

BitBiber ByteBuddyv1.0

Das ultimative Handbuch für den inklusiven Lernroboter



**Technik entdecken, Barrieren
überwinden.**

*Ein Roboter für ALLE - mit Braille-
Unterstützung und barrierefreiem
Design*

Inhalt

Revolution im Klassenzimmer	3
Die Innovation	3
Auszeichnungen & Besonderheiten	3
Mission Control - Der erste Start	4
Die Erstvorstellung	4
Die Superkräfte von ByteBuddy	5
RFID-Magie - Programmieren mit Karten	6
Terminal-Power - Serielle Befehle	7
System-Commands	7
Verkabelung – Anschlussplan	11
Verkabelungs-Checkliste für Profis	12
Troubleshooting wie die Profis	12
Barrierefreies Arbeiten mit ByteBuddy	13
Für gemeinsames Lernen	14
Challenges für die Klasse	15
Advanced Hacks & Mods	16
Lernziele & Kompetenzen	18
Support & Community	19
FAQ - Häufige Fragen	20
Schlusswort	21

Revolution im Klassenzimmer

ByteBuddy ist nicht nur ein weiterer Lernroboter. Er ist ein **vollständig barrierefreier Programmier-Lernroboter** mit integrierter Braille-Unterstützung! Entwickelt von BitBiber TechTalente e.V., ermöglicht er **ALLEN** Kindern und Jugendlichen - unabhängig von ihren Fähigkeiten - den Einstieg in die digitale Welt.

Die Innovation

- ♦ **Braille-Beschriftung** auf allen Tasten
- ♦ **Braille-Markierung** auf allen RFID-Karten
- ♦ **Akustisches Feedback** für jede Aktion
- ♦ **Taktile Programmierung** ohne Bildschirm
- ♦ **Inklusion** von Anfang an mitgedacht

Auszeichnungen & Besonderheiten

	 100% BARRIEREFREI	
	 Für Sehende & Nicht-Sehende	
	 Fördert Teamwork	
	 Computational Thinking	
	 Inklusiv by Design	

Mission Control - Der erste Start

1. USB-Kabel einstecken

↓

2. MELODIE ERTÖNT: C-E-G ♪♪

↓

3. "BitBiber ByteBuddyready for action!"

↓

4. BEREIT FÜR ABENTEUER!

Die Erstvorstellung

Lehrer*in: "Klasse, das ist ByteBuddy!"

Schüler*in (blind): *Tastet die Braille-Beschriftung* "Cool, ich kann alle Knöpfe lesen!"

Schüler*in (sehend): "Wow, was bedeuten die Punkte?"

→ **Perfekter Einstieg ins Thema Inklusion!**

Lieferumfang - Das Komplettpaket

Anzahl	Item	Besonderheit
1×	ByteBuddyRoboter	Mit Braille auf allen Tasten
17×	RFID-Karten	17 Funktionen, alle mit Braille
1×	USB-Kabel	Für Stromversorgung & Programmierung
1×	Dieses Handbuch	Inklusiv gestaltet
∞	Möglichkeiten	Unbegrenzt!

Die Superkräfte von ByteBuddy

Level 1: Bewegungs-Superkräfte

→ VORWÄRTS wie The Flash

← RÜCKWÄRTS wie ein Mondwalk

↶ LINKS wie ein Breakdancer

↷ RECHTS wie ein Eiskunstläufer

Level 2: Denk-Superkräfte

FOR-SCHLEIFEN = "Ich kann zählen und wiederholen!"

WHILE-SCHLEIFEN = "Ich höre nie auf (bis...)"

IF-BEDINGUNGEN = "Ich treffe Entscheidungen!"

VERSCHACHTELUNG = "Ich denke in mehreren Ebenen!"

Level 3: Sinnes-Superkräfte

 ULTRASCHALL = "Ich sehe Hindernisse wie eine Fledermaus!"

 PIEZO-TON = "Ich spreche in Melodien!"

 BRAILLE = "Ich kann mit Fingern gelesen werden!"

 RFID = "Ich verstehe magische Karten!"

Taster mit Braille - Barrierefreie Kontrolle

Taster	Braille	Funktion	Ton-Feedback	Geheimtipp
1	·	REC - Aufnahme	880 Hz (A5)	2× drücken = Stopp
2	:·	RUN - START	988 Hz (B5)	Startet das Abenteuer
3	∴	ACT_VOR - Vorwärts	1047 Hz (C6)	Der Weg nach vorn
4	∴·	ACT_ZURUECK - Zurück	740 Hz (F#5)	Strategischer Rückzug
5	∴	ACT_LINKS - Links	659 Hz (E5)	Gegen den Uhrzeigersinn
6	∴·	ACT_RECHTS - Rechts	831 Hz (G#5)	Im Uhrzeigersinn

Jeder Taster spielt einen anderen Ton - so können auch blinde Nutzer*innen sofort hören, welcher Knopf gedrückt wurde. Die Töne bilden zusammen einen **Akkord**!

