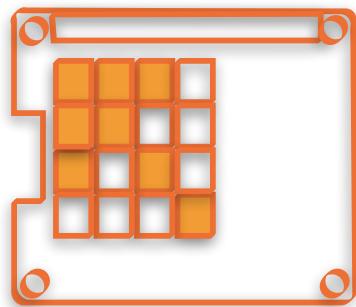
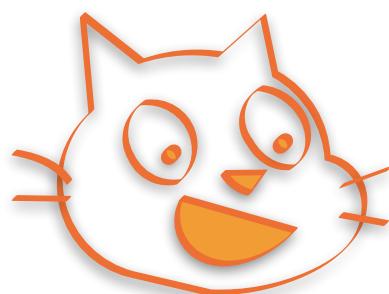
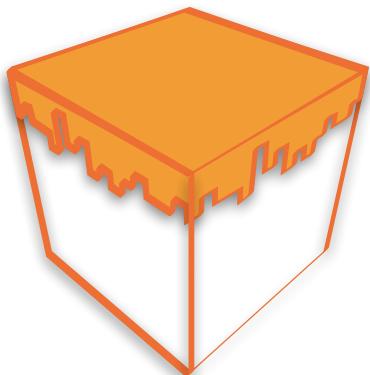
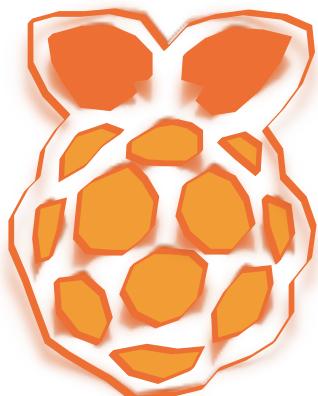
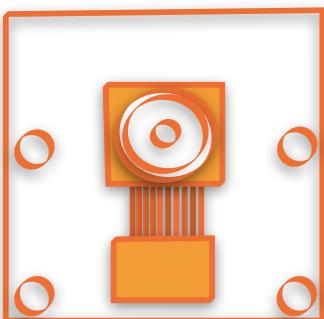


STATIONENLERNEN MIT DEM RASPBERRY PI



medienistik-lab
NR.2 /2017

VORWORT

In diesem Themenheft finden sich Vorschläge zum Stationenlernen mit dem Raspberry Pi. Neben einem Plakat, das an der Station ausgehängt werden kann, ist für jede Station auch eine kurze Anleitung vorhanden.

Je nach Station kommt unterschiedliche Hard- und Software zum Einsatz, z. B. die SenseHAT-Platine (Station 8), das Kamerabord (Station 2) oder der Cam Jam3-Roboterbausatz (Station 12).

Dieses Heft bietet (noch) keine detaillierte Handreichung für die Installation der Software, die an jeder Station benötigt wird. In der Regel handelt es sich um die Standardprogramme, die bei der Installation der Hardware automatisch mitinstalliert werden.

Bei Bedarf ergänze ich das Themenheft um detailliertere Beschreibungen der Hardware sowie weitere Tipps - momentan fand ich die Materialien einfach zu schade, um sie nicht zu teilen! :)

Noch eine kurze Info zum Schluss: Die Stationen sind unterschiedlich schwer - in der Codingschule verwenden wir rote, gelbe und grüne Ü-Ei-Controller (siehe Plakat auf Seite 3) zur Kennzeichnung.

- Stationen für Anfänger: 3, 4, 5, 8
- Stationen für Fortgeschrittene: 1, 2, 7, 9
- Stationen für Profis: 6, 10, 11, 13

Viel Spaß beim Nachmachen & bei Fragen einfach eine Mail schicken an:
info@medienistik.de

Liebe Grüße

Tobias Hübner

HUI! →

(EINE ECHTE HERAUSFORDERUNG)

OHO!

(SCHON SCHWIERIGER)

AHA!

(EINFACH)

STATION NR. 1

EINEN CONTROLLER BAUEN





STATION NR. 1: EINEN CONTROLLER BAUEN

In dieser Station lernst Du, wie Du Knöpfe und einen Joystick an den Raspberry Pi anschließt und programmierst. Dazu benötigst Du die kostenlose Software „Adafruit retrogame“. Diese ist auf Deinem Pi bereits vorinstalliert.¹

Wechsle zunächst in das Verzeichnis von retrogame mit dem Befehl:

```
cd Adafruit-Retrogame/configs
```

Kopiere nun die Datei retrogame.cfg in das Boot-Verzeichnis

```
sudo cp retrogame.cfg /boot/retrogame.cfg
```

Nun kannst Du wieder ins Verzeichnis „retrogame“ wechseln und das Programm aufrufen:

```
cd ..
```

```
sudo ./retrogame
```

Wenn Du nun eine Taste an einen GND-Anschluss und z. B. an Pin 27 angeschlossen hast, erscheint eine „3“ auf dem Bildschirm, wenn Du z. B. ein neues Terminal-Fenster öffnest.

