

StartIDE

v0.5

Schnelleinstieg

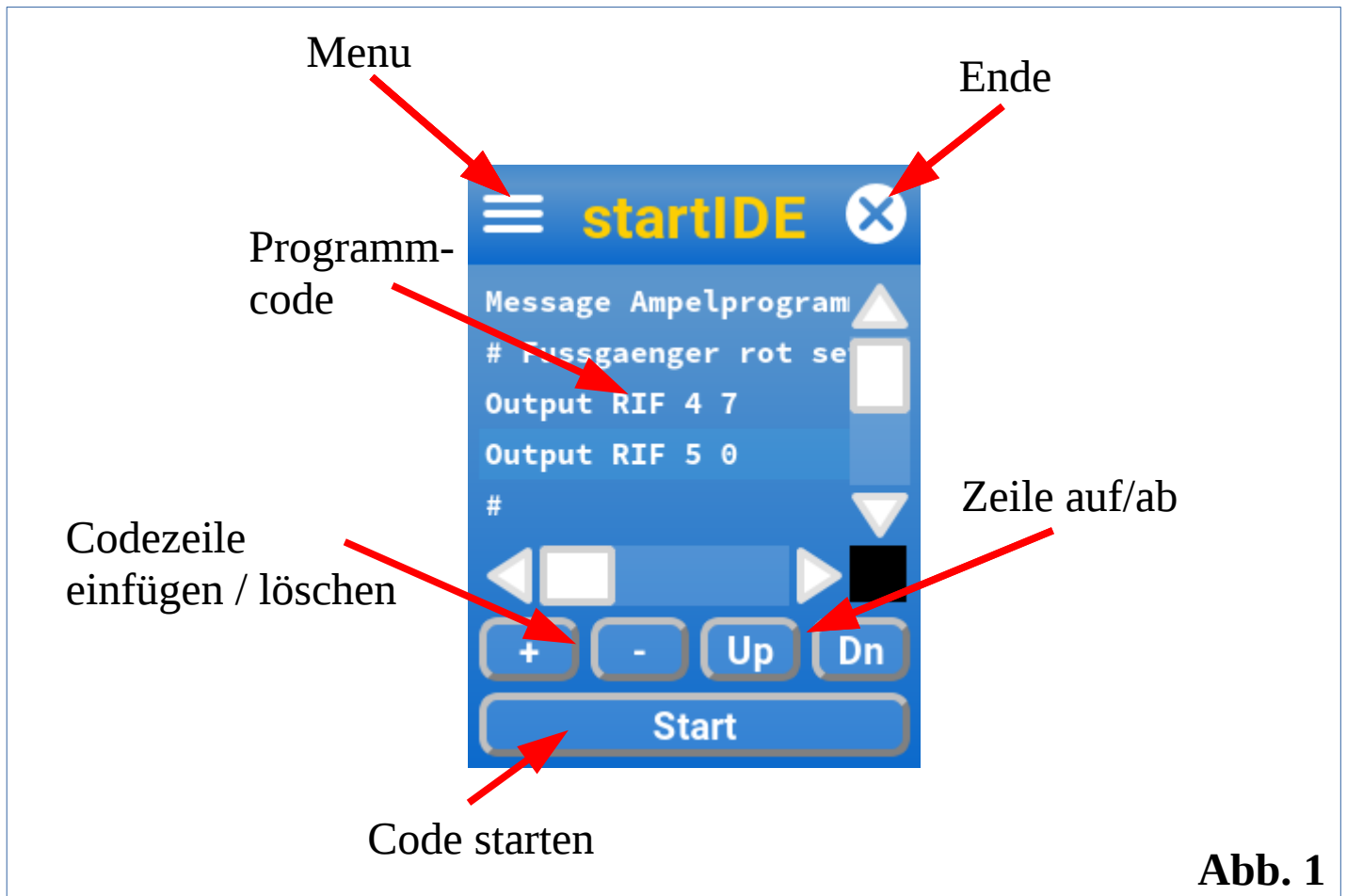


Abb. 1

startIDE ist eine Programmier-App für die community firmware des TXT Controllers, mit der sich eine Vielzahl einfacher Modelle programmieren lassen.

