

Vergleichsmessung TIA-Portal Programmiersprachen

Welcher der folgenden Programmcode wird in diesem Beispiel am schnellsten bearbeitet?

- > Programm 1 (E0.1) : AWL mit Trickprogrammierung
- > Programm 2 (E0.2) : AWL mit Dreisatz
- > Programm 3 (E0.3) : SCL mit Dreisatz
- > Programm 4 (E0.4) : FUP mit Dreisatz
- > Programm 5 (E0.5) : SCL mit Siemens Standards
- > Programm 6 (E0.6) : FUP mit Siemens Standards
- > Programm 7 (E0.7) : CEM

Programmfunktion:

- > 8 digitale Ausgänge (A0.0 bis A0.7) werden je nach gemessener Spannung 0..10V (EW10) angesteuert.
 - A0.0 \geq 10V
 - A0.1 \geq 8.75V
 - A0.2 \geq 7.5V
 - A0.3 \geq 6.25V
 - A0.4 \geq 5 V
 - A0.5 \geq 3.75V
 - A0.6 \geq 2.5V
 - A0.7 \geq 1.25V
- > Die 7 unterschiedlichen Programme können über die Eingänge E0.1 bis E0.7 bearbeitet werden.
- > Die Bearbeitungszeit wird mit dem Befehl „**RUNTIME**“ im OB1 gemessen.
- > Die Messresultate werden im „DB_Zyklusmessung“ (DB100) gespeichert.

Messaufbau:

- > S7-1500 CPU1513F-1 PN (Firmware V2.9.4)
- > TIA-Portal V17 Update 2
- > Alle Bausteine sind **optimiert** programmiert
- > Zykluszeitüberwachung von 150ms auf 6000ms erhöht
- > F-Fähigkeit ausgeschaltet
- > Während der Messung sind keine Beobachtungsfunktionen aktiv
- > Bausteine werden 30000-mal pro Zyklus bearbeitet
- > Datum der Messungen: 7.2.2022

Programm 1: AWL mit Trickprogrammierung

| Netzwerk 1: Spannungsanzeige 0..10V | | | |
|-------------------------------------|------|-----------------|--|
| 1 | L | "AE_Slider1" | |
| 2 | L | 0 | |
| 3 | <I | | // Prüfen ob der Analogwert negativ ist |
| 4 | SPB | Neg | |
| 5 | TAK | | // Akku 2 wieder im Akkul |
| 6 | SLD | 3 | // alter S5 Trick -> mal 8 multiplizieren |
| 7 | L | 27648 | // Anzahl Digits bei 10V |
| 8 | /D | | // 0..10V -> 0..27648 -> 0..8 (Analogwert * 8 / 27648) |
| 9 | L | DW#16#FFFF_FF00 | |
| 10 | SRD | | // Wert um der geschoben wird steht im Akku2 (0..8) |
| 11 | Neg: | T | "AB_Meldeleuchten0_7" |

Programm 2: AWL mit Dreisatz

| Netzwerk 1: Analogwert einlesen und normieren (0..10V -> 0..27648 -> 0..8) | | |
|--|-----|-------------------|
| 1 | L | "AE_Slider1" |
| 2 | L | 8 |
| 3 | *D | |
| 4 | L | 27648 |
| 5 | /D | |
| 6 | T | #1_iTemp |
| Netzwerk 2: Ausgänge ansteuern | | |
| 1 | L | #1_iTemp |
| 2 | L | 8 |
| 3 | >=I | |
| 4 | = | "A_Meldeleuchte0" |
| 5 | | |
| 6 | L | #1_iTemp |
| 7 | L | 7 |
| 8 | >=I | |
| 9 | = | "A_Meldeleuchte1" |
| 10 | | |
| 11 | L | #1_iTemp |
| 12 | L | 6 |
| 13 | >=I | |
| 14 | = | "A_Meldeleuchte2" |
| 15 | | |
| 16 | L | #1_iTemp |
| 17 | L | 5 |
| 18 | >=I | |
| 19 | = | "A_Meldeleuchte3" |
| 20 | | |
| 21 | L | #1_iTemp |
| 22 | L | 4 |
| 23 | >=I | |
| 24 | = | "A_Meldeleuchte4" |
| 25 | | |
| 26 | L | #1_iTemp |
| 27 | L | 3 |
| 28 | >=I | |
| 29 | = | "A_Meldeleuchte5" |
| 30 | | |
| 31 | L | #1_iTemp |
| 32 | L | 2 |
| 33 | >=I | |
| 34 | = | "A_Meldeleuchte6" |
| 35 | | |
| 36 | L | #1_iTemp |
| 37 | L | 1 |
| 38 | >=I | |
| 39 | = | "A_Meldeleuchte7" |

