

DL636B

System-Clock Slave Modul

1. Funktion	2
1.1. Datenblatt	2
1.1.1. Anwendung.....	2
1.1.2. Daten	2
1.1.3. Besonderheiten.....	2
1.1.4. Aufbau	2
1.1.5. Stromversorgung	2
1.2. Blockdiagramm.....	3
1.3. Beschreibung	3
2. Betrieb.....	4
2.1. Konfigurierung	4
2.1.1. Inputs.....	4
2.1.2. Outputs.....	4
2.2. Bedienung.....	4
2.3. Programmierung.....	4
2.3.1. Initialisierung.....	4
2.3.2. Speicherbelegung	4
2.3.3. Beispiel.....	5
3. Fertigung.....	6
3.1. Mechanik	6
3.1.1. Frontplatte	6
3.1.2. Gehäuse.....	6
3.1.3.	6
3.2. Elektronik	6
3.2.1. Schaltbild	6
3.2.2. Bestückungsplan.....	6
3.2.3. Stücklisten	6
3.2.4. Platinenunterlagen	6
3.2.5. Jumper	6
3.2.6. Abel-File	6
4. Test	11
4.1. Aufbau	11
4.2. Ergebnisse.....	11
4.3.	11
5. Modifikation.....	12
5.1. Version	12
5.2.	12
6. Anhang.....	13
6.1. Bausteinunterlagen	13
6.2.	13

1. FUNKTION

1.1. Datenblatt

1.1.1. Anwendung

System-Slave zur Synchronisation von Experimentdaten (Events).

1.1.2. Daten

Parameter	Wert	Dimension
Registertiefe	32	bit
Zeitauflösung (Master)	1	ms
Zeitbereich (Master)	ca. 49	Tage
Serielle Ausgänge	..4	NIM
Transferrate	1	Mbps
Steuerausgänge	..4	NIM

1.1.3. Besonderheiten

Serieller Eingang zur Synchronisation mit Master.

Serielle Ausgänge zur Synchronisation von weiteren Slave-Modulen.

Ausgänge programmierbar.

1.1.4. Aufbau

DL600-Submodul

1.1.5. Stromversorgung

Spannung	Strom	Leistung
+5V	380mA	1,9W
Gesamt		1,9W

1.2. Blockdiagramm

1.3. Beschreibung

Das Modul besitzt einen 16 MHz Quarzoszillator, der nach einem Vorteiler (/16) eine Clock mit der Periode von $1\mu\text{s}$ liefert. Diese dient als Zeitbasis für die serielle Kommunikation mit 1Mbps.

Die Information wird über den Eingang in serieller Form (1 Startbit, D0..D31 Datenbits, Stopbit) mit 1Mbps eingelesen. Ein 32-Bit Register fungiert hier als Schieberegister und kann nicht direkt ausgelesen werden.

Für eine unabhängige Auslese wird das Register normalerweise nach jedem kompletten Datenwort in ein 32Bit Register umgeladen und kann von dort fehlerfrei ausgelesen werden. Da das Umladen auch zwischen zwei (16Bit) Zugriffen vom Rechner erfolgen kann, muß dieses in diesem Fall ausdrücklich vor dem Lesen gesperrt und anschließend wieder freigegeben werden.

2. BETRIEB

2.1. Konfigurierung

2.1.1. Inputs

Der Eingang (oberste Lemobuchse) dient zum Empfangen der seriellen Informationen.

Die rote LED zeigt bei Input von einem Clock-Master im Sekundentakt die Funktion des Zeit-Zählers an!

2.1.2. Outputs

Die Ausgänge (4 untere Lemobuchsen) in NIM können in verschiedener und unabhängiger Weise programmiert werden:

- 1) Serieller Datenstrom zur Synchronisation von weiteren Slave-Modulen.
- 2) Ausgabe eines beliebigen Datenbits D31..D0 des Registers.
- 3) statische Ausgabe.