Mit folgender Tastenkombination beendest Du das Programm:

```
CTRL-C
```

Nun kannst Du die Belegung der Tasten ändern. Gib dazu folgenden Befehl ein:

```
sudo nano /boot/retrogame.cfg
```

Nun kannst Du beispielsweise runter scrollen und die Zeile

```
3 27 #PiTFT Button 4
```

umändern in:

```
A 27 #PiTFT Button 4
```

Speichere mit:

```
CTRL-X (Frage mit „J“ + „ENTER“ beantworten)
```

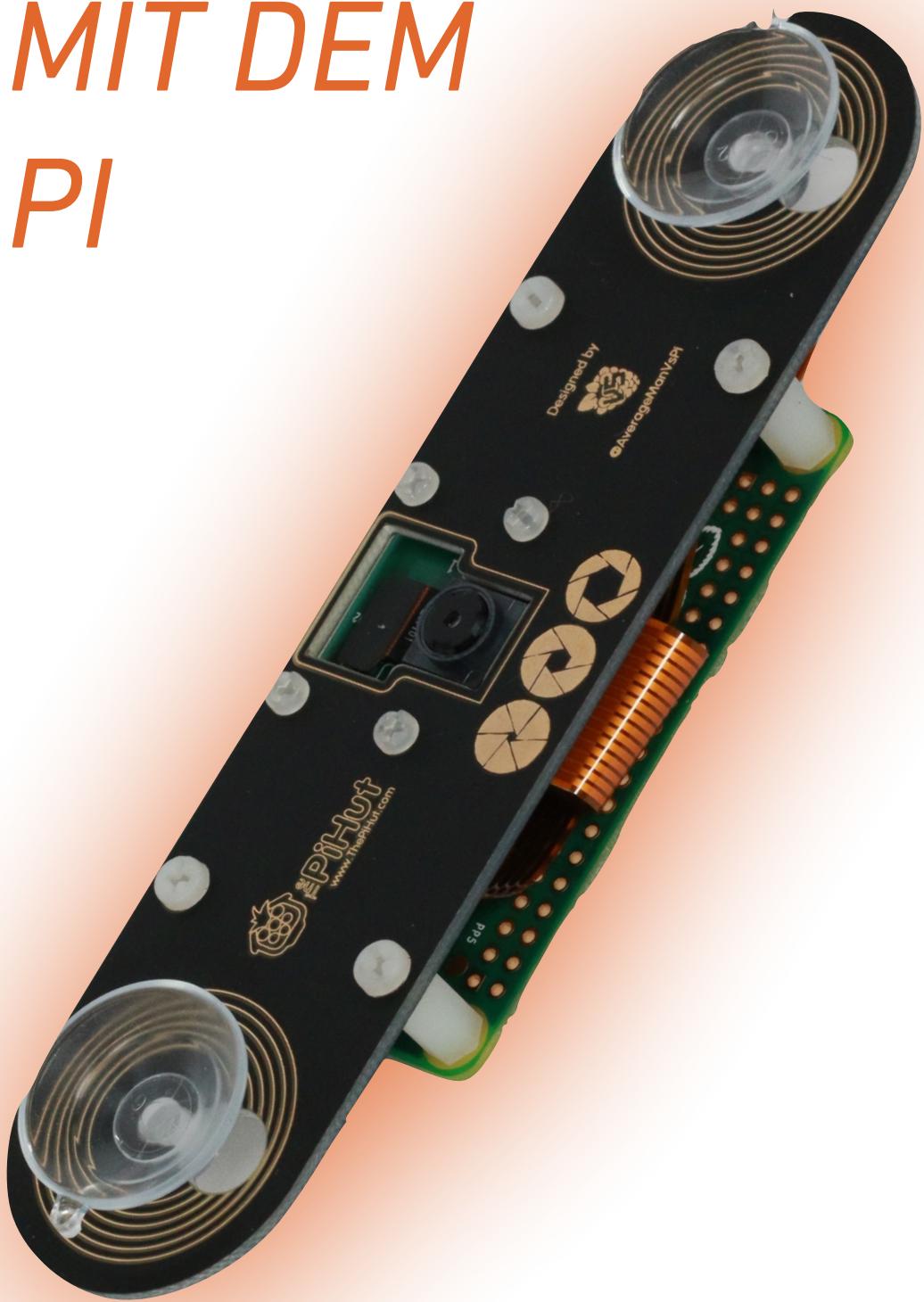
Rufe nun das Programm erneut auf (sudo ./retrogame.cfg) und drücke die Taste: Nun wird ein „a“ angezeigt.

Nun kannst Du mehrere Tasten und sogar den Joystick anschließen, die Tasten entsprechend programmieren und dann ein Spiel steuern (z. B. Minecraft oder andere Spiele aus dem „Games“-Ordner im Startmenü).

¹ Wenn Du Sie selber installieren möchtest, verbinde Deinen Pi mit dem Internet und gib einfach folgenden Befehl in die Kommandozeile ein: git clone <https://github.com/adafruit/Adafruit-Retrogame>

