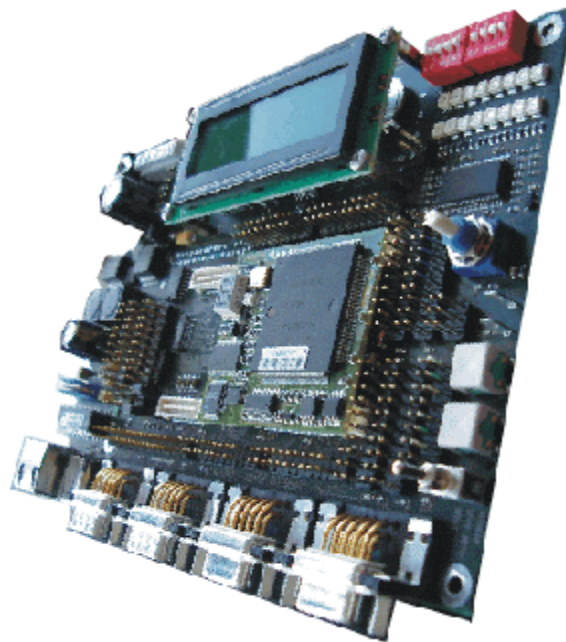




für Starterkit STK16XU



**Installationshinweise und erste Schritte der Programmierung des
STK16XU mit der grafischen Programmieroberfläche iCon-L**

Inhalt

1.	Allgemeines	3
2.	Voraussetzungen.....	3
3.	Installation der Programmieroberfläche iCon-L.....	3
4.	Firmaredownload	4
4.1	Ist die Firmware schon aktiv ?	4
4.1.1	Projekt laden	4
4.1.2	Herstellen der physischen Verbindung.....	4
4.1.3	Wechseln in die Inbetriebnahme und STK16XU anmelden	4
4.2	Was tun wenn die Firmware nicht läuft ?	5
5.	Programmieren mit iCon-L	7
5.1	Aufgabenstellung	7
5.2	Ein neues Projekt erstellen	7
5.3	Aufgabenstellung grafisch umsetzen	8
5.3.1	Digitale Eingänge einfügen.....	8
5.3.2	Digitalen Ausgang einfügen.....	9
5.3.3	UND Baustein einfügen	9
5.3.4	Baustein verbinden	9
5.4	Programm in die Steuerung (Starterkit) laden	10
6.	Bereitstellung der Konfiguration für das Starterkit STK16XU.....	12
6.1	BasicConfig.ini	12
6.1.1	Generelle Konfiguration	12
6.1.2	Portkonfigurationen.....	12
6.1.3	Registerinitialisierungen.....	12
6.2	PortConfig.ini.....	13
6.2.1	generelle Portkonfiguration.....	13
6.2.2	Beispiel einer Bit-Kanal Konfiguration (Ausgang).....	13
6.2.3	Beispiel einer Bit-Kanal Konfiguration (Eingang).....	14
6.2.4	Beispiel einer Analog-Kanal Konfiguration (Eingang).....	14
7.	Zusammenfassung und erweitertes Beispiel.....	15

1. Allgemeines

Diese Anleitung erläutert die ersten Schritte, um das Starterkit STK16XU der Firma TQ-Components mit dem grafischen Programmiersystem iCon-L in Betrieb zu nehmen.

Der Leser ist nach dem Durcharbeiten dieser Anleitung in der Lage die konfigurierten Ein- und Ausgänge bzw. das Display eines STK16XU unter Verwendung von iCon-L anzusprechen bzw. diese zu programmieren. Diese Anleitung erklärt den Umgang mit dem Programmiersystem iCon-L nur sehr eingeschränkt. Informationen zu iCon-L und dessen Bedienung entnehmen Sie bitte der Onlinehilfe von iCon-L.

Alle geschützten Markennamen sind Eigentum ihrer jeweiligen Besitzer.

2. Voraussetzungen

Um ein STK16XU mit iCon-L in Betrieb zu nehmen benötigt man:

Mindestens:

- Marktüblicher PC mit Windows – Betriebssystem (WIN9x/NT/2000/XP)
- 80 MB Platz auf der Festplatte
- eine freie RS232 –Schnittstelle
- iCon-L CD mit DevicePack TQ-Components STK16XU
- Nullmodemkabel
- STK16XU und notwendige Komponenten

3. Installation der Programmieroberfläche iCon-L

1. Legen sie die beigefügte CD ein. Wenn kein Autostart erfolgt starten sie bitte die Installation durch Doppelklick mit der Maus auf der iConL4.exe im Rootverzeichnis der CD. Wählen sie in dem darauf erscheinenden Bild die Sprache der Installation durch einfachen Klick auf die entsprechende Flagge.



2. Wählen sie in dem darauf folgendem Dialog den Menüpunkt zur Installation der entsprechenden iCon-L Installation. Sie werden in gewohnter Weise durch die Installation geführt.
3. Nach erfolgter Installation und Schließen des Browsers kann iCon-L gestartet werden.

4. Firmwaredownload

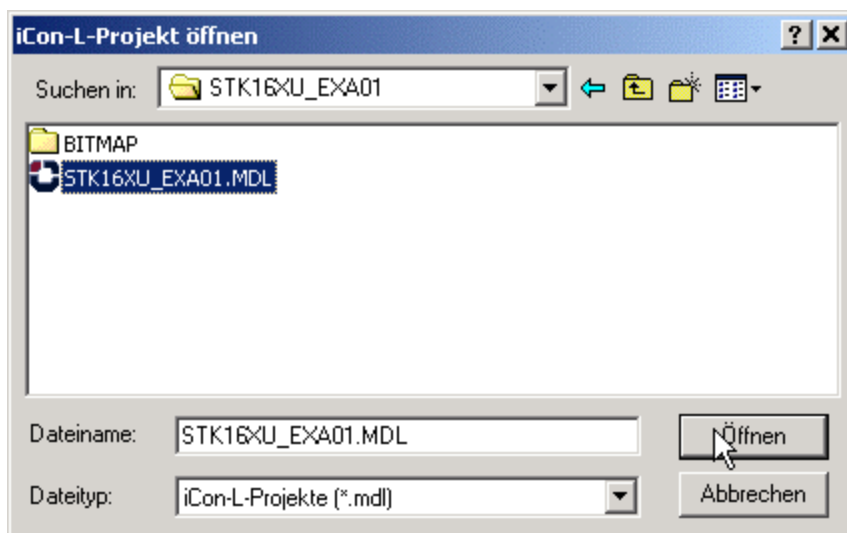
Um das STK16XU mit iCon-L programmieren zu können muss (falls dies noch nicht geschehen ist) die mitgelieferte Firmware in das STK16XU geladen werden.

