MU-Thermocouple1 CAN

PPCAN-Editor Konfigurations-Tutorial

Dokumentversion 1.0.1 (2020-03-09)

PEAK System

Berücksichtigte Produkte

Produktbezeichnung	Ausführung	Artikelnummer
MU-Thermocouple1 CAN (Messbereich J)	Messeinheit mit 8 Messkanälen im Metallgehäuse	IPEH-002205-J
MU-Thermocouple1 CAN (Messbereich K)	Messeinheit mit 8 Messkanälen im Metallgehäuse	IPEH-002205-K
MU-Thermocouple1 CAN (Messbereich T)	Messeinheit mit 8 Messkanälen im Metallgehäuse	IPEH-002205-T
PCAN-Explorer 5		IPES-005028
PPCAN-Editor 2, PCAN-View		

PEAK-System Technik GmbH
Otto-Röhm-Straße 69
64293 Darmstadt
Deutschland

Telefon: +49 (0)6151 8173-20
Telefax: +49 (0)6151 8173-29

www.peak-system.com
info@peak-system.com

Dokumentversion 1.0.1 (2020-03-09)

Inhalt

1 Einleitung

Die Verwendung des PPCAN-Editor 2 (anstelle der Software Thermocouple Configuration) setzt eine Mindestqualifikation des Benutzers hinsichtlich Hardwareverständnis und Programmierkenntnissen voraus.

Dieses Tutorial wendet sich daher an Besitzer eines MU-TC1, die komplexere Konfigurationen des Geräts benötigen und über belastbare Grundkenntnisse der Elektronik und Informatik verfügen.

Zunächst sollte der für Kunden kostenlose PPCAN-Editor 2 anhand dieses Tutorials getestet werden. Die Häufigkeit von Verständnisproblemen (technischer Natur!) beim Durcharbeiten des Dokuments kann möglicherweise als Entscheidungshilfe für einen zukünftigen Einsatz des Editors herangezogen werden. Im abschlägigen Fall bietet die PEAK-System Technik GmbH seinen Kunden einen Konfigurationsservice gemäß detaillierter Spezifikation.

1.1 Voraussetzungen für den Betrieb

Stellen Sie folgende vier Voraussetzungen sicher:

1. Das Gerät MU-TC1 wird mit Spannung versorgt.
2. Der D-Sub-Anschluss am Gerät ist mit einem anderen D-Sub-Anschluss eines PCAN-Interfaces über ein terminiertes CAN-Kabel verbunden.
3. Das terminierte CAN-Kabel ist am Computer angeschlossen.
4. Ein passender Sensor steckt an **Port 1A**.
5. Die Software **PPCAN-Editor 2** ist installiert.
6. Als CAN-Gegenstelle ist die Software **PCAN-View** oder besser **PCAN-Explorer** auf dem PC installiert.

7. **Nur für PCAN-Explorer-Anwender:**
   Ein Übertragungsnetz ist eingerichtet z. B. **Thermo_500k** mit 500 kbit/s (nur bei Verwendung des PCAN-Explorer).

Das Gerät **MU-TC1** (Measuring Unit ThermoCouple1) stellt folgende Ressourcen zum Verknüpfen bereit:

- Die Geräte-ID (4-Bit, 0..15 dez.) kann im Innern des Geräts per Schalter verändert werden.
- 1 CAN-Bus (mit der Nummer 1, nicht 0)
- CAN-Bitraten[1] (10k, 20k, 33k3, 47k6, 50k, 83k3, 95k2, 100k, 125k, 250k, 500k, 1M)
- CAN-Nachrichten (11-Bit oder 29-Bit)
- Information, ob und welche Messboards gesteckt sind.
  (5-Bit, 0..31 pro Slot)

  0 = keine Karte gesteckt (leerer Slot)

  15 = Messkarte für 2 Thermoelemente Typ **K** (grün) gesteckt

  16 = Messkarte für 2 Thermoelemente Typ **J** (schwarz) gesteckt

  17 = Messkarte für 2 Thermoelemente Typ **T** (braun) gesteckt
- Temperatur des Kompensationssensors (Referenztemperatur) jedes Messboards
  (insgesamt 4 Werte je 13-Bit, Auflösung 1/16 °C)
- Temperaturen der beiden Messfühler pro Board
  (insgesamt 8 Werte je 16-Bit, 1/16 °C)
- Je 2 Leuchtdioden pro Board (insgesamt 8);
  Status kann geschrieben und gelesen werden.

[1] Bitraten können frei eingestellt werden, die tatsächliche Funktionalität ist abhängig vom ausgestatteten Transceiver-Typ

2 Begriff Konfiguration

Die meisten Mikrocontroller-basierten Geräte von PEAK-System bieten die Möglichkeit, alle verbauten Schnittstellen miteinander zu verknüpfen. Hierzu stellt die Firmware Funktionsblöcke bereit, mit denen die Hardware-Ressourcen virtuell zu einer Konfiguration verschaltet werden. Zum Erstellen, Editieren und Verwalten von Konfigurationen, stellt PEAK-System den PPCAN-Editor 2 kostenlos zum Download bereit.

Die erstellte Datei mit der darin enthaltenen Konfiguration wird zunächst auf dem PC gespeichert, dann dem Gerät via CAN (per Upload) bekanntgegeben und dort nichtflüchtig gespeichert.

Manche Geräte können mehrere Konfigurationen speichern. Die jeweils gültige Einstellung wird mittels eines Wahlschalters selektiert. Der Wahlschalter bestimmt gleichzeitig die Geräte-ID und den Speicherort der Konfiguration innerhalb des nichtflüchtigen Speichers.

Die mit dem PPCAN-Editor 2 erstellten Dateien können mehrere Konfigurationen enthalten. Die Geräte-ID bestimmt dabei, welche davon bei Modulstart ausgeführt wird.

Daraus ergibt sich z. B. die Möglichkeit, mehrere gleiche Geräte mit unterschiedlicher ID an einem CAN-Bus zu betreiben und die identische Konfigurations-Datei auf alle Geräte gleichzeitig zu laden.

Die unterschiedliche ID sorgt dafür, dass jedes Gerät eine spezielle Konfiguration aus dem nichtflüchtigen Speicher lädt und dementsprechend seine Aufgabe ausführt.

2.1 Möglichkeiten der Konfiguration

Die Verknüpfung von internen Ressourcen erfolgt über die einfache Zuordnung und Skalierung von Werten, der Anwendung von den Methoden der CAN-Gateway-Dienste, Default-Werte, Funktionsblöcke, ereignisgesteuertes CAN-Senden, zeitgesteuerte Aktivitäten und über Kennlinien.

Geräte mit nur einem CAN-Bus unterstützen die Gateway-Dienste nicht. Des Weiteren sind zeitgesteuerte Aktivitäten nicht mit jeder Hardware möglich.

Alle verfügbaren Ressourcen eines Geräts werden dem PPCAN-Editor 2 mit einer speziellen Datei, die sich auf diese Hardware bezieht, gemeldet. Mit diesem Hardware-Profil kann der PPCAN-Editor 2 Konfigurationsmöglichkeiten entsprechend zulassen oder einschränken.

2.2 Skalierung

Das grundlegendste Mittel der Manipulation von Größen ist die Verwendung der vier Grundrechenarten. Diese werden mit den Parametern **SCALE** und **OFFSET** gesteuert, die von der bekannten Geradengleichung aus der Mathematik übernommen sind.

Der Parameter SCALE ist für die Multiplikation (wenn > 1) bzw. Division (wenn < 1) zuständig. Der Parameter OFFSET bewirkt eine Addition (wenn > 0, positiv) bzw. Subtraktion (wenn < 0, negativ). Als neutrale Voreinstellung ist daher SCALE = 1 und OFFSET = 0 gewählt.

2.3 CAN-Gateway-Dienste

Eingehende Nachrichten auf dem einen CAN-Bus können selektiv auf dem anderen CAN-Bus oder auf dem gleichen CAN-Bus mit anderer ID (z. B. Umsetzung 11-Bit <-> 29-Bit) ausgegeben werden. Oder eine eingehende Nachricht kann das Versenden einer beliebigen anderen Nachricht auslösen.

2.4 Default-Werte

Hier festgelegte Parameter bestimmen Zustände des Moduls nach dem Einschalten, wie z. B. die Bitrate des CAN-Busses, Aktivierung einer externen RS232-Schnittstelle, Aktivierung eines 5 V-Ausgangs für Sensorenversorgung, den logischen Zustand von Ports, Leuchtdioden und so weiter.

2.5 Funktionsblöcke

Falls die einfache Manipulation von Messgrößen etc. mittels SCALE und OFFSET nicht ausreicht, stellen Funktionsblöcke komplexere Möglichkeiten zur Verfügung.

Solche Funktionen sind zum Beispiel Wertzuordnung mit X/Y-Tabellen, Hysteresefunktionen, Delays, Zähler, Tiefpass-Filter, diverse mathematische und logische Operatoren bis zum komplexen PIDT1-Regelkreis. Funktionsblöcke können sequentiell oder bedingt abgearbeitet werden.

2.6 Ereignisgesteuertes Aussenden von CAN-Nachrichten

Falls CAN-Nachrichten nur bei bestimmten Gelegenheiten gesendet werden sollen, steht eine Sammlung an Triggerbedingungen zur Verfügung. CAN-Nachrichten können außerdem von extern angefordert werden (RTR-Nachrichten).

2.7 Kennlinien

Ein eingehender X-Wert ergibt die Ausgabe des zugeordneten Y-Wertes. Hier lassen sich 2 bis 31 X-Werte als Eingangsparameter eintragen, denen je ein Y-Wert als Ausgangsparameter zugeordnet ist. Für den Wertebereich zwischen zwei X-Werten wird das Y-Ergebnis linear interpoliert.

So lassen sich (wie bei der Skalierung) SCALE und OFFSET für bis zu 32 Abschnitte eines Wertebereichs einzeln einstellen. Damit ist es möglich, Kurvensegmente in ihrer Steigung zu beeinflussen, um beispielsweise Plateaus zu definieren oder nichtstetige Funktionen nachzubilden.

3 Aufgabenliste

Die folgenden Aufgaben samt Lösungsweg verschaffen Ihnen einen Eindruck von den vielfältigen Einsatzmöglichkeiten des MU-TC1.

- Aufgabe 1a: CAN-Bitrate setzen
- Aufgabe 1b: CAN-Nachrichten definieren
- Aufgabe 1c: Ausgabe der Temperatur des Sensors an Port 1A in °C mit max. Auflösung
- Aufgabe 1d: Exportieren eines Symbolfiles
- Aufgabe 1e: Erstellen eines einfachen Instruments-Panels
- Aufgabe 1f: Ausgabe der Temperatur in °C mit 0,5° Auflösung
- Aufgabe 1g: Ausgabe der Temperatur in °F mit 1° Auflösung (ganzzahlig)
- Aufgabe 1h: Verwendung der Kennlinienfunktion
- Aufgabe 2a: Einschalten der LED, wenn Sensor gesteckt
- Aufgabe 2b: Langsames Blinken der LED, wenn unterhalb Schwellenwert
- Aufgabe 2c: LED ein, wenn unterhalb Schwellenwert (mit Hysterese)
- Aufgabe 3a: Senden Alarm-Message, wenn außerhalb Bereich
- Aufgabe 3b: Senden der Temperatur bei Änderung um mind. 1°C
- Aufgabe 4a: Temperaturausgabe der internen Kompensationssensoren
- Aufgabe 4b: Ausgabe der Modul-ID
- Aufgabe 4c: Ausgabe des Kartentyps pro Slot
- Aufgabe 4d: Externes Setzen der LEDs
- Aufgabe 4e: Steuern der Blinkfunktion
- Aufgabe 5: CAN-Nachrichten auf Anforderung

4 Lösungswege mit Erläuterungen

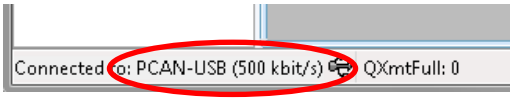
Weitere Information zur Benutzung des PPCAN-Editor 2 finden Sie in der Hilfe, die Sie im Programm über das Menü **Hilfe** oder die Taste F1 erreichen.