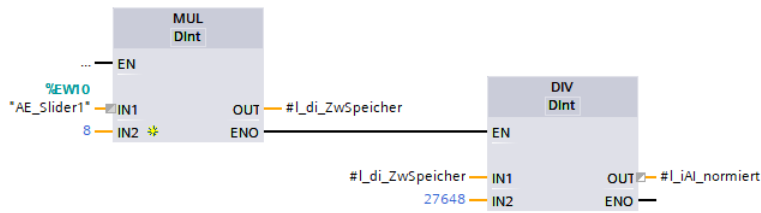
Der Befehl „TAK“ wurde bewusst nicht verwendet.

Programm 3: SCL mit Dreisatz

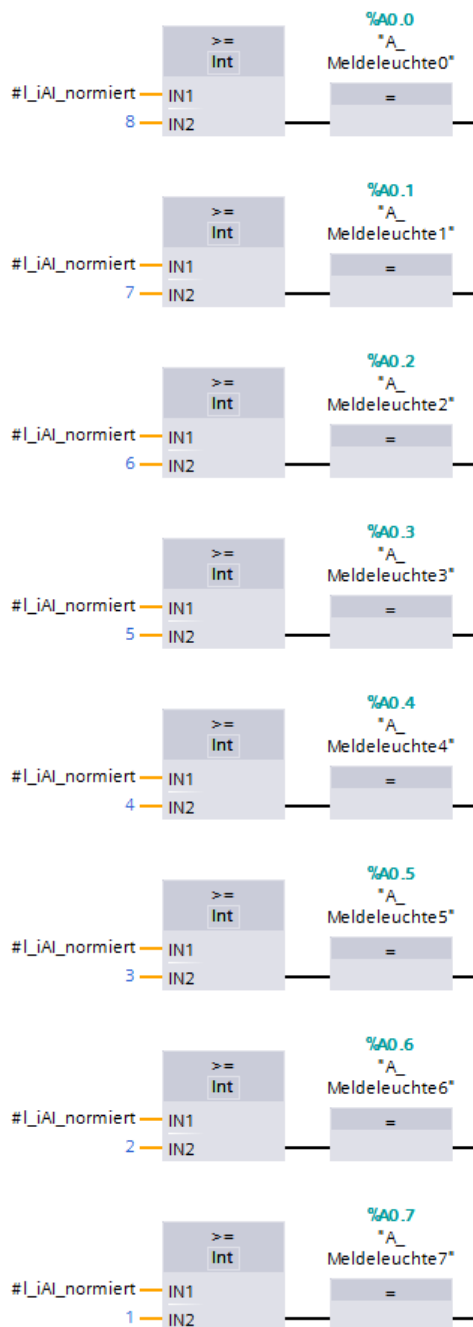
```
1
2 // Analogwert einlesen und normieren (0..10V -> 0..27648 -> 0..8)
3 #l_iTemp := DINT_TO_INT("AE_Slider1" * DINT#8 / 27648);
4
5 // Ausgänge ansteuern
6 "A_Meldeleuchte0" := #l_iTemp >= 8;
7 "A_Meldeleuchte1" := #l_iTemp >= 7;
8 "A_Meldeleuchte2" := #l_iTemp >= 6;
9 "A_Meldeleuchte3" := #l_iTemp >= 5;
10 "A_Meldeleuchte4" := #l_iTemp >= 4;
11 "A_Meldeleuchte5" := #l_iTemp >= 3;
12 "A_Meldeleuchte6" := #l_iTemp >= 2;
13 "A_Meldeleuchte7" := #l_iTemp >= 1;
```