4.1 Ist die Firmware schon aktiv ?

Diese Frage kann durch den Versuch der Verbindungsaufnahme mit dem STK16XU beantwortet werden. Benutzen sie dazu das in der Installation mitgelieferte iCon-L Projekt **STK16XU_BASIC**.

4.1.1 Projekt laden

1. Starten sie iCon-L.
2. Wählen sie im Menü **Projekt** den Punkt **Öffnen**.
3. Wählen sie im Dateidialog die Datei **Project German\STK16XU_EXA01\STK16XU_EXA01.MDL**



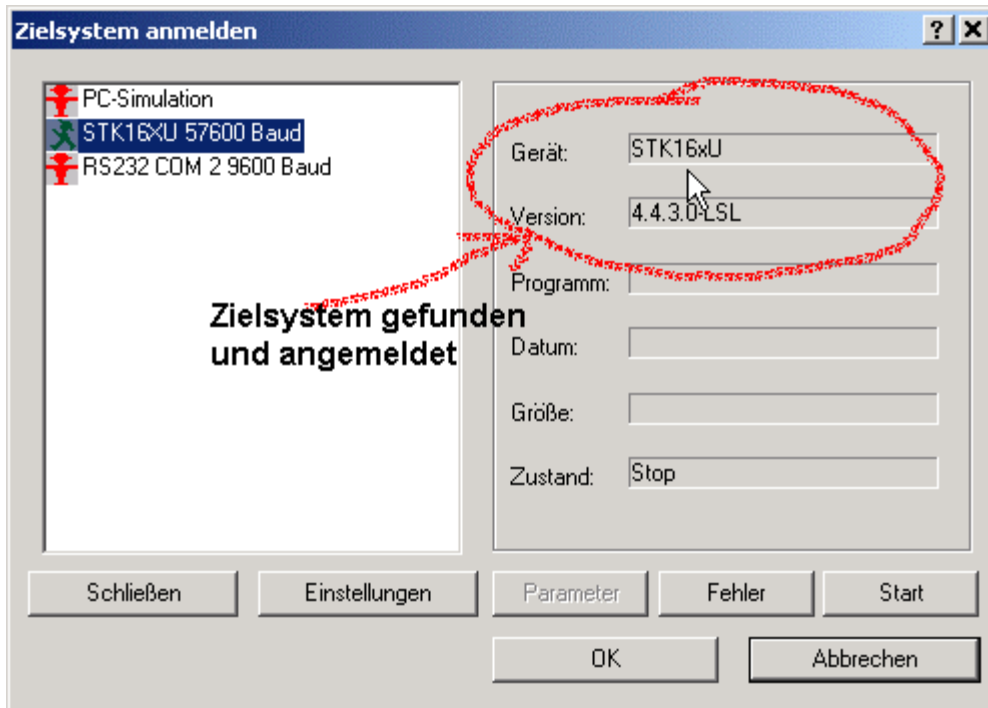
4.1.2 Herstellen der physischen Verbindung

Verbinden sie unter Verwendung des RS232-Kabels ihren PC (default COM1) mit ihrem Starterkit. Stellen sie sicher, dass die Spannungsversorgung des Starterkits arbeitet.

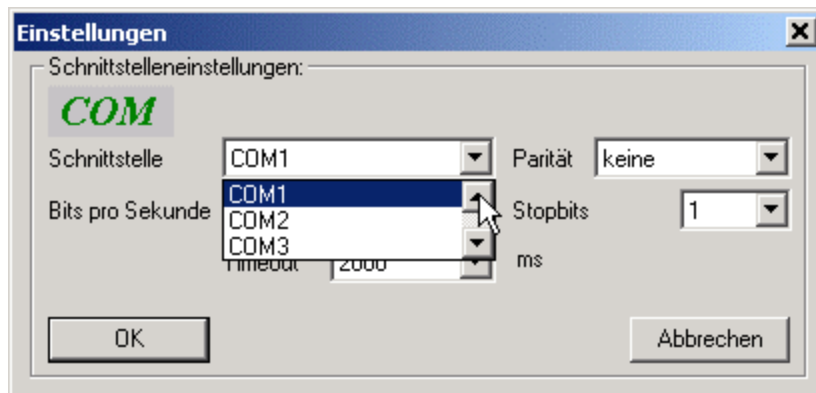
4.1.3 Wechseln in die Inbetriebnahme und STK16XU anmelden

1. Wählen sie im Menü **Inbetriebnahme** den Punkt **Eintritt**.
2. Wählen sie im Menü **Inbetriebnahme** den Punkt **Zielsystem anmelden**.
3. Wählen sie in dem daraufhin erscheinenden Dialog den Eintrag **STK16XU 57600 Baud**.

Wenn eine Verbindung aufgebaut werden konnte wird dies durch Anzeige der Informationen zu dem entsprechenden Zielsystem im rechten Teil des Dialogs angezeigt.



Ist dies nicht der Fall prüfen sie bitte die physische Verbindung bzw. ändern sie die Parameter der Kommunikation, wenn sie z.B. eine andere COM-Schnittstelle verwenden oder Erhöhen sie das Timeout auf 2000 ms bzw. größer. Dazu betätigen sie bitte den Schalter **Einstellungen**.