4.1 Aufgabe 1a: CAN-Bitrate setzen

So definieren Sie den CAN-Bus:

1. Starten Sie den PPCAN-Editor 2.
2. Verbinden Sie den PPCAN-Editor 2 mit dem CAN-Bus an dem das MU-Thermocouple1 angeschlossen ist.
3. Wählen Sie den Menüpunkt **CAN > Connect**.
4. Wählen Sie die **CAN hardware** und **Bitrate** aus.

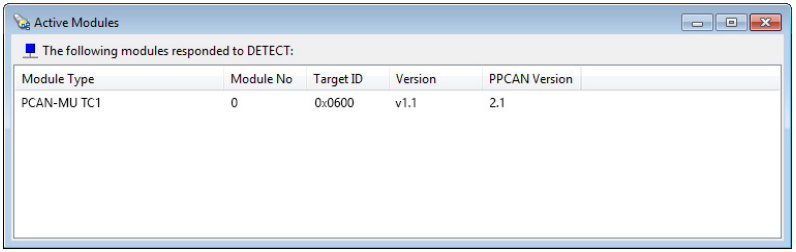
   In der Statuszeile des PPCAN-Editor 2 (unten links) erscheint ihre Auswahl.



Beispiel einer Verbindung

5. Wählen Sie den Menüpunkt **Transmit > Detect Modules** aus, um zu prüfen, ob das MU-TC1 im CAN-Netzwerk gefunden werden kann.

   Das Dialogfenster **Active Modules** erscheint. In diesem werden Statusinformationen des MU-TC1 angezeigt.



Im Feld **Module No** wird die Geräte-ID angezeigt (hier 0).
Im Feld **Version** wird die Firmware-Version angegeben.

6. Legen Sie eine leere Konfigurationsdatei an, wählen Sie den Menüpunkt **File > New** aus.

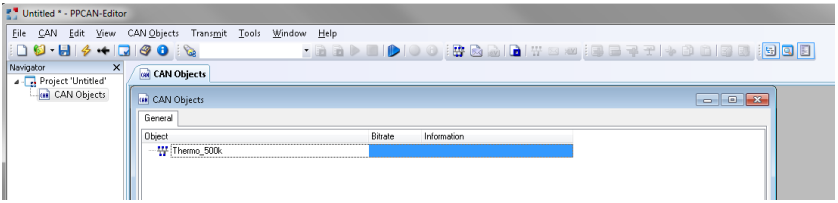
   Ein leeres Fenster mit den globalen CAN-Objekten für alle später in der Datei enthaltenen Konfigurationen erscheint.

Hinweis: Falls eine Datei mehrere Konfigurationen mit unterschiedlichen CAN-Objekten enthält, müssen alle hier definiert werden. Sie werden später selektiv in die verschiedenen Konfigurationen importiert.

Im Fenster ist bereits ein CAN-Bus als **Bus_0** definiert, unter dem man globale CAN-Objekte hierarchisch anlegen kann. Eine Konfiguration ist damit aber noch nicht angelegt.

7. Der **Bus_0** erhält den Namen **Thermo_500k** in diesem Beispiel.

8. Doppelklicken Sie auf den Namen **Bus_0** und tragen Sie den neuen Namen ein.



- **Object:** Thermo_500k (Busname)
- **Bitrate:** nur Informativ
- **Information:** Beschreibung der Zeile

4.2 Aufgabe 1b: CAN-Nachrichten definieren

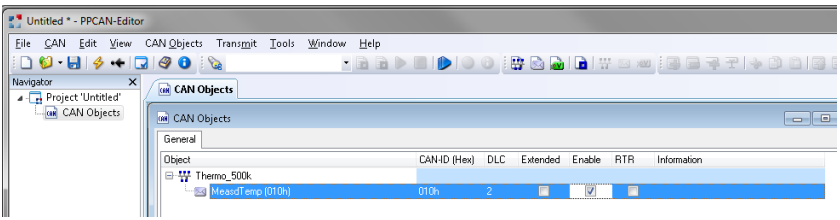
Um die geforderte Temperatur zu übertragen, genügt eine CAN-Nachricht mit 2 Bytes Länge, denn der entsprechende Messwert belegt 16 Bit.

Lösungsweg: Die CAN-Nachricht **MeasdTemp** soll, mit der ID 0x010 und der Länge von 2 Bytes, alle 300 ms zyklisch auf dem Bus **Thermo_500k** gesendet werden.

So definieren Sie die CAN-Nachricht:

1. Öffnen Sie mit einem Rechtsklick auf **Thermo_500k** das Kontextmenü.
2. Wählen Sie **Add a new Symbol** aus.

   Damit wird eine neue CAN-Nachricht auf dem Bus **Thermo_500k** definiert.
3. Tragen Sie die folgenden Parameter ein:

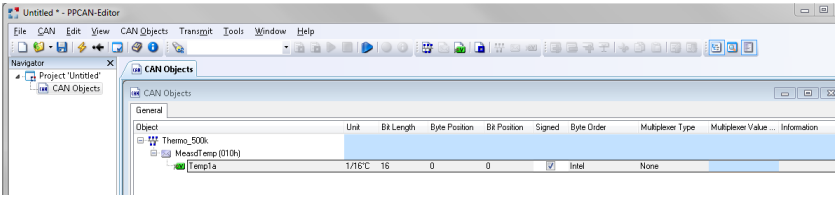


- **Object:** MeasdTemp (Symbolname)
- **CAN-ID (Hex):** 0x010
- **DLC:** 2
- **Extended:** Nein, es genügt eine 11-Bit-ID
- **Enable:** Ja
- **RTR:** Nein, die Nachricht soll nicht nur auf Anforderung gesendet werden.

- **Information:** Hier kann eine Beschreibung eingetragen werden.

Innerhalb der CAN-Nachricht muss ein 16-Bit CAN-Signal angelegt werden, das die gemessene Temperatur enthält.

4. Öffnen Sie mit einem Rechtsklick auf die CAN-Nachricht das Kontextmenü und wählen Sie **Add a new Variable** aus.
5. Tragen Sie die Parameter des Signals ein:



- **Object:** Temp1a (Variablenname)
- **Unit:** 1/16 °C (nur informativ)
- **Bit Length:** 16
- **Byte Position:** 0 (Start-Byte**)**
- **Bit Position:** 0 (Start-Bit)
- **Signed:** Ja, vorzeichenbehaftet
  (32767 größter positiver Wert, -32768 größter negativer Wert)
- **Byte Order:** Intel-Format
  (LSB in Byte 0 Bit 0, MSB in Byte 1 Bit 7)
- **Information:** Hier kann eine Beschreibung eingetragen werden.

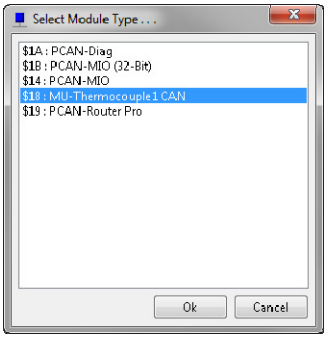
Der Rahmen der CAN-Nachricht ist damit festgelegt, aber ihr Inhalt ist noch keiner Datenquelle zugeordnet. Dazu muss eine Konfiguration erzeugt werden.

4.3 Aufgabe 1c: Ausgabe der Temperatur des Sensors an Port 1A in °C mit max. Auflösung

So legen Sie innerhalb der Konfigurationsdatei eine neue Konfiguration an:

1. Wählen Sie den Menüpunkt **Edit > New Configuration** aus.

   Ein Auswahlfenster der zu konfigurierenden Hardware erscheint.



2. Wählen Sie das Profil **MU-Thermocouple1 CAN** aus und bestätigen Sie mit **Ok**.

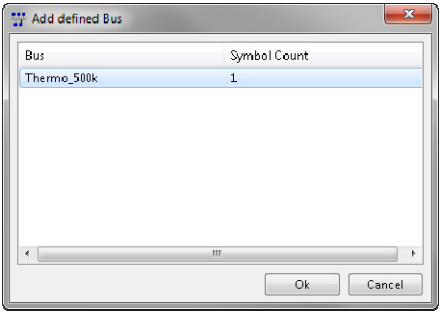
   Neben dem Tab **General** erscheint ein neuer Tab mit dem Namen der Konfiguration **Config0 I/O**.

   Ein Icon namens **Config0** ist nun im Navigationsfenster am linken Bildrand sichtbar.

   Die global definierten CAN-Busse, -Nachrichten und -Signale sollen alle in dieser Konfiguration verwendet werden. Sie müssen importiert werden.

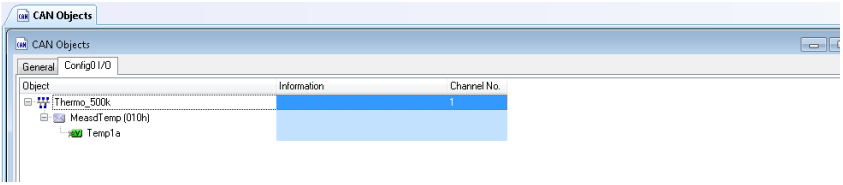
3. Klicken Sie auf den Tab **Config0 I/O**.

4. Öffnen Sie mit der rechten Maustaste das Kontextmenü und wählen Sie **Add defined bus** aus.



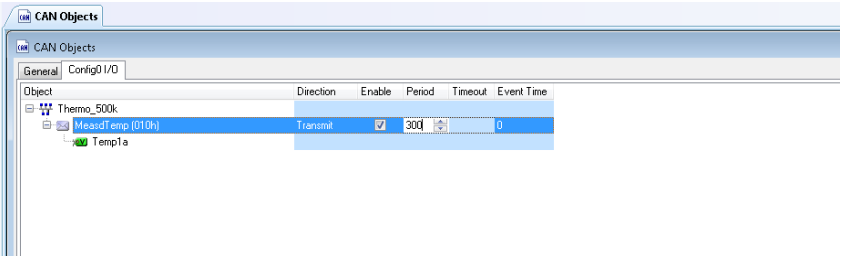
Der global bereits definierte CAN-Bus **Thermo_500k** wird mitsamt der enthaltenen Nachricht und der 16-Bit-Variable in die Konfiguration übernommen.

Wichtiger Hinweis: Die Kanalnummer beim MU-TC1 muss auf **1** gesetzt werden. Dies kann in der Listbox unter Umständen nicht ausgewählt werden (Kanal 0 existiert beim MU-TC1 nicht).



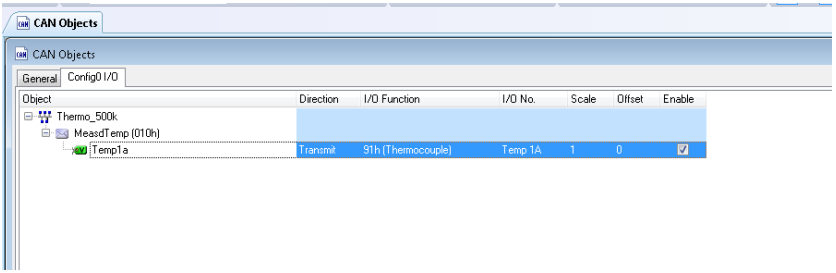
- **Channel No.**: 1 (zuweisen der Hardware-CAN-Kanäle)

5. Tragen Sie die folgenden Parameter für die Nachrichten ein:



- **Direction:** Transmit (MU-TC1 ist der Sender.)
- **Enable:** Ja, die Nachricht soll übermittelt werden.
- **Period:** 300 (Sendezykluszeit in ms)

6. Tragen sie die Parameter für die Signale ein:



- **I/O-Function:** 91-Thermocouple
  (Datenquelle: Thermocouple-Sensor)
- **I/O-No.:** Temp1A
  (Der Sensor-Port, der die Daten liefert.)
- **Scale:** 1
  (keine Verstärkung/Abschwächung der Daten, Multiplikation *1)
- **Offset:** 0 (keine Anhebung/Absenkung des Wertebereichs, Addition 0)
- **Enable:** Ja, dieses Signal (innerhalb der Nachricht) soll verwendet werden.

7. Speichern Sie die Konfiguration als `Aufgabe 1c` auf ihrem PC.
8. Übermitteln Sie die Konfigurationsdatei an das MU-TC1 über den CAN-Bus (Upload).

   Wählen Sie dazu den Menüpunkt **Transmit > Send Configuration** aus (oder wählen das entsprechend Icon).

