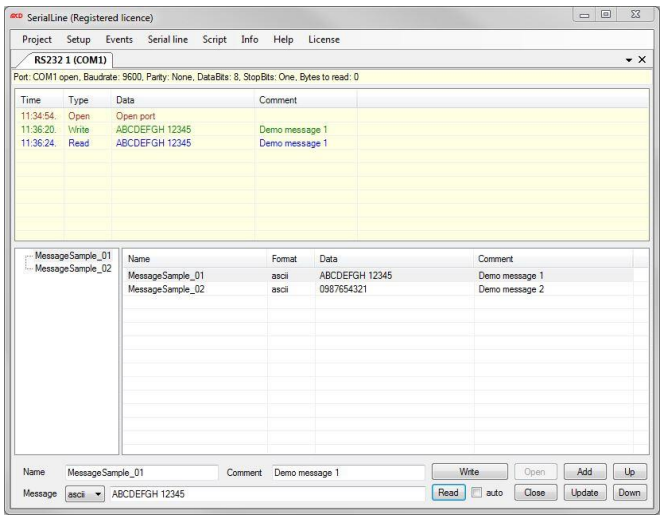
Bedienungsanleitung

Toolmonitor SerIO



GET IN touch WITH SENSITIVE TESTING

Softline

Modline

Conline

Boardline

Avidline

Pixline

Applikation

MCD Elektronik GmbH
Hoheneichstr. 52
75217 Birkenfeld
Telefon 0 72 31/78 405-0
Telefax 0 72 31/78 405-10
info@mcd-elektronik.de
www.mcd-elektronik.de
Sitz: Birkenfeld
Geschäftsführer: Bruno Hörter
Registergericht Mannheim
HRB 505692

Vorlagenversion: 5.4 / 2018-04-09
V1.1 2018-09 04 TI (TS)

Inhalt

1. ALLGEMEIN 6

2. INSTALLATION VON SOFTWARE 7

2.1. VORAUSSETZUNGEN 7

2.2. LIZENZ 7

2.3. REGISTER COM - SERVER 9

3. EINFÜHRUNG IN DEN TOOLMOINTOR SERIO 9

4. WINDOWS 11

4.1. SCAN SERIO DEVICES 11

4.2. RS232 - KOMMUNIKATION 11

4.3. SERIO - GERÄTE 12

4.3.1. MechIO 8 13

4.3.1.1. I2C - Kommunikation 13

4.3.1.2. MechIO 8 Analog Input 13

4.3.1.3. MechIO 8 Digital Input 14

4.3.1.4. MechIO 8 Digital Output 14

4.3.1.5. SerIO - Kommunikation 15

4.3.2. MechIO 16 16

4.3.2.1. I^2C - Kommunikation 16

4.3.2.2. MechIO 16 Analog Input 16

4.3.2.3. MechIO 16 Digital Input 17

4.3.2.4. MechIO 16 Digital Output 17

4.3.2.5. MechIO 16 Trigger 18

4.3.2.6. SerIO - Kommunikation 19

4.3.3. Adapter CPU 19

4.3.3.1. I^2C - Kommunikation 19

4.3.3.2. SerIO - Kommunikation 20

4.3.4. Pass / Fail - Anzeige 21

4.3.4.1. I^2C - Kommunikation 21

4.3.4.2. Pass / Fail Digital Output 22

4.3.4.3. SerIO - Kommunikation 22

4.4. I^2C - GERÄTE 23

4.4.1. I^2C Digitale IO - Geräte (PIO) 23

4.4.1.1. I²C - Kommunikation ... 24
4.4.1.2. PIO Digital IO ... 24
4.4.2. Module zur Stromversorgung ... 25
4.4.2.1. I²C - Kommunikation ... 25
4.4.2.2. Stromversorgung Analoganzeige Spannung ... 26
4.4.2.3. Stromversorgung Analoganzeige Strom ... 26
4.4.2.4. Stromversorgung Einstellungen ... 27
4.4.2.5. Stromversorgung Messung ... 27

5. KONFIGURATIONSBEISPIEL (SCHNELLSTARTANLEITUNG) ... 28
5.1. SAMPLE - PRESET AKTIVIEREN ... 28
5.2. ANGESCHLOSSENE GERÄTE SUCHEN ... 28
5.3. RS232 - PORT KONFIGURIEREN ... 29
5.4. SERIO - GERÄTE KONFIGURIEREN ... 30
5.5. I2C - GERÄT (RELMUX16) KONFIGURIEREN ... 30
5.6. ANZEIGE DER DIGITALEN UND ANALOGEN EIN - UND AUSGÄNGE ... 31

6. PROJEKTVERWALTUNG ... 32

7. EVENTS ... 33

8. SETUP ... 34
8.1. RS232 - PORTS ... 34
8.1.1. Anzahl der RS232 - Ports ... 34
8.1.2. Kommunikationsparameter ... 35
8.2. SERIO - GERÄTE ... 36
8.2.1. MechIO 8 ... 36
8.2.1.1. I²C ... 37
8.2.1.2. MechIO 8 Analog Input ... 37
8.2.1.3. MechIO 8 Digital Input ... 38
8.2.1.4. MechIO 8 Digital Ouput ... 39
8.2.2. MechIO 16 ... 40
8.2.2.1. I²C ... 40
8.2.2.2. Mech IO 16 Analog Input ... 40
8.2.2.3. MechIO 16 Digital Input ... 41
8.2.2.4. MechIO 16 Digital Output ... 42
8.2.2.5. MechIO 16 Trigger ... 42
8.2.3. Adapter CPU ... 43

8.2.3.1. I²C 43
8.2.4. Pass / Fail - Anzeige 44
8.2.4.1. I²C 44
8.2.4.2. Pass / Fail Digital Output 44
8.3. I²C 45
8.3.1. I²C Digitale IO Geräte (PIO) 46
8.3.1.1. PIO Digital IO 46
8.3.2. I²C - Module zur Stromerzeugung 47
8.3.2.1. Power Supply für Tuner Applications Master 48
8.3.2.2. Power Supply für Tuner Applications Slave 48
8.3.2.3. Power Supply für Tuner Applications Linear Slave 49
8.4. GENERAL 49
8.5. LOGGER 50
8.5.2. Log - Level 51
8.5.3. Trace - Level 51
8.5.4. Log to File 51
8.5.5. Log Monitor Background Update 51
8.5.6. Log Errors Always to File 51
9. PROGRAMMIERUNG 52
9.1. ANALOGINPUT 52
9.1.1. GetValueNames 52
9.1.2. SetValueNames 52
9.2. DIGITALINPUT 53
9.2.1. GetValuesNames 53
9.2.2. SetValueNames 53
9.3. DIGITALOUTPUT 54
9.3.1. GetValueNames 54
9.3.2. SetEventNames 54
9.3.3. SetValueNames 54
9.4. I²CBASELINE 54
9.5. I²CDIGITALIO 55
9.5.1. GetValueNames 55
9.5.2. SetEventNames 55
9.5.3. SetValueNames 56
9.6. I²CLINE 57

9.7. LINE 57

9.7.1. VirtualInterfaceCommands 57

9.8. RS232 58

9.9. SERIOBASELINE 58

9.10. SERIOLINE 58

9.11. TRIGGER 58

9.11.1. SetEventNames 58

9.11.2. SetValueNames 59

1. Allgemein

Die Hauptbereiche dieser Dokumentation sind die „Funktionen“ und „Programmierung“. Unter „Funktionen“ sind Beschreibungen zu den einzelnen Modulen der Software verfügbar. Unter dem Punkt „Programmierung“ sind die Schnittstellen, für den Anwender dieser Software, in tabellarischer Form aufgeführt. Im Interface sind die Funktionen für den Zugriff auf den Toolmonitor dokumentiert und in den jeweiligen Modulen sind die gültigen Übergabeparameter für diese Funktionen angegeben.

Der Tolmonitor SerIO basiert auf dem MCD Framework für Toolmonitore. Die Grundfunktionen des Frameworks, wie das Logging, allgemeine Einstellungen und die Lizensierung sind hier ebenfalls beschrieben.

Weiterführende Informationen zum Framework der Toolmonitore befinden sich in der allgemeinen Hilfe bzw. Beschreibung des MCD Frameworks (General Help). Diese beinhaltet insbesondere die Beschreibung der „Script-Engine“, das „Virtuelle Interface“ und die benutzerspezifische „Visualisierung“.

Bestellnummer: # 150419

2. Installation von Software

2.1. Voraussetzungen

- Betriebssystem: Windows XP® - Windows 8.1®
- Architektur: 32 Bit oder 64 Bit
- .Net Framework: Version 3.0

Zur Installation des MCD Toolmonitors SerIO ist es ausreichend, die „SerIOMonitor.exe“ in ein beliebiges Verzeichnis auf dem Zielsystem zu kopieren.

Alternativ kann der bereitgestellte Installer („SerIOInstall.msi“) ausgeführt werden.

2.2. Lizenz

Um die Software vor nicht autorisierter Benutzung zu schützen, ist es erforderlich, den Toolmonitor nach der Installation zu lizenzieren.

Zu Demonstrations - und Testzwecken kann der Toolmonitor für jeweils 30 Minuten auch ohne Lizenz betrieben werden. Einige Programmfunktionen sind dabei deaktiviert. Zur Überbrückung der Wartezeit auf eine Freischaltung (z.B. am Wochenende) kann auch eine 24h Kurzzeitlizenz aktiviert werden.

Zur Freischaltung des Toolmonitors öffnen Sie bitte den Dialog „Lizenzverwaltung“ über den Menüpunkt **License → Register**.

Abbildung 1: Dialog zum Registrieren aufrufen

1.) In dem Dialog „Aktuelle Lizensierung“ wird der Status Ihrer aktuellen Lizenz angezeigt:

Abbildung 2: Lizenzstatus abrufen

2.) Um eine permanente Lizenz für Ihre Software anzufordern gehen Sie bitte wie folgt vor:

- Wählen Sie die Registerkarte „Lizenz anfordern“ aus
- Geben Sie die Anzahl der benötigten Lizenzen (für Ihren PC) im Feld „Anzahl Lizenzen“ an
- Betätigen Sie die Schaltfläche „Anforderungsdatei erzeugen“
- Nun öffnet sich ein weiteres Fenster, in welchem Sie aufgefordert werden, die „MCD Licenser Request“ - Datei (*.mlr) abzuspeichern
- Bitte speichern Sie diese Datei ab und sende diese dann per E-Mail an info@mcd-elektronik.de.
  Bitte geben Sie eine Auftrags - oder Projektnummer an, um die Zuordnung zu erleichtern
- Im Anschluss daran erhalten Sie von MCD Elektronik eine E-Mail mit Ihrer Lizenzdatei („MCD License Key *.mlk“) im Anhang
- Als Letztes speichern Sie diese Datei entweder unter „C:\Windows“ oder in dem Verzeichnis, in dem die „.exe“ - Datei Ihrer Software installiert wurde
- Nach dem nächsten Start Ihrer Software steht diese dann mit vollem Funktionsumfang zur Verfügung

Abbildung 3: Permanente Lizenz anfordern

3.) Um eine Kurzeitlizenz (24h) zu aktivieren, wählen Sie bitte die Registerkarte „Kurzzeitlizenz“ aus. Danach geben Sie bitte die Zahlenreihe aus dem linken Fenster in das rechte Fenster ein. Falls Sie die Zahlen nicht erkennen können, betätigen Sie die Schaltfläche „Neue Zahl“, um eine neue Zahl zu erhalten. Haben Sie die Zahl richtig eingegeben, können Sie die Kurzzeitlizenz über die Schaltfläche „Lizenz aktivieren“ freischalten. Bitte beachten Sie, dass die Kurzzeitlizenz erlischt, sobald Sie die Software beenden. Sie können die Kurzzeitlizenz jedoch beliebig oft erneut aktivieren.

Abbildung 4: Kurzzeitlizenz anfordern

2.3. Register COM - Server

Über diesen Befehl kann der Toolmonitor als COM - Server registriert werden. Dies ist notwendig, wenn der Toolmonitor durch andere Programme, wie z.B. dem MCD TestManager ferngesteuert werden soll.

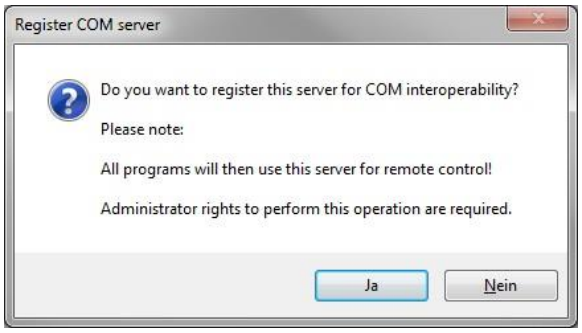


Abbildung 5: Register COM - Server

3. Einführung in den Toolmointor SerIO

Der Toolmonitor SerIO dient der Steuerung und Visualisierung der MCD Produkte und unterlagerter Geräte, welche über das MCD standardisierte SerIO - Protokoll angesprochen werden.

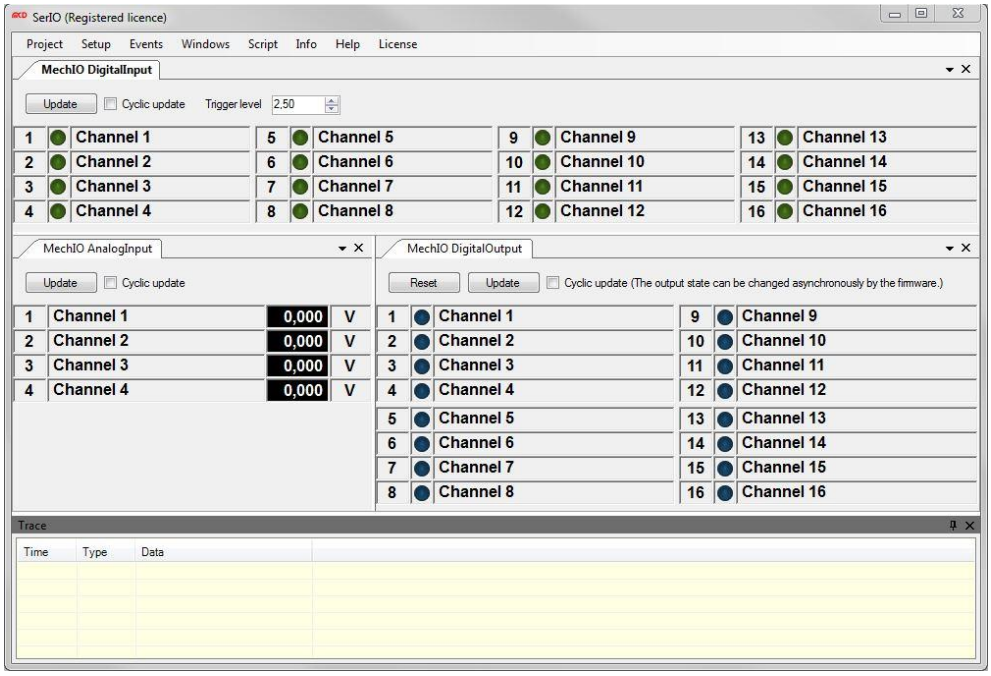


Abbildung 6: Toolmonitor SerIO

Folgende MCD Produkte mit SerIO - Protokoll werden unterstützt:

- Unknown (stellt Grundfunktionen für unspezifische Anforderungen zur Verfügung)
- MechIO 8
- MechIO 16
- Adapter CPU
- Pass / Fail - Anzeige

Folgende unterlagerte Geräte werden unterstützt:

- Unknown (stellt Grundfunktionen für unspezifische Anforderungen zur Verfügung)
- PCF8574 | PIO 8 channels per port
- PCF8574A | PIO 8 channels per port
- PCA9554 | PIO 8 channels per port
- PCA9554A | PIO 8 channels per port
- PIO 3 x PCA 9554 | B.140.010.110
- PIO 3 x PCA 9554A | B.140.010.210
- PIO 3 x PCF 8574 | B.140.010.310
- PIO 3 x PCF 8574A | B.140.010.410
- RelMUX16 | Relais multiplexer 16 channels 7442
- PST30M | Power supply for tuner applications master B.170.030.112
- PST30S | Power supply for tuner applications slave B.170.030.212
- PST30L | Power supply for tuner applications linear slave B.170.030.412

Auf einfache Art und Weise können frei definierbare oder in einer Bibliothek hinterlegte Telegramme gesendet und empfangen werden. Digitale oder analoge Ein - und Ausgänge können abgefragt oder gestellt werden.

Die Programmoberfläche kann in weitem Umfang frei gestaltet und den Anwenderanforderungen angepasst werden. Einmal erstellte Konfigurationen können in „Projektfiles“ gespeichert und bei Bedarf wieder geladen werden. Mit Hilfe einer integrierten „Script Engine“ können alle Telegramme automatisiert versendet und empfangen werden. Asynchrone Vorgänge lassen sich in den Toolmonitor auslagern. Über Fremdsoftware kann der Toolmonitor komplett ferngesteuert werden.

Als Interface kommt hierbei COM / DCOM oder eine .Net - Assembly zum Einsatz. Dadurch kann der Toolmonitor in einer Vielzahl von Applikationen eingebunden werden (Microsoft Visual Studio® (C#, C++, Visual Basic), Microsoft Office® (z.B. Excel®), Open Office®, LabVIEW®, MCD TestManager CE). Weitere Informationen hierzu befinden sich im Kapitel „Programmierung“ und in der allgemeinen Hilfe zu den Toolmonitoren (General Help).

4. Windows

Über den Menüeintrag „Windows“ im Hauptmenü des Toolmonitors SerIO kann auf alle konfigurierten Kommunikationskanäle, SerIO - Geräte und I^2C - Module zugegriffen werden.

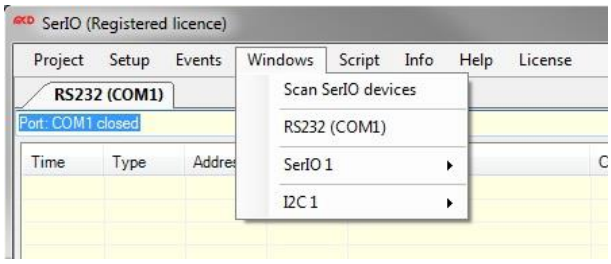


Abbildung 7: Windows - Menü

4.1. Scan SerIO Devices

Über den Menüeintrag „Scan SerIO Devices“ im Windows - Menü des Toolmonitors SerIO kann automatisch nach angeschlossenen SerIO - Geräten gesucht werden. Dabei werden zunächst alle seriellen Schnittstellen geschlossen und dann diese wiederum mit verschiedenen Baudraten und Adressen nach angeschlossenen SerIO - Geräten überprüft.

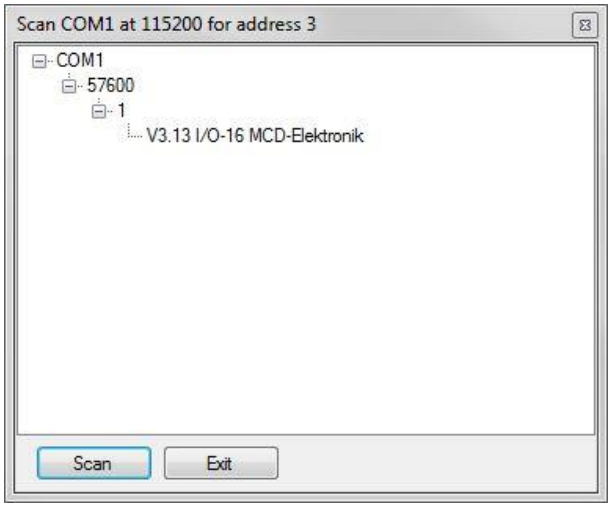


Abbildung 8: Scan SerIO Devices

4.2. RS232 - Kommunikation

Für jeden RS232 - Kommunikationskanal wird im Windows - Menü des Toolmonitors SerIO ein entsprechender Eintrag angelegt. Über diesen Eintrag, kann ein Formular zur Low - Level Analyse und Steuerung der Kommunikation über diesen RS232 - Kommunikationskanal geöffnet werden. Mit Hilfe des Formulars kann die gesamte laufende serielle Kommunikation mitgehört oder eigene Nachrichten versendet werden. Es besteht die Möglichkeit die empfangenen Daten aufzunehmen („Logging“). Weiterhin können selbst erstellte Nachrichten oder Sequenzen von Nachrichten abgespeichert und geladen werden.

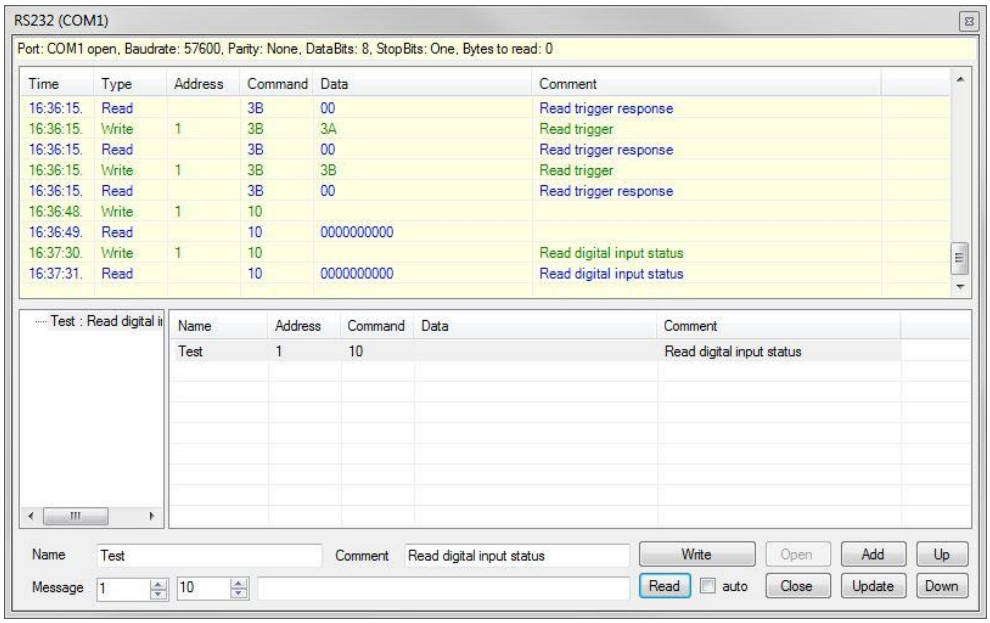


Abbildung 9: RS232 - Kommunikation

Die Kommunikation kann über die „Open“ - Schaltfläche gestartet werden. Durch das Betätigen der Schaltfläche wird der Kanal automatisch mit den Einstellungen aus den „Options“ parametriert.

