


 Kostenloser Download

PC-USB-I²C-Interface – das Test- und Entwicklungstool

Anhand der quelloffenen Demosoftware für das USB-I²C-Interface lässt sich die Ansteuerung von I²C-Modulen mithilfe des Interfaces von einem Computer aus leicht erlernen. I²C ist ein Bussystem, das aufgrund der Vielzahl der angebotenen Module und der einfachen Verwendung attraktiv für Anwender zur Integration in eigene Projekte ist.

I²C-Module einfach integriert

Mithilfe des universell einsetzbaren USB-I²C-Interfaces von ELV lassen sich mit entsprechender Software Bausteine, die über den I²C-Bus kommunizieren, einfach mittels USB-Schnittstelle und Computer ansprechen und beispielsweise als Anzeige oder zum Messen nutzen. Die Demoanwendung wurde entwickelt, um Anwendern eine Integration von I²C-Modulen in ihre eigenen Anwendungen zu erleichtern, da man sich hier schnell anhand des veröffentlichten Quelltextes einarbeiten kann.

Installation

Um das USB-I²C-Interface an einem Computer zum Programmieren zu nutzen, muss es zuvor installiert werden. Das Gerät nutzt den USB-Baustein „CP2102“ der Firma Silicon Labs (SiLabs). Die benötigten Treiber sind auf der Seite des Herstellers oder auf der Produktseite des USB-I²C-Interfaces im ELV-Web-Shop [1] zu finden. Es werden Treiber für Windows (2000, XP, Server 2003, Vista, 7, 8), Mac OS X und Linux (2.6.x und 2.4.36 Kernel) zum Download angeboten. Bei der Installation der Treiber ist zu beachten, dass das Gerät erst nach erfolgreicher Installation mit dem Computer verbunden werden darf. Eine bebilderte, schrittweise Installation auf einem Computer mit Windows-Betriebssystem findet sich im Handbuch des Geräts,

welches ebenfalls auf der genannten Produktseite zum Download bereitsteht. Das Interface wird mit dem mitgelieferten Mini-USB-Kabel mit dem Computer verbunden.

Verbindungsaufbau zwischen Computer und Interface

Die Kommunikation mit dem USB-I²C-Interface und den jeweils angeschlossenen I²C-Geräten erfolgt über einen virtuellen COM-Port. Für eine erfolgreiche Verbindung sind die in [Tabelle 1](#) aufgeführten Kommunikationsparameter notwendig. Hierbei handelt es sich um die eingestellten Parameter bei Auslieferung des Geräts. Wenn zwischenzeitlich Änderungen an diesen vorgenommen wurden, muss dies entsprechend beim Verbindungsaufbau berücksichtigt werden. Für einen Verbindungstest lässt sich ein beliebiges Terminal-

