

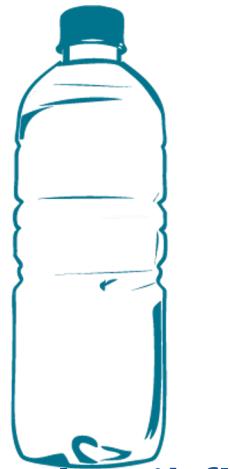


Mit MATRIX lernen, einen Roboter fernzusteuern (2)

Beispiel mit MATRIX Essential Set(MR0001)

Erstellt von Water Xu & MATRIX Robotics am 24.09.2025

Benötigte Materialien



Plastikflaschen, Kegelhütchen und Klopapierrollen können als Hindernisse genutzt werden.



MATRIX Mini Essential Set V2
* Stelle Sicher, dass du einen Joystick dabei hast.

**Was ist das Grundkonzept von Fernsteuerung?
Wie wird das Signal übertragen?**

Bekannte Prinzipien der Fernsteuerung

Infrarot

Infrarot ist ein für das menschliche Auge unsichtbares Licht, das Signale an Maschinen übertragen kann. Damit es funktioniert, muss man zumindest etwas in die Richtung des Empfängers zeigen, sonst kommt es nicht an.

Radio Wellen

Genau wie bei einem Radiosender, können Radiosignale auch durch Wände durchdringen und große Distanzen überbrücken. Häufig in Garagentoröffnern und Drohnen verbaut.

Bluetooth / W-LAN / App Steuerung

Die Signale werden von der App via Bluetooth oder W-LAN Verbindung an die Geräte geschickt. Häufig in Staubsaugerrobotern und Smart-Home Geräten verbaut.

Fernsteuerung vs Autonome Sensorsteuerung

Kategorie	Fernsteuerung	Autonome Sensorsteuerung
Steuerungsart	Nutzer gibt aktiv Kommandos	Roboter reagiert automatisch auf Veränderungen seines Umfelds
Kontrollbedarf	Benötigt Überwachung (Knöpfe drücken, Kommandos senden)	Benötigt keine Überwachung (arbeitet von selbst)
Beispiele	TV Fernbedienung, RC-Spielzeuge, Fernbedienung Klimaanlage Nutzer entscheidet, was wann passiert	Automatische Türen, Bewegungsmelder, automatische Seifenspender
Vorteile in der Bedienung	Entscheidung wann und wie etwas passiert liegt beim Nutzer	Einfach und schnell, kein Eingreifen notwendig
Anwendbare Szenarios	Wenn präzise Steuerung gebraucht wird	Für automatische Situationserfassung

Lernziele:

1. Das Grundkonzept der Fernsteuerung verstehen.
2. Zwischen Fernsteuerung und automatischer Sensorsteuerung unterscheiden können.
3. Verstehen, wie Variablen und Mapping genutzt werden, um Joystick Daten zu verarbeiten
4. Lernen, die Achsen des Joysticks mit Motoren zu verbinden.
5. Einen ferngesteuerten Roboter bauen und testen, der auf Input vom Joystick reagiert.



Praktisches Beispiel

(30 Minuten)

1. Variablen für Werte Links und Rechts des Joysticks erstellen.
2. Verwende den “Map” Block, um Werte des Joysticks (0-255) in Motorgeschwindigkeit (-100 - 100) umzuwandeln.
3. Verbinde Joysticks mit der Motorleistung und teste deinen Roboter.

The screenshot shows the MATRIX Robotics System software interface. The top bar includes the logo, file navigation (File, Save, Open, Help), and the file name ***Example6.mbn2**. On the left, a sidebar lists categories with icons: Mini Core (blue circle), Serial (orange circle), Sensing (teal circle), Extension Boxes (blue circle), 3rd Party Sensors (orange circle), Control (orange circle), Operators (green circle), and Variables (orange circle). The main workspace is a grid-based area where blocks are placed. Two blue blocks are visible: one in the 'Setup' area labeled 'MJ2 Joystick Begin' and one in the 'Loop' area labeled 'MJ2 Joystick Polling Data'. A 'Variables' section is also present in the workspace.