Die grünen Leuchtdioden zeigen die Pegel der Ausgänge (leuchtet bei NIM=16mA).

2.2. Bedienung

Keine manuellen Bedienungselemente.

2.3. Programmierung

2.3.1. Initialisierung

Bei Sys-Reset werden ALLE Bits (Modes) auf Null gesetzt.

2.3.2. Speicherbelegung

Alle Adressen sind als Offset zur Modul-Adresse definiert. Die Zählweise hier ist Byte-orientiert.

D.h. für VME-Module (DL600) sind die Adressen wie angegeben zur Modul-Adresse zu addieren, z.B:

\$F200 = DL600 Modul-Basisadresse (Short IO; Hex Address Switch=\$F2);

\$0040 = DL635D Submodul-Adresse (Modul 2);

\$000A = Subadresse Select Mode Output 3 (siehe Tabelle)

=====

\$F24A = Adresse für Funktion

R. \$0000	D15	HighByte	D7	LowByte	D0	
	D15..		..D0			Auslese Register Low Word

W=Write, R=Read, SH=Short, STD=Standard, EXT=Extended, \$=HEX, ss=Address-Switches;

Die Bedeutung der Datenbits D5..D0 bei Select Mode Output 1..4 ist wie folgt:

D5	D4	D3	D2	D1	D0	Bedeutung
0	0	x	x	x	x	Ausgang = Serieller Eingang (Default)
0	1	x	x	x	s	Ausgang = s (1= NIM aktiv)
1	b	b	b	b	b	Ausgang = Datenbit Db (b=0..31)

2.3.3. Beispiel

```

const
  VMESubModuleBase = $F240;
  RegisterLow = VMESubModuleBase+$0;
  RegisterHigh = VMESubModuleBase+$2;
  Inhibit = VMESubModuleBase+$4;
  Mode1 = VMESubModuleBase+$6;
  Mode2 = VMESubModuleBase+$8;
  Mode3 = VMESubModuleBase+$A;
  Mode4 = VMESubModuleBase+$C;
  Reset = VMESubModuleBase+$E;
begin
  Reset^ := 0;      {Daten irrelevant, setzt alle ModeBits auf 0}

  Mode1^ := 0;      {Ausgang1 auf Seriell Out zum Treiben eines weiteren Moduls gesetzt}
  Mode2^ := 32 + 9; {Ausgang2 gibt Bit9 des Registers aus}
  Mode3^ := 16 + 0; {Ausgang3 auf statisch NIM=inaktiv gesetzt}
  Mode4^ := 16 + 1; {Ausgang4 auf statisch MIM=aktiv gesetzt}

  Inhibit^ := 1;    {Disable Update}
  Register := (RegisterHigh^ shl 16) + RegisterLow^; {Einlesen des Registers}
  Inhibit^ := 0;    {Freigabe Update}
end;

```

3. FERTIGUNG

3.1. Mechanik

3.1.1. Frontplatte

3.1.2. Gehäuse

3.1.3. ...

3.2. Elektronik

3.2.1. Schaltbild

3.2.2. Bestückungsplan

3.2.3. Stücklisten

3.2.4. Platinenunterlagen

3.2.5. Jumper

J1.2-3; J2.2-3; J3.2-3: Lemobuchsen als Ausgänge;

J20 On : Clock auf Pin 80;

J23 On : -5V aus DC/DC-Wandler;

Alle weiteren Jumper Off;

3.2.6. Abel-File

Siehe File DL636B.ABL!

4. TEST

4.1. Aufbau

4.2. Ergebnisse

4.3. ...

5. MODIFIKATION

5.1. Version

Der Vorgänger dieses Moduls ist das Modul DL635E. Für Neuentwicklungen wird die Verwendung des Submoduls DL636 mit den neuen Bausteinen Lattice 1048E empfohlen.

6. ANHANG

6.1. Bausteinunterlagen

6.2. ...