

Scheda video: PWM

Con una scheda Arduino si comandano in PWM (Pulse Width Modulation) la luminosità di una lampadina e la velocità di un motore utilizzando la piattaforma di programmazione "Snap for Arduino". Un oscilloscopio misura la tensione sull'interruttore su ciascuno dei carichi. Un grafico generato dal programma sul PC illustra la tensione sulla lampadina simulando un oscilloscopio ed uno strumento analogico.

Preparazione

Lo schema elettrico rappresentato alla destra comprende:

- il **carico L** composto da una lampadina dicroica 12V, 20W che assorbe 1,7A
- l'**alimentazione V_{batt}**, fornita da una batteria al Pb da 12V
- l'**interruttore T** di tipo elettronico che è parte della circuiteria di controllo realizzata con la scheda Arduino
- il **generatore di segnale** ad onda quadra a duty cycle variabile per il comando dell'interruttore
- l'**oscilloscopio** che misura l'uscita ai capi dell'interruttore e mostra la forma d'onda.

Con lo stesso schema ma su altra uscita viene alimentato un motoriduttore 6V, 130rpm, che assorbe 400mA a vuoto (nota: per brevi periodi il motoriduttore può sopportare anche tensioni di 12V)

Il generatore di segnale, realizzato con l'uso della scheda Arduino. Poiché le correnti assorbite da ciascun carico superano ampiamente le possibilità di output della scheda Arduino (circa 40mA), viene utilizzato una scheda "RGB shield" che utilizza il MOSFET STP36NE06 sulle uscite 3, 5, 6 che è capace di correnti fino a 36A con una $R_{on} = 0,040\Omega$ (figura A).

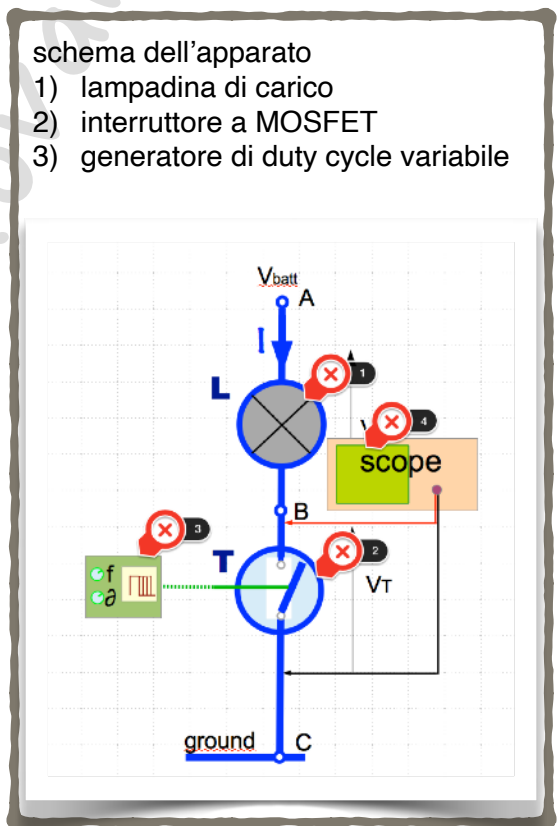
L'interruttore T nello schema è realizzato con il MOSFET della scheda RGB.

L'alimentazione della scheda viene posta dal lato "power" per non caricare l'apparato della scheda Arduino ed il PC che la alimenta.

La scheda Arduino genera su alcuni piedini un segnale analogico in PWM che fornisce valori a due livelli logici ma con durata dell'impulso variabile da zero al 100% inviando valori compresi fra 0 e 255 e viene comandata dal software di "Snap for Arduino".

Il software consente di aumentare o diminuire il valore del duty cycle da 0 a 100 tramite tastiera e di calcolare il valore di PWM da inviare alla scheda Arduino nella gamma suddetta.