Schritt 1 : Ziehe den “MJ2 Joystick Begin” Block in den “Setup” Bereich.

Schritt 2 : Ziehe den "MJ2 Joystick Polling Data" in den "LOOP" Bereich.

The screenshot shows the MATRIX Robotics System software interface. On the left, a sidebar lists categories: Mini Core (Variables), Serial, Sensing, Extension Boxes, 3rd Party Sensors, Control, Operators, Variables, and My Blocks. A red arrow points from a callout box to the "Make a Variable" button under the Variables category. The main workspace displays a code editor with the following C++ code:

```
1 #include "MatrixMini.h"
2
3 void setup()
4 {
5     Mini.begin(LI_2, 0, 9600);
6     Serial.begin(9600);
7     Mini.PS2.begin();
8 }
9
10 void loop()
11 {
12     Mini.PS2.polling();
13 }
14
```

A callout box with a red border contains the text: "Klicke ‘Make a Variable’ um eine neue Variable zu erstellen". Below the code editor, there is a preview window showing the software's graphical interface with tabs for "Setup" and "Loop" containing code blocks labeled "MJ2 Joystick Begin" and "MJ2 Joystick Polling Data".

Variable

In der Programmierung ist eine **Variable** wie eine **Box**, die **Informationen speichert**. Du kannst eine Zahl, Text oder auch Sensordaten dort hineinlegen. Jede Box hat einen Namen, wie Temperatur, Score oder welchen Namen du ihr auch gegeben hast. Mit diesem Namen kannst du jederzeit in die Box hineinschauen.

Das besondere an einer Variable ist, dass sich **ihr Wert ändern kann**. Du kannst ihn entweder selbst ändern, oder das Programm anweisen, ihn automatisch für dich zu ändern, basierend auf Sensoren, Knopfdruck oder Ereignissen. Deswegen nennt man es **Variable** – ein Wert, der variabel ist.

MATRIX ROBOTICS SYSTEM

*Example6.mbn2

/dev/tty.usbse...

Variables

Mini Core

Serial

Sensing

Extension Boxes

3rd Party Sensors

Control

Operators

Variables

My Blocks

Make a Variable

Make a Block

1. Erstelle eine Variable namens "left".

New Variable

Left

OK Cancel

```
1 #include "MatrixMini.h"
2
3 void setup()
4 {
5     Mini.begin(LI_2, 0, 9600);
6     Serial.begin(9600);
7     Mini.PS2.begin();
8 }
9
10 void loop()
11 {
12     Mini.PS2.polling();
13 }
14 }
```

STEAM EDUCATION, FUTURE TECHNOLOGY.

MATRIX ROBOTICS SYSTEM

*Example6.mbn2

/dev/tty.usbse...

Variables

Mini Core

Serial

Sensing

Extension Boxes

My Blocks

3rd Party Sensors

Control

Operators

Variables

My Blocks

Make a Variable

Left

set Left to 0

change Left by 1

Make a Block

New Variable

Right

OK Cancel

```
#include "MatrixMini.h"
void setup()
{
    Mini.begin(LI_2, 9600);
    Serial.begin(9600);
    Mini.PS2.begin();
}
void loop()
{
    Mini.PS2.polling();
}
```

STEAM EDUCATION, FUTURE TECHNOLOGY.

The screenshot shows the MATRIX Robotics System software interface. The top bar includes the logo, file menu, and a file named *Example6.mbn2. The left sidebar lists categories: Variables, Mini Core, Serial, Sensing, Extension Boxes, 3rd Party Sensors, My Blocks, Control, Operators, Variables, and My Blocks. The main workspace contains a Scratch-like script:

```
Mini Begin: 18650x2, UART: On, Baud: 9600  
Setup: MJ2 Joystick Begin  
Loop: MJ2 Joystick Polling Data  
    set [Left v] to [0]  
    change [Left v] by [1]
```

A red arrow points from a callout box to the "set Left to 0" block in the loop.

Ziehe den "set Left " Variablenblock aus dem Panel.



