



Hochschule für Technik, Wirtschaft und Kultur
Fakultät Ingenieurwissenschaften

Belegarbeit

Factory Automation

Thema: Erweiterung der Fertigungsstrecke auf AS-interface

Autoren: Leon Arnecke 79511,
Johannes Trummer 87051,
Darius Nock 86960,
Jonas Endler 74636

Version vom: 6. August 2025

Prüfer: Prof. Dr.-Ing Tilo Heimbold

Inhaltsverzeichnis

1 Einführung	3
2 Umverdrahtung der Anlage	4
2.1 Spannungsversorgung	4
2.2 ASi-Leitungen	4
2.3 Verbindung der Ein- und Ausgänge	4
3 Installation und Konfiguration der Software	9
3.1 Installation und Konfiguration	9
3.2 Softwarekomponenten	12
4 Konfiguration des ASi-Netzwerkes	13
4.1 Zugriff auf das Master-Display	13
4.2 Webinterface via Browserzugriff	16
4.3 Hinzufügen neuer Teilnehmer	18
5 Programmierung des Masters	19
5.1 Variablen	20
5.2 Funktionen	20
6 Aufbau und Bedienung der Debug-Schnittstelle	23
7 Ausblick	26
Abbildungsverzeichnis	27
Literaturverzeichnis	28

1 Einführung

Im Rahmen des Moduls *Factory Automation* wurde die Aufgabe gestellt, eine bestehende Anlage, die bislang in konventioneller Sternverkabelung betrieben wurde, auf AS-Interface (ASi) umzustellen. Diese Umstellung verfolgt das Ziel, die Vorteile der ASi Technologie – insbesondere im Hinblick auf die Vereinfachung der Verkabelung, die Flexibilität bei Änderungen sowie die Reduzierung von Verdrahtungsaufwand und Fehlerquellen – praktisch zu erproben.

Im vorliegenden Projekt waren die ASi Module bereits vorab in der Anlage verbaut und die grundlegende Verkabelung vorbereitet. Die wesentlichen Aufgaben bestanden darin, die bestehende Infrastruktur zu vervollständigen, sich mit der zugehörigen Softwareumgebung vertraut zu machen, die ASi Slaves korrekt zu adressieren und zu parametrieren sowie den ASi Master entsprechend zu programmieren. Abschließend wurde die Inbetriebnahme der Anlage mit dem ASi System durchgeführt, um die Funktionsfähigkeit zu überprüfen.

Diese Arbeit dokumentiert die dabei durchgeführten Arbeitsschritte, gewonnenen Erkenntnisse sowie auftretenden Herausforderungen und deren Lösungen im Verlauf der Umstellung auf AS-Interface.

2 Umverdrahtung der Anlage

Um die Anlage mit AS-Interface zu betreiben, musste sie entsprechend verdrahtet werden, sodass der Master und die Slaves miteinander kommunizieren können, sämtliche Baugruppen mit Spannung versorgt werden und die Ein- und Ausgänge der Slaves mit den zugehörigen Sensoren und Aktoren verbunden sind.

2.1 Spannungsversorgung

Die Spannungsversorgung der Anlage erfolgt über ein primäres Netzteil, das die 230V-Netzspannung in eine 24V-Gleichspannung umwandelt. Diese 24V werden allen Steckboards zur Verfügung gestellt. Ein weiteres Netzteil (DC/DC-Wandler) erzeugt aus den 24V eine 30V-Gleichspannung, die den ASi Master sowie die ASi Slaves mit Energie versorgt. Die Anschlüsse für die 24V-Versorgungsspannung sind im unteren Bereich der Boards angeordnet und stellen die Spannungsversorgung sämtlicher Boards sicher.

2.2 ASi-Leitungen

Die beiden ASi-Leitungen, ASi Plus und ASi Minus, gehen vom Master aus und werden mit allen Boards verbunden. Über diese Leitungen wird einerseits die 30V-Gleichspannung zur Versorgung der Slaves bereitgestellt und andererseits die Kommunikation zwischen Master und Slaves realisiert. Das bedeutet, dass über dasselbe Leitungspaar gleichzeitig Energie- und Datenübertragung erfolgen.

Die beiden Leitungen werden ebenfalls an der Unterseite der Boards geführt und sind mit "ASi+" und "ASi-" gekennzeichnet.

2.3 Verbindung der Ein- und Ausgänge

Auf den Boards, die die Verbindung zu den Tastern, Schaltern und Aktoren herstellen, besteht die Möglichkeit, diese mittels Bananensteckern flexibel auf die Ein- und Ausgänge der Slave-I/O-Module zu schalten. An dieser Stelle ist es eine Frage der Philosophie, wie die einzelnen Funktionen der Anlage verdrahtet werden. In diesem Fall wurde entschieden, die Funktionen exakt in der Reihenfolge auf die Slave-Module zu legen, wie sie auf den Boards angeordnet sind – also von links oben nach rechts unten. Befindet sich beispielsweise die Ansteuerung eines Ventils links oben, so ist diese auch auf den entsprechenden Ausgang links oben verdrahtet.

Dieses Vorgehen ignoriert zwar bewusst die chronologische Nummerierung der Ein- und Ausgänge, bietet jedoch den Vorteil, dass auf einen Blick erkennbar ist, welcher Ausgang verwendet wird. Ein lästiges "Kabelverfolgen", um den richtigen Ausgang zu identifizieren, entfällt somit.

Daraus entstehen die folgenden Tabellen:

Lager

Funktion	Beschreibung	I/O	Verdrahtung
Bandende Links	B6	I	I/O 1
Palette in Position	B1	I	I/O 2
Hub unten	B3	I	I/O 3
Schalter	S1	I	I/O 4
Bandende rechts	B5	I	I/O 5
Füllstand	B2	I	I/O 6
Hub oben	B4	I	I/O 7
Schalter	S2	I	I/O 8

Tabelle 1: Eingänge der Station Lager

Funktion	Beschreibung	I/O	Verdrahtung
Band Linkslauf	M4	O	I/O 9
Stopper	M1	O	I/O 10
Vereinzeln	M3	O	I/O 11
LED rot	P1	O	I/O 12
Band Rechtlauf	M5	O	I/O 13
Hubzylinder	M2	O	I/O 14
LED grün	P2	O	I/O 16

Tabelle 2: Ausgänge der Station Lager

Handling

An der Handling-Station sind zwei Slaves installiert, die jeweils über 16 digitale I/Os verfügen. Es wurde eine spezifische Aufteilung realisiert: Der erste Slave erfasst sämtliche Eingangssignale, während der zweite Slave die Ausgangssignale schaltet. Die detaillierte Belegung ist der nachfolgenden Tabelle zu entnehmen.