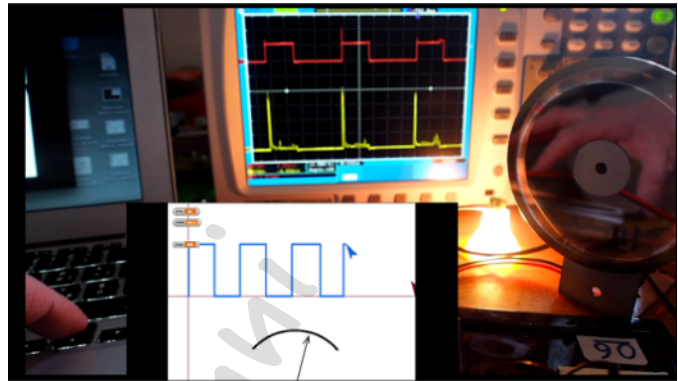
Lo stesso programma disegna, con la grafica di cui è dotato, il grafico della tensione sul carico e la posizione dell'indice di uno strumento analogico.



video: PWM

Composizione del quadro a video

- a sinistra: un Pc con la piattaforma “Snap for Arduino”, usata per pilotare la scheda Arduino e per realizzare l’animazione che illustra la tensione sul carico
- in centro: lo schermo dell’oscilloscopio e sotto, il disegno animato con i due strumenti che simulano la misura sul carico
- a destra la lampadina e il motore con un disco calettato (sul disco lucidato a specchio si vede riflessa la mano che manovra i tasti freccia sul PC)



Cosa si vede nel video

- 00:15** viene presentato lo schema elettrico del dispositivo con le due posizioni dell’interruttore (aperto, chiuso) ed il corrispondente stato della lampadina (spenta, accesa);
- 00:27** viene simulata la commutazione, prima lenta e poi rapida (malgrado sia tutto simulato, con la commutazione rapida si comincia ad intravedere una luminosità intermedia dovuta alla permanenza dell’immagine sulla retina, non credo al tempo di risposta dei pixel);
- 00:32** viene presentato lo schema elettrico dell’apparato di prova;
- 00:44** viene illustrato l’apparato come è disposto nel video:
- 01:06** ATTENZIONE, l’oscilloscopio misura la VT (tensione sull’interruttore MOSFET) mentre il grafico disegnato riproduce la tensione sul CARICO per cui le due onde quadre sono complementari. L’oscilloscopio mostra in rosso la traccia sul MOSFET della lampadina ed in giallo (con scala dimezzata) la tensione sul MOSFET del motore dove sono visibili le sovratensioni di apertura;
- 01:09** viene indicata la tensione sull’interruttore della lampadina (in rosso);
- 01:13** viene indicata la tensione sul carico come viene disegnata da “Snap for Arduino”;
- 01:23** inizia l’alimentazione del carico: il tempo di conduzione ton è nullo, la tensione è zero e la freccia dello strumento analogico simulato è tutta a sinistra; con la pressione del tasto “freccia su” viene allungata progressivamente la durata “on” dell’onda quadra e si osserva che la luminosità e la velocità aumentano (in realtà il motore si avvia solo con $d=25\%$): il duty cycle cresce (il suono iniziale che si sente è dovuto alle vibrazioni indotte da una tensione a frequenza 980Hz generata dal timer interno che Arduino usa sul pin 5 per generare il PWM, successivamente gli si sovrappongono i rumori della meccanica del motore);
- 01:41** il duty cycle è al 50%, ton è uguale a toff e la lancetta dello strumento analogico è a metà;
- 02:07** il funzionamento è a piena tensione, la luminosità e la velocità sono massime, il toff è nullo, la lancetta è tutta a destra;
- 02:09** con il tasto “freccia giù” si inizia la diminuzione del duty cycle, la luminosità diminuisce e la velocità decresce;
- 02:32** il duty cycle scende al 15% e si vede che è consentita una velocità minima che nella fase di accelerazione era impedita;
- 02:56** vengono presentati gli oscillogrammi a varie d% (riportati nelle figg. D).

video: PWM

Descrizione del programma

Il programma è composto da 4 sprites:

- a. **main**: presiede al coordinamento delle funzioni da svolgere
 1. all'avvio: vengono impostati i pin 3, 5, 6 di Arduino al funzionamento come uscite in PWM,
 2. vengono azzerate le variabili *PWM* e *duty*
 3. viene attivato il loop "per sempre" del calcolo di *duty* e *PWM* ed invio del valore *PWM* ai pin 3, 5, 6 contemporaneamente
 4. all'avvio: viene inviato il messaggio per il disegno degli assi cartesiani ed al termine viene inviato il messaggio per il disegno del grafico terminato il quale si ritorna all'inizio del ciclo "per sempre" e infine
 5. quando viene premuto uno dei "tasti freccia" viene assegnato il valore alla variabile *duty* con limitazione del valore minimo a zero e del valore massimo a cento; *duty* è una variabile globale ed il suo valore sarà utilizzato dallo sprite "scope".
- b. **assi**: quando riceve il messaggio disegna gli assi cartesiani e quindi le tacche sull'asse x; le tacche sono pensate per aiutare a vedere la percentuale del tempo di conduzione (ton) sul periodo di 10 tacche (sul piano cartesiano si possono vedere quattro periodi);

```
when I receive assi
hide
point in direction 180
clear
pen up
set pen size to 1
go to x: -240 y: 0
set pen color to red
pen down
go to x: 240 y: 0
pen up
go to x: -200 y: -180
set pen color to red
pen down
go to x: -200 y: 180
broadcast tacche and wait

when I receive tacche
pen up
set time to 0
repeat 8
  repeat 4
    change time by 10
    set pen size to 1
    pen up
    go to x: time - 200 y: -5
    pen down
    go to x: time - 200 y: 5
  repeat 4
    change time by 10
    set pen size to 3
    pen up
    go to x: time - 200 y: -5
    pen down
    go to x: time - 200 y: 5
```

```
MAIN
when clicked
  setup digital pin 5 as PWM
  setup digital pin 3 as PWM
  setup digital pin 6 as PWM
  set PWM to 0
  set duty to 0
  forever
    set PWM to duty * 2.5
    set PWM pin 5 to PWM
    set PWM pin 3 to PWM
    set PWM pin 6 to PWM

when clicked
  forever
    broadcast assi and wait
    broadcast scope and wait
    wait 1 secs

when up arrow key pressed
  change duty by 5
  if duty > 100
    set duty to 100

when down arrow key pressed
  change duty by -5
  if duty < 0
    set duty to 0
```

video: PWM

c. **scope**: quando riceve il messaggio ripete per 400 volte il calcolo per ogni punto lungo l'asse x per posizionare la penna, alto o basso, a seconda se si trova in fase on o in fase off;

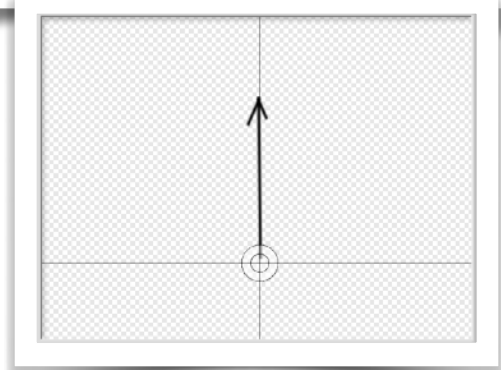
d. **analogic**: fa puntare lo sprite a forma di freccia ad angoli diversi a seconda del valore di *duty* per simulare l'indice di uno strumento analogico più adatto a visualizzare le variazioni della tensione. Lo sprite è una freccia con il centro posizionato all'estremità della coda.

A Scratch script titled "SCOPE" in green text. The script starts with a "when I receive" block set to "scope". It then sets "time" to 0, "pen size" to 3, and "pen color" to blue. It performs "pen up", "go to x: -200 y: 0", and "pen down". A "repeat" block with 400 iterations follows. Inside the repeat block, it "change x by 1". An "if" block checks if "time mod 100 < duty". If true, it "set y to 100"; otherwise, it "set y to 0". Finally, it "change time by 1".

```
when I receive scope
set time to 0
set pen size to 3
set pen color to blue
pen up
go to x: -200 y: 0
pen down
repeat 400
  change x by 1
  if time mod 100 < duty
    set y to 100
  else
    set y to 0
  change time by 1
```

A Scratch script titled "analogic". It starts with a "when clicked" block. It then "go to x: 8 y: -175", "set size to 55 %", and "point in direction 40". A "forever" loop contains a "point in direction" block with "40 + duty".

```
when clicked
go to x: 8 y: -175
set size to 55 %
point in direction 40
forever
  point in direction 40 + duty
```



Il video qui illustrato è reperibile a <https://youtu.be/wtvsJEHEspQ>

e fa parte dell'articolo: <http://www.mastropaolo.net/laboratorio/misure-di-pwm-con-arduino>

