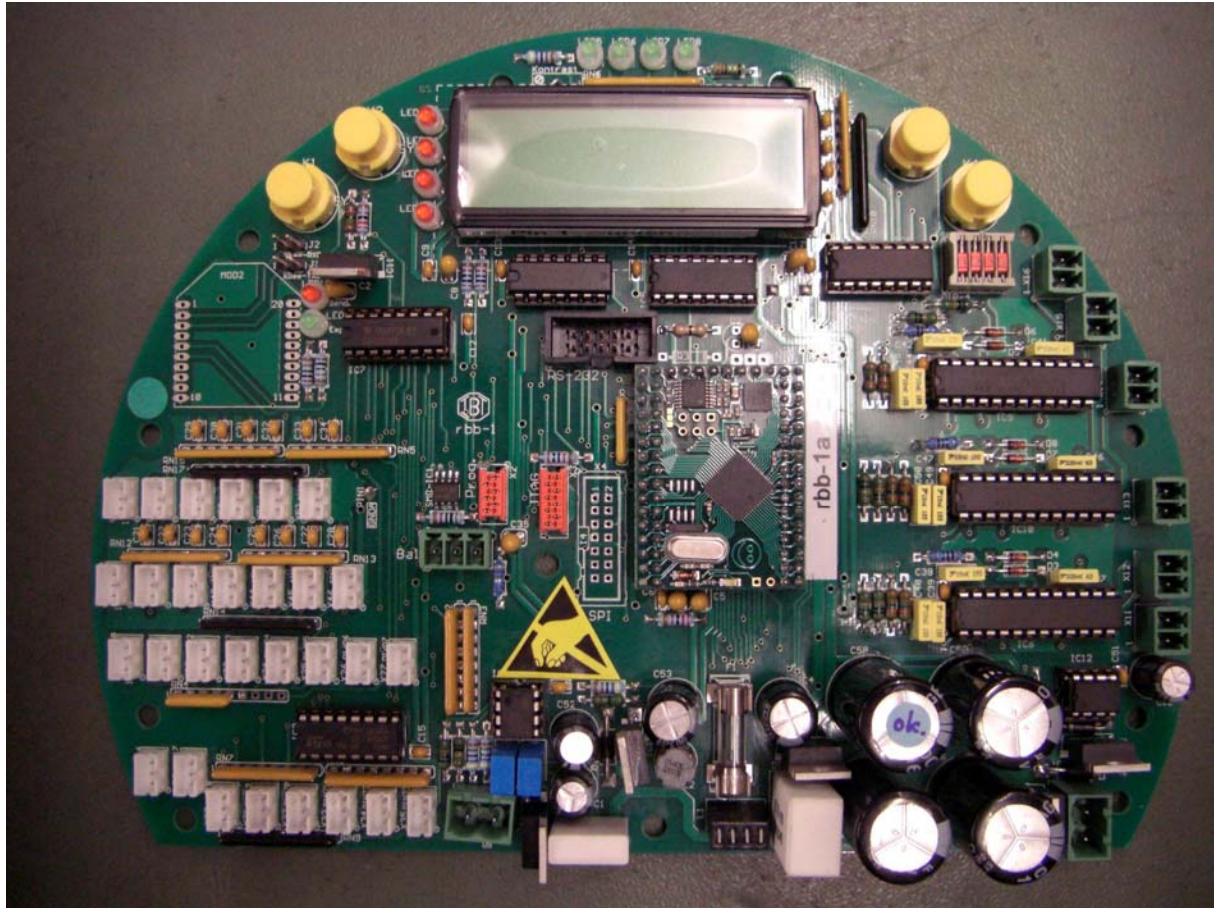


## Bedienungsanleitung

# rbb 1a

Robotic-Board



## IBT Universal-Robotic-Board rbb-1a

Industrie-Elektronik  
Mess- und Prüftechnik

IBT - Electronic



## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b> .....	<b>3</b>
<b>2</b>	<b>Wichtige Sicherheits-Hinweise</b> .....	<b>4</b>
2.1	Statische Elektrizität (ESD) kann das <i>rbb-1a</i> Board zerstören.....	4
2.2	Wackelkontakte bei den Motoranschlüssen zerstören die Motortreiber .....	4
2.3	Nur den mitgelieferten Programmieradapter mit Original-Kabel verwenden .....	4
2.4	Baugruppen / Sensoren /Elektronik-Komponenten nur im spannungsfreien Zustand einbauen/anschiessen .....	4
<b>3</b>	<b>Wichtige Betriebs-Hinweise</b> .....	<b>4</b>
<b>4</b>	<b>Sonstige Hinweise</b> .....	<b>5</b>
<b>5</b>	<b>WinAVR Compiler</b> .....	<b>5</b>
<b>6</b>	<b>Programmieradapter</b> .....	<b>6</b>
6.1	Treiber .....	6
6.2	Dip-Schalter (Programmieradapter „mySmartUSB“).....	6
6.3	Anschluss an rbb-1a .....	6
<b>7</b>	<b>rbb-1a Betriebssystem</b> .....	<b>7</b>
7.1	Installation.....	7
7.2	Bibliotheksfunktionen .....	7
7.3	Programmierung.....	7
<b>8</b>	<b>Anschluss der Sensoren</b> .....	<b>8</b>
8.1	Analogeingänge.....	8
8.2	Digitaleingänge .....	8
8.3	Kompass-Sensor.....	8
8.4	Ultraschall-Abstandssensoren .....	8
<b>9</b>	<b>Schuss-Magnet-Schaltstufe</b> .....	<b>8</b>
<b>10</b>	<b>Anpassen des Makefiles</b> .....	<b>9</b>
10.1	COM-Port des Programmieradapters .....	9
10.2	Weitere Quellcode-Dateien einbinden .....	10