Funktion	Beschreibung	I/O	Verdrahtung
Bandende Links	B11	I	I/O 1
Anschlag POS. 1 eingefahren	B1	I	I/O 2
Palette in Position	B3	I	I/O 3
Arm über Band	B5	I	I/O 4
Bandende Rechts	B12	I	I/O 5
Anschlag POS. 2 eingefahren	B2	I	I/O 6
Hub eingefahren	B4	I	I/O 7
Arm über Anlage	B6	I	I/O 8
Werkstück erkannt	B7	I	I/O 9
Schalter	S1	I	I/O 10
nicht belegt			I/O 11
nicht belegt			I/O 12
Unterdruckschalter am Sauger	B8	I	I/O 13
Taster	S2	I	I/O 14
nicht belegt			I/O 15
nicht belegt			I/O 16

Tabelle 3: Eingänge der Station Handling

Funktion	Beschreibung	I/O	Verdrahtung
Band Linkslauf	M11	O	I/O 1
Anschlag POS. 1 einfahren	M3	O	I/O 2
Stopper ausfahren	M5	O	I/O 3
Arm Richtung Band	M7	O	I/O 4
Band Rechtslauf	M12	O	I/O 5
Anschlag POS. 2 einfahren	M4	O	I/O 6
Hubzylinder ausfahren	M6	O	I/O 7
Arm Richtung Anlage	M8	O	I/O 8
Sauger anschalten	M9	O	I/O 9
LED rot	P1	O	I/O 10
LED gelb	P3	O	I/O 11
nicht belegt			I/O 12
nicht belegt			I/O 13
LED grün	P2	O	I/O 14
LED blau	P4	O	I/O 15
nicht belegt			I/O 16

Tabelle 4: Ausgänge der Station Handling

Montage

Funktion	Beschreibung	I/O	Verdrahtung
Bandende Links	B4	I	I/O 1
Presse unten	B6	I	I/O 2
Schalter	S1	I	I/O 3
Bandende Rechts	B2	I	I/O 5
Palette da	B5	I	I/O 6
Taster	S2	I	I/O 7
nicht belegt		I	I/O 8

Tabelle 5: Eingänge der Station Montage

Funktion	Beschreibung	I/O	Verdrahtung
Band Linkslauf	M11	O	I/O 1
Stopper	M3	O	I/O 2
LED rot	P1	O	I/O 3
LED gelb	P3	O	I/O 4
Band Rechtslauf	M12	O	I/O 5
Presse	M4	O	I/O 6
LED grün	P2	O	I/O 7
LED blau	P4	O	I/O 8

Tabelle 6: Ausgänge der Station Montage

Demontage

Funktion	Beschreibung	I/O	Verdrahtung
Bandende Links	B4	I	I/O 1
Presse unten	B6	I	I/O 2
Schalter	S1	I	I/O 3
nicht belegt		I	I/O 4
Bandende Rechts	B2	I	I/O 5
Palette da	B5	I	I/O 6
Taster	S2	I	I/O 7
nicht belegt		I	I/O 8

Tabelle 7: Eingänge der Station Demontage

Funktion	Beschreibung	I/O	Verdrahtung
Band Linkslauf	M11	O	I/O 9
Stopper	M3	O	I/O 10
LED rot	P1	O	I/O 11
LED gelb	P3	O	I/O 12
Band Rechtslauf	M12	O	I/O 13
Presse	M4	O	I/O 14
LED grün	P2	O	I/O 15
LED blau	P4	O	I/O 16

Tabelle 8: Ausgänge der Station Demontage

3 Installation und Konfiguration der Software

3.1 Installation und Konfiguration

Die für dieses Projekt benötigte Software sowie sämtliche zugehörigen Tools und Resourcen stehen über eine Cloud-Plattform zum Download bereit. Dabei ist zu beachten, dass die Softwarelizenz an ein spezifisches Endgerät gebunden ist: Eine Installation ist jeweils nur auf einem einzelnen Laptop möglich. Soll die Software auf einem anderen Gerät eingesetzt werden, muss sie zuvor auf dem ursprünglich verwendeten Gerät vollständig deinstalliert werden. Insgesamt ist die Nutzung auf maximal drei verschiedenen Geräten erlaubt, was bei der Planung und Durchführung des Projekts berücksichtigt werden muss.

Die Programmierung des ASi Masters erfolgt über die integrierte Entwicklungsumgebung (IDE) *Eclipse*. Um Eclipse lauffähig zu machen, ist die Installation des Java Development Kit (JDK) erforderlich, wobei zwingend auf eine 32-Bit-Version[9] des JDK zurückgegriffen werden muss. Die eigentliche Programmierung des Masters wird in der Programmiersprache C unter Verwendung herstellerspezifischer Bibliotheken durchgeführt. Für den Zugriff und das Verständnis dieser Bibliotheken ist eine gesonderte Dokumentation vom jeweiligen Hersteller einzuholen, da diese Informationen nicht frei zugänglich zur Verfügung stehen. Die in dieser Arbeit verwendeten Funktionen werden an den Stellen erklärt, an denen sie gebraucht werden.

Für die Softwareinstallation kann grundsätzlich die Standardinstallation gewählt werden. Dennoch empfiehlt es sich, auf alle Installationspfade, Lizenzschlüssel und Konfigurationsdateien zu achten und diese zu speichern, da solche Informationen im späteren Projektverlauf relevant werden könnten.