Standardkommunikationsparameter für die Verbindung mittels virtuellen COM-Ports

Tabelle 1

Baudrate	115.200
Datenbits	8
Parität	none/keine
Stoppbits	1

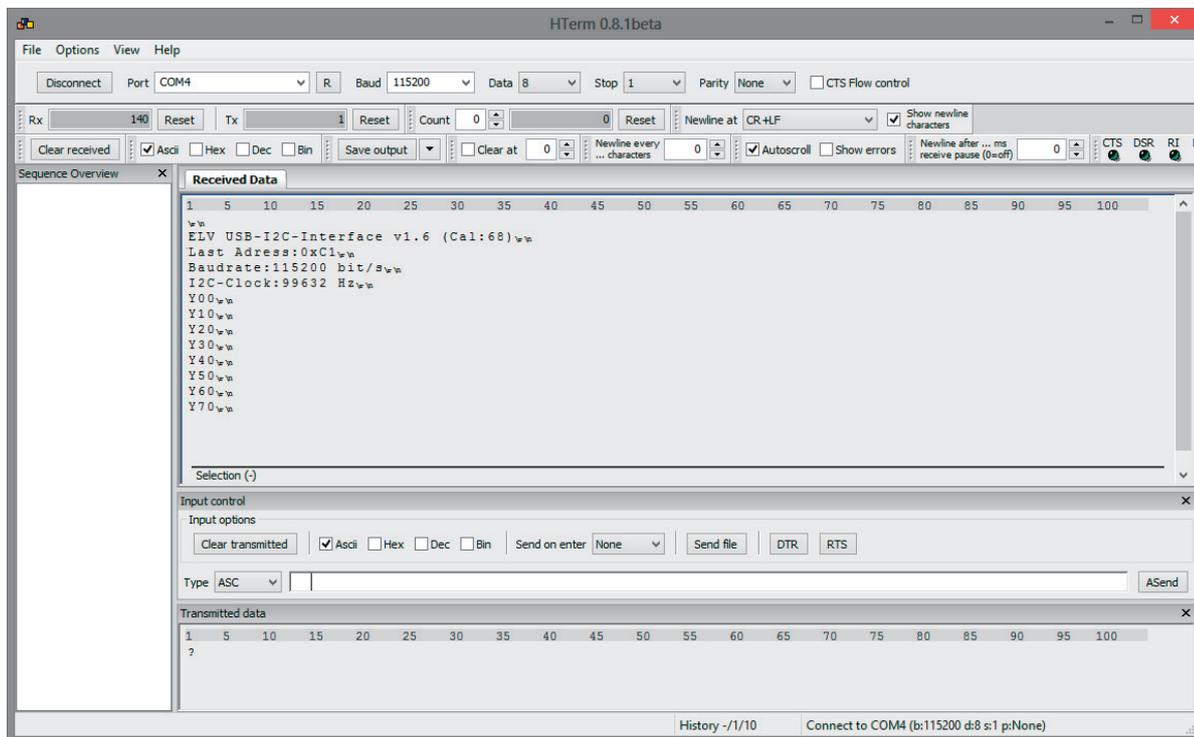


Bild 1: Erfolgreich hergestellte Verbindung zwischen Computer und USB-I²C-Interface mittels „HTerm“

Programm verwenden, beispielsweise das von Tobias Hammer entwickelte „HTerm“ [2], welches privat und kommerziell kostenfrei genutzt werden darf. Nach Übernahme der jeweiligen Kommunikationsparameter antwortet das USB-I²C-Interface bei korrekten Verbindungseinstellungen nach Senden des Befehls „?“ wie in Bild 1 dargestellt.

Das Kommunikationsprotokoll

Das USB-I²C-Interface lässt sich mittels an das Interface gesendeter Zeichenketten steuern, die das Interface anschließend in I²C-Befehle für die daran ange-

schlossenen Geräte übersetzt. Dies sorgt dafür, dass sich die Geräte mittels eines ASCII-basierten Protokolls steuern lassen und sich der Programmierer nicht mit der Kommunikation auf dem I²C-Bus bzw. deren Ablauf beschäftigen muss.

Die grundlegenden Befehle für die Steuerung der I²C-Geräte sind in Tabelle 2 zu finden. Das Interface unterstützt neben Befehlen zur Steuerung auch jene zum Kommentieren von Befehlen bzw. zur Konfiguration, womit sich beispielsweise Befehlsketten in Form von Makros zum Abruf im Gerät selbst abspeichern lassen. Eine detaillierte Auflistung aller Befehle mit

