**Einführung in Physical Computing und Programmierung mit Scratch und Python für Einsteiger**

## Kurzbeschreibung und pädagogischer Ansatz

Durch den Experimentierkoffer "Joypi" https://joy-it.net/de/products/rb-joypi wird ein erster Einstieg in die Welt der Elektronik und der Programmierung mit Scratch und Python ermöglicht, das kompakte und modulare Design vermeidet den Kauf von separaten Sensoren und Modulen und macht das Erlebnis eines Workshops sehr komfortabel und praktisch.

Der Tutor muss die Raspberry pi-Plattform kennen, die Details der Verwendung des Raspbian-Betriebssystems und die Grundlagen der Hardware-Architektur, der Elektronik, der Eingangs- und Ausgangsports sowie Erfahrung im Anschluss von Sensoren, Modulen und Aktoren und Programmierkenntnisse in Scratch3 und Python3 haben.

Die Aufgabe des Tutors im ersten Teil des Workshops ist es, eine kurze Einführung in die grundlegende Navigation der GUI des Raspbian-Betriebssystems zu geben, eine kurze Einführung in die Programmierumgebung Scratch zu geben und zu erklären, wie die Code-Blöcke konfiguriert werden, die die Steuerung der Ein- und Ausgabeports des Raspberry pi ermöglichen,

Der zweite Teil des Workshops konzentriert sich auf die Steuerung der Sensoren und Module des Koffers, die notwendigerweise in Python programmiert werden müssen. In diesem Teil gibt der Tutor eine kurze Einführung in die Programmiersprache Python, den Code-Editor GIMP und die Grundstruktur eines Skripts.
Der Joypi Koffer enthält eine Dokumentation mit Codebeispielen in der Programmierumgebung Scratch3 und Python3, die als Grundlage für die Entwicklung des Workshops dient.

Ziel des Kurses ist es, die Teilnehmer zu motivieren, ein Projekt zu realisieren, bei dem sie die im Koffer vorhandenen Ressourcen nach dem Prinzip [Dateneingabe -> Datenverarbeitung -> Ausgabe] nutzen. Die Eingangsdaten sind z. B. Werte, die von einem Sensor erfasst werden. Der Prozess kann die Logik sein, die programmiert wird, um diese Daten zu verarbeiten, die Ausgabe kann eine Information auf dem Bildschirm sein, ein Alarm, ein Licht von einer LED, die sich einschaltet, ein Relais, das eine Lampe einschaltet, ein Motor, der sich bewegt, usw.

Einige konkretere Beispiele könnten eine Geräuschstation, eine Wetterstation, eine Multimedia-Installation oder ein einfaches Animation sein.
Die vom Hersteller zur Verfügung gestellte Dokumentation ist sehr gut und die Herausforderung besteht darin, die zur Verfügung gestellten Codestücke zu verstehen und sie so zu modifizieren, dass sie die von uns vorgeschlagene Funktion erfüllen.

## Gruppengröße

6 Teilnehmer*innen bei einem Betreuungsschlüssel von maximal 1:6

**Ausrüstung für den Tutor:**
Laptop und Raspberry Pi Koffer-
Webcam
Beamer / Leinwand
Interner Verbindung Stabil min 20Mbits/sek. download
**Ausrüstung für Teilnehmer**
Joypi Koffer.
Es wird empfohlen, dass jeder Teilnehmer einen Laptop mit sich führt, da die Größe des Displays und der zur Verfügung gestellten Tastatur sehr klein ist. Dazu wird es notwendig sein, eine Remote-Verbindung des PC-Laptops - Rasberry über die VNC-Software einzurichten, was die Arbeit wesentlich komfortabler macht.