Inhalt:

1. Der Hauptbildschirm
2. Das Menu
3. Funktionsübersicht
 - 3.1. Inputs
 - 3.2. Outputs
 - 3.3. Controls
 - 3.4. Modules
 - 3.4. Interaction
4. Beispiele
5. Tips und Tricks

1. Der Hauptbildschirm

Die wesentlichen Elemente des Hauptbildschirms sind in **Abb. 1** gekennzeichnet.

Oben links befindet sich der „**Menu**“-Knopf, dessen Funktionen in Kap. 2 erläutert werden.

Mit dem Kreuz-Knopf („**Ende**“) wird startIDE beendet. Der aktuelle Programmcode geht nicht verloren und steht beim nächsten Aufruf von startIDE wieder zur Verfügung.

Mit dem **Plus**- und **Minus**-Knopf wird eine neue Programmzeile eingefügt bzw. die aktuelle Zeile gelöscht. Zum Löschen ist ein schnelles zweimaliges Antippen des Minus-Knopfs erforderlich.

Mit dem **Up**- bzw. **Down**-Knopf wird die aktuelle Zeile nach oben bzw. unten verschoben.

Der „**Start**“-Knopf schließlich startet die Ausführung des Programmes.

2. Das Menu

Das Menu bietet die Optionen



Unter „**Project**“ ist es möglich,

- mit „**New**“ ein neues Projekt anzulegen. Dabei wird der Programmspeicher gelöscht.
- mit „**Load**“ ein vorher gespeichertes Projekt wieder zu laden
- mit „**Save**“ das aktuelle Projekt abzuspeichern
- mit „**Delete**“ ein abgespeichertes Projekt dauerhaft zu löschen

Der Speicherort für die Projekte ist dabei ein Unterverzeichnis des startIDE-Verzeichnisses auf der SD-Karte. In **Kap. 5** wird u.a. erläutert, wie man von außen (PC) auf diese Daten zugreifen kann.

Mit den unter „**Modules**“ angebotenen Optionen können

- mit „**Import**“ Programmmodule von SD-Karte zum aktuellen Code dazugeladen werden
- mit „**Export**“ Programmmodule aus dem aktuellen Programm auf SD-Karte exportiert werden
- mit „**Delete**“ Module von SD-Karte dauerhaft gelöscht werden

Zur Erklärung, was ein Modul im Sinne von startIDE ist, siehe **Kap. 3**.

Der Menüpunkt „**Interfaces**“ öffnet eine Benachrichtigung, die anzeigt, welche Hardware-Interfaces (TXT und/oder Robo Interface Familie) aktuell gefunden wurden.

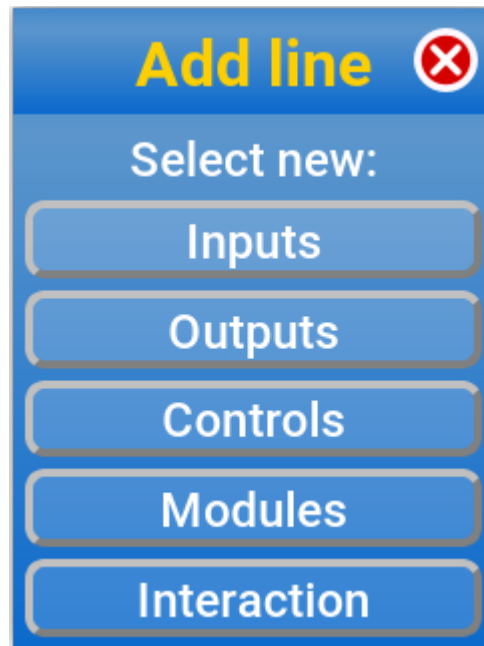
Wurde ein externes Interface nach dem Start von startIDE angeschlossen, so muß dieser Menüpunkt aufgerufen werden, um das Interface in startIDE nutzen zu können.

startIDE kann neben der TXT-Hardware das Robo Interface, die Robo I/O Extension, das Robo LT Interface und das Robo RF Funkmodul ansprechen. Es wird jedoch nur das unter „Interfaces“ angezeigte Gerät verwendet, auch wenn mehrere Interfaces gleichzeitig angeschlossen sind.

Auf dem TXT ist es jedoch möglich, in einem Programm gleichzeitig die TXT-Hardware **und** ein weiteres, per USB an den TXT angeschlossenes Interface zu nutzen.