Befehle für die I²C-Kommunikation

ASCII-Zeichen	Folgebyte(S) *	Funktionsbeschreibung
S		initiiert Start-Ereignis auf I ² C-Bus
	7-Bit-Adresse + Write-Bit (0) + Datenbyte(s)	nachfolgende Daten ins adressierte I ² C-Gerät schreiben (geringwertigstes Bit [LSB] des Adressbytes muss 0 sein)
	7-Bit-Adresse + Read-Bit (1) + Byteanzahl	Daten aus dem adressierten I ² C-Gerät lesen, dem Adressbyte folgt die Anzahl der zu lesenden Bytes (geringwertigstes Bit [LSB] des Adressbytes muss 1 sein)
P		initiiert Stopp-Ereignis auf I ² C-Bus (Bus im Leerlauf = idle)
W	Byte1 Byte2 Byte3...	schreibt „Byte1, Byte2, Byte3...“ ins zuletzt adressierte I ² C-Gerät
R	Byteanzahl	liest Datenbytes (1...255) aus dem zuletzt adressierten I ² C-Gerät
:		wartet mit der Ausführung der nachfolgenden Befehle bis zum nächsten Zeilenumbruch (0x0D oder 0x0A); ist sinnvoll, wenn das Terminal-Programm jedes Zeichen sofort nach der Eingabe überträgt; Ausführung erst nach Abschluss mit Eingabetaste
L	Byte 1 Byte 2	fügt eine Wartepause von 1 bis 65.535 ms (0001...FFFF) in Hex-Schreibweise (16 Bit) in die Befehlsausführung ein (z. B. innerhalb von Makros)
N		lässt Master nach letztem gelesenem Byte mit NACK antworten, wenn die automatische NACK-Antwort mit Y21 deaktiviert wurde

Tabelle 2

*Jedes Byte (hexadezimal) wird mit 2 ASCII-Zeichen geschrieben, z. B. 0x1F = 1F

Erläuterung eines Befehls an das I²C-Interface

S 34	Der folgende Befehl ist an das I ² C-Gerät mit der Slave-Adresse 0x34 gerichtet.
28	Das Register 0x28 ist von der gewünschten Operation betroffen.
R 01	Es soll ein Byte aus dem Register gelesen werden.
P	Initiiere Stopp-Ereignis auf dem I ² C-Bus.

