

DATA SCIENCE 1

TUTORIALDAY - LEGO SPIKE EDUCATION

PROF. DR. CHRISTIAN BOCKERMANN

HOCHSCHULE BOCHUM

WINTERSEMESTER 2025/2026

- 1 Motivation - Lernen mit LEGO?
- 2 LEGO Hardware + Umgebung
- 3 Python + LEGO: spike Modul
- 4 Aufgaben

Warum mit LEGO spielen?

LEGO Spike Education Programm

- Praxisorientiertes Lernen
- Haptische Selbsterfahrung
- Fokus u.a. auf MINT Fächer



Warum mit LEGO spielen?

LEGO Spike Education Programm

- Praxisorientiertes Lernen
- Haptische Selbsterfahrung
- Fokus u.a. auf MINT Fächer

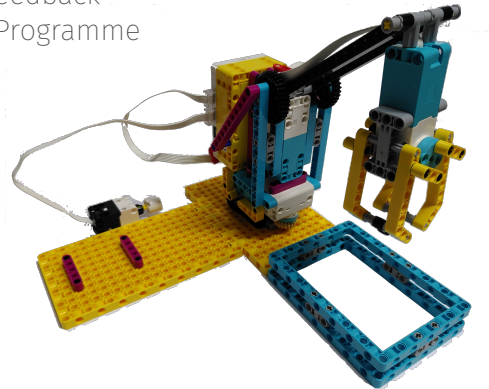


LEGO Serious Play

- Moderierte Innovationsprozess (u.a. in Unternehmen)
- Förderung von Kreativität/Innovation durch haptische Modellierung
- Verbessertes Verständnis dargestellter Themen

Motivation durch reale Modelle

- Programmieren lernen mit physikalischen Modellen
- Haptisches Feedback der eigenen Programme



LEGO Hub

- Microcontroller
- 100 MHz M4 Prozessor, 320 KB RAM
- 32 MB Flash Speicher für Daten
- mit Bluetooth LE Modul
- **MicroPython** “Betriebssystem”



LEGO Hub

- Microcontroller
- 100 MHz M4 Prozessor, 320 KB RAM
- 32 MB Flash Speicher für Daten
- mit Bluetooth LE Modul
- **MicroPython** “Betriebssystem”

Anschlüsse

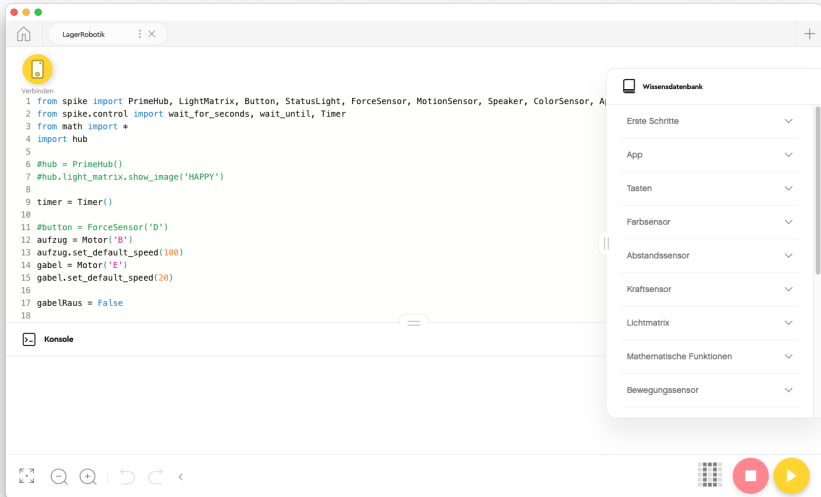
- USB zum Laden, Programmieren
- 6 Anschlüsse ([Ports](#)) für Motoren, Sensoren, usw.
- 5x5 LED Matrix (Anzeige)



LEGO Spike Legacy App

- Programmierumgebung für LEGO Hub
- USB-Verbindung mit HUB möglich
- Unterstützt graphische Programmierung + Python
- direktes Auslesen der Sensoren





Python + LEGO: spike Modul

Python Modul **spike**

- Python Modul zum Zugriff auf den Hub
- Funktionen für Hardware (Farbsensor, Motor, usw.)

```
from spike import PrimeHub

# Hub Objekt erzeugen
hub = PrimeHub()

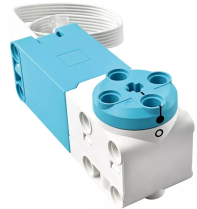
# Pixel in der Mitte anschalten:
hub.light_matrix.set_pixel(3,3, brightness=100)
```

