

# PWM per PLC S7

Il generatore di impulsi interno al PLC gestisce autonomamente l'emissione di treni di impulsi simmetrici (funzione PTO, *Pulse Train Output*, duty cycle = 50%) o modulabili in durata (funzione PWM, *Pulse Width Modulation*, duty cycle variabile, **fig. 1**).



**Fig. 1.** Segnale PWM.

Nel PLC S7-200 il tempo di ciclo e la durata degli impulsi sono predefiniti in microsecondi o millisecondi, in word da 16 bit (valori fino a 65.535).  
 Come indicato nella **tab. 1**, impostando la durata dell'impulso uguale al tempo di ciclo (d. c. = 100%), l'uscita viene attivata continuamente, mentre se si imposta la durata dell'impulso a zero (d. c. = 0%), l'uscita rimane sempre disattivata.

Tab. 1 – Relazioni particolari tra durata degli impulsi e tempo di ciclo	
Durata degli impulsi / tempo di ciclo	Reazione
durata impulso $\geq$ tempo di ciclo	duty cycle = 100% (uscita sempre attiva)
durata impulso = 0	duty cycle = 0% (uscita sempre spenta)
tempo di ciclo < 2	tempo di ciclo = 2

### Assistente PWM per STEP 7-Micro/Win

Le funzioni PWM disponibili nel PLC S7-200, CPU 224XP, sono due.

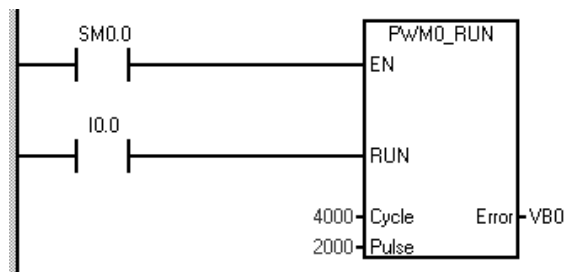
L'esempio proposto (**fig. 2**) utilizza lo strumento *Assistente di controllo posizionamento* per attivare la funzione PWM sull'uscita Q0.0.

Dalla barra *Strumenti* selezionare *Assistente di controllo posizionamento* e spuntare *Configura PTO/PWM* su Q 0.0, uscita per PWM, con base tempi *microsecondi*.

In questo modo, il sistema aggiunge al progetto il sottoprogramma PWM0\_RUN, da tenere abilitato

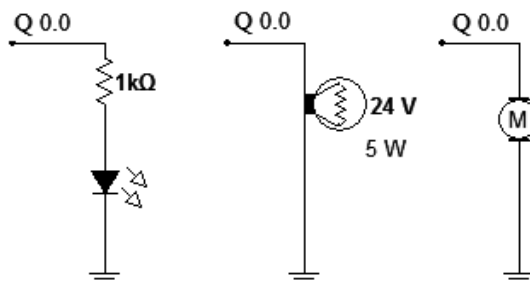
mentre è attiva l'uscita PWM, e rende disponibile il box PWM0\_RUN tra i componenti.

Nello schema di **fig. 1**, il sottoprogramma è sempre abilitato e l'emissione del segnale PWM è controllata dall'ingresso I 0.0



**Fig. 2.** Box PWM0\_RUN.

Il funzionamento dell'uscita PWM a 24 V su Q 0.0 può essere osservato applicando come carico un LED con 1 k $\Omega$  in serie (**fig. 3**), oppure una lampada da 5 W (massimi) a 24 V, o ancora un motorino c. c. con resistenza di armatura superiore a 32  $\Omega$  (corrente di spunto < 0,75 A).



**Fig. 3.** Carichi diretti per uscita PWM.

Modificare la durata di Pulse in modo da osservare la diversa quantità di energia trasferita al carico. Portare Cycle = 40.000 e osservare il comportamento del carico.

### Variazione manuale del duty cycle

Si supponga di voler variare manualmente e con continuità, mediante il potenziometro presente sulla CPU, la velocità di un motorino in continua a magneti permanenti, utilizzando un'uscita con segnale PWM.