*Example6.mbn2

Variables

Mini Core



Serial



Sensing



Extension Boxes



3rd Party Sensors



Control



Operators



Variables



My Blocks

Make a Variable

Left

Right

set Left ▾ to 0

change Left ▾ by 1

My Blocks

Make a Block

Ziehe den "set Right " Variablenblock aus dem Panel.





Map

“Map” konvertiert eine Zahl von einer Spanne in eine Andere. In diesem Fall konvertiert es die Joystick Werte (0-255) in Werte für die Motorleistung (-100 bis 100).

Das ist wie einen Maßstab in einen anderen umzurechnen – damit wenn sich der Joystick bewegt, die Motorleistung korrekt und stetig angepasst wird.

MATRIX ROBOTICS SYSTEM

*Example6.mbn2

Mini Core

Serial

Sensing

Extension Boxes

3rd Party Sensors

Control

Operators

Variables

My Blocks

Mini Begin 18650x2 ▾ UART: On ▾ Baud: 9600 ▾

Setup

Loop

Mini RGB LED LED1 ▾ R: 255 G: 0 B: 0

Mini DC Motor M1 ▾ Power: 50

Mini Servo Motor RC1 ▾ Angle: 50

Mini Button Button1 ▾ is Pressed?

Mini Ultrasonic D1 ▾ Distance(cm)

Mini GPIO

Mini D1 ▾ Digital Signal

Mini D1 ▾ Set to HIGH ▾

Mini A1 ▾ Analog Signal

Ziehe den "Joystick Stick Value" Block aus dem Panel.

Dieser Block liest den X-Achsen Wert des linken Joysticks auf dem Controller aus.

- Der Wert kann von **0 bis 255** reichen, wobei **128** die Ausgangsposition des Sticks ist.
- Er wird benutzt, um zu messen wie weit und in welche Richtung der Joystick bewegt wurde.