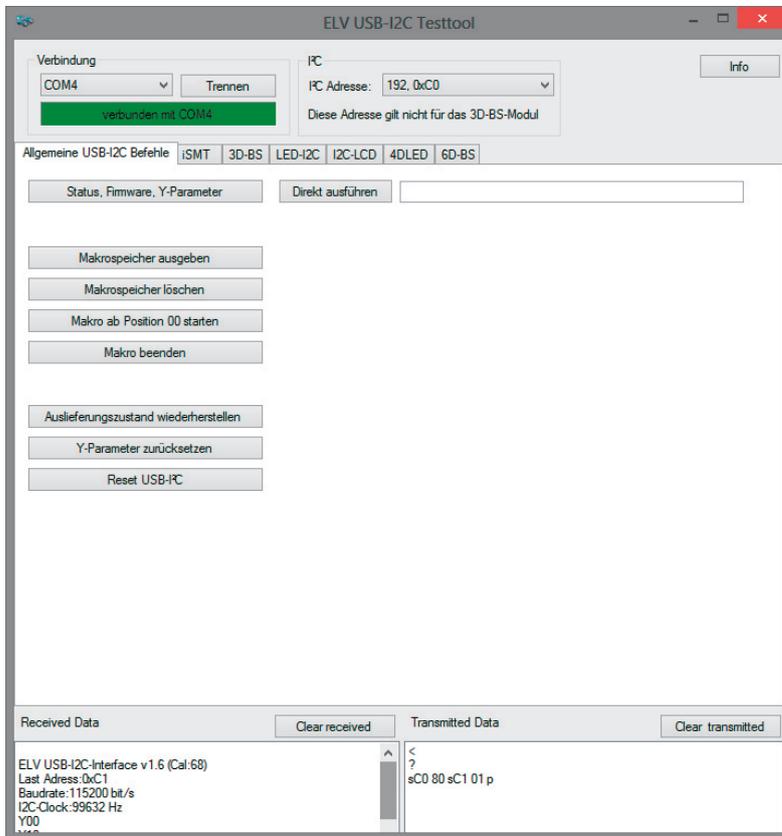


Bild 2: Die Demoanwendung für das USB-I²C-Interface

Unterstützte I²C-Module

iSMT	Es handelt sich um ein Schrittmotor-Treibermodul. Die Software besitzt entsprechende Benutzeroberflächen zum Erhalten und Setzen der Motorparameter, zum Bedienen des Motors sowie zum Erstellen eines Programmablaufs mittels der Makrofunktion.
3D-BS	Für den 3-Achsen-Beschleunigungssensor 3D-BS wird eine Abfrage von Funktionsparametern und das Setzen der Bandbreitenfrequenz sowie der Sensitivität zur Verfügung gestellt.
LED-I ² C	Das LED-I ² C bietet sich mit seinen 16 LEDs als Anzeige für vielerlei Anwendungszwecke an. Es werden eine Verwaltung der Funktionsparameter, das Setzen einzelner oder mehrerer LEDs, die Auswahl der jeweiligen Helligkeitswerte als auch das Zusammenstellen von Makro bereitgestellt.
I ² C-LCD	Dieses Modul wird primär als Anzeigemodul genutzt, es unterstützt sowohl die Darstellung von Zahlen als auch von Maßeinheiten. Mit der Software lässt sich das Display ansteuern, die LEDs setzen und die angebrachten Taster abfragen.
4DLED	Dieses Modul besteht aus einer 7-Segment-Anzeige mit jeweiligem Punkt und einem verbauten Temperatursensor. In der Demosoftware lässt sich sowohl zyklisch das Display ansteuern als auch die Temperatur messen.
6D-BS	Der 6-Achsen-Beschleunigungssensor und dessen Integration in die Demoanwendung werden in dieser Ausgabe vorgestellt.

jeweiligen Beispielen und weiteren Hinweisen findet sich im Handbuch des USB-I²C-Interfaces.

Für eigene Anwendungen sind primär die Befehle „S“ zum Starten des Ereignisses, „W“ zum Schreiben von Bytes in Register des Geräts, „R“ zum Auslesen von Registern und „P“ zum Initiieren des Stopp-Ereignisses von Bedeutung, um mit I²C-Geräten zu kommunizieren.

Um beispielsweise das Register 0x28 eines Geräts mit den Slave-Adressen 0x34/0x35 (jeweils zum Lesen und Schreiben, siehe Befehlstabelle) auszulesen, ist der Befehl „S 34 28 R 01 P“ einzusetzen. Der Aufbau des Befehls ist in [Tabelle 3](#) erläutert.

Auch Schreiboperationen gehen aufgrund des ASCII-Protokolls einfach von der Hand, der Befehl für das Schreiben des Bytes 0xAB in das gleiche Register des vorangehenden Geräts lautet „S34 28 AB P“. Die Demoanwendung wickelt anhand dieses Befehlsusters einen Großteil der Gerätekommunikation ab.

Für eine vereinfachte Kommunikation kann, da aus der Geräte-Adresse anhand des Zustands des geringwertigsten Bits hervorgeht, dass eine Schreiboperation erfolgen wird, auf den Befehl „W“ verzichtet werden. Ebenso kann auch für Leseoperationen die Geräteadresse zum Schreiben verwendet werden, das Interface korrigiert dies beim Übersetzen in I²C-Befehle automatisch.

Features der Demoanwendung

Auch die Demoanwendung ([Bild 2](#)) steht auf der Produktseite des USB-I²C-Interfaces zum kostenfreien Download bereit. Sie wurde mit der Programmiersprache C# für Windows-Betriebssysteme entwickelt, für die Benutzeroberfläche wurde Windows Forms genutzt. In der linken oberen Ecke lässt sich der für die Verbindung notwendige COM-Port einstellen. Das Handbuch des Interfaces beschreibt mithilfe einer bebilderten Anleitung den Weg, wie sich auf einem Windows-Betriebssystem der COM-Port des Interfaces anhand des Geräte-Managers ermitteln lässt. Nach erfolgreicher Verbindung wird dem Benutzer dies visuell durch eine grüne Markierung des Verbindungszustands dargestellt. Die Demoanwendung benutzt die in [Tabelle 1](#) dargestellten Kommunikationsparameter.