Terminal-Power - Serielle Befehle

NEU: BEFEHLSKETTEN-TECHNOLOGIE!

Statt mühsam jeden Befehl einzeln einzutippen, könnt ihr jetzt **KETTEN** erstellen:

Für Anfänger (ohne Ketten):

REC

ACT_VOR

ACT_LINKS

ACT_VOR

ACT_RECHTS

REC

RUN

Für Profis:

REC, ACT_VOR, ACT_LINKS, ACT_VOR, ACT_RECHTS, REC, RUN

System-Commands

Befehl	Action	Coolness-Faktor
HELP	Zeigt alle Befehle	★ ★
REC	Aufnahme Start/Stop	★ ★ ★
RUN	Programm ausführen	★ ★ ★ ★
PROG DUMP	Zeigt Programm-Code	★ ★ ★ ★ ★
PROG CLEAR	Löscht alles	★ ★

One-Line-Wonder Beispiele

Das Quadrat in einer Zeile!

REC,SET_N_4,FOR_BEGIN,ACT_VOR,ACT_LINKS,END,REC,RUN

Die Spirale des Wahnsinns!

REC,SET_N_10,FOR_BEGIN,ACT_VOR,ACT_LINKS,ACT_VOR,END,REC,RUN

Der Ausweich-Algorithmus!

REC,WHILE_FRONT_FREE,ACT_VOR,END,ACT_LINKS,ACT_LINKS,REC,RUN

Level 1: Erste Schritte - Die Basics

Mission 1: Der erste Move

Aufgabe: Fahre 3 Felder vorwärts

Lösung: ACT_VOR, ACT_VOR, ACT_VOR

Mission 2: Das L

Aufgabe: Fahre ein L-Muster

Schema: ↑ ↑ → →

Lösung: ACT_VOR, ACT_VOR, ACT_RECHTS, ACT_VOR, ACT_VOR

Mission 3: Der Moonwalk

Aufgabe: Vor, vor, zurück, zurück - wie Michael Jackson!

Lösung: ACT_VOR, ACT_VOR, ACT_ZURUECK, ACT_ZURUECK

Level 2: Schleifen-Meister

Mission 4: Das perfekte Quadrat

Pseudocode

für 4 mal:

 fahre_vor()

 drehe_links()

ByteBuddy-Code

SET_N_4

FOR_BEGIN

 ACT_VOR

 ACT_LINKS

END

Mission 5: Die Endlos-Spirale

Verschachtelte Schleifen!

```
SET_N_4
FOR_BEGIN
  SET_N_3
  FOR_BEGIN
    ACT_VOR
  END
  ACT_LINKS
END
```

Level 3: Bedingungen & KI

Mission 7: Der Crash-Test-Dummy

```
# Fahre bis zur Wand
WHILE_FRONT_FREE
  ACT_VOR
END
ACT_ZURUECK
ACT_ZURUECK
```

Mission 8: Der Ausweicher

Intelligente Navigation

```
IF_FRONT_FREE
  ACT_VOR
  ACT_VOR
END
IF_FRONT_BLOCKED
  ACT_LINKS
  ACT_VOR
  ACT_RECHTS
  ACT_VOR
END
```

Mission 9: Der Labyrinth-Löser

Rechte-Hand-Regel

SET_N_20

FOR_BEGIN

ACT_RECHTS

IF_FRONT_FREE

ACT_VOR

END

IF_FRONT_BLOCKED

ACT_LINKS

END

END

Level 4: Profi-Challenges

Mission 10: Der Tanz-Battle

Programmiere eine Choreografie mit mindestens 20 Moves und 3 verschiedenen Schleifen!

Mission 11: Der Parkour

Baue einen Hindernisparcours und programmiere ByteBuddyso, dass er automatisch durchkommt!