Mini Begin 18650x2 ▾ UART: On ▾ Baud: 9600 ▾

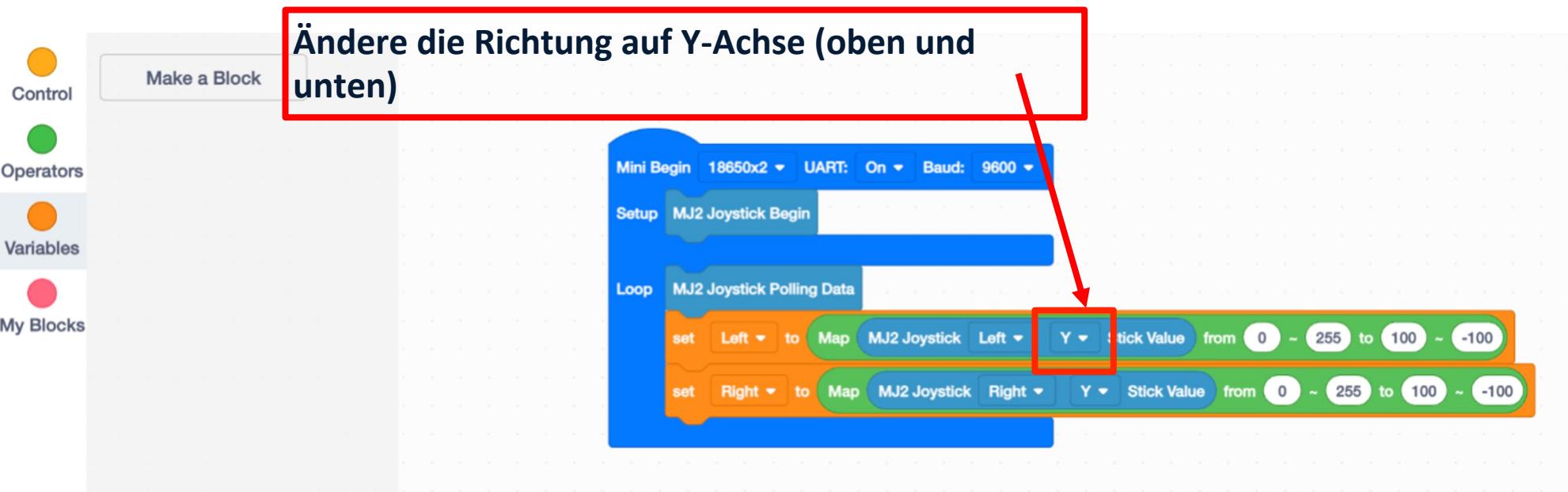
Setup MJ2 Joystick Begin

Loop MJ2 Joystick Polling Data

set Left ▾ to Map MJ2 Joystick Left ▾ X ▾ Stick Value from 0 ~ 255 to 100 ~ -100

set Right ▾ to Map MJ2 Joystick Right ▾ X ▾ Stick Value from 0 ~ 255 to 100 ~ -100

```
script
  setup
  loop
    set [Left v] to (map (MJ2 Joystick Left) (X) (Stick Value) (from 0 ~ 255 to 100 ~ -100))
    set [Right v] to (map (MJ2 Joystick Right) (X) (Stick Value) (from 0 ~ 255 to 100 ~ -100))
end
end
```



Warum auf die Y-Achse wechseln?

- Die X-Achse des Joysticks prüft Bewegungen nach links und rechts.
- Die Y-Achse kontrolliert die Bewegung des Joysticks nach oben und unten.

Nachdem unser Motor nur nach vorne oder hinten drehen kann, nutzen wir die Y-Achse zur Steuerung:

- Nach oben drücken → Motor rotiert vorwärts
- Nach unten ziehen → Motor rotiert rückwärts

Dadurch wird die Steuerung intuitiver für den Benutzer.

MATRIX ROTATION SYSTEM

*Example6.mbn2

No Device

The interface includes a sidebar with categories: Mini Core, Serial, Sensing, Extension Boxes, 3rd Party Sensors, Control, Operators, Variables, My Blocks, and Serial. The main area shows a script editor with the following blocks:

```

1 #include "MatrixMini.h"
2
3 float Left;
4 float Right;
5
6 void setup()
7 {
8     Mini.begin(LI_2, 0, 9600);
9     Serial.begin(9600);
10    Mini.PS2.begin();
11 }
12
13 void loop()
14 {
15     Mini.PS2.polling();
16     Left = map(Mini.PS2.LY, 0, 255, 1
17     Right = map(Mini.PS2.RY, 0, 255,
18     Mini.M1.set(50);
19     Mini.M2.set(50);
20 }
21

```

Übersetze die Spanne 0–255 auf -100–100.

Ziehe die “Mini DC Motor M1/M2” Kontrollblöcke aus dem Panel.

MATRIX ROBOTICS SYSTEM

*Example6.mbn2

Variables

Mini Core

Serial

Sensing

Extension Boxes

3rd Party Sensors

My Blocks

Control

Operators

Variables

My Blocks

Make a Variable

Left

Right

set Left to 0

change Left by 1

Make a Block

Setup MJ2 Joystick Begin

Loop MJ2 Joystick Polling Data

set Left to Map MJ2 Joystick Left Y Stick Value from 0 ~ 255 to 100 ~ -100

set Right to Map MJ2 Joystick Right Y Stick Value from 0 ~ 255 to 100 ~ -100

Mini DC Motor M1 Power: Left

Mini DC Motor M2 Power: Right

Nutze die Variable left für die Leistung von Motor M1;
Nutze die Variable rechts für die Leistung von Motor M2.

```
script
  setup [MJ2 Joystick Begin]
  loop
    [MJ2 Joystick Polling Data
      set [Left v] to [Map [MJ2 Joystick Left v] [Y v] [Stick Value] from 0 ~ 255 to 100 ~ -100]
      set [Right v] to [Map [MJ2 Joystick Right v] [Y v] [Stick Value] from 0 ~ 255 to 100 ~ -100]
      mini dc motor [M1 v] [Power: Left]
      mini dc motor [M2 v] [Power: Right]
    ]
  end
end
```

Aufgabe für den Unterricht

(10 Minuten)

Verwende verschiedene Objekte um einen Hindernisparcours zu bauen.
Fahre deinen Roboter mit Fernsteuerung und versuche um die
Hindernisse herumzufahren!