

Merkzettel, Reminder Sheet

AUTHOR: BERND HEUFELDER

18. April 2019



Inhaltsverzeichnis

1	Verbindungsaufbau mit der Maschine	1
2	Maschineninbetriebnahme	2
3	ComET Command Prompt	2
3.1	Hilfreiche Kommandos	2
4	Maschinenbedienung	2
5	Servo Tools	2
6	Servo-Bundle	2
6.1	XML-File	3
7	Theorie	3
7.1	Encoder	3
7.1.1	SinCos-Encoder	3
7.1.2	EnDat Encoder	4
7.2	Bussysteme	4
7.2.1	EtherCAT	4
7.2.2	ETEL Systeme	5
7.2.3	CAN-Bus	5
7.2.4	Ginlink	5
8	Firmenspezifisch	5
8.1	Firmenstruktur	5
8.1.1	Software	5

8.2	Ansprechpersonen	6
-----	----------------------------	---

1 Verbindungsaufbau mit der Maschine

- PuttY starten
- Maschinennamen eingeben, z.B. fc-4438
- Benutzername: ses
Passwort: dmsu
- Mushell starten
- In der Mushell: `Ultimet_startETNE` eingeben um Maschine "freizuschalten"
- Im Port-File muss die Maschine definiert sein, dann kann übers Dropdown-Menü die Verbindung/Maschine ausgewählt werden um eine Verbindung herzustellen.

Beispiele für *port.properties* Einträge verschiedener Maschinen. Die Grünen Einträge sind immer gleich und rot ist variabel.

- **FlipChip (fc)/8k8**

```
port.8.name=fc-4169           Beliebiger Name, dieser wird im Dropdown Menü angezeigt
port.8.driver=etn://fc-4169:1160 Maschinennamen, muss mit der Beschilderung auf der Maschine übereinstimmen
```

```
port.8.protocol=ETB
port.8.accept.0.driver=ETN:0
port.8.start=auto
```

- **Lupus Testunit**

```
port.8.name=Lupus testunit-401
port.8.driver=etn://testunit-401:1160
port.8.protocol=ETB
port.8.accept.0.driver=ETN:0
port.8.start=auto
```

- **Evo/2k2**

```
port.8.name=EVO 2-modulig evo-005089   Beliebiger Name, dieser wird im Dropdown Menü angezeigt
```

```
port.8.driver=etn://evo-005089:1160
port.8.protocol=ETB
port.8.accept.0.driver=ETN:0
port.8.start=auto
```

- **Beliebige Maschine**

```
port.8.name=VCB-02_ssd           Als Beispiel die VCB-Maschine
port.8.driver=etn://10.10.18.109:1160 Statt Maschinennamen kann auch direkt über die IP-Adresse verbunden werden
```

```
port.8.protocol=ETB
port.8.accept.0.driver=ETN:0
port.8.start=auto
```

2 Maschineninbetriebnahme

Prinzipiell wird das Phasing aller Achsen durchgeführt. Überprüfung aller Encoder. Können alle Achsen gehomed werden? Die Regelung ist bereits getuned.

3 ComET Command Prompt

3.1 Hilfreiche Kommandos

Rufzeichen ! für alle Achsen und .x führt das Kommando nur für die aktuell ausgewählte Achse aus.

name.x	als Rückgabewert erhält man die Benennung der Achse mit welcher man gerade verbunden ist
ver.x	Versionsnummer der gerade gewählten Achse (e.g. 0x316xxxx) -i die ersten 3 Ziffern sind die Versionsnummer und die darauf folgenden kennzeichnen die Revision

3.2 Undocumented Parameters

C9	MLTI Multiplikator
K260	
K261	

4 Maschinenbedienung

Kommandos und Tastenkombinationen

Strg+Shift+F2	Run Application (z.b. zum Start vom Indel-INCO-Explorer) <ul style="list-style-type: none">• konquer: Öffnet File-Explorer des Maschinenpc's<ul style="list-style-type: none">+ Motion-Files (Seq., Reg., ...) sind im Pfad <i>Datacon/share/config/-Servotools</i>
----------------------	---

5 Servo Tools

Erweiterung der Servotool durch Eingabe von *servoToolsDialogExpertMode 2* in der Mushell. Z.B. für den Fall wenn sich Register-Files nicht vom PC auf den Controller updaten lassen, kann dies über die Servotools im Expert-Mode manuell geschehen.

expertMode -i Force overwrite

Terminal -i Registerwerte auslesen

6 Servo-Bundle

Ist ein .zip-File, in welchem sich das XML-File, sämtliche benötigte Register-Files, Sequenzen und Firmware-Files befinden. Auf jeder Maschine befindet sich das Gesamte zip-File. Ein Servobundle kann für alle Maschinenplattformen verwendet werden, da sich die besagten Files für alle Maschinen darin befinden. Jedes Servobundle hat ein version_info.txt, in welchem die Änderungen der einzelnen Versionen gelistet sind.