3. Funktionsübersicht

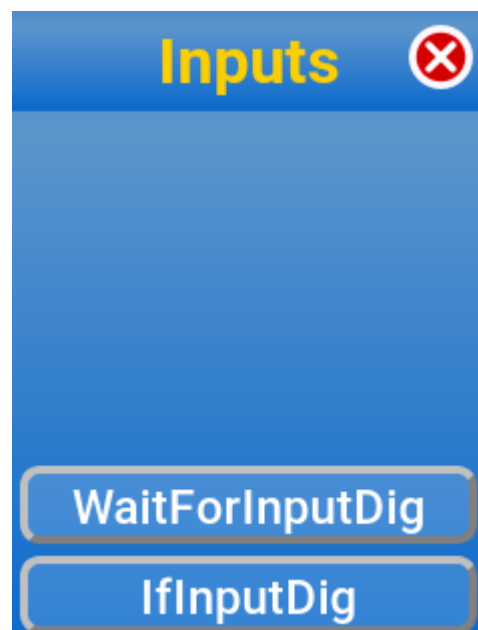
Über den „+“-Knopf auf dem Hauptbildschirm können Funktionen aus den folgenden Gruppen in den Programmcode eingefügt werden:



Mit dem Kreuz-Knopf oben rechts kann die Auswahl abgebrochen werden, ohne eine neue Code-Zeile einzufügen.

3.1. Inputs

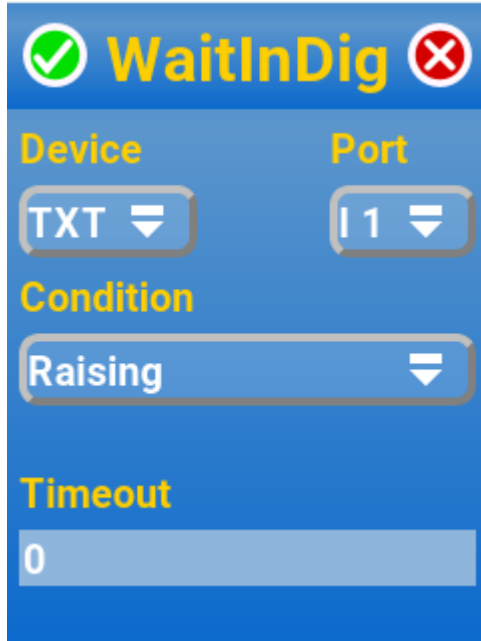
Hier sind die Funktionen



verfügbar.

3.1.1. WaitForInputDig

Der Befehl „WaitForInputDig“ wartet auf eine Änderung des Signals an einem Digitaleingang.



Es gibt folgende Optionen:

Device: TXT oder RIF
Es wird erst beim Programmstart überprüft, ob das gewählte Interface tatsächlich vorhanden ist.

Port: Die Nummer des Anschlusses

Condition: Soll auf eine steigende („Raising“) oder fallende („Falling“) Signalflanke gewartet werden?

Steigende Flanke bedeutet, daß der Kontakt geschlossen wird, fallend dementsprechend ein Öffnen des Kontaktes.

Timeout: maximale Wartezeit in ms, bis auch ohne Signaländerung fortgefahren wird. 0 steht dabei für „unendlich lange“ Wartezeit.

Im Code-Abschnitt des Hauptbildschirms ist die Syntax

```
WaitInDig <Device> <Port> <Condition> <Timeout>
```

also z.B.

```
WaitInDig RIF 1 Raising 500
```

Wartet max. 500ms darauf, daß der Kontakt an RoboInterface I1 geschlossen wird. Nach Schließen des Kontakts oder 500ms wird mit dem Programmablauf fortgefahren.

3.1.2. IfInDig

Der Befehl „IfInDig“ überprüft den Zustand eines Digitaleingangs und springt mit der Programmausführung abhängig vom Ergebnis ggf. zu einer Sprungmarke.

Vor dem Einfügen des IfInDig-Befehls muß mindestens eine Sprungmarke definiert sein.

Zur Definition von Sprungmarken siehe Kap. 3.3., Befehl „Tag“

Device: TXT oder RIF
Es wird erst beim Programmstart überprüft, ob das gewählte Interface tatsächlich vorhanden ist.

