

V3.x

# PLC-BDE-Bibliothek productionLog.lib für TwinCAT-PLC

Beckhoff Automation GmbH  
8.6.2018



# ProductionLog.LIB V 3.2.x

## Historie:

### Erweiterungen in Version 3.2.0

- Statusinformationen für Standardsignale als globale Variablenstruktur auslesbar
- Informationsstruktur angelegt
- Informationsvariable in STRUCT\_ProductionLogConfigFromBDA definiert

### Erweiterungen in Version 3.1.0

- Reset-Funktionen werden gesondert behandelt, da sie kein Schreiben für BDE-Meldungen auslösen
- alter MsgTyp "ProdReset" wird kompatibel behandelt > bBarProdReset"
- BDEOnOff wird als Ein/Aus-Schalter ausgewertet.
- Schreiben von Bereichsnutzungen auf Basis der „Area“-Vorgabe als Wartungseinträge
- Update von Schmierungseinträgen in Wartungstabelle (BearbNr: -2 oder -3)  
Es wird die aufgelaufene Automatikzeit aufgerechnet  
(Hierzu muss der aktuelle ProdStatServer (12/2016) aktiviert sein)
- Autoeinträge für genutzte Maschinenbereiche, abschaltbar über PLCOption
- Begründungen für Timeouts (Fehlzeit, Stopp) werden in die Statusmeldung selbst geschrieben

## Eigenschaften von 3.1.x:

- die Registry-Freigabecodes werden statisch gesetzt, es sei denn sie werden gezielt abgefragt
- Kompatibilität zu V1.6.x, V2.x.x  
(Die alten Funktionalitäten sollten nur aus Kompatibilitätsgründen genutzt werden, und/oder durch aktuelle Funktionsaufrufe ersetzt werden!)
- Möglichkeit die BDE-Funktionen auch ohne die BDA-Applikation genutzt werden können
- Sprachabhängige Defaulttexte für Standardmeldungen
- Sprachwahl über CONSTANT : BDEDEF\_LANGUAGE:INT := 7; möglich
- Sprachumschaltung Online per *Lib-Instanz*.ProductionConfig.Language := x;
- Sprachwahl: 7=Deutsch, 9=Englisch, 12=Französisch, 0: alles bleibt wie einmal gesetzt
- Autoeinträge für Maschinenbereichsnutzungen in Wartungstabellen
- Update vom Schmierungseinträgen in Wartungstabelle

## Voraussetzungen:

BDA-Version
Die Oberfläche muss die Version von 28.12.2016 oder jünger sein.
ProdStatserver-Version
Der Server muss in Version 1.241 oder jünger aktiv sein.
projAutostart.exe-Version
Das Startupprogramm in Version vom 28.12.2016 oder jünger aktiviert sein.

## Funktionsbaustein

Version: 3.1.x

Stati-  
on

FBWRITEPRODUCTIONLOG	
— ProductionConfig : STRUCT_PRODUCTIONLOGCONFIGFROMPLC	bBusy : BOOL
— ProductionLog : STRUCT_PRODUCTIONLOG	bWritingXML : BOOL
— LogDatei : STRUCT_LOGDATEI	bReadingREG : BOOL
— ProductionState : STRUCT_PRODUCTIONSTATE	errID_Enable : UDINT
— bCloseActEntry : BOOL	ProductionComOut : STRUCT_PRODUCTIONLOGCOM_OUT
— ProductionComIn : STRUCT_PRODUCTIONLOGCOM_IN	

### Übergabeparameter:

<b>ProductionConfig</b>
In den Konfigurationsdaten können Maschinenparameter zur Einstellung von Optionen oder Pfaden übergeben werden.
<b>ProductionLog</b>
Der Eingangsparameter dient zur Übergabe von Meldungen  Meldungen können über Ihren Typ definiert werden. Die Datenübergabe ist in der Regel für alle Meldungen gleich. In den übergebenen Meldungstexten, muss auf die Übergabe von Sonderzeichen, so wie Umlaute, verzichtet werden. Die Übernahme der gesetzten Meldungen erfolgt über den Aufruf der Instanz nach dem Setzen der Meldungsinformationen und dem Setzen des Schreibbits. Es kann immer nur eine Meldung übergeben werden. Die Größe des Meldungspuffers ist auf ,30' beschränkt. Der aber zyklisch abgearbeitet wird.
<b>LogDatei</b>
Der Parameter dient zur Übergabe von Meldungen die nicht als BDE-Meldungen erfasst werden, sondern direkt in das Logbuch der Maschinenapplikation geschrieben werden.  Meldungen können über Ihren Typ definiert werden. Die Datenübergabe ist für alle Meldungen gleich. In den übergebenen Meldungstexten, muss auf die Übergabe von Sonderzeichen, so wie Umlaute, verzichtet werden. Die Übernahme der gesetzten Meldungen erfolgt über den Aufruf der Instanz nach dem Setzen der Meldungsinformationen und dem Setzen des Schreibbits. Es kann immer nur eine Meldung übergeben werden. Die Größe des Meldungspuffers ist auf ,30' beschränkt. Der aber zyklisch abgearbeitet wird.
<b>ProductionState</b>
In der Übergabestruktur können von der Maschine erfasste Signalzustände übergeben werden. Auf Grund dessen werden die Produktionszeiten und Produktionszyklen erfasst.
<b>bCloseActEntry</b>
Über das Setzen dieses Bits können alle aktuell gepufferten Meldungen und Produktionsstatuszeiten abgeschlossen werden. Mit dem Abschluss werden die Daten als BDE-Meldungen weggeschrieben. Der aktuelle Zustand bleibt aber erhalten und wird weiter verarbeitet. Das Bit wird nach Abschluss der Funktion zurückgesetzt.
<b>ProductionComIn</b>
Über die Struktur „ProductionComIn“ müssen Quittierungen von abgesetzten Events erfol-

gen. Diese korrespondieren mit der Ausgabestruktur „ProductionComOut“.  
(Nur aus Kompatibilitätsgründen zu V 1.6.4 verwenden!)

## Rückgabeparameter:

bBusy	True: Während Übernahme der Meldungen False: Nach erfolgter Übernahme
bWritingXML	True: Während Schreibfunktion für XML-BDE-Daten aktiv. False: Schreibfunktion im Wartezustand
bReadingREG	True: Während Freigaben aus Windowsregistrierung gelesen werden. False: Wenn Funktion beendet.
errID_Enable	<>0: Fehlercode wenn Freigaben nicht gelesen werden konnten. = 0: Freigaben aus Windowsregistrierung konnten gelesen werden.
ProductionComOut	Struktur mit boolschen Variablen zur Auslösungen von Events (Nur aus Kompatibilitätsgründen zu V 1.6.4 verwenden!)

## ProductionConfig:

Konfigurationsdaten für globale Einstellungen der Bibliothek:

```

TYPE STRUCT_ProductionLogConfigFromPLC :
STRUCT
  UserName      :STRING;    (* aktueller Benutzer name (User) *)
  ShiftNo       :INT;       (* aktuelle Schichtnummer (ShiftNo) *)
  Machine       :STRING;    (* aktueller Maschinennamen *)
  Area          :STRING;    (* aktueller Maschinenbereich *)
  xmlPath       :STRING;    (* Zielpfad XML-BDE-Dateien *)
  bLogOnOff     :BOOL;      (* Logging On/Off *)
  Language      :INT;       (* Sprachwahl per Code für Defaulttexte *)

  (* Additional options *)
  xmlPathToolCount :STRING; (* Zielpfad XML-BDE-Dateien für Werkzeug/Wartungsfunktionen *)
  xmlPathLogEntry  :STRING; (* Zielpfad XML-LogDatei.mdb *)

  InstanceCode    :STRING;  (* PreCode für BDE-Ausgangsdateien (leer wenn nur eine Instance oder Maschinenrechner) *)

  bCheckRegistry  :BOOL;    (* Teste auf neue Registrierungseinträge *)
  iPieceCountCode :INT;     (* Code für Teilezählungsfunktion *)

  tCycleTime      :TIME;    (* Zykluszeit für Maschine in Sekunden / Auswertung mit GlobalAutomatic *)
  tCycleTimeout   :TIME;    (* Timeout für Zyklusticks => BDA-MsgBox: SystemReason / Zeit in Sekunden *)
  tCycleTimerange :TIME;    (* *** Old *** Please use tCycleTimeout Or tLevelMakroFault *)

  tCycleTimeToolCount :TIME; (* Zyklus für Schreibfunktion Werkzeugzählung in Sekunden *)
  tCycleTimeMsgRefresh:TIME; (* Zyklus für ProductionMsg-Refresh 0=Aus / in Minuten *)

  tLevelStopReason :TIME;   (* Maximale unbegr. Zeit zwischen Autom.Stop und Autom.Start > Level = Get UserReason *)

  tLevelMakroFault :TIME;   (* Zeit zwischen zwei Lifeticks > Level = MarkoFault Autom.Stopp *)
  iLevelMakroFault :UINT;   (* Zähler MakroStops > Level = Production msg AutomaticFault *)

  tEPAutomatic    :TIME;    (* Entprellzeit Automatiksignal *)

  plcOptions      :STRUCT_PLCOPTIONS; (* Switch On/Off special Options *)
END_STRUCT
END_TYPE