STATION NR. 2

FOTOS & VIDEOS MIT DEM PI





► STATION NR. 2: FOTOS & VIDEOS MIT DEM PI

Schließe den Pi Zero mit der Kamera an den Monitor an. Das Raspberry Pi Kamerabord hat 8 Megapixel und kann sowohl Fotos als auch Videos aufnehmen. Probier es einfach mal aus, indem Du ein Terminalfenster öffnest und folgenden Befehl eingibst:

raspistill -d -t 4000

Nun startet eine 40-sekündige Demo, die verschiedene Einstellungsmöglichkeiten und Effekte zeigt. Wenn Du ein Foto aufnehmen möchtest, gib einfach folgenden Befehl ein:

raspistill -o test.jpg

Das Programm bietet viele Einstellungsmöglichkeiten. Wenn Du z. B. ein Foto mit einem Effekt aufnehmen möchtest, gib einfach folgenden Befehl ein:

raspistill -ifx sketch -o test2.jpg

Nun erstellt die Kamera nach einer kurzen Vorschau ein Bild und speichert es unter dem Namen „test.jpg“ ab. Du kannst das Bild aufrufen, indem Du den Dateimanager oben links anklickst und das Bild, das sich im Ordner /home/pi befindet, mit einem Doppelklick aufruft. Mit einem rechten Mausklick und der Auswahl des Befehls „Löschen“ kannst Du es in den Papierkorb verschieben. Nimm nun ein Video auf mit dem Befehl:

raspivid -o test.h264

Du kannst es abspielen mit diesem Befehl:

omxplayer test.h264

Mit der Kamera lassen sich auch Slow-Motion-Aufnahmen erstellen. Gib dazu folgenden Befehl ein:

raspivid -w 640 -h 480 -fps 90 -t 8000 -o slowmo.h264

Das Video muss anschließend noch in ein anderen Format umgewandelt werden:

MP4Box -add slowmo.h264 slowmo.mp4

Nun kannst Du es abspielen:

omxplayer slow.h264

Wenn Du mit dem Befehl:

sudo nano Ausloeser.py

ein neues Programm schreibst und den Knopf an GPIO-Pin 14 und einen GND-Anschluss anschließt, kannst Du einen Auslöser basteln. Probier es doch mal aus:

(Die Anleitung geht auf der Rückseite weiter!)



GNU nano 2.2.6 Datei: Ausloeser.py

```
from picamera import PiCamera
from time import sleep
from gpiozero import Button

Knopf = Button(14)
Kamera = PiCamera()

Kamera.start_preview()
Knopf.wait_for_press()
Kamera.capture('/home/pi/Desktop/Bild1.jpg')
Kamera.stop_preview()
```

Speichere das Programm mit dem Befehl:

CTRL-X (Mit „J“ + „ENTER“ bestätigen)

Und führe es aus mit dem Befehl:

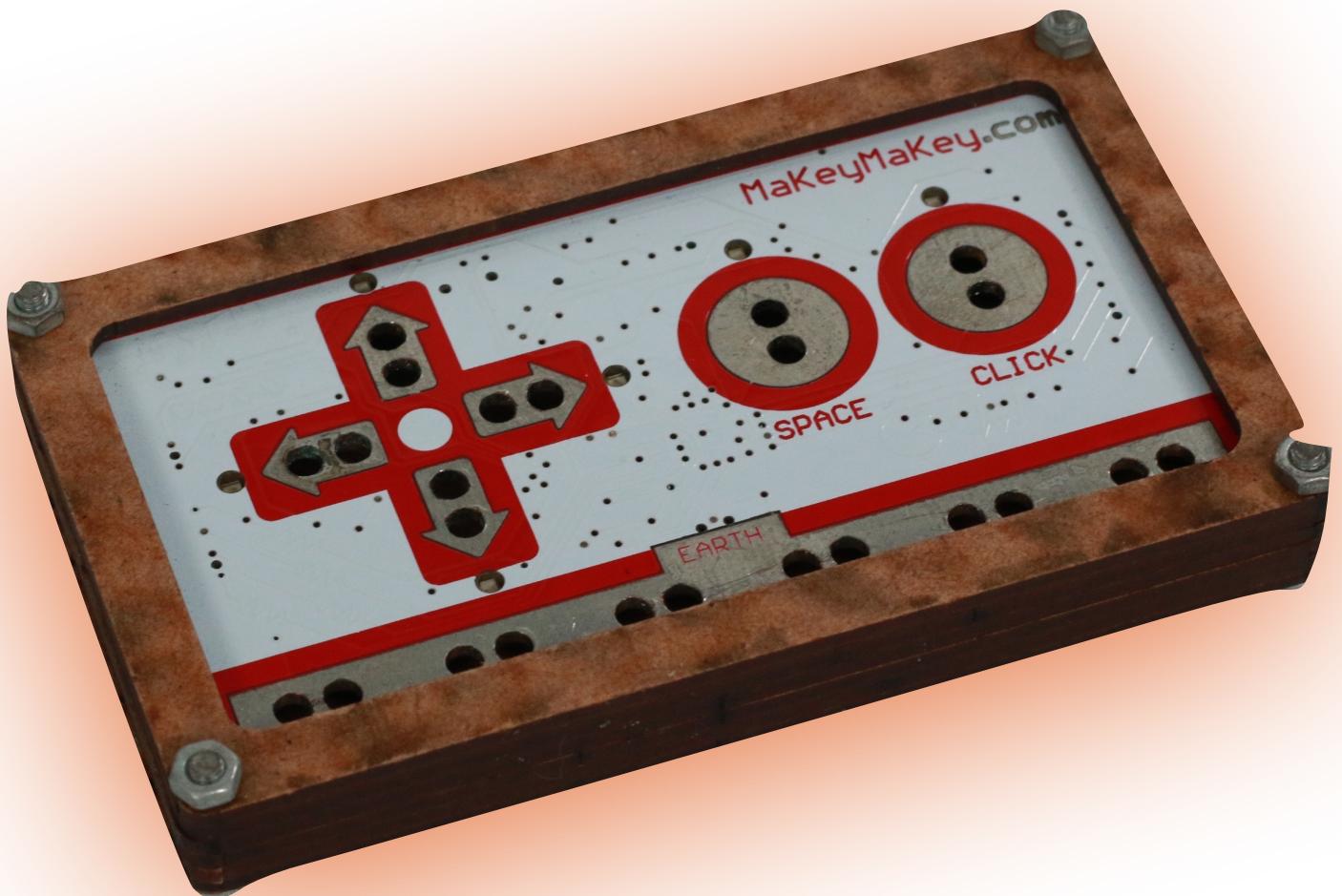
sudo python Ausloeser.py

Nun siehst Du eine Vorschau und wenn Du den Knopf drückst, wird ein Bild mit dem Namen „Bild1.jpg“ aufgenommen.

Denke daran, alle Deine Bilder und Videos zu löschen, bevor Du die Station wieder verlässt.

STATION NR. 3

MAKEY MAKEY- CONTROLLER



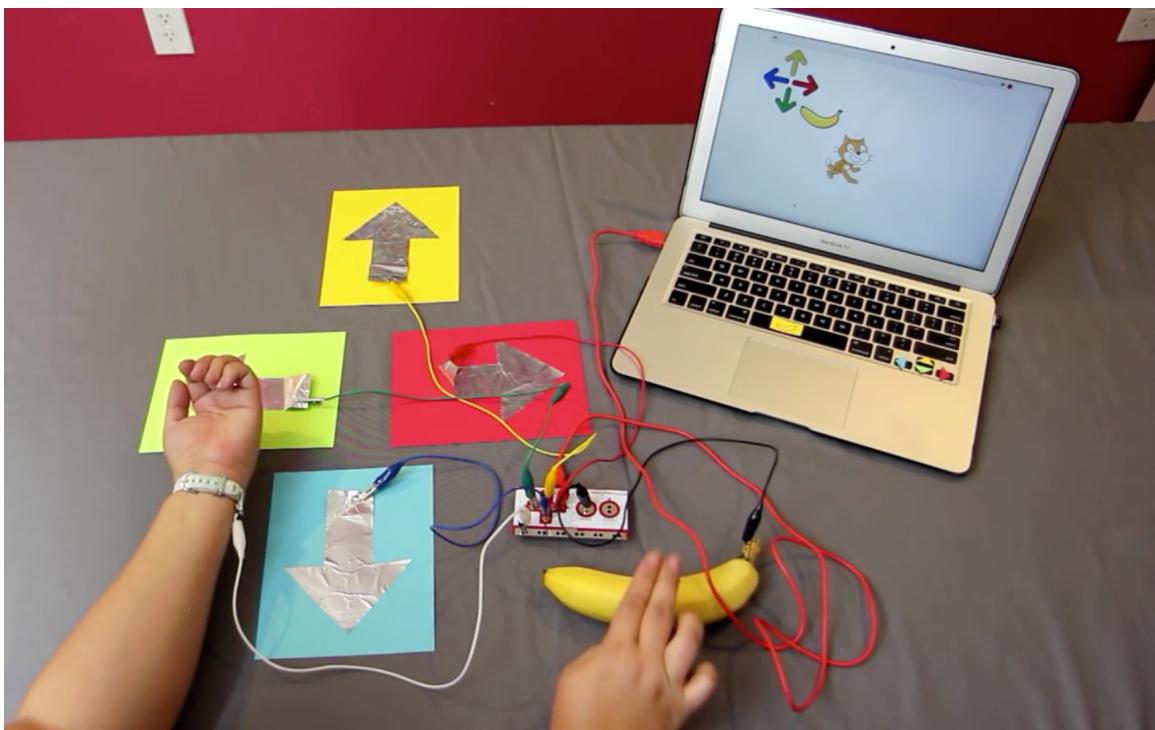
STATION NR. 3: MAKEY MAKEY

Der Makey Makey ist eine faszinierende Platine, mit der man sich ganz einfach einen eigenen Controller basteln kann. Schließ ihn einfach mit dem USB-Kabel an einen Laptop oder den Raspberry Pi an.

Anschließend kannst Du das beiliegende Armband an einen der „EARTH“-Anschlüsse unten auf der Platine anschließen. Wenn Du nun mit dem Finger eines der Felder auf der Platine berührst, wird der entsprechende Tasturbefehl an den Computer weitergeleitet.

Auf der Rückseite finden sich noch mehr Anschlüsse. Wenn Du dort ein Jumper-Kabel anschließt, kannst Du z. B. auch die Maus steuern. Probier es doch mal aus!

Nun hast Du Zeit, mit der ausliegenden Alufolie (oder einem anderen leitenden Gegenstand, etwa einer Banane) einen eigenen Controller zu bauen, der z. B. so aussieht:



Bildquelle: Video „Introduction to Makey Makey“ von „Teaching Robots“ (https://www.youtube.com/watch?v=-X3hb_YnM)

Schreibe nun ein einfaches Programm mit Scratch, das Du mit Deinem Controller steuern kannst!