Über die Eingabemaske können eigene Nachrichten angelegt werden. Hierzu kann der Nachrichtentyp und der Dateninhalt eingegeben werden.

Im Bereich „Message“ sind die „SerIO - Adresse“, das „SerIO - Kommando“ und die zugehörigen Daten einzutragen.

Über die Schaltfläche „Write“ wird die Nachricht versendet. Mit Hilfe von „Read“ können Nachrichten, sofern vorhanden, empfangen werden. Ist die Option „Auto“ aktiviert, werden eigehende Nachrichten automatisch empfangen und angezeigt.

In den Feldern „Name“ und „Comment“ können weitere Informationen zu der Nachricht hinterlegt und diese dann über die „Add“ - Schaltfläche gespeichert werden.

Die Kommunikation kann über die „Close“ - Schaltfläche beendet werden.

4.3. SerIO - Geräte

Für jedes SerIO - Gerät wird im Windows - Menü des SerIO - Toolmonitors ein entsprechender Eintrag angelegt. Über diesen Eintrag, könne verschiedene Formulare zur Steuerung des SerIO - Gerätes geöffnet werden.

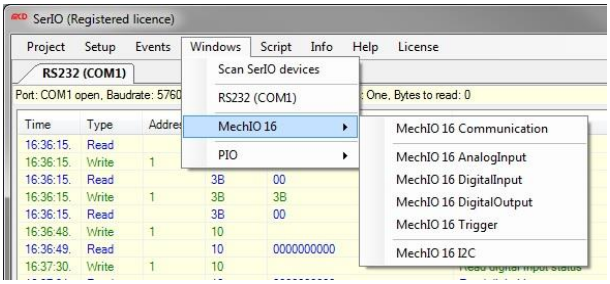


Abbildung 10: Menü für SerIO - Geräte

4.3.1. MechIO 8

4.3.1.1. I²C - Kommunikation

Mit Hilfe des Formulars kann die gesamte laufende I²C - Kommunikation des entsprechenden SerIO - Gerätes mitgehört oder eigene Nachrichten versendet werden.

Es besteht die Möglichkeit die empfangenen Daten aufzunehmen („Logging“). Weiterhin können selbst erstellte Nachrichten oder Sequenzen von Nachrichten abgespeichert und geladen werden.

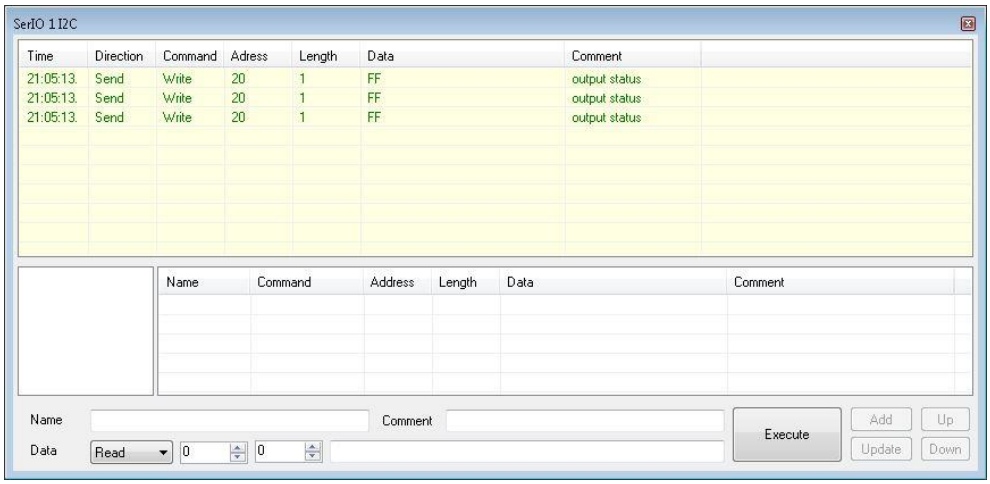


Abbildung 11: I²C - Kommunikation

Über die Eingabemaske können eigene Nachricht angelegt werden. Hierzu kann der Nachrichtentyp und der Dateninhalt eingegeben werden.

Im Bereich „Message“ sind das „I²C - Kommando“, die „I²C - Adresse“ und die zugehörigen Daten einzutragen.

Über die Schaltfläche „Execute“ wird die Nachricht versendet.

In den Feldern „Name“ und „Comment“ können weitere Informationen zu der Nachricht hinterlegt und diese dann über die „Add“ - Schaltfläche gespeichert werden.

4.3.1.2. MechIO 8 Analog Input

Hier erfolgt die Anzeige und Steuerung der analogen Eingänge der Mechanik - Steuerung. Über den Button „Update“ kann der aktuelle Zustand der Eingänge aktualisiert werden. Die Check - Box „Cyclic update“ aktiviert eine zyklische Aktualisierung der Eingänge.

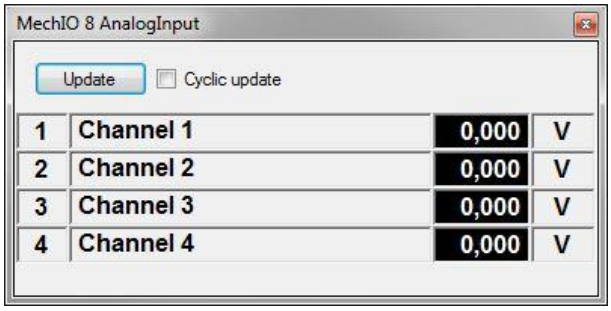


Abbildung 12: Analoge Eingänge

4.3.1.3. MechIO 8 Digital Input

Hier erfolgt die Anzeige und Steuerung der digitalen Eingänge der Mechanik - Steuerung. Über den Button „Update“ kann der aktuelle Zustand der Eingänge aktualisiert werden. Die Check - Box „Cyclic update“ aktiviert eine zyklische Aktualisierung der Eingänge. Mit Hilfe von „Trigger level“ kann die Triggerschwelle (in Volt) für die digitalen Eingänge eingestellt werden.

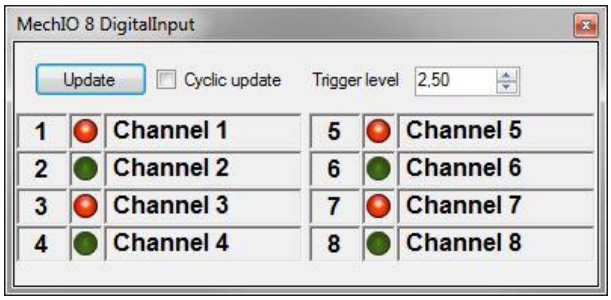


Abbildung 13: Digitale Eingänge

4.3.1.4. MechIO 8 Digital Output

Hier erfolgt die Anzeige und Steuerung der digitalen Ausgänge der Mechanik - Steuerung. Über den Button „Update“ kann der aktuelle Zustand der Ausgänge aktualisiert werden. Die Check - Box „Cyclic update“ aktiviert eine zyklische Aktualisierung der Ausgänge.

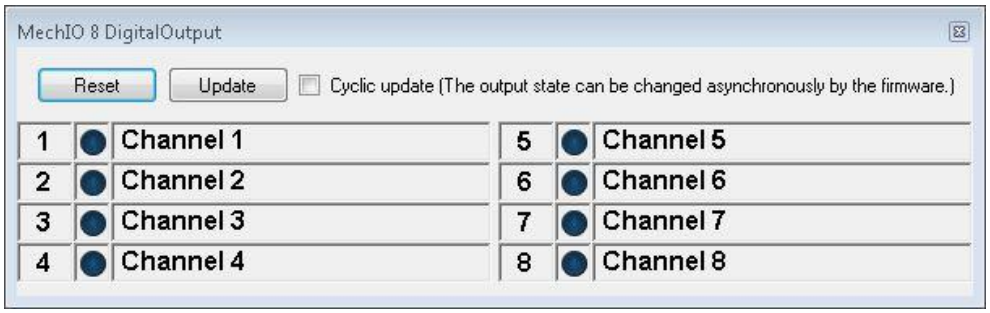


Abbildung 14: Digitale Ausgänge

4.3.1.5. SerIO - Kommunikation

Mit Hilfe des Formulars kann die laufende SerIO - Kommunikation des betreffenden Gerätes mitgehört oder eigene Nachrichten versendet werden.

Es besteht die Möglichkeit die empfangenen Daten aufzunehmen (Logging). Weiterhin können selbst erstellte Nachrichten oder Sequenzen von Nachrichten abgespeichert und geladen werden.

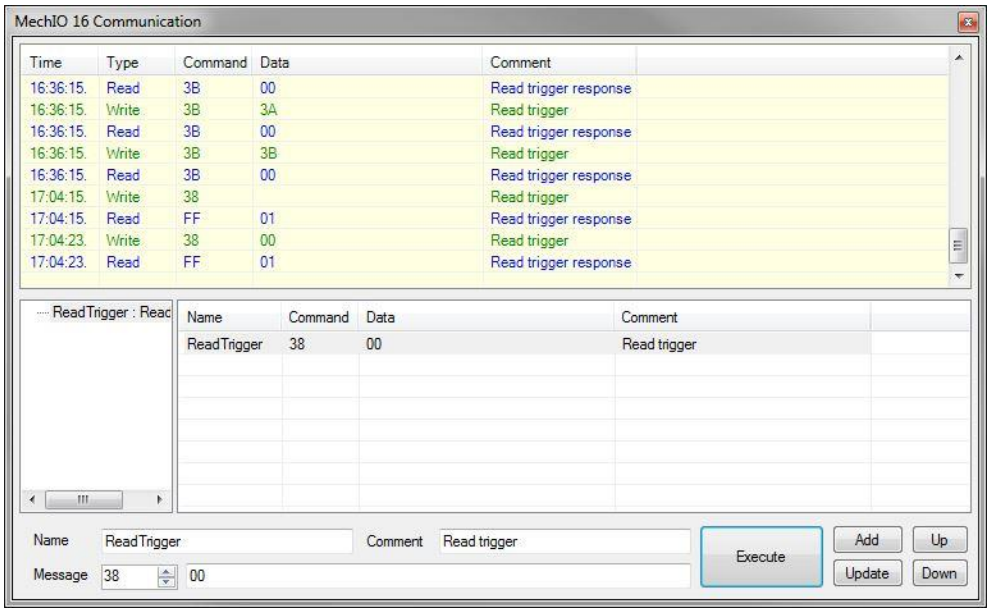


Abbildung 15: SerIO - Kommunikation

Über die Eingabemaske können eigene Nachrichten angelegt werden. Hierzu kann der Nachrichtentyp und der Dateninhalt eingegeben werden.

Im Bereich „Message“ sind das „SerIO - Kommando“ und die zugehörigen Daten einzutragen.

Über die Schaltfläche „Execute“ wird die Nachricht versendet.

In den Feldern „Name“ und „Comment“ können weitere Informationen zu der Nachricht hinterlegt und diese dann über die „Add“ - Schaltfläche gespeichert werden.

4.3.2. MechIO 16

4.3.2.1. I²C - Kommunikation

Mit Hilfe des Formulars kann die gesamte laufende I²C - Kommunikation des entsprechenden SerIO - Gerätes mitgehört oder eigene Nachrichten versendet werden.

Es besteht die Möglichkeit die empfangenen Daten aufzunehmen („Logging“). Weiterhin können selbst erstellte Nachrichten oder Sequenzen von Nachrichten abgespeichert und geladen werden.

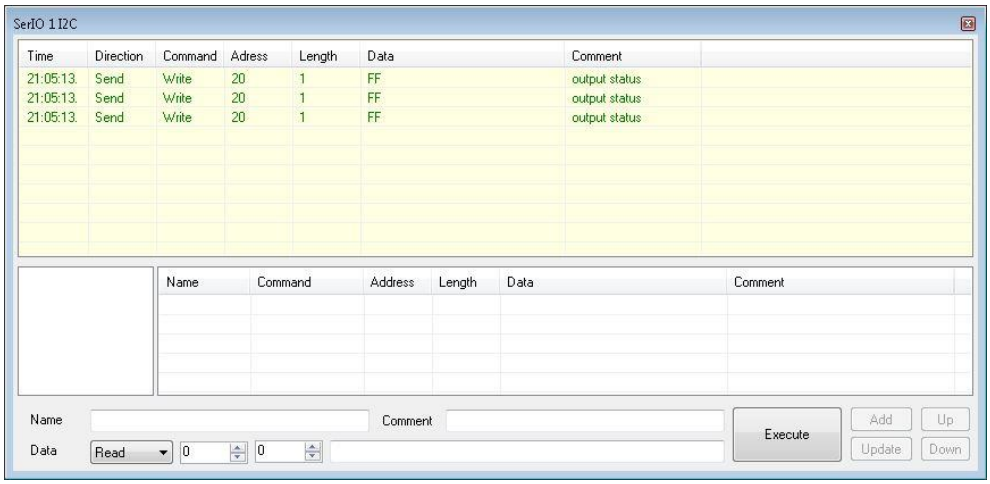


Abbildung 16: I²C - Kommunikation

Über die Eingabemaske können eigene Nachrichten angelegt werden. Hierzu kann der Nachrichtentyp und der Dateninhalt eingegeben werden.

Im Bereich „Message“ sind das „I²C - Kommando“, die „I²C - Adresse“ und die zugehörigen Daten einzutragen.

Über die Schaltfläche „Execute“ wird die Nachricht versendet.

In den Feldern „Name“ und „Comment“ können weitere Informationen zu der Nachricht hinterlegt und diese dann über die „Add“ - Schaltfläche gespeichert werden.

4.3.2.2. MechIO 16 Analog Input

Hier erfolgt die Anzeige und Steuerung der analogen Eingänge der Mechanik - Steuerung. Über den Button „Update“ kann der aktuelle Zustand der Eingänge aktualisiert werden. Die Check - Box „Cyclic update“ aktiviert eine zyklische Aktualisierung der Eingänge.

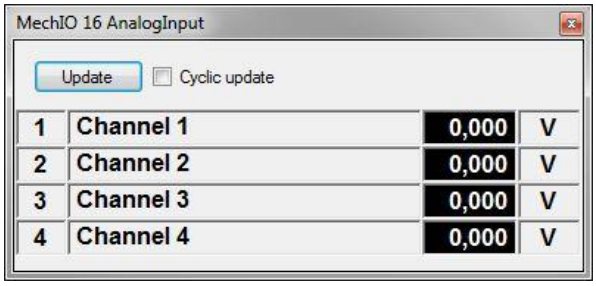


Abbildung 17: Analoge Eingänge

4.3.2.3. MechIO 16 Digital Input

Hier erfolgt die Anzeige und Steuerung der digitalen Eingänge der Mechanik - Steuerung. Über den Button „Update“ kann der aktuelle Zustand der Eingänge aktualisiert werden. Die Check - Box „Cyclic update“ aktiviert eine zyklische Aktualisierung der Eingänge. Mit Hilfe von „Trigger level“ kann die Triggerschwelle (in Volt) für die digitalen Eingänge eingestellt werden.

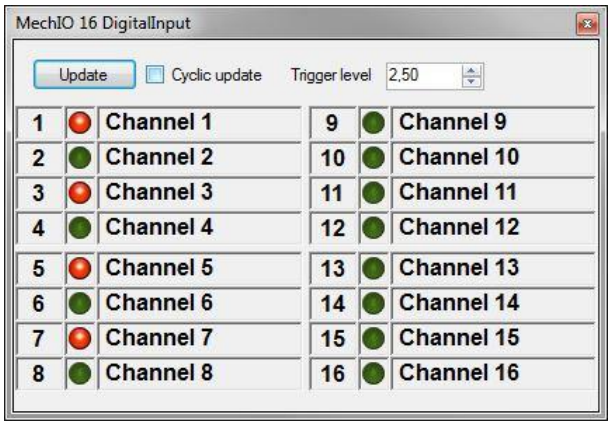


Abbildung 18: Digitale Eingänge

4.3.2.4. MechIO 16 Digital Output

Hier erfolgt die Anzeige und Steuerung der digitalen Ausgänge der Mechanik - Steuerung. Über den Button „Update“ kann der aktuelle Zustand der Ausgänge aktualisiert werden. Die Check - Box „Cyclic update“ aktiviert eine zyklische Aktualisierung der Ausgänge.

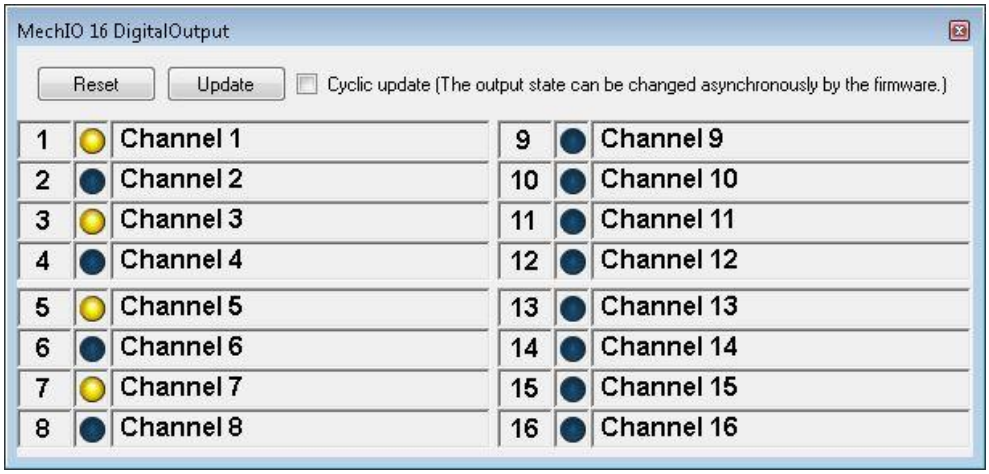


Abbildung 19: Digitale Ausgänge

4.3.2.5. MechIO 16 Trigger

Mit Hilfe des Formulars könne die Trigger der Mechanik - Steuerung eingestellt werden.

Es können maximal acht interne Trigger definiert werden. Die Mechanik - Steuerung kontrolliert zyklisch die Eingänge und setzt nach Übereinstimmung mit der Triggerbedingung die entsprechenden Ausgangszustände bzw. das entsprechende „Merkerflag“. Die Priorität fällt mit steigendem „Index“. Das bedeutet, dass „Index 0“ immer den Ausgabezustand von z.B. „Index 1“ überschreibt. „Trigger Index 0“ wird nach „Power - UP“ aus dem EEPROM geladen und hat höchste Priorität.

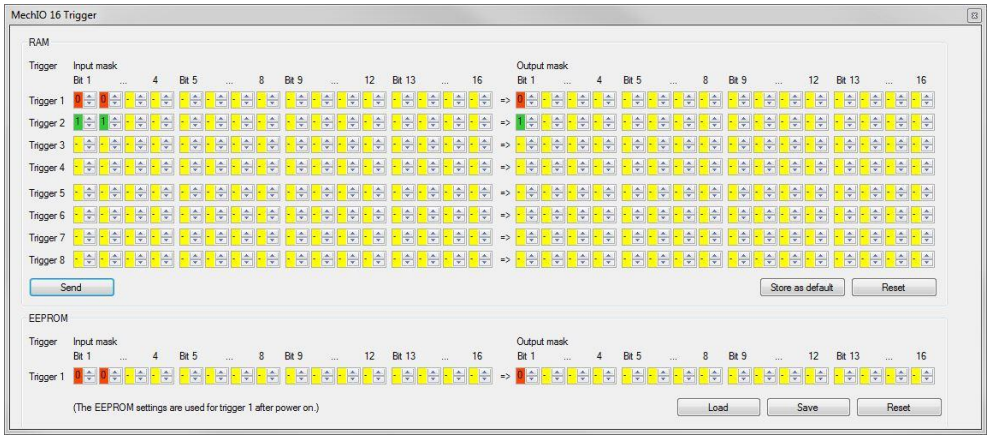


Abbildung 20: Trigger Interface

Das Trigger Interface besteht aus folgenden Bereichen:

- Input mask: Hier wird das Eingangsmuster für den entsprechenden Trigger eingestellt. Dabei bedeutet ein „-“, dass das entsprechende Bit für den Trigger irrelevant ist, eine „0“, dass das entsprechende Bit „Low“ sein muss und eine „1“, dass das entsprechende Bit „High“ sein muss
- Output mask: Hier wird das zu setzende Ausgangsmuster für den entsprechenden Trigger eingestellt. Dabei bedeutet ein „-“, dass das entsprechende Bit für den Trigger irrelevant ist, eine „0“, dass das entsprechende Bit auf „Low“ gesetzt wird und eine „1“, dass das entsprechende Bit „High“ gesetzt wird
- Send: Mit Hilfe dieses Buttons wird die Triggerkonfiguration an die Mechanik - Steuerung übertragen
- Store as default: Mit Hilfe dieses Buttons wird die aktuelle Triggerkonfiguration in das Setup des Toolmonitors übernommen
- Reset: Mit Hilfe dieses Buttons wird die Triggerkonfiguration zurückgesetzt
- EEPROM: Es besteht die Möglichkeit die Konfiguration für den „Trigger 1“ persistent im EEPROM der Mechanik - Steuerung zu hinterlegen. Dies ist über diesen Bereich des Triggerformulars möglich

4.3.2.6. SerIO - Kommunikation

Mit Hilfe des Formulars kann die laufende SerIO - Kommunikation des betreffenden Gerätes mitgehört oder eigene Nachrichten versendet werden.

Es besteht die Möglichkeit die empfangenen Daten aufzunehmen („Logging“). Weiterhin können selbst erstellte Nachrichten oder Sequenzen von Nachrichten abgespeichert und geladen werden.