Considerando l'inerzia del motore, una frequenza di 250 Hz può essere sufficiente ( $T = 4000 \mu\text{s}$ ).

Poiché la posizione di rotazione del potenziometro 0, presente sulla CPU, è contenuta nella variabile SMB28 come valore ad 8 bit compreso tra 0 e 255, si tratta di portare questa quantità nel range intero tra 0 e 4000, per utilizzarla come valore del duty cycle della forma d'onda. Il fattore di moltiplicazione è quindi:

$$\frac{4000}{255} = 15,686$$

Con questo fattore, in corrispondenza del valore massimo 255, si ottiene difatti:

$$255 \cdot 15,686 = 3999,93, \text{ arrotondabile a } 4000$$

Il programma in AWL riportato in **fig. 4** utilizza l'operatore PLS 0 per attivare la funzione PWM sull'uscita Q0.0.

```

Segmento 1 Pot. 0 -> MW0 = 0 - 4.000
|-----|
|trasforma 8 bit di Pot. 0 (0 - 255) in intero 16 bit 0 - 4.000|
|-----|
LD      SM0.0          // ad ogni ciclo
MOVW   MW0, MW2       // salva valore precedente
MOVB   0, MB0
MOVB   SMB28, MB1     // MW0= MB0(=0)+MB1(=Pot.0)
MOVR   15.686, VD4    // costante moltiplicativa
ITD    MW0, AC1       // intero da 16 a 32 bit
DTR    AC1, VD0       // da int. 32 a reale in VD0
*R     VD4, VD0       // moltiplica tra num. reali
ROUND  VD0, VD8       // da num. reale a intero 32 bit
MOVW   VW10, MW0      // 16 bit meno sign.-->MW0(=0-4000)
MOVW   4000, MW4      // periodo

Segmento 2 se diverso dal precedente aggiorna PTO
|-----|
|eventuale aggiornamento del d.c. del PWM|
|-----|
LDW=   MW0, MW2
NOT
MOVB   16#D3, SMB67   // se il val.calc. è div. dal prec.
MOVW   MW4, SMW68    // abilita aggiornamento
MOVW   MW4, SMW68    // periodo
MOVW   VW10, SMW70   // d.c.
PLS    0              // attiva nuovi valori

```

**Fig. 4.** Listato AWL per generazione PWM variabile.

Poiché il moltiplicatore è con la virgola, si tratta di lavorare con i numeri reali, trasformando l'intero a 8 bit (MB1 = SMB28) prima a 16 bit (MW0) e poi a 32 bit (ITD, *Int To Double*, in AC1) e da questo nel numero reale corrispondente (DTR, *Double To Real*, in VD0). Una volta eseguita la moltiplicazione tra i due valori reali (\*R), il risultato (VD0) va arrotondato all'intero a 32 bit più prossimo (ROUND, in VD8) e da questo si estraggono i soli 16 bit significativi (VW10 ≤ 4000) da assegnare poi al duty cycle.

L'aggiornamento dei valori del periodo (SMW68 = 4000) e del d. c. (SMW70 da VW10) è ottenuto mediante il comando D3<sub>H</sub> a 8 bit scritto nel byte di controllo SMB67, con ciascun bit che assume un significato specifico:

- D3<sub>H</sub> = 1101 0011;
- b<sub>7</sub> = 1 → abilita blocco PTO/PWM;
- b<sub>6</sub> = 1 → seleziona funzione PWM;
- b<sub>4</sub> = 1 → aggiornamenti sincroni con la fine dell'impulso precedente;
- b<sub>3</sub> = 0 → unità espresse in μs;
- b<sub>1</sub> = 1 → segue nuovo valore di t<sub>on</sub>;
- b<sub>0</sub> = 1 → segue nuovo valore del periodo.

L'istruzione PLS 0 attiva finalmente la nuova configurazione sull'uscita Q0.0.