabili dai singoli piedini e quale sia il limite massimo di corrente assorbibile da tutti i piedini di output contemporaneamente attivi. I datasheets del ATmega328 ci offrono molte informazioni tra cui anche quelle che stiamo cercando:

- Corrente massima in uscita da un piedino di output: 40 mA
- Somma delle correnti in uscita massima: 200 mA

Con una tensione di 5V significa una potenza massima di 0,2W per pin e 1W totali.
Giusto per fare qualche esempio un piccolo motore DC alimentato a 5V assorbe facilmente 250mA o la bobina di un relay tipo lo Zettler AZ8222-2C-5DSE assorbe 40 mA.

Tradotto in altri termini questo significa che non possiamo collegare ad Arduino carichi che assorbano potenze superiori a quelle massime erogabili da ciascun pin e da tutta la scheda pena la distruzione del microcontrollore.

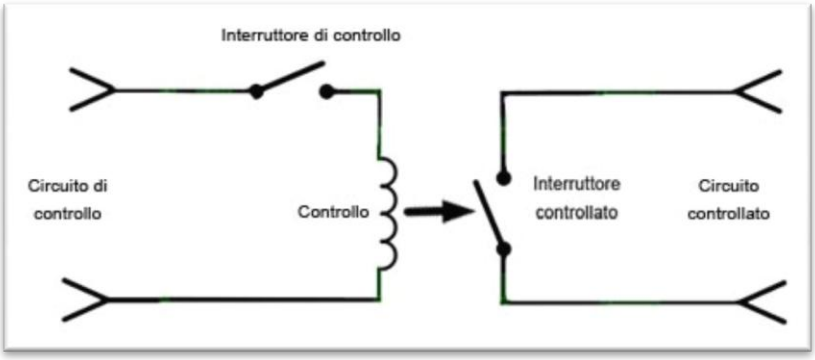

Figura 1: interruttore controllato

Per superare questi limiti si utilizzano le uscite come segnali di controllo per pilotare degli "interruttori controllati" o "comandati" posti tra l'alimentazione principale e il carico da controllare. Quando l'uscita è alta, cioè +5V, l'interruttore è chiuso e collega il carico alla alimentazione (acceso); Quando l'uscita è bassa, cioè 0V, l'interruttore è aperto ed il carico è scollegato dall'alimentazione (spento).

Il transistor come interruttore

Il modo più semplice di realizzare in elettronica un "interruttore comandato" è quello di utilizzare un transistor in commutazione ovvero un transistor che lavori nelle zone di saturazione/interdizione.

In queste zone infatti il dispositivo si comporta in modo molto simile ad un interruttore facendo passare tutta la corrente possibile o bloccandola completamente tra due dei sui tre piedini in funzione del segnale applicato tra il terzo piedino ed un piedino comune.

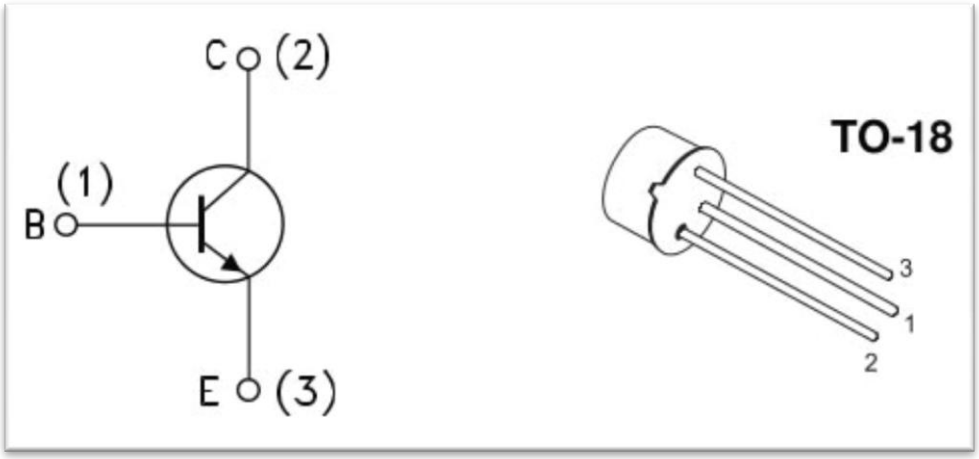

Figura 2: Transitor con base B, collettore C e emettitore E
(fonte 2N2222A datasheet – STMicroelectronics)