## rbb-1a Robotic-Board

**ACHTUNG** Beachten sie unbedingt die „**Sicherheitshinweise**“ und die „**Wichtigen Betriebs-Hinweise**“, ansonsten werden die elektronischen Baugruppen **beschädigt oder zerstört**

### 1 Einleitung

Das *rbb-1a* Robotic-Board ist eine elektronische Leiterplatte, mit der z.B. ein Fußballroboter aufgebaut werden kann.

- Mikroprozessormodul mit ATMEGA 128 (20 MHz)
- Anzeigemodul (LCD 4 x 20 Zeichen)
- 8 LED's
- 4 Taster und ein 4-fach Dip-Schalter
- 13 freie Analogeingänge (0 – 5V)
- 1 fest zugewiesener Analogeingang für die Gesamt-Stromaufnahme-Messung
- 1 fest zugewiesener Digitaleingang für den „Ball erkannt“-Sensor
- 1 fest zugewiesener Digitaleingang für den Magnetfeldsensor
- 4 fest zugewiesenen Digitaleingänge für Ultraschall-Abstandssensoren
- 3 freie Digital-Ein/-Ausgänge (TTL)
- 8 freie Digitalgänge (TTL)
- Leistungs-Schaltstufe für Schussmagnet (max. 16V/10A)
- 6 PWM-Motorstufen (max. 16V/0,5 ... 2 A) z. Zt. auf 0,5 A-Begrenzung eingestellt
- Verpolschutz für den Batterieanschluss (9 ... max. 16V)

Die grundlegenden Softwarefunktionen/Treiber werden in Form von C++ Bibliotheken zur Verfügung gestellt.

Steckertyp und-Steckerbelegung der Digital/Analogsignale sind identisch mit dem QFIX-Soccerboard.

Zur Spannungsversorgung kann ein Doppel-Lithium-Ionenakku (15V, z. Bsp. Farnell Bestell-Nr. 118-9014) verwendet werden.

## 2 Wichtige Sicherheits-Hinweise

### **Eine Nichtbeachtung führt zur Zerstörung der Baugruppen**

#### 2.1 Statische Elektrizität (ESD) kann das *rbb-1a* Board zerstören.

Entladen Sie sich immer, bevor sie das *rbb-1a* Board oder den Programmieradapter berühren. Berühren Sie dazu einen leitfähigen/metallischen geerdeten Gegenstand. Ideal wäre die Verwendung einer ESD-Schutzmatte.

