

# Python Stellwerk

Stellwerk-Start ohne Arduino	2
Stellwerk-Start mit Arduino	2
Stellwerk-Start mit Anlage	3
Stromversorgung	4
Software	5
Externe Module	5

## Stellwerk-Start ohne Arduino

Für den Programmstart des Python-Stellwerks bietet das Betriebssystem des Raspberry (oder PC) mehrere Möglichkeiten bis hin zum Start mit einem Icon auf dem Desktop. Für den weniger erfahrenen User sollte der Weg über die Entwicklungsoberfläche gewählt werden.

Das Python-Stellwerk kann unmittelbar gestartet werden, ohne erkannte USB-Verbindung erhält man eine Fehlermeldung etwa in der Art

```
"could not open port /dev/ttyUSB0 ..."
```

Zunächst sind dann die im DiMo-Beitrag beschriebenen Schritte notwendig, damit Python die Schnittstelle akzeptiert.

Der Stellwerksbetrieb ist dann auch ohne angeschlossenen Anlagenbus möglich. In den nächsten Ausbaustufen des Stellwerks werden dann fehlende Verbindungen über die Software erkannt.

Der Funktionsumfang des Stellwerks ist für die erste Inbetriebnahme vorgesehen, es stellt noch nicht den gesamten Funktionsumfang dar. Einige Grafikelemente erhalten noch eine Feinkorrektur

Startet die Software ohne Fehlermeldung der Entwicklungsoberfläche, beginnt der Ablauf mit dem Lichtwechsel der Signale P2 und K2.

Ist kein Arduino vorhanden aber die USB-Schnittstelle bereit, dann ist nach einiger Zeit die Meldung „timeout“ in der untersten Zeile des Stellwerks zu sehen. Zusätzlich fehlen die Sequenznummern.

Meldungen und Zeiten werden mit dem Stellwerksausbau auf praktikable Werte angepasst.

## Stellwerk-Start mit Arduino

Der Arduino ist mit zwei verschiedenen Aufsetzplatinen zu betreiben, dies sind die Platinen für das Touchscreen-Stellwerk und die Platinen für das Fahrpult. Eine weitere Variante ist der Eigenbau des XBus-Interfaces oder des LocoNet-Interfaces.

Die Software muss die Unterschiede zwischen XBus, LocoNet und der Aufsetzplatinen berücksichtigen:

Platine	Software	XNetSRPin	LN_TX_PIN
Touchscreen Platine XBus	STWTXxx	5	-
Touchscreen Platine LocoNet	STWTLxx	-	6
Fahrpultplatine XBus	STWPXxx	13	-
Fahrpultplatine LocoNet	STWPLxx	-	2

xx: Versionsnummer

Für den XBus ist die Software in \*.hex Form für beide Platinenversionen im Download enthalten. LocoNet-Betreiber können die Versionen unter der angegebenen E-Mail anfordern oder den ebenfalls enthaltenen Quellcode anpassen.

Der Arduino gibt über seine USB-Schnittstelle eine Startmeldung aus. Mit Arduino-IDE ist diese mit dem Werkzeug *Serieller Monitor* sichtbar.

```
STWTX01  
XWAIT:1100
```

Ohne Arduino-IDE ist die Meldung mit einem Terminal-Programm, z.B. TeraTerm ebenfalls sichtbar. Die Baudrate ist auf 9600 Bd einzustellen.

Die vom Raspberry oder PC kommenden Schaltkommandos und ihre Weiterleitung an die Anlage sind ebenfalls sichtbar.

## Stellwerk-Start mit Anlage

Es empfiehlt sich, zunächst die Stromversorgung der verschiedenen Komponenten sicherzustellen. Im Abschnitt Stromversorgung sind die Details benannt.

Das Schalten der Weichen, Signale und der Schaltdecoder erfolgt über Adressen, die in der CSV-Datei enthalten sind. Die CSV-Datei beschreibt das Stellwerk vollständig. Die CSV-Zeilen sind im Dokument `Element Mapxx.pdf` beschrieben. Einige Eigenschaften sind noch nicht implementiert und andere nicht mehr relevant, weil Raspberry oder PC erweiterte Anzeigemöglichkeiten gegenüber dem Touchscreen bieten.

Für einen Test der eigenen Anlage ist mindestens eine passende Adresse einer Weiche und eines Signals erforderlich.

Die Adressen werden auch sichtbar, wenn mit der Maus ein entsprechendes Element im Gleisplan angeklickt wird.

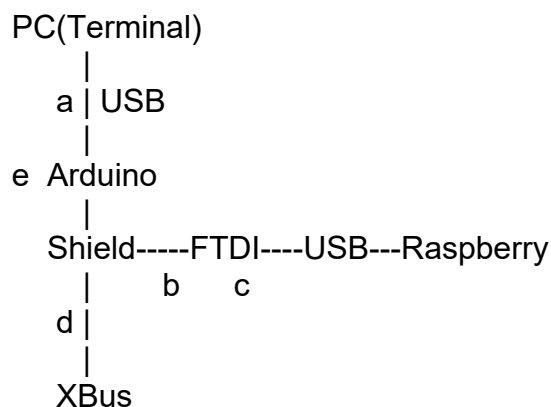
Für die DKW, für die Doppelweichen und Hauptsignale sind ggf. zwei Adressen vorhanden. Die Auswahl erfolgt dann über die blauen bzw. die roten Felder.