Angeboten wird von der Anwendung die Nutzung von allgemeinen USB-I²C-Befehlen auf der ersten Registerseite, die unabhängig von den angeschlossenen I²C-Modulen funktionieren. Auch hält die Demoanwendung für sechs verschiedene I²C-Module von ELV entsprechende gerätespezifische Befehle und Nutzungsmöglichkeiten parat ([siehe Tabelle 4](#)). Alle ausgehenden Befehle, aber auch die Antworten von Geräten werden in den jeweiligen Listen im unteren Bereich der Anwendung erfasst. Auf der ersten Registerseite lassen sich der aktuelle Status, Firmware und Y-Parameter des Interfaces ausgeben, der Makrospeicher verwalten als auch ein I²C-Befehl direkt ausführen.

Diese Aufgaben erfüllt die Anwendung primär durch das Abfragen des aktuellen Zustands der Benutzeroberfläche und das Zusammensetzen von entsprechenden Befehlen. Für die Kapselung der Verbindung über den virtuellen COM-Port wurde die eigene Klasse „UsbI2C“, die die Kommunikation mit dem USB-I²C-

Interface verwaltet und Methoden zum Senden von Befehlen und Empfangen von Antworten anbietet, verwendet. Eine Erweiterung dieser Demoanwendung lässt sich durch das Hinzufügen einer weiteren Registerkarte und der Nutzung der zuvor beschriebenen Klasse leicht bewerkstelligen.

Integration von I²C-Modulen in eigene Projekte

Das USB-I²C-Interface bietet aufgrund des genutzten ASCII-basierten Protokolls und der Kapselung der Kommunikation mit dem I²C-Bus eine attraktive Schnittstelle, um I²C-Module in eigene Projekte ohne großen Mehraufwand zu integrieren.

Drei von den sechs in der Demoanwendung aufgeführten I²C-Modulen lassen sich als externe Anzeige nutzen, beispielsweise zur Darstellung von aktuellen Statusinformationen oder von Messergebnissen. Hierfür bietet sich insbesondere das I²C-LCD an, da es neben vier 15-Segment-Anzeigen auch eine Vielzahl unterschiedlicher Maßeinheiten darstellen kann. Ebenfalls lassen sich angeschlossene Geräte über die Taster des Moduls steuern, die Abfrage aller vier angebrachten Taster ist mittels eines einzigen ASCII-Befehls möglich. Die montierten LEDs können als zusätzliche Statusinformation zu den dargestellten Informationen dienen.

Beide Beschleunigungssensoren können als Bausteine für die Realisierung von Projekten mit Bewegungssensorik, beispielsweise dem Bau eines eigenen Quadropters oder einer Bewegungssteuerung, dienen. Hier erleichtert eine eigene Testansteuerung insbesondere die langwierige und feinfühligte Abstimmung des Sensors auf den restlichen Aufbau.

Im Allgemeinen kann das Interface die Realisierung und das Testen von Applikationen für I²C-Module erheblich erleichtern. Auf der einen Seite können über die Demosoftware leicht Befehle für ein einzelnes Modul erstellt und versendet werden, auf der anderen Seite kann die Funktionsweise des Moduls mittels einer eigenen softwarebasierten Testansteuerung ohne

großen Aufwand überprüft werden.

Außerdem ist die Nutzung des USB-I²C-Interfaces als Ansteuerung für einen Schrittmotor mittels des Moduls „iSMT“ (siehe Tabelle 4) möglich. Das Treibermodul wird hierbei im Lauf der Installation auf einen Schrittmotor montiert. Nun kann nach Einstellung von auf das Gerät abgestimmten Motorparametern die Ansteuerung mit einer eigenen Anwendung beginnen. Mithilfe der Demoanwendung können leicht die Befehle für die richtigen Motoreinstellungen als auch zum Starten und Stoppen des Motors ermittelt werden und letztlich für eine eigene Anwendung genutzt werden.