Programm 4: FUP mit Dreisatz

Netzwerk 1: Analogeingang 0..10V (0..27648) auf den Wert 0..8 umrechnen



Netzwerk 2: Je nach grössse der eingelesenen Spannung werden mehr oder weniger Ausgänge angesteuert

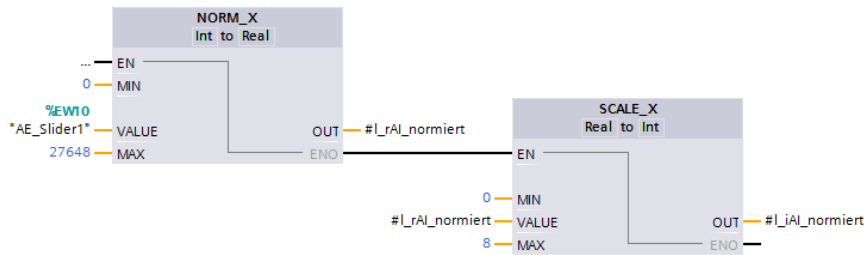


Programm 5: SCL mit Siemens Standards

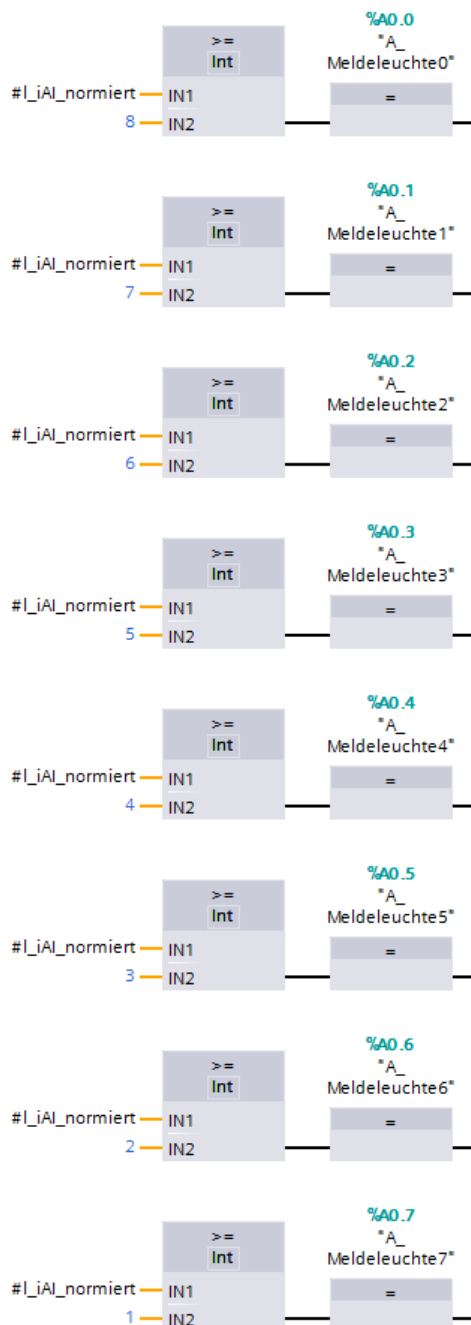
```
1
2 // Analogwert einlesen und normieren (0..10V -> 0..27648 -> 0..8)
3 #l_iTemp := SCALE_X(MIN := INT#0, VALUE := NORM_X(MIN := INT#0, VALUE := "AE_Slider1", MAX := INT#27648), MAX := INT#8);
4
5 // Ausgänge ansteuern
6 "A_Meldeleuchte0" := #l_iTemp >= 8;
7 "A_Meldeleuchte1" := #l_iTemp >= 7;
8 "A_Meldeleuchte2" := #l_iTemp >= 6;
9 "A_Meldeleuchte3" := #l_iTemp >= 5;
10 "A_Meldeleuchte4" := #l_iTemp >= 4;
11 "A_Meldeleuchte5" := #l_iTemp >= 3;
12 "A_Meldeleuchte6" := #l_iTemp >= 2;
13 "A_Meldeleuchte7" := #l_iTemp >= 1;
```