Lizenz hinzufügen

Neue Lizenz hinzufügen

Bitte geben Sie Ihre Lizenzdaten ein

Name: Marco Braun

Firma: HTWK Leipzig

E-Mail: marco.braun@htwk-leipzig.de

E-Mail (Wiederholung): marco.braun@htwk-leipzig.de

Ich habe die [Datenschutzerklärung](#) zur Kenntnis genommen

Bitte wählen Sie, ob Sie einen Gutscheincode online einlösen möchten oder bereits einen Lizenzschlüssel besitzen:

Gutscheincode: [REDACTED]

Hinweis: Dieser Vorgang erfordert Zugriff auf das Internet. Wenn Sie auf diesem Computer keinen Internetzugriff besitzen, besuchen Sie bitte die Registrierungs-Website auf einem anderen Computer, um Ihren Gutscheincode einzulösen und einen Lizenzschlüssel zu erhalten. Sie benötigen eventuell die unten stehende Hardware-ID.

Registrierungs-Website: <https://licensing.bihl-wiedemann.de>

Ihre Hardware-ID: 042A 683F 111A 88FE 3960 7013

Ja, ich bin damit einverstanden, dass ich nach dem Kauf von zeitlich befristeten Softwarelizenzen von Bihl+Wiedemann per E-Mail über den bevorstehenden Ablauf der Lizenzen informiert werde. Ich habe das Recht, diesen Erinnerungs-E-Mails jederzeit zu widersprechen.

Lizenzschlüssel: [REDACTED]

[Lizenzdaten vergessen?](#)

Weiter > Abbrechen Hilfe

Abbildung 1: Installationsdaten

Nach erfolgreicher Installation stehen mehrere Softwarekomponenten zur Verfügung:

- Sicherheitsmonitor Simulator
- ASIMON360
- ASi Control Tools
- Diagnosesoftware
- Bihl+Wiedemann Suite
- Firmware-Update Tool
- Eclipse IDE

Für jedes dieser Programme ist eine eigenständige Einarbeitung notwendig, da sie unterschiedliche Aufgaben und Funktionen übernehmen und sehr komplex sind. Hier soll ausschließlich eine Übersicht gegeben werden. Die Programmübermittlung zum Master muss ebenfalls erst mittels Lizenzschlüssel freigegeben werden. Dafür musste dieser auf der Website eingegeben werden um einen weiteren Code zu generieren, der dann zur Freigabe genutzt werden kann. Hier sollen die hinterlegten Daten nochmal zusammengefasst dargestellt werden.

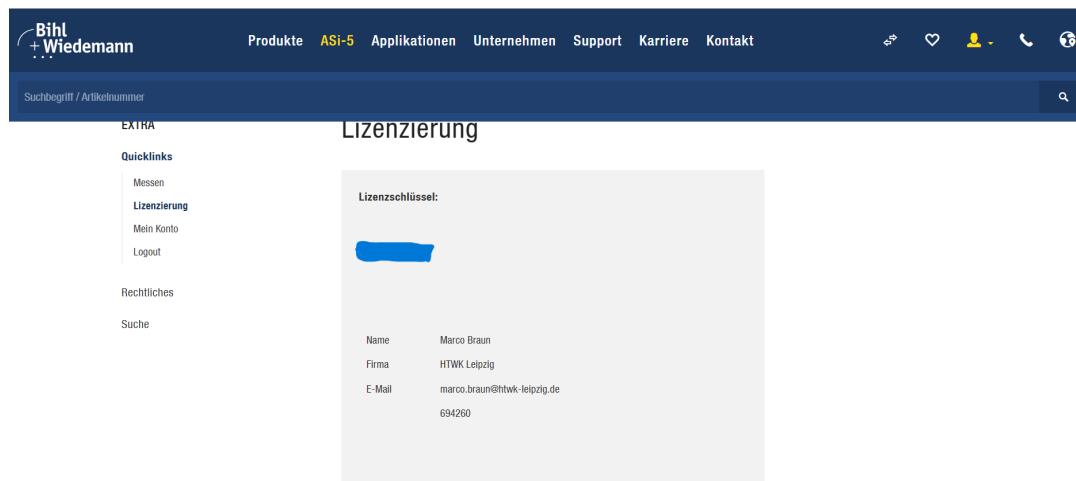


Abbildung 2: Onlinefreigabe

Der Beleuchtung des Webservers ist das nachfolgende Kapitel gewidmet, da über dieses die Konfiguration der ASi-Teilnehmer im ASi-Netzwerk festgelegt wird.

3.2 Softwarekomponenten

Im Folgenden werden die wichtigsten Softwarepakete von Bihl+Wiedemann kurz vorgestellt. Sie dienen zur Konfiguration, Diagnose, Simulation und Aktualisierung von ASi-Komponenten und sicherheitstechnischen Anwendungen.

Sicherheitsmonitor Simulator: Der Sicherheitsmonitor Simulator ist Bestandteil der Safety Suite und ermöglicht das Testen von Sicherheitskonfigurationen auf simulierten ASi-Geräten, bevor reale Hardware genutzt wird [3].

ASIMON360: ASIMON360 ist ein umfassendes Tool zur grafischen Konfiguration, Adressierung und Inbetriebnahme von ASi-3 und ASi-5 Netzwerken, unterstützt durch Video-Tutorials [4].

Diagnosesoftware: Die Diagnosesoftware erkennt und visualisiert Netzwerkfehler in Echtzeit sowohl in Standard- als auch Safety-ASi-Netzen und unterstützt schnell die Fehlersuche [5].

Bihl+Wiedemann Suite: Die Suite vereint ASIMON360, Diagnosesoftware und ASi Control Tools in einem Paket – ideal für Installation, Inbetriebnahme und Wartung [6].

Firmware-Update Tool: Das Firmware-Update Tool prüft installierte Firmware gegen die aktuelle Version und ermöglicht gesicherte Updates im Feld – z.B. als integrierte Funktion in ASi Control Tools360 [7].

4 Konfiguration des ASi-Netzwerkes

In diesem Kapitel wird die Einrichtung eines ASi-5 Netzwerks über das Gerät selbst sowie über das Webinterface des Masters erläutert.