Mission 12: Die KI-Challenge

Erstelle einen Algorithmus, der ByteBuddy aus JEDEM Labyrinth führt

Verkabelung – Anschlussplan

Komplette Pin-Übersicht

Komponente	Arduino Pin	Spannung	Funktion	Code-Definition
RFID SDA/SS	D8	5V tolerant	Chip Select	#define SS_PIN 8
RFID RST	D5	5V tolerant	Reset	#define RST_PIN 5
RFID SCK	D13	5V tolerant	SPI Clock	Automatisch
RFID MOSI	D11	5V tolerant	Daten → RFID	Automatisch
RFID MISO	D12	5V tolerant	Daten ← RFID	Automatisch
RFID 3.3V	3.3V	NUR 3.3V!	Stromversorgung	-
RFID GND	GND	-	Masse	-
RFID IRQ	-	-	Nicht benutzt	-
Servo Links	D9	5V	PWM Signal	-
Servo Rechts	D10	5V	PWM Signal	-
Summer	D3	5V	Töne	-
US-Trig	D7	5V	Trigger	-
US-Echo	D6	5V	Echo	-
Taster	A0	5V	Analog	-

Verkabelungs-Checkliste für Profis

RFID-Anschlüsse:

- ☐ SDA → D8 (Gelbes Kabel empfohlen)
- ☐ SCK → D13 (Grünes Kabel empfohlen)
- ☐ MOSI → D11 (Blaues Kabel empfohlen)
- ☐ MISO → D12 (Oranges Kabel empfohlen)
- ☐ IRQ → NICHT VERBINDEN!
- ☐ GND → GND (Schwarzes Kabel)
- ☐ RST → D5 (Weißes Kabel empfohlen)
- ☐ 3.3V → 3.3V (Rotes Kabel) ⚠️ WICHTIG! 5V beschädigen das Modul!

Andere Komponenten:

- ☐ Servo Links → D9
- ☐ Servo Rechts → D10
- ☐ Summer → D3
- ☐ Ultraschall Trigger → D7
- ☐ Ultraschall Echo → D6
- ☐ Taster-Array → A0

Troubleshooting wie die Profis

Problem: RFID-Karte wird nicht erkannt

Symptome:

- Kein Piepton
- Serial Monitor zeigt "Unbekannte UID"

Lösungen:

1. Check: 3.3V Stromversorgung? (NICHT 5V!)
2. Check: Alle SPI-Kabel richtig?
3. Check: Karte flach auflegen, 2 Sek warten
4. Check: Metallische Gegenstände entfernen
5. Pro-Tipp: UID notieren und im Code ergänzen

Für gemeinsames Lernen

Tandem-Programmierung:

- Sehender *Schülerin*: Liest Bildschirm-Feedback
- Blinder *Schülerin*: Bedient Taster und Karten
- **Beide lernen voneinander!**
- Sehende Schüler*innen: Zeichnen einen Plan
- Gemeinsam: Lösungsalgorithmus entwickeln
- ByteBuddy: Führt das Programm aus

Konzept 3: Die Choreografie-Designer (60 Min)

Aufgabe: Erstellt einen Roboter-Tanz zu Musik!

Teams: 3-4 Personen (gemischt)

- Choreograf*in: Plant die Moves
- Programmierer*in: Legt RFID-Karten
- Tester*in: Prüft mit Tasten
- DJ: Wählt passende Musik

Starter-Level

- Einfache Sequenzen (3-5 Befehle)
- Keine Schleifen
- Klare Wege

Medium-Level

- FOR-Schleifen verwenden
- 10-15 Befehle
- Einfache Muster

Expert-Level

- Verschachtelte Strukturen
- IF/WHILE-Bedingungen
- Komplexe Algorithmen

Challenges für die Klasse

Challenge 1: Das Rennen

Aufbau: 3 Meter Strecke mit Hindernissen Ziel: Schnellster Weg von A nach B Twist: Jedes Team darf nur 20 Befehle verwenden!

Challenge 2: Der Schatzsucher

Aufbau: 4×4 Feld mit verstecktem "Schatz" Ziel: Systematisch das Feld absuchen Bonus: Wenigste Befehle gewinnt!

Challenge 3: Die Zirkusnummer

Aufgabe: 1-Minuten-Show programmieren Bewertung:

- Kreativität (1-10 Punkte)
- Technische Komplexität (1-10 Punkte)
- Teamwork (1-10 Punkte)

Challenge 4: Der Code-Cracker

Lehrer*in programmiert geheimes Muster Teams müssen durch Beobachtung herausfinden:

- Welche Befehle wurden verwendet?
- In welcher Reihenfolge?
- Können sie es nachprogrammieren?