Con riferimento alla figura, il piedino comune è detto emettitore ed è collegato a massa, il piedino al quale applicare il segnale proveniente dal pin di controllo è detto base, il pedino a cui collegare un terminale del carico (nello schema rappresentato da Rc) è detto collettore (l'altro terminale del carico sarà collegato alla alimentazione).

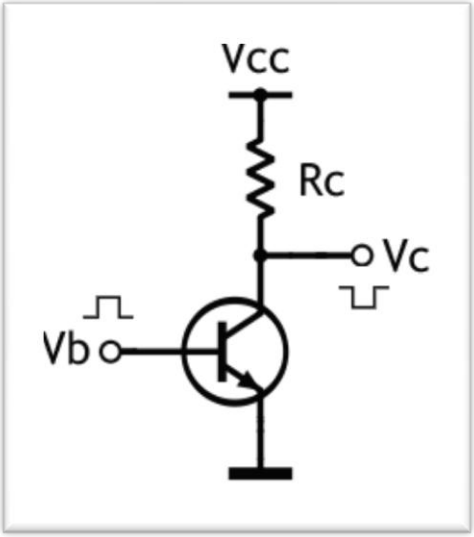

Figura 3: Transistor come interruttore

Quando il segnale di controllo applicato alla base è positivo (HIGH logico), passerà tra base e emettitore una corrente positiva che, se sufficientemente grande, porterà il transistor in saturazione. Quando il transistor si trova in questo stato la tensione tra collettore e emettitore tende a zero, e tutta la tensione di alimentazione è applicata al carico.

Viceversa quando il segnale di controllo applicato alla base è zero (LOW logico), tra base e emettitore non scorrerà corrente e il transistor lavorerà in interdizione. Quando il transistor si trova in questo stato tutta la tensione di alimentazione "cade" tra collettore e emettitore e il carico risulta non alimentato.

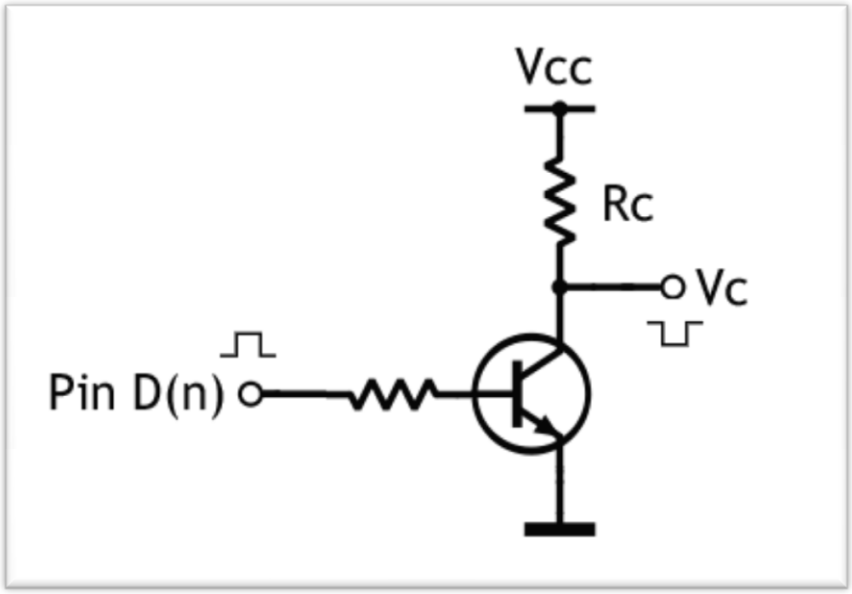

Figura 4: Transistor collegato ad Arduino con Rb

Affinché la corrente che passa nella base del transistor sia sufficiente per portare il dispositivo in saturazione ma non troppo intensa da danneggiarlo è necessario porre tra il pin di controllo e il terminale di base un resistenza. Il suo calcolo è abbastanza semplice.

