

SGDSL – Kurzreferenz

Spritzguss-DSL (SGDSL) ist eine Hochsprache fuer ASPW-Steuerungsprogramme auf KuASY-Spritzgiessmaschinen (DDR). Sie kompiliert zu bit-identischem EPROM-Code.

3 Abstraktionsebenen

EPROM (Hex)	KuASY-DSL (decompiliert)	SGDSL (Hochsprache)
C2 20 LU Port(6,0,2)	@ A.Ventil.FormSchliessen	ODER Grundzustand:
52 48 B0 Port(5,2,0).1	M.Init.Startmerker	ZWEIG: A.Ventil.FormSchliessen
42 21 SU Port(4,2,1)	& !A.Ventil.FormOeffnen	ZWEIG: M.Init.Startmerker, !A.Ventil.FormOeffnen
62 48 A0 Port(5,2,0).1	-> M.Init.Startmerker	AUSGANG: M.Init.Startmerker

Dateistruktur

```
MASCHINE "KuASY 810/160-IK"
SYMDB "kuasy810_160-IK.toml"
```

```
# GRUPPE 1: INITIALISIERUNG
UND InitMerker:
  WENN: M.Init.Startmerker
  DANN: M.Init.ErstlaufMerker
```

Signalnamen

Praefix	Bedeutung	Beispiel
E.	Eingang (Sensor, Schalter)	E.Hydr.FormGeschlossen
A.	Ausgang (Ventil, Lampe)	A.Ventil.FormSchliessen
M.	Merker (internes RAM)	M.Betrieb.MaschineAktiv
I0.	I/O-Parameter (DAC, BCD)	I0.BCD.EinerBit1
SR1./SR2.	Schieberegister	SR1.Feedback.Stufe3

!Signal = negiert, SELF = Selbsthaltung (Ausgang referenziert sich selbst)

Operatoren (Raw-Modus)

Operator	ASPW-Opcode	Bedeutung
@&	LU	Satzanfang AND
@	LBO	Satzanfang OR
&	SU	weiteres AND
	BO	weiteres OR
->	AO	Ausgang (OR-Kontext)
=>	AU	Ausgang (AND-Kontext)
:=	SRU	Register-Kopie

Block-Typen

UND – Alle Bedingungen muessen wahr sein

```
UND BCDStrobe:
  WENN: M.BCD.Strobe1, E.Hydr.FormGeschlossen
  DANN: M.BCD.Strobe2
```

→ Wenn Strobe1 aktiv UND Form geschlossen → Strobe2 setzen

ODER – Einer von mehreren Zweigen genuegt

```
ODER Grundzustand:
  ZWEIG: A.Ventil.FormSchliessen
  ZWEIG: M.Init.Startmerker, !A.Ventil.FormOeffnen
  AUSGANG: M.Init.Startmerker
```

→ FormSchliessen ODER (Startmarker UND nicht FormOeffnen) → Startmarker

FLIPFLOP — Setzen/Halten mit Selbsthaltung

```
FLIPFLOP Betriebsbereit:
SETZEN: E.Schalter.Start, M.Aktivierung
HALTEN: M.Freigabe, !E.Fehler.NotAus
AUSGANG: M.Betriebsbereit
```

→ Start ODER Aktivierung setzt, haelt sich solange Freigabe UND kein NotAus

VERKETTUNG — Raw-Modus (direkte @-Zeilen)

```
VERKETTUNG SR1InitKaskade:
@& M.Zeitbasis.480ms | SR2.Steuer.TimerZustand & M.Zeitbasis.120ms
| SR2.Steuer.TimerZustand & M.Umschalt.480ms | M.Init.SR1StartTrigger
& SR1.Feedback.Stufe1 -> SR1.Kette.Stufe1
```

→ Fallback fuer komplexe Logik die nicht in Hochsprache passt

TIMER_KETTE — Frequenzteiler-Kaskade

```
TIMER_KETTE TimerTeiler:
TAKT: Timer.Basis.Zweitakt
STUFE: Timer.Umschalt.30MS, Timer.Basis.30MS
STUFE: Timer.Umschalt.60MS, Timer.Basis.60MS
ENDE: Timer.Umschalt.120MS
```