Sequenzen müssen kompiliert werden bevor sie auf die Maschine überspielt werden. Um ein Kompilieren auf der Maschine zu umgehen wurde der Offline-Sequence-Compiler entworfen. Damit werden die .txt Files in .eseq Files umgewandelt. Im Servo-Bundle befinden sich nur die kompilierten Versionen der Sequenzen.

6.1 XML-File

Im XML-File sind für jede einzelne Maschine die Folgenden Zuweisungen für jede Achse definiert

- Auf welchem Controller-Typ hängt die Achse
- Welche Controller-ID hat die Achse
- Mit welcher Firmware soll die Achse laufen
- Welche Sequence läuft für die Achse
- Welches Register-File ist der Achse zugewiesen
- Welches Homing und Phasing soll die Achse verwenden

7 Theorie

7.1 Encoder

7.1.1 SinCos-Encoder

Im Abtastkopf leuchtet eine Infrarot-LED seitlich auf eine Maßverkörperung mit reflektierenden und nicht-reflektierenden Bereichen. Das Licht wird von den reflektierenden Bereichen durch ein transparentes Phasengitter zurückreflektiert.

In der Erkennungsebene des Abtastkopfes entstehen hierdurch sinusförmige Interferenzstreifen. Das optische System integriert aus mehreren Teilungsperioden einen Durchschnittswert und filtert Signale heraus, die nicht von der Maßverkörperung reflektiert werden. Dadurch wird die Stabilität des Signals auch dann gewährleistet, wenn die Maßverkörperung verschmutzt oder leicht beschädigt ist.

Durch die einzigartige Optikkonstruktion des Abtastkopfes sowie die Funktionen Auto Gain Control (AGC) und Auto Offset Control (AOC) des REF Interface sind geringe Kurzzeitfehler gewährleistet, die typischerweise einen zyklischen Fehler (SDE) von weniger als $\pm 0,05$ um ergeben.

Das REF Interface verfügt über eine Einstell-LED, die durch ein grünes Licht das Erreichen der optimalen Einstellung anzeigt.

Eine Referenzmarke oder ein einfacher Endschalter sind bei diesem Abtastkopf erhältlich. Die Referenzmarke dient als Referenz oder Nullpunkt für den Abtastkopf, während der Endschalter das Ende des Verfahrbereiches signalisiert.

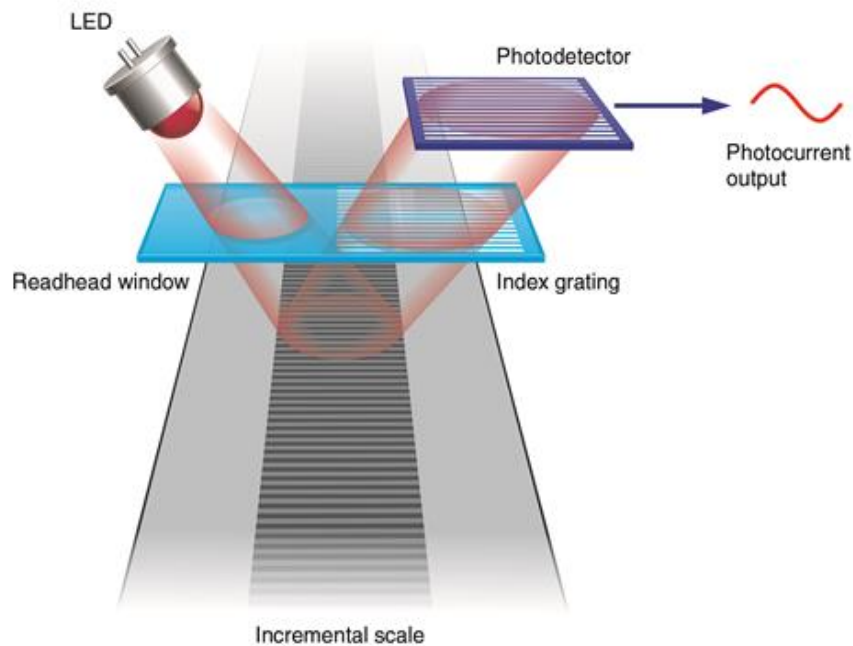


Abbildung 1: SinCos Encoder Hardwareaufbau des Encoderkopfs mit Lineal. Das Index Grating ist nur eine transparente Platte.