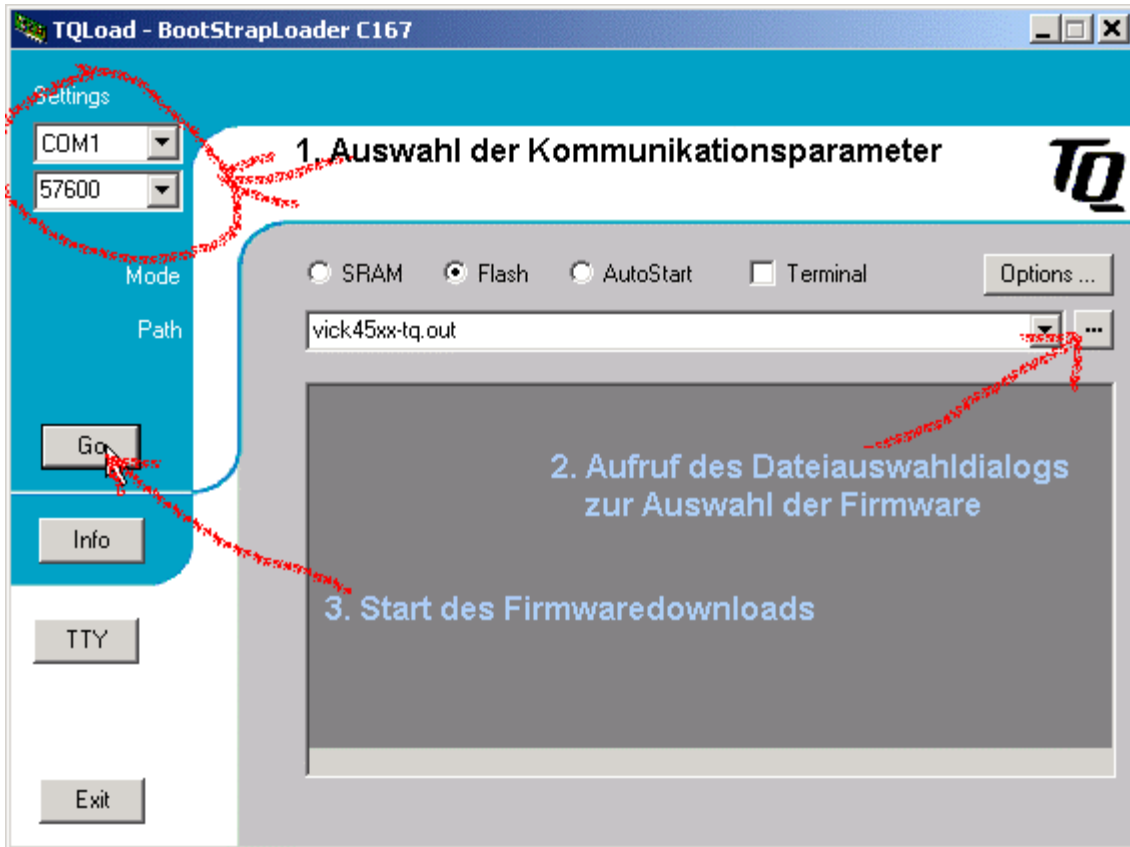


4.2 Was tun wenn die Firmware nicht läuft ?

Über das Menü „Extras“ könne sie einen Firmwareloader aufrufen. Um diesen zu aktivieren gehen sie bitte vor wie im Punkt 4.1.1 und 4.1.2 beschrieben.

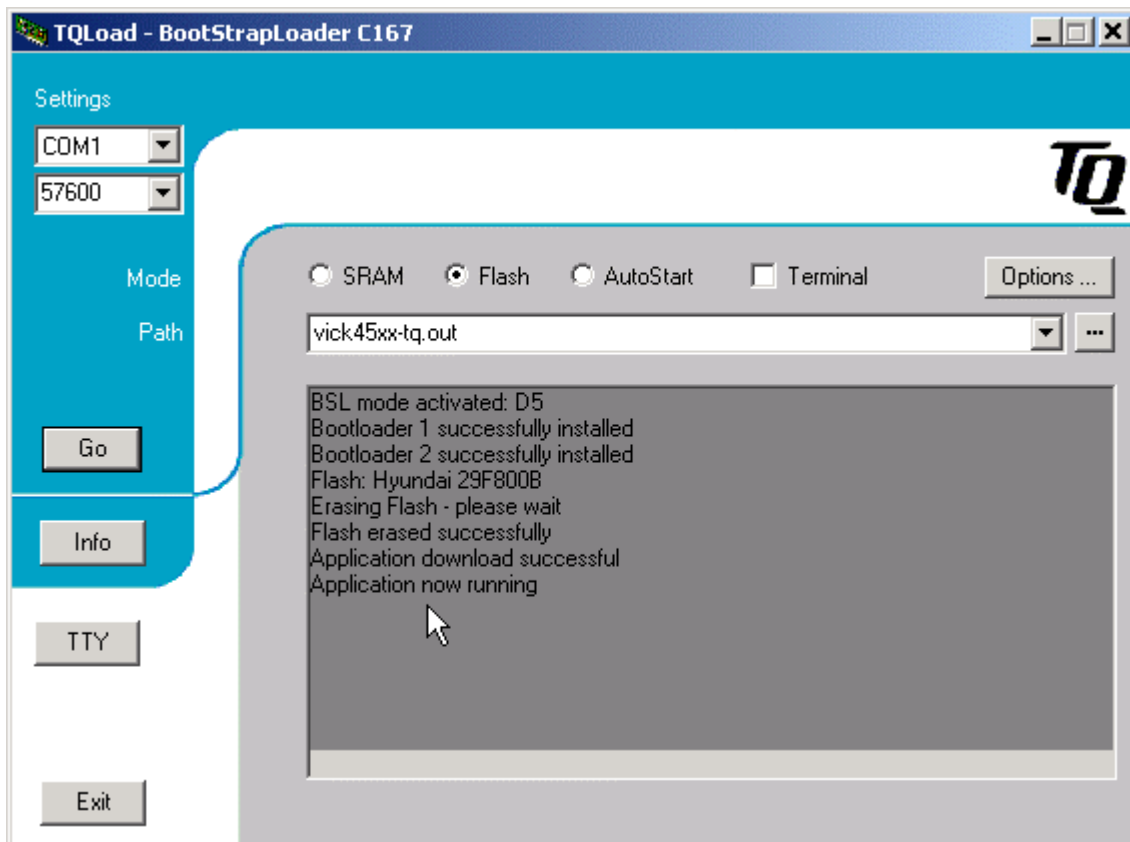
Jetzt können sie die folgenden Punkte abarbeiten:

1. Wählen sie im Menü **TQ / Extras** den Punkt **TQ Firmwareloader**.
2. Als Default für die Kommunikation sind eingestellt
COM1
57600 Baud
Ändern sie diese, wenn sie z.B. eine andere COM - Schnittstelle benutzen.
3. Wählen sie die entsprechende Datei zum Downloaden. Diese finden sie im Verzeichnis Firmware der aktuellen iCon-L Installation.



Starten sie jetzt den Firmwaredownload durch Klick mit der linken Maustaste auf dem Schalter **Go**.

Der Flash – Vorgang wird im Messagefenster des Dialogs entsprechend dokumentiert. War der Firmwaredownload erfolgreich, dann sollten die Meldungen wie im nachfolgend dargestellten Bild lauten.