Wichtiger Hinweis: Am oberen Fensterrand in der Toolbar muss der Eintrag **Config0** in der Listbox eingestellt sein.

Die Power-LED des MU-TC1 blinkt während der Übertragung und Verarbeitung der Konfigurationsdatei unrhythmisch. Sobald die LEDs an den Sensorports kurz aufblitzen, wurde die Konfiguration verarbeitet und ein automatischer Geräte-Reset durchgeführt.

Nun ist MU-TC1 mit seiner neuen Konfiguration betriebsbereit. Die Power-LED blinkt mit 1 Hz. Die gemessene Temperatur wird übertragen (in 1/16 °C).

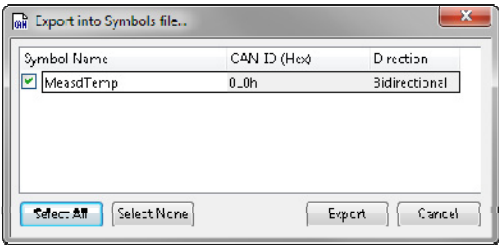
4.4 Aufgabe 1d: Exportieren eines Symbolfiles

Mit dem PCAN-Explorer kann der Lesewert einfach mit einer Symboldatei in Klartext dekodieren. Die Symboldatei wird im PPCAN-Editor 2 aus den CAN-Daten generiert und als Datei abgespeichert.

So exportieren Sie eine Symboldatei im PPCAN-Editor 2:

1. Klicken Sie auf den Tab **General** um die CAN-Datenbasis auszuwählen.
2. Wählen Sie den Menüpunkt **File > Export > Export into Symbol File** aus.
3. Speichern Sie die Datei als `Aufgabe 1d` auf ihrem PC.

   Ein Auswahlfenster mit CAN-Objekten erscheint.
4. Wählen Sie die zu exportierenden CAN-Objekte aus.



5. Klicken Sie auf Export, um die Symboldatei zu erzeugen.

So laden und aktivieren Sie die Symboldatei im PCAN-Explorer:

1. Gehen Sie im Menü auf **File > Open**.
2. Klicken Sie auf **File > Apply,** um die Datei zu aktivieren.

   Im PCAN-Explorer werden die beiden gesendeten Bytes des Messwertes im Feld **Daten** angezeigt. In diesem Fall ist der Messwert nicht wie gewünscht.

3. Führen Sie einen Rechtsklick über die Symboldatei im **Project Browser** aus und öffnen Sie die Datei.

   Der Editor öffnet sich im PCAN-Explorer.

4. Suchen Sie die folgende Zeile:

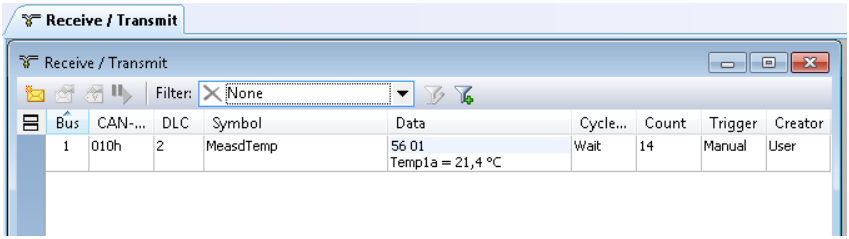
   ```
   a=Temp1a        signed
   ```

5. Ergänzen Sie diese um folgende Parameter:

   ```
   a=Temp1a        signed /u:°C /f:0,0625
   ```

   Dies ist notwendig, weil zum einen ein physikalischer Messwert mit der Einheit **°C** übertragen wird, zum anderen weil dieser Messwert in 1/16 Grad-Schritten (=0,0625) aufgelöst ist.

Nach der manuellen Korrektur des Symbolfiles (mit Abspeichern und erneuter Bestätigung) wird im Feld **Daten** der aktuelle Messwert in **°C** angezeigt.



4.5 Aufgabe 1e: Erstellen eines einfachen Instruments-Panels

Besitzer des PCAN-Explorer können den Messwert noch grafisch aufbereiten, in dem ein Instrument-Panel angelegt wird.

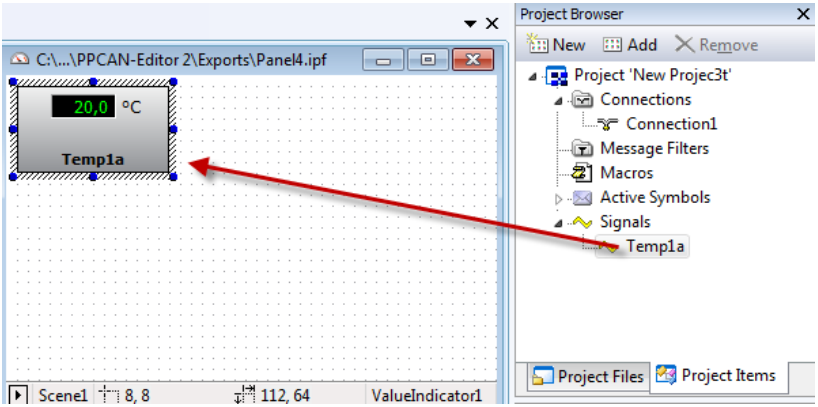
So erstellen Sie ein Instrument-Panel im PCAN-Explorer:

1. Wählen Sie den Menüpunkt **Tools > Instruments Panel > Create Value Indicator** aus.

   Ein leeres Panel mit einem digitalen Anzeigeinstrument erscheint, dem eine Datenquelle zugewiesen werden muss.

2. Ziehen Sie mit der linken Maustaste die Variable **Temp1a** aus dem **Project Browser** auf das Instrument.

   Das Instrument zeigt nun den Namen der Variablen und erhält die passende Formatierung.



3. Speichern Sie die Datei als `Aufgabe 1e` auf ihrem PC.

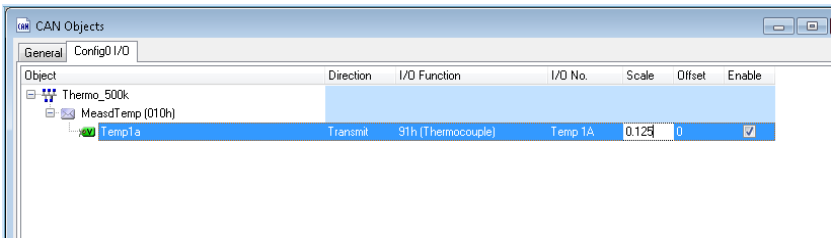
4.6 Aufgabe 1f: Ausgabe der Temperatur in °C mit 0,5° Auflösung

Die Auflösung 1/16 °C ist zwar die maximal mögliche, ihre Verwendung ist oft aber praxisfern. Stattdessen könnte z. B. in 0,5 °C Schritten gemessen werden. Dafür genügt die Manipulation des Scale-Wertes (1/16 * 2), also eine Multiplikation um den Faktor 0,125. Hardwareseitig wird weiterhin in 1/16 °C Schritten gemessen. Lediglich die Skalierung der übertragenen Daten wird verändert.

So erhalten Sie die Temperatur mit 0,5° Auflösung:

1. Klicken Sie im Tab **Config0 I/O** auf das CAN-Signal **Temp1A** und ändern Sie den **Scale**-Wert von 1 auf 0,125.

   Dadurch wird die Schrittzahl pro Grad auf 16*0,125 = 2 reduziert.



2. Speichern Sie die Konfiguration als `Aufgabe 1f` auf ihrem PC.
3. Übermitteln Sie die Konfigurationsdatei an das MU-TC1 über den CAN-Bus (Upload).

Tipp: Nur ein 1/8 des Wertebereiches ist in einer CAN-Nachricht erforderlich. 3 Bit könnten einspart werden. Dann wäre die Variable Temp1a nur noch 13 Bit lang (statt 16 Bit).

```
Picture=aaaaaaaa aaaaaaaa // Original
Picture=aaaaaaaa ---aaaaa // angepasst
a=Temp1a    signed  /u:°C /f:0,0625 // Original
a=Temp1a    signed  /u:°C /f:c      // angepasst
```

4.7 Aufgabe 1g: Ausgabe der Temperatur in °F mit 1° Auflösung (ganzzahlig)

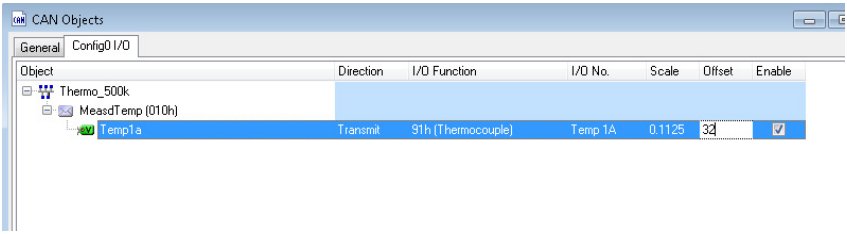
Durch die Parameter Scale und Offset lässt sich die Temperatur in ganzen **° Fahrenheit** anzeigen. Das Gerät misst intern nach wie vor in 1/16° Celsius. Der Umrechnungsfaktor ist wie folgt:

$$T_{Fahrenheit} = ((T_{Celsius} \times 9) / 5) + 32$$

Daraus errechnet sich ein Scale-Wert von 1/16 *9 / 5 = **0,1125**. Der Offset-Wert von **32** wird übernommen, da die Übertragung auf volle Grad umgerechnet wurde. Bei einer ¼ Grad Auflösung müsste sowohl bei Scale als auch bei Offset ein Faktor 4 mitgerechnet werden (also Scale: 0,1125 *4 = **0,45** und Offset: 32 *4 = **128**).

So erhalten Sie die Temperatur in Fahrenheit:

1. Klicken Sie im Tab **Config0 I/O** auf das CAN-Signal **Temp1A**.
2. Ändern Sie den **Scale**-Wert von 1 auf 0,1125.
3. Ändern Sie den **Offset**-Wert von 0 auf 32.



4. Speichern Sie die Konfiguration als `Aufgabe 1g` auf ihrem PC.
5. Übermitteln Sie die Konfigurationsdatei an das MU-TC1 über den CAN-Bus (Upload).

```
a=Temp1a    signed  /u:°C /f:0,0625 // Original
a=Temp1a    signed  /u:°F           // angepasst
```

4.8 Aufgabe 1h: Verwendung der Kennlinienfunktion

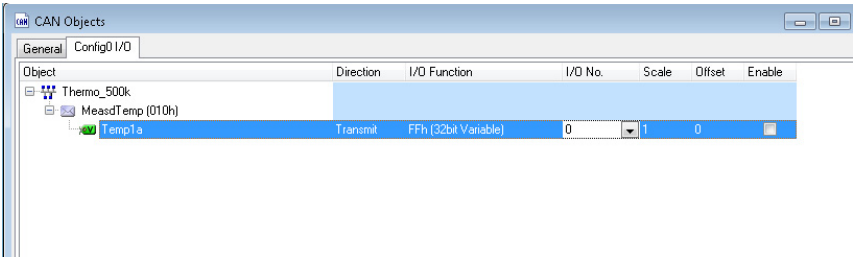
Die Kühlmitteltemperatur soll zwischen 80 °C und 105 °C als konstant 90 °C angezeigt werden (um Schwankungen zu glätten bzw. den Fahrer nicht zu beunruhigen). Oder der Vorlauf eines Tachometers soll stufenweise mit der Geschwindigkeit wachsen.

Da man auf dem Schreibtisch aber nicht mit hohen Temperaturen oder Geschwindigkeiten experimentieren soll, wurde ein anderes Beispiel gewählt, das den Mechanismus begreiflich macht. Der übertragene Messwert (= Temperatur) des Sensors von Port 1A soll im Bereich 0..50 °C invertiert werden.

Die gemessene Temperatur wurde bisher direkt in das CAN-Signal geleitet und gegebenenfalls per Scale und Offset etwas formatiert. Bei komplexeren Umrechnungen muss ein Funktionsblock aus der Firmware herangezogen werden.

In diesem Fall einer, der den Messwert über eine Kennlinie führt (weist jedem X-Wert einen neuen Y-Wert zu) und das Ergebnis in eine 32-Bit-Variable schreibt. Deren neuer Inhalt (der transformierte Messwert) kann jetzt in das CAN-Signal eingesetzt werden.