→ Jede Stufe halbiert die Frequenz (Divide-by-2 Kette)

SCHIEBEREGISTER — Kopierkette (rueckwaerts kompiliert)

```
SCHIEBEREGISTER SR_Koeffizienten:
KETTE: K0.1X, K0.2X, K0.3X, K0.4X, K0.5X
```

→ Schiebt Bits von Stufe zu Stufe (letzte zuerst, kein Ueberschreiben)

DEKODER — BCD zu 1-aus-N

```
DEKODER NachdruckDekoder:
BITS: M.Nachdruck.N1, M.Nachdruck.N2, M.Nachdruck.N3
ZUSTAND 0: M.Nachdruck.Niveau0
ZUSTAND 1: M.Nachdruck.Niveau1
ZUSTAND 4: M.Nachdruck.Niveau4
```

→ 3-Bit Eingang → genau ein Ausgang aktiv je nach Binaerwert

BINAERZAEHLER — XOR-Toggle Ripple-Counter

```
BINAERZAEHLER NachdruckZaehler:
TAKT: M.Zaehler.Z3
STUFE: M.Nachdruck.N2U, M.Nachdruck.N1
STUFE: M.Nachdruck.N3U, M.Nachdruck.N2
STUFE: M.Nachdruck.N3
```

→ Jede Stufe toggelt bei Carry der vorherigen (kaskadierter Zaehler)

Weitere Typen

Typ	Zweck
ZUWEISUNG	Wie UND, semantisch: direkte Durchleitung
INIT	Power-On: alle Register loeschen (. \$0F Maske)
SCHRITTKETTE	Sequenzieller Ablauf mit PHASE/EINGANG/AUSGANG
FEHLERSPEICHER	Flankengetriggertes Latch-Array (DIREKT/LATCH)
BITREGISTER	Transparent Latch mit Ladebedingung

Programmstruktur (KuASY 810/160-IK)

GRUPPE 1:	Init & Grundzustand	(3 Saetze)
GRUPPE 2:	SR1-Init, MasterEnable, Takt	(33 Saetze)
GRUPPE 3:	Timer-Teilerkette	(22 Saetze)
GRUPPE 4:	Betriebsart & Zykluslogik	(50 Saetze)
GRUPPE 5:	Anzeigen & Statusregister	(11 Saetze)
GRUPPE 6:	Druck-/Geschwindigkeitsprofile	(65 Saetze)
GRUPPE 7:	SR2-Timer-Paare	(55 Saetze)
GRUPPE 8:	DAC-Dekoder & BCD-Zaehler	(62 Saetze)
GRUPPE 9:	Schaltschrank-Konfiguration	(29 Saetze)
GRUPPE 10:	Ausgangs-Zuordnung	(104 Saetze)

434 Saetze, 3929 Steuerworte

Spritzguss-Zyklus

1. Startbedingung → Betriebsbereitschaft herstellen
2. Form schliessen → Sicherheitsverriegelung pruefen
3. Duese vorfahren → Aggregat an Form anlegen
4. Einspritzen → Schmelze unter Druck in Form pressen
5. Nachdruck → Volumenschwund kompensieren
6. Dosieren → Schnecke dreht, neue Schmelze aufbereiten
7. Kuehlen → Formteil erstarren lassen
8. Form oeffnen → Werkzeughaelften trennen
9. Auswerfen → Formteil entnehmen
10. Wiederholen → bei Automatikbetrieb

Toolchain

```
# Kompilieren: SGDSL → EPROM
asplib sgdsl compile -i programm.sgdsl \
    -hi hi.bin -lo lo.bin -symdb signale.toml

# Verifizieren: gegen Original-EPROMs
asplib sgdsl verify -i programm.sgdsl \
    -hi original_hi.bin -lo original_lo.bin -symdb signale.toml

# Decompilieren: EPROM → KuASY-DSL
asplib kuasy decompile -hi hi.bin -lo lo.bin \
    --symdb signale.toml -o programm.kuasy
```