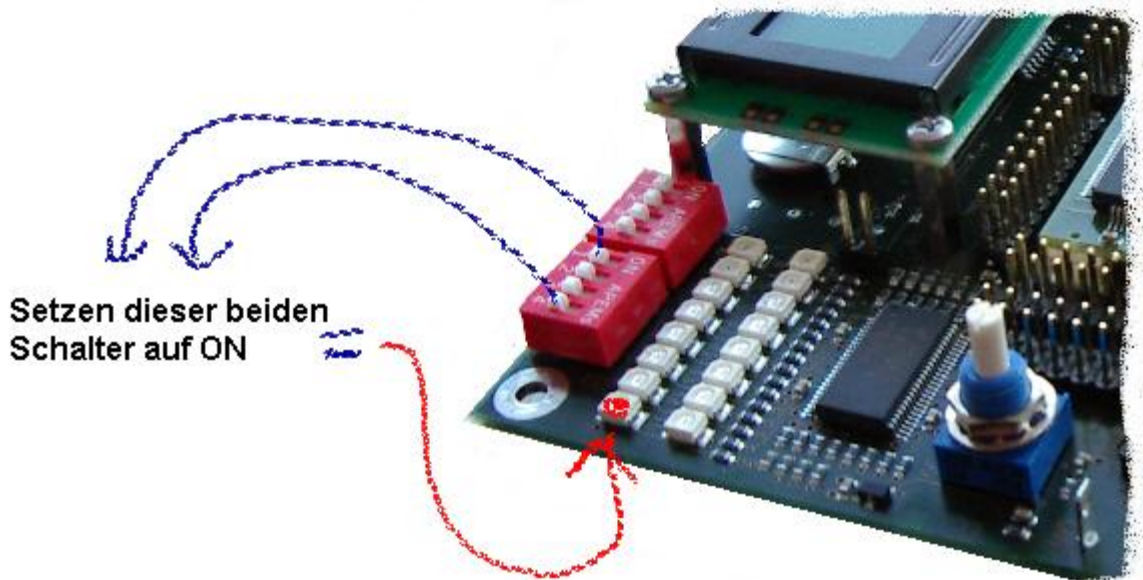


5. Programmieren mit iCon-L

Wenn der iCon-L Laufzeitkern (Firmware) auf dem Starterkit läuft kann mit iCon-L auf die entsprechenden IO-s zugegriffen werden. Um, wenn auch nur ansatzweise, die Vorteile der grafischen Programmierung mit iCon-L darzustellen, wird hier das Erstellen eines Minimalprogramms erklärt.

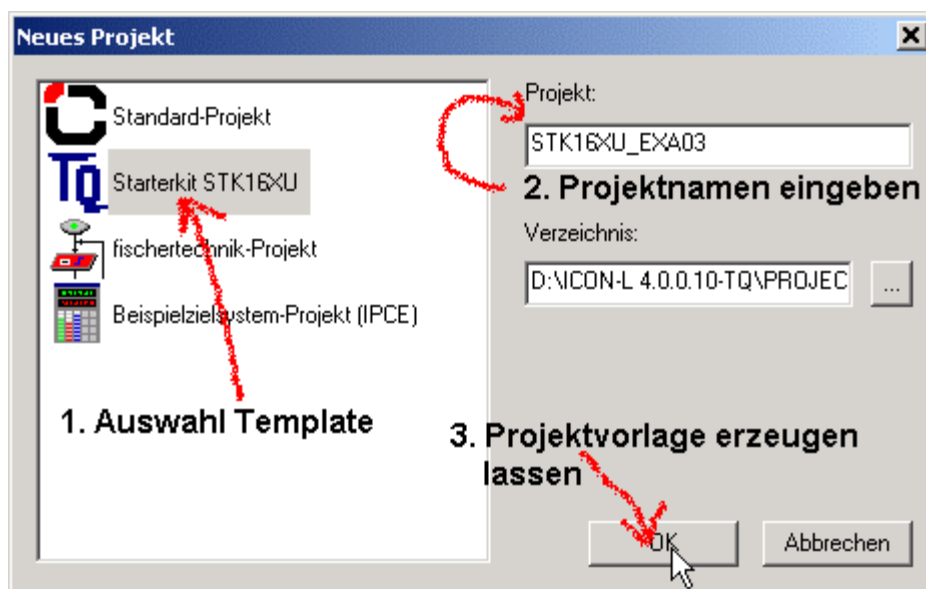
5.1 Aufgabenstellung

Das Setzen der DIP-Schalter 1 und 4 des zweiten Blocks soll die 8. LED schalten.



5.2 Ein neues Projekt erstellen

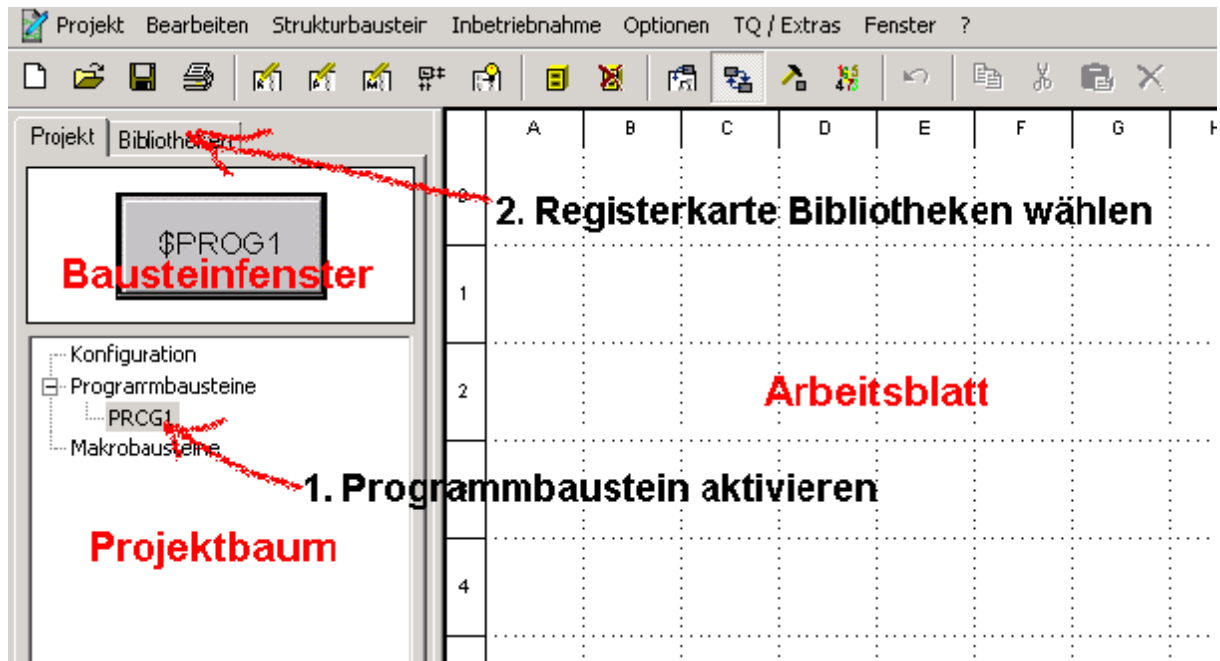
Starten sie zuerst iCon-L. Wählen sie im Menü **Projekt** den Punkt **Neu**. In dem darauf folgenden Dialog markieren sie das Template zur Erzeugung einer Projektvorlage für das Starterkit STK16XU und geben danach den Namen des Projekts (in unserem Beispiel STK16XU_EXA03) im Eingabefeld Projekt ein.