Beginnend mit dem letzten Schritt, wird in der CAN-Datenbasis nicht mehr direkt der Messwert von Port 1A gesendet, sondern der Inhalt der 32-Bit-Variable #0. Basierend auf dem Beispiel 1c ist beim CAN-Signal die I/O-Function und die I/O-No. wie folgt zu ändern:



- **I/O-Function**: FF 32bit-Variable
- **I/O-No.**: 0

Als nächstes wird eine fallende Kennlinie (z. B. Nummer #7) eingerichtet, die 2 Punkte hat:

	X	Y
Punkt 1	X = 0	Y = 800 (entspricht 50°C)
Punkt 2	X = 800	Y = 0 (entspricht 0°C)

So erstellen Sie eine Kennlinie:

1. Führen Sie am linken Bildrand im Navigationsfenster einen Doppelklick auf die Konfiguration **Config0** aus.

   Ein neues Fenster mit dem Titel **Config0** öffnet sich.

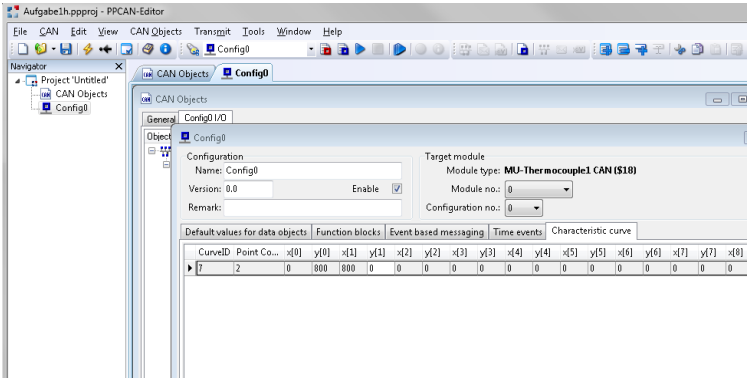
   Hier können komplexe Verknüpfungen von Ressourcen vorgenommen werden.

2. Wählen Sie im Fenster den Tab **Characteristic curve** aus.

3. Öffnen Sie mit der rechten Maustaste das Kontextmenü und wählen Sie **Add Record** aus.

   Eine Tabellenzeile erscheint, die eine Kennlinie repräsentiert.

4. Tragen Sie die folgenden Werte ein:



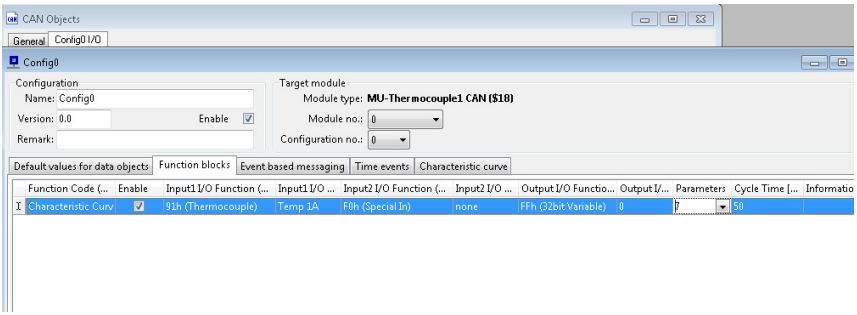
- **Curve ID**: 7 (eine willkürlich gewählte Nummer)
- **Point Count**: 2 (Anzahl der X/Y-Wertepaare)
- **Wertepaar 1**: X = 0 soll Y = 800 ergeben
- **Wertepaar 2**: X = 800 soll Y = 0 ergeben
  (man erhält so eine fallende Gerade)
- **Information**: Beschreibung der Zeile.

Zuletzt muss dafür gesorgt werden, dass der Messwert zu den X-Koordinaten der Kennlinie gelangt und der entsprechende Y-Wert in die 32-Bit-Variable #0 geschrieben wird, um auf dem CAN-Signal übermittelt zu werden. Dazu wird ein Funktionsblock benötigt, der die Handhabung der Kennlinie übernimmt.

So erstellen Sie den benötigten Funktionsblock:

1. Wählen Sie den Tab **Function blocks** aus und legen Sie einen neuen Funktionsblock an.
2. Öffnen Sie mit der rechten Maustaste das Kontextmenü und wählen Sie **Add Record** aus.

   Eine Tabellenzeile erscheint, die einen Funktionsblock darstellt.
3. Tragen Sie die folgenden Werte ein:



- **Function Code**: Characteristic Curve
  (Kennlinien-Bedien-Funktion)
- **Enable**: Ja, dieser Block soll aktiv sein.
- **Input1**: "91-Thermocouple" und "Temperature1a"
  (Quelle der X-Werte)
- **Input2**: "F0-Special In" und "none" (nicht benutzt)
- **Output**: "FF-32bit-Variable" und "0"
  (Y-Ergebnis wird in die 32bit-Variable #0 gespeichert.)
- **Parameters**: 7
  (Nummer der bereits definierten Kennlinie)
- **Cycle Time**: 50
  (Die Umsetzung des Messwerts findet alle 50 ms statt.)
- **Information**: Beschreibung der Zeile

4. Speichern Sie die Konfiguration als `Aufgabe 1h` auf ihrem PC.
5. Übermitteln Sie die Konfigurationsdatei an das MU-TC1 über den CAN-Bus (Upload).

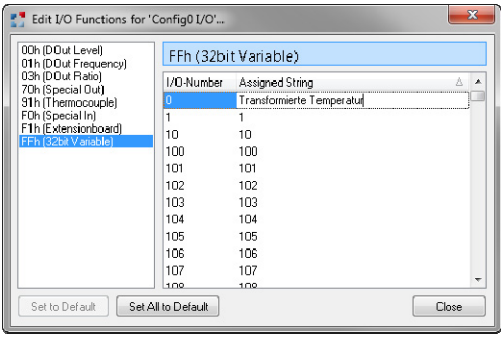
Sobald der intakte Sensor in Port 1A steckt, wird eine Temperatur angezeigt, die 50°C - des gemessenen Wertes entspricht. Wenn der Sensor durch die Finger erwärmt wird, fällt die angezeigte Temperatur entsprechend.

Zudem kann jeder 32-Bit-Variablen (und auch anderen Ressourcen) kann ein lesbarer Name zugewiesen werden.

So weisen Sie jeder 32-Bit-Variablen einen Namen zu:

1. Gehen Sie auf den **Function blocks** Tab.
2. Wählen Sie den Menüpunkt **Edit > I/O-Function** aus.

   Folgendes Fenster erscheint:



3. Benennen Sie die 32-Bit-Variable #0 in **Transformierte Temperatur**.

   Ab sofort kann man in der Konfiguration mit diesem Namen arbeiten, was Fehler durch Verwechslung einer Variablennummer erheblich reduziert.

4.9 Aufgabe 2a: Einschalten der LED, wenn Sensor gesteckt

Wenn man die Konfigurationsdatei aus Aufgabe 1c erneut lädt und den Temperatursensor vom Port 1A abzieht, wird ein Rohwert von 0x8000 übertragen.

Da Temperaturen unterhalb der Naturkonstante K_0 = -273,15 °C nicht vorkommen können (dies entspräche Messwerten von 0x8000 bis 0xEEEC hex), wird dieser Wertebereich zum Darstellen von Fehlern genutzt, z. B.:

- **0x8000**: Keine Thermospannung gemessen: Sensor am Port nicht eingesteckt oder Kabelbruch.
- **0xE000**: Karte nicht gesteckt bzw. nicht erkannt.

Die Messwerte vom Sensor können also anhand des Wertebereichs in gültig und ungültig unterschieden werden.

Der mathematische Vergleich des Messwertes mit einer Konstanten (z.B. 0x8000) erfordert die Verwendung eines Funktionsblocks der Firmware.

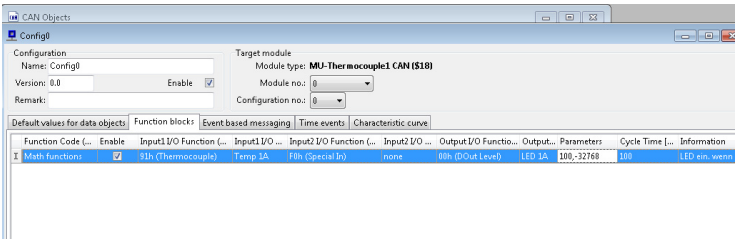
So verwenden Sie den Funktionsblock:

1. Öffnen Sie die Aufgabe 1c und speichern diese unter `Aufgabe2a`.
2. Doppelklicken Sie am linken Bildrand im Navigationsfenster die Konfiguration **Config0**.

   Ein neues Fenster namens **Config0** öffnet sich.
3. Wählen Sie den Tab **Function blocks** aus.
4. Öffnen Sie mit der rechten Maustaste das Kontextmenü und wählen Sie **Add Record** aus.

   Eine Tabellenzeile erscheint, die einen Funktionsblock repräsentiert.

5. Tragen Sie die folgenden Werte für den oben beschriebenen mathematischen Vergleich ein:



- **Function Code**: MathFunction
- **Enable**: Ja, dieser Block soll aktiv sein.
- **Input1**: "91-Thermocouple" und "Temperature1a"
- **Input2**: "F0-Special In" und "none" (nicht benutzt)
- **Output**: "00-Dout Level" und "LED 1A"
  (Leuchtdiode zum Port 1A)
- **Parameter**: Art der mathematischen Funktion:
  Logischer Vergleich mit 0x8000

  Als Funktion wird gewählt: **In1 > const** und als Wert wird **-32768** eingegeben. Das Ergebnis ist ein Bool'scher Wert (TRUE oder FALSE), der die LED steuert.
- **Cycle Time**: z.B. 100
  (Logischer Vergleich findet alle 100 ms statt.)
- **Information**: Beschreibung der Zeile.

6. Speichern Sie die Konfiguration als `Aufgabe 2a` auf ihrem PC.
7. Übermitteln Sie die Konfigurationsdatei an das MU-TC1 über den CAN-Bus (Upload).

Sobald der intakte Sensor in Port 1A steckt, leuchtet die LED. Wenn der Sensor abgezogen wird, geht die LED aus.

4.10 Aufgabe 2b: Langsames Blinken der LED, wenn unterhalb Schwellenwert

LED 1A soll blinken, wenn der Messwert 30 °C unterschritten wird. Die LED soll aus sein, wenn der Messwert mindestens 30 °C erreicht. Sie soll zudem dauerhaft leuchten, wenn kein Sensor gesteckt ist.

In diesem Beispiel wird ein bedingtes Ausführen von Funktionsblöcken demonstriert. Hier sind drei verschiedene Messbereiche zu prüfen:

1. Wenn das Ergebnis TRUE ist, wird die Folgezeile ausgeführt, um die LED entsprechend zu setzen.

2. Bei Ergebnis FALSE wird die Folgezeile übersprungen.

   Die Anzahl der zu überspringenden Zeilen wird im **Parameter**-Feld der jeweiligen Bereichsprüfung angegeben (hier: jeweils 1). So lassen sich CASE-ähnliche Strukturen erstellen.

Blinken wird über die Funktion **PWM** realisiert. Per Defaultwert wird eine feste PWM-Frequenz eingestellt. Das Tastverhältnis entscheidet über den Zustand der LED 1A: **aus**, **blinken** oder **ein**.

Die Frequenz kann in 0,1 Hz-Schritten zwischen 0,1 und 10 Hz eingestellt werden. Das Tastverhältnis (Ratio) wird von 0 bis 255 eingestellt. Der Wert 127 entspricht dabei 50 %.

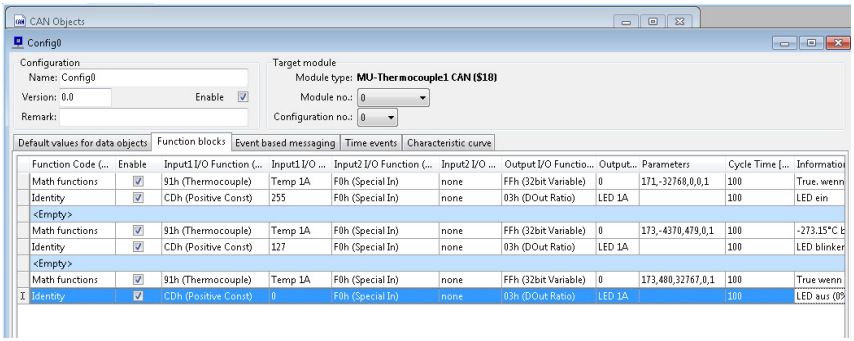
Drei Funktionsblöcke für drei Bereichsprüfungen sind notwendig, sowie jeweils ein zugeordneter Funktionsblock, um die PWM-Ratio (= der entsprechenden Blinkart) zu setzen. Zur Gliederung einer Konfiguration können Trennzeilen eingefügt werden (**Menü Edit > Insert Space Record)**.