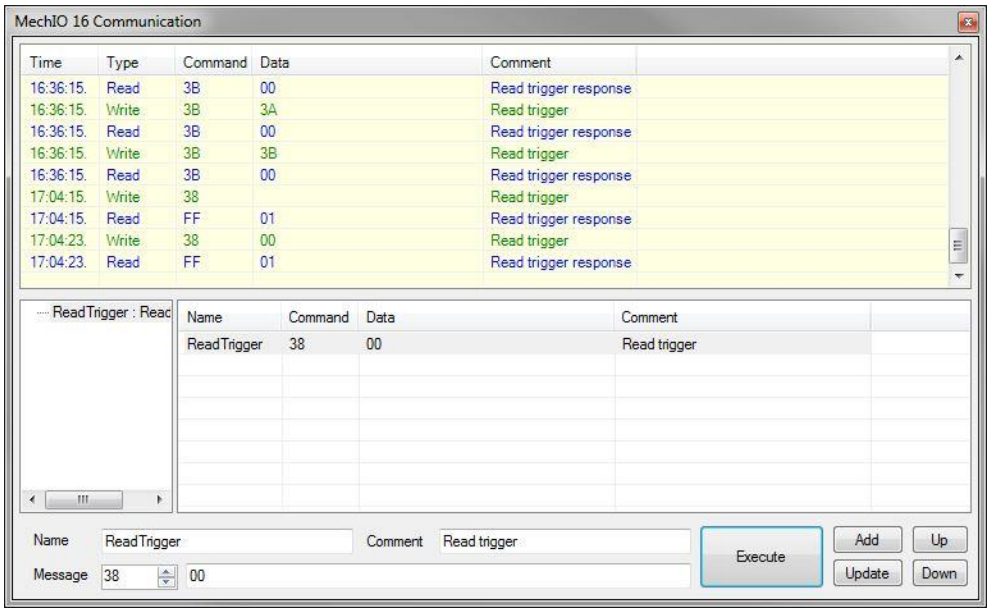


Abbildung 21: SerIO - Kommunikation

Über die Eingabemaske können eigene Nachrichten angelegt werden. Hierzu kann der Nachrichtentyp und der Dateninhalt eingegeben werden.

Im Bereich „Message“ sind das „SerIO - Kommando“ und die zugehörigen Daten einzutragen.

Über die Schaltfläche „Execute“ wird die Nachricht versendet.

In den Feldern „Name“ und „Comment“ können weitere Informationen zu der Nachricht hinterlegt und diese dann über die „Add“ - Schaltfläche gespeichert werden.

4.3.3. Adapter CPU

Dieser SerIO - Gerätetyp stellt die MCD Mechanik - Steuerung „Adapter - CPU“ dar.

4.3.3.1. I^2C - Kommunikation

Mit Hilfe des Formulars kann die gesamte laufende I^2C - Kommunikation des entsprechenden SerIO - Gerätes mitgehört oder eigene Nachrichten versendet werden.

Es besteht die Möglichkeit die empfangenen Daten aufzunehmen („Logging“). Weiterhin können selbst erstellte Nachrichten oder Sequenzen von Nachrichten abgespeichert und geladen werden.

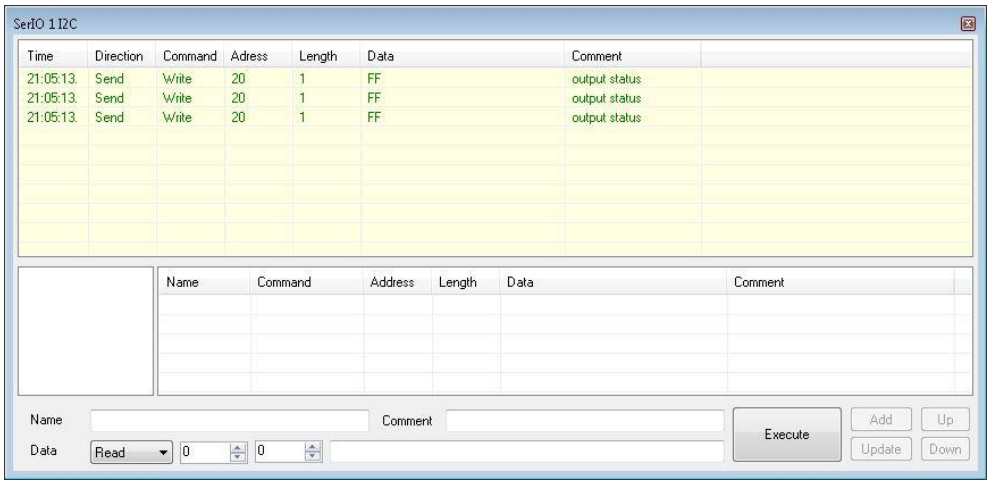


Abbildung 22: I^2C - Kommunikation

Über die Eingabemaske können eigene Nachrichten angelegt werden. Hierzu kann der Nachrichtentyp und der Dateninhalt eingegeben werden.

Im Bereich „Message“ sind das „I^2C - Kommando“, die „I^2C - Adresse“ und die zugehörigen Daten einzutragen.

Über die Schaltfläche „Execute“ wird die Nachricht versendet.

In den Feldern „Name“ und „Comment“ können weitere Informationen zu der Nachricht hinterlegt und diese dann über die „Add“ - Schaltfläche gespeichert werden.

4.3.3.2. SerIO - Kommunikation

Mit Hilfe des Formulars kann die laufende SerIO - Kommunikation des betreffenden Gerätes mitgehört oder eigene Nachrichten versendet werden.

Es besteht die Möglichkeit die empfangenen Daten aufzunehmen („Logging“). Weiterhin können selbst erstellte Nachrichten oder Sequenzen von Nachrichten abgespeichert und geladen werden.

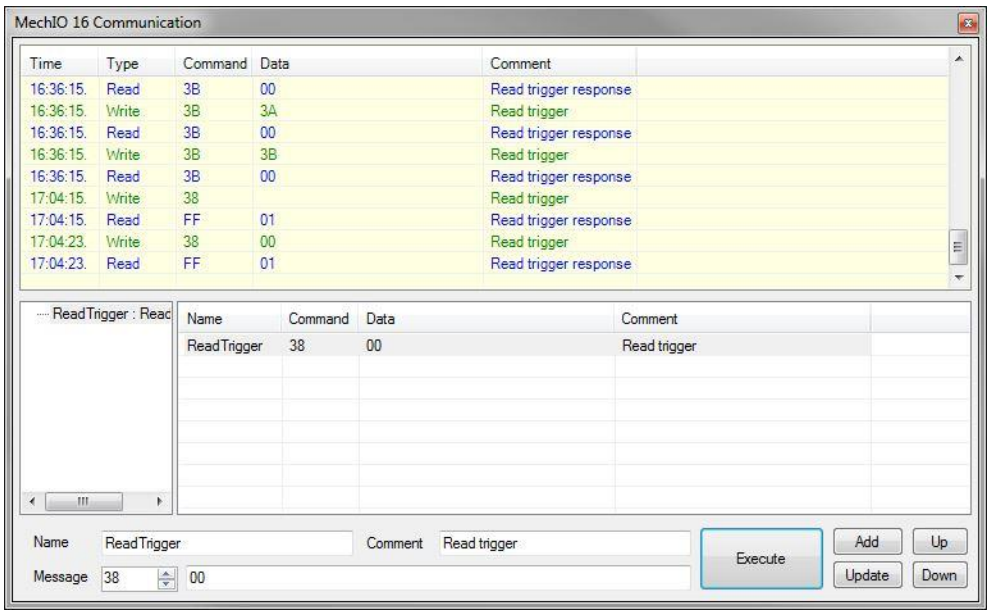


Abbildung 23: SerIO - Kommunikation

Über die Eingabemaske können eigene Nachricht angelegt werden. Hierzu kann der Nachrichtentyp und der Dateninhalt eingegeben werden.

Im Bereich „Message“ sind das SerIO - Kommando und die zugehörigen Daten einzutragen.

Über die Schaltfläche „Execute“ wird die Nachricht versendet.

In den Feldern „Name“ und „Comment“ können weitere Informationen zu der Nachricht hinterlegt und diese dann über die „Add“ - Schaltfläche gespeichert werden.

4.3.4. Pass / Fail - Anzeige

Dieser SerIO - Gerätetyp stellt die MCD Pass / Fail - Anzeige dar.

4.3.4.1. I^2C - Kommunikation

Mit Hilfe des Formulars kann die gesamte laufende I^2C - Kommunikation des entsprechenden SerIO - Gerätes mitgehört oder eigene Nachrichten versendet werden.

Es besteht die Möglichkeit die empfangenen Daten aufzunehmen („Logging“). Weiterhin können selbst erstellte Nachrichten oder Sequenzen von Nachrichten abgespeichert und geladen werden.

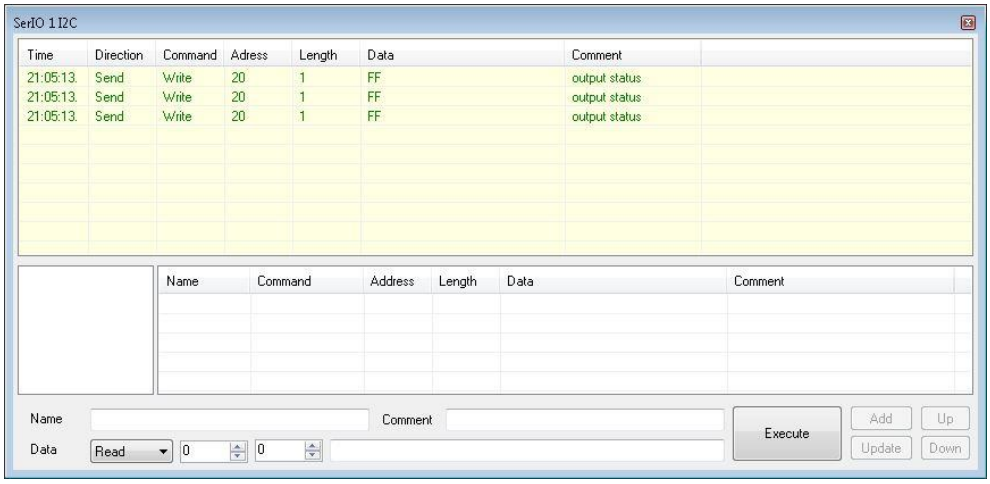


Abbildung 24: I^2C - Kommunikation

Über die Eingabemaske können eigene Nachrichten angelegt werden. Hierzu kann der Nachrichtentyp und der Dateninhalt eingegeben werden.

Im Bereich „Message“ sind das „I^2C - Kommando“, die „I^2C - Adresse“ und die zugehörigen Daten einzutragen.

Über die Schaltfläche „Execute“ wird die Nachricht versendet.

In den Feldern „Name“ und „Comment“ können weitere Informationen zu der Nachricht hinterlegt und diese dann über die „Add“ - Schaltfläche gespeichert werden.

4.3.4.2. Pass / Fail Digital Output

Hier erfolgt die Anzeige und Steuerung der digitalen Ausgänge der Pass / Fail - Anzeige. Über den Button „Reset“ können alle Ausgänge zurückgesetzt werden.

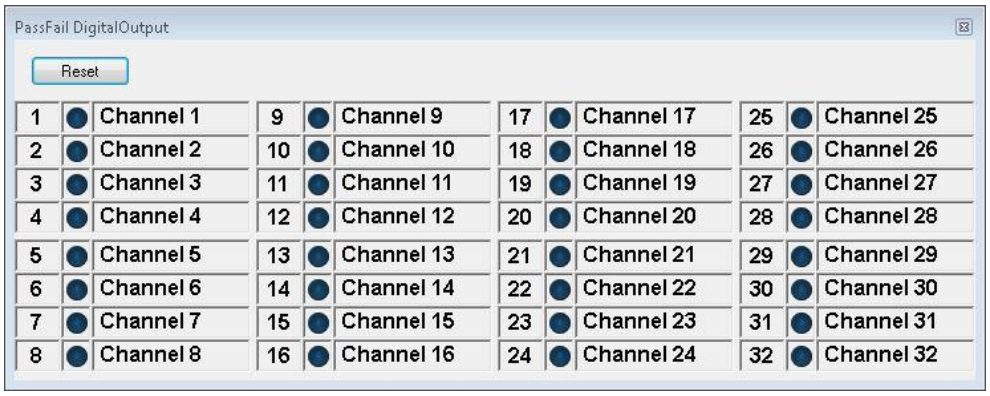


Abbildung 25: Digitale Ausgänge

4.3.4.3. SerIO - Kommunikation

Mit Hilfe des Formulars kann die laufende SerIO - Kommunikation des betreffenden Gerätes mitgehört oder eigene Nachrichten versendet werden.

Es besteht die Möglichkeit die empfangenen Daten aufzunehmen („Logging“). Weiterhin können selbst erstellte Nachrichten oder Sequenzen von Nachrichten abgespeichert und geladen werden.

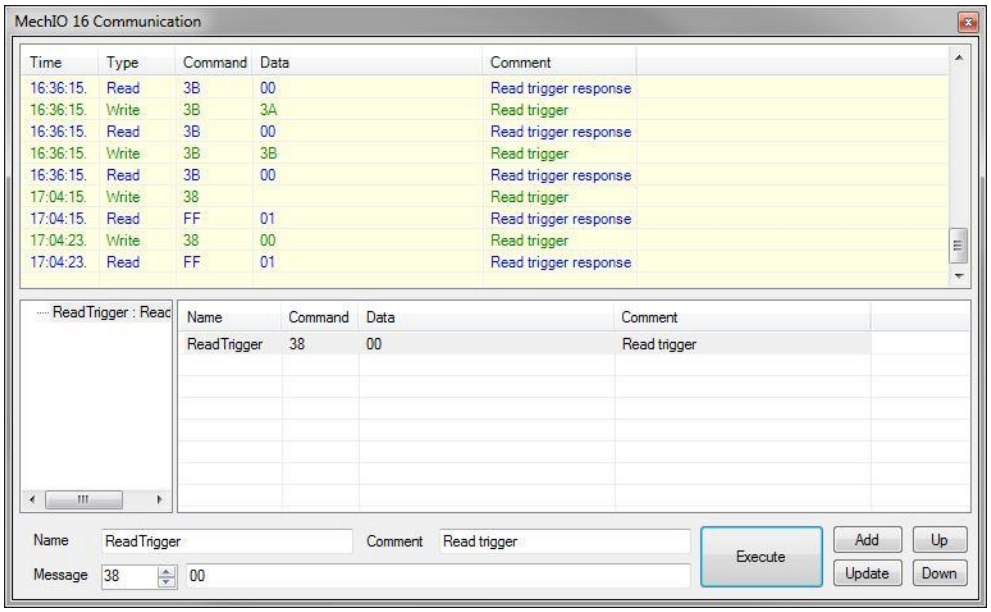


Abbildung 26: SerIO - Kommunikation

Über die Eingabemaske können eigene Nachrichten angelegt werden. Hierzu kann der Nachrichtentyp und der Dateninhalt eingegeben werden.

Im Bereich „Message“ sind das „SerIO - Kommando“ und die zugehörigen Daten einzutragen.

Über die Schaltfläche „Execute“ wird die Nachricht versendet.

In den Feldern „Name“ und „Comment“ können weitere Informationen zu der Nachricht hinterlegt und diese dann über die „Add“ - Schaltfläche gespeichert werden.

4.4. I²C - Geräte

Für jedes I²C - Gerät wird im Windows - Menü des Toolmonitors SerIO ein entsprechender Eintrag angelegt. Über diesen Eintrag, könne verschiedene Formulare zur Steuerung des I²C - Gerätes geöffnet werden.

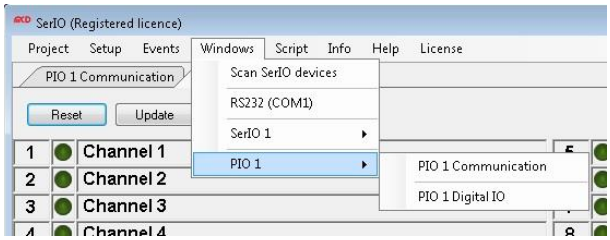


Abbildung 27: Menü für I²C - Geräte

4.4.1. I²C Digitale IO - Geräte (PIO)

Eine Reihe von I²C - Geräten dienen der digitalen Ein - und Ausgabe und basieren auf einer oder mehrerer I²C - PIO und werden auf die gleiche Art und Weise konfiguriert.

Folgende Geräte sind davon betroffen:

- PCF8574 | PIO 8 channels per port
- PCF8574A | PIO 8 channels per port
- PCA9554 | PIO 8 channels per port
- PCA9554A | PIO 8 channels per port
- PIO 3 x PCA 9554 | B.140.010.110
- PIO 3 x PCA 9554A | B.140.010.210
- PIO 3 x PCF 8574 | B.140.010.310
- PIO 3 x PCF 8574A | B.140.010.410
- RelMUX16 | Relais multiplexer 16 channels 7442

4.4.1.1. I²C - Kommunikation

Mit Hilfe des Formulars kann die laufende I²C - Kommunikation des entsprechenden I²C - Moduls mitgehört oder eigene Nachrichten versendet werden.

Es besteht die Möglichkeit die empfangenen Daten aufzunehmen („Logging“). Weiterhin können selbst erstellte Nachrichten oder Sequenzen von Nachrichten abgespeichert und geladen werden.

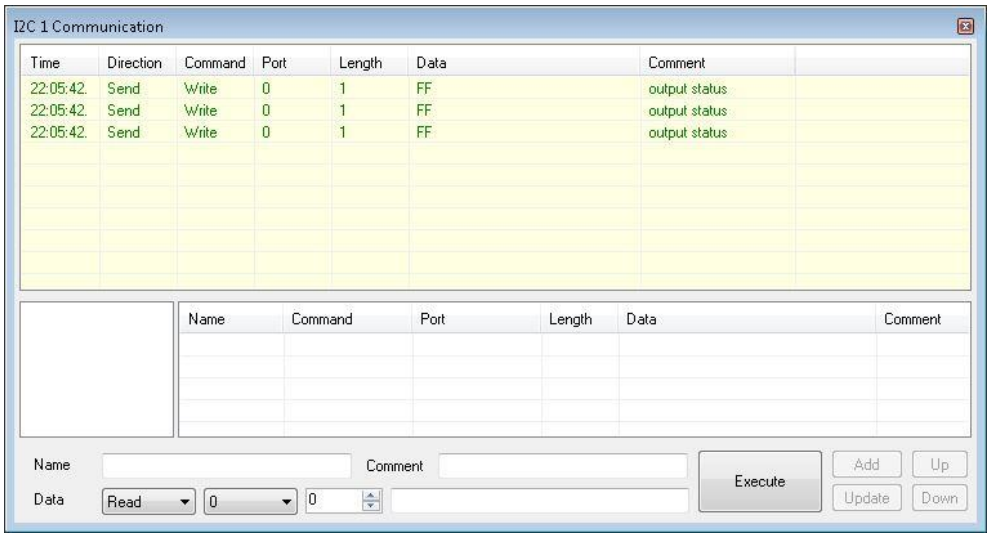


Abbildung 28: I²C - Kommunikation

Über die Eingabemaske können eigene Nachrichten angelegt werden. Hierzu kann der Nachrichtentyp und der Dateninhalt eingegeben werden.

Im Bereich „Message“ sind das „I²C - Kommando“, die „I²C - Bank“ und die zugehörigen Daten einzutragen.

Über die Schaltfläche „Execute“ wird die Nachricht versendet.

In den Feldern „Name“ und „Comment“ können weitere Informationen zu der Nachricht hinterlegt und diese dann über die „Add“ - Schaltfläche gespeichert werden.

4.4.1.2. PIO Digital IO

Hier erfolgt die Anzeige und Steuerung der digitalen Ein - und Ausgänge der „I²C - PIO - Module“. Über den Button „Reset“ können alle Ausgänge zurückgesetzt werden.

Über den Button „Update“ kann der aktuelle Zustand der Ein - und Ausgänge aktualisiert werden. Die Check - Box „Cyclic update“ aktiviert eine zyklische Aktualisierung der Ein - und Ausgänge.

Je nach Konfiguration des PIO - Moduls kann die Betriebsart (Ein - oder Ausgang) online umgeschaltet werden. Diese Option ist nicht für alle PIO - Module verfügbar und von der Konfiguration abhängig.

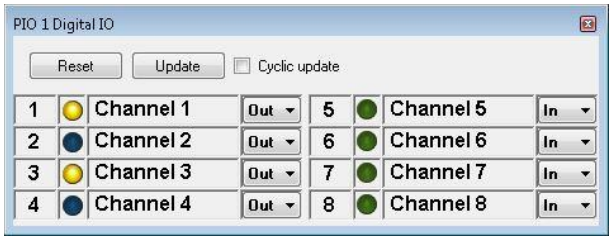


Abbildung 29: Digitale Ein - und Ausgänge

4.4.2. Module zur Stromversorgung

Eine Reihe von I²C - Geräten dienen der Stromversorgung über spezielle Stromversorgungsmodule mit einstellbarer Frequenz des Reglerbausteins.

Folgende Geräte sind davon betroffen:

- PST30M | Power supply for tuner applications master B.170.030.112
- PST30S | Power supply for tuner applications slave B.170.030.212
- PST30L | Power supply for tuner applications linear slave B.170.030.412

4.4.2.1. I²C - Kommunikation

Mit Hilfe des Formulars kann die laufende I²C - Kommunikation des entsprechenden I²C - Moduls mitgehört oder eigene Nachrichten versendet werden.

Es besteht die Möglichkeit die empfangenen Daten aufzunehmen („Logging“). Weiterhin können selbst erstellte Nachrichten oder Sequenzen von Nachrichten abgespeichert und geladen werden.