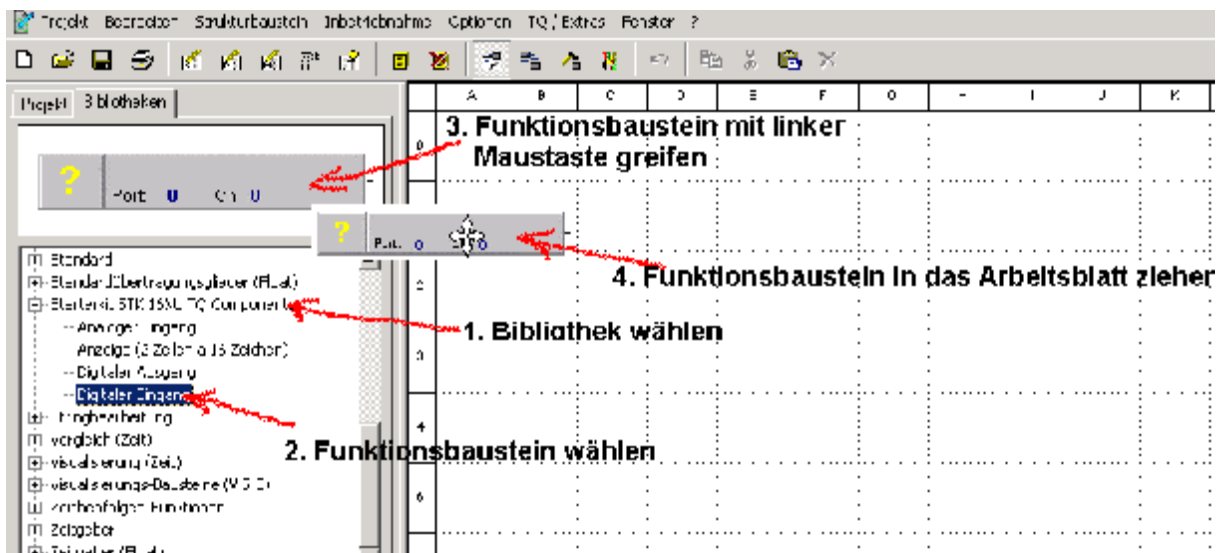
Nach Betätigung des Schalters OK wird ein neues Projekt erstellt, welches die Spezifika des Starterkits berücksichtigt (Bibliotheken, Konfigurationsdateien usw.).

5.3 Aufgabenstellung grafisch umsetzen

Nach dem das neue Projekt erstellt wurde können sie nun die notwendigen Funktionsbausteine platzieren und verknüpfen. Aktivieren sie dazu den Programmbaustein 1 durch Doppelklick mit der Maus auf dem Eintrag PROG1 im Projektbaum.



Aktivieren sie nun die Bibliotheks- und Bausteinanzeige durch Klick mit der Maus auf der Registerkarte Bibliotheken.



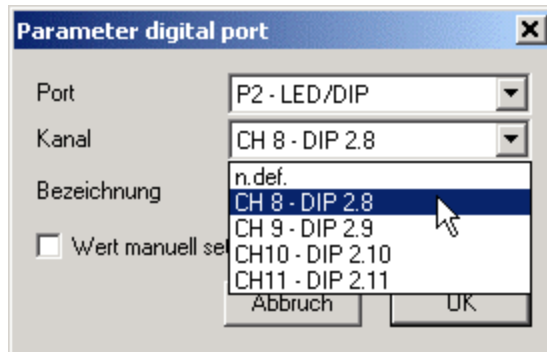
5.3.1 Digitale Eingänge einfügen

Gehen sie zum Einfügen von Funktionsbausteinen nach folgendem Schema vor:

1. wählen sie die Bibliothek im Projektbaum und öffnen sie diese (Doppelklick linke Maustaste) hier **Starterkit STK16XU TQ Components**
2. markieren sie den gewünschten Funktionsbaustein (einfacher Klick linke Maustaste) hier zuerst **Digitaler Eingang**
3. bewegen sie die Maus in das Bausteinfenster, in dem der markierte Baustein abgebildet wird, über diesen und halten sie nun die linke Maustaste gedrückt.
4. bewegen sie die Maus zusammen mit dem Baustein in das Arbeitsblatt an die gewünschte Position und lassen jetzt die linke Maustaste los.

Bei Funktionsbausteinen die parametrieren wird an dieser Stelle der Parametrierdialog aufgerufen.

In unserem Beispiel müssen sie den Port und den Kanal auswählen. Wie die Daten der verfügbaren Ports und Kanäle zur Verfügung gestellt werden wird im **Punkt 6** dieser Kurzanleitung beschrieben. Um die Aufgabenstellung umsetzen zu können wählen sie im Parametrierdialog für den Port **P2 – LED /DIP** und für den Kanal **CH 8 – DIP 2.8** .



Wiederholen sie den Vorgang des Einfügens eines Funktionsbausteins vom gleichen Typ und geben sie als Parameter für den Port **P2 – LED /DIP** und für den Kanal **CH 8 – DIP 2.11** vor.

Diese Parameter können nachträglich auch noch geändert werden, wenn sie über dem Baustein die **rechte Maustaste** drücken und den entsprechenden Menüpunkt auswählen. Hier finden sie auch eine Möglichkeit sich eine Hilfe zu dem entsprechenden Baustein anzeigen zu lassen.

5.3.2 Digitalen Ausgang einfügen

Gehen sie vor wie beim Digitalen Eingang nur dass sie nun den Funktionsbaustein Digitaler Ausgang wählen.

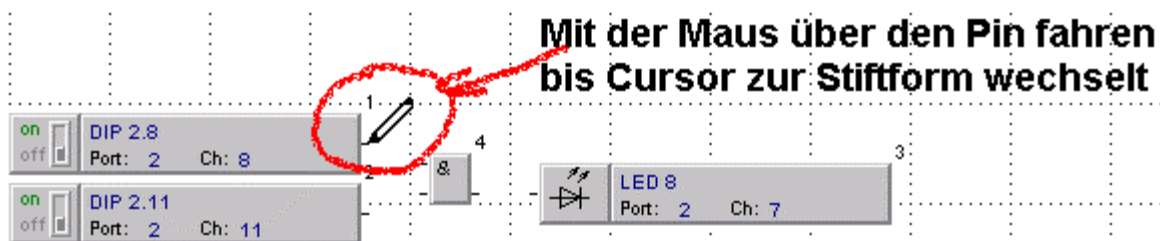
Um die Aufgabenstellung umsetzen zu können wählen sie im Parametrierdialog für den Port **P2 – LED /DIP** und für den Kanal **CH 7 – LED 8** .