So blinkt die LED langsam unterhalb eines Schwellenwertes:

1. Doppelklicken Sie auf das Icon **Config0** im linken Navigationsfenster.

   Ein neues Fenster namens **Config0** öffnet sich.

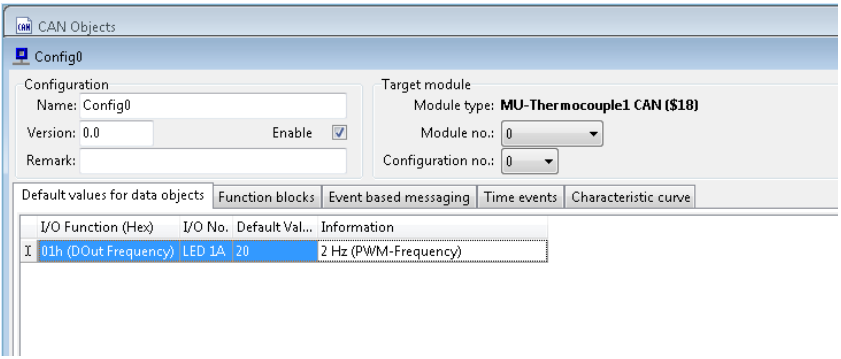
2. Wählen Sie den Tab **Function blocks** aus.
3. Öffnen Sie mit der rechten Maustaste das Kontextmenü und wählen Sie **Add Record** aus.
4. Tragen Sie die folgenden Werte ein:



Nun ist noch die Blinkfrequenz 2 Hz (per Default-Wert) festzulegen.

So legen Sie die Blinkfrequenz fest:

1. Wählen Sie den Tab **Default values for data objects** aus.
2. Öffnen Sie mit der rechten Maustaste das Kontextmenü und wählen Sie **Add Record** aus.
3. Tragen die folgenden Werte ein:



- **I/O Function:** 01h (DOut Frequency)
- **I/O No.:** LED 1A
- **Default Value:** 20
- **Information:** Beschreibung der Zeile.

4. Speichern Sie die Konfiguration als `Aufgabe 2b` auf ihrem PC.
5. Übermitteln Sie die Konfigurationsdatei an das MU-TC1 über den CAN-Bus (Upload). Wenn kein Sensor in Port 1A steckt, leuchtet die LED.

Wenn ein Sensor gesteckt ist und die Messtemperatur unterhalb 30 °C liegt, blinkt die LED. Steigt die Temperatur auf mindestens 30 °C, geht die LED aus.

4.11 Aufgabe 2c: LED ein, wenn unterhalb Schwellenwert (mit Hysterese)

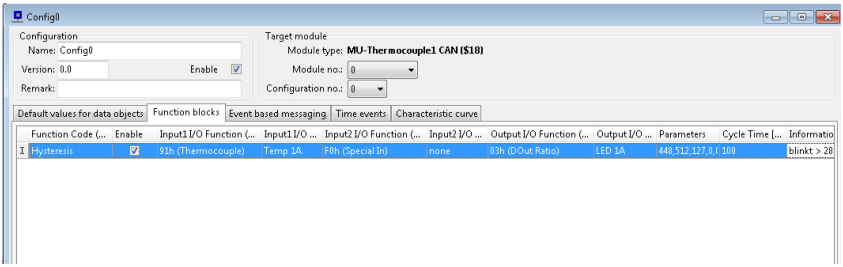
Die Leuchtdiode LED 1A soll blinken, bis der steigende Messwert 32 °C erreicht wird. Sie soll aus bleiben, bis der Messwert 28 °C unterschritten wird, darunter aber wieder blinken.

Blinken wird über die Funktion **PWM** realisiert. Per Defaultwert wird eine feste PWM-Frequenz eingestellt. Das Tastverhältnis entscheidet über den Zustand der Leuchtdiode 1A: aus oder blinken.

So schaltet die LED unter einem bestimmten Schwellenwert ein:

1. Öffnen Sie die Aufgabe 1c und speichern diese unter `Aufgabe 2c`.
2. Klicken Sie am linken Bildrand im Navigationsfenster die Konfiguration **Config0** an.

   Ein neues Fenster namens **Config0** erscheint.
3. Wählen Sie den Tab **Function blocks** aus.
4. Öffnen Sie mit der rechten Maustaste das Kontextmenü und wählen Sie **Add Record** aus.
5. Tragen Sie die folgenden Werte ein:



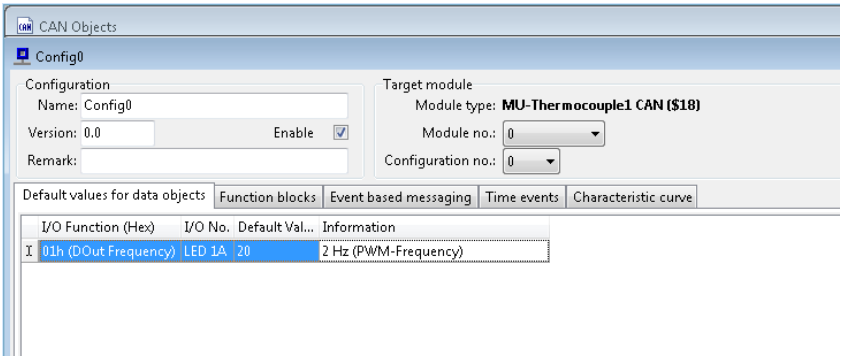
- **Function Code:** Hysterese
- **Input I/O-Function:** 90-Thermocouple (Auswerten der Messtemperatur)

- **Input I/O-No.**: Temp1A
- **Output I/O-Function**: PWM-Ratio
  (Ergebnis wirkt auf Ratio der LED)
- **Output I/O-No.**: LED 1A
- **Hysterese-Parameter**: 448 (= 28 °C), 512 (= 32 °C), 127 (= 50 % Ratio), 0 (= LED aus)

Als Nächstes ist die Blinkfrequenz 2 Hz (per Default-Wert) festzulegen.

So legen Sie die Blinkfrequenz fest:

1. Wählen Sie den Tab **Default values for data objects** aus.
2. Öffnen Sie mit der rechten Maustaste das Kontextmenü und wählen Sie **Add Record** aus.
3. Tragen die folgenden Werte ein:



- **I/O Function**: 01h (DOut Frequency)
- **I/O No.**: LED 1A
- **Default Value**: 20
- **Information**: Beschreibung der Zeile.

4. Speichern Sie die Konfiguration als `Aufgabe 2c` auf ihrem PC.

5. Übermitteln Sie die Konfigurationsdatei an das MU-TC1 über den CAN-Bus (Upload).

Erwärmen des Sensors: Blinken beginnt über 32 ° C.

Abkühlen des Sensors: Blinken beginnt unter 28 °C.

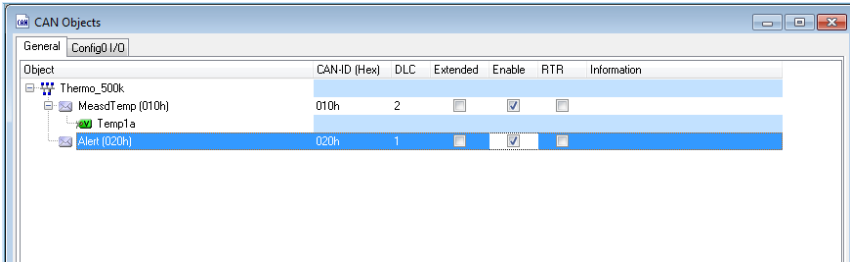
4.12 Aufgabe 3a: Senden Alarm-Message, wenn außerhalb Bereich

Im folgenden Beispiel (basierend auf Aufgabe 1c) soll eine separate CAN-Nachricht gesendet werden, wenn der Messwert einen bestimmten Bereich verlässt. Dazu muss zunächst die CAN-Datenbasis um eine zusätzliche Nachricht erweitert werden.

Lösungsweg: Senden der CAN-Nachricht **Alert** mit der ID 0x020 und der Länge von 1 Byte auf dem Bus **Thermo_500k**, aber nur wenn der Sensor an Port 1A den Bereich von 28 bis 33 °C verlässt. Diese Nachricht enthält das Signal **ErrFlag**.

So richten Sie eine Alarm-Nachricht ein:

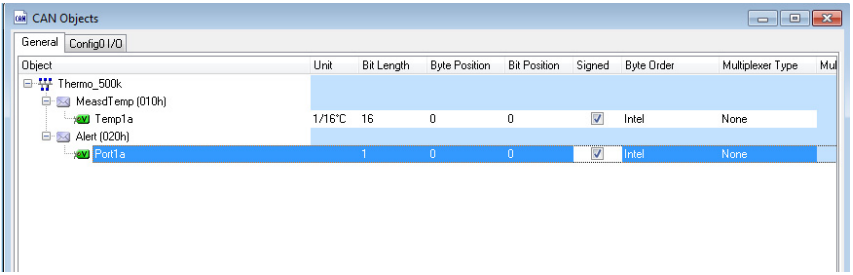
1. Fügen Sie im Tab **General** die Alarm-Nachricht hinzu.
2. Öffnen Sie mit der rechten Maustaste das Kontextmenü und wählen Sie **Add a new Symbol** aus.
3. Tragen Sie die folgenden Werte ein:



- **Object:** Alert (Symbolname)
- **ID:** 20h
- **DLC:** 1
- **Extended:** Nein, weil eine 11bit-ID genügt.
- **Enabled:** ja

- **RTR:** Nein, die Nachricht soll nicht nur auf Anforderung gesendet werden.

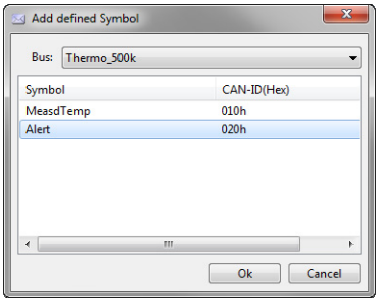
4. Legen Sie innerhalb der CAN-Nachricht ein neues Signal an.
5. Öffnen Sie mit der rechten Maustaste das Kontextmenü und wählen Sie **Add a new Variable** aus.
6. Tragen Sie die folgenden Werte ein:



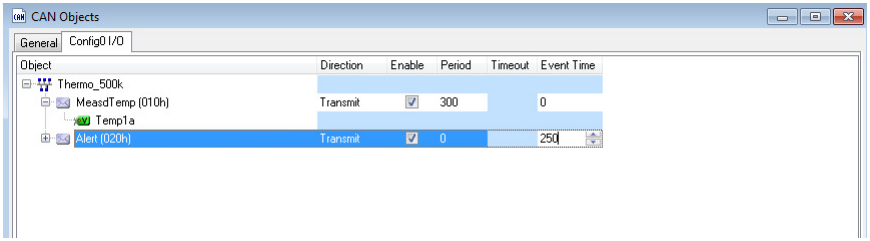
- **Object:** Port1a (Variablen-Name)
- **Unit:** -
- **Bit length:** 1
- **Byte Position:** 0 (Start-Byte)
- **Bit Position:** 0 (Start-Bit)
- **Signed:** ja (bei 1-Bit-Werten unrelevant)
- **Byte Order:** Intel-Format (bei 1-Bit-Werten unrelevant)

Der Rahmen der CAN-Nachricht ist damit festgelegt. Importieren Sie diese in die Konfigurationsdatei **Config0**.

7. Holen Sie den Tab **Config0 I/O** in den Vordergrund.
8. Öffnen Sie mit der rechten Maustaste das Kontextmenü und wählen Sie **Add defined Symbol** aus.
9. Wählen Sie die Alert-Nachricht aus.