4.1 Zugriff auf das Master-Display

Der Master selbst verfügt über ein integriertes Display mit Menü-Navigation und Status-LEDs. So ist auch ohne PC-Zugriff eine grundlegende Konfiguration möglich. Das Display kann jetzt genutzt werden, um mit dem Master in das Netzwerk der Wahl zu gehen:

1. Menü aufrufen durch drücken der Taste *OK*
2. Bis zum Punkt *Diag. Interface* nach unten scrollen und diesen auswählen
3. *Configuration* auswählen
4. IP einstellen
5. passende Subnetzmase auswählen, z.B. (192.168.1.1) und (255.255.255.0)
6. Einstellungen speichern



(a) allgemeine Ansicht



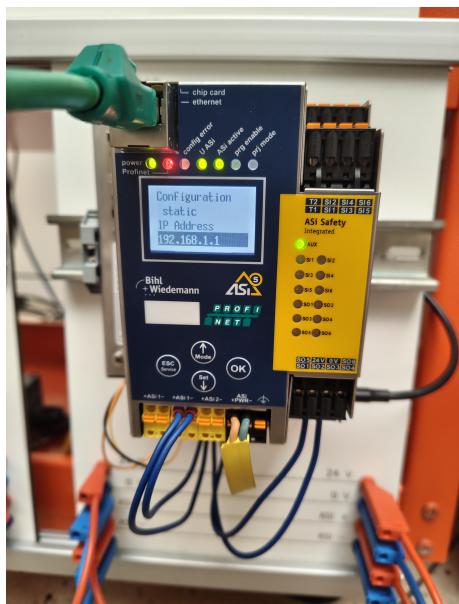
(b) Menü



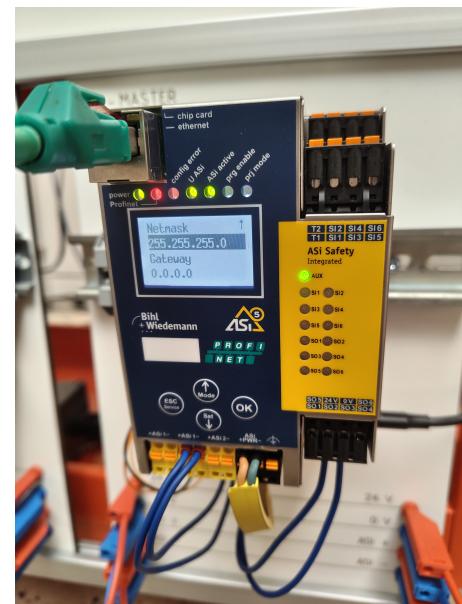
(a) Diagn. Interface



(b) Configuration



(c) IP-Adresse



(d) Subnetzmase

Abbildung 4: Einrichtung der Master-IP-Adresse

Damit der PC über eine Ethernet-Leitung mit dem Master kommunizieren kann, muss er ebenfalls Teilnehmer desselben Netzwerks sein:

1. Windows-Taste drücken, *Netzwerkverbindungen anzeigen* eingeben und auswählen
2. Rechtsklick auf die Ethernet Verbindung mit dem Master, *Eigenschaften* auswählen
3. Doppelklick auf Internetprotokoll Version 4
4. Einstellen der richtigen Adresse und Subnetzmaske, z.B. *192.168.1.10* und *255.255.255.0*, dann Speichern und Schließen