```

Strukturelemente:

	Die sechs nachfolgenden Parameter werden automatisch von der Oberfläche übergeben.
UserName	Benutzername des an der Maschine eingeloggten Benutzers. Der Benutzername wird automatisch beim Login oder Benutzerwechsel übergeben.
ShiftNo	Schichtnummer. Die Schichtnummer kann beim Benutzerwechsel gesetzt werden und wird automatisch übergeben.
Machine	Maschinenbezeichnung aus Konfigurationsdaten. Diese wird ebenfalls automatisch übergeben.
Area	Bezeichnung Maschinenteil. Die Bezeichnung wird automatisch übergeben.
xmlPath	Standardmäßiger Zielpfad für die erzeugten XML-Dateien. Dieser sollte auf der Verzeichnis ProductionLog im Applikationsverzeichnis der Maschine zeigen. Werden die Daten für eine externe Auswertung zur Verfügung gestellt, dann muss dieser Pfad freigegeben sein.
bLogOnOff	= True, kennzeichnet das Schreiben der oben aufgeführten Parameter und wird von der Bibliothek nach der Übernahme zurückgenommen.
	Die nachfolgenden Parameter können projektspezifisch gesetzt werden und müssten entsprechend in einer Parameterliste vorgebar sein.
xmlPathToolCount	Pfad für die XML-Dateien, die Information zur Werkzeugzählung beinhalten. Wird hier nichts vorgegeben, dann wird automatisch der xmlPath als Zielpfad angenommen.
xmlPathLogEntry	Pfad für die XML-Dateien, die Logbucheinträge beinhalten. Dieser sollte standardmäßig auf das Verzeichnis zeigen auf das der ProdStatServer arbeitet, der Einträge in der Maschine speichert. Wird hier nicht vorgegeben, wird automatisch der xmlPath als Ziel angenommen.

InstanceCode	<p>Die hier programmierte Zeichenfolge dient als eindeutige Kennung für die Ausgabedateien der erzeugten Instanz. Die Zeichen werden vor den aus Datum und Uhrzeit erzeugten Dateinamen der Ausgabedateien gestellt. Somit ist ein Erzeugen von Dateien zur selben Uhrzeit aus verschiedenen Instanzen möglich.</p> <p>Dies ist auch notwendig, wenn auf mehreren Rechner an einer Maschine Betriebsdaten erfasst werden und zum Hauptrechner kopiert werden.</p>
bCheckRegistry	Abfrage der Registryeinträge zur Freigabe der Optionen zur Laufzeit des Programms. Im Startmoment wird automatisch die Registry abgefragt. Das Aufrufbit wird nach erfolgter Auslesung zurückgesetzt.
iPieceCountCode	<p>0 = zählen der produzierten Teile über den BarReady-Aufruf (Default)</p> <p>1 = zählen der produzierten Teile über den PieceReady-Aufruf</p> <p>2 = zählen der produzierten Teile über bGlobalPieceTick</p> <p>(siehe: Datentypen - ENUM_LOGOPTIONCODES)</p>
tCycleTime	Taktzeit der Maschine, sollte in Sekunden vorgegeben werden. Wird nichts vorgegeben, wird eine Taktzeit von 20sec angenommen.
tCycleTimeout	<p>Zeitfenster für Meldung von Produktionsstörzeiten. Wird nach Ablauf dieser Zeit kein neuer Lifetick gesetzt, wird eine Begründung für diesen Maschinenstopp angefragt.</p> <p>Der Wert „0“ schaltet diese Funktion aus.</p> <p>Minimalwert 15 Sekunde, Standardwert &gt;= 70 Sekunden</p>
tCycleTimeToolCount	<p>Taktzeit für das Wegschreiben der gezählten Werkzeugeinsätze. Die Zeitvorgabe sollte in Sekunden erfolgen. Wird nichts vorgegeben, wird eine Taktzeit von 15Minuten angenommen.</p> <p>Intern steht ein Ringpuffer mit 25 Plätzen zur Verfügung, der aber zyklisch nur bei einem Belegungsgrad &gt; 80% weggeschrieben wird, um die Einträge in der BDE-Datenbank zu minimieren.</p> <p>Steht das Signal ‚bGlobalAutomatic‘ zur Verfügung, werden die gezählten Werkzeuge schon vorzeitig bei der fallenden Flanke des Signals weggeschrieben.</p>
tCycleTimeRefresh	<p>Aktualisierungszeit für anstehende Automatismeldungen. Durch die Funktion wird der aktuelle Status auch in der BDA sichtbar.</p> <p>Der Wert „0“ schaltet diese Funktion aus.</p> <p>Mindestvorgabe 30Sec (Default: 70Sec)</p> <p><b>!! ProdstatServer in Version 1.201 oder jünger !!</b></p>
tLevelStopReason	<p>Maximale Zeit die zwischen Automatikstopp und erneuten Automatikstart verstrichen sein darf ohne dass eine Begründung abgefragt wird. Ergibt sich eine größere Zeit, muss eine Begründung für diesen langen Stopp angegeben werden.</p> <p>Der Wert „0“ schaltet die Funktion aus.</p> <p>Minimalwert ist 20 Minuten, Standardwert &gt;= 20 Minuten</p>
tLevelMakroFault	<p>Zeitspanne zwischen zwei Lifeticks, ab der ein interner Stopp als „Makro-Stop“ interpretiert werden soll. Der Wert ist in Sekunden vorzugeben.</p> <p>Der Wert „0“ schaltet die Funktion aus.</p>
iLevelMakroFault	<p>Zählgrenze für aufeinanderfolgende Makrostopps ab den einen besonderen Automatikseintrag erfolgt. Der Eintrag erfolgt immer dann wenn die Maschine wieder im „normalen“ Taktrhythmus ist oder mit Automatik aus.</p> <p>Der Wert „0“ schaltet die Funktion aus.</p>
tEPAutomatic	<p>Entprellzeit für das Signal „bGlobalAutomatic“. Default wird 1 Sekunde angenommen wenn keine gezielte Vorgabe.</p> <p>Entprellt wird positive und negative Flanke über die eingestellte Entprellzeit.</p>

plcOptions	Freigabebits für besondere Optionen:	
	bBDERunsOnSlave	Die BDE läuft auf Slaverechner, Produktionsstatussignale und Meldungen werden nicht abgesetzt. Die Verarbeitung muss am Hauptrechner erfolgen.
	bBDEWriteSingelPiece	Schreibe jede PieceReadyMsg per BDE-Msg. Die Summierung ist ausgeschaltet.
	bBDEFilterZeroProd	Schreibe Stoppzustand auch wenn kein Lifetick gekommen ist.
	bBDEWriteOnlyMakroFault	Die Fehlzeit wird nur erfasst wenn sie größer der Vorgabe „tLevelMakroFault“ ist.
	bBDEWriteProdDuration	Nutze Stabstart- und -stopmeldung für Stabzeitmessung
	bBDEWithoutBDAAppl	BDE arbeitet ohne die BDA-Applikation
	bBDEWithoutAutoAreaMsg	BDE erzeugt keine Wartungsmeldung auf Basis der Bereichsvorgabe (Wartungsfunktion: -93) und keine Betriebsstundenzählung auf Basis der Werkzeugwartungsfunktion „-2“, „-3“
	iBDEProdChangedCode	Code für Erkennung Produktionswechsel anhand der Stabdaten: 1: Wechsel bei Jobwechsel 2: Wechsel bei Profilwechsel 4: Wechsel bei Farbwechsel 8: Wechsel bei Profiltypwechsel Default ist Summe (=15) alles aktiv