Il programma per Snap for Arduino utilizzato per il video e questa scheda è qui:

<https://drive.google.com/file/d/0B2pMljvhp1Gem4tOXZpczJrVTA/view?usp=sharing>

E' disponibile on line un versione del programma migliorata rispetto a quella considerata nel video che è utilizzabile anche senza scheda Arduino collegata e quindi più adatta per effettuare simulazioni in aula:

<https://drive.google.com/file/d/0B2pMljvhp1GaEFzQm5zSW1MYzQ/view?usp=sharing>

video: PWM

Schemi, fotografie, software

fig. A: schema della scheda RGB shield di Futura Elettronica

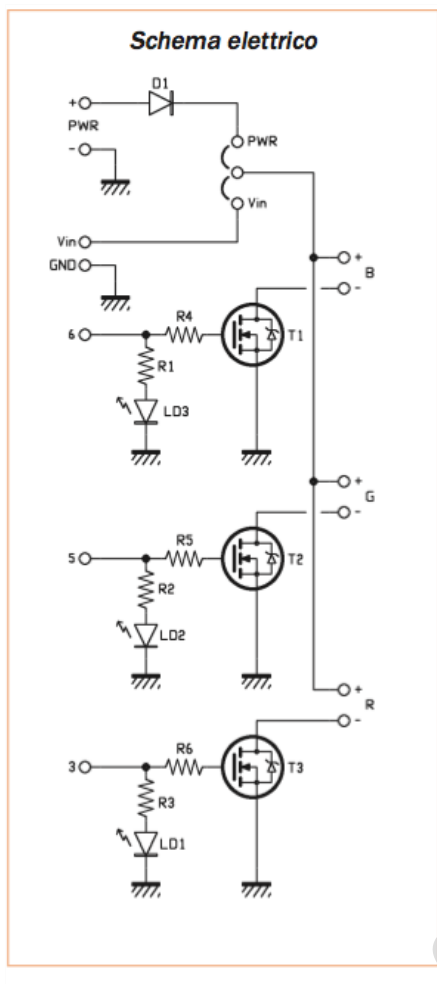


fig. B: la scheda Arduino con sopra montata la scheda RGB, la lampadina, il motoriduttore e le due sonde osciloscopiche.

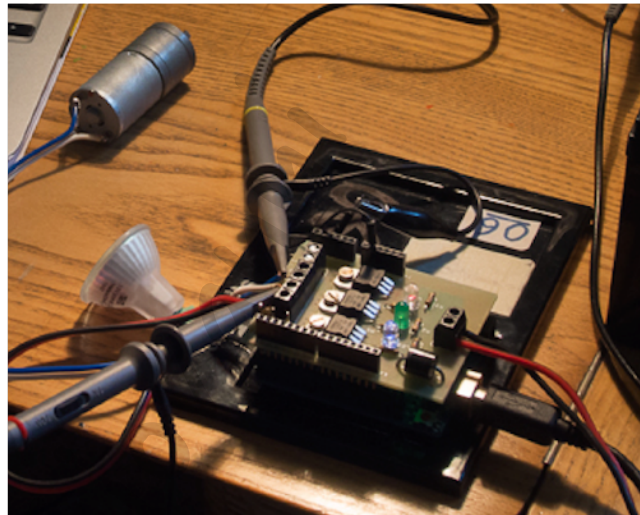
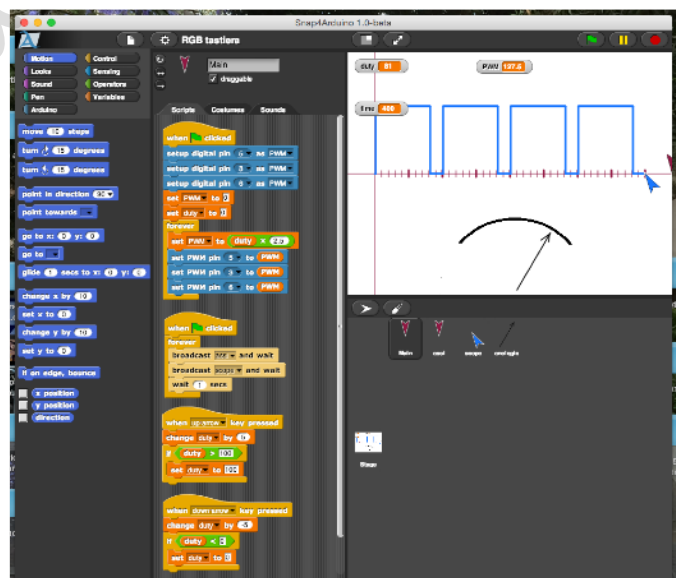


