

Ghid de Formule și Explicații pentru Microcontrolere AVR

Acest document sumarizează relațiile matematice fundamentale utilizate în configurarea și analiza perifericelor pentru microcontrolerele din familia AVR (ATmega16, ATmega1280).

1. Temporizatoare (Timers / Counters)

A. Calculul duratei unui interval măsurat: Se utilizează pentru a determina timpul scurs între două valori citite din registrul de numărare ($TCNT$).

$$T_{măsurat} = \frac{(Val_{final} - Val_{inițial}) \cdot Prescaler}{f_{osc}} \quad (1)$$

Explicație: Un timer incrementează registrul $TCNT$ la fiecare interval de timp egal cu $\frac{Prescaler}{f_{osc}}$. Înmulțind numărul de pași făcuți cu durata unui pas, obținem timpul total.

B. Timpul până la întreruperea de depășire (Overflow): Se utilizează pentru a calcula perioada la care un timer generează o întrerupere automată.

$$T_{ovf} = \frac{2^n \cdot Prescaler}{f_{osc}} \quad (2)$$

Explicație: Un timer pe n biți se resetează după 2^n pași (256 pentru 8 biți, 65536 pentru 16 biți). Această formulă este esențială pentru a crea baze de timp precise (ex: o secundă).

2. Modularea în lățime a impulsului (PWM)

A. Durata impulsului activ (High) în modul Fast PWM: Se utilizează pentru a calcula cât timp semnalul de ieșire este la nivel logic "1" într-o perioadă.

$$t_{high} = \frac{(OCRnx + 1) \cdot Prescaler}{f_{osc}} \quad (3)$$

Explicație: În modul Fast PWM (ne-inversat), pinul de ieșire trece în 0 când $TCNT$ atinge valoarea registrului de comparație ($OCRnx$). Adunăm 1 deoarece numărarea începe de la 0.

3. Comunicația Serială (USART)

A. Configurarea vitezei de transmisie (Baud Rate): Se utilizează pentru a determina valoarea ce trebuie scrisă în registrul $UBRR$ pentru a obține o viteză de comunicație standard.

$$UBRR = \frac{f_{osc}}{16 \cdot \text{Baud}} - 1 \quad (\text{Mod Normal}) \quad (4)$$

$$UBRR = \frac{f_{osc}}{8 \cdot \text{Baud}} - 1 \quad (\text{Mod Double Speed, U2X=1}) \quad (5)$$

Explicație: Divizorul hardware împarte frecvența cristalului pentru a genera "ceasul" de bit. Scădem 1 deoarece registrul acceptă valori de la 0 la $n - 1$.

B. Timpul de transmisie a datelor: Se utilizează pentru a calcula latența comunicației.

$$T_{transmisie} = \frac{N_{octeți} \cdot \text{Biți/Cadru}}{\text{Baud Rate}} \quad (6)$$

Explicație: Un cadru serial conține de regulă 10 biți (1 start + 8 date + 1 stop). Împărțind numărul total de biți la viteza de transmisie (biți/secundă), obținem durata transferului în secunde.

4. Arhitectură și Execuție CPU

A. Frecvența reală de lucru: Se utilizează pentru calibrarea oscilatorului sau verificarea duratei de execuție a codului.

$$f_{real} = \frac{\text{Nr. Cicluri Ceas}}{T_{execu\c{t}ie}} \quad (7)$$

Explicație: Instrucțiunile precum `_delay_cycles(N)` consumă exact N perioade de ceas. Dacă știm timpul real de execuție (măsurat), putem deduce frecvența la care rulează de fapt cipul.

B. Lățimea magistralei de adrese: Se utilizează pentru a determina câți biți sunt necesari pentru a adresa o anumită capacitate de memorie.

$$N_{bi\c{t}i} = \log_2(\text{Nr. Locații de Memorie}) \quad (8)$$

Explicație: De exemplu, ATmega1280 are 128 KB Flash. Deoarece codul este organizat în cuvinte de 16 biți (2 octeți), avem 64k locații. $\log_2(65536) = 16$ biți necesari pentru magistrala de adrese a codului.