Ersatzweise können alle elektronischen Teile (möglichst noch in der ESD-Schutzverpackung) auf eine Pappkarton-Unterlage gestellt werden. Dann kann durch Berühren des Pappkartons für ca. 5 Sekunden ein elektrischer Ladungsausgleich zwischen Elektronikteilen und Anwender stattfinden.

#### **Auch im eingebauten Zustand müssen ESD-Sicherheitsregeln beachtet werden:**

- niemals elektronische Bauteile des *rbb-1a*-Boards berühren
- bei Spielbeginn vor Bedienen der Tasten/Schalter muss sich jeder über den Entladungspunkt des *rbb-1a*-Boards (siehe Bild) entladen.

#### 2.2 Wackelkontakte bei den Motoranschlüssen zerstören die Motortreiber

- **Schrauben der Kabel am Motor-Steckverbinder und am Motor fest anziehen bzw. befestigen.** (bei lockeren Anschlüssen bzw. Wackelkontakt an den Motoranschlüssen entstehen Spannungsfunken, welche die Motortreiber zerstören.)

#### 2.3 Nur den mitgelieferten Programmieradapter mit Original-Kabel verwenden

Andere Programmieradapter oder Kabel dürfen nur nach Rücksprache mit dem *rbb-1a*-Hersteller verwendet werden, selbst dann, wenn der Programmierstecker zu passen scheint (die Belegung der Stecker könnte unterschiedlich sein, und das *rbb-1a* Board beschädigen). Der Stecker ist zwar mit dem QFIX-Socboard mechanisch identisch aber elektrisch anders belegt

#### 2.4 Baugruppen / Sensoren /Elektronik-Komponenten nur im spannungsfreien

##### Zustand einbauen/anschiessen

Dazu sollte der Batteriespannungsanschluss möglichst über einen geeigneten Schalter angeschlossen werden.

## 3 Wichtige Betriebs-Hinweise

- **Stellen Sie für den Programmieradapter die richtige COM-Schnittstelle ein**  
Einstellen der COM-Schnittstelle siehe Kapitel „Programmieradapter“ .
- Batterie nur über einen geeigneten Schalter anschliessen

---

## 4 Sonstige Hinweise

- Die Beschreibung der C++-Bibliotheksfunktionen finden sie in der Datei „**rbb1\_Programmfunktionen.txt**“

Diese Datei befindet sich im Beispielprojekt.

## 5 WinAVR Compiler

Die Bibliotheken sind für den Freeware C++ Compiler „WinAVR“ geschrieben. Getestet wurde mit der Version 4.3.0 (20080610).

Zur Installation von WinAVR führen Sie das Installationsprogramm „WinAVR-20080610-install.exe“ aus. Belassen sie die Auswahl der Komponenten in der Standardeinstellung.

Als Programmierumgebung wird das in WinAVR enthaltene „Programmers Notepad“ verwendet. Zum Aufruf der Programmierumgebung starten Sie „Programmers Notepad [WinAVR]“ im, bei der Installation angelegten, WinAVR-Eintrag unter „Programme“.

Konfigurieren der Funktionstasten:

Um das Compilieren und Programmieren schnell über Tastaturkürzel zu starten, wählen Sie im Programmers Notepad den Menüpunkt „Tools/Options“ und im dann erscheinenden Fenster den Eintrag „Tools“. Wählen Sie im Feld „Scheme“ den Eintrag „None – Global Tools“.

Klicken Sie in der Liste auf den Eintrag „[WinAVR] Make All“ und anschließend auf den Knopf „Edit“. Klicken sie auf den Wert im Feld „Shortcut“ und drücken sie anschließend die gewünschte Tastaturkürzel-Taste, z.B. eine Funktionstaste. Die Taste erscheint jetzt im Shortcut-Feld.

Klicken Sie auf „OK“ und wiederholen Sie den Vorgang für den Eintrag „[WinAVR] Program“.

## 6 Programmieradapter

Standardmäßig wird der Programmieradapter „mySmartUSB“ verwendet.

**Andere Programmieradapter dürfen nur nach Rücksprache verwendet werden**, siehe Wichtige Sicherheits-Hinweise .

### 6.1 Treiber

Die Treiber und eine Installationsanleitung im pdf-Format werden mitgeliefert.