## ProductionLog:

BDE-Meldungsvorgabe:

```

TYPE STRUCT_ProductionLog :
STRUCT
    bWriteMsg      :BOOL;          (* set to True: Write Msg *)
    msgType        :STRUCT_MSGTYPE; (* selected Msgtyp *)
    msgText        :STRUCT_MSGTEXT; (* Msg-Data *)
    msgLogInfo     :STRUCT_MSGDEFAULTS; (* Msg-Default LogData *)
    msgTextProdData :STRUCT_PRODDATA; (* Msg-Data / Production *)
    msgTextToolCount:STRUCT_TOOLDATA; (* Msg-Data / ToolCount *)
END_STRUCT
END_TYPE

```

Strukturelemente:

bWriteMsg	Bit für Übergabe einer Meldung, wird nach Übernahme zurückgesetzt. Maximal können 30 Meldungen gepuffert werden.
msgType	<p>Meldungstypen: Zuordnung der übergebenen Meldung zu einem bestimmten Meldungstyp. (siehe zusätzliche Beschreibung zur Betriebsdatenerfassung).</p> <pre> TYPE STRUCT_MSGTYPE : STRUCT     Plc      :STRUCT_PLMSGTYPEN; (* SPS-Meldungen: BufferEmpty, NoQuit... *)     User     :STRUCT_USERMSGTYPEN; (* Bedienermeldungen *)     Maintenance :STRUCT_MAINTENANCEMSGTYPEN; (* Wartungsmeldungen ... *)     Machine  :STRUCT_MACHINEMSGTYPEN; (* Maschinenmeldungen ... *)     Shift    :STRUCT_SHIFTMSGTYPEN; (* Schichtmeldungen: Start, Stop *)     Production :STRUCT_PRODMSGTYPEN; (* Eintrag in Produktionszahlendaten ... *)     Tool     :STRUCT_TOOLMSGTYPEN; (* Werkzeugzählung *)     Station  :STRUCT_STATIONMSGTYPEN; (* Station(0 -31) im Fehlerzustand *) END_STRUCT END_TYPE </pre>
msgText	<p>Meldungsdaten:</p> <pre> TYPE STRUCT_MSGTEXT : STRUCT     MsgID      :UINT; (* MsgID für Meldungstext *)     MsgText    :STRING; (* Meldungstext *)     MsgDescription :STRING; (* zusätzliche Beschreibung *)     Duration   :DINT; (* extern erfasste Zeitspanne *)     Counter    :UDINT; (* Meldungsanzahl *) END_STRUCT END_TYPE </pre> <p>MsgID: Muss nur bei „PLC.bSpecialMsgID“ gezielt gesetzt werden.  Duration: Dauer muss in Sekunden vorgeben werden.  Counter: Anzahl der zu schreibenden Meldungen (Default = 1 Eintrag)</p>
msgLogInfo:	<p>Meldungsgrundlagen:</p> <pre> TYPE STRUCT_MSGDEFAULTS : STRUCT     UserName   :STRING; (* Bedienername für Meldungseintrag *)     ShiftNo    :INT; (* Schichtnummer für Meldungseintrag *)     Machine    :STRING; (* Maschinenname für Meldungseintrag *)     Area       :STRING; (* Maschinenbereich für Meldungseintrag *)     MsgDate    :STRING; (* Meldungsdatum für Meldungseintrag *)     MsgTime    :STRING; (* Meldungszeit für Meldungseintrag *) END_STRUCT END_TYPE </pre> <p>Hierüber können Vorgaben für Datum, Uhrzeit, Maschine, Maschinenteil usw. gemacht werden, die von den in ProductionConfig gemachten Vorgaben abweichen können.  Werden diese Vorgaben nicht gesetzt, erfolgen die entsprechenden Einträge aus der Übergabe in ProductionConfig. Datum und Uhrzeit werden während des Eintrags automatisch erzeugt. Das gilt für alle Einträge.</p>

msgTextProdData	<p>Produktionsdaten: Hier muss die Information zu einer Fertigmeldung übergeben werden.</p> <pre> TYPE STRUCT_PRODDATA : STRUCT   Job           :STRING;    (* LosName *)   BarNo         :UINT;      (* StabNr *)   PieceNo       :UINT;      (* TeileNr / only writing with bPieceReady... *)   ProfileName   :STRING;    (* ProfilName *)   Color         :STRING;    (* Profilfarbe *)   ProfileTyp    :STRING;    (* Profiletyp: String user defined in proflidata: FL=Flügel, RA=Rahmen *)   Commission    :STRING;    (* Kundendaten: Kommission ... *)   Designation   :STRING;    (* Kundendaten: Bezeichnung ... *)   BarLength     :LREAL;     (* Stabrohlänge *)   Pieces        :UDINT;     (* Teilezahl im Stab *)   PieceLength   :LREAL;     (* Summe Teilleängen im Stab / oder Teilleänge bei bPieceReady *)   Course        :INT;       (* Laufbahnnummer *)   Counter       :INT;       (* Eintragszähler für bBarReady *)   RestLength    :LREAL;     (* Restteillänge im Stab *)   WasteLength   :LREAL;     (* Abfalllänge im Stab *)   RestPieces    :UINT;      (* Anzahl Restteile *)   WastePieces   :UINT;      (* Anzahl Abfallstücke *)   BarMeasureHeight:LREAL;   (* Gemessene Stabhöhe *)   BarMeasureWidth:LREAL;    (* Gemessene Stabbreite *)   BarMeasureData:STRING;    (* Allgemeine Information zur Messung *)   TreatmentCounter:UINT;    (* Anzahl Bearbeitungen im Stab / Teil *)   TreatmentTime :UDINT;     (* Bearbeitungszeit Stab / Teil *) END_STRUCT END_TYPE </pre> <p>TreatmentTime muss in 1/1000 Sekunden übergeben werden. In der Auswertedatenbank wird dann in 1/10 Sekunden gezählt. Die Teile- und Stablängen müssen in Millimetern übergeben werden, Nachkommastellen werden von der Bibliotheksfunktion automatisch abgeschnitten. Die Felder BarMeasure... sind standardmäßig nicht in der BDE-Datenbank angelegt, falls notwendig sind sie also anzulegen.</p>
msgTextToolCount	<p>Werkzeugzählungsdaten: Hier muss die Information zu einem Werkzeug übergeben werden, dessen Einsatz gezählt werden soll.</p> <pre> TYPE STRUCT_TOOLDATA : STRUCT   ToolNo        :DINT;      (* Werkzeugnr. *)   ToolDescription:STRING;    (* Werkzeugbeschreibung *)   TreatmentNo    :INT;       (* Bearbeitungsnr. *)   TreatmentDescription:STRING; (* Bearbeitungsbeschreibung *)   TreatmentCounter:INT;      (* Zählerinkrement *)   TreatmentTime  :DINT;      (* Zeitinkrement *) END_STRUCT END_TYPE </pre> <p>Als Minimalinformation muss die ToolNo und ToolDescription übergeben werden, in diesem Fall wird der TreatmentCounter der Wert ,1' angenommen. Alle weiteren Angaben dienen als zusätzliche Information. Bei Vorgabe einer Bearbeitungszeit über TreatmentTime muss diese Übergabe in 1/10-Sekunden erfolgen. Wenn die Übergabe nicht gewünscht ist sollte der Wert ,0' übergeben werden oder keine Übergabe erfolgen. Es ist zu beachten, dass die gezählten Werkzeugeinsätze nur zyklisch über den Timer „tCycleTimeToolCount“ weggeschrieben werden.</p>

## LogDatei:

Logdatei-Meldungsvorgabe:

```
TYPE STRUCT_LogDatei :  
STRUCT  
    bWriteMsg      :BOOL;          (* set to True: Write LogDatei-Msg *)  
    msgType        :STRUCT_LOGMSGTYPEN; (* selected MsgTyp *)  
    msgLogInfo     :STRUCT_LOGENTRYDEFAULTS; (* Defaultdata *)  
    msgLogEntry    :STRUCT_LOGENTRYTEXT;  (* Logbucheintrag *)  
END_STRUCT  
END_TYPE
```