Programm 6: FUP mit Siemens Standards

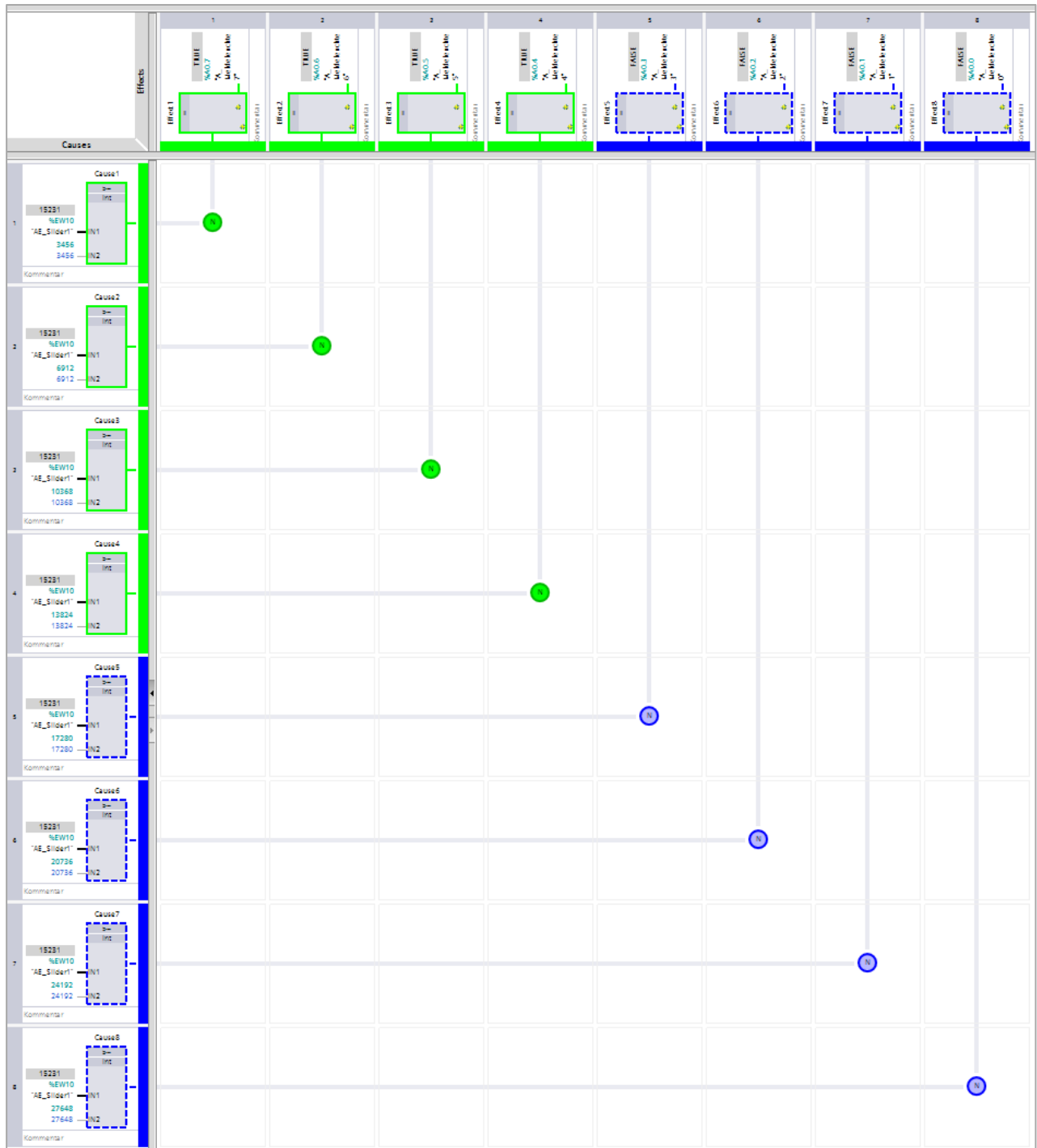
Netzwerk 1: Analogeingang 0..10V (0..27648) auf den Wert 0..8 umrechnen



Netzwerk 2: Je nach grösse der eingelesenen Spannung werden mehr oder weniger Ausgänge angesteuert



