

| |
|--|
| <h1 style="margin: 0;">DL635D</h1> <h2 style="margin: 0;">System-Clock Master Modul</h2> |
|--|

| | |
|---------------------------------|----|
| 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. Input | 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 | 4 |
| 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 | 9 |
| 4.1. Aufbau | 9 |
| 4.2. Ergebnisse | 9 |
| 4.3. | 9 |
| 5. Modifikation | 10 |
| 5.1. Version | 10 |
| 5.2. | 10 |
| 6. Anhang | 11 |
| 6.1. Bausteinunterlagen | 11 |
| 6.2. | 11 |

1. FUNKTION

1.1. Datenblatt

1.1.1. Anwendung

System-Clock zur Synchronisation von Experimentdaten (Events).

1.1.2. Daten

| Parameter | Wert | Dimension |
|-------------------|--------|-----------|
| Zählertiefe | 32 | bit |
| Zeitauflösung | 1 | ms |
| Zeitbereich | ca. 49 | Tage |
| Serielle Ausgänge | ..4 | NIM |
| Transferrate | 1 | Mbps |

1.1.3. Besonderheiten

Serielle Ausgänge zur Synchronisation von Slave-Modulen.

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

Ein weiterer Counter (/1000) liefert die eigentlichen Zählimpulse (1ms) für die System-Clock, einen Zähler mit 32 Bit. Dieser Zähler fungiert beim Serialisieren auch als Schieberegister und kann nicht direkt ausgelesen werden.

Der Zählerstand wird nach jedem Weiterzählen über die Ausgänge in serieller Form (1 Startbit, D0..D31 Datenbits, Stopbit) mit 1Mbps ausgegeben

Für eine unabhängige Auslese wird die Zählerinformation normalerweise nach jedem Weiterzählen 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. Input

Der Eingang (oberste Lemobuchse) hat keine Funktion.

Die rote LED zeigt im Sekudentakt die Funktion des Zeit-Zählers an!

2.1.2. Outputs

Die Ausgänge (4 untere Lemobuchsen) in NIM liefern den seriellen Datenstrom zur Synchronisation von weiteren Slave-Modulen.

Die grünen Leuchtdioden zeigen den Status der Ausgänge.

2.2. Bedienung

Keine manuellen Bedienungselemente.

2.3. Programmierung

2.3.1. Initialisierung

Bei Sys-Reset werden alle Bits (z.B. Update) auf Null gesetzt, nicht jedoch der Counter. Die Funktion ModulReset (siehe Tabelle) setzt zusätzlich auch noch den Counter auf Null!

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:

\$FE00 = DL600 Modul-Basisadresse (Short IO; Hex Address Switch=\$FE);

\$0060 = DL635D Submodul-Adresse (oberster Steckplatz, Modul 3);

\$0004 = Subadresse Inhibit/_Enable Update (siehe Tabelle)

=====

\$FE64 = Adresse für Funktion

| | | | | | | |
|-----------|------------|----------|-----------|---------|-----------|--------------------------|
| R. \$0000 | D15 | HighByte | D7 | LowByte | D0 | |
| | D15.. | | ..D0 | | | Auslese Counter Low Word |

```
Inhibit = VMESubModuleBase+$4;
Reset = VMESubModuleBase+$E;
begin
  Reset^ := 0;      {Zugriff auf Speicherstelle, Daten irrelevant, setzt Counter = 0}
  Inhibit^ := 1;   {Disable Update}
  Counter := (CounterHigh^ shl 16) + CounterLow^; {Einlesen des Counters}
  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 75;

Alle weiteren Jumper Off;

3.2.6. Abel-File

Module DL635D

Title 'ClockMaster vwa010796 V2.0'

" This is a Clock Master module, which counts time (typ 1ms) and sends data in a
" serial stream to according slave modules.

" The data can be read out.

" For consistent readout set MUpdate to High (disables update of register), read data
" and then clear MUpdate again"

Declarations

