

# Produktinformation Xtm

## **Einleitung**

Xtm ist eine Umgebung zur Steuerung von vernetzten Rechnerknoten (z.B Steuergeräte), sowie zur Visualisierung und Modifikation innerer Zustände dieser Rechnerknoten. ausserdem ist es mit Hilfe von Xtm möglich, die Kommunikation dieser vernetzten Rechnerknoten darzustellen, zu analysieren und bei Bedarf, fehlende Rechnerknoten durch die Simulation Ihres Netzwerkverhaltens zu ersetzen, bzw. die Bestehende Kommunikation zu beeinflussen und/oder zu ergänzen.

Xtm ist gedacht, um über eine einfach zu bedienende Oberfläche schnell Einblick in vernetzte Rechnersysteme zu erhalten, sowie schnell Bedien-/Simulations-Anwendungen zu erstellen.

Xtm zeichnet sich durch niedrigen Ressourcen-Bedarf, einfache Installation und Handhabung aus.

Im Moment unterstützte Netzwerke sind CAN, LIN und - mit Einschränkungen - auch Ethernet (nur UDP). Des weiteren kann die serielle Schnittstelle des PCs und die K-Leitung (mit entsprechendem Adapter für die LAPcan Karte) zur Kommunikation mit einem externen Rechnerknoten verwendet werden.

Xtm hat einen modularen Aufbau: Auf der unteren Ebene befinden sich die Schnittstellen zu Hardware und Schnittstellen die z.B. ein bestimmtes Kommunikationsprotokoll implementieren. Diese Schnittstellen werden angelegt und parametrisiert, sind jedoch im Normalbetrieb von Xtm nicht direkt sichtbar. Sichtbar sind die Daten und Signale die von diesen Schnittstellen generiert werden.

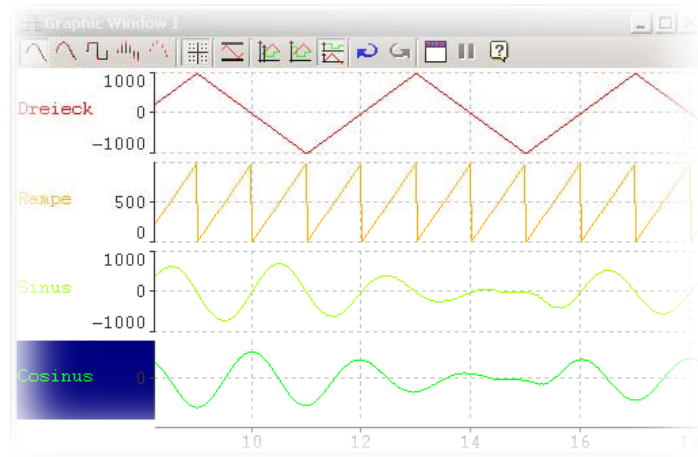
## **Eigenschaften**

- Hardware-Schnittstellen zu CAN, LIN, K-Leitung, Ethernet und Serielle Schnittstelle
- Script-Interpreter zur Simulation von Netzwerkknoten.
- Echtzeitumgebung auf Basis von Windows-Threads die mit höchster Priorität ausgeführt werden.
- Visualisierungs-Fenster als Schnittstelle zum Anwender.
- Modularer Aufbau: Neue Hardware-Schnittstellen und Visualisierungs-Fenster können mit wenig Aufwand hinzugefügt werden, sodass auf Basis von Xtm kundenspezifische Anwendungen erstellt werden können.
- Geringer Ressourcenverbrauch sowie einfache Installation.

## Grafik-/Oszilloskop-Fenster

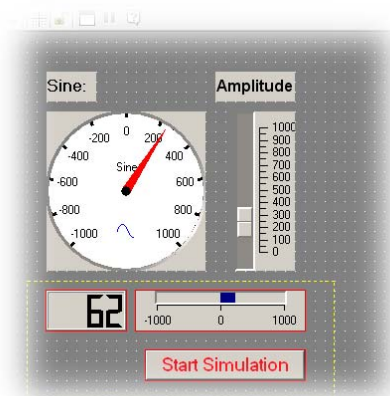
In diesem Fenster können Analoge Signale in einem Zeit-Werte-Diagramm dargestellt werden.

Eine Reihe unterschiedlicher Darstellungsmodi werden unterstützt. Zoom-, Scrollfunktionen in vertikaler und horizontaler Richtung. Mess-Cursor, Raster, Ein-/Ausblenden von Signalen.



## Panel-Fenster

Um Messgrößen zu visualisieren und Script-Programme einfach zu steuern, steht das Panel-Fenster zur Verfügung. Eine beliebige Anzahl von Panel-Fenstern kann geöffnet werden, und Anzeige- und Steuerelemente können darin frei angeordnet werden. Den Anzeige- und Steuerelementen können Signale und Aktionen des Script-Interpreters einfach zugeordnet werden.



