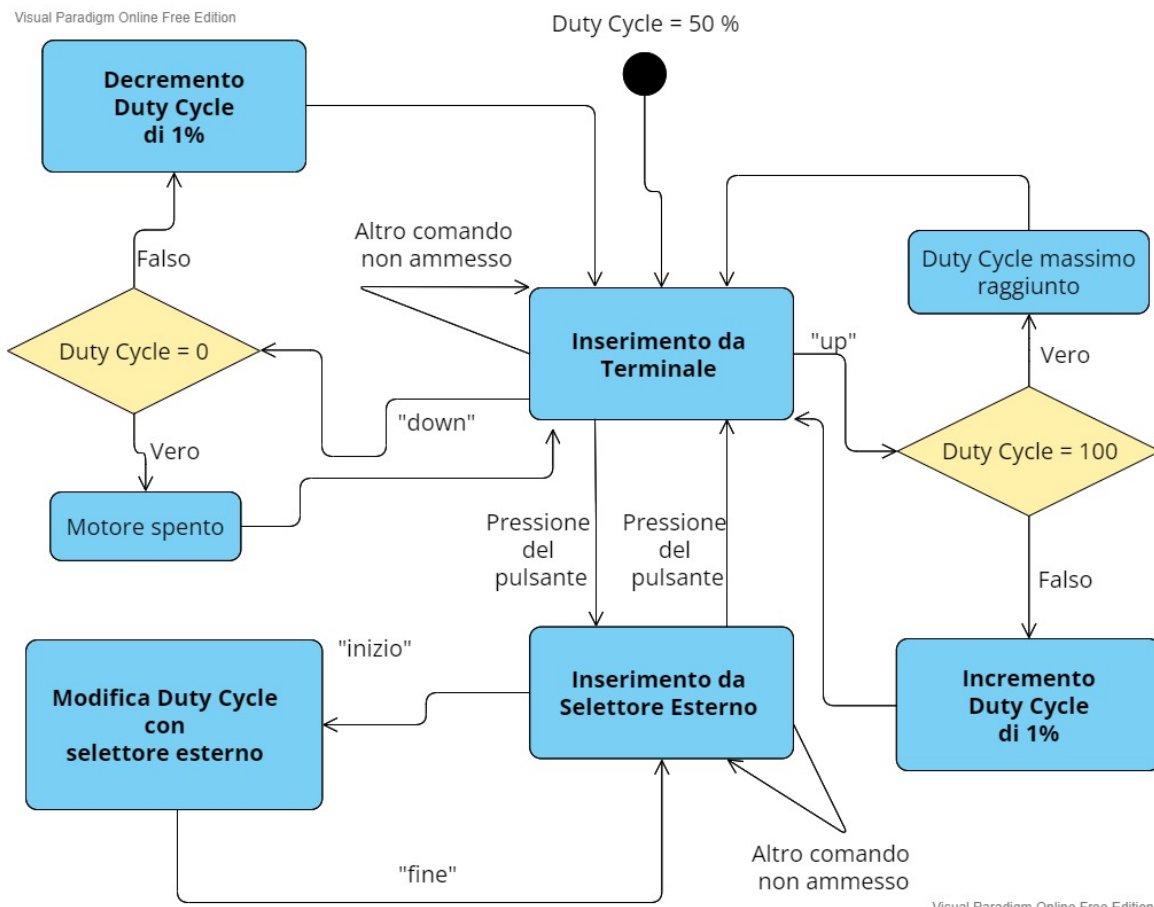


# Complementi di Elettronica Digitale e Microprocessori - Progetto d'seame 2020/2021

Frenki Shqepa  
Università degli Studi di Brescia

December 13, 2021



## Scelta del timer

Il microprocessore utilizzato, ATMEGA-328P, comprende tre timers/counters, TC0, TC1, TC2, rispettivamente da 8, 16 e 8 bit. Questo significa che con TC0 e TC2 si possono rappresentare  $2^8 = 256$  valori diversi, mentre con TC1 posso avere  $2^{16} = 65536$  numeri. Tra le specifiche c'è la richiesta di avere una variazione di 1% tra una velocità e l'altra. Vengono richiesti 100 livelli di velocità, a cui aggiungiamo anche quello in cui il Duty Cycle è uguale a '0'. In quest'ultimo caso non utilizziamo un Duty Cycle di 0%, ma spegniamo del tutto il timer. Utilizziamo il Timer Counter TC0 e sfruttiamo tutti i possibili 255 valori da 0 a 255. Per rappresentare 100 valori di Duty Cycle, nel codice si utilizza una variabile per memorizzare il valore percentuale attuale del Duty Cycle, per poi mapparlo in un range tra 0 e 255 attraverso la seguente proporzione:

$$ValorePercentualeAttuale : 100 = x : UserTop \quad (1)$$

dove:

- 'x' è il valore del Duty Cycle, mappato in un numero tra 0 e 255;
- UserTop è uguale a 255 ed è il numero fino a cui deve contare il Timer Counter TC0.

Essendo il risultato ottenuto non intero, si utilizza la funzione ceil come segue:

$$OCR0B = \text{ceil}(\text{valoreDC} \cdot \text{top} / 100);$$

Per pilotare il motore viene sfruttato il Circuito Integrato L293D, che permette di pilotare fino a due motori in due direzioni. Questo accetta due alimentazioni, una per alimentare se stesso, l'altra per alimentare il motore. Uno degli inputs del driver è il segnale di Enable (ENx), in cui si fa entrare il segnale PWM. Questo attiva il 3 State Buffer (vedere schema circuitale L293D), che permette quindi di far passare il segnale IN in OUT, che determinano il verso di rotazione del motore.