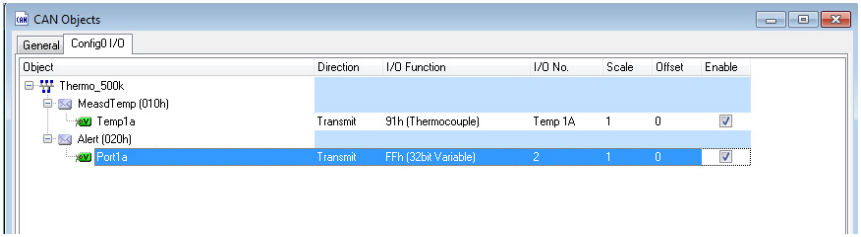


10. Tragen Sie die Parameter der Nachricht ein:



- **Direction:** Transmit (MU-TC1 soll der Sender sein.)
- **Enable:** Ja, diese Nachricht soll übermittelt werden.
- **Period:** 0 (kein zyklisches Senden, sondern nur im Alarmfall)
- **Event Time:** 250 ms
  (Liegt der Alarm statisch an, erfolgt das zyklische Senden zwischen zwei aufeinanderfolgenden Alert-Nachrichten.)

11. Tragen Sie die Parameter des Signals ein:



- **I/O-Function**: FF 32bit-Variable
  (Variable enthält das Error-Flag.)
- **I/O-No.**: 2 (Variablen-Nummer).
- **Scale**: 1
- **Offset**: 0
- **Enable**: Ja, Signal soll in der Nachricht verwendet werden.

Zwei weitere Schritte sind notwendig, um die Aufgabe zu lösen:

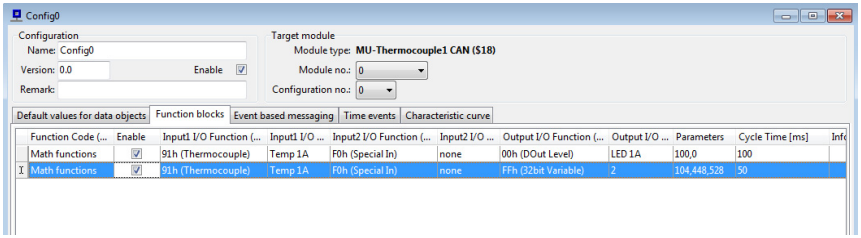
- Erstens ist eine zyklische Prüfung notwendig, ob die an Port 1A gemessene Temperatur den Wertebereich verlassen hat (Bereichsprüfung mit bool'schem Wert als Ergebnis).
- Zweitens muss das Senden der Nachricht **Alert** getriggert werden.

So erstellen Sie eine zyklische Prüfung:

1. Doppelklicken Sie auf das Icon **Config0** im linken Navigationsfenster.

   Ein neues Fenster namens **Config0** erscheint.
2. Wählen Sie den Tab **Function blocks** aus.
3. Öffnen Sie mit der rechten Maustaste das Kontextmenü und wählen Sie **Add Record** aus.

   Eine Tabellenzeile erscheint, die einen Funktionsblock repräsentiert.
4. Tragen Sie die folgenden Werte ein:



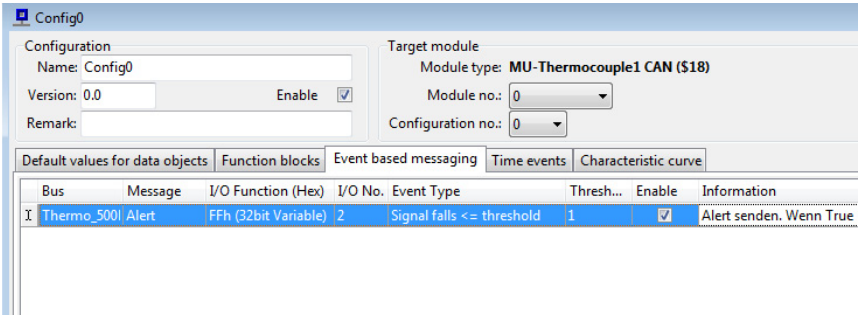
- **Function Code**: MathFunction
- **Enable**: Ja, dieser Block soll aktiv sein.
- **Input1**: "91-Thermocouple" und "Temp1A"
- **Input2**: "F0-Special In" und "none" (nicht benutzt)
- **Output**: "FF 32bit-Variable" und #2 (willkürlich gewählt)
- **Parameters**: Art der mathematischen Funktion:
  TRUE, wenn außerhalb fester Grenzen:
  Untergrenze 448 (= 28 °C)
  Obergrenze 528 (= 33 °C)
- **Cycle Time**: 50
  (Der logische Vergleich findet alle 50 ms statt.)
- **Information**: Beschreibung der Zeile

Abschließend soll anhand des bool'schen Wertes in der Variablen #2 eine CAN-Nachricht getriggert werden.

So erstellen Sie eine getriggerte CAN-Nachricht:

1. Wählen Sie den Tab **Event based Messaging** aus.
2. Öffnen Sie mit der rechten Maustaste das Kontextmenü und wählen Sie **Add Record** aus.

   Eine Tabellenzeile erscheint.
3. Tragen Sie die folgenden Werte ein:



- **Bus:** Thermo_500k
- **Message:** Alert
  (CAN-Nachricht soll im Alarm-Fall gesendet werden.)
- **I/O-Function:** 32bit-Variable #2
  (Prüft das Ergebnis im definierten Bereich.)
- **Event Type:** Wenn das Ergebnis >= 1 (also TRUE) ist.
- **Threshold:** 1 (TRUE)
- **Enable:** Ja, der Trigger soll wirksam sein.
- **Information:** Beschreibung der Zeile

4. Speichern Sie die Konfiguration als `Aufgabe 3a`.
5. Übermitteln (Upload) Sie die Konfigurationsdatei an das MU-TC1 über den CAN-Bus.

Bei einer Raumtemperatur unterhalb 28 °C wird im Abstand von 250 ms die Nachricht **Alert** (ID = 0x020) gesendet. Wenn man den Sensor mit dem Finger erwärmt, bleibt die Übertragung ab 28 °C stehen, bis 33 °C überschritten sind. Dann wird erneut die Nachricht gesendet und kein Alarm wird zwischen 28 und 33°C ausgelöst.

Je nach Jahreszeit / Temperatur können andere Temperaturschwellen sinnvoll sein. Man könnte in der Alert-Nachricht auch die Messtemperatur übertragen, oder ein Oberhalb- / Unterhalb-Flag.

4.13 Aufgabe 3b: Senden der Temperatur bei Änderung um mind. 1°C

Im folgenden Beispiel (basierend auf Aufgabe 1c) soll die CAN-Nachricht **MeasdTemp** nur noch dann gesendet werden, wenn sich die gemessene Temperatur um mindestens 1 °C verändert hat.

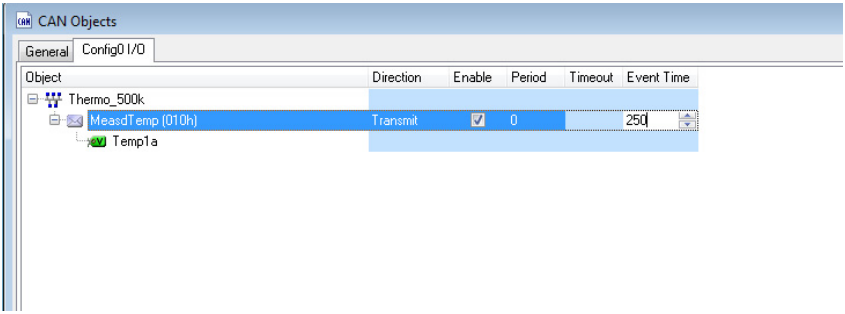
Die Nachricht **MeasdTemp** muss in der CAN-Datenbasis modifiziert und ein entsprechender Sende-Trigger definiert werden.

So senden Sie eine Nachricht bei einer Temperaturänderung:

1. Ändern Sie im Tab **Config0 I/O** in der CAN-Nachricht **MeasdTemp** die Sendeperiode auf **0**.

2. Setzen Sie ein **Event-Time** von **250 ms**.

   Die Nachricht wird nicht mehr zyklisch gesendet.



   Damit die Nachricht hin und wieder gesendet wird, muss der Auslöser definiert werden.

3. Doppelklicken Sie am linken Bildrand im Navigationsfenster die Konfiguration **Config0**.

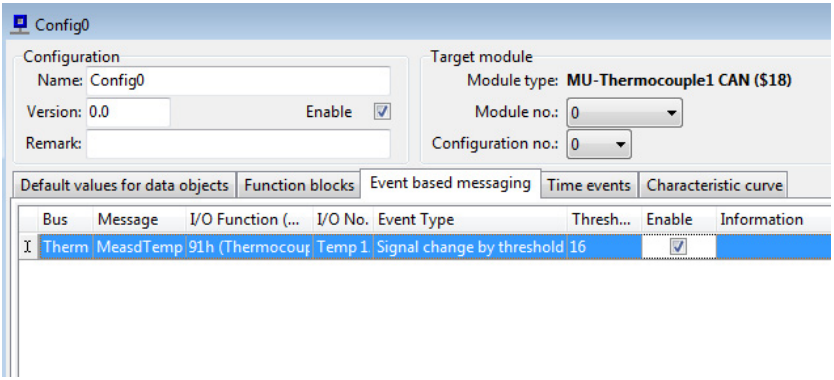
   Ein neues Fenster namens **Config0** erscheint.

4. Wählen Sie den Tab **Event based Messaging** aus.

5. Öffnen Sie mit der rechten Maustaste das Kontextmenü und wählen Sie **Add Record** aus.

   Eine Tabellenzeile erscheint.

6. Tragen Sie die folgenden Werte ein:



- **Bus:** Therm_500k
- **Message:** MeasdTemp
  (Temperatur-Nachricht soll gesendet werden.)
- **I/O-Function:** 91-Thermocouple
  (Temperatur-Messwert soll bewertet werden.)
- **I/O_Number:** Temp1a (Temperatur-Messwert an Port 1A)
- **Event Type:** Wenn sich der Messwert um einen bestimmten Betrag (Threshold) ändert.
- **Threshold:** 16
  (Entspricht 1 °C, weil Messung in 1/16tel Grad Auflösung erfolgt.)
- **Enable:** Ja, der Trigger soll wirksam sein.
- **Information:** Beschreibung der Zeile

7. Speichern Sie die Konfiguration als `Aufgabe 3b` auf ihrem PC.

8. Übermitteln Sie die Konfigurationsdatei an das MU-TC1 über den CAN-Bus (Upload).

Die CAN-Nachricht **MeasdTemp** wird nur gesendet, wenn sich der Sensor erwärmt/abkühlt oder gezogen/gesteckt wird.

4.14 Aufgabe 4a: Temperaturausgabe der internen Kompensationssensoren

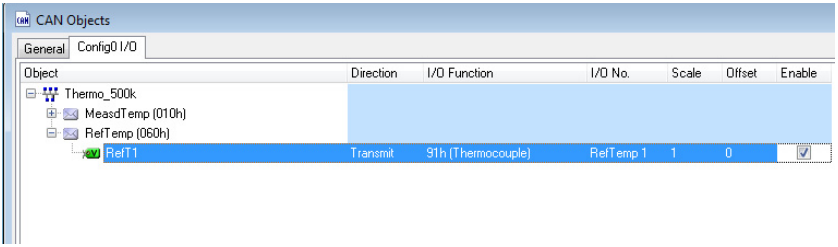
Die zur Temperaturmessung herangezogene Thermospannung an der Spitze des Messsensors kann durch verschiedene Einflüsse merklich verfälscht werden. Eine wesentliche Fehlerquelle sind die metallischen Kontakte der 8 Anschlussports. Hier entstehen Thermospannungen, zum Beispiel durch Lötstellen, deren Einfluss mit zunehmender Umgebungstemperatur wächst.

Zur Kompensation wird die Temperatur an diesen Kontaktstellen gemessen und das mathematische Äquivalent einer entgegengesetzten Spannung in den Messwert eingerechnet. Die Temperatur an den Kontaktstellen (sogenannte Referenztemperatur) kann ausgelesen werden.

Lösungsweg: Die CAN-Nachricht **RefTemp** soll mit der ID 0x060 und Länge von 8 Bytes auf dem Bus **Thermo_500k** gesendet werden. Die Nachricht enthält das Signal **RefT1**, das die Referenztemperatur der linken Karte (Steckplatz #1) enthält.