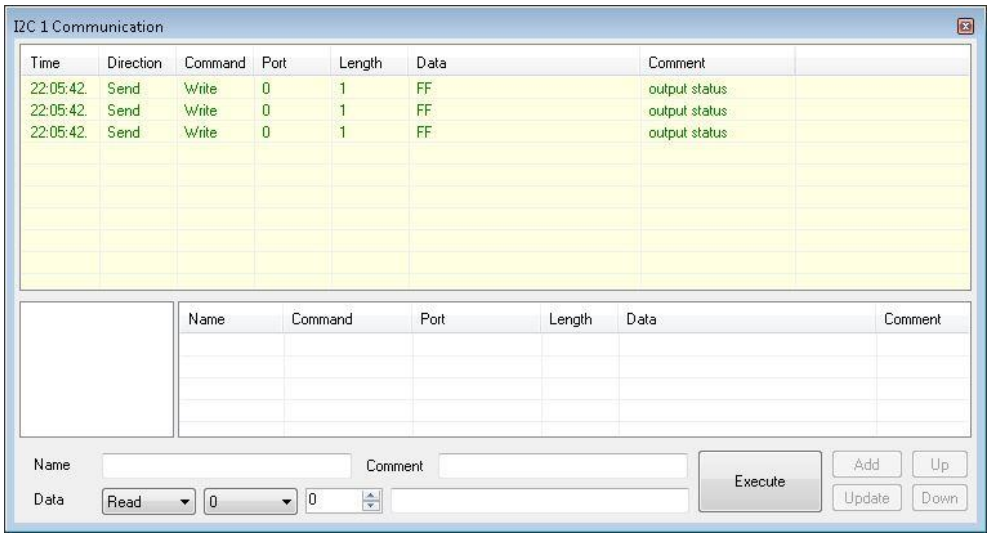


Abbildung 30: I²C - Kommunikation

Über die Eingabemaske können eigene Nachricht angelegt werden. Hierzu kann der Nachrichtentyp und der Dateninhalt eingegeben werden.

Im Bereich „Message“ sind das „I²C - Kommando“, die „I²C - Bank“ und die zugehörigen Daten einzutragen.

Über die Schaltfläche „Execute“ wird die Nachricht versendet.

In den Feldern „Name“ und „Comment“ können weitere Informationen zu der Nachricht hinterlegt und diese dann über die „Add“ - Schaltfläche gespeichert werden.

4.4.2.2. Stromversorgung Analoganzeige Spannung

Mit Hilfe dieses Formulars kann die gemessene Spannung angezeigt werden. Über ein Kontextmenü kann der Anzeigebereich eingestellt werden.

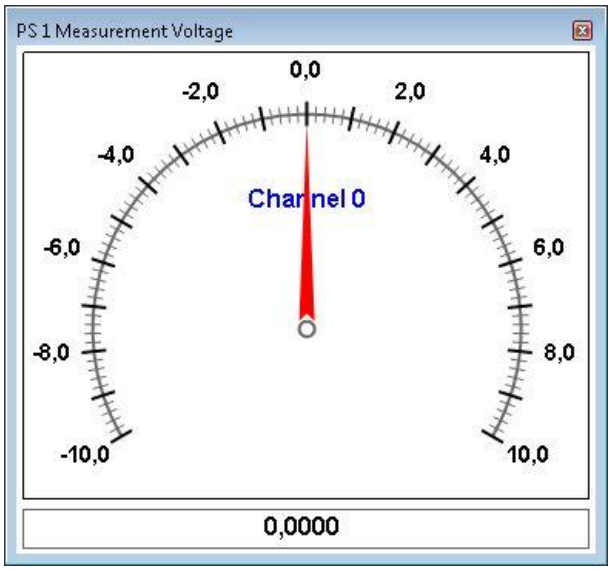


Abbildung 31: Stromversorgung Analoganzeige Spannung

4.4.2.3. Stromversorgung Analoganzeige Strom

Mit Hilfe dieses Formulars kann der gemessene Strom angezeigt werden. Über ein Kontextmenü kann der Anzeigebereich eingestellt werden.

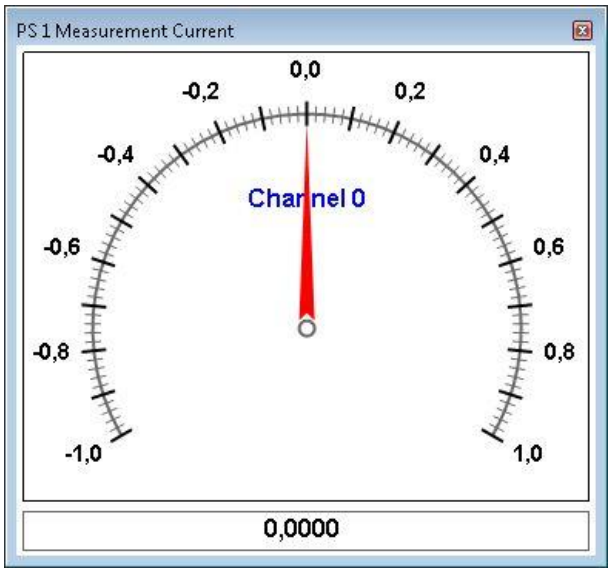


Abbildung 32: Stromversorgung Analoganzeige Strom

4.4.2.4. Stromversorgung Einstellungen

Mit Hilfe dieses Formulars kann das zugehörige Stromversorgungsmodul gesteuert werden. Es kann die Spannung und die „Sense“ ein - und ausgeschaltet werden.

Bei einem „Maste“ - Modul kann außerdem die Frequenz des Reglerbausteins eingestellt werden. Diese Option ist nicht bei allen Modulen verfügbar.

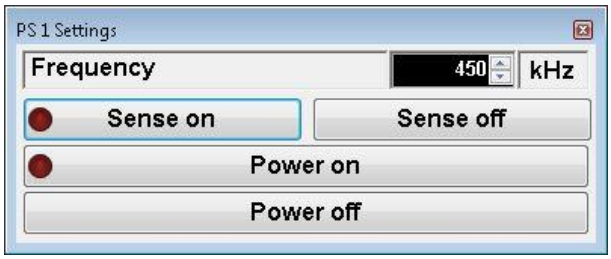


Abbildung 33: Stromversorgung Einstellung

4.4.2.5. Stromversorgung Messung

Mit Hilfe dieses Formulars kann die Spannung und der Strom des zugehörigen Stromversorgungsmoduls ausgelesen und angezeigt werden.

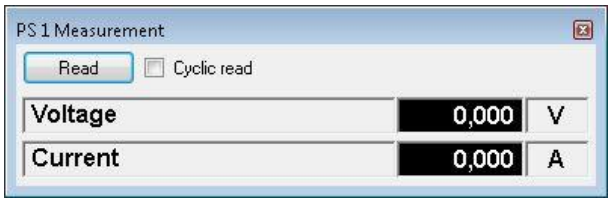


Abbildung 34: Stromversorgung Messung

5. Konfigurationsbeispiel (Schnellstartanleitung)

Dieser Abschnitt beschreibt schrittweise und beispielhaft die Konfiguration des Toolmonitors SerIO. In diesem Beispiel wird davon ausgegangen, dass eine MCD Mechanik - Steuerung MechIO 16 an „COM1“ des PCs angeschlossen ist. Weiterhin befindet sich am I^2C - Anschluss der Mechanik - Steuerung eine Relaisplatine (RelMUX16 | Relais multiplexer 16 channels 7442).

5.1. Sample - Preset aktivieren

Um die Konfiguration zurückzusetzen, im „Projekt“ - Menü unter „Presets“ den „Default“ - Preset laden.

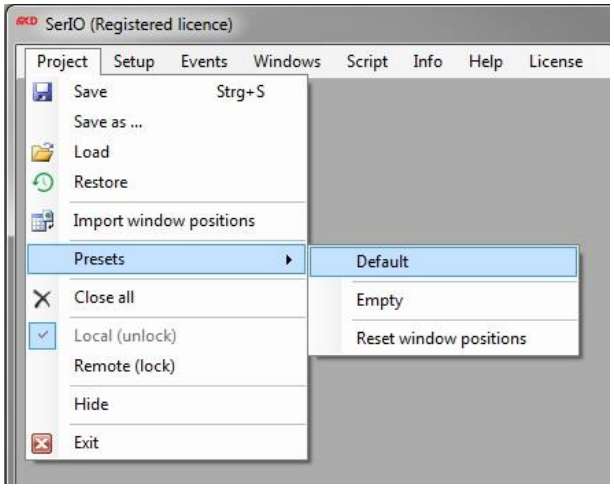


Abbildung 35: Sample - Preset aktivieren

5.2. Angeschlossene Geräte suchen

Um nach angeschlossenen Geräten zu suchen, im „Windows“ - Menü den Befehl „Scan SerIO devices“ aufrufen.

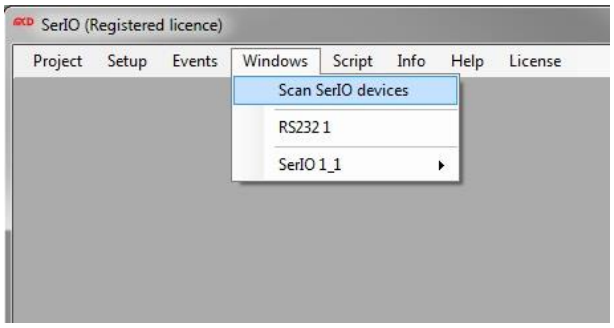


Abbildung 36: Angeschlossene Geräte suchen

Es wird dann nach angeschlossenen Geräten gesucht und diese werden angezeigt:

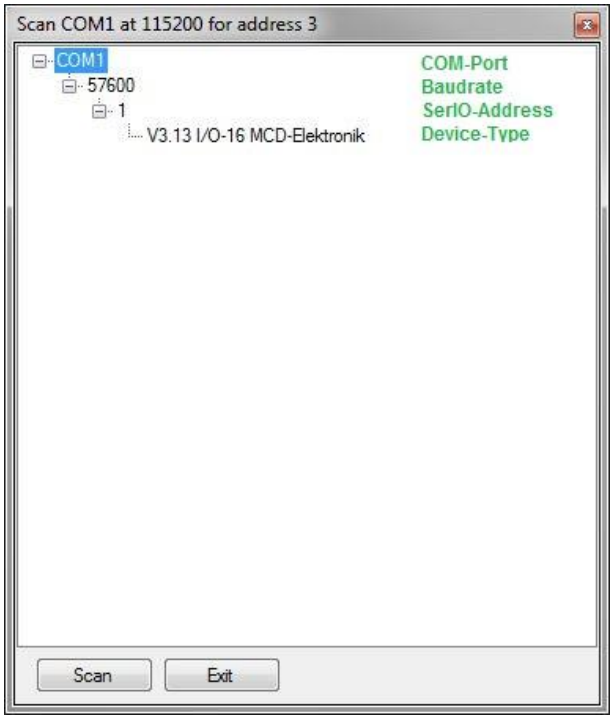


Abbildung 37: Darstellung angeschlossener Geräte

5.3. RS232 - Port konfigurieren

Über das Menü „Setup“ die „Options“ aufrufen und den RS232 - Port einstellen. Dann den „Options“ - Dialog mit „OK“ beenden.

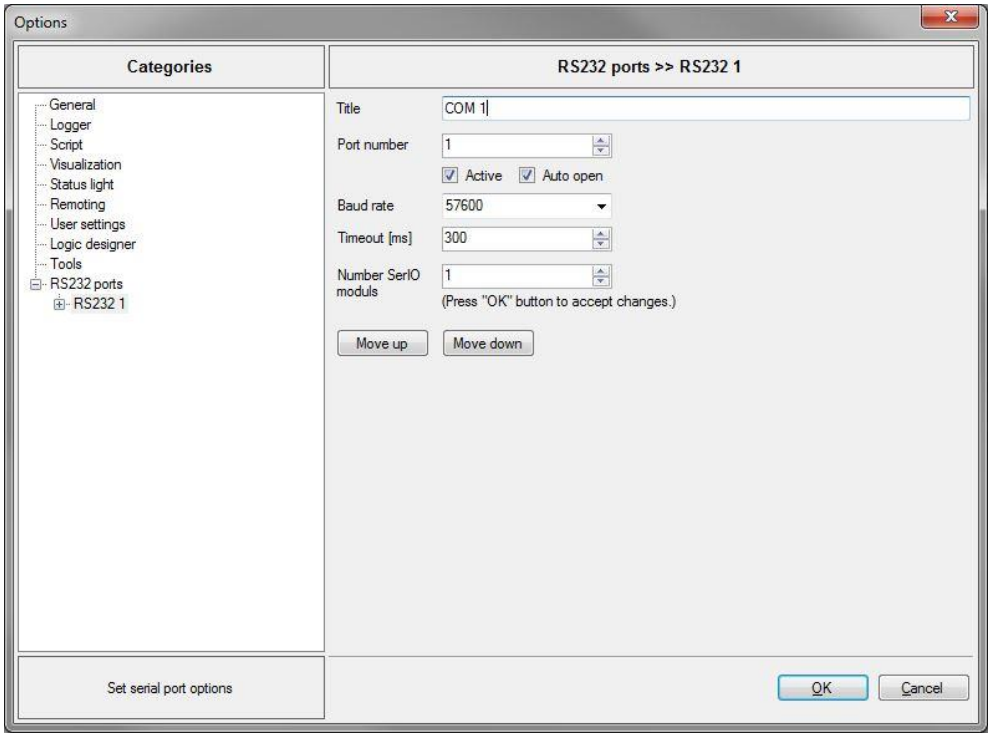


Abbildung 38: RS232 - Port konfigurieren

5.4. SerIO - Geräte konfigurieren

Den „Options“ - Dialog wieder öffnen und das SerIO - Gerät einstellen. Dann den „Options“ - Dialog mit „OK“ beenden.

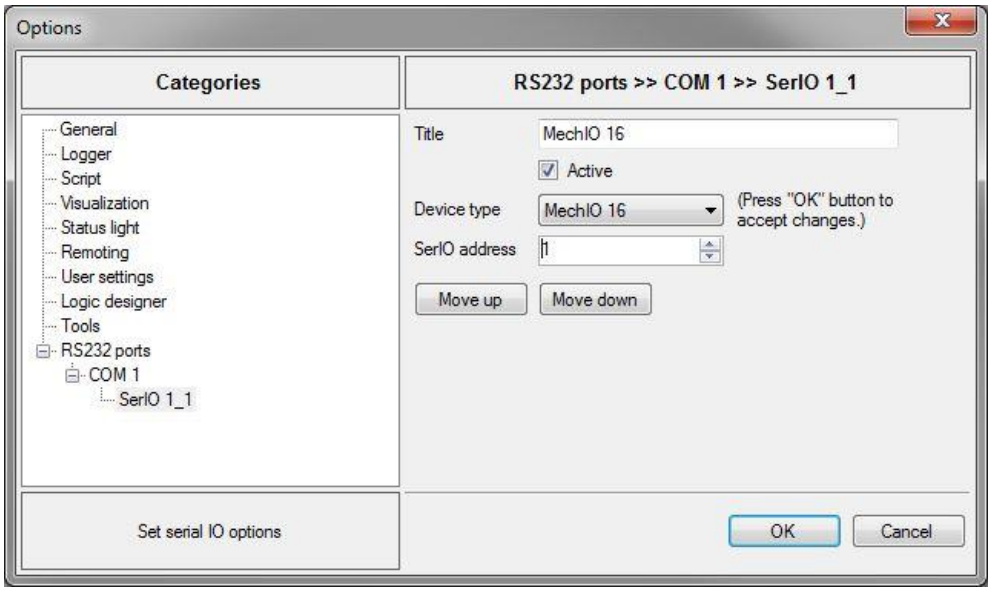


Abbildung 39: SerIO - Geräte konfigurieren

5.5. I²C - Gerät (RelMUX16) konfigurieren

Den „Options“ - Dialog wieder öffnen und das I²C - Gerät (RelMUX16) einstellen. Dann den „Options“ - Dialog mit „OK“ beenden.

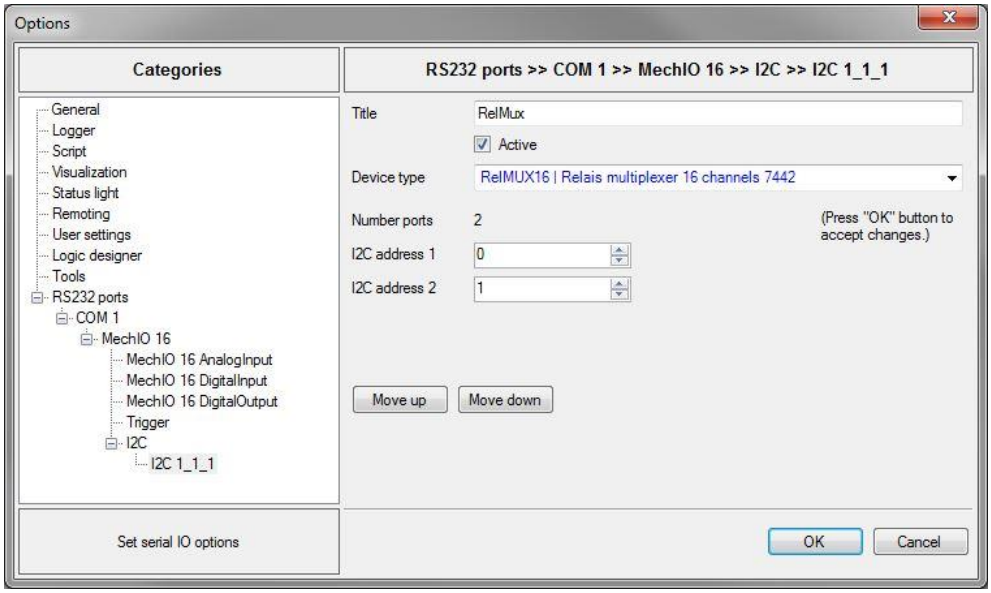


Abbildung 40: I²C - Gerät (RelMUX16) konfigurieren

5.6. Anzeige der digitalen und analogen Ein - und Ausgänge

Über das „Windows“ - Menü können nun die einzelnen Formulare für die digitalen und analogen Ein - und Ausgänge ausgewählt und angezeigt werden. Die Position der Formulare kann frei gewählt werden.

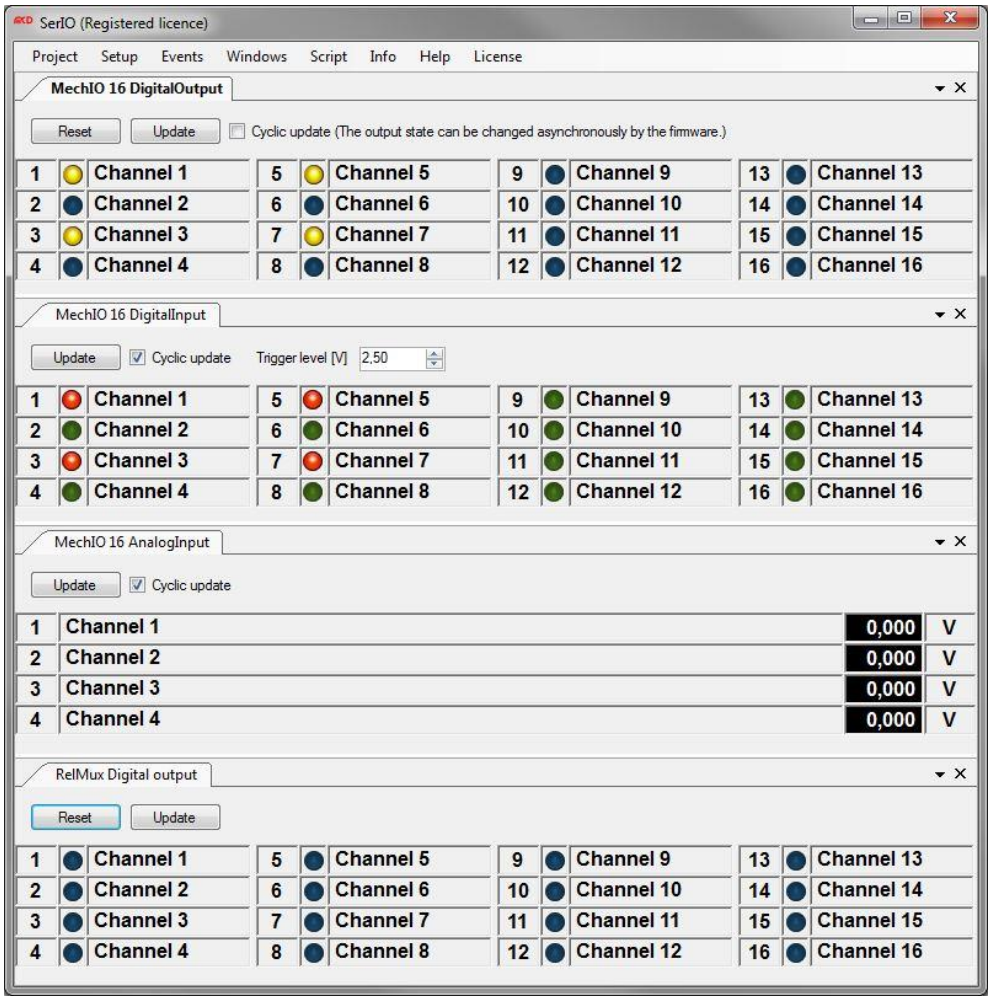


Abbildung 41: Anzeige der digitalen und analogen Ein - und Ausgänge

6. Projektverwaltung

In der Projektverwaltung können die aktuellen Einstellungen und das Layout des Toolmonitors gespeichert und geladen werden. Sämtliche Fenster können frei positioniert und entsprechend den eigenen Erfordernissen angeordnet werden.