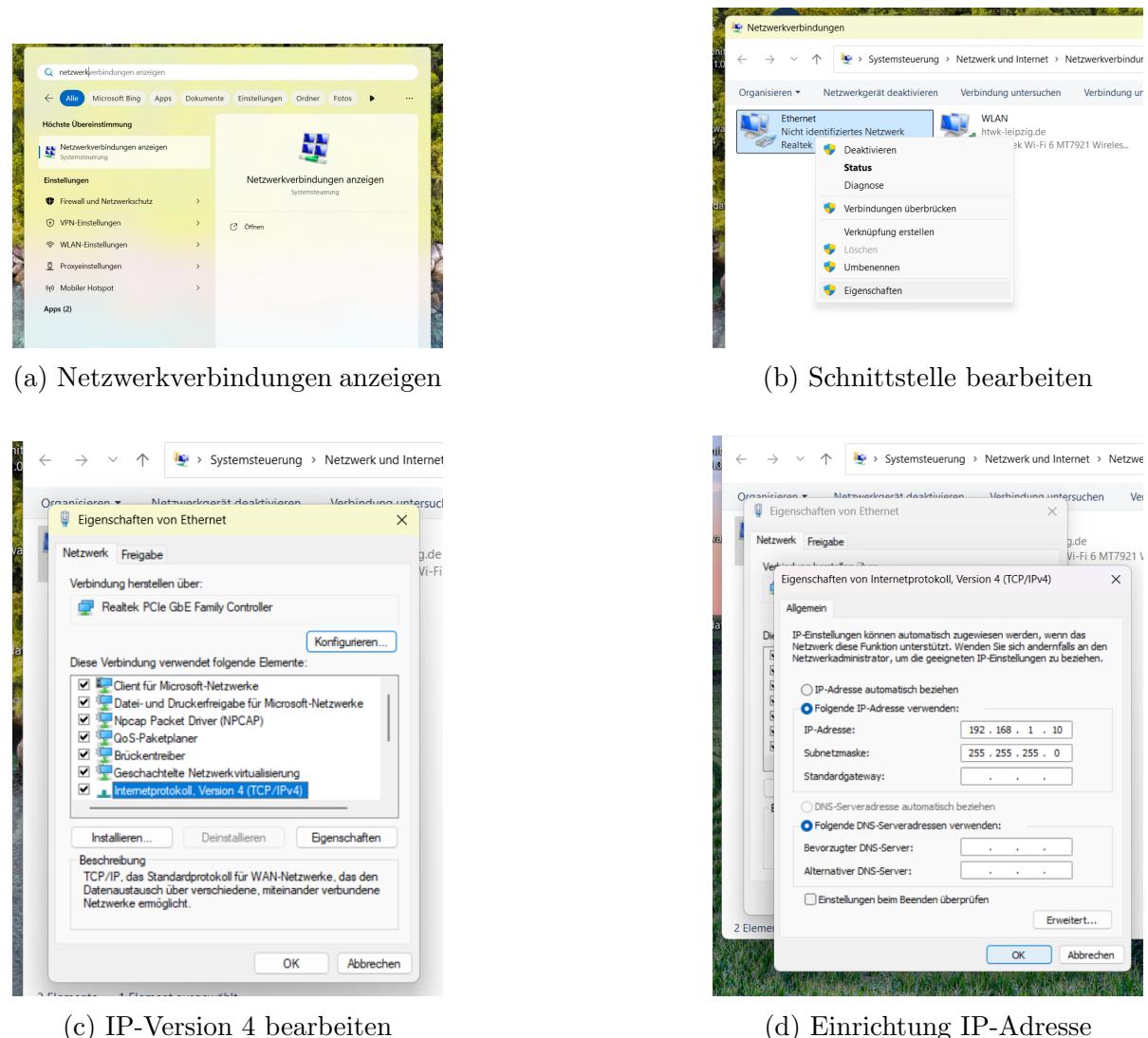


Abbildung 5: Einrichtung der PC-IP-Adresse

Jetzt sind Master und PC im selben Netzwerk und können miteinander kommunizieren, sodass der nächste Schritt erfolgen kann:

4.2 Webinterface via Browserzugriff

Die weitere Konfiguration erfolgt ab jetzt über den Browser:

1. IP-Adresse des Masters (*192.168.1.1*) im Browser eingeben
2. Authentifizierung mit Initialpasswort *0000*
3. Wechsel in den Projektierungsmodus zum Anlernen neuer Teilnehmer über den Reiter *Gateway*
4. Bearbeiten des Netzwerks im Reiter des entsprechenden ASi-Circuits

(a) Webinterface aufrufen

(b) Authentifizierung

The screenshot shows the Bihl+Wiedemann web interface. At the top, there's a navigation bar with links for Home, Gateway, ASi Circuit 1, ASi Circuit 2, Safety (with a red exclamation mark), Safe Link, Certificates, and Diagnostics. The main content area has three sections: 'Spontaneous diagnostics' (showing ASI Circuit 1 OK, ASI Circuit 2 OK, Safety Not OK, and No data exchange with the controller), 'Gateway information | protected mode' (listing device name, article number, identifier, electronic serial number, software package ID, webserver version, monitor version, and mode), and a 'Warning' box stating 'Outputs may be set and PLC may lose control'. Below these are buttons for 'Save & protected mode' and 'Save & configuration mode'. A note at the bottom says 'ASI-5 node reset' and 'Note: All selected ASI-5 nodes will be reset and (optionally) the stored configuration will be deleted.'

(a) Modus wechseln

This screenshot shows the ASi network overview. It features a 'Spontaneous diagnostics' section with status for ASI Circuit 1 (OK), ASI Circuit 2 (OK), Safety (Not OK), and No data exchange with the controller. Below this is the 'ASi-5' section, which displays four nodes (001 to 004) each with an ASi logo and a green checkmark under 'ASI-5'. The 'ASi-3' section shows a table with 16 columns labeled 0 through 15, all of which are currently empty.

(b) Übersicht ASi-Netzwerk

Abbildung 7: Einrichtung Webinterface

4.3 Hinzufügen neuer Teilnehmer

Mit dem eingerichteten Webinterface kann das ASi-Netzwerk nun um die Teilnehmer erweitert werden.

1. Im Projektierungsmodus ASi-Teilnehmer über ASi-Leitung anschließen und wenn nötig mit *Auxillary Power* versorgen
2. Eine gelbe 1 markiert einen nicht projektierten Teilnehmer im Netzwerk.
3. Den Teilnehmer dem Netzwerk hinzufügen. Die Adresse wird jetzt automatisch vergeben und ist konfliktsicher - eventuell erkannte doppelte Adressen werden gemeldet
4. Bestätigung durch Umschalten vom Projektierungsmodus in den geschützten Modus unter *Gateway*

(a) Teilnehmer anschließen

(b) Hinzufügen des Teilnehmers

(c) Teilnehmer hinzugefügt

(d) Teilnehmer projektiert

Abbildung 8: Einrichtung Teilnehmer

5 Programmierung des Masters

Zur Programmierung des Masters muss die Entwicklungsumgebung (IDE) *Control III* der Firma Bihl+Wiedemann GmbH (B+W) verwendet werden. B+W stellt für die IDE ein Handbuch zur Verfügung, das hier[8] heruntergeladen werden kann. Die Erklärungen dieses Kapitels stützen sich auf die Inhalte des Handbuchs. An manchen Stellen sind die Erklärungen des Handbuchs ausführlicher, an anderen werden sie durch die Erfahrungen aus der praktischen Erprobung ergänzt.

Die IDE basiert auf Eclipse. Eine Erklärung der Benutzeroberfläche findet sich im Benutzerhandbuch. Ein Unterschied der aktuellen Benutzeroberfläche zum Handbuch sind die Beispielprojekte, die im Projektexplorer angezeigt werden. Die Projekte *EthernetIP*, *Profibus* und *Profinet* wurden durch *ASi-5-Gateway* und *Alternative Gateway* ersetzt. In der praktischen Erprobung wurde das Projekt *ASi-5-Gateway* dupliziert und als Grundgerüst für neue Projekte genutzt. Alternativ können auch neue Projekte erstellt und anschließend ein Template (Beispielprojekt) importiert werden (Handbuch s.20). Die Programmierung der gesamten Fertigungsstrecke muss innerhalb eines Projekts erfolgen.

Hinweis: Es empfiehlt sich, zu Testzwecken jeweils neue Projekte zur Erprobung isolierter Funktionen anzulegen und diese sukzessive ineinander zu integrieren.

Beim erstmaligen Programmieren des Masters muss darauf geachtet werden, dass in der Datei *settings.mak* die richtige IP-Adresse des Masters eingestellt ist. Außerdem muss einmalig das Konfigurationstool zum Freischalten des Masters gestartet und der zuvor generierte Code dort eingegeben werden (Handbuch s.16).