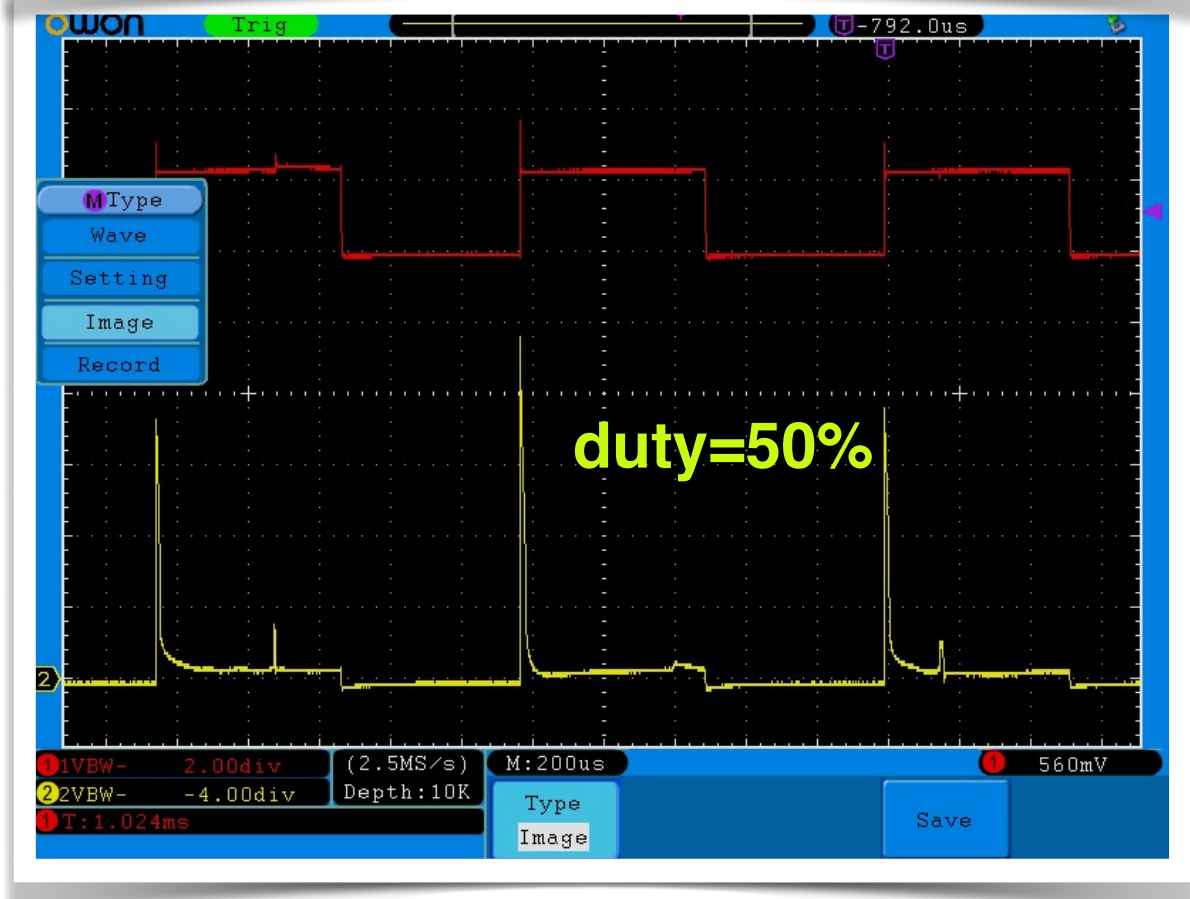
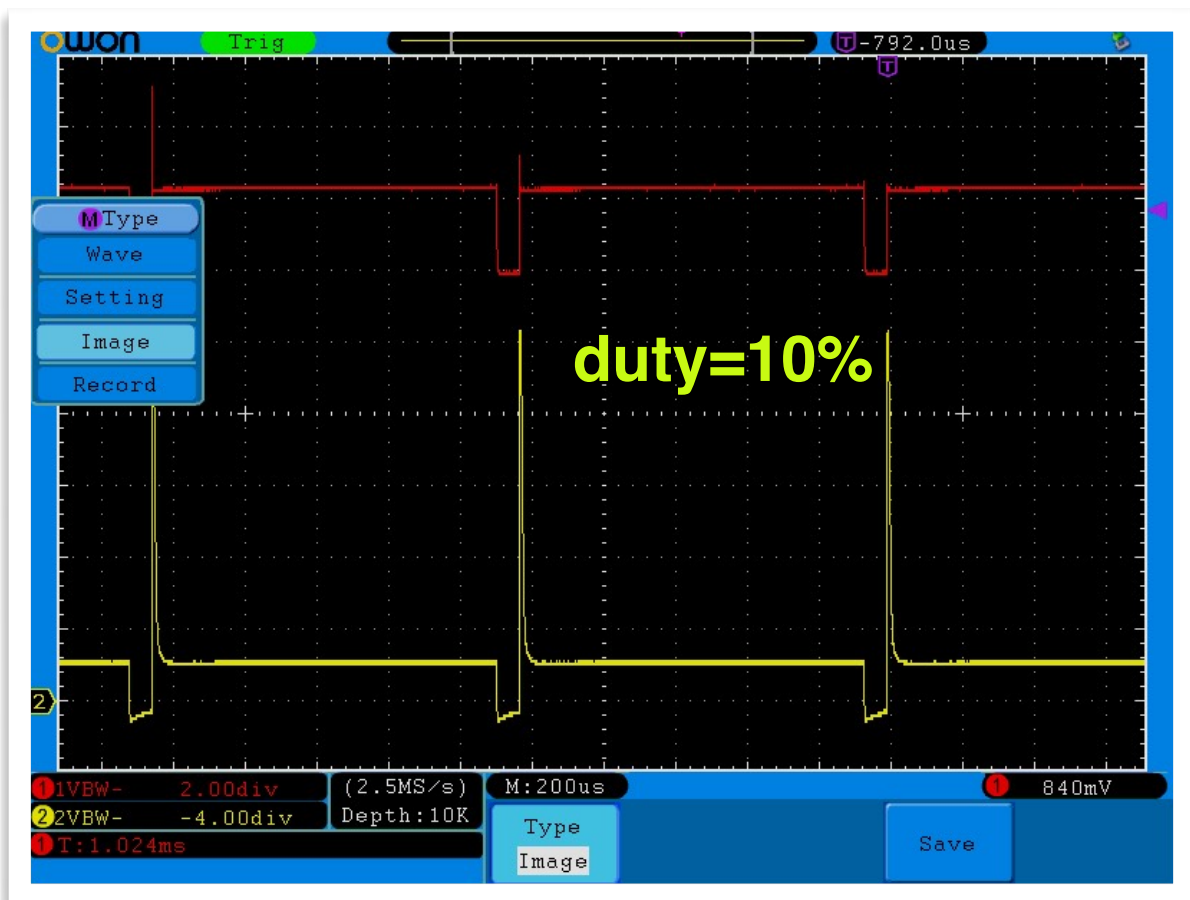
Fig. C: La piattaforma "Snap for Arduino" pronta per l'esercitazione