Calcolo Rb

Per calcolare la resistenza di base si procede a ritroso partendo dalla analisi della parte di circuito relativa al carico. Innanzitutto si deve determinare la corrente che scorrerà attraverso il carico, e quindi attraverso il collettore, quando il transistor sarà in saturazione:

```
Ic = (Vcc – Vce sat.) / Rc
```

La corrente di collettore sarà pari alla tensione presente sul carico (Vcc – Vcesat dove Vcc è la tensione di alimentazione e Vcesat è la tensione tra collettore e emettitore del transistor quando si trova in saturazione – dato disponibile sui datasheets del componente) diviso la resistenza del carico.

Una volta nota la corrente di collettore possiamo calcolare la corrente di base sapendo che Ic = hfe * Ib (dove Ib è la corrente di base e hfe il guadagno del transistor come da datasheets) quindi:

```
Ib = Ic / hfe
```

A questo punto si può analizzare il circuito di ingresso considerando la tensione del segnale di controllo, nel nostro caso 5V (valore alto di una uscita digitale di Arduino) la tensione tra emettitore e base Vbe (che si trova sui datasheets) e la corrente Ib desiderata e calcolata al passo precedente. Dato che:

```
Ib = (Vpin - Vbe ) / Rb
```

avremo che:

```
Rb = (Vpin - Vbe) / Ib
```

Questa è la resistenza massima da utilizzare sulla base del transistor per assicurare che vada in saturazione (cioè quella che garantisce una Ib sufficiente alla saturazione). Valori di resistenza minori comportano Ib maggiori e quindi una maggiore sicurezza di far lavorare il transistor nella zona corretta.

Esiste però un limite inferiore dettato da due parametri fondamentali:

- La corrente massima erogabile da un pin di Arduino
- La corrente massima assorbibile dalla basse del transistor

Chiamando il valore minore tra i due Ib(max) avremo che:

```
Rb(min) = (Vpin - Vbe) / Ib(max)
dove
Ib(max) = Ic(max) / hfe
```

Il valore finale di Rb dovrà essere compreso tra Rb e Rb(min) e scelto in funzione della serie di resistori a disposizione (ad esempio E12).

Esempio di calcolo

Vediamo concretamente come calcolare la resistenza di base nel caso in cui vogliamo alimentare un piccolo motore DC la cui tensione di alimentazione è 5V e la resistenza interna è di 20 Ohm con un transistor 2N2222 usato come interruttore.

$V_{CE(sat)}$*	Collector-Emitter Saturation Voltage	I_C = 150 mA I_C = 500 mA	I_B = 15 mA I_B = 50 mA			0.3 1	V V
$V_{BE(sat)}$*	Base-Emitter Saturation Voltage	I_C = 150 mA I_C = 500 mA	I_B = 15 mA I_B = 50 mA	0.6		1.2 2	V V
h_{FE}*	DC Current Gain	I_C = 0.1 mA I_C = 1 mA I_C = 10 mA I_C = 150 mA I_C = 500 mA I_C = 150 mA I_C = 10 mA T_{amb} = -55 °C	V_{CE} = 10 V V_{CE} = 10 V V_{CE} = 10 V V_{CE} = 10 V V_{CE} = 10 V V_{CE} = 1 V V_{CE} = 10 V	35 50 75 100 40 50 35		300	

Tabella: transistor 2N2222, estratto datasheets
(fonte 2N2222A datasheet – STMicroelectronics)

I dati sul transistor che ci interessano e che sono disponibili sui datasheets del componente sono i seguenti:

- Vcesat = 0.3 V
- Vbesat = 0.6 V
- hfe = 100
- Ic(max) = 600 mA

I dati del motore necessari per il calcolo sono invece:

- Ic = 250 mA

Quindi:

```
Ib = 250 mA / 100 = 2,5 mA
```

da cui:

```
Rb = (5 V - 0.6 V) / 2,5 mA = 1760 Ohm
```

Nota sull'esempio:
I motori sono carichi induttivi che richiedono alcuni accorgimenti ulteriori per evitare che correnti di scarica possano circolare in modo incontrollato nel transistor e causarne la rottura. Per evitare queste correnti anche dette di fly-back è necessario posizionare un diodo ai capi del carico induttivo con il catodo collegato al terminale connesso all'alimentazione e l'anodo collegato al terminale connesso al collettore del transistor. Questo vale anche per gli esempi successivi in cui si utilizzano relay (che al loro interno hanno un elettromagnete, cioè un induttanza) per aumentare la potenza del carico comandato.