5.3.3 UND Baustein einfügen

Gehen sie nach dem für den Digitalen Eingang beschriebenen Schema vor.

Wählen sie die Bibliothek Logic (Bit) und daraus den Funktionsbaustein UND. Bestätigen sie im Parameterdialog die als default eingestellten 2 Eingänge.

5.3.4 Baustein verbinden



Um die Ein- und Ausgänge der Funktionsbausteine zu verbinden verfahren sie wie folgt:

1. die Maus über den Pin des Eingangsbausteins **DIP 2.8** bewegen bis der Cursor in die Stiftform wechselt
2. linke Maustaste drücken; es erscheint ein Fadenkreuz
3. das Zentrum des Fadenkreuzes über den **oberen Pin des UND** – Bausteins bewegen
4. linke Maustaste drücken; die Verbindung wurde hergestellt

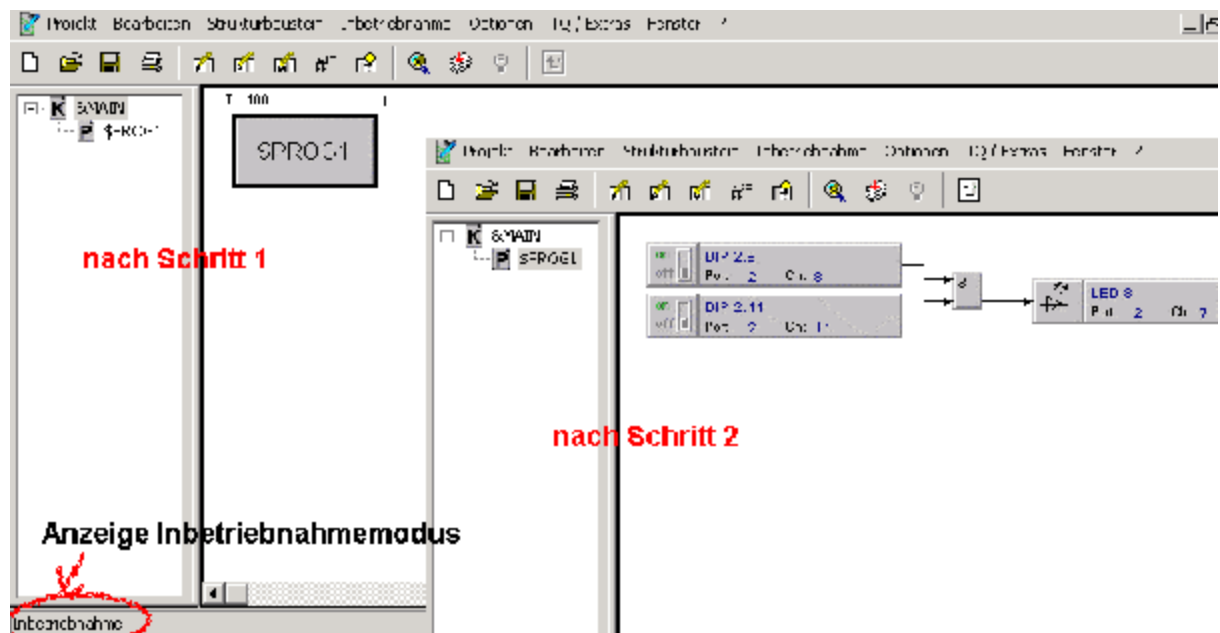
5. die Maus über den Pin des Eingangsbausteins **DIP 2.11** bewegen bis der Cursor in die Stiftform wechselt
6. linke Maustaste drücken; es erscheint ein Fadenkreuz
7. das Zentrum des Fadenkreuzes über den **unteren Pin des UND** – Bausteins bewegen
8. linke Maustaste drücken; die Verbindung wurde hergestellt
9. wiederholen sie dies mit dem Ausgang des UND – Bausteins und dem Eingang des Digitalen Ausgangs

Damit ist die Aufgabenstellung abgeschlossen und das Programm voll funktionsfähig.

5.4 Programm in die Steuerung (Starterkit) laden

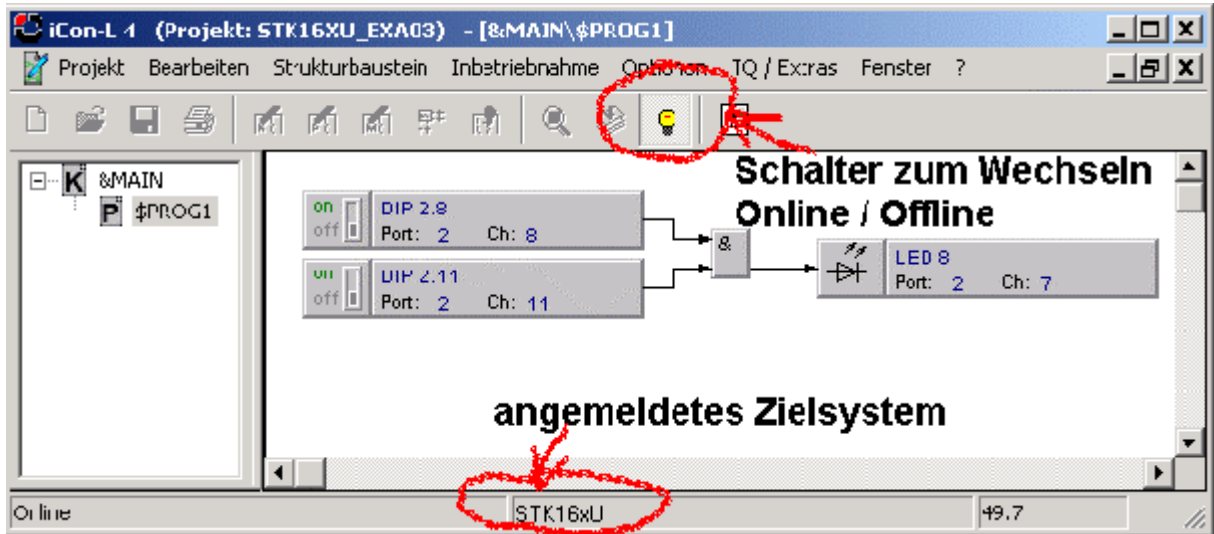
Stellen sie sicher, dass Punkt 4 dieser Kurzanleitung erfolgreich abgearbeitet wurde.

1. Wählen sie im Menü **Inbetriebnahme** den Punkt **Eintritt**.
Nach dem Wechsel in die Inbetriebnahme wird ihnen die Konfigurationsebene angezeigt (näheres dazu in der iCon-L Hilfe).
2. Klicken sie mit der Maus auf den Programmbaustein \$PROG1. Hier finden sie das von ihnen erstellte Programm.