STATION NR. 4

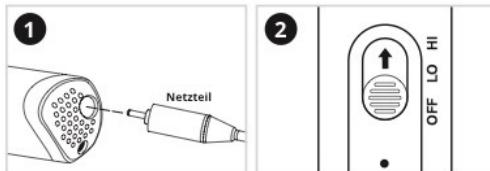
3DOODLE - WIE FUNKTIONIERT EIN 3D-DRUCKER?



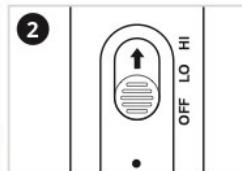
STATION NR. 4: 3doodler

Der 3doodle-Stift funktioniert so wie ein 3D-Drucker. Nimm Dir eine der Schablonen und probier ihn aus. Eine Anleitung liegt an der Station aus.

Schritt 1: Schließen Sie Ihren 3Doodler an, und warten Sie darauf, dass er sich erhitzt



Schließen Sie den **3Doodler-Stift** an.



Stellen Sie den **Bedienschalter** auf **HI**.

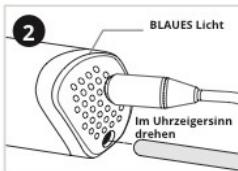


Das **Licht** wird **ROT** leuchten, während der **Stift** die **Temperatur** erreicht, die er benötigt, um Ihren **Kunststoffstift** zu schmelzen. Sobald das **Licht BLAU** ist, ist Ihr **Stift** zum **Extrudieren** von **Kunststoff** bereit.

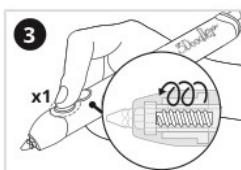
Schritt 2: Laden und extrudieren Sie Kunststoff



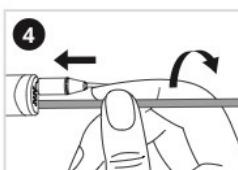
Wählen Sie einen Stift **ABS (MATTEN) Kunststoff** aus (kommt in der **3Doodler-Verpackung**).



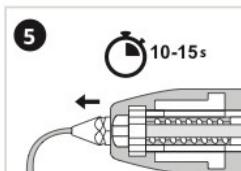
Versichern Sie sich, dass das **Licht** immer noch **BLAU** ist und laden Sie den **Kunststoffstift** in die **Ladevorrichtung**.



Klicken Sie einmal auf den **FAST-Knopf** und lassen Sie ihn wieder los. Sie werden hören, dass sich der **Antrieb in Gang setzt**.



Greifen Sie mit Ihrem **Daumen** und **Zeigefinger** vorsichtig nach dem **Kunststoff** und drehen Sie ihn im **Uhrzeigersinn**, während Sie ihn in die **Ladevorrichtung** drücken, bis Sie fühlen, dass der Stift von alleine durch das **Antriebsrad** gezogen wird.



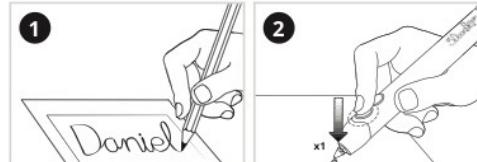
Nach 10-15 Sekunden wird der **Kunststoff** durch die **Düse** extrudiert. Der extrudierte **Kunststoff** wird sich nach wenigen **Sekunden** erhärten.



Drücken Sie einmal auf den **FAST-Knopf**, um mit dem **Extrudieren** zu stoppen.

Schritt 3: Doodlen Sie Ihren Namen

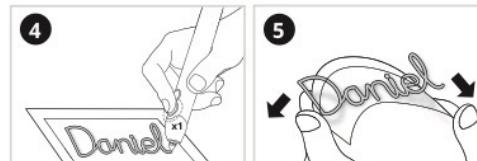
Benutzen Sie das Kästchen unten, um Ihr erstes Doodle zu schaffen - Ihren Namen!



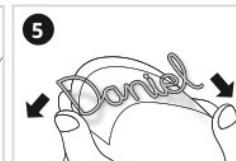
Schreiben Sie Ihren Namen in das **Kästchen** unten, mit einem Marker, einem Füller, einem Bleistift oder jedem anderen Stift Ihrer Wahl. Wir empfehlen **Schreibschrift** oder **Blockschrift** mit miteinander verbundenen **Buchstaben**.



Doodlen Sie Ihren Namen in einem fortgesetzten, ununterbrochenen Doodle, indem Sie den **Kunststoff** über das Papier ziehen, als ob Sie mit einem Bleistift schreiben würden, alle **Buchstaben** dabei miteinander verbunden. Bewegen Sie den 3Doodler langsam und gleichmäßig.