Strukturelemente:

bWriteMsg	Bit für Übergabe einer Meldung, wird nach Übernahme zurückgesetzt. Die Einträge erfolgen nicht in der ProductionLog-Datenbank, sondern im Logbuch (LogDatei.mdb) der Maschine. Daher sollte der Pfad für diese Meldungen immer auf dem Pfad der Maschine liegen und nicht ausgelagert sein.
msgType	Meldungstyp:  <pre>TYPE STRUCT_LOGMSGTYPEN : STRUCT     bLogHint      :BOOL;  (* LogDateieintragstyp: Hinweis *)     bLogFault     :BOOL;  (* LogDateieintragstyp: Fehler *)     bLogWarning   :BOOL;  (* LogDateieintragstyp: Warnung *) END_STRUCT END_TYPE</pre>
msgLogInfo	Hier können abweichend zu den in ProductionLogConfig übergebenen Daten andere Vorgaben gemacht werden. Wird hier nichts übergeben, erfolgen die Einträge automatisch. Das gilt für alle Einträge.  <pre>TYPE STRUCT_LOGENTRYDEFAULTS : STRUCT     Area      :STRING;  (* Meldungsbereich *)     MsgDate   :STRING;  (* Meldungsdatum *)     MsgTime   :STRING;  (* Meldungszeit *) END_STRUCT END_TYPE</pre>
msgLogEntry	Hier muss die in das Logbuch einzutragende Meldung übergeben werden.  <pre>TYPE STRUCT_LOGENTRYTEXT : STRUCT     MsgText     :STRING;  (* Meldungstext *)     Duration    :DINT;    (* Zeitspanne für Meldungsstatus *) END_STRUCT END_TYPE</pre> <p>Duration / Dauer muss in Sekunden vorgeben werden.</p>

## ProductionState:

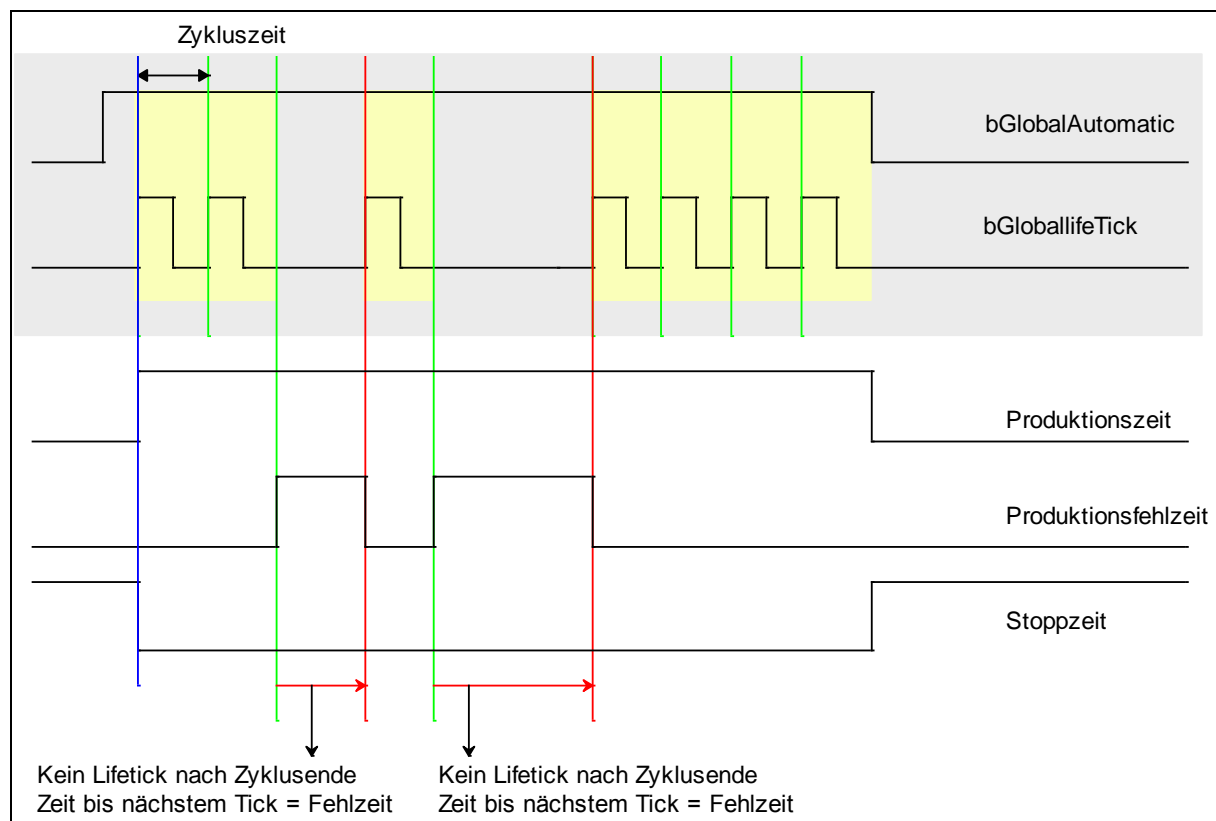
```

TYPE STRUCT_ProductionState :
STRUCT
    bGlobalAutomatic:      BOOL;      (* pFlanke: Maschine im Automatikstatus *)
    bGlobalPause:          BOOL;      (* pFlanke: Maschine im Pausenstatus *)
    bGlobalPieceTick:      BOOL;      (* pFlanke: Life ticker mit Teilezählung *)
    bGlobalLifeTick:       BOOL;      (* pFlanke: Life ticker ohne Teilezählung *)
    bGlobalEmergencyStop:  BOOL;      (* True: mit nFlanke an Automatik = Grundabfrage *)
END_STRUCT
END_TYPE
  
```

Strukturelemente:

bGlobalAutomatic	0 = wenn Gesamtanlage in Ruhezustand / Grundposition 1 = wenn Gesamtanlage Automatik aktiv
bGlobalPause	0 = keine Pause aktiv 1 = Pause aktiv
bGlobalPieceTick	Teilezähler, wird nach Übernahme zurückgesetzt
bGlobalLifeTick	Produktionsaktivität, wird nach Übernahme zurückgesetzt
bGlobalEmergencyStop	„Automatik aus“ wurde durch Not-Aus-Stopp aktiviert

Zeitdiagramm:



Die Produktionszeiten werden in Sekunden erfasst.

Der BDE-Eintrag ‚ProductionStatusMsg‘ erfolgt immer beim Wechsel des Produktionszustands. In dem Fall werden immer die erfassten Produktionsdaten des Vorzustands weggeschrieben.

Eine Fehlzeit entsteht immer dann wenn spätestens bis Zyklusende kein neuer Lifetick erkannt wurde. Die Fehlzeit wird nur innerhalb der Automatikzeit erfasst und dann als Summe mit der Automatikzeit weggeschrieben. Wenn aktiviert kann noch zwischen „Mikro-“ und „Makrostopps“ unterschieden werden.

Das Bit ‚bGlobalLifeTick‘ stellt die Möglichkeit bereit die Produktionszeit unabhängig vom Teilezähler (bGlobalPieceTick) zu starten. Für die Produktionszeit gilt ‚bGlobalLifeTick‘ OR ‚bGlobalPieceTick‘.

Die geschriebene Stoppzeit ist die Zeit, die zwischen dem Ende des vorherigen Automatiklaufs und dem Start des aktuellen Automatiklaufs verstrichen ist. Die Stoppzeit beim ersten Eintrag der Produktionszeiten wird vom Start der Maschine aus gemessen.

Die globale Produktionszeiterfassung läuft unabhängig von den anderen Meldungen. Die Anzahl der Bearbeitungen wird nur gezählt, wenn diese über das Bit ‚.....msgType.tool.bToolCount‘ der BDE gemeldet werden. Ist das nicht der Fall wird ‚0‘ als Anzahl der Bearbeitungen gemeldet.

Über welchen Mechanismus die Teilezählung für die globale Produktion erfolgt, kann über den Konfigurationsparameter ‚ProductionConfig.iPieceCountCode‘ vorgegeben werden:

Modus	Zählauf
0	Die Anzahl der produzierten Teile wird über die Teileanzahl bei der Übergabe ‚BarReady‘ gezählt.
1	Die Anzahl der produzierten Teile wird über die Teileanzahl bei der Übergabe ‚PieceReady‘ gezählt.
2	Die Anzahl der produzierten Teile wird mit dem globalen Bit ‚bGlobalPieceTick‘ gezählt. In diesem Fall könnte das Bit parallel zu oder für das Bit ‚bGlobalLifeTick‘ übergeben werden.

Folgendes ist bei verschiedenen Teilezählmoden zu beachten:

Modus	Besonderheiten
0	Bei einem Wechsel des Profils, des Profiltyps oder des Auftrags wird automatisch ein Produktionswechsel über die Produktionsstatusmeldung (Change Production) abgesetzt.
1	Bei einem Wechsel des Profils, des Profiltyps oder des Auftrags wird automatisch ein Produktionswechsel über die Produktionsstatusmeldung (Change Production) abgesetzt.
	Die zusätzlichen Informationen für die Produktionsstatusmeldung (Profil, Profiltyp und Auftrag) werden der Stab- bzw. Teilefertigmeldung entnommen.
2	Ein Produktionswechsel anhand der Auftragsdaten wird nicht erkannt. Die Stückzahlen und Bearbeitungszahlen gelten immer für den erfassten Produktionszeitraum.