Technische Spezifikationen

| Max. Programmlänge | 128 Befehle | Reicht für komplexe Programme |

| Max. Verschachtelung | 16 Ebenen | Mehr als genug! |

| Stack-Größe | 16 Frames | Für verschachtelte Strukturen |

| Ultraschall-Range | 2-400 cm | Perfekt für Klassenzimmer |

| Servo-Auflösung | $\pm 250 \mu s$ | Präzise Steuerung |

| Speicher | 2KB RAM | Klein aber oho! |

Timing-Parameter

// Diese Werte können im Code angepasst werden:

```
const int ACTION_MS = 600;    // Bewegungsdauer
const int ENTPRELL_MS = 200;  // Taster-Entprellung
const int IDLE_DETACH_MS = 800; // Servo-Abschaltung
const int SICHERHEITS_ABSTAND_CM = 10; // Kollisionsvermeidung
```

Kommunikation

- **Serial:** 9600 Baud, 8N1
- **SPI:** Hardware-SPI für RFID
- **PWM:** Pins 9 & 10 für Servos
- **ADC:** A0 für Taster-Array

Advanced Hacks & Mods

Hack 1: Neue RFID-Karten hinzufügen

So fügt ihr neue Karten hinzu:

1. Unbekannte Karte auflegen
2. UID aus Serial Monitor kopieren (z.B. "A3 B4 C5 D6")
3. Im Code suchen: RfidMap rfidMap[]
4. Ersetzen:

```
{"XX XX XX 01", "ACT_VOR"}, // ALT
```

```
{"A3 B4 C5 D6", "ACT_VOR"}, // NEU
```

5. Upload!

Hack 2: Eigene Melodien

Start-Melodie ändern:

```
void setup() {
    //Star Wars Theme:
    tone(PIN_SUMMER, 392, 350); // G4
    delay(350);
}
```

```
tone(PIN_SUMMER, 392, 350); // G4
delay(350);
tone(PIN_SUMMER, 392, 350); // G4
delay(350);
tone(PIN_SUMMER, 311, 250); // Eb4
delay(325);
tone(PIN_SUMMER, 466, 100); // Bb4
delay(100);
tone(PIN_SUMMER, 392, 500); // G4
}
```

Hack 3: Turbo-Modus

// Für Speed-Freaks:

```
int GESCHWINDIGKEIT = 400; // Statt 250
```

```
const int ACTION_MS = 300; // Statt 600
```

Lernziele & Kompetenzen

Informatik-Kompetenzen

- ✓ Algorithmisches Denken
- ✓ Sequenzielle Programmierung
- ✓ Schleifen und Iteration
- ✓ Bedingte Anweisungen
- ✓ Fehlersuche (Debugging)
- ✓ Abstraktion

Soziale Kompetenzen

- ✓ Teamwork
- ✓ Inklusion leben
- ✓ Kommunikation
- ✓ Problemlösung
- ✓ Kreativität
- ✓ Empathie

21st Century Skills

- ✓ Computational Thinking
 - ✓ Digital Literacy
 - ✓ Critical Thinking
 - ✓ Collaboration
 - ✓ Creativity
 - ✓ Communication
-

Support & Community

Direkter Kontakt

BitBiber TechTalente e.V.

Schweidnitzer Str. 17

45891 Gelsenkirchen

Deutschland

 E-Mail: maik@bitbiber.de

 Web: www.bitbiber.de

FAQ - Häufige Fragen

F: Kann ByteBuddy auch ohne RFID verwendet werden?

A: Ja! Taster und serielle Befehle funktionieren immer.

F: Können eigene Sensoren hinzugefügt werden?

A: Ja! Freie Pins: D2, D4, A1-A5

F: Gibt es Updates?

A: Check unser Webseite oder kontaktiert uns direkt!

F: Kann ich den Code verändern?

A: Absolut! Open Source FTW! 🇩🇪 Gebt uns aber bitte Credits 😊

Schlusswort

Die Vision

ByteBuddy ist mehr als nur ein Roboter. Er ist ein **Symbol für Inklusion**, ein **Werkzeug für Bildung** und ein **Brückenbauer zwischen Welten**.

Wenn sehende und blinde Kinder gemeinsam programmieren, wenn Technik Barrieren überwindet statt sie zu schaffen, wenn ALLE Kinder die gleichen Chancen haben - dann haben wir unser Ziel erreicht.

Der Auftrag

Liebe Lehrer*innen,

Ihr habt jetzt die Macht, Kindern nicht nur Programmieren beizubringen, sondern auch:

- Dass Verschiedenheit eine Stärke ist
- Dass Technik für ALLE da ist
- Dass Barrieren überwunden werden können
- Dass gemeinsam mehr möglich ist

Nutzt diese Macht weise! 🚀

Das Versprechen

Jedes Kind, das mit ByteBuddy arbeitet, lernt mehr als nur Code. Es lernt, dass in einer inklusiven Welt ALLE einen Platz haben und dass Technik dabei helfen kann, diese Welt zu schaffen.

🤖 **Happy Coding!** 🤖

BitBiber ByteBuddyv1.0

Technik entdecken, Barrieren überwinden.

© 2024 BitBiber TechTalente e.V.

Made with ❤️ in Gelsenkirchen