Der grundlegende Workflow zum Laden und Ausführen eines Projekts besteht aus folgenden Schritten:

1. Alle geänderten Dateien speichern
2. Build-Tools: Clean (entfernt alte Ausführungsdateien)
3. Build-Tools: Release (Projekt für Master kompilieren, neue Ausführungsdateien erstellen)
4. Konfigurationstools: Stop Control (Programmausführung im Master stoppen)
5. Konfigurationstools: Download + Start Control (Neues Programm laden und ausführen)

Die Programmierung erfolgt in C. Die Datei *main.c* enthält das Programmgrundgerüst, welches sich in einen Setup- und einen Loop-Teil aufteilt. Je nach Bedarf können eigene Bibliotheken entwickelt und hier eingebunden werden. Alle ASi-Komponenten der

Fertigungsstrecke sind ASi-5-kompatibel. Es werden daher auch nur die Bibliotheksfunktionen für ASi-5 verwendet und hier erklärt.

Das Grundprogramm arbeitet nach dem EVA-Prinzip. Übertragen auf ASi bedeutet das, dass zyklisch das IDI gelesen und das ODI geschrieben werden muss. Dafür sind folgende Programmelemente notwendig:

5.1 Variablen

- `unsigned char ctrl_flags`:
Enthält Bits, die bei externen Nutzereingaben zum Stopp des Programms führen.
- `unsigned char asi5_accodi[]`:
Legt bitweise fest, welche Ports der I/O-Module prinzipiell als Output genutzt werden können. Auch wenn alle Ports im Setup freigeschaltet werden, kann ein Port danach noch als Input genutzt werden. Da jedes BWU-8334 Modul 16 Ports besitzt, müssen für jedes Modul 2 Array-Elemente eingeplant werden.
- `unsigned char asi5_odi[]`:
Das Output-Data-Image legt bitweise fest, welche Output-Ports der I/O-Module geschaltet werden. Da jedes BWU-8334 Modul 16 Ports besitzt, müssen für jedes Modul 2 Array-Elemente eingeplant werden.
- `unsigned char asi5_idi[]`:
Im Input-Data-Image werden bitweise die Schaltpiegel der Input-Ports gespeichert. Da jedes BWU-8334 Modul 16 Ports besitzt, müssen für jedes Modul 2 Array-Elemente eingeplant werden.

5.2 Funktionen

- `cctrl_func.Asi5WriteCtrlAccODI()`:

Schreibt die Daten aus `asi5_accodi[]` an das jeweilige Modul.

Beispiel:

```
cctrl_func.Asi5WriteCtrlAccODI(0, &asi5_accodi[2], 2, 3);
```

V.r.n.l. (Von rechts nach links): Schreibt an Gerät mit Adresse 3, 2 Bytes Daten, angefangen bei AccOdi-Element Nummer 2, im ASi-Kreis 0.

Hinweis:

Auch wenn sich hier um die AccOdi-Daten für das zweite Modul handelt, hat es die

logische Adresse 3. Das liegt daran, dass jedes Modul 2 Byte Nutzdaten überträgt und deshalb 2 Adressen verbraucht.

- `cctrl_func.Asi5WriteASi5Odi()`:

Schreibt die Daten aus `asi5_odi[]` an das jeweilige Modul.

Beispiel:

```
cctrl_func.Asi5WriteASi5Odi(0, &asi5_odi[2], 2, 3);
```

V.r.n.l.: Schreibt an Gerät mit Adresse 3, 2 Bytes Daten, angefangen bei Odi-Element Nummer 2, im ASi-Kreis 0.

Hinweis:

Auch wenn es sich hier um die ODI-Daten für das zweite Modul handelt, hat es die logische Adresse 3. Das liegt daran, dass jedes Modul 2 Byte Nutzdaten überträgt und deshalb 2 Adressen verbraucht.

- `cctrl_func.Asi5ReadASi5Idi()`:

Liest die Daten in `asi5_idi[]` vom jeweiligen Modul ein.

Beispiel:

```
cctrl_func.Asi5ReadASi5Idi(0, &asi5_idi[2], 2, 3);
```

V.r.n.l.: Liest von Gerät mit Adresse 3, 2 Bytes Daten ein, angefangen bei Idi-Element Nummer 2, im ASi-Kreis 0.

Hinweis:

Auch wenn es sich hier um die IDI-Daten für das zweite Modul handelt, hat es die logische Adresse 3. Das liegt daran, dass jedes Modul 2 Byte Nutzdaten überträgt und deshalb 2 Adressen verbraucht.

Die einzelnen Bits des IDI und ODI können mit Bitshift-Operationen oder Bitmasken ausgelesen und gesetzt werden. Beispiel:

```
if (((asi5_idi[1]<<8)|asi5_idi[0]) & (0x01 << 7)),
```

prüft ob Port 9 am ersten Modul geschaltet ist. Synonym dazu wäre:

```
if (asi5_idi[1] & (0x01 << 0))
```

In den Beispielprojekten wird im Setup-Teil eine Timerfunktion initialisiert:

`cctrl_func.CCctrlInitTimer(1, timer_function)`, siehe Handbuch S.80. Dabei wird angegeben, wie oft die Funktion aufgerufen werden soll (in Millisekunden) und der Name der aufzurufenden Funktion. Die aufzurufende Timerfunktion ist im Setup-Teil definiert und kann angepasst werden. Die globale Timervariable `system_ticks`, die in den Beispielprojekten verwendet wird, kommt in der Hauptschleife zum Einsatz, um die oben erwähnte EVA-Programmroutine in definierten Zeitabständen abzuarbeiten.

Treten bei der Programmierung Fehlermeldungen auf, können diese im Handbuch ab Seite 98 nachgeschlagen werden.

