

Biwak - Fehler #743

I2c TIming variable not calculated

18.03.2026 17:46 - Maximilian Seesslen

Status:	Neu	Beginn:	18.03.2026
Priorität:	Normal	Abgabedatum:	
Zugewiesen an:		% erledigt:	0%
Kategorie:		Geschätzter Aufwand:	0.00 Stunde
Zielversion:		Aufgewendete Zeit:	0.00 Stunde
CS Zielversion:			

Beschreibung

```
consteval uint32_t i2cTiming(
    uint32_t fPCLK,
    uint32_t fI2C,
    uint32_t riseTime,
    uint32_t fallTime,
    uint8_t dnf,
    uint8_t analogFilt
){
    // 1. Grundtakte
    double tI2C = 1.0 / fI2C;          // Zielperiode
    double tPCLK = 1.0 / fPCLK;        // APB1-Taktperiode

    // 2. Filterzeiten
    double tFilter = dnf * tPCLK;
    if (analogFilt)
        tFilter += 50e-9;              // 50 ns analog filter

    // 3. High/Low Zeiten (vereinfachte AN4235-Formeln)
    double tLow  = (tI2C / 2.0) - riseTime*1e-9 - tFilter;
    double tHigh = (tI2C / 2.0) - fallTime*1e-9 - tFilter;

    // 4. PRESC suchen (kleinster gültiger Wert)
    uint8_t presc = 0;
    double tPresc = tPCLK;

    while (tLow < tPresc || tHigh < tPresc) {
        presc++;
        tPresc = (presc + 1) * tPCLK;
        if (presc > 15) break;
    }

    // 5. SCLL / SCLH berechnen
    uint8_t scll = (uint8_t)(tLow / tPresc) - 1;
    uint8_t sclh = (uint8_t)(tHigh / tPresc) - 1;

    if (scll > 255) scll = 255;
    if (sclh > 255) sclh = 255;

    // 6. SDADEL / SCLDEL (vereinfachte sichere Werte)
    uint8_t sdadel = (uint8_t)((fallTime*1e-9 + tFilter) / tPresc);
    uint8_t scldel = (uint8_t)((riseTime*1e-9 + tFilter) / tPresc);

    if (sdadel > 15) sdadel = 15;
    if (scldel > 15) scldel = 15;

    // 7. TIMINGR zusammensetzen
    uint32_t timing =
        ((uint32_t)presc << 28) |
        ((uint32_t)scldel << 20) |
        ((uint32_t)sdadel << 16) |
```

```
((uint32_t)sclh << 8 ) |  
((uint32_t)scll);  
  
return timing;  
}
```

The variable can be calculated with an "consteval" function.