Geänderte CSV-Dateien sollten mit einem anlagenspezifischen Namen gespeichert werden. Dieser neue Name ist in der Python-Datei `Variables8.py` bei `CSV_FILE = ...` einzutragen.

## Stromversorgung

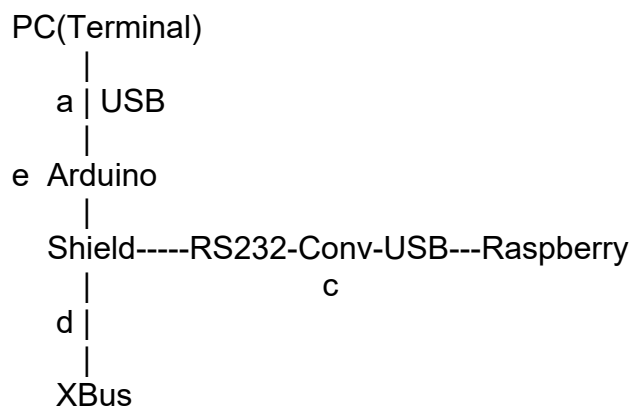
Bis zu vier 5V-Versorgungen sind an der Spannungsversorgung beteiligt.  
Nur eine Spannung sollte den Arduino samt Shield versorgen.  
Gefährdungen treten bei parallelen 5V-Verbindungen normalerweise nicht auf.  
Über die gemeinsame Masse Verbindungen können aber Querströme auftreten,  
welche die Datensignale beeinflussen können.

Für die Verbindung a zum PC sollte die 5V-Leitung des USB-Kabels unterbrochen sein. Das Flashen des Arduinos sollte mit einem normalen USB-Kabel ohne XBus und Raspberry-Verbindung erfolgen.



- a: Kann gesteckt sein, +5V unterbrechen
- b: FTDI kann Shield und Arduino mitversorgen
- c: Rasp.USB versorgt immer FTDI
- d: Shield mit Spannungswandler, 5V-Versorgung nur für Shield und Arduino
- e: Arduino und Shield mit eigenem Netzteil versorgt

Bei der Verwendung eines RS232-USB-Converters ist der Arduino über e oder d zu versorgen. Das Shield muss die RS232-Signale auf TTL-Pegel umsetzen.



Statt „Raspberry“ kann auch der das Gleisbild anzeigende PC über USB den FTDI-Adapter samt Arduino versorgen.

## Software

Die hier beschriebene Software ist Teil des Projektes „Das Python Stellwerk“ veröffentlicht unter dem Titel „Die Schlange im Stellwerk“ der Zeitschrift Digitale Modellbahn (DiMo), Ausgabe 1-2024.

Diese Software selbst und die darin enthaltenen Module ist und sind für den privaten Gebrauch frei (Open Source). Eine kommerzielle Verwendung bedarf der schriftlichen Zustimmung der jeweiligen Autoren.

Die Entwicklung der Python-Programme erfolgte auf Windows 10 PC, auf Raspberry Pi 3B+ und auf Raspberry 400.

Die Entwicklung und Übersetzung der Programme für den Arduino erfolgten auf einem Windows 10 PC mit der Arduino- IDE Version 1.8.19.

## Externe Module

### XBus

Externes Modul für die XBus-Übersetzung

Modulname	Version	Download-Website / Info
XpressNet.h	3.1.1	<a href="http://pgahtow.de/wiki/index.php?title=XpressNet">http://pgahtow.de/wiki/index.php?title=XpressNet</a> Stand: 29.09.2022

Diese XpressNet Version ist im Quellcode-Verzeichnis enthalten und wird mit übersetzt.

Bei der Übersetzung für den XBus ist im Arduino Quellcode die Zeile define-Zeile „`// #define LOCONET`“ als Kommentar (`//`) markiert und inaktiv.

Die zum Zeitpunkt der Veröffentlichung in der DiMo downloadbare XpressNet-Version 4.1.0 (<https://github.com/Digital-MoBa/XpressNetMaster>) wurde noch nicht geprüft.

Für die XBus-Entwicklung verwendete Zentrale:

- Roco z21 Firmware- Stand V1.42

### LocoNet

Externes Modul für die LocoNet-Übersetzung

Datei	Version	Download-Website / Info
LocoNet-master.zip	1.1.13	<a href="https://mrrwa.org/loconet-interface/">https://mrrwa.org/loconet-interface/</a>

Die LocoNet-Bibliothek hat offiziellen Arduino-Status und ist somit über die Arduino-IDE direkt installierbar.

Bei der Übersetzung für LocoNet ist im Arduino Quellcode die Zeile „`#define LOCONET`“ aktiv.

Für die LocoNet-Entwicklung verwendete Zentrale:

- Uhlenbrock-Zentrale Intellibox II (Artikelnummer 65100) mit der Firmware-Version 1.025.

-/-