**Dauer**
Eine Tag mit vier Stunden Workshop – bestenfalls zuzüglich einer gemeinsamen Mittagspause

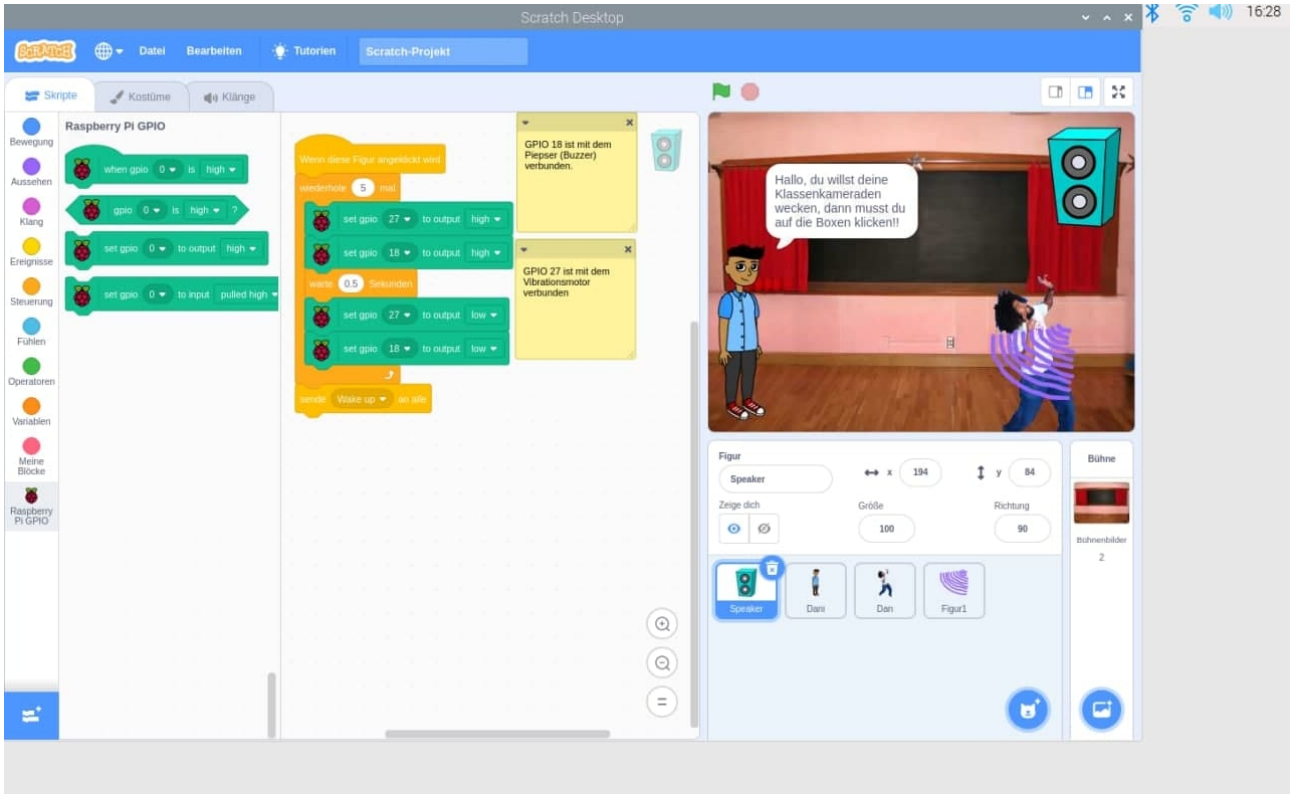
**Zielgruppe**
Kinder und Jugendliche zwischen 10 und 14 Jahren

**Aufwand**
Solange die Lehrer mit dem Thema vertraut sind, ist die Vorbereitung des Themas durchschnittlich, Der Lehrer wird einen Videoprojektor benötigen, da einige Dinge live erklärt werden müssen.