So geben Sie die Temperatur der internen Sensoren aus:

1. Öffnen Sie die Aufgabe 1c und speichern diese unter `Aufgabe 4a`.
2. Legen Sie eine neue Nachricht **RefTemp** im Tab **General** an.
3. Legen Sie eine neues Signal (13-Bit) im Tab **General** an.
4. Importieren Sie die Nachricht inklusive des CAN-Signals in die Konfiguration und tragen Sie die Parameter ein:



5. Speichern Sie die Konfiguration als `Aufgabe 4a` auf ihrem PC.
6. Übermitteln Sie die Konfigurationsdatei an das MU-TC1 über den CAN-Bus (Upload).

Die Nachricht **RefTemp** überträgt (mit der ID 0x060) die Temperatur.
Diese wird an den Sensorports 1A und 1B gemessen.
Die Temperatur sollte in etwa der Raumtemperatur entsprechen.

4.15 Aufgabe 4b: Ausgabe der Modul-ID

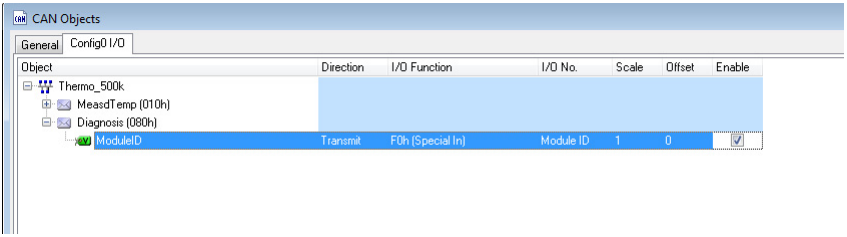
Die Modul-ID ist ein 4-Bit-Wert, der bei Auslieferung auf 0 eingestellt ist und im Innern des MU-TC1 per DIP-Schalter verändert werden kann. Die Modul-ID hat verschiedene Funktionen, darunter das Auswählen der passenden Konfiguration je nach Schalterstellung. Beispielsweise kann zwischen verschiedenen Konfigurationen geschaltet werden, die in einer PPCAN-Projektdatei enthalten sind.

Hinweis: Bei unerklärlichem Verhalten einer neuen Konfiguration ist einer der ersten Debugging-Schritte das Feststellen der Modul-ID. Möglicherweise wird jedes Mal eine andere Konfig ausgeführt.

Lösungsweg: Die CAN-Nachricht **Diagnosis** soll mit der ID 0x080 und der Länge von 8 Bytes auf dem Bus **Thermo_500k** gesendet werden. Diese Nachricht enthält das Signal **ModuleID**, dass die derzeit eingestellte Modul-ID anzeigt.

So geben Sie die Modul-ID aus:

1. Öffnen Sie die Konfiguration aus Aufgabe 1c und speichern diese unter `Aufgabe 4a`.
2. Legen Sie eine neue Nachricht Diagnosis im Tab **General** an.
3. Legen Sie ein neues Signal (4-Bit, unsigned) im Tab **General** an.
4. Importieren Sie die Nachricht inklusive des CAN-Signals in die Konfiguration.
5. Tragen Sie die folgenden Parameter ein:



- **I/O-Function:** F0-Special In
  (eine von mehreren Statusinformationen)
- **I/O-No.:** Module ID
  Der I/O-Nummer 16 wurde bereits ein Name zugeordnet.
- **Scale**: 1
- **Offset**: 0
- **Enable**: Ja.

6. Speichern Sie die Konfiguration als `Aufgabe 4b` auf ihrem PC.
7. Übermitteln Sie die Konfigurationsdatei an das MU-TC1 über den CAN-Bus (Upload).

Die Nachricht **Diagnosis** (mit der ID 0x080) überträgt die Modul-ID. Eine Änderung der Modul-ID wird erst nach Reset des Geräts aktiv (z.B. Power Off / On).

4.16 Aufgabe 4c: Ausgabe des Kartentyps pro Slot

Das Controllerboard des MU-TC-1 kann 31 verschiedene Messkartentypen erkennen. Der entsprechende Wert von 0 bis 31 kann gelesen und angezeigt werden. Derzeit sind definiert:

0 = Keine Karte gesteckt (leerer Slot).

15 = Messkarte für 2 Thermoelemente Typ K (grün) gesteckt.

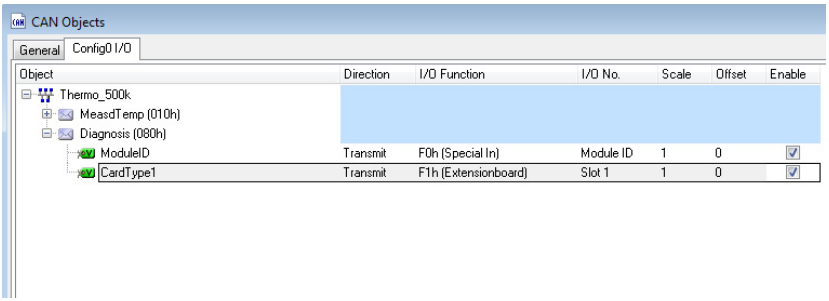
16 = Messkarte für 2 Thermoelemente Typ J (schwarz) gesteckt.

17 = Messkarte für 2 Thermoelemente Typ T (braun) gesteckt.

Lösung: Die CAN-Nachricht **Diagnosis** soll mit der ID 0x080 und der Länge von 8 Bytes auf dem Bus **Thermo_500k** gesendet werden. Diese Nachricht enthält das Signal **CardType1**, das den Typ der in Slot 1 erkannten Karte angibt.

So geben Sie den Kartentyp pro Slot aus:

1. Öffnen Sie die Konfiguration aus Aufgabe 4b und speichern diese unter `Aufgabe 4c`.
2. Legen Sie ein Signal (5-Bit, unsigned) im Tab **General** an.
3. Importieren Sie die Nachricht inklusive des CAN-Signals in die Konfiguration und tragen Sie die Parameter ein:



- **I/O Function:** F1-ExtensionBoard
  (eine von mehreren Statusinformationen)
- **I/O No:** Slot 1 (hier wurde der I/O-Nummer 0 bereits ein Name zugeordnet)
- **Scale:** 1
- **Offset:** 0
- **Enable:** Ja

4. Speichern Sie die Konfiguration als `Aufgabe 4c` auf ihrem PC.
5. Übermitteln Sie die Konfigurationsdatei an das MU-TC1 über den CAN-Bus (Upload).

Die Nachricht **Diagnosis** überträgt (mit der ID 0x080) außer der Modul-ID (in Byte 0) noch den Kartentyp im Slot 1 (in Byte 4).

Zusätzlich kann im PCAN-Explorer eine Nummernfolge in Klartext umgewandelt werden. Bezogen auf die Aufgabe 4c kann der Wertebereich von 0 bis 31 entsprechend zu Kartentypen dekodiert werden.

Im Symbol-File finden Sie den Abschnitt **ENUMS**. Hier können Sie Aufzählungen definieren:

```
{ENUMS}
  enum BoardType(15="K(green)", 16="J(black)", 17="T(brown)",
  0="empty"
```

In der Nachrichten-Definition können Sie die definierten **ENUMS** verwenden:

```
{RECEIVE}
  [Diagnosis]
  ID=80h
  Picture=----aaaa -------- -------- -------- ---bbbbb --------
  -------- --------
  a=Module ID       unsigned
  b=Board-0 unsigned /e:BoardTyp
```

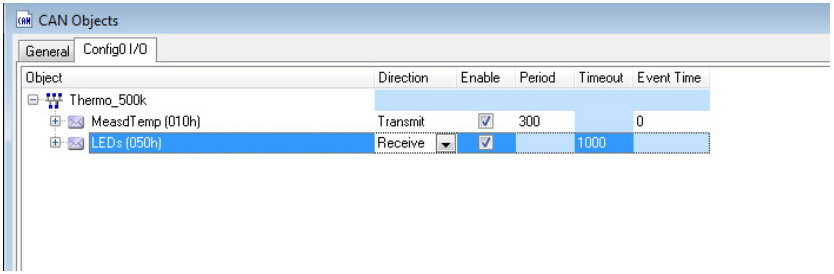
4.17 Aufgabe 4d: Externes Setzen der LEDs

Das Setzen der Leuchtdioden aufgrund von geräteinternen Entscheidungen wurde bereits in Aufgabe 2 gezeigt. Das Setzen der LEDs von außerhalb durch eine CAN-Empfangsnachricht ist ebenfalls möglich.

Lösung: Die CAN-Nachricht **LEDs** soll mit der ID 0x050 und Länge von 4 Bytes auf dem Bus **Thermo_500k** empfangen werden. Diese Nachricht enthält das Signal **LED 1A,** das mit der LED für den Messfühler-Port 1A korrespondiert. Wenn die Nachricht ausbleibt, soll die LED nach 1 Sekunde ausgehen.

So setzen Sie LEDs von außerhalb durch eine CAN-Nachricht:

1. Öffnen Sie die Konfiguration aus Aufgabe 1c und speichern diese unter `Aufgabe 4d`.
2. Legen Sie eine neue Nachricht LEDs im Tab **General** an.
3. Legen Sie eine neues Signal (1-Bit) im Tab **General** an.
4. Importieren Sie die Nachricht inklusive des CAN-Signals in die Konfiguration.
5. Tragen Sie die folgenden Parameter ein:



- **Direction:** Receive
  (Diese Nachricht kommt von außen und wird ausgewertet.)
- **Enable**: Ja, die Nachricht soll verwendet werden.

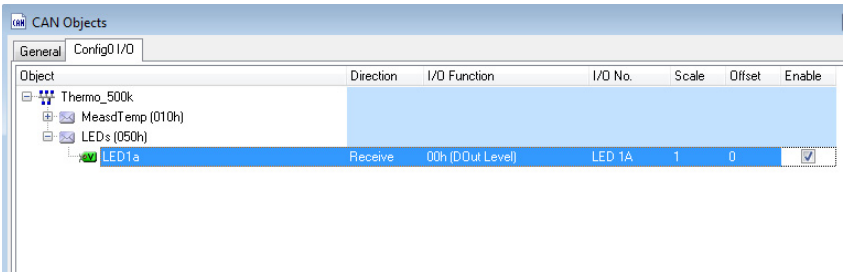
- **Period**: Sendezyklus, entfällt bei Receive-Nachrichten.
- **Timeout**: 1000 ms
  (Danach wird der Default-Wert angenommen.)
- **EventTime**: Sendezyklus wenn ein Event vorliegt.
  (Dieser entfällt bei Receive-Nachrichten)

Die Nachricht **LEDs** wird als Empfangs-Nachricht definiert und hat deshalb keine Sendeperiode.

Das **Timeout** für diese Nachricht beträgt 1000 ms. Danach wird der Default-Wert wirksam.

Das Empfangssignal **LED 1A** wirkt sich direkt auf die entsprechende Ressource aus.

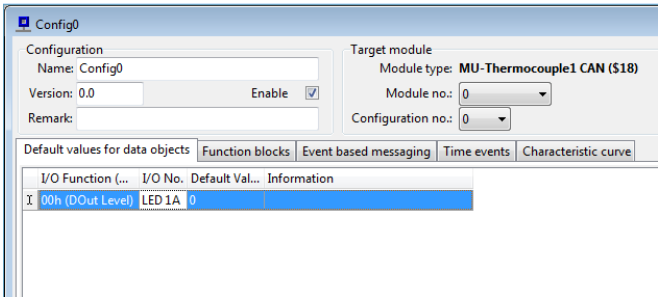
6. Tragen Sie die folgenden Parameter für **LED 1A** ein:



- **I/O-Function**: 00-Dout Level (eine der Leuchtdioden)
- **I/O-No.**: LED 1A (Leuchtdiode zum Port 1A)
- **Scale**: 1
- **Offset**: 0
- **Enable**: Ja, dieses Signal soll verwendet werden.

Abschließend muss noch der Defaultwert für die Leuchtdiode gesetzt werden. Bei PowerOn und im Timeout-Fall soll die LED aus sein: **Default = 0**.