Il driver L293D accetta segnali di EN con frequenza massima di 5KHz, quindi bisogna calcolare la frequenza PWM adatta al nostro scopo. Come già detto, il valore di TOP è 255. Si usa la formula

$$F_{PWM} = \frac{F_{CPU}}{TOP \cdot PRE} - 1 \quad (2)$$

che, al variare di PRE, mi permette di ottenere la Fpwm. Stiamo usando la modalità Fast PWM, in modalità non invertente. Questo significa che il valore di TOP viene memorizzato nel registro OCR0A e c'è il "clear" di OCR0B al momento del "Compare Match".

L'obiettivo del progetto è il pilotaggio di un motore. E' stato utilizzato in continua da 12V. Bisogna scegliere quindi la giusta frequenza di pwm per il pilotaggio di questo motore.

PRE	$F_{PWM}$ [Hz]
1	62744
8	7842
64	979
256	244
1024	60

Table 1: Valori di  $F_{PWM}$  al variare del valore di PRE

Il driver L293D è adatto per l'uso in applicazioni dove ho commutazione a frequenze minori di 5kHz. Significa che non posso utilizzare i fattori di prescaler uguali a 8 e 1. Utilizzare un fattore di prescaler basso permette di avere una risoluzione temporale migliore del segnale ricostruito, quindi, secondo questo ragionamento, bisognerebbe scegliere il fattore di prescaler 64. E' consigliato utilizzare motori DC con frequenze dai 50Hz ai 100Hz, facendo quindi virare la scelta del fattore di prescaler a 1024.

## Trasmissione dati via terminale RS232

Quando si implementa una comunicazione seriale, sono essenziali i seguenti aspetti:

- Il trasmettitore ed il ricevitore operano allo stesso Baud Rate;
- Il trasmettitore ed il ricevitore sono a conoscenza della lunghezza del pacchetto che viene inviato e successivamente ricevuto;
- E' noto lo stato quiescente della linea;
- Il trasmettitore deve mettere la linea dati in stato di quiescenza per almeno un tempo di bit tra due frame consecutivi. Questo prende il nome di Stop Bit.

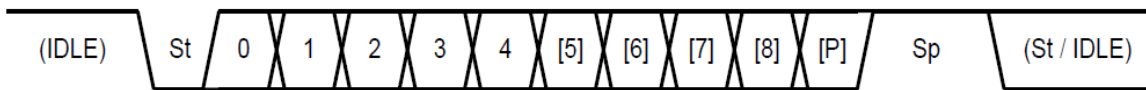


Figure 1: Costruzione di un pacchetto per la periferica USART.

Per implementare queste funzionalità si utilizza la periferica USART disponibile nel controllore XplainedMini. Per scegliere il Baud Rate (bps), consideriamo l'applicazione della periferica nel sistema sotto esame. Il solo ruolo della periferica è la trasmissione di messaggi di utilità all'utente, quindi si può capire che non è essenziale la massima velocità di trasmissione. Ci si ferma allora allo standard Baud Rate di 9600 bps. Nel pacchetto si possono inserire bit di parità, utilizzati per la rilevazione di errori ma, sempre per una

questione di non estrema importanza nella applicazione in esame, tralasciamo questo tipo di bit. Si utilizza un solo bit di stop e si inviano pacchetti con dati da un byte, quindi otto bit.

## **Pattern di sviluppo**

Si sceglie la programmazione di una macchina a stati. Al primo avvio ci si trova in uno stato dove il Duty Cycle è al 50%, in particolare nello stato " TerminaleAttivo". Se si scrive uno tra i comandi "up" e "down", si entra nello stato "ModificaDCTerminale", in cui avviene la effettiva modifica del Duty Cycle a seconda del comando inserito. Se si schiaccia il pulsante sulla scheda XplainedMini, si entra nello stato "SelettoreEsternoAttivo", dove si aspetta che l'utente inserisca "inizio" a terminale, in modo da poter iniziare la modifica del Duty Cycle attraverso il selettore esterno. Quando l'utente invia il comando "fine", si passa allo stato "ModificaDCSelettore", dove il Duty Cycle viene modificato. Si è scelto di effettuare la modifica tramite selettore esterno dopo apposito comando inviato a terminale, per dare tempo all'utente di inserire il valore desiderato, evitando quindi che nel frattempo il Duty Cycle si modifichi a valori non desiderati. Solo quando l'utente è soddisfatto, può consentire la modifica del Duty Cycle.

## **Inserimento da Selettore Esterno**

Per scegliere il Duty Cycle dal selettore esterno, è stato scelto di utilizzare gli interrupt pin change. Da specifiche, sono richieste cifre in codifica BCD, quindi c'è bisogno di nove linee. Ognuna di queste linee è collegata ad un Pin di tipo I/O. Si può scegliere tra 23 pin per la generazione dell'interrupt, nel caso ci fosse un cambio dello stato della linea. Bisogna però fare attenzione a non utilizzare pin utilizzati per altre funzionalità. Per memorizzare questi valori BCD, si usano tre vettori, dichiarati volatile, in cui gli elementi vengono modificati dalle ISR a seconda dello stato della linea. Si è usata una configurazione attiva alta, quindi per avere il bit a '1', bisogna chiudere lo switch, o collegare il jumper ad alimentazione. Quando l'utente decide che ha finito di inserire il valore, di questi vettori viene calcolato il corrispondente decimale, con la funzione BinToDec(), per poi essere messe come argomenti della funzione SwitchConcat(), che unisce le tre cifre per formare un numero da '0' a '100'.