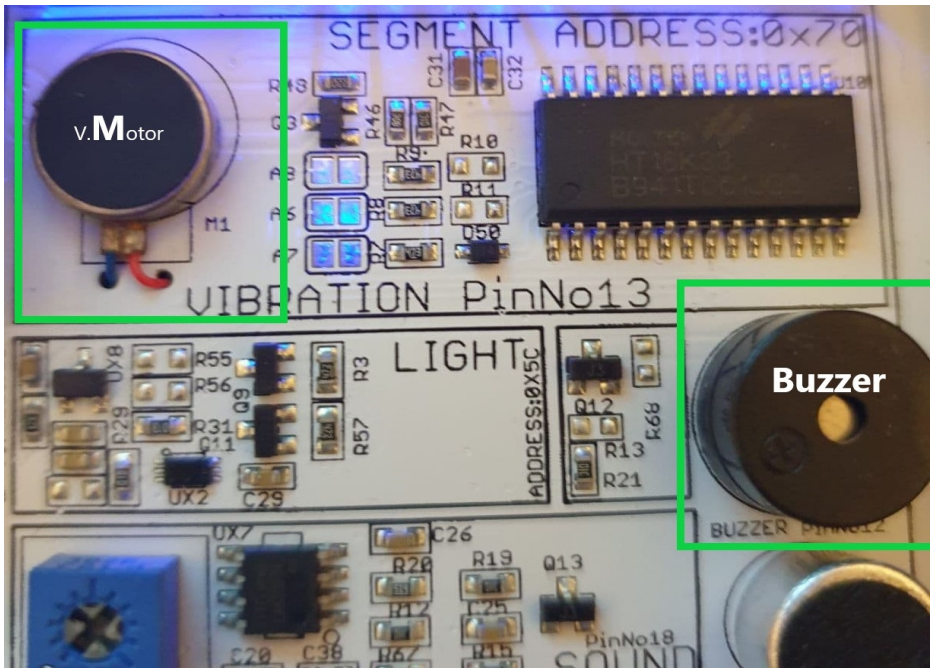
| **Workshop Teil** | **Inhalt** | **Pädagogische Hilfsmittel** | **Zeitdauer** |
|---|---|---|---|
| Teil 1 | Joypi -Tour, GPIOS/ Sensoren Modulen | Power Point, Camera,Frontalunterricht | 30 min |
| Teil 2 | Raspbian Oberfläche | Live-Beaemer | 30 min |
| Teil 3 | Scratch3 | Live-Beamer | 30 min |
| Teil 4 | Hands On | Internet, Frontal, | 1 Stunde |
| Teil 5 | Python3 | Live Beamer | 30 min |
| Teil 6 | Hands on | Frontal Live Beamer | 1 Stunde |