## Automatische Einträge:

Folgende Einträge werden automatisch erzeugt:

Meldung	Auslösung durch...
ShiftMsg	Schichtan- oder -abmeldung Wechsel der übergebenen Schichtnummer in ProductionConfig.ShiftNo oder Übergabe von Schichtnummer ,0' für Schichtabmeldung Der Eintrag erfolgt erst, wenn die Automatik beendet wurde oder nach Schichtabmeldung eine neue Schichtanmeldung erfolgt.
ProductionStatusMsg	„bGlobalAutomatic“, wenn Teile produziert wurden oder bGlobalPause

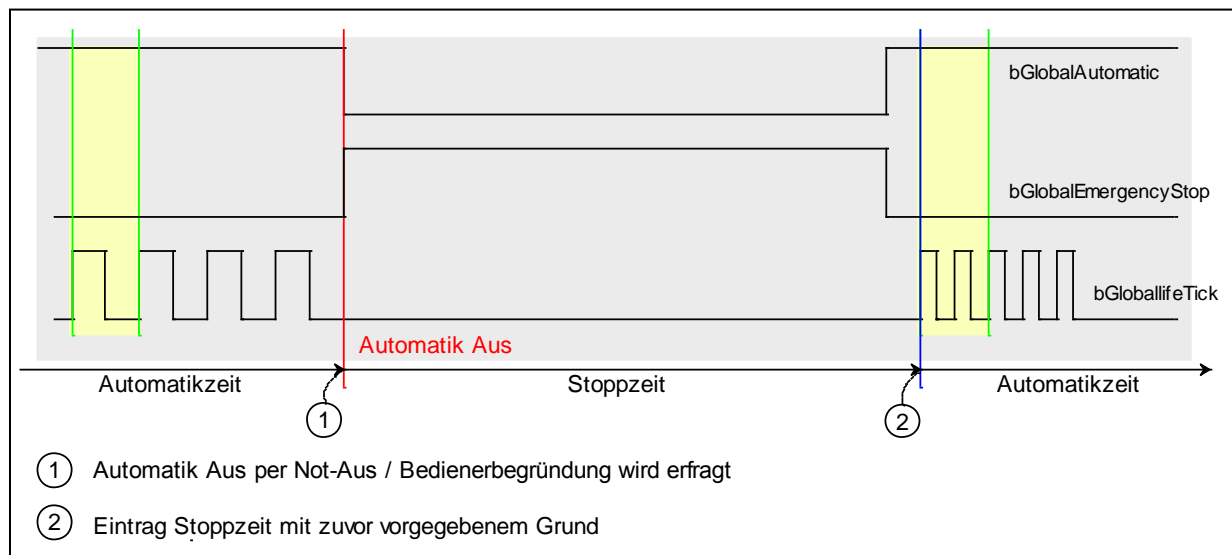
## Not-Aus-Begründung:

Wird die Automatikfunktion der Maschine über die Not-Aus-Funktion abgebrochen, kann die SPS durch Setzen eines Kennungsbits eine Bedienerbegründung für diesen Not-Aus anfordern. Es öffnet sich dann ein Fenster in dem die möglichen Gründe angezeigt. Der Bediener klickt dann zum Meldungsabschluss den aktuellen Grund an. Dieser wird dann übernommen und später als Stoppgrund in die Tabelle eingetragen.

```

TYPE STRUCT_ProductionState :
STRUCT
    bGlobalAutomatic:    BOOL;    (* pFlanke: Maschine im Automatikstatus *)
    bGlobalPause:        BOOL;    (* pFlanke: Maschine im Pausenstatus *)
    bGlobalPieceTick:    BOOL;    (* pFlanke: Life ticker mit Teilezählung *)
    bGlobalLifeTick:     BOOL;    (* pFlanke: Life ticker ohne Teilezählung *)
    bGlobalEmergencyStop: BOOL;    (* True: mit nFlanke an Automatik = Grundabfrage *)
END_STRUCT
END_TYPE

```



Das Signal „bGlobalEmergencyStop“ muss bei Rücknahme des Automatiksignals „bGlobalAutomatic“ entsprechend gesetzt sein.

Auswertung erfolgt nur bei negativer Flanke von „bGlobalAutomatic“ !

## Fertigmeldungen:

Fertigmeldungen können für Stäbe und Teile erfolgen. Wobei jede Fertigmeldeart getrennt zu sehen ist und Einträge in verschiedenen Tabellen erfolgen.

Aktivierungsstruktur:

```
TYPE STRUCT_PRODMSGTYPEN :  
STRUCT  
    bBarReady          :BOOL;  (* Stabfertigmeldung *)  
    bPieceReady        :BOOL;  (* Teilefertigmeldung *)  
    bBarStart          :BOOL;  (* Start: Stabbearbeitung *)  
    bPieceStart        :BOOL;  (* *** NOT IMPLEMENTED *** Start: Teilebearbeitung *)  
    bBarProdReset      :BOOL;  (* Reset: Stabproduktionsdaten *)  
    bPieceProdReset    :BOOL;  (* Reset: Teiledatenpuffer *)  
    bProdReset         :BOOL;  (* *** Old *** Please use bBarProdReset *)  
END_STRUCT  
END_TYPE
```

Datenübergaben:

```
TYPE STRUCT_PRODDATA :  
STRUCT  
    Job                :STRING;  (* LosName *)  
    BarNo              :UINT;    (* StabNr *)  
    PieceNo            :UINT;    (* TeileNr / only writing with bPieceReady... *)  
    ProfileName        :STRING;  (* ProfilName *)  
    Color              :STRING;  (* Profilarbe *)  
    ProfileType        :STRING;  (* Profiletyp: String user defined in proflidata: FL=Flügel, RA=Rahmen *)  
    Commission         :STRING;  (* Kundendaten: Kommission ... *)  
    Designation        :STRING;  (* Kundendaten: Bezeichnung ... *)  
    BarLength          :LREAL;   (* Stabrohlänge *)  
    Pieces             :UDINT;   (* Teilezahl im Stab *)  
    PieceLength        :LREAL;   (* Summe Teilleängen im Stab / oder Teillelänge bei bPieceReady *)  
    Course             :INT;     (* Laufbahnnummer *)  
    Counter            :INT;     (* Eintragszähler für bBarReady *)  
    RestLength         :LREAL;   (* Restteillänge im Stab *)  
    WasteLength        :LREAL;   (* Abfalllänge im Stab *)  
    RestPieces         :UINT;    (* Anzahl Restteile *)  
    WastePieces        :UINT;    (* Anzahl Abfallstücke *)  
    BarMeasureLength   :LREAL;   (* Gemessene Stablänge *)  
    BarMeasureWidth    :LREAL;   (* Gemessene Stabbreite *)  
    TreatmentCounter   :UINT;    (* Anzahl Bearbeitungen im Stab / Teil *)  
    TreatmentTime      :UDINT;   (* Bearbeitungszeit Stab / Teil *)  
END_STRUCT  
END_TYPE
```

Meldung	Funktion
bBarStart	Übernahme des Zeitpunktes für den Bearbeitungsstart des Stabes. Zwingende Datenübergaben sind: Job & BarNo
bBarReady	Fertigmeldung für Stab mit Datenübergabe über msgTextProdData. Daten werden abgelegt in der Tabelle: BarReady
bPieceReady	Fertigmeldung für Teil mit Datenübergabe über msgTextProdData. Es brauchen aber nur die für ein Teil relevanten Daten übergeben werden. Standardmäßig wird die Teilezahl erst weggeschrieben wenn sich ein Datenwechsel in den Produktionsvorgaben zeigt, der Profiltyp wechselt oder die Automatik beendet wird. Es findet eine automatische Aufsummierung der Teileanzahl statt.  Für die Teilelaufzeit wird immer die Automatik- und Stoppzeit erfasst, die bis zum Wegschreiben der Teilemeldung erfasst wurden. Daten werden abgelegt in der Tabelle: PieceReady

## Beispiel-Meldungsaufrufe:

In den nachfolgenden Beispielen ist „prodLog“ die im SPS-Programm angelegte Instanz des BDE-ProductionLog-Funktionsbausteins