## Script-Interpreter

Ein leistungsfähiger Script-Interpreter ist eingebaut und kann immer dann verwendet werden, wenn anspruchsvolle Aufgaben, wie z.B. Restbus-Simulation, Filterung, Verrechnung von Signalen etc. durchgeführt werden sollen. Bei dem Script-Interpreter handelt es sich um eine C-ähnliche Programmiersprache, die speziell für die Anwendung in Simulationsumgebungen optimiert wurde. Einsprungpunkte für bestimmte Ereignisse wie z.B. Timer, CAN-Botschaften, Panel-Elemente können definiert werden. Beim Eintritt eines dieser Ereignisse wird der zugehörige Programmcode ausgeführt. Script-Programme werden dem jeweiligen Hardware-Interface zugeordnet. Jedes Script-Programme wird in einem eigenen Context ausgeführt und läuft somit unabhängig von allen anderen Script-Programmen. Zur Kommunikation zwischen Script-Programmen dienen Umgebungsvariablen.

```

on action "Start Simulation" { /* Starts sending
  printf ("Start Simulation\n");
  timerStart (txTimer);
}

on timer txTimer {
  timerStart (txTimer);

  msg_ramp.Rampe.Phys = value_ramp++ % 1000;
  msg_ramp.Dreieck.Phys = value_dreieck;

  value_dreieck = value_dreieck + dir_dreieck;
  if (value_dreieck >= 1000) dir_dreieck = -1;
  if (value_dreieck <= -1000) dir_dreieck = 1;
  canSendMessage (msg_ramp);

  msg_signals.Sinus.Phys = amplitude * sin (t);
  msg_signals.Cosinus.Phys = amplitude * cos (t);
  canSendMessage (msg_signals);

  time += 1;
}

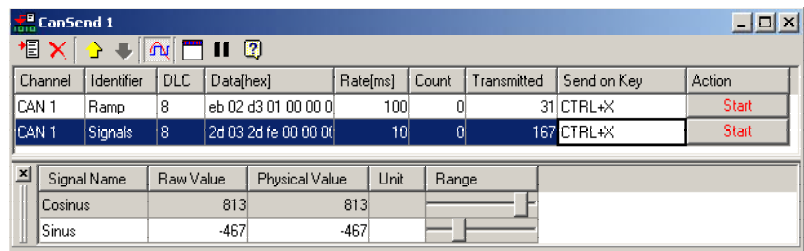
```

## CAN-/LIN-Interface

Obwohl Xtm nicht ausschliesslich auf CAN/LIN ausgerichtet ist, liegt ein grosser Schwerpunkt auf der CAN-/LIN-Funktionalität. Xtm stellt spezielle Schnittstellen für CAN und LIN zur Verfügung. DBC- und LDF-Dateien, die den CAN-/LIN-Datenverkehr beschreiben können geladen werden. Spezielle Einsprungspunkte im Sccript-Interpreter ermöglichen eine Reaktion auf ankommende CAN- und LIN-Botschaften sowie das Dekodieren der Botschaften auf Basis der DBC- und LDF-Dateien.

### CAN-Sende-Fenster

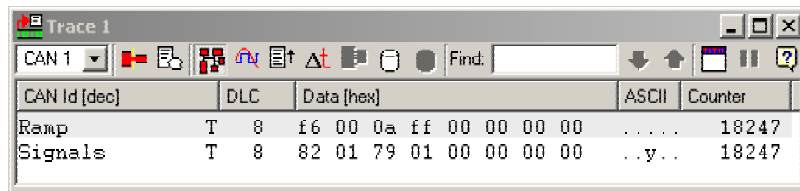
Um auf einfache Art und Weise CAN-/LIN- Botschaften auf den angeschlossenen Bus zu versenden, kann das CAN-Sende-Fenster eingesetzt werden. Das Versenden kann auf unterschiedliche Art und Weise durchgeführt werden. Es kann auf Grund eines Pushbuttons, eines Trigger-Ereignisses oder über eine Tastenkombination gesendet werden.



Botschaften können einmalig oder zyklisch (bis zu 1ms Auflösung) gesendet werden. Die Sende-Daten können im Hexadezimalformat oder auf Grund der DBC-/LDF-Beschreibungen auf Signalebene manipuliert werden.

### CAN-Trace-Fenster

Zur Darstellung von CAN-/LIN-Botschaften steht das CAN-Trace-Fenster zur Verfügung. Es ermöglicht die chronologische, sowie eine nach Identifier sortierte Darstellung der Botschaften. Botschaften können zur übersichtlichen Darstellung gefiltert werden. Bei Bedarf können die Daten auch in eine Datei geschrieben werden. Nach dem Ende der Messung ist eine Offline-Analyse möglich.



## Systemvoraussetzungen

Empfohlene Grundvoraussetzung ist ein PC mit dem Betriebssystem WindowsNT, Windows2000 oder WindowsXP. Auf Rechnern mit den Betriebssystemen Windows98/ME kann Xtm zwar betrieben werden, allerdings ist das Zeitverhalten gegenüber den Systemen mit WindowsNT als Basis etwas schlechter. Ausserdem ist es für den Betrieb der Demo-Version von Xtm erforderlich, die CAN-Treiber für die Kvaser CAN-Hardware vor dem Start von Xtm zu installieren.

Auf Rechnern mit einem auf WindowsNT basierenden Betriebssystem, können die Demo-Anwendungen von Xtm mit CAN ohne die vorhergehende Installation der Kvaser CAN-Treiber erfolgen, da Xtm bei Bedarf automatisch den Treiber für die virtuellen (simulierten) CAN-Kanäle lädt.

In Abhängigkeit der konkreten Aufgabe die mit Xtm durchgeführt werden soll, ist weitere Hardware Voraussetzung:

- CAN-Hardware der Firma Kvaser AB (LAPcan, USBcan, PCican, DRVlin, etc.)
- Serielle Schnittstelle.
- Netzwerk-Interface.

An CAN-Hardware werden zur Zeit die CAN-Adapter der Firma Kvaser AB, Schweden unterstützt. In Abhängigkeit der installierten Treiber ist auch der Betrieb mit der CANcardX und CANcardXL der Firma Vector Informatik GmbH möglich.