**Hands On**

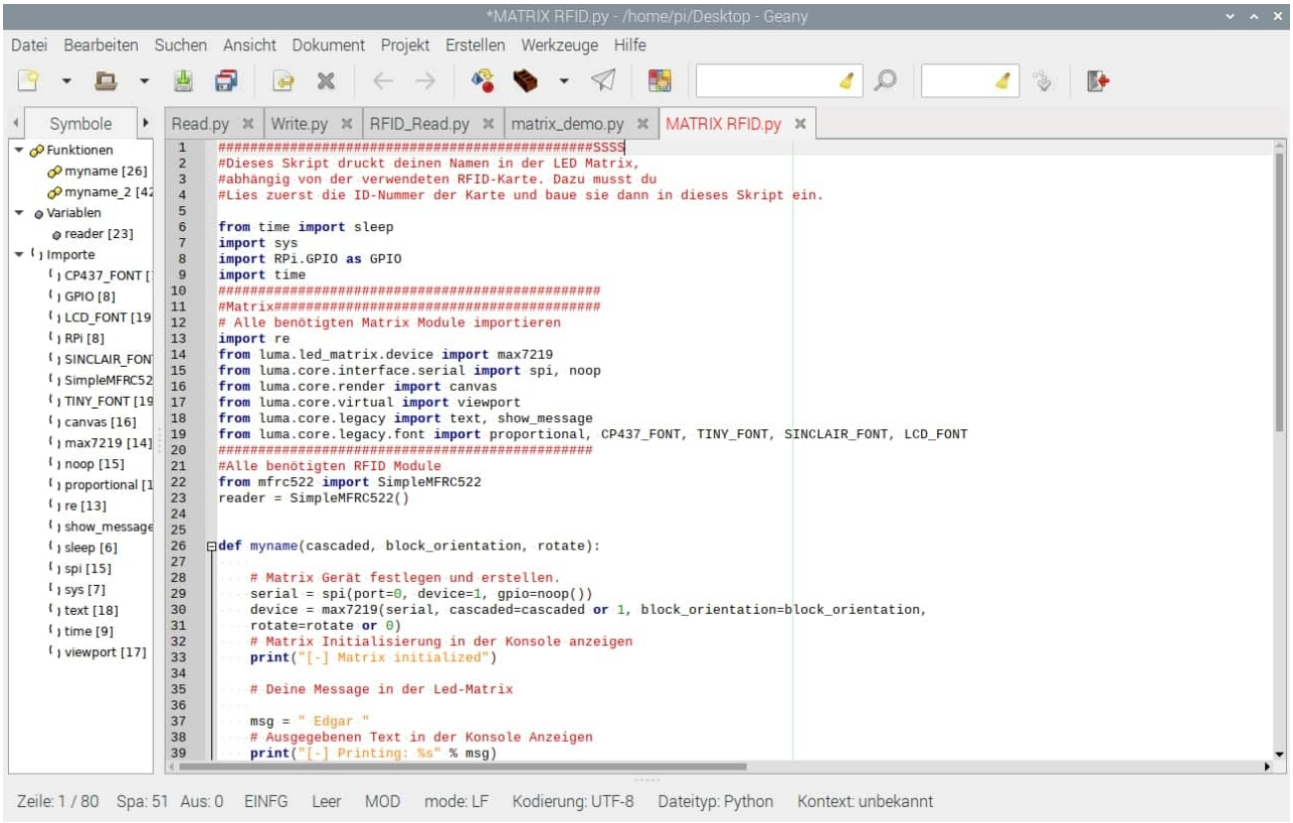
Scratch 3 integriert die Blöcke zur Steuerung der IO ports  des Raspberry Pi, so dass es möglich ist, Animationen zu erstellen, die mit den Sensoren und Modulen des Joypi Koffer verbunden sind.

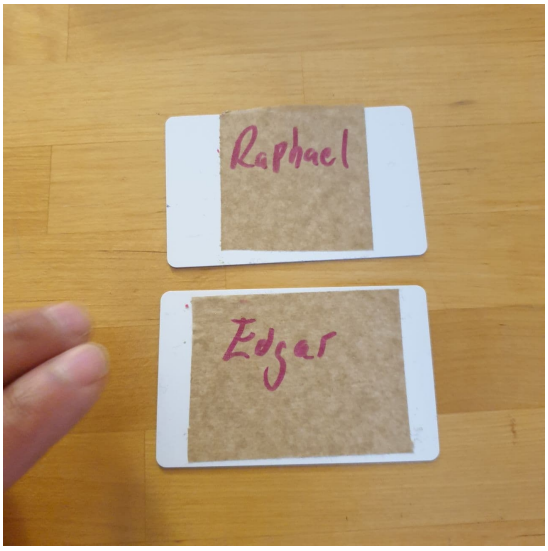


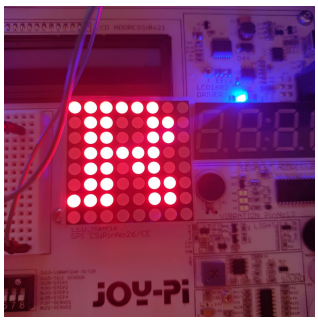

Dies sind die beiden Module, die im Scratch-Code integriert sind:

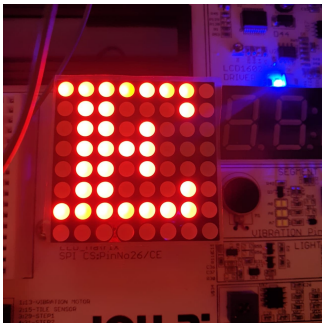


Im Abschnitt Pyhton 3 werden wir Geany als Code-Editor verwenden und wir können unser Programm modifizieren, um zum beispiel, RFID-Tags zu lesen und eine Karte zu verwenden, die, wenn sie vom Lesegerät gelesen wird, unseren Namen in der LED-Matrix anzeigt.