```

maxTime                = 999;                "Counter/ms

DataIn                 PIN 27;
Out1..Out4             PIN 11,12,9,10;
TP1..TP8               PIN 36,34,23,21,7,5,117,115;
Clock                  PIN 75;
!SysReset              PIN 82;
!RD,!WR                PIN 81,80;
A4..A1                 PIN 83..86;
D15..D0                PIN 72,70,68,66,64,62,57,59,71,69,67,65,63,61,58,60;

MUpdate                NODE isType 'reg_d,buffer';
MUpdateSync            NODE isType 'reg_d,buffer';
P3..P0                 NODE isType 'reg_d,buffer'; "ms Precounter
T9..T0                 NODE isType 'reg_d,buffer'; "Time PreCounter
S1..S0                 NODE isType 'reg_d,buffer'; "Status
B4..B0                 NODE isType 'reg_d,buffer'; "Bitcounter
C31..C0                NODE istype 'reg_d,buffer'; "Time Counter
R31..R0                NODE istype 'reg_d,buffer'; "Register
SerBit                 NODE;
```

```

Addr                = [A4..A1];
Data                = [D15..D0];
PreCnt              = [P3..P0];
Time                = [T9..T0];
Counter             = [C31..C0];
CounterS            = [C30..C0,C31];
CounterL            = [C15..C0];
CounterH            = [C31..C16];
Register            = [R31..R0];
RegisterL           = [R15..R0];
RegisterH           = [R31..R16];
Bits                = [B4..B0];
Status              = [S1..S0];
SIDLE               = 0;
SSTART              = 1;
SDATA               = 2;
SSTOP               = 3;

RCounterL           = (Addr==0) & RD;
RCounterH           = (Addr==1) & RD;
WCounterL           = (Addr==0) & WR;
WCounterH           = (Addr==1) & WR;
WUpdate             = (Addr==2) & WR;
MReset              = (Addr==7) & WR;
SReset              = MReset # SysReset;

BitNum              = [1,1,1,1,1];

Equations
TP1                 = P3;
TP2                 = T9;
TP3                 = R9; "LED red
TP4                 = !Out1; "LED green
TP5                 = !Out2; "LED green
TP6                 = !Out3; "LED green;
TP7                 = !Out4; "LED green;
TP8                 = MUpdate;

MUpdate.ACLR = SReset;
MUpdate.CLK  = !WUpdate;
MUpdate      := D0;

MUpdateSync.ACLR = SReset;
MUpdateSync.CLK  = Clock;
MUpdateSync      := MUpdate;

Status.ACLR      = SReset;
Status.CLK       = P3;

PreCnt.ACLR      = SReset;
PreCnt.CLK       = Clock;
PreCnt           := PreCnt+1;

Bits.ACLR        = SReset;
Bits.CLK         = P3;

Time.ACLR        = SReset;
Time.CLK         = P3;
when (Time==0) then Time:=maxTime else Time := Time-1;

Counter.ACLR     = MReset;
Counter.CLK      = P3;

Register.ACLR    = SReset;
Register.CLK     = P3;

```

```
Data          = RegisterL & RCounterL;
Data          = RegisterH & RCounterH;
Data.OE      = RCounterL # RCounterH ;

Out1         = SerBit;
Out2         = SerBit;
Out3         = SerBit;
Out4         = SerBit;
```

State_Diagram Status;

STATE SIDLE:

```
SerBit = 1;
when !MUpdateSync then Register:=Counter else Register:=Register;
Counter:=Counter;
If (Time==0) then SSTART else SIDLE;
```

STATE SSTART:

```
SerBit = 0;
Register:=Register;
Counter:=Counter+1;
Bits := BitNum;
Goto SDATA;
```

STATE SDATA:

```
SerBit=C0;
Register:=Register;
CounterS:=Counter;
Bits:=Bits-1;
if Bits==0 then SIDLE else SDATA;
```

End

4. TEST

4.1. Aufbau

4.2. Ergebnisse

4.3. ...

5. MODIFIKATION

5.1. Version

5.2. ...

6. ANHANG

6.1. Bausteinunterlagen

6.2. ...