Diese Konstruktion lässt sich beispielsweise während des Testens von Rauchmeldern in einem Rauchkanal einsetzen (Bild 3). Ein Rauchkanal ist eine Vorrichtung zum Testen von Rauchmeldern, wobei die eingelassenen Rauchmelder einer stetig steigenden Konzentration von Rauch ausgesetzt sind. Es wird gemessen, ab welcher Konzentration der Rauchmelder auslöst. Hierfür ist es vonnöten, alle relevanten Positionen/Winkel zu testen bzw. den Rauchmelder entsprechend zu drehen, um sicherzustellen, dass eine vollständige Funktionalität des Rauchmelders aus jeder möglichen Position gegeben ist. Der Testdurchlauf soll weitestgehend automatisiert werden, was eine automatische Ansteuerung von Rauchkanal und Drehvorrichtung des Rauchmelders nötig macht. Hier kommt der Schrittmotor zum Einsatz, die Drehung erfolgt automatisiert im Kanal zwischen zwei Testdurchläufen nach einer angemessenen Anzahl von Wiederholungen.

ELV



Weitere Infos:

- [1] www.elv.de/usb-i2c-interface-komplettbausatz-inkl-gehaeuse-bearbeitet-und-bedruckt-usb-kabel-3-anschlusskabel.html
- [2] www.der-hammer.info/terminal/

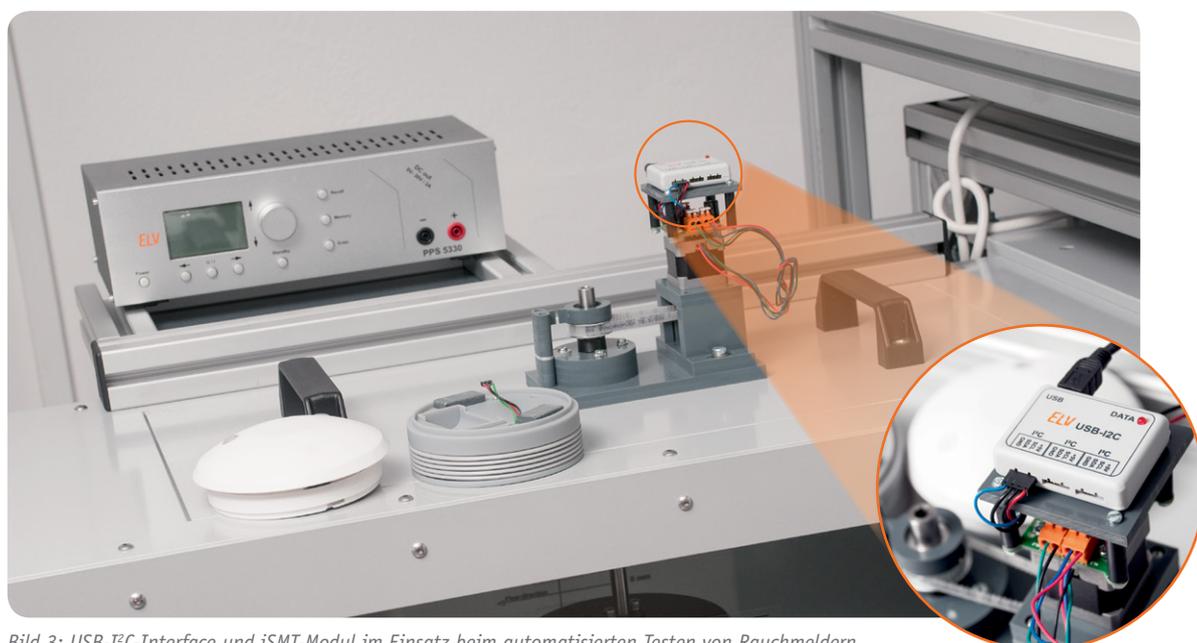


Bild 3: USB-I²C-Interface und iSMT-Modul im Einsatz beim automatisierten Testen von Rauchmeldern