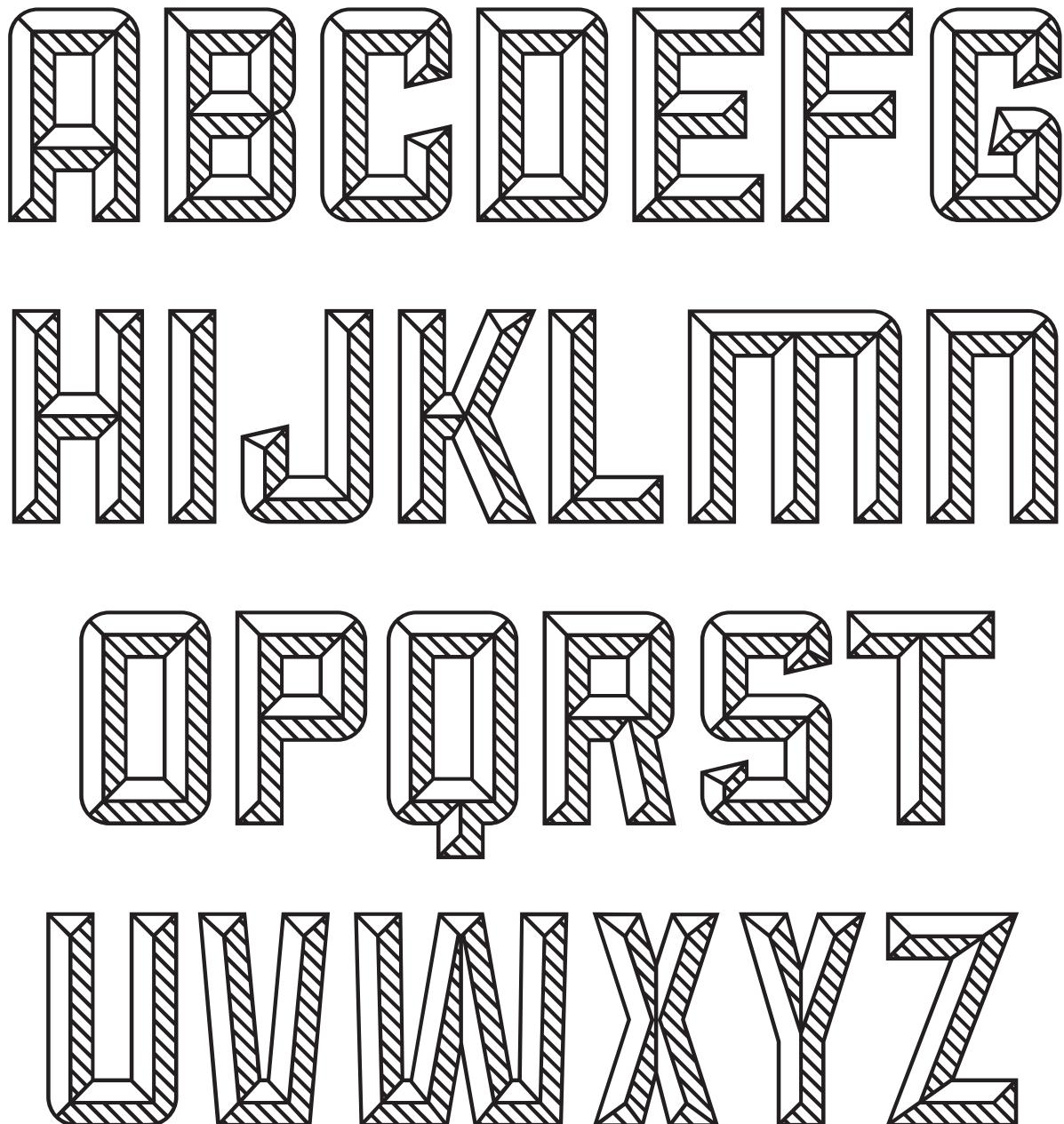


Wenn Sie das Ende Ihres Namens erreicht haben, beenden Sie das Extrudieren, indem Sie einmal auf den **FAST-Knopf** drücken.



Biegen Sie das Papier nach außen, um Ihren gedodelten Namen davon zu entfernen.

Hier kannst Du die Initialen Deines Namens nachzeichnen:



3Doodler

WWW.THE3DOODLER.COM

STATION NR. 5

MUSIK KOMPONIEREN MIT SONIC PI



The screenshot shows the Sonic Pi interface with the following code:

```
1 # Welcome to Sonic Pi v2.10
2
3 # Coded by Sam Aaron
4
5 use_debug false
6 load_samples :guit_em9, :bd_haus
7
8 live_loop :low do
9   tick
10  synth :zawa, wave: 0, phase: 1|, release: 5, note: (knit :e1, 12, :c1, 4).look, cuto
11  sleep 4
12 end
13
14 with_fx :reverb, room: 1 do
15   live_loop :lands, auto_cue: false do
16     use_synth :dsaw
17     use_random_seed 310003
18     ns = (scale :e2, :minor_pentatonic, num_octaves: 4).take(4)
19     16.times do
20       play ns.choose, detune: 12, release: 0.1, amp: 2, amp: rand + 0.5, cutoff: rran
21       sleep 0.125
22     end
23
```

The code defines two live loops: ':low' and ':lands'. The ':low' loop plays a 'zawa' synth with specific parameters and sleeps for 4 seconds. The ':lands' loop uses a 'dsaw' synth, sets a random seed, and plays a minor pentatonic scale 16 times with varying amplitudes and cutoff frequencies.