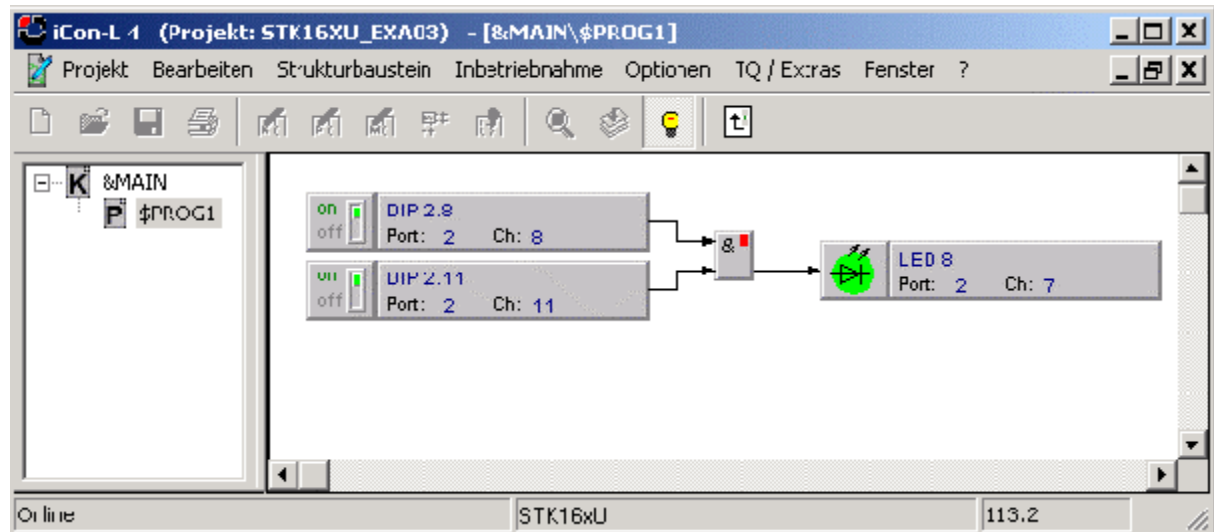


3. Wählen sie im Menü **Inbetriebnahme** den Punkt **Zielsystem anmelden**.
4. Wählen sie in dem daraufhin erscheinenden Dialog den Eintrag **STK16XU 57600 Baud**.
5. Nach erfolgreichem Aufbau der Verbindung zum Zielsystem (wie in Punkt 4 beschrieben) wählen sie im Menü **Inbetriebnahme** den Punkt **Anwenderprogramm laden**.

Nach dem Download dieses Programms schaltet das Programmiersystem in den Online – Modus. Ab jetzt können sie mit dem Starterkit kommunizieren.



Wenn sie jetzt die in der Aufgabenstellung angegebenen DIP – Schalter betätigen wird die in der Aufgabenstellung angegebene LED leuchten. Das spiegelt sich im Programmiersystem wie im nachfolgendem Bild dargestellt wieder.



6. Bereitstellung der Konfiguration für das Starterkit STK16XU

Dieser Abschnitt enthält weiterführende Informationen die nur für den erfahrenen Anwender des Starterkits im Zusammenhang mit iCon-L interessant sind.

Eine Änderung der hier beschriebenen Dateien kann dazu führen, dass die Applikationen nicht laufen.

Um Zugriff auf die IO-s des Starterkits zu erhalten muss dem Programmiersystem mitgeteilt werden, welche Ports verfügbar sind, wie deren Kanäle konfiguriert wurden usw.

Dies erfolgt über ein externes Tool, welches an dieser Stelle nicht beschrieben wird. Das Ergebnis dieses Konfigurationstools sind zwei spezifische Dateien, die der Projektvorlage beim Erstellen eines neuen Projekts automatisch beigefügt werden.

Somit wird man in die Lage versetzt auf Änderungen der Konfiguration problemlos zu reagieren.

Die in dieser Installation zur Verfügung gestellten Konfigurationsdateien erlauben den Zugriff auf die Ports 2 und 5 d.h. auf die LED-s, das Potentiometer, den Thermosensor und die DIP Schalter sowie auf das LCD- Display.

6.1 BasicConfig.ini

In dieser Datei werden die Basisdaten konfiguriert.

6.1.1 Generelle Konfiguration

[GENERAL]	
NUMOFPORTS=8	Anzahl der Ports
DEFAULTDEVICENAME=STK16XU	Default Gerätename (nur nach Rücksprache ändern)
DEFAULTTRESNAME=STK16XU.def	Basisdatei der Ressourcenanforderung (nur nach Rücksprache ändern)
RESNAMEFORPORTINIT=PORT_INITRES0	Name der Ressource in der systemintern die Daten der Portinitialisierungen eingetragen werden (nur nach Rücksprache ändern)

6.1.2 Portkonfigurationen

Die hier angegebenen Daten wurden der Beschreibung zum 16 – Bit Single – Chip Microcontroller C167CS-4R / C167CS-L (Data Sheet, V2.2 August 2001) entnommen.

[PORT_2]	
PHYSICALADDRESS=FFC0	physische Adresse
INPUTOUTPUT=IO	konfiguriert als Input / Output
[PORT_5]	
PHYSICALADDRESS=FFA2	physische Adresse
INPUTOUTPUT=I	konfiguriert als Input

6.1.3 Registerinitialisierungen

Die hier angegebenen Daten wurden der Beschreibung zum 16 – Bit Single – Chip Microcontroller C167CS-4R / C167CS-L (Data Sheet, V2.2 August 2001) entnommen.

[ODP2]	
PHYSICALADDRESS=F1C2	physische Adresse
LENGTH=2	Länge in Byte
VALUE=0000	Initialisierungswert

```

[P2]
PHYSICALADDRESS=FFC0      physische Adresse
LENGTH=2                 Länge in Byte
VALUE=0000                Initialisierungswert

[POCON2]
PHYSICALADDRESS=F088      physische Adresse
LENGTH=2                 Länge in Byte
VALUE=0000                Initialisierungswert

[DP2]
PHYSICALADDRESS=FFC2      physische Adresse
LENGTH=2                 Länge in Byte
VALUE=F0FF                Initialisierungswert

[ADDAT2]
PHYSICALADDRESS=F0A0      physische Adresse
LENGTH=2                 Länge in Byte
VALUE=0000                Initialisierungswert

[P1DIDIS]
PHYSICALADDRESS=FEE4      physische Adresse
LENGTH=2                 Länge in Byte
VALUE=0000                Initialisierungswert

[P5DIDIS]
PHYSICALADDRESS=FFA4      physische Adresse
LENGTH=2                 Länge in Byte
VALUE=FFFF                Initialisierungswert
    
```

