

# 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

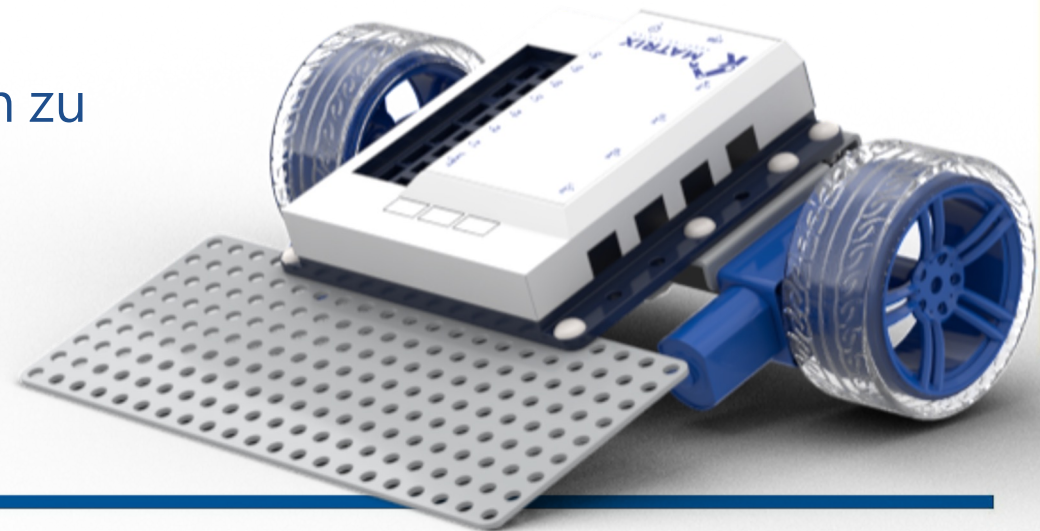
Die Signale werden von der App via Bluetooth oder W-LAN Verbindung an die Geräte geschickt. Häufig in Staubsaugerrobotern und Smart-Home Geräten verbaut.

# 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.





\*Example6.mbn2

Mini Core

Serial

Sensing

Extension  
Boxs

3rd Party  
Sensors

Control

Operators

Variables

Variables

Make a Variable

My Blocks

Make a Block

Schritt 1 : Ziehe den **"MJ2 Joystick Begin"** Block in den **"Setup"** Bereich.

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

Mini Begin

18650x2

UART: On

Baud: 9600

Setup

MJ2 Joystick Begin

Loop

MJ2 Joystick Polling Data



The screenshot shows the Matrix IDE interface. On the left, there is a sidebar with categories: Mini Core, Serial, Sensing, Extension, Boxes, 3rd Party, Sensors, Control, Operators, Variables, and My Blocks. The main workspace is a grid where a 'New Variable' dialog box is open. The dialog box has a title 'New Variable' and a text input field containing the word 'Left'. Below the input field are 'OK' and 'Cancel' buttons. A red rectangular box is overlaid on the workspace, containing the text '1. Erstelle eine Variable namens "left".' in white. On the right side of the IDE, there is a code editor showing C++ code for a Mini Core project. The code includes a header file 'MatrixMini.h' and defines 'setup()' and 'loop()' functions. The 'loop()' function calls 'Mini.PS2.polling()'. The status bar at the bottom right shows the line number '1591' and a 'Send' button.

1. Erstelle eine Variable namens "left".

```
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 }
```

MATRIX

\*Example6.mbn2

dev/tty.usbse...

Variables

Mini Core

Serial

Sensing

Extension

Boys

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

1. Erstelle eine neue Variable namens "right".

New Variable


Right





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 }
```

1591

Send





\*Example6.mbn2

Mini Core

Serial

Sensing

Extension

Boxs

3rd Party

Sensors

Control

Operators

Variables

My Blocks

Variables

Make a Variable

☐

Left

☐

Right

setLeft to 0

changeLeft by 1

My Blocks

Make a Block

Mini Begin18650x2UART: OnBaud: 9600

SetupMJ2 Joystick Begin

LoopMJ2 Joystick Polling Data

setLeft to 0

Ziehe den“set Left ” Variablenblock aus dem Panel.

**Variables**

Mini Core

Serial

Sensing

Extension  
Boxs

3rd Party  
Sensors

Control

Operators

Variables

My Blocks

Make a Variable

Left

Right

set Left to 0

change Left by 1

Make a Block

Ziehe den "set Right " Variablenblock aus dem Panel.



The screenshot shows the MATRIX programming environment interface. On the left, a panel titled 'Operators' contains various blocks categorized by color: Mini Core (blue), Serial (orange), Sensing (light blue), Extension (dark blue), Boxs (light orange), and 3rd Party (orange). The 'Map' block is highlighted in green. A red arrow points from a text box to this 'Map' block. The text box contains the instruction: 'Ziehe den "Map" Block aus dem Panel.' In the main workspace, a code block is visible with sections for 'Mini Begin', 'Setup', and 'Loop'. The 'Loop' section contains a 'MJ2 Joystick Polling Data' block, which has two 'set' blocks for 'Left' and 'Right' joysticks. Each 'set' block is followed by a 'Map' block. These 'Map' blocks are highlighted with a red rectangle. The 'Map' blocks are configured to convert joystick values from a range of 0 to 255 to a range of 0 to 1023.

**Ziehe den "Map" Block 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.

Mini Core

Mini Core

Serial

Sensing

Extension Boxs

3rd Party Sensors

Control

Operators

Variables

My Blocks

Mini Core

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



### 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** \*Example6.mbn2 No Device

**Mini Core**

- Mini Core
- Serial
- Sensing
- Extension Boxes
- 3rd Party Sensors
- Control
- Operators
- Variables
- My Blocks

**Mini GPIO**

- Mini D1 Digital Signal
- Mini D1 Set to HIGH
- Mini A1 Analog Signal

**Serial**

- Serial Print Hello
- Serial Print World with New Line
- Serial Write (ASCII) 65
- Serial Chart DataSet 32 64 128 , Interval 500 ms
- Is Serial Available?

**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**

**Serial**

**Serial Print Hello**

**Serial Print World with New Line**

**Serial Write (ASCII) 65**

**Serial Chart DataSet 32 64 128 , Interval 500 ms**

**Is Serial Available?**

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

**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: 50**

**Mini DC Motor M2 Power: 50**


```
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, 100, -100);
17     Right = map(Mini.PS2.RY, 0, 255, 100, -100);
18     Mini.M1.set(50);
19     Mini.M2.set(50);
20 }
21 }
```


Uploading, Serial Auto Close


1591 Send


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


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











\*Example6.mbn2

Mini Core

Serial

Sensing

Extension

Boxx

3rd Party

Sensors

Control

Operators

Variables

My Blocks

Make a Variable

Left

Right

set

Left

to

0

change

Left

by

1

Make a Block

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

Mini Begin

18650x2

UART: On

Baud: 9600

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

# 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!