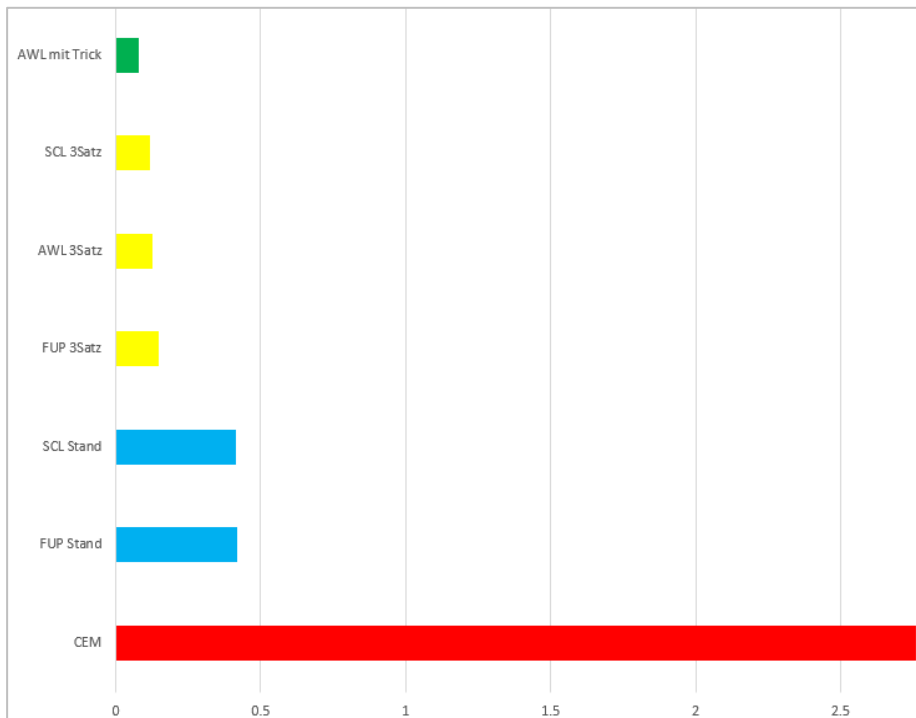
Programm 7: CEM



Messresultat:

| Strukt | Struct | | | | | Messresultat |
|---------------------------|--------|-----|--------------------|-------------------------------------|--------------------------|--|
| AWL_mit_Trick | LReal | 0.0 | 0.0790936844444445 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Messresultat - AWL mit Trickprogrammierung [s] |
| AWL_mit_3Satz | LReal | 0.0 | 0.126604097777778 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Messresultat - AWL mit Dreisatz [s] |
| SCL_mit_3Satz | LReal | 0.0 | 0.11836892 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Messresultat - SCL mit Dreisatz [s] |
| FUP_mit_3Satz | LReal | 0.0 | 0.148586697777778 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Messresultat - FUP mit Dreisatz [s] |
| SCL_mit_Siemens_Standards | LReal | 0.0 | 0.415012711111111 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Messresultat - SCL mit Siemens Standards [s] |
| FUP_mit_Siemens_Standards | LReal | 0.0 | 0.418615493333333 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Messresultat - FUP mit Siemens Standards [s] |
| CEM | LReal | 0.0 | 2.76192971111111 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Messresultat - CEM [s] |

- Rang 1: 0.079ms AWL mit Trickprogrammierung
- Rang 2: 0.118ms SCL mit Dreisatz
- Rang 3: 0.127ms AWL mit Dreisatz
- Rang 4: 0.149ms FUP mit Dreisatz
- Rang 5: 0.415ms SCL mit Siemens Standards
- Rang 6: 0.419ms FUP mit Siemens Standards
- Rang 7: 2.762ms CEM



Fazit:

- > Bei diesem Beispiel siegt der AWL Code, wenn tief in die Trickkiste gegriffen wird. Nachvollziehbar ist dieser Code nur für die wenigsten.
- > FUP, SCL und normal programmierter AWL-Code sind in etwa identisch.
- > CEM ist mit Abstand am langsamsten. Ziel von CEM ist nicht die Geschwindigkeit, sondern intuitive graphische Darstellung von Zusammenhängen.
- > Ich persönlich bevorzuge die SCL Programmierung da dieser Code gut lesbar ist und auch auf einer S7-1200 ablauffähig ist.

Schlusswort:

Diese Messung ist nicht repräsentativ, sondern zeigt nur eine grobe Tendenz auf.