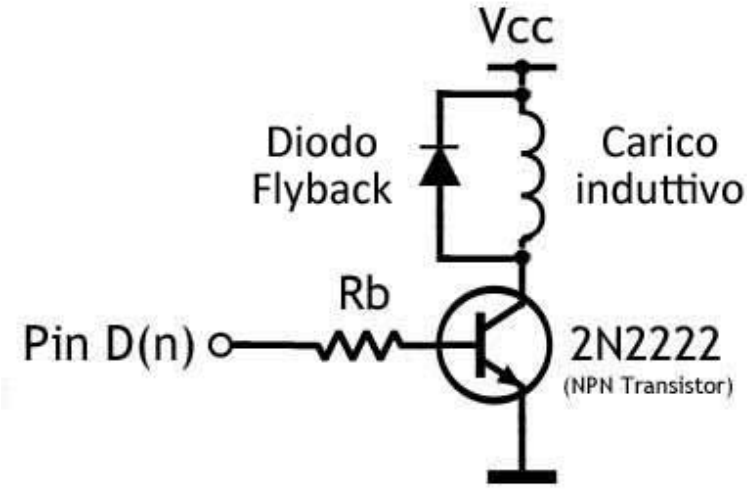

Figura 5: Diodo Flyback (antiparallelo) su carico induttivo

Transistor a doppio stadio (darlington)

Nel caso in cui le correnti di collettore siano tali da non poter utilizzare un transistor BJT semplice come il 2N2222 è possibile passare a componenti capaci di sopportare correnti più elevate. Il problema dei transistor più potenti è che necessitano di correnti di base maggiori di quelle erogabili da Arduino. La soluzione è utilizzare uno schema a due stadi collegati come in Figura 6.

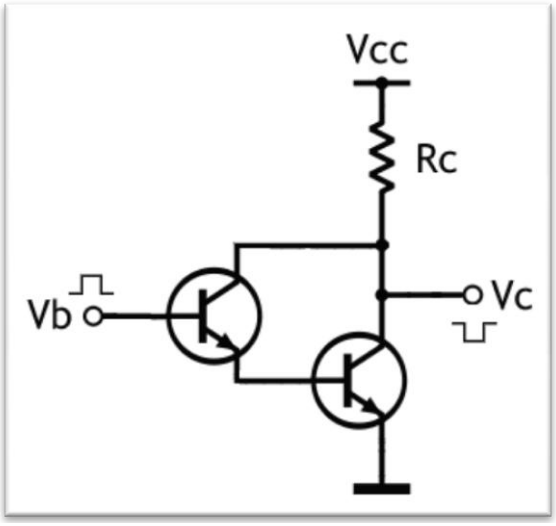

Figura 6: Schema doppio stadio, configurazione darlington

Esistono componenti tipo i **TIP120**: coppia di transistor in configurazione darlington, che realizzano questa configurazione in un unico dispositivo.

MOSFET, una alternative ai BJT

Se i transistor bipolari (BJT) sono una ottima scelta quando le correnti richieste dal carico rimangono nell'ordine delle centinaia di mA (600mA/800mA) e le tensioni di alimentazione intorno alle decine di volts (20V/30V), per correnti nell'ordine degli A (1A/10A) e tensioni intorno al centinaio di V iniziano ad essere interessanti i MOSFET.