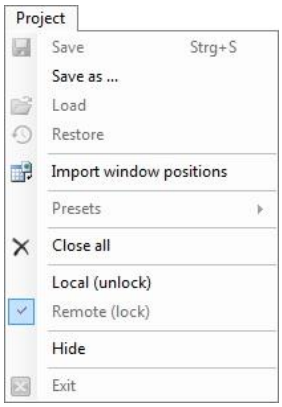


Abbildung 42: Project - Menü

Das Project - Menü besteht aus folgenden Bereichen:

- Save / Save as: Über den Befehl „Save“ können alle aktuellen Einstellungen in einer Projektdatei gespeichert werden. Auch die aktuellen Fensterpositionen werden darin festgehalten
- Load: Über den Befehl „Load“ können zuvor gespeicherte Einstellungen wieder geladen werden. Auch die ursprünglichen Fensterpositionen werden wieder hergestellt
- Import window positions: Über den Befehl „Import window positions“ können aus einem gespeicherten Setup die Fensterpositionen importiert werden. Alle ursprünglichen Einstellungen bleiben davon unberührt
- Presets: Hier können beispielhaft vordefinierte Einstellungen aufgerufen werden
- Close all: Dadurch werden alle offenen Formulare geschlossen. Der Toolmonitor bleibt aktiv
- Local: Wird der Toolmonitor ferngesteuert oder ist das „Setup“ mit einem Passwort geschützt, sind die meisten Benutzeraktion vor direkter Eingabe geschützt. Durch den Aufruf von „Local“ wird dieser Schutz aufgehoben und alle Bedienelemente sind wieder zugänglich. Wurde im „Setup“ ein Passwort vergeben, muss dieses zur Freigabe des Toolmonitors eingegeben werden
- Remote: Hiermit kann der Toolmonitor wieder in den geschützten Modus zurückversetzt werden
- Hide: Hiermit wird der Toolmonitor ausgeblendet, bleibt aber aktiv. Wird er nicht ferngesteuert kann der Toolmonitor über ein Icon in der Taskleiste wieder aktiviert werden
- Exit: Hiermit wird der Toolmonitor beendet (nicht verfügbar, wenn der Toolmonitor ferngesteuert wird)

7. Events

Hierüber können die Anzeigen für die Log - und die Trace - Meldungen aufgerufen werden.

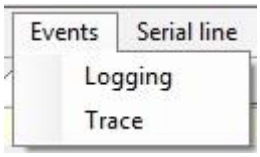


Abbildung 43: Events - Menü

Das Events - Menü besteht aus folgenden Bereichen:

- Logging: Über dieses Menü wird die Anzeige der Log - Meldungen für allgemeine Ereignisse, Warnungen, Fehler usw. aufgerufen
- Trace: Über dieses Menü wird die Anzeige der Trace - Meldungen (gesendeten oder empfangen Nachrichten) aufgerufen

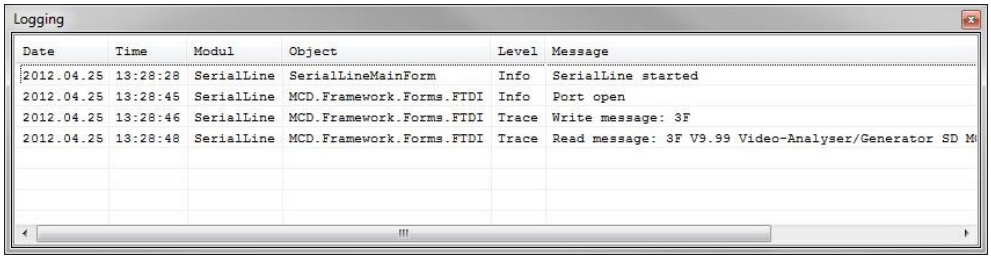


Abbildung 44: Log - Monitor

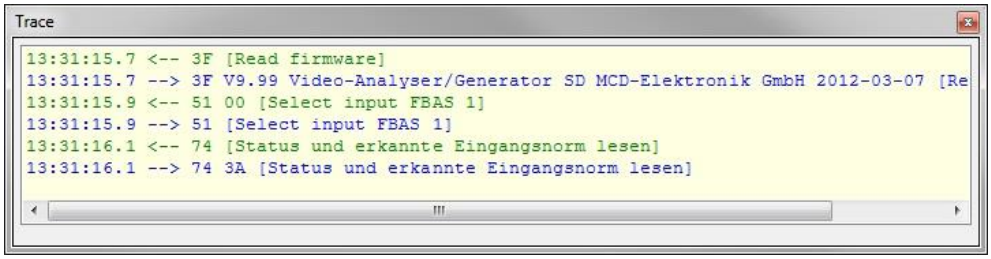


Abbildung 45: Trace - Monitor

8. Setup

Über das Setup erfolgen der Aufruf der projektspezifischen Optionen und die Registrierung des Toolmonitors als COM - Server.

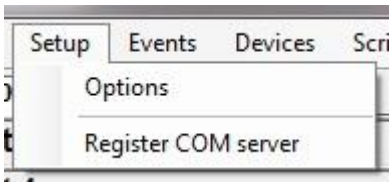


Abbildung 46: Setup - Menü

8.1. RS232 - Ports

Hier erfolgt die Konfiguration der seriellen Schnittstellen.

8.1.1. Anzahl der RS232 - Ports

Hier erfolgt die Festlegung, wie viele RS232 - Schnittstellen vom Toolmonitor SerIO verwendet werden sollen.

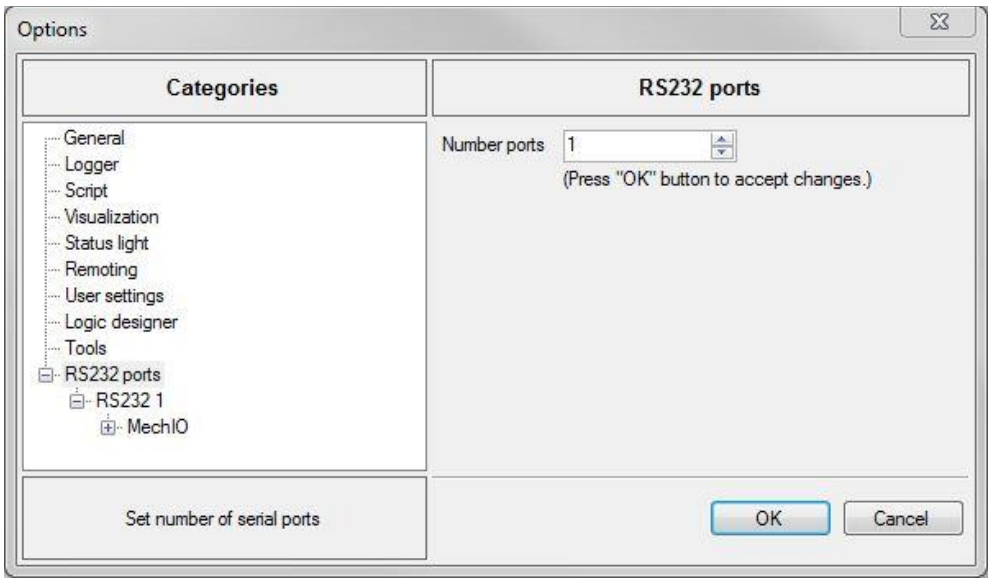


Abbildung 47: Anzahl der RS232 - Ports auswählen

8.1.2. Kommunikationsparameter

Für jeden zu verwendenden RS232-Port können hier die Kommunikationsparameter konfiguriert werden.

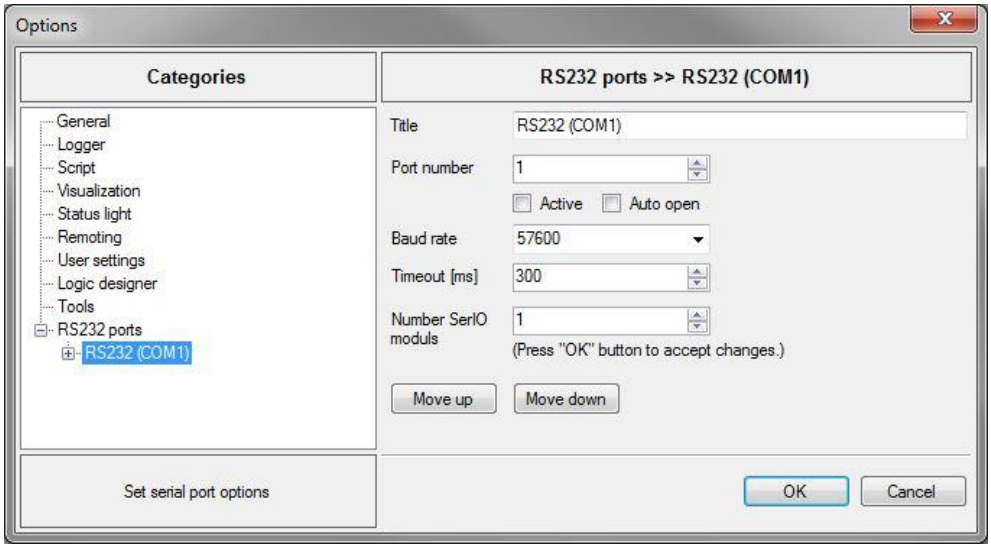


Abbildung 48: Konfiguration eines seriellen Ports

Folgende Bereiche können konfiguriert werden:

- Title: Ein Name, welcher für den Port zur Anzeige und Ansteuerung verwendet werden soll
- Port number: Der zu verwendende COM - Port
- Active: Ein „Flag“, welches kennzeichnet, dass der Port verwendet werden kann
- Auto open: Ein „Flag“, welches kennzeichnet, dass der Port automatisch beim Start des Toolmonitors geöffnet werden soll (Hinweis: auch wenn dieses „Flag“ abgeschaltet ist, so kann der Port trotzdem automatisch geöffnet werden, wenn ein unterlagertes Modul dies erfordert)
- Timeout: Eine Zeitspanne in Millisekunden, welche maximal beim Senden und Empfangen gewartet wird, bis der Vorgang abgeschlossen sein muss
- Number SerIO moduls: Die Anzahl an SerIO - Modulen, welche an diesen RS232 - Port angeschlossen sind
- Move up / Move down: Mit Hilfe dieser Buttons kann der ausgewählte RS232 - Port in der Liste der RS232 - Ports verschoben werden

8.2. SerIO - Geräte

Für jedes zu verwendende SerIO - Gerät kann hier der Typ und die SerIO - Adresse ausgewählt werden. Folgende Typen stehen zur Verfügung:

- Unknown (stellt Grundfunktionen für unspezifische Anforderungen zur Verfügung)
- MechIO 8
- MechIO 16
- Adapter CPU
- Pass / Fail - Anzeige

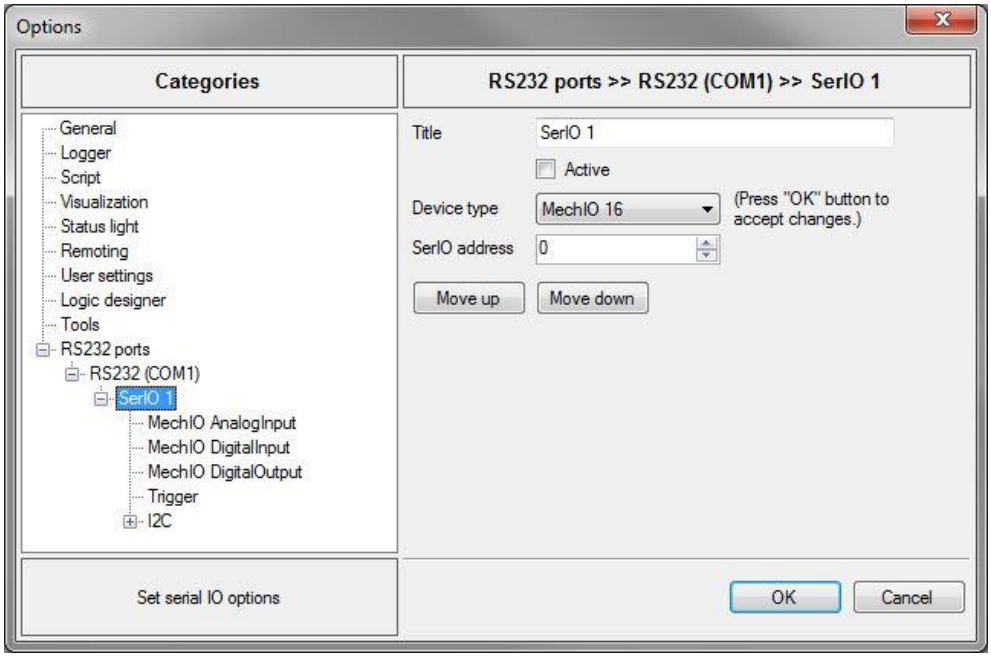


Abbildung 49: Konfiguration eines SerIO - Gerätes

Folgende Bereiche können konfiguriert werden:

- Title: Ein Name, welcher für das SerIO - Gerät zur Anzeige und Ansteuerung verwendet werden soll
- Active: Ein „Flag“, welches kennzeichnet, dass das Gerät verwendet werden kann
- Device type: Der SerIO - Gerätetyp (siehe oben)
- SerIO address: Die SerIO - Adresse für das SerIO - Gerät (jedes an einem gemeinsamen RS232 - Port angeschlossene SerIO - Gerät muss mit einer eindeutigen SerIO - Adresse konfiguriert werden)
- Move up / Move down: Mit Hilfe dieser Buttons kann das ausgewählte SerIO - Gerät in der Liste der SerIO - Geräte verschoben werden

8.2.1. MechIO 8

Dieser SerIO - Gerätetyp stellt die MCD Mechanik - Steuerung MechIO 8 dar.

8.2.1.1. I²C

Hier erfolgt die Festlegung, wie viele I²C - Ports von der MechIO 8 angesteuert werden sollen.

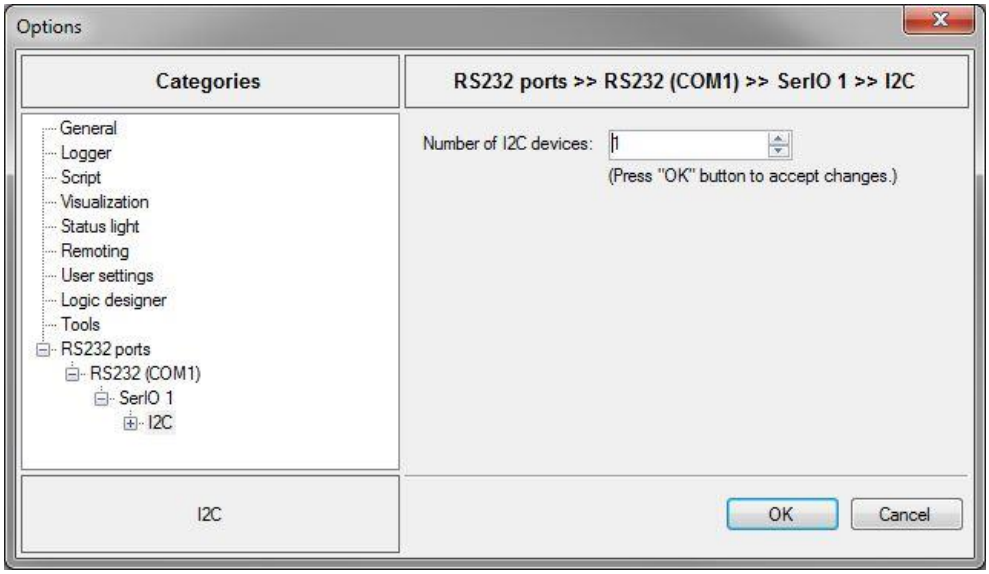


Abbildung 50: Auswahl der Anzahl an I²C - Ports

8.2.1.2. MechIO 8 Analog Input

Hier erfolgt die Konfiguration der analogen Eingänge der Mechanik - Steuerung.

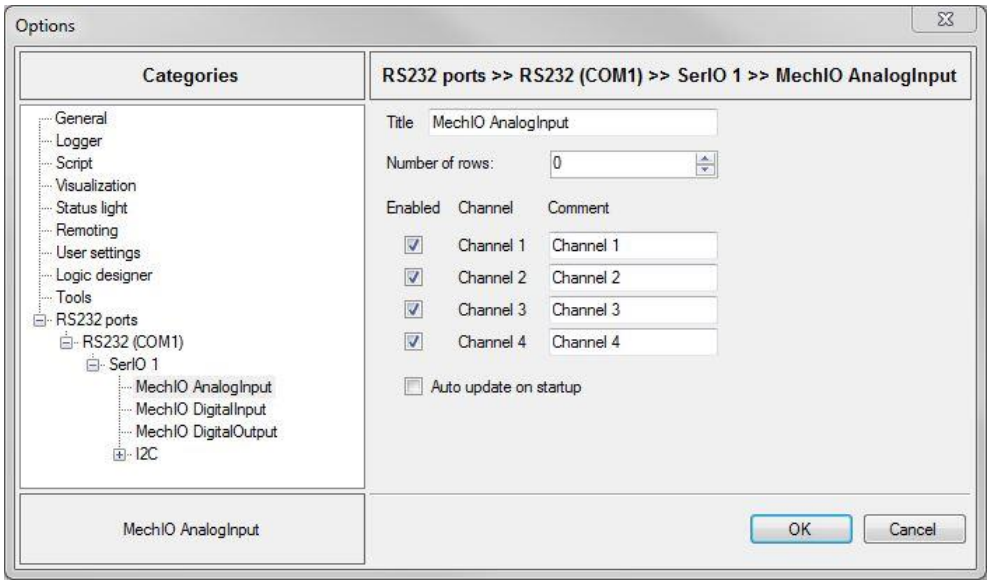


Abbildung 51: Konfiguration der analogen Eingänge

Folgende Bereiche können konfiguriert werden:

- Title: Ein Name, welcher für den analogen Eingang zur Anzeige und Ansteuerung verwendet werden soll
- Number of rows: Die Anzahl an Reihen, welche in der Anzeige für die Eingangskanäle verwendet werden soll. Ist hier der Wert „0“ eingetragen, so wird die Anzahl der Reihen und Spalten automatisch ermittelt, um die Darstellung möglichst platzsparend zu halten
- Enabled: Für jeden Kanal kann ausgewählt werden, ob dieser angezeigt werden soll (nicht benutzte Kanäle können somit ausgeblendet werden)
- Comment: Für jeden Kanal kann ein Name vergeben werden, welcher zur Anzeige und Ansteuerung verwendet wird
- Auto update on startup: Ist dieses „Flag“ aktiv, so wird der zugehörige Kommunikationskanal beim Start des Toolmonitors automatisch geöffnet und der Status der Eingangskanäle wird einmalig ausgelesen und angezeigt

8.2.1.3. MechIO 8 Digital Input

Hier erfolgt die Konfiguration der digitalen Eingänge der Mechanik - Steuerung.

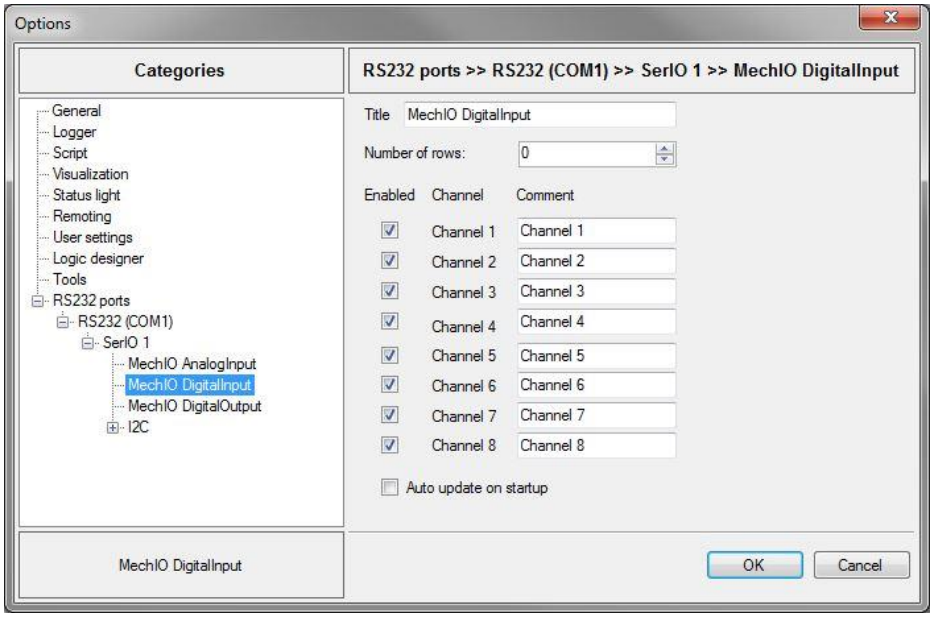


Abbildung 52: Konfiguration der digitalen Eingänge

Folgende Bereiche können konfiguriert werden:

- Title: Ein Name, welcher für die digitalen Eingänge zur Anzeige und Ansteuerung verwendet werden soll
- Number of rows: Die Anzahl an Reihen, welche in der Anzeige für die Eingangskanäle verwendet werden soll. Ist hier der Wert „0“ eingetragen, so wird die Anzahl der Reihen und Spalten automatisch ermittelt, um die Darstellung möglichst platzsparend zu halten
- Enabled: Für jeden Kanal kann ausgewählt werden, ob dieser angezeigt werden soll (nicht benutzte Kanäle können somit ausgeblendet werden)
- Comment: Für jeden Kanal kann ein Name vergeben werden, welcher zur Anzeige und Ansteuerung verwendet wird
- Auto update on startup: Ist dieses „Flag“ aktiv, so wird der zugehörige Kommunikationskanal beim Start des Toolmonitors automatisch geöffnet und der Status der Eingangskanäle wird einmalig ausgelesen und angezeigt

8.2.1.4. MechIO 8 Digital Ouput

Hier erfolgt die Konfiguration der digitalen Ausgänge der Mechanik - Steuerung.