6 Aufbau und Bedienung der Debug-Schnittstelle

Zur Fehlersuche im Code ist der Einsatz eines Debuggers unerlässlich. Leider erweist sich der in *Control III* integrierte Debugger als wenig intuitiv, und die vorhandene Dokumentation bietet bei auftretenden Problemen nur unzureichende Unterstützung. Da insbesondere bei größeren Projekten ein funktionales Diagnosetool notwendig ist, wurde im Rahmen dieser Arbeit eine alternative Debug-Möglichkeit realisiert. Hierfür wurde ein USB-zu-TTL-Wandler an den Master angeschlossen, um über die serielle Schnittstelle Laufzeitinformationen direkt auf einem PC – beispielsweise mittels *PuTTY* – ausgeben zu können. Im Folgenden wird der Aufbau dieser Schnittstelle beschrieben und deren Anwendung demonstriert.

Zunächst wurden ein geeigneter USB-TTL-Konverter[1] sowie ein passendes Datenkabel[2] beschafft. An die Pins **RX**, **GND** und **VCC** des Konverters wurden feine Drähte angelötet, welche anschließend mit den entsprechenden Pads (**TX**, **GND**, **VCC**) auf der untersten Platine des Masters verbunden wurden. Hierfür musste das Gerät vorsichtig geöffnet werden. Die Vorgehensweise orientierte sich an der Masterarbeit von Tobias Haft, aus der auch die folgende Abbildung stammt, welche die Lötpunkte im Detail zeigt:

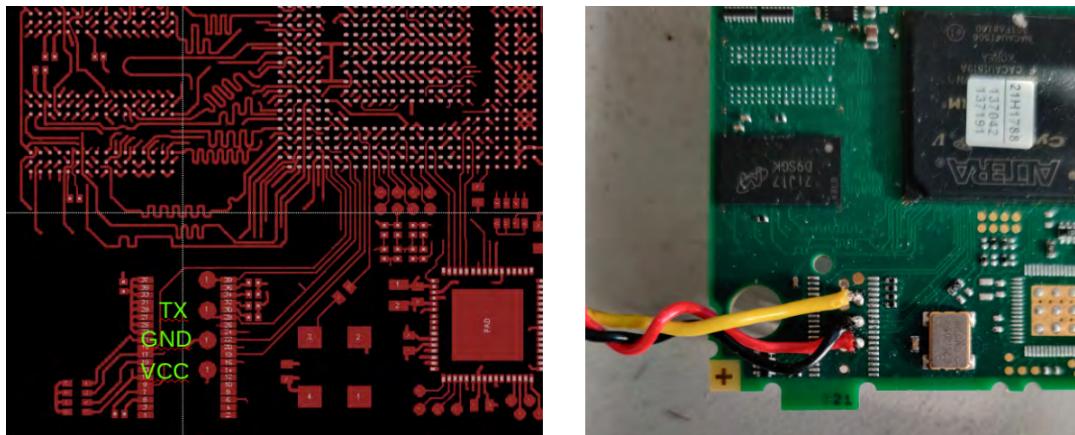


Abbildung 9: Anschluss des TTL-Konverters an die Platine des Masters

Die Drahtfarben wurden in unserem Projekt analog zur Abbildung verwendet: Gelb für die Datenleitung, Schwarz für Masse (GND) und Rot (bzw. Orange) für die Betriebsspannung (VCC). Es ist zwingend darauf zu achten, dass der Jumper des Konverters auf **3,3 V** und nicht auf 5 V gesetzt wird, um eine Beschädigung der Elektronik zu vermeiden.

Nach dem Zusammenbau und der erfolgreichen Inbetriebnahme des Masters kann an beliebiger Stelle im Programm der folgende Funktionsaufruf integriert werden:

```
cctrl_func.CCctrlPrintf("%d", &test);
```

Dabei ist zu beachten, dass der übergebene Wert zuvor in einer Variable gespeichert werden muss, da es sonst zu fehlerhaften oder nicht interpretierbaren Ausgaben kommt.

Sobald das Programm kompiliert und auf den Master übertragen wurde und der TTL-Konverter mit dem PC verbunden ist, kann die Ausgabe über ein Terminalprogramm wie *PuTTY* betrachtet werden. Dazu wird der entsprechende COM-Port (z. B. COM5) ausgewählt, die Baudrate auf 115200 eingestellt und die Verbindung geöffnet. Auf dem Hochschul-Laptop wurde diese Konfiguration dauerhaft gespeichert:

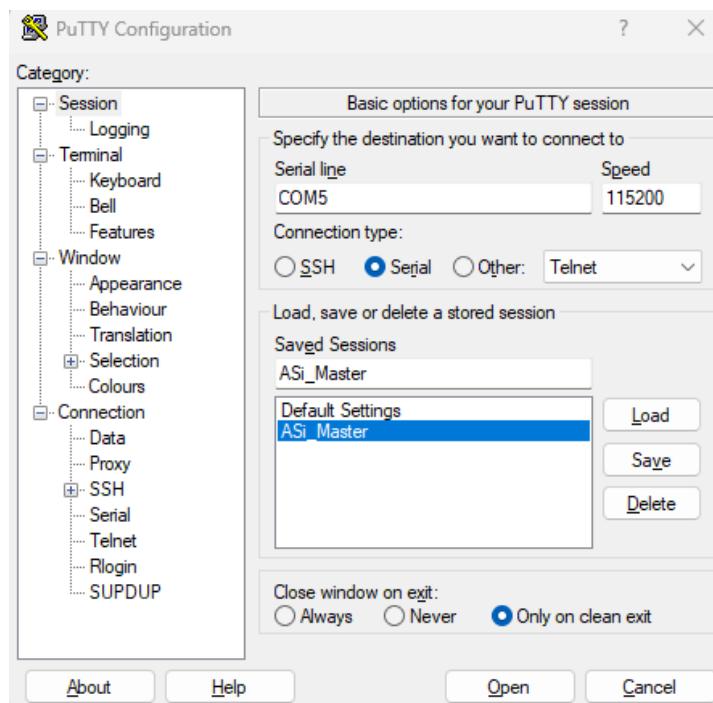


Abbildung 10: Konfiguration der seriellen Schnittstelle in PuTTY

Die Terminalausgabe ist nun sichtbar. Zum Test wurde in unserem Fall die Zahl 5 dauerhaft ausgegeben. Grundsätzlich kann jede beliebige Information ausgegeben werden, sofern die Formatierung der `printf`-ähnlichen Funktion korrekt erfolgt und dem C-Standard entspricht.