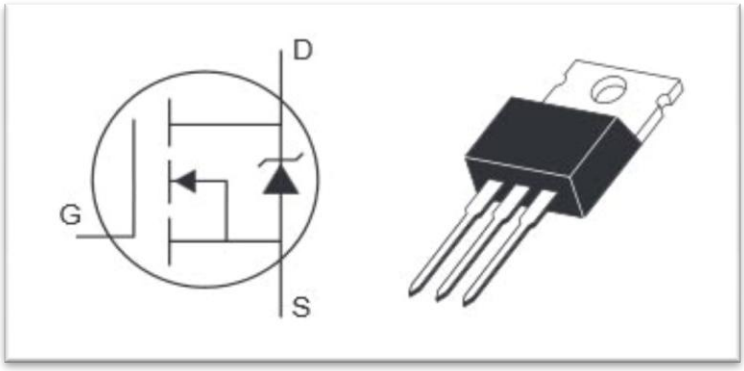

Figura 7: simblo e aspetto di un MOSFET

A differenza dei BJT che si comandano in corrente, i MOSFET sono comandati in tensione e questo nel campo dei microcontrollori TTL può costituire un problema dato che solitamente la tensione necessaria per farli andare in conduzione è dell'ordine dei 10V, di gran lunga maggiore dei 5V a cui lavora Arduino.

PRODUCT SUMMARY		
V_{DS} (V)	100	
$R_{DS(on)}$ (Ω)	V_{GS} = 5.0 V	0.077

Figura 8: Dati su resistenza Drain-Surce a Vgs=5V

(Fonte: datasheet Vishay)

Esistono fortunatamente dei MOSFET tipo l'IRL540 che sono appositamente progettati per andare in 'OnState' con una tensione Vgs in ingresso di 5V.

L'altra grande differenza riguarda la frequenza alla quale far lavorare il componente: mentre un BJT può essere spinto senza troppi accorgimenti a frequenze dell'ordine dei 100kHz, un MOSFET già a qualche kHz necessita di particolari accorgimenti.

Solitamente però, nella applicazioni a micocontrollore, questo aspetto non è un problema.

MOSFET in parallelo

A differenza dei BJT i MOSFET possono essere collegati in parallelo e quindi permettono una soluzione 'scalabile' che consente cioè di gestire correnti maggiori semplicemente aumentando il numero di dispositivi impigati.

Uno sguardo alle applicazione in AC

Fino ad ora abbiamo considerato applicazioni in corrente continua. Esistono però numerose applicazioni in cui il carico di potenza è un dispositivo che funziona in corrente alternata.

Una buona soluzione per gestire carichi in corrente alternata tramite un microcontrollore è l'utilizzo di un TRIAC.

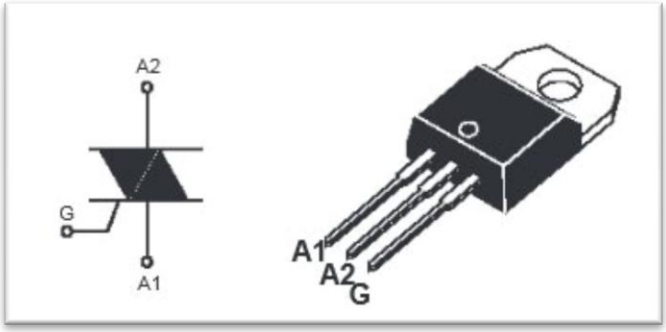

Figura 9: Simbolo e aspetto di un TRIAC

Per potenze ancora maggiori

Il passo successivo è quello di utilizzare al posto di un secondo stadio a semiconduttore come nel caso del tipo120, un relay.