Nach der Installation kann unter der Windows-Gerätemanager die Nummer des virtuellen Com-Ports, unter dem der Programmieradapter angesprochen wird, eingestellt werden (siehe pdf-Installationsanleitung).

**Hier sollte „COM4“ eingestellt werden.**

Sollte dies nicht möglich sein, muss die richtige COM-Schnittstelle im Makefile eingestellt werden, siehe Kapitel „Anpassen des Makefiles“.

### 6.2 Dip-Schalter (Programmieradapter „mySmartUSB“)

Die Standardeinstellung der Dip-Schalter auf dem Programmieradapter ist

- 1 off
- 2 off
- 3 off
- 4 off

### 6.3 Anschluss an rbb-1a

Der mySmartUSB-Programmieradapter darf **nur mit dem mitgelieferten Kabel** an das **rbb-1a** Board angeschlossen werden. Der Stecker ist zwar mit dem QFIX-Socboard mechanisch identisch aber elektrisch anders belegt.

## 7 rbb-1a Betriebssystem

Die grundlegenden Systemfunktionen werden in Form von Bibliotheken zur Verfügung gestellt. Dabei sind folgende Funktionen bereits fest zugewiesen:

<b>Digital-E/A</b>	<b>Funktion</b>	<b>Kommentar</b>	
DI/O 3	1x Magnetfeldsensor-Anschluss	für CMPS03	
DI/O 4 - 7	4x Ultraschall-Entfernungsmesser-Anschluss	für SRF05	

### 7.1 Installation

Zur Installation entpacken Sie die Archivdatei rbb1\_Vx\_x.zip in ein beliebiges Verzeichnis.

Starten Sie Programmers Notepad, wählen Sie den Menüeintrag „File/Open Project(s)“, navigieren Sie zu den eben entpackten Dateien und öffnen Sie „ibtRoboticBoard1.pnproj“.

Dieses Projekt enthält ein bereits lauffähiges Beispielprogramm.

Zum Testen wählen Sie „Tools/[WinAVR] Make All“. Das Programm sollte ohne Fehlermeldungen kompiliert werden. Sollte das Fenster mit den Compilermeldungen nicht sichtbar sei, wählen Sie „View/Output“ um es aufzurufen.

Um das Programm auf das rbb-1a Board zu schreiben wählen Sie „Tools“[WinAVR] Program“. Sollten Sie eine Fehlermeldung erhalten, prüfen Sie, ob der Programmieradapter auf die richtige COM-Schnittstelle eingestellt ist (siehe Kapitel „Programmieradapter“).

### 7.2 Bibliotheksfunktionen

Eine Beschreibung der Programmfunktionen, die vom Betriebssystem zur Verfügung gestellt werden finden Sie in der Textdatei „**rbb1\_Programmfunktionen.txt**“.

### 7.3 Programmierung

Zur Erstellung Ihres eigenen Programms verwenden sie am besten das Beispielprojekt als Grundlage. Diese Beispielprojekt ist bereits lauffähig.

- Laden Sie das Projekt in Programmers Notepad (File/Open Project(s))
- Kompilieren Sie das Projekt (Tools/[WinAVR] Make All)
- Programmieren sie das rbb-1a Board (Tools/[WinAVR] Program).

Die Datei mit dem Quellcode muss „main.cpp“ heißen. Wie man andere Dateien einbindet, siehe unten.

---

## **8 Anschluss der Sensoren**

### **8.1 Analogeingänge**

0 bis 5 Volt Analogsignale könne an den Steckern AIN0 bis AIN12 angeschlossen werden. AIN13 ist intern fest mit der Gesamtstrom-Messung verbunden.

### **8.2 Digitaleingänge**

0 Volt / 5 Volt Digitaleingänge können an den Steckern DIN16 bis DIN23 angeschlossen werden.

### **8.3 Kompass-Sensor**

Der Kompass-Sensor CMPS03 muss am Stecker DI/O3 angeschlossen werden

### **8.4 Ultraschall-Abstandssensoren**

Es können bis zu vier Ultraschall-Abstandssensoren SFR05 an den Steckern DI/O4 bis DI/O7 angeschlossen werden.