figg. D: Schermate dell'oscilloscopio settembre 2015

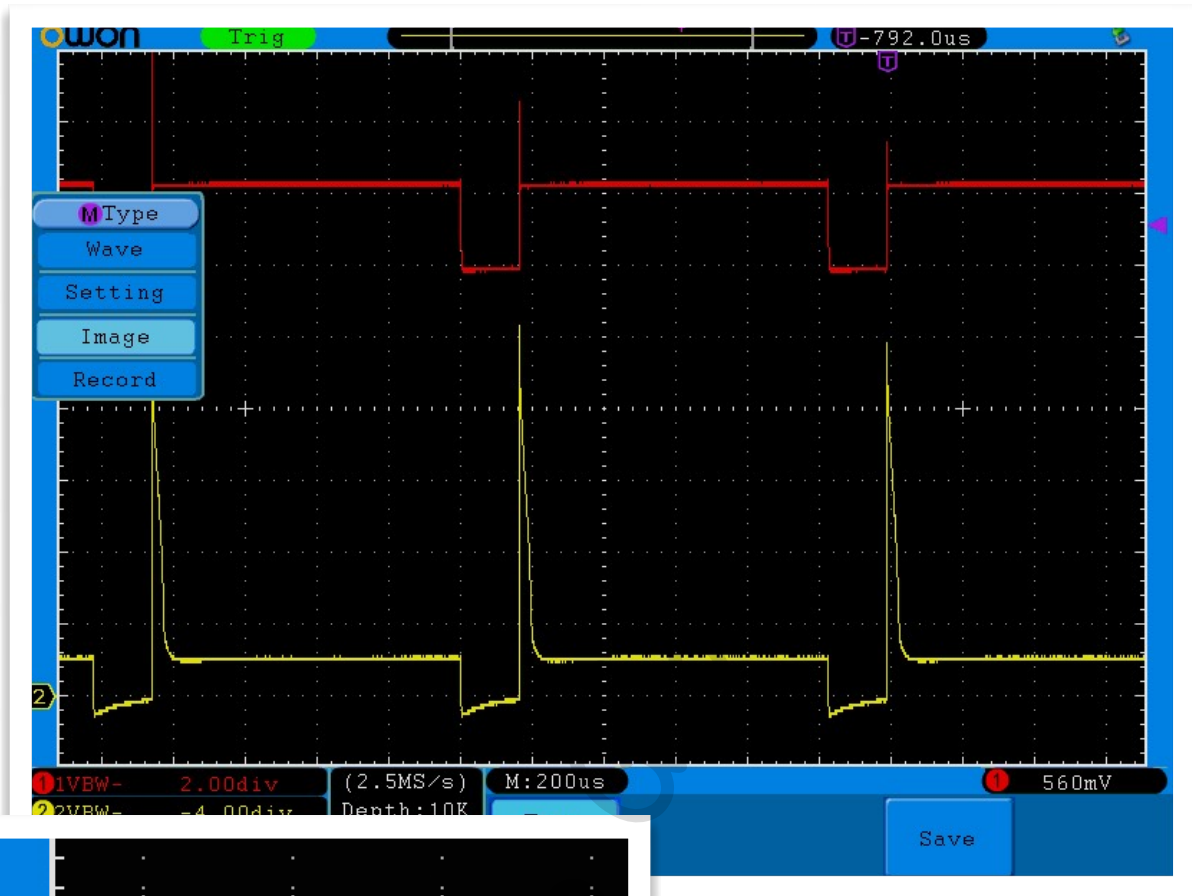
video: PWM

Vene mostrato il quadro dell'oscilloscopio in altra situazione con lo scopo di approfondirne lo



video: PWM

studio:



① 1VBW- 2.00div
② 2VBW- -4.00div
① T:1.024ms

I valori delle scale di misura sono a fianco (occorre sapere che la sonda era impostata su 10x per entrambi i canali e ricordare che si misura la tensione ai capi del MOSFET)

sul grafico in rosso:

- 1VBW- significa che la scala delle tensioni è 1 volt a divisione (per divisioni si intendono i quadretti di circa un cm di lato) con accoppiamento dell'ingresso in cc
- 2.00div significa che lo zero volt è posizionato a 2divisioni (al secondo quadretto sopra la linea di mezzo)
- T:1.024ms significa che il periodo è 1,024 ms e quindi frequenza 977Hz
- M:200 μ s significa che la scala dei tempi è 200 μ s per divisione

dall'esame del grafico in rosso si misurano quindi:

- valore minimo 0div e valore massimo 1,2div (trascurando lo spike brevissimo di sovratensione) che corrispondono a 0V e 12V dato che la sonda è posizionata su 10x
- periodo 5,1div che corrispondono a $5,1 \cdot 200\mu\text{s} = 1020\mu\text{s}$ ovvero 1,020ms (la differenza è solo del 4 per mille rispetto a quanto indicato dall'oscilloscopio, è trascurabile e la causa va imputata agli inevitabili errori di apprezzamento)
- durata della conduzione 0,8div che significa $\text{duty cycle} = 0,8/5,1 = 0,16$ ovvero $d\% = 16\%$

Sul grafico in giallo si osservano una sovratensione all'apertura di T che raggiunge 5div ovvero 100V ed una tensione negativa alla chiusura. Ricordo che si tratta di un motore DC.