► STATION NR. 5: Sonic Pi-Experimente

Sonic-Pi ist eine sehr interessante Programmiersprache, die Musik produziert. Du benötigst also den Kopfhörer, der an der Station ausliegt, und die SD-Karte. Wenn Du das Programm startest, siehst Du links einen „Workspace“, indem Du Dein Programm schreiben kannst. Beginne mit dem Befehl:

play 60

und klicke auf den „Play“-Knopf oben links. Wenn du stattdessen:

play 60

eingibst, siehst du, dass ein Fehler angezeigt wird. Schreibe nun folgenden Code:

```
play 60
play 67
play 69
```

Du hörst, dass die Töne sehr schnell hintereinander abgespielt werden. Füge also Pausen hinzu und wiederhole die Tonfolge einmal mit einer Schleife:

```
2.times do
play 60
sleep 0.5
play 62
sleep 0.5
play 64
sleep 0.5
play 60
sleep 0.5
end
```

Wenn du über diesen Code folgenden Befehl schreibst:

with_synth „fm“

änderst Du das Instrument, das den Ton abspielt. Es gibt auch andere Instrumente, nämlich: „pretty_bell“, „dull_bell“, „beep“ und „saw_beep“. Nun kannst du das folgende Programm von der [Sonic Pi-Website](#) abtippen:

```
#Hiermit setzt Du das Tempo der Melodie fest.
with_tempo 350
2.times do
  play_pattern [40,25,45,25,25,50,50]
  play_pattern [25,50,25,30,35,40,45,50]
  play_pattern [25,50,25,30,35,40,45,50].reverse
end
2.times do
  with_synth „saw_beep“
  -Fortsetzung auf der nächsten Seite-
```

```

play_pattern [25,50,25,30,35,40,45,50].shuffle
play_pattern [25,50,25,30,35,40,45,50].reverse
end
# Nun spielen wir mehrere Melodien auf einmal.
in_thread do
  with_synth "saw_beep"
  10.times do
    if rand < 0.5
      play 37
    else
      play 49
    end
    sleep 2
  end
end
in_thread do
  with_synth "pretty_bell"
  20.times do
    play 49
    sleep 1
  end
end

```

Nun kannst du versuchen, eine eigene Melodie zu programmieren.

Auf der rechten Seite siehst du, welche Taste auf dem Klavier welcher Zahl entspricht. Vielleicht kennst Du ja eine einfache Melodie, die Du mit Sonic Pi programmieren kannst, z. B. diese hier:

```

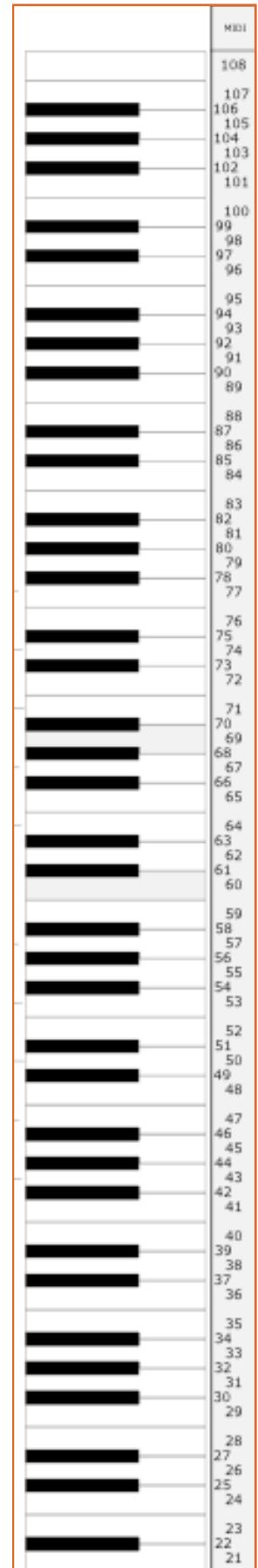
3.times do
  play 55
  sleep 0.5
  play 55
  sleep 0.3
  play 55
  sleep 0.1
  play 55
  sleep 0.25
  play 55
  sleep 0.1
  play 55
  sleep 0.25
  play 57
  sleep 0.25
  play 59
  sleep 0.25
end

```

Der Code geht hier weiter..

↑

play 60
sleep 0.5
play 59
sleep 0.5
play 57
sleep 0.5
play 59
sleep 0.5



Bildquelle: [Wikimedia Commons](#)