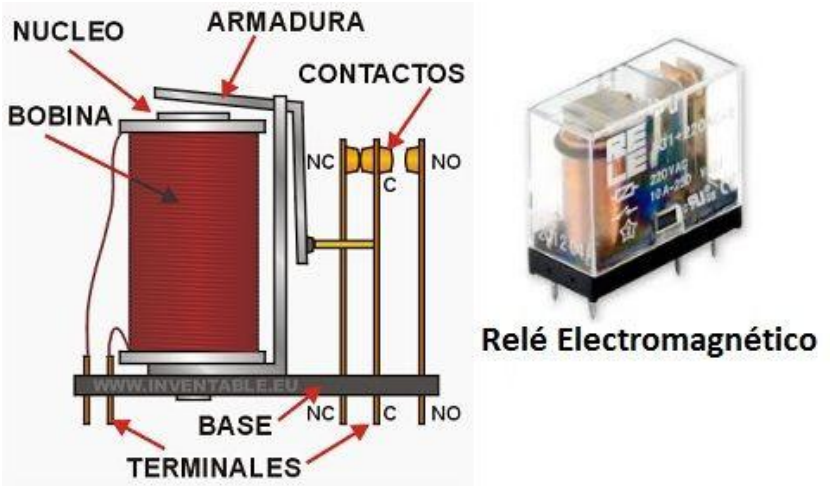

Figura 10: relay

In questa configurazione il transistor garantisce la potenza necessaria per comandare la bobina del relay la quale comanda a sua volta dei contatti meccanici. Per quanto riguarda il carico, in questo caso, l'interruttore comandato è a tutti gli effetti un interruttore.

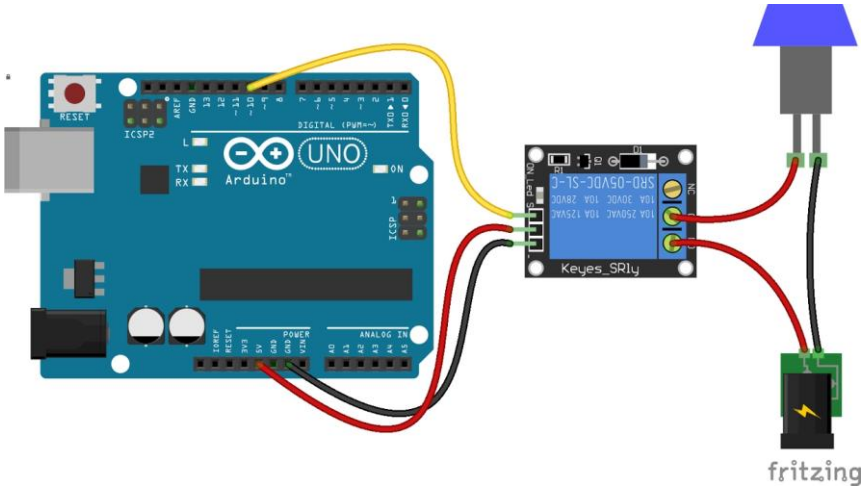

Figura 11: Arduino comanda relay

Se non è ancora abbastanza si potrebbe pensare di utilizzare un relay a 5V per comandare un contattore, tipo l'Hager ES224 o ES424 al quale collegare il carico finale.

Una tipica applicazione in cui applicare un modello di questo tipo è il controllo di sistemi di riscaldamento/raffrescamento degli ambienti.

Nota sulle applicazioni in alta tensione

Gli ultimi esempi sono stati inclusi in questo articolo per completezza ma sono volutamente generici perchè riguardano applicazioni che richiedono conoscenze specifiche nella manutenzione degli impianti elettrici in alta tensione. Intervenire su impianti in alta tensione senza adottare le necessarie precauzioni comporta rischio di folgoramento e di morte!

Trade off potenza/velocita'/costo

Le tre soluzioni proposte sopra: transistor, transistor doppio stadio e relay hanno ovviamente ambiti di applicazione molto diversi identificati soprattutto dai fattori: potenza, velocità di commutazione e costo.

Per quanto riguarda la potenza gestibile abbiamo visto che cresce passando dal transistor semplice, al doppio stadio al relay.

Anche per quanto riguarda il costo dei componenti la progressione segue lo stesso ordine: la soluzione transistor+relay è la più costosa, soprattutto per la presenza del relay, il doppio stadio una via intermedia e il transistor semplice la più economica.

Per quanto riguarda la velocità di commutazione l'ordine è invece l'inverso: il più veloce è il transistor singolo. Segue il doppio stadio.

Chiude la classifica il relay che a causa delle componenti meccaniche ha tempi di commutazione limitati dalla fisicità dei contatti che si devono muovere (circa 50 ms per una doppia commutazione -> 200 volte al secondo).