7.1.2 EnDat Encoder

Es gibt ein grobes Lineal in welchem die absolute Position kodiert ist und ein feineres für die inkrementalen Schritte (feinschritt). Die Digitalisierung des Positionswertes erfolgt oft direkt am Encoderkopf.

7.2 Bussysteme

7.2.1 EtherCAT

EtherCAT (Ethernet for Control Automation Technology) ist ein von der Firma **Beckhoff Automation** initiiertes Echtzeit-Ethernet.

EtherCAT unterscheidet sich wesentlich von anderen Industrial Ethernet Lösungen. Während bei diesen der vom Master versendete Standard Ethernet Frame (gemäß IEEE 802.3) in jeder Anschaltung zunächst empfangen, dann interpretiert und die Prozessdaten weiterkopiert werden, entnehmen beim EtherCAT die EtherCAT Slave-Geräte die für sie bestimmten Daten, während das Telegramm das Gerät durchläuft. Ebenso werden Eingangsdaten im Durchlauf in das Telegramm eingefügt.

Zykluszeiten $\leq 100\mu s$ mit niedrigem Jitter.

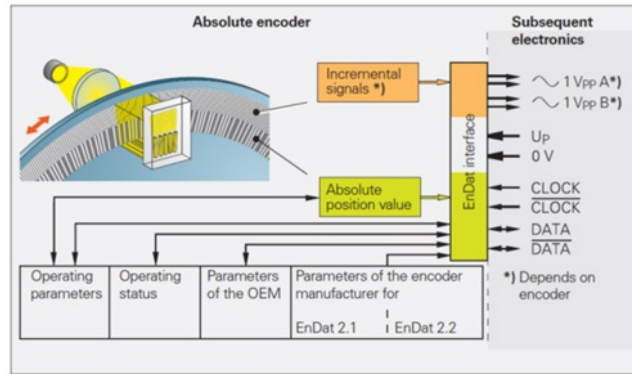
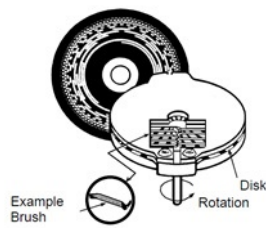


Abbildung 2: Absolutencoder

7.2.2 ETEL Systeme

TransnET Für Echtzeit-Anwendungen mit mehreren Achsen. Die TransnET Synchronisation mit Jitter im Nanosekunden-Bereich ist die richtige Lösung für deterministische und zuverlässige Kommunikation innerhalb von Maschinen.

Turbo-ETEL-Bus Multi-axis fast bus mit 100Mbps

7.2.3 CAN-Bus

Der CAN-Bus (Controller Area Network) ist ein serielles Bussystem und gehört zu den Feldbussen. Feldbus ist ein Bussystem zur Kommunikation für Sensoren, Aktoren in einer Anlage bzw. Maschine.

Funktionsprinzip: Der CAN-Bus arbeitet nach dem „Multi-Master-Prinzip“ d. h., er verbindet mehrere gleichberechtigte Steuergeräte. Ein CSMA/CR-Verfahren löst Kollisionen (gleichzeitiger Buszugriff) auf, ohne dass die gewinnende, höher priorisierte Nachricht beschädigt wird. Der CAN-Bus ist nicht echtzeitfähig, da nicht garantiert werden kann ob der Bus zu gegebenen Zeitpunkt gerade frei ist.

7.2.4 Ginlink

Busverbindung vom COPMAS zum Hammer.

8 Firmenspezifisch

8.1 Firmenstruktur

8.1.1 Software

BIB Softwareabteilung, bestehend aus OMA und Betriebssystem für die Maschine (Linux).

OMA

Hardwareabstraktion auf Softwareebene; Applikation erhält möglichst einheitliche Objekte für unterschiedliche Hardware. Die Applikation verwendet diese Hardware dann entsprechend und

programmiert damit einen Prozessablauf. Kommunikation der HW über CAN-Bus und PC. OMA ist nicht echtzeitfähig (z.b. Rückmeldung nach Bewegungsende). Bildverarbeitung ist eigenständig und nicht Teil der OMA. Auch wird durch die OMA die Topologie der Hardware, also welche HW an welchem Knoten hängt, verschleiert.

Link zur Doku: <https://confluence.besi.com/display/MA/interferometric+systems+RENISHAW>

Applikation Programiert einen Prozessablauf und seine Bewegungen. Ebenso wird die Benutzeroberfläche auf der Maschine von der Applikation bereitgestellt.

8.2 Ansprechpersonen

Simon Holzknecht	Chef Software Tester, Ansprechpartner zum Reservieren von Maschinen
Michael Kirchler	Zuständiger System Engineer für 2k2
Bernd Jandl	Manager Machine Integration Software