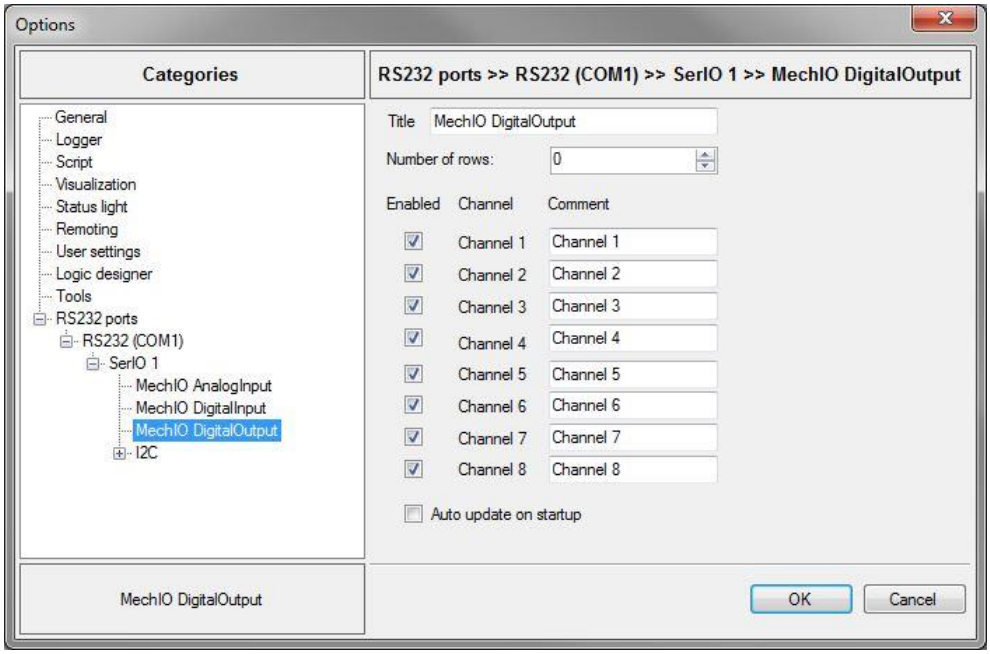


Abbildung 53: Konfiguration der digitalen Ausgänge

Folgende Bereiche können konfiguriert werden:

- Title: Ein Name, welcher für die digitalen Ausgänge zur Anzeige und Ansteuerung verwendet werden soll
- Number of rows: Die Anzahl an Reihen, welche in der Anzeige für die Ausgangskanäle verwendet werden soll. Ist hier der Wert „0“ eingetragen, so wird die Anzahl der Reihen und Spalten automatisch ermittelt, um die Darstellung möglichst platzsparend zu halten
- Enabled: Für jeden Kanal kann ausgewählt werden, ob dieser angezeigt werden soll (nicht benutzte Kanäle können somit ausgeblendet werden)
- Comment: Für jeden Kanal kann ein Name vergeben werden, welcher zur Anzeige und Ansteuerung verwendet wird
- Auto update on startup: Ist dieses „Flag“ aktiv, so wird der zugehörige Kommunikationskanal beim Start des Toolmonitors automatisch geöffnet und der Status der Ausgangskanäle wird einmalig ausgelesen und angezeigt

8.2.2. MechIO 16

8.2.2.1. I²C

Hier erfolgt die Festlegung, wie viele I²C - Ports von der MechIO 16 angesteuert werden sollen.

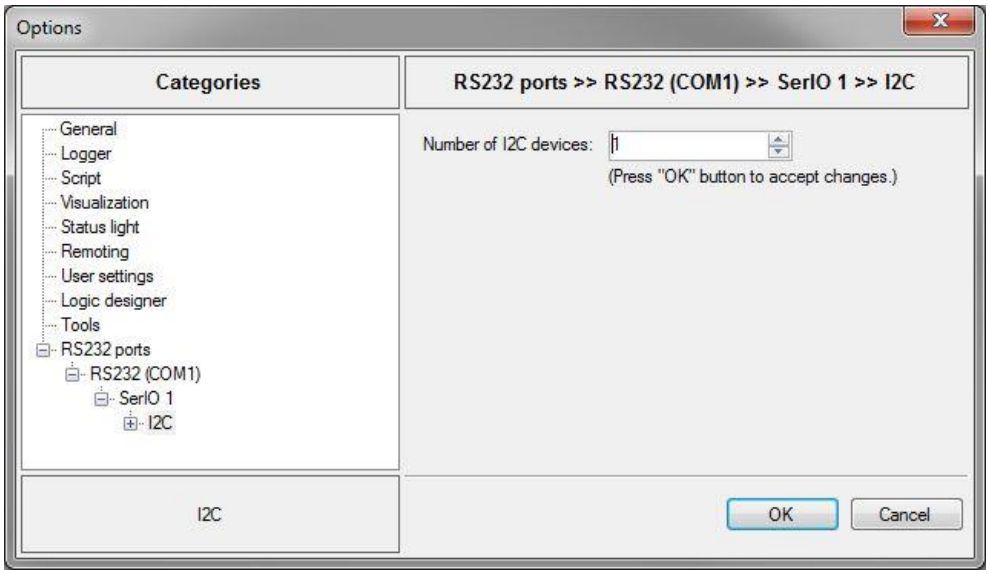


Abbildung 54: Auswahl der Anzahl der I²C - Ports

8.2.2.2. Mech IO 16 Analog Input

Hier erfolgt die Konfiguration der analogen Eingänge der Mechanik - Steuerung.

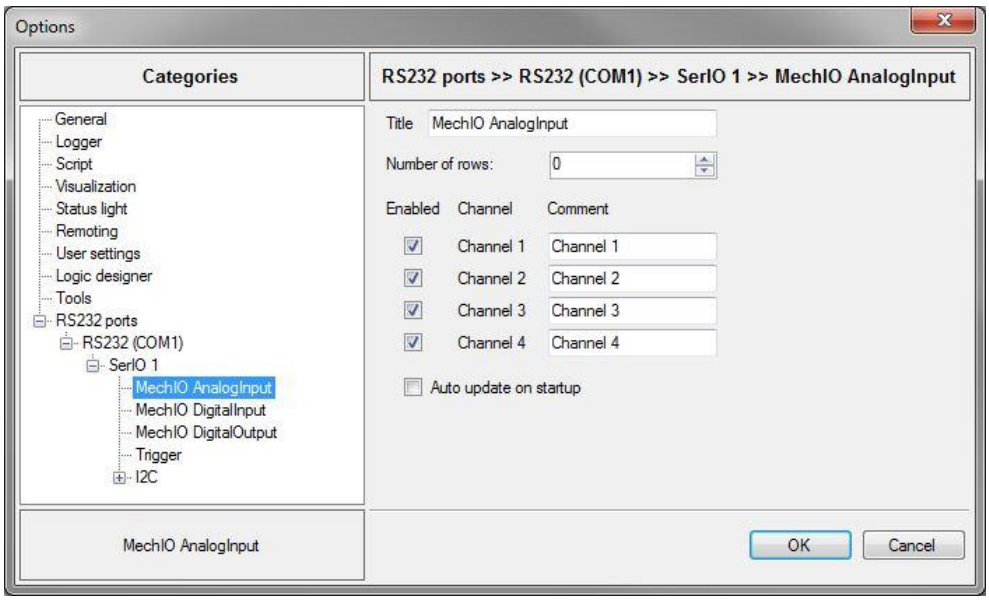


Abbildung 55: Konfiguration der analogen Eingänge

Folgende Bereiche können konfiguriert werden:

- Title: Ein Name, welcher für den analogen Eingang zur Anzeige und Ansteuerung verwendet werden soll
- Number of rows: Die Anzahl an Reihen, welche in der Anzeige für die Eingangskanäle verwendet werden soll. Ist hier der Wert „0“ eingetragen, so wird die Anzahl der Reihen und Spalten automatisch ermittelt, um die Darstellung möglichst platzsparend zu halten
- Enabled: Für jeden Kanal kann ausgewählt werden, ob dieser angezeigt werden soll (nicht benutzte Kanäle können somit ausgeblendet werden)
- Comment: Für jeden Kanal kann ein Name vergeben werden, welcher zur Anzeige und Ansteuerung verwendet wird
- Auto update on startup: Ist dieses „Flag“ aktiv, so wird der zugehörige Kommunikationskanal beim Start des Toolmonitors automatisch geöffnet und der Status der Eingangskanäle wird einmalig ausgelesen und angezeigt

8.2.2.3. MechIO 16 Digital Input

Hier erfolgt die Konfiguration der digitalen Eingänge der Mechanik - Steuerung.

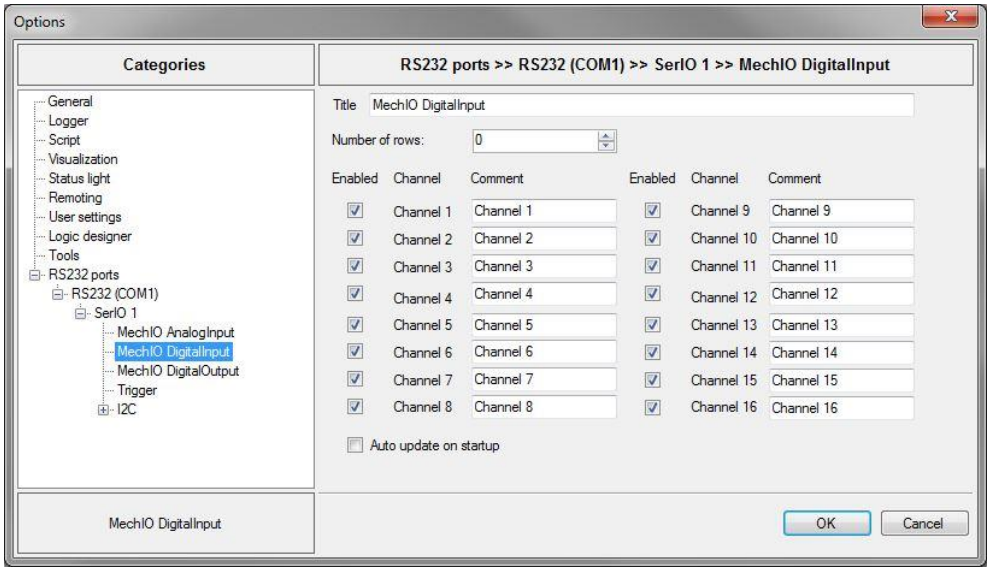


Abbildung 56: Konfiguration der digitalen Eingänge

Folgende Bereiche können konfiguriert werden:

- Title: Ein Name, welcher für die digitalen Eingänge zur Anzeige und Ansteuerung verwendet werden soll
- Number of rows: Die Anzahl an Reihen, welche in der Anzeige für die Eingangskanäle verwendet werden soll. Ist hier der Wert „0“ eingetragen, so wird die Anzahl der Reihen und Spalten automatisch ermittelt, um die Darstellung möglichst platzsparend zu halten
- Enabled: Für jeden Kanal kann ausgewählt werden, ob dieser angezeigt werden soll (nicht benutzte Kanäle können somit ausgeblendet werden)
- Comment: Für jeden Kanal kann ein Name vergeben werden, welcher zur Anzeige und Ansteuerung verwendet wird
- Auto update on startup: Ist dieses „Flag“ aktiv, so wird der zugehörige Kommunikationskanal beim Start des Toolmonitors automatisch geöffnet und der Status der Eingangskanäle wird einmalig ausgelesen und angezeigt

8.2.2.4. MechIO 16 Digital Output

Hier erfolgt die Konfiguration der digitalen Ausgänge der Mechanik - Steuerung.

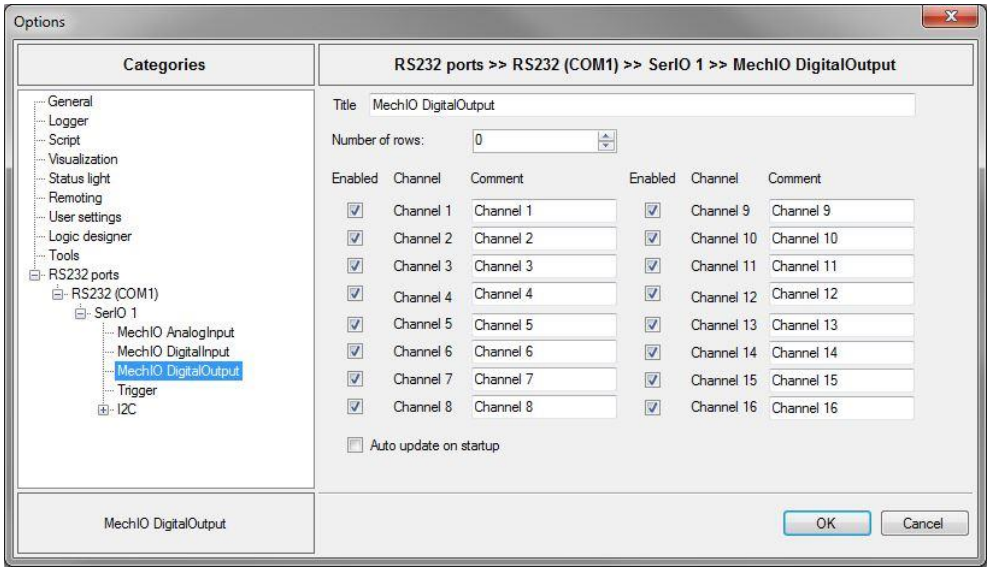


Abbildung 57: Konfiguration der digitalen Ausgänge

Folgende Bereiche können konfiguriert werden:

- Title: Ein Name, welcher für die digitalen Ausgänge zur Anzeige und Ansteuerung verwendet werden soll
- Number of rows: Die Anzahl an Reihen, welche in der Anzeige für die Ausgangskanäle verwendet werden soll. Ist hier der Wert „0“ eingetragen, so wird die Anzahl der Reihen und Spalten automatisch ermittelt, um die Darstellung möglichst platzsparend zu halten
- Enabled: Für jeden Kanal kann ausgewählt werden, ob dieser angezeigt werden soll (nicht benutzte Kanäle können somit ausgeblendet werden)
- Comment: Für jeden Kanal kann ein Name vergeben werden, welcher zur Anzeige und Ansteuerung verwendet wird
- Auto update on startup: Ist dieses „Flag“ aktiv, so wird der zugehörige Kommunikationskanal beim Start des Toolmonitors automatisch geöffnet und der Status der Ausgangskanäle wird einmalig ausgelesen und angezeigt

8.2.2.5. MechIO 16 Trigger

Hier erfolgt die Konfiguration der internen Trigger der Mechaniksteuerung.

Es können maximal acht interne Trigger definiert werden. Die Mechanik - Steuerung kontrolliert zyklisch die Eingänge und setzt nach Übereinstimmung mit der Triggerbedingung die entsprechenden Ausgangszustände bzw. das entsprechende „Merkerflag“. Die Priorität fällt mit steigendem „Index“. Das bedeutet, dass „Index 0“ immer den Ausgabezustand von z.B. „Index 1“ überschreibt. „Trigger Index 0“ wird nach „Power - UP“ aus dem EEPROM geladen und hat höchste Priorität.

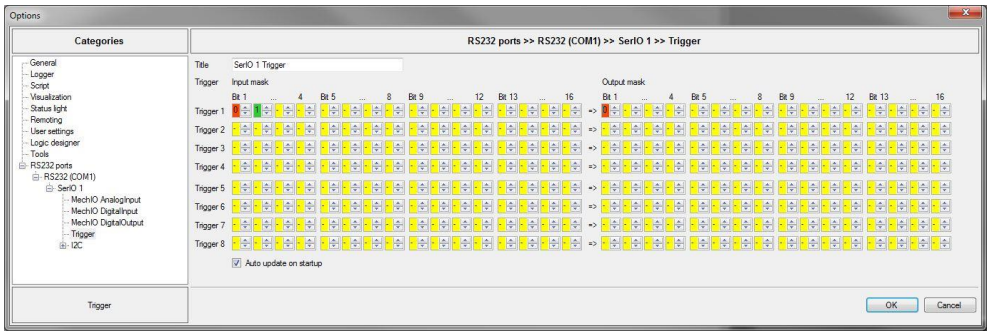


Abbildung 58: Konfiguration der internen Trigger der MechIO 16

Das Trigger Interface besteht aus folgenden Bereichen:

- Title: Ein Name, welcher für den internen Trigger zur Anzeige und Ansteuerung verwendet werden soll
- Input mask: Hier wird das Eingangsmuster für den entsprechenden Trigger eingestellt. Dabei bedeutet ein „-“, dass das entsprechende Bit für den Trigger irrelevant ist, eine „0“, dass das entsprechende Bit „Low“ sein muss und eine „1“, dass das entsprechende Bit „High“ sein muss
- Output mask: Hier wird das zu setzende Ausgangsmuster für den entsprechenden Trigger eingestellt. Dabei bedeutet ein „-“, dass das entsprechende Bit für den Trigger irrelevant ist, eine „0“, dass das entsprechende Bit auf „Low“ gesetzt wird und eine „1“, dass das entsprechende Bit „High“ gesetzt wird
- Auto update on startup: Ist dieses „Flag“ aktiv, so wird der zugehörige Kommunikationskanal beim Start des Toolmonitors automatisch geöffnet und die Trigger - Konfiguration an die Mechanik - Steuerung gesendet. Ist dieses „Flag“ nicht gesetzt, erfolgt keine automatische Übertragung an die Mechanik - Steuerung. Die Trigger - Konfiguration kann dann nur noch durch manuelles Auslösen oder über das *virtuelle Interface* gesetzt werden

8.2.3. Adapter CPU

Dieser SerIO - Gerätetyp stellt die MCD Mechanik - Steuerung „Adapter CPU“ dar.

8.2.3.1. I^2C

Hier erfolgt die Festlegung, wie viele I^2C - Ports von der „Adapter CPU“ angesteuert werden sollen.

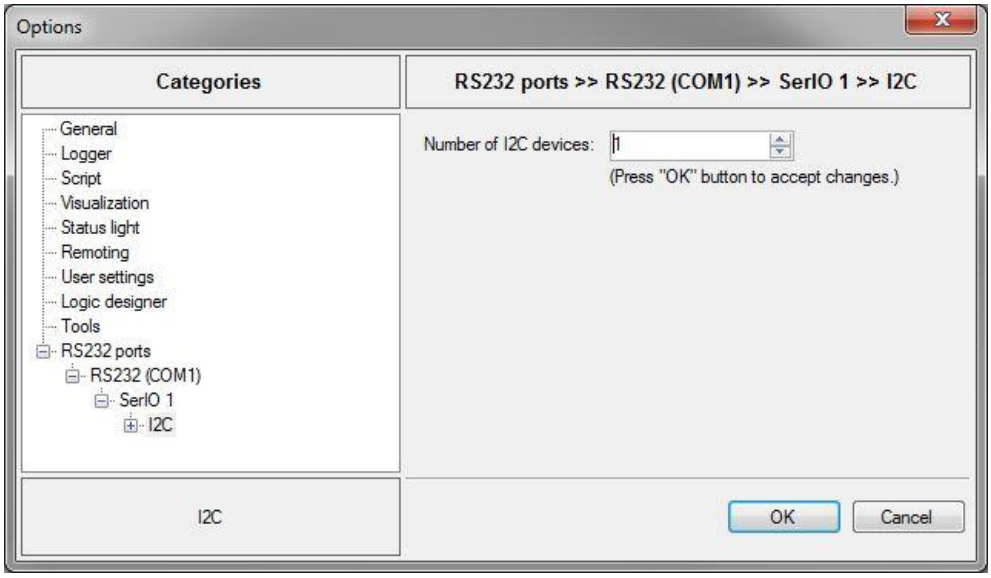


Abbildung 59: Auswahl der Anzahl der I^2C - Ports

8.2.4. Pass / Fail - Anzeige

Dieser SerIO - Gerätetyp stellt die MCD Pass / Fail - Anzeige dar.

8.2.4.1. I^2C

Hier erfolgt die Festlegung, wie viele I^2C - Ports von der Pass / Fail - Anzeige angesteuert werden sollen.

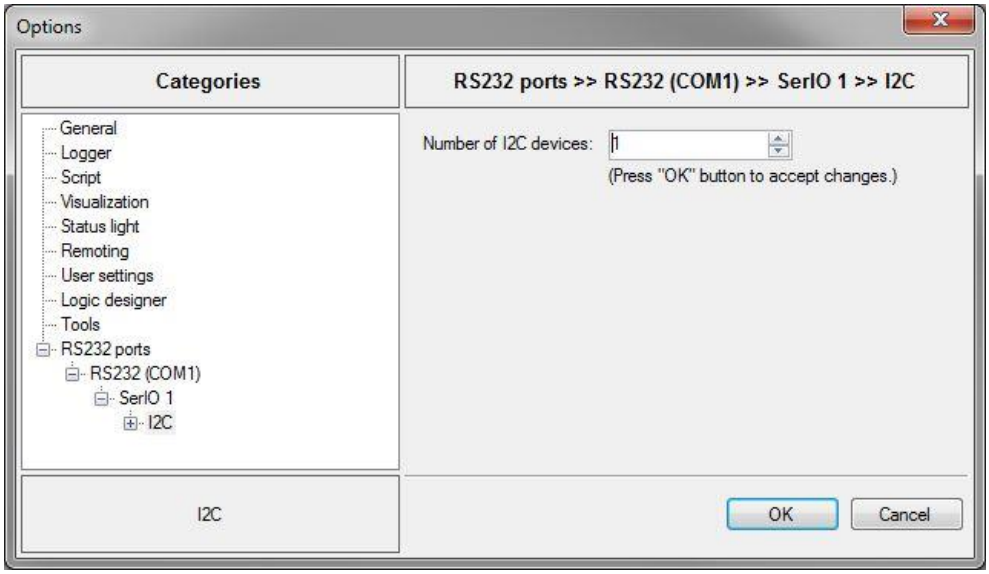


Abbildung 60: Auswahl der I^2C - Ports

8.2.4.2. Pass / Fail Digital Output

Hier erfolgt die Konfiguration der digitalen Ausgänge der Pass / Fail - Anzeige.