Port: Die Nummer des Anschlusses

Condition: „True“ (Wahr, Eingang ist aktiv / Kontakt geschlossen) oder „False“ (Falsch, Eingang ist deaktiv / Kontakt offen)

Jump Tag: Die Sprungmarke, die angesprungen werden soll

Im Codeabschnitt des Hauptbildschirms ist die Syntas

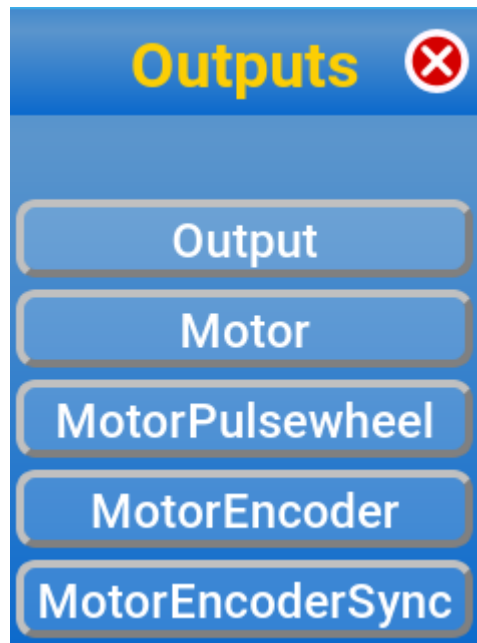
```
IfInDig <Device> <Port> <Condition> <Jump Tag>
```

also z.B.

```
IfInDig TXT 2 False ende
```

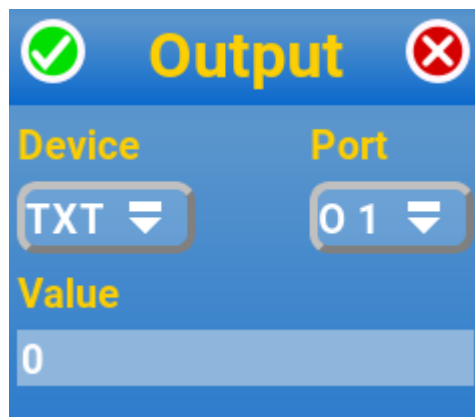
Springt zur Marke „ende“, wenn I2 am TXT Falsch (also Kontakt geöffnet) ist. Andernfalls wird mit der direkt folgenden Programmzeile fortgefahren.

3.2. Outputs



3.2.1. Output

Der Befehl „Output“ dient zum Schalten eines einzelnen Ausganges.



Device: TXT oder RIF

Es wird erst beim Programmstart überprüft, ob das gewählte Interface tatsächlich vorhanden ist.

Port:

Die Nummer des Anschlusses

Value: Einstellender Wert, zwischen 0 (= Aus) und 7 am RIF bzw. 0 und 512 am TXT.

Im Codeabschnitt des Hauptfensters ist die Syntax

```
Output <Device> <Port> <Value>
```

also z.B.

```
Output TXT 1 255
```

Setzt den Ausgang O1 am TXT auf 255, also ungefähr halbe Leistung (255/512).

3.2.2. Motor

Der „Motor“ Befehl dient zum Ansteuern eines Motorausgangs.

Device: TXT oder RIF

Es wird erst beim Programmstart überprüft, ob das gewählte Interface tatsächlich vorhanden ist.

Port:

Die Nummer des Anschlusses

Value: Einzustellender Wert, zwischen 0 (= Aus) und 7 am RIF bzw. 0 und 512 am TXT.

Direction: Drehrichtung „right“, „left“ oder „stop“. Dabei ist zu beachten, daß die tatsächliche Drehrichtung von der Polung des Motors abhängt.

Im Codeabschnitt des Hauptfensters ist die Syntax

```
Motor <Device> <Port> <Value> <Direction>
```

also z.B.

```
Motor TXT 1 512 r
```

Setzt den Ausgang M1 am TXT auf 512, also volle Leistung und Drehrichtung „rechts“.

3.2.3. MotorPulsewheel

Zur Ansteuerung eines Motors, der über einen gekoppelten Impulsgeber (üblicherweise Impulszahnrad, das einen Taster betätigt) und einen Endschalter überwacht wird. Damit läßt sich z.B. eine genaue Zielposition ansteuern.