6.2 PortConfig.ini

Dies Daten organisieren die Verwaltung der IO-s im Programmiersystem. Sie basieren teilweise auf den Daten der BasicConfig.ini. Hier werden die einzelnen Ports und die entsprechenden Kanäle konfiguriert.

```

[PORT_2]                  Port
                           der Sektionsname muss entsprechend
                           folgender Maske gebildet werden
                           PORT_9
    
```

6.2.1 generelle Portkonfiguration

```

NAME=P2                   Name
DESCRIPTION=LED/DIP       beschreibender Text
NUMINITREG=4             Anzahl der zu initialisierenden Register
INITREG001=ODP2          Register (in BasicConfig.ini beschrieben)
INITREG002=P2            Register (in BasicConfig.ini beschrieben)
INITREG003=POCON2        Register (in BasicConfig.ini beschrieben)
INITREG004=DP2           Register (in BasicConfig.ini beschrieben)
DirectionReg=DP2         Register (in BasicConfig.ini beschrieben)
    
```

6.2.2 Beispiel einer Bit-Kanal Konfiguration (Ausgang)

Die Abschnitte der Bezeichner der folgenden Schlüsseleinträge vor dem _ müssen entsprechend folgender Maske gebildet werden
CH99_

```

CH00_TYP=1                Kanal vom Typ Digitaler Ausgang
                           interner Typ des Programmiersystems
    
```

	(wird in aller erster Linie zur Identifikation der Visualisierungseigenschaften genutzt)
CH00_SIGNALTYP=0	Signaltyp BIT
CH00_NAME=LED 1	beschreibender Text
CH00_BITMASK=0001	Position des Wertes
CH00_DIRECTIONBIT=0001	Position des Directionbit (Input / Output)

6.2.3 Beispiel einer Bit-Kanal Konfiguration (Eingang)

Die Abschnitte der Bezeichner der folgenden Schlüsseleinträge vor dem _ müssen entsprechend folgender Maske gebildet werden
CH99_

CH08_TYP=2	Kanal vom Typ Digitaler Eingang interner Typ des Programmiersystems (wird in aller erster Linie zur Identifikation der Visualisierungseigenschaften genutzt)
CH08_SIGNALTYP=0	Signaltyp BIT
CH08_NAME=DIP 2.8	beschreibender Text
CH08_BITMASK=0100	Position des Wertes
CH08_DIRECTIONBIT=0100	Position des Directionbit (Input / Output)

6.2.4 Beispiel einer Analog-Kanal Konfiguration (Eingang)

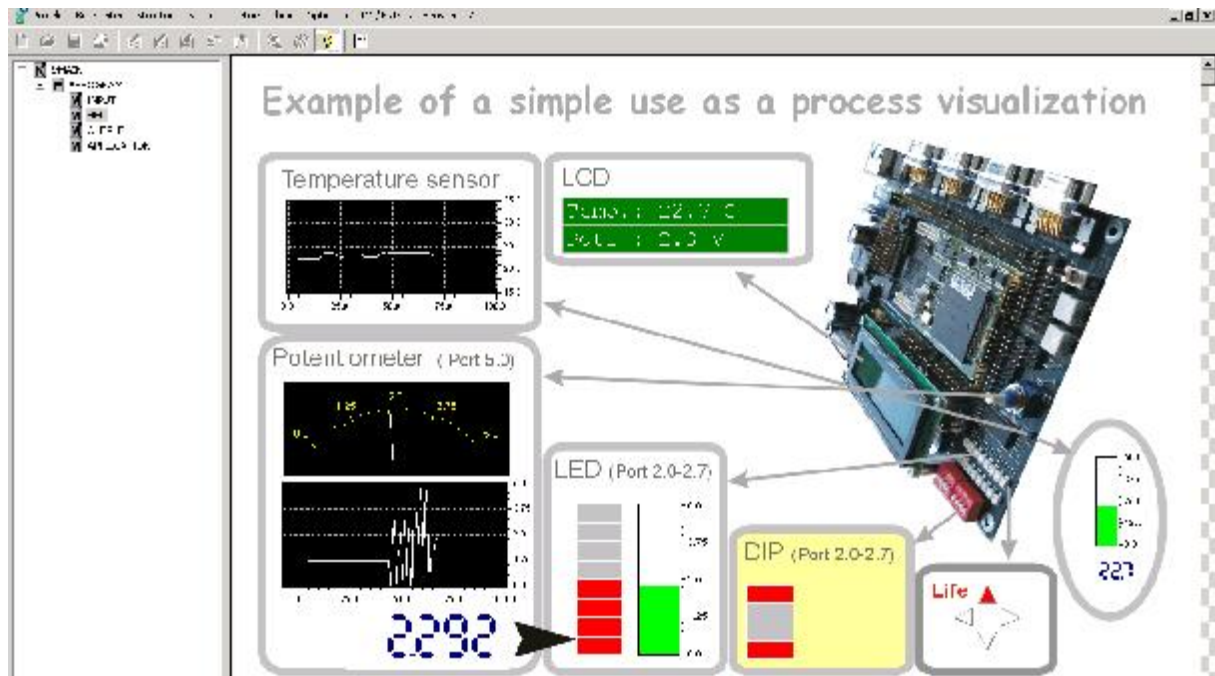
Die hier angeführten Kanäle sind Bestandteil der Port-Konfiguration für Port 5.
Die Abschnitte der Bezeichner der folgenden Schlüsseleinträge vor dem _ müssen entsprechend folgender Maske gebildet werden
CH99_

CH00_TYP=3	Kanal vom Typ Potentiometer interner Typ des Programmiersystems (wird in aller erster Linie zur Identifikation der Visualisierungseigenschaften genutzt)
CH00_SIGNALTYP=1	Signaltyp FLOAT
CH00_NAME=Potentiometer	beschreibender Name

7. Zusammenfassung und erweitertes Beispiel

Nach dem Durcharbeiten dieser Kurzanleitung haben sie einen kleinen Überblick darüber erhalten wie sie schnell und sicher ihre Hardware für ihre Zwecke verfügbar und programmierbar machen können ohne auch nur eine Zeile Code schreiben zu müssen.

Dies kann allerdings nur ein kleiner Einblick in die Möglichkeiten von iCon-L sein. Einen wesentlich besseren Überblick darüber was iCon-L kann, bietet das in ihrer aktuellen Installation im Projektverzeichnis enthaltene Beispiel STK16XU_EXA02. Starten sie dazu iCon-L, laden sie das Projekt in die Programmieroberfläche und danach in die Steuerung (wie in der Kurzanleitung beschrieben). Jetzt erhalten sie einen Eindruck davon wie leistungsfähig iCon-L im Zusammenwirken mit dem STK16XU ist.



Notizen

16;1;14;3;12;5;10;7
2;15;4;13;6;11;8;9