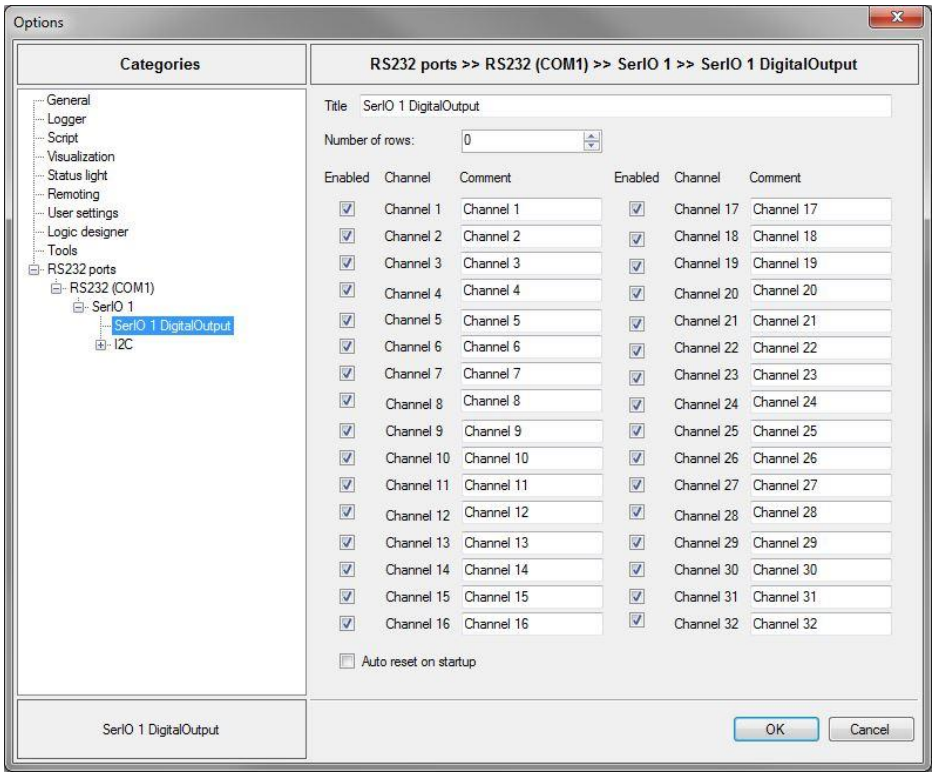


Abbildung 61: Konfiguration der digitalen Ausgänge

Folgende Bereiche können konfiguriert werden:

- Title: Ein Name, welcher für die digitalen Ausgänge zur Anzeige und Ansteuerung verwendet werden soll
- Number of rows: Die Anzahl an Reihen, welche in der Anzeige für die Ausgangskanäle verwendet werden soll. Ist hier der Wert „0“ eingetragen, so wird die Anzahl der Reihen und Spalten automatisch ermittelt, um die Darstellung möglichst platzsparend zu halten
- Enabled: Für jeden Kanal kann ausgewählt werden, ob dieser angezeigt werden soll (nicht benutzte Kanäle können somit ausgeblendet werden)
- Comment: Für jeden Kanal kann ein Name vergeben werden, welcher zur Anzeige und Ansteuerung verwendet wird
- Auto update on startup: Ist dieses „Flag“ aktiv, so wird der zugehörige Kommunikationskanal beim Start des Toolmonitors automatisch geöffnet und der Status der Ausgangskanäle wird einmalig zurückgesetzt

8.3. I²C

Für jedes zu verwendende I²C - Gerät kann hier der Typ und die I²C - Adresse(n) ausgewählt werden.

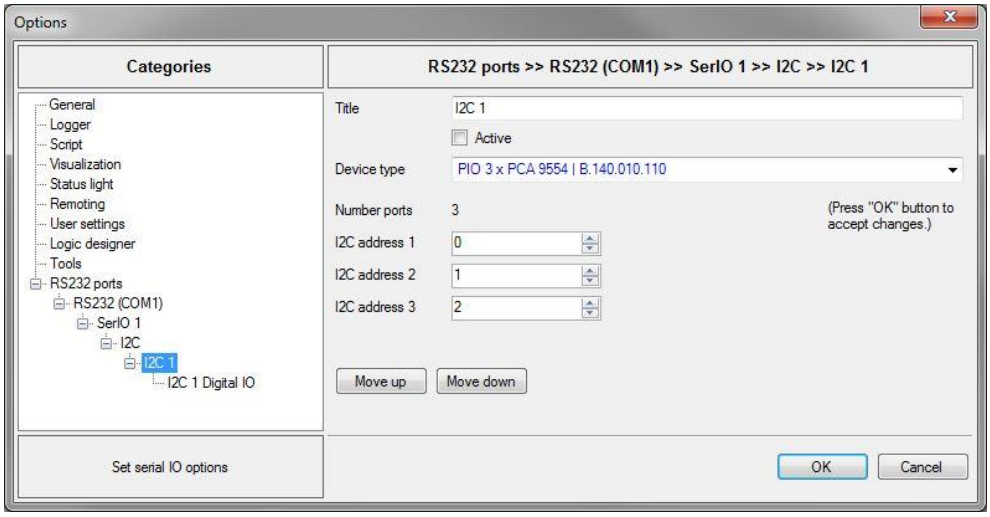


Abbildung 62: Konfiguration eines I²C - Gerätes

Folgende Typen stehen zur Verfügung:

- Unknown (stellt Grundfunktionen für unspezifische Anforderungen zur Verfügung)
- PCF8574 | PIO 8 channels per port
- PCF8574A | PIO 8 channels per port
- PCA9554 | PIO 8 channels per port
- PCA9554A | PIO 8 channels per port
- PIO 3 x PCA 9554 | B.140.010.110
- PIO 3 x PCA 9554A | B.140.010.210
- PIO 3 x PCF 8574 | B.140.010.310
- PIO 3 x PCF 8574A | B.140.010.410
- RelMUX16 | Relais multiplexer 16 channels 7442
- PST30M | Power supply for tuner applications master B.170.030.112
- PST30S | Power supply for tuner applications slave B.170.030.212
- PST30L | Power supply for tuner applications linear slave B.170.030.412

Folgende Bereiche können konfiguriert werden:

- Title: Ein Name, welcher für das I^2C - Gerät zur Anzeige und Ansteuerung verwendet werden soll
- Active: Ein „Flag“, welches kennzeichnet, dass das Gerät verwendet werden kann
- Device type: Der I^2C - Gerätetyp (siehe oben)
- I^2C address 1 .. n: Die I^2C - Adresse (n) für das I^2C - Gerät (jedes an einem gemeinsamen I^2C - Port angeschlossene I^2C - Gerät muss mit einer oder mehreren eindeutigen I^2C - Adresse (n) konfiguriert werden, Anzahl der notwendigen I^2C Adressen hängt von I^2C - Gerätetyp ab)
- Move up / Move down: Mit Hilfe dieser Buttons kann das ausgewählte I^2C - Gerät in der Liste der I^2C - Geräte verschoben werden

8.3.1. I^2C Digitale IO Geräte (PIO)

Eine Reihe von I^2C - Geräten dienen der digitalen Ein - und Ausgabe und basieren auf einer oder mehrerer I^2C - PIO und werden auf die gleiche Art und Weise konfiguriert.

Folgende Typen stehen zur Verfügung:

- PCF8574 | PIO 8 channels per port
- PCF8574A | PIO 8 channels per port
- PCA9554 | PIO 8 channels per port
- PCA9554A | PIO 8 channels per port
- PIO 3 x PCA 9554 | B.140.010.110
- PIO 3 x PCA 9554A | B.140.010.210
- PIO 3 x PCF 8574 | B.140.010.310
- PIO 3 x PCF 8574A | B.140.010.410
- RelMUX16 | Relais multiplexer 16 channels 7442

8.3.1.1. PIO Digital IO

Hier erfolgt die Konfiguration der digitalen Ein - und Ausgänge, welche über das konfigurierte I^2C - Gerät gesteuert werden sollen.

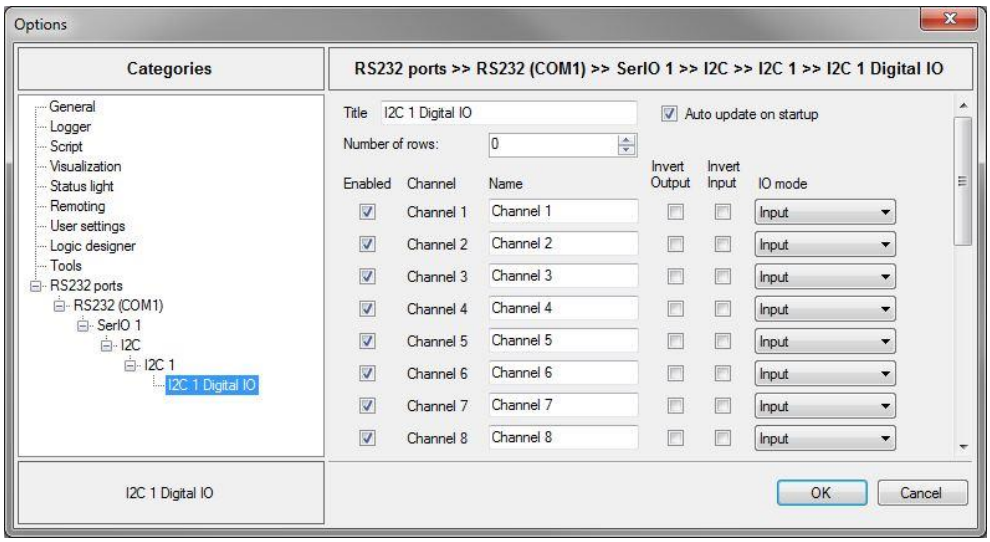


Abbildung 63: Konfiguration der digitalen Ein - und Ausgänge

Folgende Bereiche können konfiguriert werden:

- Title: Ein Name, welcher für die digitalen Ein - und Ausgänge zur Anzeige und Ansteuerung verwendet werden soll
- Auto update on startup: Ist dieses „Flag“ aktiv, so wird der zugehörige Kommunikationskanal beim Start des Toolmonitors automatisch geöffnet und der Status der Ein - und Ausgangskanäle wird einmalig ausgelesen und angezeigt
- Number of rows: Die Anzahl an Reihen, welche in der Anzeige für die Ein - und Ausgangskanäle verwendet werden soll. Ist hier der Wert „0“ eingetragen, so wird die Anzahl der Reihen und Spalten automatisch ermittelt, um die Darstellung möglichst platzsparend zu halten
- Enabled: Für jeden Kanal kann ausgewählt werden, ob dieser angezeigt werden soll (nicht benutzte Kanäle können somit ausgeblendet werden)
- Name: Für jeden Kanal kann ein Name vergeben werden, welcher zur Anzeige und Ansteuerung verwendet wird
- Invert Output: Für jeden Kanal kann hier angegeben werden, ob der Eingang invertiert werden soll. Diese Option ist nicht für alle I^2C - Geräte verfügbar
- Invert Input: Für jeden Kanal kann hier angegeben werden, ob der Ausgang invertiert werden soll. Diese Option ist nicht für alle I^2C - Geräte verfügbar
- IO mode: Für jeden Kanal kann hier angegeben werden, ob der Kanal als Eingang, Ausgang oder als bidirektionaler Port (mit entsprechendem „Default“) einzustellen ist (Option nicht für alle I^2C - Geräte verfügbar)

8.3.2. I^2C - Module zur Stromerzeugung

Eine Reihe von I^2C - Geräten dienen der Stromversorgung über spezielle Stromversorgungsmodule mit einstellbarer Frequenz des Reglerbausteins.

Folgende Geräte sind davon betroffen:

- PST30M | Power supply for tuner applications master B.170.030.112
- PST30S | Power supply for tuner applications slave B.170.030.212
- PST30L | Power supply for tuner applications linear slave B.170.030.412

8.3.2.1. Power Supply für Tuner Applications Master

Bei diesem I²C - Gerät handelt es sich um eine Stromversorgung mit einstellbarer Frequenz des Reglerbausteins.

Eingestellt werden können die Namen für die Anzeige und Steuerung der Sollwerte („Settings“) und der Messwerte („Measurement“), die Frequenz des Reglerbausteines und der „Shunt“ für die Strommessung.

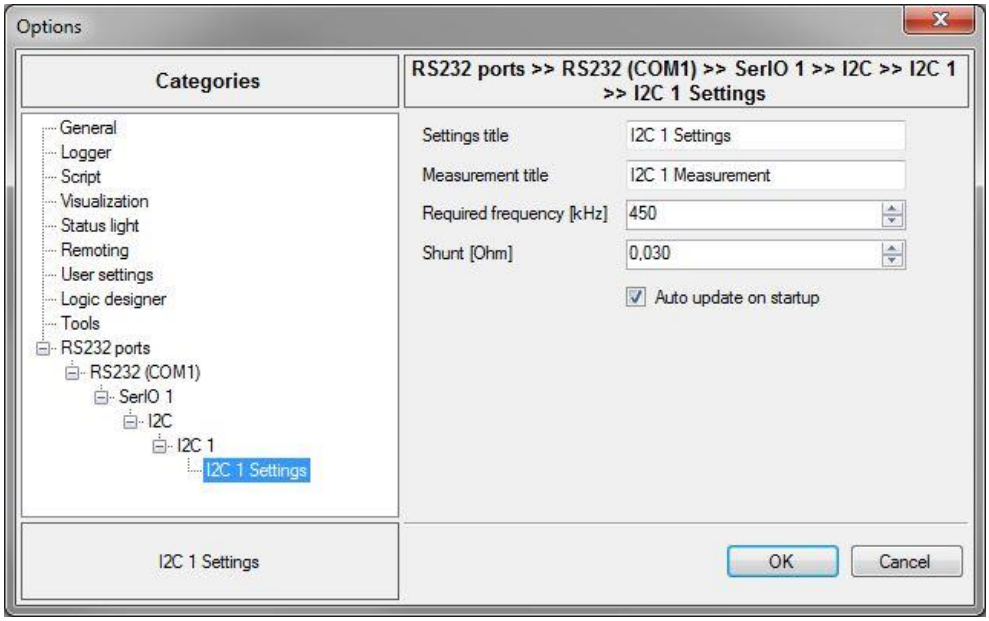


Abbildung 64: Konfiguration des Power Supply für Tuner Applications Master

8.3.2.2. Power Supply für Tuner Applications Slave

Bei diesem I²C - Gerät handelt es sich um ein „Slave“ - Modul für eine Stromversorgung mit einstellbarer Frequenz des Reglerbausteins.

Eingestellt werden können die Namen für die Anzeige und Steuerung der Sollwerte („Settings“) und der Messwerte („Measurement“) und der Shunt für die Strommessung.

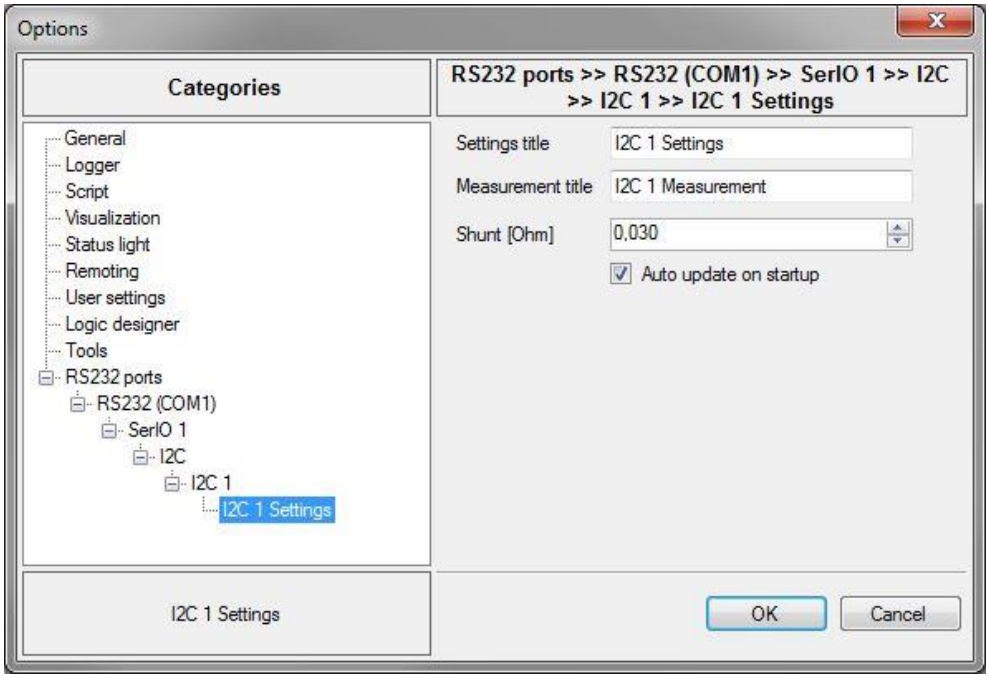


Abbildung 65: Konfiguration des Power Supply for Tuner Applications Slaves

8.3.2.3. Power Supply für Tuner Applications Linear Slave

Bei diesem I^2C - Gerät handelt es sich um ein „Slave“ - Modul mit Linearregler für eine Stromversorgung mit einstellbarer Frequenz des Reglerbausteins.

Eingestellt werden können die Namen für die Anzeige und Steuerung der Sollwerte („Settings“) und der Messwerte („Measurement“) und der „Shunt“ für die Strommessung.

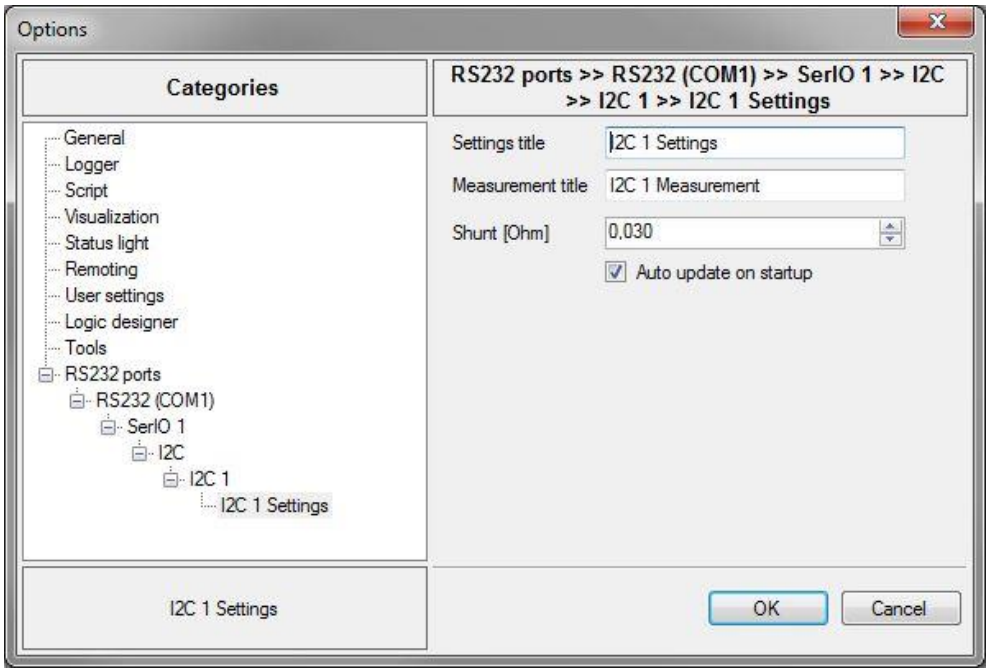


Abbildung 66: Konfiguration des Power Supply für Tuner Applications Linear Slaves

8.4. General

Hier können allgemeine Einstellungen für den Toolmonitor vorgenommen werden.

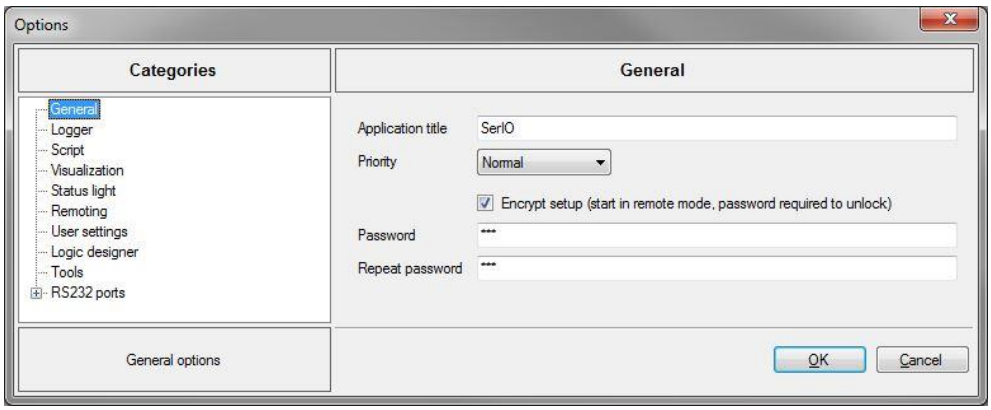


Abbildung 67: Allgemeine Einstellungen

Folgende Bereiche können konfiguriert werden:

- Applications title: Hier kann der Fenstertitel des Toolmonitors angepasst werden
- Priority: Hier kann Basispriorität des Toolmonitors eingestellt werden. Dieser Wert sollte nur bei wirklichem Bedarf geändert werden
- Encrypt setup: Hier kann festgelegt werden, ob das Setup verschlüsselt abgespeichert werden soll. In diesem Fall wird der Toolmonitor im Remote - Mode gestartet. Außerdem ist ein Passwort festzulegen, mit welchem der Toolmonitor freigeschaltet werden kann. Zur Kontrolle ist dieses Passwort zu bestätigen. Geht das Passwort verloren, kann das zugehörige Setup nicht mehr geändert werden (wird der Toolmonitor mit einem Passwort geschützt, wird der Local - / Remote - Mode nicht mehr ferngesteuert)

8.5. Logger

Hier können Einstellungen für das „Logging“ im Toolmonitor vorgenommen werden.