## **9 Schuss-Magnet-Schaltstufe**

Zur Begrenzung der Stromaufnahme werden die Speicher-Kondensatoren für den Schuss-Schalter über eine 150 OHM-Widerstand aufgeladen. Dadurch kann nur alle ca. 10 sec ein Schuss erfolgen. Damit ein Schuss nur bei wirklich anliegendem Ball erfolgt, muss das Signal des Digitaleinganges am Stecker X7/Klemme 3 (dreipoliger grüner Stecker RM 3,81mm) ausgewertet werden ( siehe (BallPresent()) –Funktion)



## 10 Anpassen des Makefiles

Das Makefile befindet sich in dem Verzeichnis, in das das Beispielprojekt entpackt wurde. Das Makefile ist eine Textdatei, die Vorschriften zur Erstellung des Programms und zum Programmieren des rbb-1a Boards enthält.

Das Makefile kann in Programmers Notepad geöffnet und verändert werden.

Achtung: Bei falschen Einstellungen kann das Projekt evtl. nicht mehr compiliert werden.

Nehmen Sie Änderungen am Makefile mit Bedacht vor. Machen Sie vor jeder Änderung eine Sicherheitskopie des bisherigen Makefiles, die Sie im Falle eines Fehlers wieder zurückkopieren können.

### 10.1 COM-Port des Programmieradapters

Die Einstellungen für den Programmieradapter befinden sich in den Abschnitten „program“ und „programfusebits“

Belassen sie, wenn möglich die vorhandene Einstellung auf COM4 und stellen Sie im Gerätemanager für den Programmieradapter COM4 ein (siehe pdf-Installationsanleitung des mySmartUSB-Programmieradapters).

Sollte dies nicht möglich sein, können sie im Makefile einen anderen COM-Port einstellen. um z.B. COM1 einzustellen suchen Sie die Einträge

*program:*

```
avrduke -p $(MCU) -c avr911 -P com4 -U flash:w:$(TARGET).hex
```

*programfusebits:*

```
avrduke -p $(MCU) -c avr910 -P com4 -U hfuse:w:"hfuse.hex":i -U lfuse:w:"lfuse.hex":i -U efuse:w:"efuse.hex":i
```

und ändern Sie „com4“ in „com1“.

## 10.2 Weitere Quellcode-Dateien einbinden

Um ein größeres Programm übersichtlicher zu gestalten kann es sinnvoll sein es in mehrere Quellcode-Dateien aufzuteilen.

Nehmen wir an, Sie wollen den Quellcode in einer Datei „Bewegung.cpp“ einbinden.

Um diese Datei zu compilieren, suchen sie im Makefile den Eintrag

*compile:*

```
@echo Compiling:  
avr-gcc -c -mmcu=$(MCU) -DF_CPU=$(F_CPU) $(CPPFLAGS) main.cpp -o main.o  
@echo
```

und ergänzen ihn um eine Compilierungsvorschrift für Bewegung.cpp

*compile:*

```
@echo Compiling:  
avr-gcc -c -mmcu=$(MCU) -DF_CPU=$(F_CPU) $(CPPFLAGS) main.cpp -o main.o  
avr-gcc -c -mmcu=$(MCU) -DF_CPU=$(F_CPU) $(CPPFLAGS) Bewegung.cpp -o Bewegung.o  
@echo
```

Anschließend muss dem Linker noch mitgeteilt werden, dass er die compilierte „Bewegung.cpp“ einbinden soll.

Suchen Sie dazu den Eintrag

*link:*

```
@echo Linking:  
avr-gcc -mmcu=$(MCU) -DF_CPU=$(F_CPU) $(LDFLAGS) main.o rbb1_main.o rbb1_lcd.o  
rbb1_dinout.o rbb1_motor.o rbb1_adc.o -o $(TARGET).elf  
@echo
```

und ergänzen ihn um „Bewegung.o“:

*link:*

```
@echo Linking:  
avr-gcc -mmcu=$(MCU) -DF_CPU=$(F_CPU) $(LDFLAGS) main.o rbb1_main.o rbb1_lcd.o  
rbb1_dinout.o rbb1_motor.o rbb1_adc.o Bewegung.o -o $(TARGET).elf  
@echo
```