Malen mit der LichtMatrix

```
from spike import PrimeHub  
  
hub = PrimeHub()  
  
for x in range(5):  
    for y in range(5):  
        if x == y:  
            hub.light_matrix.set_pixel(x,y)
```

LEGO Winkelmotor

- Winkelgenauer Motor
- Anschluss an einen der 6 Ports
- Drehen mit Winkel, Geschwindigkeit
- aktuellen Winkel auslesen



```
m = Motor('A')  
  
# Winkel abfragen  
winkel = m.get_position()  
  
# Um 90 Grad drehen mit Geschwindigkeit 10  
m.run_for_degrees(90, speed=10)
```

LEGO Farbsensor

- Vordefinierte Farben erkennen
- RGB Farbwerte auslesen



```
senor = ColorSensor('B')  
  
# Farbe als String auslesen  
farbe = sensor.get_color()  
  
# Farbe ausgeben  
print(farbe)
```

Aufgaben

L1 Verbindung mit Hub herstellen

- Starten Sie das LEGO Spike Legacy Programm
- Verbinden Sie den HUB mit dem PC und schalten Sie ihn ein
- Legen Sie ein neues Projekt an (Python!!)
- Falls der Hub nicht automatisch verbunden wird: klicken Sie auf das *Verbinden* Icon und folgen Sie den Anweisungen



Verbinden

L2: Ein erstes Programm

Schreiben und starten Sie das folgende Programm:

```
from spike import PrimeHub

hub = PrimeHub()

for x in range(5):
    for y in range(5):
        if x == y:
            hub.light_matrix.set_pixel(x,y)
```

L2.2: Den Rand einschalten:

Verändern Sie das folgende Programm so, dass alle Pixel am Rand des Displays leuchten!

```
from spike import PrimeHub  
  
hub = PrimeHub()  
  
for x in range(5):  
    for y in range(5):  
        if x == y:  
            hub.light_matrix.set_pixel(x,y)
```

Was muss dafür geändert werden?

L3: Programm mit Motor

Legen Sie das folgende Programm in LEGO Spike Legacy an:

```
from spike import Motor

m = Motor('A')

# Motor auf Winkel 0 fahren
m.run_to_position(0)

# Motor 4 Umdrehungen laufen lassen
m.run_for_rotations(4)
```

Starten Sie das Programm auf dem angeschlossenen Hub über den Start-Button unten rechts in der Spike Legacy Anwendung.

L4: Motor-Winkel auslesen

Benutzen Sie das folgende Programm, um den aktuellen Motor-Winkel auszulesen:

```
from spike import Motor

m = Motor('A')

while True:
    winkel = m.get_position()
    print("Winkel: ", winkel)
```

Starten Sie das Programm und drehen Sie am Motor!

L5: Programm: Motor ausrichten

Mit der Funktion `m.run_to_position(winkel)` fährt der Motor `m` zu einer Winkelposition `winkel`.

Es kann zusätzlich der Parameter `speed=X` angegeben werden, um die Geschwindigkeit des Motors zu kontrollieren.

- Schreiben Sie ein Programm, dass den Motor nacheinander auf die Winkel 0, 30, 60, 90, 120, 150, 180 und 210 fährt!
- Benutzen Sie als Geschwindigkeit den Wert 20.
- Ändern Sie das Programm so, dass der Motor in 5-Grad Schritten eine komplette Umdrehung macht (also die Winkel 0, 5, 10, usw. abfährt)!

L6: Spaß mit Farben

Mit **ColorSensor** kann auf den Farbsensor zugegriffen werden. Die Methode **get_color()** liefert einen Farb-Wert als String zurück:

```
farbSensor = ColorSensor('B')  
  
farbe = farbSensor.get_color()  
  
print("Farbe: " + farbe)
```

Starten Sie das Programm mehrfach und legen Sie dabei unterschiedlich farbige LEGO-Steine vor den Farbsensor!

L7: Spaß mit Farben

- Schreiben Sie ein Programm, dass kontinuierlich den Farbsensor ausliest und die erkannten Farben ausgibt!
- Testen Sie das Programm, in dem Sie unterschiedlich farbige LEGO-Steine vor den Farbsensor legen!

L7: Farben und Winkel

- Schreiben Sie ein Programm, das auf unterschiedliche Farben reagiert und in Abhängigkeit der Farbe den Motor auf einen bestimmten Winkel fährt.
- Z.B. bei der Farbe blau, soll der Motor zum Winkel 30 Grad fahren, bei grün zu 60 Grad.