Die fertige Lösung ist hier zu sehen:

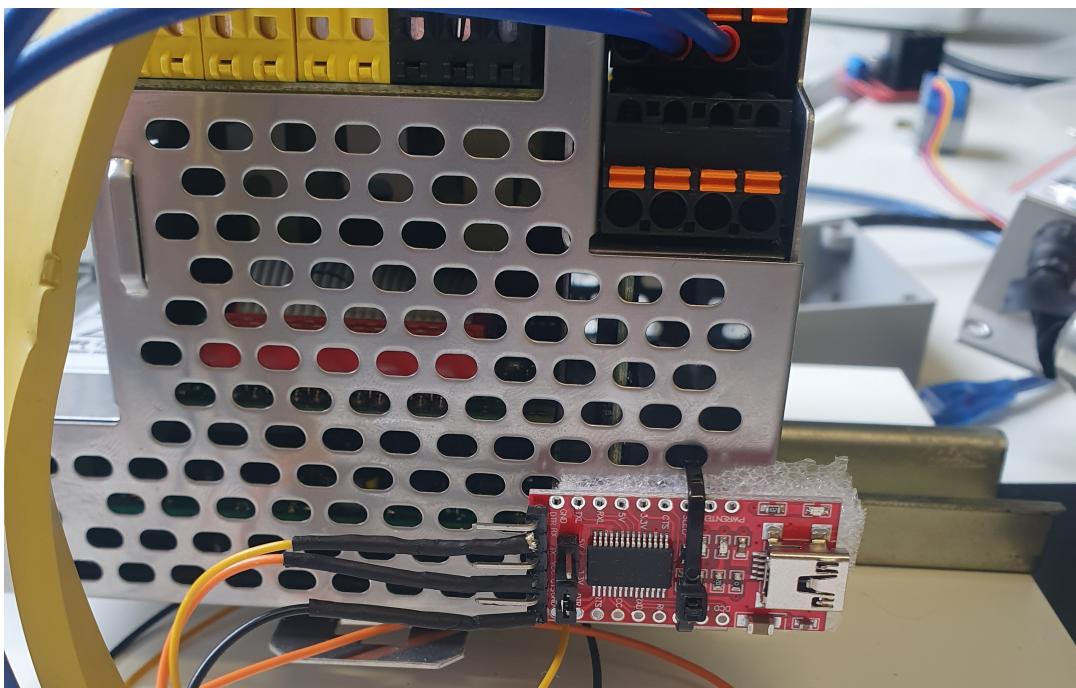


Abbildung 11: Fertiger Aufbau

7 Ausblick

Für zukünftige Praktikumsdurchgänge im Bereich ASi-5 ergeben sich verschiedene sinnvolle Ansätze zur Weiterentwicklung der bisherigen Übungen. Zunächst sollte angestrebt werden, mehrere Laptops mit der notwendigen Software auszustatten, sodass mehrere Gruppen parallel arbeiten können und Wartezeiten reduziert werden. Dies würde nicht nur den praktischen Ablauf verbessern, sondern auch die Effizienz der Durchführung insgesamt steigern.

Ein weiterer Ansatzpunkt besteht darin, aus den bisherigen grundlegenden Funktionen handhabbare, modulare Funktionsbausteine zu entwickeln. Diese könnten von den Studierenden einfacher genutzt und besser verstanden werden, da sie praxisnähere Anwendungen ermöglichen und die Komplexität reduzieren.

Darüber hinaus wäre eine Erweiterung des bestehenden Versuchsaufbaus um sicherheitsrelevante Aktorik und Sensorik denkbar. Hierdurch ließen sich wichtige Aspekte der funktionalen Sicherheit im Umfeld von ASi-5 anschaulich vermitteln und praktisch erproben.

Abschließend wäre es hilfreich, typische Fehlerbilder und deren Ursachen systematisch aufzubereiten und mit konkreten Hinweisen zur Behebung zu ergänzen. Dadurch könnten zukünftige Teilnehmer nicht nur schneller eigenständig Probleme erkennen und lösen, sondern erhielten zugleich ein tieferes Verständnis für die zugrunde liegende Technologie.

Abbildungsverzeichnis

1	Installationsdaten	10
2	Onlinefreigabe	11
4	Einrichtung der Master-IP-Adresse	14
5	Einrichtung der PC-IP-Adresse	15
7	Einrichtung Webinterface	17
8	Einrichtung Teilnehmer	18
9	Anschluss des TTL-Konverters an die Platine des Masters	23
10	Konfiguration der seriellen Schnittstelle in PuTTY	24
11	Fertiger Aufbau	25

Literatur

- [1] amazon. https://www.amazon.de/dp/B07Q7ZCL7L?ref=ppx_yo2ov_dt_b_fed_asin_title.
- [2] amzaon. https://www.amazon.de/dp/B078MBX55Q?ref=ppx_yo2ov_dt_b_fed_asin_title&th=1.
- [3] Bihl+Wiedemann GmbH. <https://www.bihl-wiedemann.de/de/produkte/zubehoer/diagnoseinbetriebnahme/produktuebersicht-diagnoseinbetriebnahme/l/BW2916>.
- [4] Bihl+Wiedemann GmbH. <https://www.bihl-wiedemann.de/de/produkte/software/produktuebersicht-asimon360>.
- [5] Bihl+Wiedemann GmbH. <https://www.bihl-wiedemann.de/de/produkte/software/diagnosesoftware>.
- [6] Bihl+Wiedemann GmbH. <https://www.bihl-wiedemann.de/en/products/software/diagnostics-software/t/BW2902>.
- [7] Bihl+Wiedemann GmbH. <https://www.bihl-wiedemann.de/en/support/downloads/lp-software>.
- [8] Bihl+Wiedemann GmbH. <https://www.bihl-wiedemann.de/de/produkte/zubehoer/diagnoseinbetriebnahme/produktuebersicht-diagnoseinbetriebnahme/l/BW2582>.
- [9] Java. <https://www.java.com/de/download/manual.jsp>.