### 1. Einfache Meldung bei PVC-Puffer belegt

```
prodLog.ProductionLog.msgText.Duration := d_MsgPufferDuration; (* Zeitspanne Fehlermeldung *)
prodLog.ProductionLog.msgType.Plc. bNoMaterialPVCLoad :=TRUE; (* msgTyp: PVC-Beladung nicht belegt *)
prodLog.ProductionLog.bWriteMsg:=TRUE; (* Kennung: Meldung schreiben *)
prodLog(); (* Aufruf: Meldung übernehmen *)
```

Bei diesen Meldungen wird die Eintragszeit (MsgTime) über die Zeitspanne zurückgerechnet!  
 $\text{MsgTime} + \text{Zeitspanne} = \text{Ende der Fehlersituation}$

### 2. Einfache Fehlermeldung bei nicht Quittierung mit Textübergabe

```
prodLog.ProductionLog.msgText.MsgText := 'Error XY'; (* Infotext für Fehlersituation *)
prodLog.ProductionLog.msgLogInfo.Area := 'PBZ1';
prodLog.ProductionLog.msgText.Duration := d_MsgPufferDuration; (* Zeitspanne Fehlermeldung *)
prodLog.ProductionLog.msgType.Plc. bNoMachineErrorQuit :=TRUE; (* msgTyp: PVC-Beladung nicht belegt *)
prodLog.ProductionLog.bWriteMsg:=TRUE; (* Kennung: Meldung schreiben *)
prodLog(); (* Aufruf: Meldung übernehmen *)
```

### 3. Stabfertigmeldung

```
prodLog.ProductionLog.msgTextProdData.BarNo:=i_BarNo; (* Stabnummer *)
prodLog.ProductionLog.msgTextProdData.Job:=s_JobName; (* Auftragsbezeichnung *)
prodLog.ProductionLog.msgTextProdData.ProfileName:=s_ProfilName; (* Profilbezeichnung *)
prodLog.ProductionLog.msgTextProdData.Color :=s_ProfilColor; (* Profilfarbe *)
prodLog.ProductionLog.msgTextProdData.ProfileType := s_Profiltyp; (* Profiltyp *)
prodLog.ProductionLog.msgTextProdData.BarLength:=l_ProfilbarLength; (* Profistablänge *)
prodLog.ProductionLog.msgTextProdData.PieceLength:=l_PieceLength; (* hier ist gesamte Teilelänge im Stab gemeint *)
prodLog.ProductionLog.msgTextProdData.Pieces:= i_PieceCount; (* Anzahl Teile im Stab *)
prodLog.ProductionLog.msgTextProdData.TreatmentCounter := i_ToolCount; (* Anzahl Bearbeitungen im Stab *)
prodLog.ProductionLog.msgTextProdData.Counter:= 1;
```

(\* verbleibende Restlänge bewerten \*)

```
prodLog.ProductionLog.msgTextProdData.RestLength:=
    l_ProfilbarLength - prodLog.ProductionLog.msgTextProdData.PieceLength;
```

```
IF prodLog.ProductionLog.msgTextProdData.RestLength < 800 THEN
    prodLog.ProductionLog.msgTextProdData.WasteLength := prodLog.ProductionLog.msgTextProdData.RestLength;
    prodLog.ProductionLog.msgTextProdData.RestLength := 0;
ELSE
    prodLog.ProductionLog.msgTextProdData.WasteLength:=0;
END_IF
```

(\* Übergabe für Stabvermessungen, wenn an Maschine vorhanden (optional) \*)

```
prodLog.ProductionLog.msgTextProdData.BarMeasureHeight := 1.254;
prodLog.ProductionLog.msgTextProdData.BarMeasureWidth := -0.987;
prodLog.ProductionLog.msgTextProdData.BarMeasureData := 'Messdaten';
*)
```

```
prodLog.ProductionLog.msgType.production.bBarReady:=TRUE; (* msgTyp: Stabinformation *)
prodLog.ProductionLog.bWriteMsg:=TRUE; (* Kennung: Meldung schreiben *)
prodLog(); (* Aufruf: Meldung übernehmen *)
```

### 4. Werkzeugzählung

```
prodLog.ProductionLog.msgLogInfo.Area := 'PC2_SBZ1'; (* Kennung für Station *)
prodLog.ProductionLog.msgTextToolCount.ToolNo := 1120033; (* Werkzeugnummer *)
prodLog.ProductionLog.msgTextToolCount.TreatmentCounter := 3; (* Zählwert ausgeführter Bearbeitungen *)
prodLog.ProductionLog.msgTextToolCount.ToolDescription := 'ToolCount: 1120033 / 3'; (*Beschreibung *)
```

```
prodLog.ProductionLog.msgType.Tool.bToolCount :=TRUE; (* msgTyp: Werkzeugzählung *)
prodLog.ProductionLog.bWriteMsg:=TRUE; (* Kennung: Meldung schreiben *)
prodLog(); (* Aufruf: Meldung übernehmen *)
```



## 5. Stationstransportfehler / Stationsstillstand auf Grund eine Fehlers

```
(* 1. Aufruf: Fehlersituation kommt: Station kann Teil nicht austransportieren *)
prodLog.ProductionLog.msgType.Station.bStationError[0] := TRUE;      (* Meldebit für Station xy (Index [xy]) *)
prodLog.ProductionLog.msgType.Station.bErrorBegin := TRUE;          (* Meldezeit beginnt *)
prodLog.ProductionLog.bWriteMsg:=TRUE;                               (* Kennung: Meldung schreiben *)
prodLog();                                                           (* Aufruf: Meldung übernehmen *)

(* 2. Aufruf: Fehlersituation ist behoben: Station kann wieder produzieren *)
prodLog.ProductionLog.msgText.MsgText := 'Werkzeugbruch';           (* Infotext für Fehlersituation *)
prodLog.ProductionLog.msgLogInfo.Area := 'SBZ1';                    (* Kennung für Station *)
prodLog.ProductionLog.msgType.Station.bStationError[0] := TRUE;    (* Meldebit für Station xy (Index [xy]) *)
prodLog.ProductionLog.msgType.Station.bErrorEnd := TRUE;            (* Meldung endet / Standzeit wird berechnet *)
prodLog.ProductionLog.bWriteMsg:=TRUE;                               (* Kennung: Meldung schreiben *)
prodLog();                                                           (* Aufruf: Meldung übernehmen *)
```

Bei diesen Meldungen wird die Eintragszeit (MsgTime) über die erfasste Zeitspanne ebenfalls zurückgerechnet!

Es wird auch die Zykluszeit beachtet, Meldungen < der 1,5fachen Zykluszeit werden nicht als Fehlerstandzeitmeldung der Station abgesetzt.

Wenn man beim 2. Aufruf eine MsgText.Duration in Sekunden übergibt, kann auch eine variable Grenze für das Absetzen der Meldung genutzt werden.

## 6. Die BDE läuft auf einem Unterrechner mit Verbindung zum Hauptrechner der Maschine

Gefordert:

- Der Lifeticker der von den Modulen die durch den zweiten Rechner bedient werden müssen zum Hauptrechner weitergeleitet werden.
- Das Flag prodLog.ProductionConfig.plcOptions.bBDERunsOnSlave := **TRUE**; setzen!
- Der ProdStatServer muss als Copy-Server aktiviert werden, wobei der Zielpfad für die Copy-Funktion auf das ProductionLog-Verzeichnis des Hauptrechners verweist.

## 7. Aktualisierungsfunktion für Produktionszeiten aktivieren

Gefordert:

- Die Zeitvorgabe prodLog.ProductionConfig. tCycleTimeMsgRefresh := **#1m30s**; wie gewünscht setzen. (Sollte aber in Minutengröße sein!)

## Freigabe der Optionen:

Die Freigabe der Optionen erfolgt über Einträge in der Windows-Registry. Man kann die einzelnen Meldungsfunktionen abschalten, bei nicht aktiver Funktion kann der Meldetext von der SPS trotzdem abgesetzt werden, die Übernahme wird quittiert, nur das Wegschreiben in die XML-Datei erfolgt nicht.