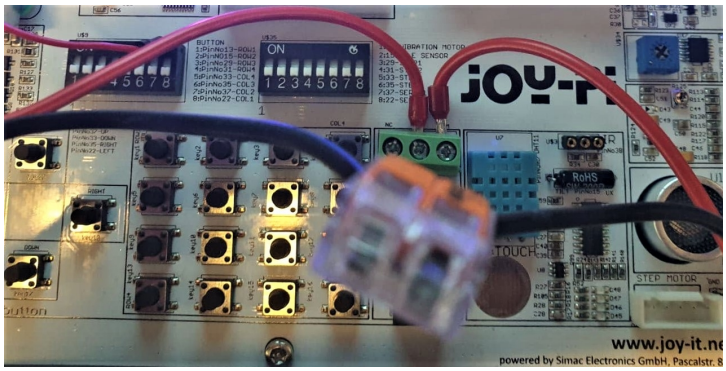
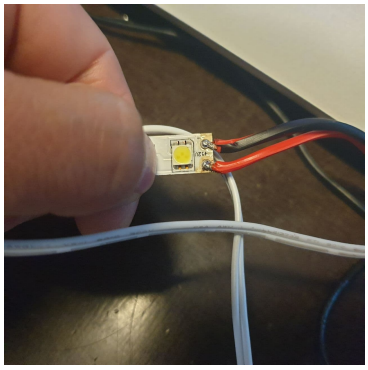



Man kann das Relaismodul verwenden, aber niemals nur Drähte mit Gleichstrom bis 12 Volt anschließen, niemals Wechselstrom oder 220V. Der Anschluss dieser Art von Experimenten muss immer von einem qualifizierten Erwachsenen beaufsichtigt werden.

*RFID_Read_Relay.py - /home/pi/Desktop/Joy-Pi/MFRC522-python - Geany

Datei Bearbeiten Suchen Ansicht Dokument Projekt Erstellen Werkzeuge Hilfe

Symbole | Read.py | Write.py | RFID_Read.py | RFID_Read_Relay.py

- Variablen
  - reader [18]
  - relay_pin [11]
- Importe
  - GPIO [8]
  - RPi [8]
  - SimpleMFRC52
  - sleep [6]
  - sys [7]
  - time [9]

```python
#Mit diesem Skript werden wir einen 12V-LED-Streifen einschalten,
#Der RFID-Kartenleser erkennt, ob es sich um die Karte mit der ID-Nummer handelt,
#und wenn dies der Fall ist, aktiviert er das Relais für 5 Sekunden,
#danach kehrt das Relais in den Zustand "Nicht angeschlossen" zurück.

from time import sleep
import sys
import RPi.GPIO as GPIO
import time
# definiere relay pin
relay_pin = 40
# Board Modus GPIO.BOARD
GPIO.setmode(GPIO.BOARD)
# relay_pin als Ausgang
GPIO.setup(relay_pin, GPIO.OUT)

from mfrc522 import SimpleMFRC522
reader = SimpleMFRC522()
GPIO.output(relay_pin, GPIO.HIGH)

try:
    while True:
        print("Halte eine RFID-Karte in die Nähe des Lesegerats")
        id, text = reader.read()
        print("ID: %s\nText: %s" % (id,text))
        #sleep(5)
        if id == 974890146809:
            GPIO.output(relay_pin, GPIO.LOW)
            print("Lampe An 5 sekunde")
            sleep (5)
            GPIO.output(relay_pin, GPIO.HIGH)

        else:
            GPIO.output(relay_pin, GPIO.HIGH)
            print("Lampe aus")


except KeyboardInterrupt:
    GPIO.cleanup()
    raise
```