7. Tragen Sie die folgenden Parameter für **LED 1A** in **Config0** ein:



- **I/O-Function:** 00-Dout-Level (eine der Leuchtdioden)
- **I/O-No.:** LED 1A (Leuchtdiode zum Port 1A)
- **Default-Value:** 0 (LED soll aus sein.)
- **Information:** Beschreibung der Zeile

8. Speichern Sie die Konfiguration als `Aufgabe 4d` ab.
9. Übermitteln Sie die Konfigurationsdatei an das MU-TC1 über den CAN-Bus (Upload).

Der PCAN-Explorer (alternativ PCAN-View) sendet eine CAN-Nachricht mit der ID 0x050, in der Byte 0 und Bit 0 = 1 gesetzt sind. Die Leuchtdiode geht an. Wird innerhalb einer Sekunde eine weitere Nachricht gesendet, bleibt die Leuchtdiode an.
Wird für mehr als eine Sekunde die Nachricht nicht gesendet, tritt der Default-Wert in Kraft und die Leuchtdiode geht aus.

Geben Sie in der PCAN-Explorer-Symboldatei den folgenden Code ein, um eine Sende-Nachricht zu definieren:

```
{SEND}
  [Leds]
  ID=50h
  //      Byte 0   Byte 1   Byte 2   Byte 3   Byte 4   Byte 5   Byte 6   Byte 7
  //    76543210 76543210 76543210 76543210 76543210 76543210 76543210 76543210
  Picture=-------a -------- -------- --------
  a=Led1a bit
```

4.18 Aufgabe 4e: Steuern der Blinkfunktion

Blinkende Leuchtdioden können auch über eine PWM-Funktion realisiert werden. Dazu stehen (wie bei PWM zu erwarten) die Parameter **Frequenz** und **Ratio** zur Verfügung.

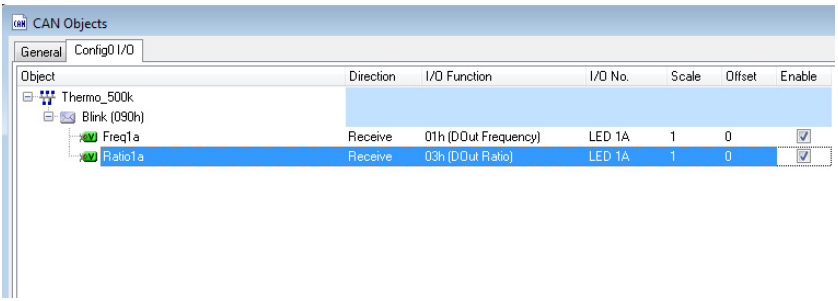
	X	Y
Frequenz0	0, 1 .. 10 Hz	0, 1 Hz
Ratio (Puls-Pause-Verhältnis)	0 255 entspricht 0..100%	1

Lösung: Die CAN-Nachricht **Blink** mit der ID 0x090 und Länge 2 Bytes soll auf dem Bus **Thermo_500k** empfangen werden. Diese Nachricht enthält die beiden 8bit-Signale **Freq1a** und **Ratio1a**, die direkt die **LED 1A** beeinflussen.

So steuern Sie die Blinkfunktion:

1. Legen Sie eine neue Empfangsnachricht inklusiv der beiden Signale an.

   Diese werden direkt auf die I/O-Funktionen **Frequency** und **Ratio** der LED 1A geschrieben.



2. Speichern Sie die Konfiguration als `Aufgabe 4e`.
3. Übermitteln Sie die Konfigurationsdatei an das MU-TC1 über den CAN-Bus (Upload).

Der PCAN-Explorer (alternativ PCAN-View) sendet eine 2 Bytes lange CAN-Nachricht mit der ID 0x090, in der Byte 0 und Byte 1 mit entsprechenden Werten für **Frequency** und **Ratio** belegt sind. Die Leuchtdiode blinkt entsprechend der eingestellten Werte.

So definieren Sie eine Sende-Nachricht mit dem PCAN-Explorer in der Symboldatei aus Aufgabe 4e:

```
{SEND}
  [Blink]
  ID=90h
  //      Byte 0   Byte 1   Byte 2   Byte 3   Byte 4   Byte 5   Byte 6   Byte 7
  //    76543210 76543210 76543210 76543210 76543210 76543210 76543210 76543210
  Picture=aaaaaaaa bbbbbbbb
  a=Freq1a unsigned
  b=Ratio1a unsigned
```

So gehen Sie im PCAN-Explorer vor:

1. Wählen Sie im Menü **Tools > Instruments Panel > Create vertical Slider Control** aus.

   Ein leeres Panel mit einem senkrechten Schieberegler erscheint.

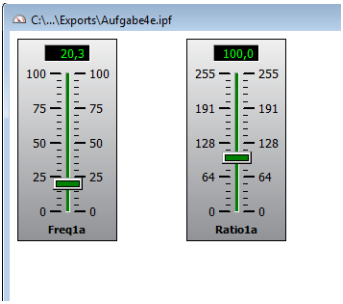
2. Wählen Sie erneut den Menüpunkt **Tools > Instruments Panel > Create vertical Slider Control** aus.

   Ein weiterer senkrechter Schieberegler wird dargestellt, den Sie neben dem ersten Instrument platzieren können.

3. Ziehen Sie mit der linken Maustaste die Variablen **Freq1a** und **Ratio1a** nacheinander aus dem **Project Browser** auf je einen Schieberegler.

   Beiden Reglern sind nun Sendeparameter zugewiesen. Dadurch zeigen sie die Namen der Variablen in einer passenden Formatierung.

Hinweis: Eventuell ist der Wertebereich des Reglers **Freq1a** im Kontextmenü unter **Properties** auf **100** zu begrenzen.



4. Speichern Sie die Datei als `Aufgabe 4e` auf ihrem PC.

4.19 Aufgabe 5: CAN-Nachrichten auf Anforderung

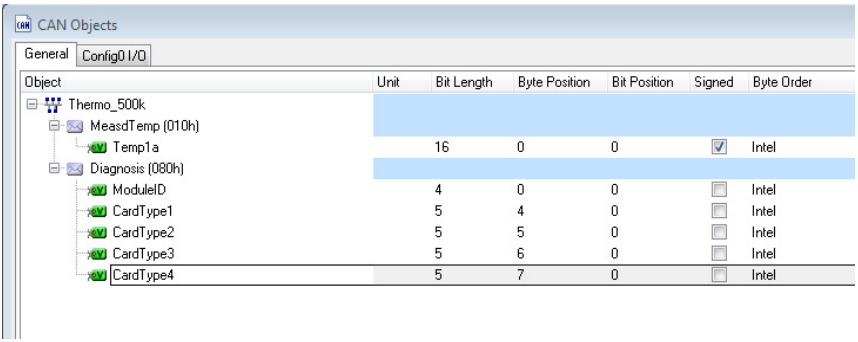
Eine Statusinformation soll nur auf Anfrage ausgegeben werden. Ausgehend von Aufgabe 4c sollen die Typen aller gesteckten Karten über CAN ausgegeben werden.

Lösung: Die CAN-Nachricht **Diagnosis** soll mit der ID 0x080 und der Länge 8 Bytes auf dem Bus **Thermo_500k** auf Anforderung (Remote Transmission Request) gesendet werden. Diese Nachricht enthält die Signale **CardType1** bis **CardType4**, die den Typ der in den Slots erkannten Karten angibt.

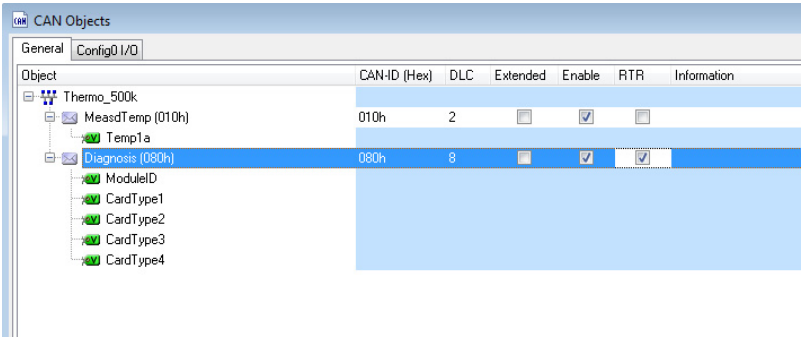
So erhalten Sie CAN-Nachrichten auf Anforderung:

1. Öffnen Sie die Konfiguration der Aufgabe 4c und speichern diese unter `Aufgabe 5` auf ihrem PC.

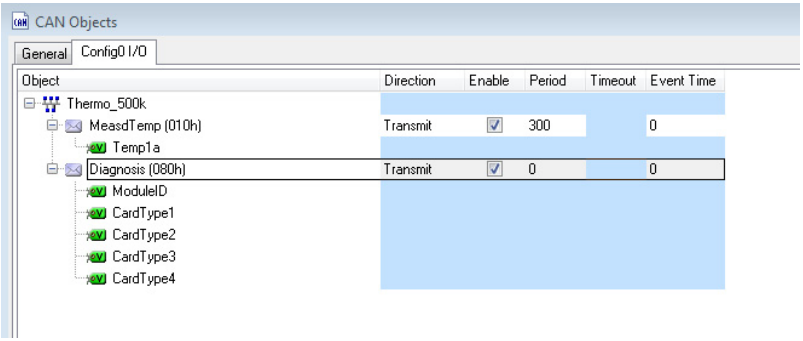
2. Fügen Sie 3 weitere CAN-Signale (je 5-Bit, unsigned) hinzu und setzen Sie die Parameter wie folgt:



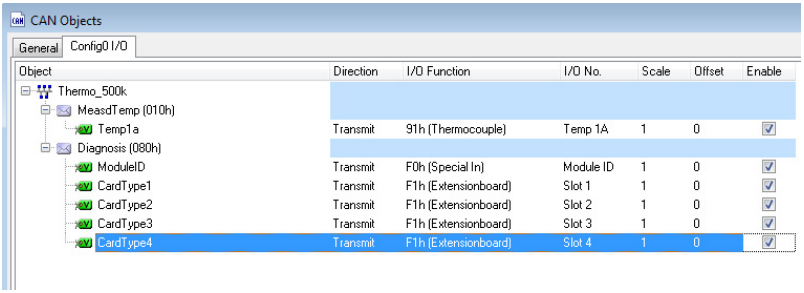
3. Aktivieren Sie die Checkbox **RTR**, um die CAN-Nachricht auf **Transmission Request** umzustellen.



4. Importieren Sie die drei neuen Signale in die Konfiguration: Öffnen Sie mit der rechten Maustaste das Kontextmenü und wählen Sie **Add defined Variable** aus.



5. Geben Sie diese Werte ein und weisen Sie die virtuellen Modulressourcen zu:



6. Speichern Sie die Konfiguration als `Aufgabe 5` auf ihrem PC.
7. Übermitteln Sie die Konfigurationsdatei an das MU-TC1 über den CAN-Bus (Upload).

Der PCAN-Explorer (alternativ PCAN-View) sendet eine CAN-Nachricht mit der ID 0x080 und der Länge DLC = 0. Das MU-TC1 sendet daraufhin die 8 Byte lange Nachricht **Diagnosis**, die unter anderem die 4 Kartentypen enthält.

Im PCAN-Explorer-Symbolfile kann die Nachricht **Diagnosis** entsprechend ergänzt werden.

- Geben Sie folgenden Code um die Nachricht zu ergänzen:

```
[Diagnosis]
  ID=80h
  //      Byte 0   Byte 1   Byte 2   Byte 3   Byte 4   Byte 5   Byte 6   Byte 7
  //    76543210 76543210 76543210 76543210 76543210 76543210 76543210 76543210
  Picture=----aaaa -------- -------- -------- ---eeeee ---fffff ---ggggg ---hhhhh
  a=Module ID unsigned
  e=Board-0 unsigned /e:BoardType
  f=Board-1 unsigned /e:BoardType
  g=Board-2 unsigned /e:BoardType
  h=Board-3 unsigned /e:BoardType
```

Anhang A Referenzen und weiterführende Literatur

- MU-TC1-CAN Hardware Handbuch
- PPCAN-Editor 2 Online-Hilfe
- Referenzliste der Funktionsblöcke
- PCAN-View Online-Hilfe
- PCAN-Explorer Online-Hilfe
- PPCAN-Protokoll Referenz