Im Einzelnen kann man freigeben:

### - Betriebsmeldungen:

Das betrifft folgende Meldungstypen:

```
TYPE STRUCT_MSGTYPE :  
STRUCT  
    Plc          :STRUCT_PLCMMSGTYPEN;  
    User         :STRUCT_USERMSGTYPEN;  
    Maintenance  :STRUCT_MAINTENANCEMSGTYPEN;  
    Machine      :STRUCT_MACHINEMSGTYPEN;  
    Shift        :STRUCT_SHIFTMSGTYPEN;  
    Production   :STRUCT_PRODMSGTYPEN;  
    Tool         :STRUCT_TOOLMSGTYPEN;  
END_STRUCT  
END_TYPE
```

zusätzlich auch die Logbucheinträge

### - Produktionszahlen:

Das betrifft folgende Meldungstypen:

```
TYPE STRUCT_MSGTYPE :  
STRUCT  
    Plc          :STRUCT_PLCMMSGTYPEN;  
    User         :STRUCT_USERMSGTYPEN;  
    Maintenance  :STRUCT_MAINTENANCEMSGTYPEN;  
    Machine      :STRUCT_MACHINEMSGTYPEN;  
    Shift        :STRUCT_SHIFTMSGTYPEN;  
    Production   :STRUCT_PRODMSGTYPEN;  
    Tool         :STRUCT_TOOLMSGTYPEN;  
END_STRUCT  
END_TYPE
```

Man kann:

### - Werkzeugzählfunktion:

Das betrifft folgende Meldungstypen:

```
TYPE STRUCT_MSGTYPE :  
STRUCT  
    Plc          :STRUCT_PLCMMSGTYPEN;  
    User         :STRUCT_USERMSGTYPEN;  
    Maintenance  :STRUCT_MAINTENANCEMSGTYPEN;  
    Machine      :STRUCT_MACHINEMSGTYPEN;  
    Shift        :STRUCT_SHIFTMSGTYPEN;  
    Production   :STRUCT_PRODMSGTYPEN;  
    Tool         :STRUCT_TOOLMSGTYPEN;  
END_STRUCT  
END_TYPE
```

- Produktionsstatus:

Das betrifft folgende Funktionen:

```
TYPE STRUCT_ProductionState :  
STRUCT  
    bGlobalAutomatic:    BOOL;  
    bGlobalPause:        BOOL;  
    bGlobalPieceTick:    BOOL;  
    bGlobalLifeTick:     BOOL;  
    bGlobalEmergencyStop: BOOL;  
END_STRUCT  
END_TYPE
```

---

## Globale Variablen der Bibliothek:

Als globale Variable steht für eine Instanz der Bibliothek die globale Variable „ProductionConfig“ vom Typ „STRUCT\_ProductionLogConfigFromBDA“ zur Verfügung. Über diese Variable ist es auch möglich Konfigurationsvariablen zu setzen, wenn dieses von extern erfolgen soll.

Der Datentyp ist folgendermaßen definiert:

```
TYPE STRUCT_ProductionLogConfigFromBDA :
STRUCT
  (* BDE > LIB *)
  bLogOnOff:      BOOL;          (* Logging On/Off / Handshake Var *)
  UserName:       STRING;        (* current default User *)
  ShiftNo:        INT;           (* current default ShiftNo *)
  Machine:        STRING;        (* current default Machinenname *)
  Area:           STRING;        (* current default Area *)
  xmlPath:        STRING;        (* XML-Destination Path *)

  ApplicationStart: BOOL;        (* Eintrag Applikationsstart *)
  ApplicationStop:  BOOL;        (* Eintrag Applikationsstopp *)
  ShutDown:        BOOL;        (* Rechnsr ShutDown *)

  CloseActEntrys:  BOOL;        (* Abschluss aller anstehenden Meldungen *)

  UserReason:      STRUCT_UserReason;
  SystemReason:    STRUCT_SystemReason;

  PauseStart:      BOOL;        (* Start einfache Pause *)
  PauseStop:       BOOL;        (* Stop einfache Pause *)

  ReasonEnd:       BOOL;        (* Begründete Pause beenden, neue beginnen *)

  ReasonFaultText:  STRING;
  ReasonFaultStartTime: STRING;
  ReasonFaultDuration: UDINT;

  msgTexte:        ARRAY [0..MAX_PRODLOG_MSGTEXTE] OF STRING;

  (* LIB > BDA *)
  Pause:          BOOL;          (* TRUE: allgemeine Pause ist in BDE aktiv *)
  LifeState:      BOOL;          (* TRUE: Lifeticker von SPS (Weiterleitung= *)
  InfoState:      STRUCT_INFOSTATE; (* Information zum aktuellen Status der Maschine *)

END_STRUCT
END_TYPE
```

Es ist aber zu beachten, dass standardmäßig die Übergabeschnittstelle durch die laufende Maschinenapplikation belegt ist und somit nicht für den allgemeinen Gebrauch bestimmt ist. Eine Ausnahme stellen die Variablen nach dem Kommentar (\* LIB > BDA \*) dar. Diese können von extern ausgelesen werden.

Die Variable „Pause“ ist als boolsche Variable definiert und kennzeichnet im True-Zustand, dass der Bediener in die Pause gegangen ist. Dies bedeutet, aber nicht, dass die Maschine sich im Stillstand befindet, sondern nur das der Bediener nicht Vorort ist. Die Maschine ist in dem Zustand wie sie verlassen wurde.

Die Variable „LifeState“ ist als boolsche Variable definiert und kennzeichnet im True-Zustand den Aktivzustand der Maschine. Das heißt die Maschine ist in „Automatik“ und bearbeitet Stäbe oder Teile. Der False-Zustand entsteht dann, wenn die Maschine im Automatikmodus nicht produziert, oder die Automatik ausgeschaltet ist.

Die Variable „InfoState“ vom Typ „STRUCT\_INFOSTATE“ stellt eine zyklisch geschriebene Variable zur Verfügung über die folgenden Daten ausgelesen werden können:

```

TYPE STRUCT_INFSTATE :
STRUCT
  bBdeFbActive      : BOOL;          (* Gesetz wenn SPS Baustein initialisiert *)
  bTrig             : BOOL;          (* Trigger Statusmeldung abgesetzt (toggelt True <> False) *)
  bLifeState        : BOOL;          (* Weiterleitung des Lifeticksignals *)
  iRunningMode      : DINT;          (* aktiver Modus oder zu aktivierender Modus *)
  iStateCode        : DINT;          (* Status des aktiven Modus *)
  sStateText        : STRING(255);   (* Klartext zum Status *)
  iErrorCode        : UINT;          (* Melde-ID (nur im Fehlerfall) *)
  sMachineName      : STRING;        (* Bezeichnung / Nummer der Maschine *)
  iBarCount         : UDINT;         (* Zählwert Stäbe *)
  iPieceCount       : UDINT;         (* Zählwert Teile *)
  iReCutBarCount    : UDINT;         (* Zählwert Nachfertigungsstäbe *)
  iReCutCount       : UDINT;         (* Zählwert Nachfertigungsteile *)
END_STRUCT
END_TYPE

```

Strukturelemente:

bBdeFbActive	Bool	True, wenn BDE-Funktion auf Maschine aktiviert
bTrig	Bool	Diese Variable ändert zyklisch ihren Zustand und dient nur zum triggern der Lesefunktion auf der Gegenseite.
iRunningMode	DINT	Aktiver Produktionsmodus der Maschine 0 = Maschine ist nicht im Produktionsmodus 1 = Maschine ist im Produktionsmodus
iStateCode	DINT	Status des aktiven Modus 0 = Alles Okay, Maschine ist in Produktion 1 = Automatik aktiv aber keine Produktion 2 = Bediener hat Pause aktiviert
sStateText	String	Hier wird ein Klartext zum Status der Maschine geschrieben Der Text ist also abhängig vom Statuscode: 0: leer 1: Meldetext der durch den Bediener festgelegt wurde, solange kein Text gewählt wurde, wird Standardtext ausgegeben. 2: Text = Pause
sMachineName	String	Hier wird eine Maschinebezeichnung zurückgegeben
iBarCount	UDINT	Anzahl der per BDE-Funktion erfassten Stäbe
iPieceCount	UDINT	Anzahl der produzierten Teile
iReCutBarCount	UDINT	Anzahl Stäbe für Nachfertigungen die per BDE erfasst wurden
iReCutCount	UDINT	Anzahl nachgefertigter Teile

## BDA-Oberfläche:

Die BDA-Applikation muss immer auf der Maschine installiert sein!