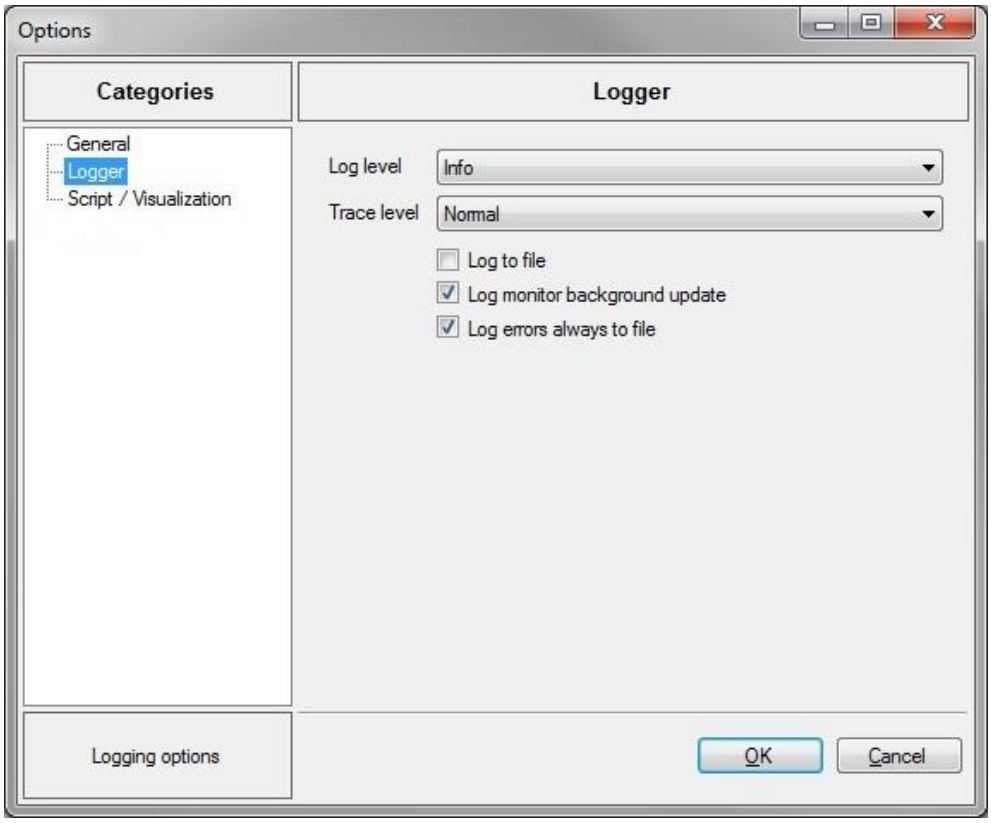


Abbildung 68: Logger

8.5.1. Log - Level

Hier wird das Log - Level für allgemeine Ereignismeldungen, Warnungen, Fehler usw. eingestellt.

Zur Auswahl stehen:

- ExtendedDebug
- Debug
- Trace
- Info
- Warning
- ErrorTrace
- CriticalWarning
- Error

8.5.2. Trace - Level

Hier wird das Trace - Level für die Protokollierung von gesendeten oder empfangen Nachrichten eingestellt.

Zur Auswahl stehen:

- Streaming
- Cyclic
- Normal

8.5.3. Log to File

Ist diese Option aktiviert, so werden aller Ereignismeldungen in einer Datei abgelegt. Die Datei setzt sich aus dem Namen des Toolmonitors und einem Zeitstempel zusammen. Die Datei wird im selben Ordner angelegt, in dem sich der Toolmonitor befindet.

8.5.4. Log Monitor Background Update

Normalerweise werden die Ereignismeldungen immer erzeugt, auch wenn der Monitor zur Anzeige der Ereignismeldungen gerade nicht aktiv ist. Ist dies nicht gewünscht, kann diese Funktion hier ausgeschaltet werden.

8.5.5. Log Errors Always to File

Normalerweise werden Fehlermeldungen immer in eine zusätzliche Log - Datei geschrieben. Ist dies nicht gewünscht, kann diese Funktion hier ausgeschaltet werden. Diese Datei setzt sich aus dem Namen des Toolmonitors und dem Zusatz „Exceptions“ zusammen. Die Datei wird im selben Ordner angelegt, in dem sich der Toolmonitor befindet.

9. Programmierung

9.1. AnalogInput

Hierbei handelt es sich um eine Basisklasse für die analogen Eingänge der SerIO - Geräte. Diese Klasse stellt Funktionen zur Steuerung der analogen Eingangskanäle der SerIO - Geräte zur Verfügung.

9.1.1. GetValueNames

Hier werden die Namen für die Befehle zum Abfragen von Werten definiert. Mit diesen Namen wird der Befehl „GetValue“ des virtuellen Interfaces für die analogen Eingänge ausgelöst.

Beispiel:

```
GetValue("Channel 1")
GetValue("Channel 1.HasAutoRange")
GetValue("IsValidMeasurementType.VoltDC")
```

Aufzählungswerte:

HasAutoRange	Abfrage, ob der Kanal „AutoRange“ unterstützt (hier immer „0“)
MaxRangeIndex	Abfrage des maximalen Index für den Messbereich (hier immer „0“)
MaxAverages	Abfrage der maximalen Anzahl für die Mittelwertbildung des Messbereichs (hier immer „1“)
ChannelName	Abfrage der Spannung des analogen Eingangs
IsValidMeasurementType	Abfrage, ob das übergebene Attribut ein gültiger Messbereich ist

9.1.2. SetValueNames

Hier werden die Namen für die Befehle zum Setzen von Werten definiert. Mit diesen Namen wird der Befehl „SetValue“ des *virtuellen Interfaces* für die analogen Eingänge ausgelöst.

Beispiel:

```
SetValue("Channel 1.Averages", 0)
```

Aufzählungswerte:

Averages	Anzahl der Mittelwerte (hier immer „1“)
MeasurementRange	Umschaltung des Messbereichs (Wert muss zwischen „0“ und „5“ liegen)
RangeIndex	Messbereichsindex (hier immer „0“)
MeasurementType	Art der Messung (hier immer „VoltDC“)
Speed	Geschwindigkeit der Messung (zulässig sind „Fast“, „Slow“, „Normal“, hier aber irrelevant)

9.2. DigitalInput

Hierbei handelt es sich um eine Basisklasse für die digitalen Eingänge der SerIO - Geräte. Diese Klasse stellt Funktionen zur Steuerung der digitalen Eingangskanäle der SerIO - Geräte zur Verfügung.

9.2.1. GetValuesNames

Hier werden die Namen für die Befehle zum Abfragen von Werten definiert. Mit diesen Namen wird der Befehl „GetValue“ des *virtuellen Interfaces* für die digitalen Eingänge ausgelöst.

Beispiel:

```
GetValue("Channel 1")
GetValue("Ports")
GetValue("Channel 1.HasAutoRange")
GetValue("IsValidMeasurementType.VoltDC")
```

Aufzählungswerte:

HasAutoRange	Abfrage, ob der Kanal „AutoRange“ unterstützt (hier immer „0“)
MaxRangeIndex	Abfrage des maximalen Index für den Messbereich (hier immer „0“)
MaxAverages	Abfrage der maximalen Anzahl für die Mittelwertbildung des Messbereichs (hier immer „1“)
ChannelName	Abfrage des Zustands des digitalen Eingangs („0“ oder „1“)
IsValidMeasurementType	Abfrage, ob das übergebene Attribut ein gültiger Messbereich ist
TriggerLevel	Abfrage des Trigger - Levels
Ports	Abfrage des Zustands aller digitalen Eingangskanäle (bitweise kodiert)

9.2.2. SetValueNames

Hier werden die Namen für die Befehle zum Setzen von Werten definiert. Mit diesen Namen wird der Befehl „SetValue“ des *virtuellen Interfaces* für die digitalen Eingänge ausgelöst.

Beispiel:

```
SetValue("Channel 1.TriggerLevel", 3.5)
```

Aufzählungswerte:

Averages	Anzahl der Mittelwerte (hier immer „1“)
MeasurementRange	Umschaltung des Messbereichs (Wert muss zwischen „0“ und „5“ liegen)
RangeIndex	Messbereichsindex (hier immer „0“)
MeasurementType	Art der Messung (hier immer „VoltDC“)
Speed	Geschwindigkeit der Messung (zulässig sind „Fast“, „Slow“, „Normal“, hier aber irrelevant)
TriggerLevel	Setzen des Trigger - Levels

9.3. DigitalOutput

Hierbei handelt es sich um eine Basisklasse für die digitalen Ausgänge der SerIO - Geräte. Diese Klasse stellt Funktionen zur Steuerung der digitalen Ausgangskanäle der SerIO - Geräte zur Verfügung.

9.3.1. GetValueNames

Hier werden die Namen für die Befehle zum Abfragen von Werten definiert. Mit diesen Namen wird der Befehl „GetValue“ des virtuellen Interfaces für die digitalen Ausgänge ausgelöst.

Beispiel:

```
GetValue("Channel 1")
GetValue("Ports")
```

Aufzählungswerte:

ChannelName	Abfrage des Zustands des digitalen Ausgangs („0“ oder „1“)
Ports	Abfrage des Zustands aller digitalen Ausgangskanäle (bitweise kodiert)

9.3.2. SetEventNames

Hier werden die Namen für Events definiert. Mit diesen Namen wird der Befehl „SetEvent“ des *virtuellen Interfaces* für die digitalen Ausgänge ausgelöst.

Aufzählungswerte:

Reset	Setzen aller Ausgänge auf den Wert „0“ bzw. „Low“

9.3.3. SetValueNames

Hier werden die Namen für die Befehle zum Setzen von Werten definiert. Mit diesen Namen wird der Befehl „SetValue“ des *virtuellen Interfaces* für die digitalen Ausgänge ausgelöst.

Beispiel:

```
SetValue("Channel 1", 0)
```

Aufzählungswerte:

ChannelName	Setzen des Zustands des digitalen Ausgangs („0“ oder „1“)
Ports	Setzen des Zustands aller digitalen Ausgangskanäle (bitweise kodiert)

9.4. I^2CBaseLine

Hierbei handelt es sich um eine Basisklasse für die I^2C - Basiskommunikation. Diese Klasse implementiert einen Kommunikationskanal für die gesamte I^2C - Kommunikation eines SerIO - Gerätes.

9.5. I²CDigitalIO

Hierbei handelt es sich um eine Basisklasse für die digitalen Ein - und Ausgänge der SerIO - Geräte.

9.5.1. GetValueNames

Hier werden die Namen für die Befehle zum Abfragen von Werten definiert. Mit diesen Namen wird der Befehl „GetValue“ des *virtuellen Interfaces* für die digitalen Ein - und Ausgänge ausgelöst.

Beispiel:

```
GetValue("Channel 1")
GetValue("Ports")
GetValue("Channel 1.HasAutoRange")
GetValue("IsValidMeasurementType.VoltDC")
```

Aufzählungswerte:

HasAutoRange	Abfrage, ob der Kanal „AutoRange“ unterstützt (hier immer „0“)
MaxRangeIndex	Abfrage des maximalen Index für den Messbereich (hier immer „0“)
MaxAverages	Abfrage der maximalen Anzahl für die Mittelwertbildung des Messbereichs (hier immer „1“)
ChannelName	Abfrage des Zustands des digitalen Eingangs („0“ oder „1“)
IsValidMeasurementType	Abfrage, ob das übergebene Attribut ein gültiger Messbereich ist
Port0	Abfrage des Zustands aller digitalen Eingangskanäle von Port 0 (8 Kanäle, bitweise kodiert)
Port1	Abfrage des Zustands aller digitalen Eingangskanäle von Port 1 (8 Kanäle, bitweise kodiert)
Port2	Abfrage des Zustands aller digitalen Eingangskanäle von Port 2 (8 Kanäle, bitweise kodiert)
Port3	Abfrage des Zustands aller digitalen Eingangskanäle von Port 3 (8 Kanäle, bitweise kodiert)
Ports	Abfrage des Zustands aller digitalen Eingangskanäle (bitweise kodiert)

9.5.2. SetEventNames

Hier werden die Namen für Events definiert. Mit diesen Namen wird der Befehl „SetEvent“ des *virtuellen Interfaces* für die digitalen Ein - und Ausgänge ausgelöst.

Aufzählungswerte:

Reset	Setzen aller digitalen Ein - und Ausgänge auf den Wert entsprechend den Grundeinstellungen zurück. Ausgänge werden auf „0“ bzw. „Low“ gesetzt

9.5.3. SetValueNames

Hier werden die Namen für die Befehle zum Setzen von Werten definiert. Mit diesen Namen wird der Befehl „SetValue“ des *virtuellen Interfaces* für die digitalen Ein - und Ausgänge ausgelöst.

Beispiel:

```
SetValue("Channel 1", 1)
SetValue("Channel 1.Mode", 1)
SetValue("Ports0Mode", 0)
```

Aufzählungswerte:

Averages	Anzahl der Mittelwerte (hier immer „1“)
MeasurementRange	Umschaltung des Messbereichs (Wert muss zwischen „0“ und „5“ liegen)
RangeIndex	Messbereichsindex (hier immer „0“)
MeasurementType	Art der Messung (hier immer „VoltDC“)
Speed	Geschwindigkeit der Messung (zulässig sind „Fast“, „Slow“, „Normal“, hier aber irrelevant)
Mode	Setzen des Kanals als Eingang oder Ausgang. Dieser Befehl ist nur zulässig, wenn der Kanal dies unterstützt und in den Grundeinstellungen ein Umschalten zugelassen ist (Mode = 0: Eingang, Mode = 1: Ausgang)
PortsMode	Setzen aller Kanäle als Eingang oder Ausgang. Dieser Befehl ist nur zulässig, wenn das Modul dies unterstützt und in den Grundeinstellungen ein Umschalten zugelassen ist (Mode = 0: Eingang, Mode = 1: Ausgang)
Port0Mode	Setzen aller Kanäle von Port 0 als Eingang oder Ausgang. Dieser Befehl ist nur zulässig, wenn das Modul dies unterstützt und in den Grundeinstellungen ein Umschalten zugelassen ist (Mode = 0: Eingang, Mode = 1: Ausgang)
Port1Mode	Setzen aller Kanäle von Port 1 als Eingang oder Ausgang. Dieser Befehl ist nur zulässig, wenn das Modul dies unterstützt und in den Grundeinstellungen ein Umschalten zugelassen ist (Mode = 0: Eingang, Mode = 1: Ausgang)
Port2Mode	Setzen aller Kanäle von Port 2 als Eingang oder Ausgang. Dieser Befehl ist nur zulässig, wenn das Modul dies unterstützt und in den Grundeinstellungen ein Umschalten zugelassen ist (Mode = 0: Eingang, Mode = 1: Ausgang)
Port3Mode	Setzen aller Kanäle von Port 3 als Eingang oder Ausgang. Dieser Befehl ist nur zulässig, wenn das Modul dies unterstützt und in den Grundeinstellungen ein Umschalten zugelassen ist (Mode = 0: Eingang, Mode = 1: Ausgang)
Port0	Setzen des Zustands aller digitalen Ausgangskanäle von Port 0 (8 Kanäle, bitweise kodiert)
Port1	Setzen des Zustands aller digitalen Ausgangskanäle von Port 1 (8 Kanäle, bitweise kodiert)
Port2	Setzen des Zustands aller digitalen Ausgangskanäle von Port 2 (8 Kanäle, bitweise kodiert)
Port3	Setzen des Zustands aller digitalen Ausgangskanäle von Port 3 (8 Kanäle, bitweise kodiert)
Ports	Setzen des Zustands aller digitalen Ausgangskanäle (bitweise kodiert)

9.6. I²CLine

Hierbei handelt es sich um eine Basisklasse für die I²C - Kommunikation. Diese Klasse implementiert einen Kommunikationskanal für die I²C - Kommunikation eines I²C - Moduls.

9.7. Line

Hierbei handelt es sich um eine Basisklasse für die serielle Kommunikation über verschiedene Kommunikationswege. Dazu gehören I²CBaseLine, I²CLine, SerIOLine und RS232.

9.7.1. VirtualInterfaceCommands

Hier werden die Namen für die Befehle des *virtuellen Interfaces* des Kommunikationskanals definiert. Mit diesen Namen werden die Befehle „SetEvent“, „SetValue“, „GetValue“, und „GetString“ des *virtuellen Interfaces* für den Kommunikationskanal ausgelöst.

Aufzählungswerte:

Open	Öffnen des Kommunikationskanals
Close	Schließen des Kommunikationskanals
Clear	Löschen der Daten im Sende - / Empfangspuffer des Kommunikationskanals
IsActive	Rückgabe, ob Kommunikationskanal in den Grundeinstellungen aktiviert wurde
IsOpen	Rückgabe, ob Kommunikationskanal geöffnet ist
DataAvailable	Rückgabe, ob Daten zum Empfang vorhanden sind (wird nicht von allen Kommunikationskanälen unterstützt)
BytesAvailable	Rückgabe der Anzahl an Bytes, die zum Empfang vorhanden sind (wird nicht von allen Kommunikationskanälen unterstützt)
Write	Ausgeben der übergebenen Daten über dem Kommunikationskanal
WriteMessage	Ausgeben der unter dem vorgegebenen Namen gespeicherten Nachricht über dem Kommunikationskanal
WriteSequence	Ausgeben der unter dem vorgegebenen Namen gespeicherten Sequenz über dem Kommunikationskanal
Read	Auslesen der Daten aus dem Kommunikationskanal
Execute	Ausführen des übergebenen Kommandos über dem Kommunikationskanal und Speichern der zugehörigen Antwort unter „Response“ (Kommando nur für Kommunikationskanäle ohne getrennte Write - /Read - Funktion vorgesehen)
ExecuteMessage	Ausführen des unter dem vorgegebenen Namen gespeicherten Kommandos über dem Kommunikationskanal und Speichern der zugehörigen Antwort unter „Response“ (Kommando nur für Kommunikationskanäle ohne getrennte Write - / Read - Funktion vorgesehen)
ExecuteSequence	Ausführen der unter dem vorgegebenen Namen gespeicherten Sequenz über dem Kommunikationskanal und Speichern der zugehörigen Antwort unter „Response“ (Kommando nur für Kommunikationskanäle ohne getrennte Write - / Read - Funktion vorgesehen)
Response	Rückgabe der zuletzt über einen „Execute“ - Befehl empfangenen Nachricht

TraceLevel	Ändern des Trace - Levels beim Senden und Empfangen über *das virtuelle Interface* (zulässige Werte: Streaming = „0“, Cycling = „1“, Normal = „2“, None = „3“)
Timeout	Ändern des Timeouts beim Senden und Empfangen über das *virtuelle Interface*
Count	Ändern der Byteanzahl beim Senden und Empfangen über das *virtuelle Interface*
Comment	Ändern des Kommentars beim Senden und Empfangen über das *virtuelle Interface*
EndOfLine	Ändern des EndOfLine - Strings beim Senden und Empfangen über das *virtuelle Interface*
ThrowTimeoutException	Ein - oder Ausschalten der Timeout - Exception beim Senden und Empfangen über das *virtuelle Interface*
GetMessage	Rückgabe der unter den angegebenen Namen gespeicherten Nachricht

9.8. RS232

Hierbei handelt es sich um eine Basisklasse für die RS232 - Kommunikation. Diese Klasse implementiert einen Kommunikationskanal über eine RS232 - Schnittstelle. Sie wird abgeleitet von der Basisklasse „Line“ und ist selbst die Ableitung für die Basisklasse „SerIOBaseLine“.

9.9. SerIOBaseLine

Hierbei handelt es sich um eine Basisklasse für die SerIO - Kommunikation via RS232. Diese Klasse implementiert einen Kommunikationskanal über eine RS232 - Schnittstelle.

9.10. SerIOLine

Hierbei handelt es sich um eine Basisklasse für die SerIO - Kommunikation. Diese Klasse implementiert einen Kommunikationskanal für die Kommunikation zu einem SerIO - Gerät.

9.11. Trigger

Hierbei handelt es sich um eine Basisklasse zur Steuerung der Trigger der Mechanik - Steuerung MechIO 16. Diese Klasse stellt Funktionen zur Steuerung der Trigger der Mechanik - Steuerung MechIO 16 bereit.

9.11.1. SetEventNames

Hier werden die Namen für Events definiert. Mit diesen Namen wird der Befehl „SetEvent“ des *virtuellen Interfaces* für die Trigger ausgelöst.

Beispiel:

```
SetEvent("Reset");
SetEvent("WriteAll");
SetEvent("Trigger1.Write");
```

Aufzählungswerte:

Reset	Zurücksetzen aller Trigger
WriteAll	Senden der Triggerkonfiguration für alle Trigger an die Mechanik - Steuerung
Write	Schreiben der Konfiguration des betreffenden Triggers an die Mechanik - Steuerung (der jeweilige Trigger muss vorangestellt werden, siehe Beispiel)
TriggerN	Name für den vorangestellten Trigger beim Befehl „Write“ (N = 1..16)

9.11.2. SetValueNames

Hier werden die Namen für die Befehle zum Setzen von Werten definiert. Mit diesen Namen wird der Befehl „SetValue“ des *virtuellen Interfaces* für den Trigger ausgelöst.

Beispiel:

```
SetValue("Trigger 1.InputMask", 3)
SetValue("Trigger 1.InputValue", 2)
SetValue("Trigger 1.OutputMask", 255)
SetValue("Trigger 1.OutputValue", 0)
SetValue("Trigger 1.Input.Bit1", -1)
SetValue("Trigger 1.Input.Bit1", 0)
SetValue("Trigger 1.Input.Bit1", 1)
SetValue("Trigger 1.Output.Bit1", -1)
SetValue("Trigger 1.Output.Bit1", 0)
SetValue("Trigger 1.Output.Bit1", 1)
```

Aufzählungswerte:

InputMask	Setzen der Eingangsmaske des entsprechenden Triggers, dabei handelt es sich um eine bitweise Kodierung der Eingänge, welche für den Trigger relevant sind (der jeweilige Trigger muss vorangestellt werden, siehe Beispiel)
InputValue	Setzen der Eingangsbedingungen des entsprechenden Triggers, dabei handelt es sich um eine bitweise Kodierung der Eingangswerte, welche für den Trigger erforderlich sind (der jeweilige Trigger muss vorangestellt werden, siehe Beispiel)
OutputMask	Setzen der Ausgangsmaske des entsprechenden Triggers, dabei handelt es sich um eine bitweise Kodierung der Ausgänge, welche für den Trigger relevant sind (der jeweilige Trigger muss vorangestellt werden, siehe Beispiel)
OutputValue	Setzen der Ausgangsbedingungen des entsprechenden Triggers, dabei handelt es sich um eine bitweise Kodierung der Ausgangswerte, welche für den Trigger erforderlich sind (der jeweilige Trigger muss vorangestellt werden, siehe Beispiel)
Input	Direkte Konfiguration der Input - Bits (siehe Beispiel)
Output	Direkte Konfiguration der Output - Bits (siehe Beispiel)
BitN	Gezieltes Ansprechen der einzelnen Bits der Eingangs - und Ausgangskonfiguration, um die Trigger zu konfigurieren (N = 1..16, zulässige Werte: „-1“ = Bit ist nicht relevant, „0“ = Bit ist „Low“ bzw. muss „Low“ sein, „1“ = Bit ist „High“ bzw. muss „High“ sein)