Folgende Funktionen sind darin integriert:

- Auswahlfenster für Begründungen von Fehlersituationen
- Vorgabe von Bedienermeldungen die den Maschinenzustand beschreiben
- Wartungsdatenanzeigen / Wartungstabellen
- Auswertung BDE in Tabellen
- Anzeige aktueller Übersicht über Produktionsdaten
- Pausenanwahl, -abwahl

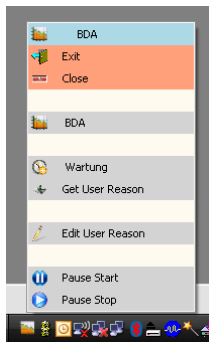
Die BDA-Applikation muss im Programmverzeichnis der Maschinenapplikation gespeichert sein und mit der Applikation nach TwinCat-Start gestartet werden.

Die BDA-Applikation klinkt sich mit ihren Icon in der Notification Area des TaskBars ein:



Funktionen über Kontextmenü:

Mit dem Klick auf das Icon öffnet sich das Kontextmenü der BDA-Applikation:



Funktion	Beschreibung
BDA	Kopfzeile des BDA-Kontextmenüs
Exit	Beenden der BDA-Applikation
Close	Schließen des Kontextmenüs
BDA	Aufruf der BDA-Applikationsoberfläche
Wartung	Aufruf des Wartungsmenüs mit Anzeige der aktuellen Zählstände
Get User Reason	Aufruf der Bedienerbegründungen zum Absetzen einer Meldung
Edit User Reason	Öffnen des Bearbeitungsmenüs der Bedienerbegründungen (wird nur mit gedrückt gehaltener Strg-Taste angezeigt)
Pause Start	Start einer allgemeinen Pause an der Maschine
Pause Stop	Beenden der zuvor gestarteten Pause mit BDE-Eintrag

Ressourceneinstellungen in BDA-Settings:

Parameter	Beschreibung	Wert
Teile-Taktzyklus (sec)	Defaultwert Zykluszeit für Maschinentakt in Sekunden, wenn der Maschine kein spezieller Takt über die Maschineigenschaften zu geordnet	25
ChartAxisMaxAutomatik (%)	Maximalwert für die Höhe des Produktivitätsbalken im Chart	120
Where Clause PLCMsg	Datenbankabfrage für anzuzeigende SPS-Meldungen	MsgId > 3
ShowUserFaultMsg	Anzeige von Bedienerbegründungen in der Produktionszeit-Tabelle.	1
MachineVersion	Aktivierung der Maschinenversion mit allen Funktionen	1
BDAOnOff	Aktivierung der Übergabe von Konfigurationsdaten	1

	aus der BDA an die BDE in der SPS.	
BDAMachineName	Maschinenname für BDE-Einträge. Wird nur übergeben wenn BDAOnOff = 1	
BDAAreaName	Maschinenbereichsname für BDE-Einträge. Wird nur übergeben wenn BDAOnOff = 1	
ReasonAnzeigeMaxSpalten	Spaltenzahl für Auswahlmenü der Fehlerbegründungen	2
ReasonAnzeigeMaxZeilen	Zeilenzahl für Auswahlmenü der Fehlerbegründungen	10
ReasonAnzeigeFontSize	Schriftgröße für Text die im Auswahlmenü der Fehlergründe angezeigt werden	18
WartungsMsgIntervall (Min)	Intervall zur Anzeige anstehender Wartungen in Minuten	30
LetzterWartungsTest		
TopMostReason	Aktiviert, dass die Begründungsabfrage bis zur Bestätigung in den Vordergrund gebracht wird.	1
CompletionTimeX	Zeitpunkte an dem aktuell erfasste Betriebsdaten gezielt abgeschlossen werden sollen. Die Vorgabe muss im Zeitformat (hh:mm) erfolgen. Die leere Vorgabe deaktiviert die Funktion.	
TimeShiftX	Fixe Startzeit des Schichtbeginns. Die Vorgabe muss im Zeitformat (hh:mm) erfolgen. Die leere Vorgabe deaktiviert die Funktion. Bei der Vorgabe muss die zeitliche Reihenfolge von Schicht 1 – 5 eingehalten werden.	
PausenStartzeitX#Dauer (Min)	Fixe Startzeit einer Pause und ihre Dauer. Die Vorgabe muss im Zeitformat (hh:mm) erfolgen. Die Dauer muss mit Raute getrennt in Minuten übergeben werden. Die leere Vorgabe deaktiviert die Funktion.	09:00#15

## BDA-Server:

Der Produktions-Status-Meldungsserver dient zur Übernahme der BDE-Meldungen in die BDE-Datenbank auf der Maschine oder zur Übermittlung der Daten an eine externe BDE.

Parameter	Beschreibung
Server Path	Geben Sie bitte in diesem Eingabefeld den vollständigen Pfad vor, in dem Betriebsmeldungen von der SPS übergeben werden. Standardmäßig ist hier das Verzeichnis „..\ProductionLog“ im Applikationsverzeichnis vorzugeben. Es kann aber auch jedes andere Verzeichnis gewählt werden. Dabei muss nur sichergestellt sein, dass das Verzeichnis vorhanden ist und die Daten dort von der SPS hinterlegt werden.
Anlagen Path	In der Eingabezeile ‚Anlagen Path‘ müssen Sie den Pfad auf das System-Verzeichnis der Applikation vorgeben. Der Server greift über diese Pfadvorgabe auf die Datenbank ‚ProductionLog‘ zu und legt darin die Betriebsmeldungen ab. Existiert die Datenbank nicht, wird diese vom Server automatisch in diesem Verzeichnis erzeugt.
	Beide Vorgaben müssen gesetzt sein, auch wenn der Server nur als Copy-Server arbeitet.
Copy Path	Hier können Sie einen Zielpfad und Verzeichnis vorgeben, in das die XML-Dateien kopiert werden, wenn die Auswertung extern erfolgt. Diese Funktion ist nur wirksam, wenn der Schalter ‚Copy-Server‘ gesetzt ist.
Connection String	Hier kann eine spezielle Verbindungszeichenfolge vorgegeben werden, wenn z.B. mit einer SQL-Datenbank gearbeitet wird. Die Schaltfläche rechts daneben ruft eine Eingabehilfe zur Erstellung der Zeichenfolge auf. Wird mit der lokalen ProductionLog.mdb gearbeitet, braucht hier kein Eintrag erfolgen!
Copy-Server	<input type="checkbox"/> Der Server arbeitet auf die maschineninterne BDE-Datenbank. Der Verzeichnispfad zur Datenbank muss unter ‚Anlagen Path:‘ vorgegeben werden.
	<input checked="" type="checkbox"/> Der Server arbeitet im Copy-Mode und kopiert die XML-Meldungsdateien der BDE in das unter ‚Copy Path:‘ vorgegebene Verzeichnis. Es kann sich dabei auch um ein im Netzwerk freigegebenes Verzeichnis handeln. Einträge in die maschineninterne Datenbank erfolgen nicht.
Yearly Backup	<input type="checkbox"/> Eine jährliche Sicherung der Betriebsdaten wird nicht automatisch erstellt. Nutzen Sie die Menüfunktion „Backup“.
	<input checked="" type="checkbox"/> Der Server erstellt automatisch bei Erkennung des Jahreswechsels eine Datensicherung der Betriebsdaten. Die Datensicherung befindet sich dann im Unterverzeichnis „OldProductionLog“, das sich wiederum im Verzeichnis der Betriebsdatendatei befindet. In den Dateinamen ist das Datum „Von-Bis“ eingefügt
Deutsches Datumsformat	<input type="checkbox"/> Es wird mit dem englischen Datumsformat gearbeitet. (>SQL-Datenbank)
	<input checked="" type="checkbox"/> Es wird mit dem deutschen Datumsformat gearbeitet. (>SQL-Datenbank)



Delete double Records	<input type="checkbox"/>	Funktionalität ist nicht aktiv
	<input checked="" type="checkbox"/>	Einträge mit gleichen Datum & Uhrzeit werden automatisch zusammen gefasst
History	<input type="checkbox"/>	Die Meldungshistorie wird nicht angezeigt.
	<input checked="" type="checkbox"/>	Unterhalb der Eingabefelder wird Ihnen eine Historie der vom Server erzeugen Meldungen angezeigt.
RefreshRate (ms)	Hier kann das Intervall in Millisekunden festgelegt werden in dem das Vorhandensein neuer Betriebsmeldungen geprüft werden soll.	

### BDA-Startup:

Zur Wartung der ProductionLog.mdb muss auf der Maschine das Programm „projAutoStart.exe“ aktiviert sein. Dort muss beim Löschdatum mindesten ein Eintrag gemacht sein (30#5).  
Im Startmoment des Rechners komprimiert das Programm die BDE-Datenbank und bereinigt sie zu dem.

Programmversion : 1.0.0 vom 10.11.2014 oder jünger