STATION NR. 6

MIT EDU-KITS

BASTELN



STATION NR. 6: EDU-KITS

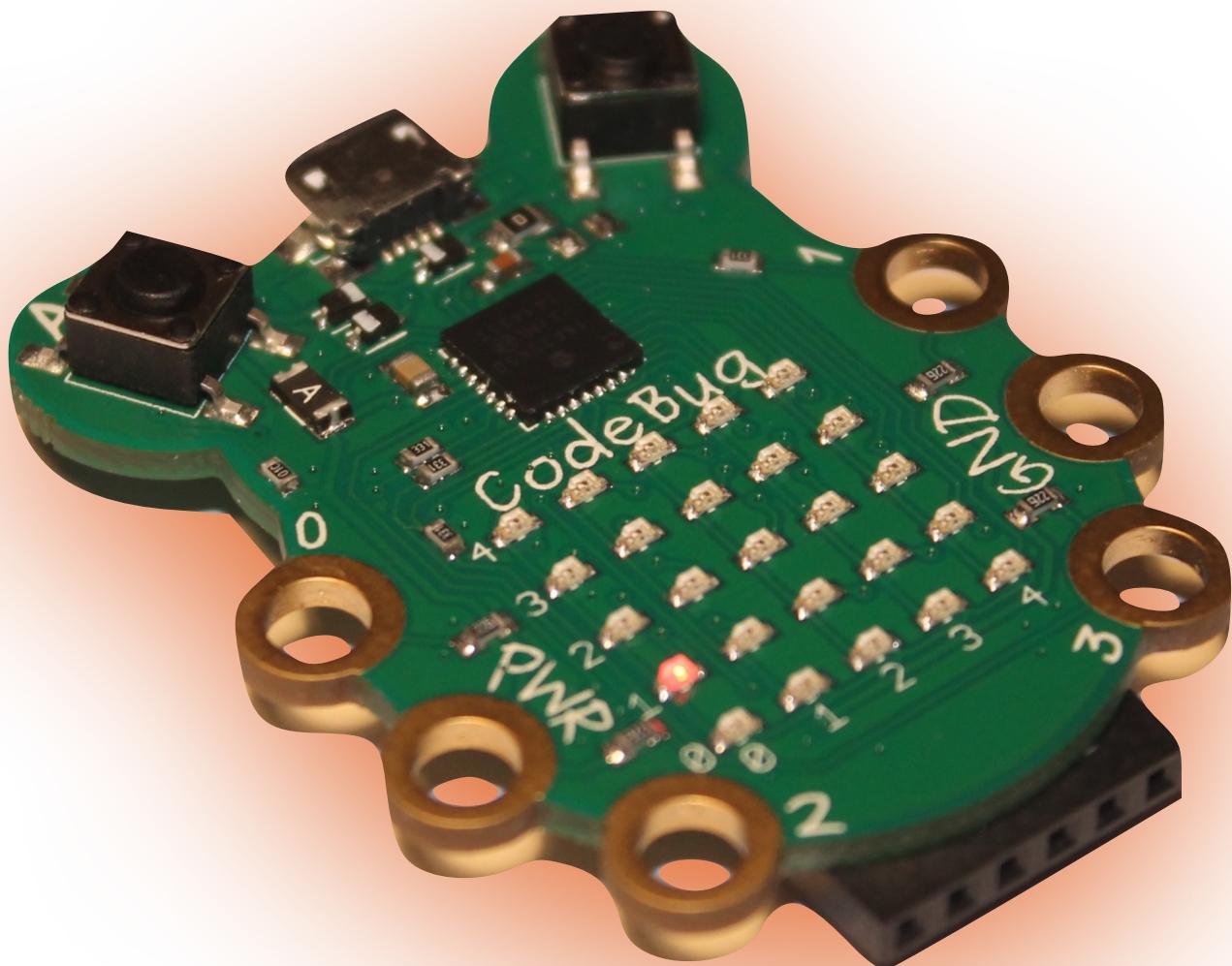
Mit den Edu-Kits Nr. 1 & 2 kannst du tolle kleine Elektronik-Projekte basteln. Anleitungen hierfür findest Du im Ordner „Documents/CamJam Materialien“. Probier es einfach einmal aus.



Viel Erfolg!

STATION NR. 7

MIKRO- CONTROLLER



STATION NR. 7: MIKROCONTROLLER

Es gibt neben dem Raspberry Pi noch andere kleine Computer, mit denen man interessante Projekte realisieren kann - es sind so genannte Mikrocontroller wie der micro:bit¹ oder der Codebug. An dieser Station kannst Du eines dieser Projekte ausprobieren. Das folgende Beispiel stammt aus dem Buch „Adventures in Minecraft“ von Martin O’Hanlon und David Whale. Der erste Schritt besteht darin, den micro:bit so zu programmieren, dass er ein ärgerliches Gesicht zeigt, wenn er geschüttelt wird. Schließe den micro:bit dazu an den Raspberry Pi mit dem USB-Kabel an. Öffne dann im Menü „Entwicklung“ das Programm „mu“. Klicke auf „New“ und ergänze dann den Code wie folgt:

```
from microbit import *
while True:
    if accelerometer.current_gesture() == "shake":
        display.show(Image.ANGRY)
    else:
        display.show(Image.HAPPY)
```

Klicke auf „Flash“, um das Programm auf den angeschlossenen micro:bit zu überspielen. Nun soll der micro:bit mit dem Raspberry Pi kommunizieren. Wenn der micro:bit geschüttelt wird, soll ein Signal an Pin0 gesendet werden, wenn man den Knopf A drückt - so sieht der Code dafür aus:

```
from microbit import *
while True:
    if accelerometer.current_gesture() == "shake":
        display.show(Image.ANGRY)
        pin0.write_digital(1)
    elif button_a.is_pressed():
        display.show("A")
        pin1.write_digital(1)
    else:
        display.show(Image.HAPPY)
        pin0.write_digital(0)
        pin1.write_digital(0)
```

Schließe nun den micro:bit an den Raspberry Pi an. Pin0 auf dem micro:bit muss mit GPIO Pin17 verbunden werden, Pin1 (micro:bit) mit GPIO Pin 27 (Raspberry Pi).

¹ Der micro:bit wurde beispielsweise an alle Schülerinnen und Schüler der siebten Klassen in England kostenlos verteilt. Er wurde vom öffentlichen Rundfunk (der BBC) produziert und es gibt zahlreiche kostenlose Materialien im Netz, die zeigen, wie man den micro:bit programmiert.

Rufe nun Minecraft auf und erstelle eine neue Welt. Öffne anschließend den IDLE3-Editor und erstelle ein neues Programm. Speichere es als „mc_micro.py“. Das Programm soll folgenden Code enthalten:

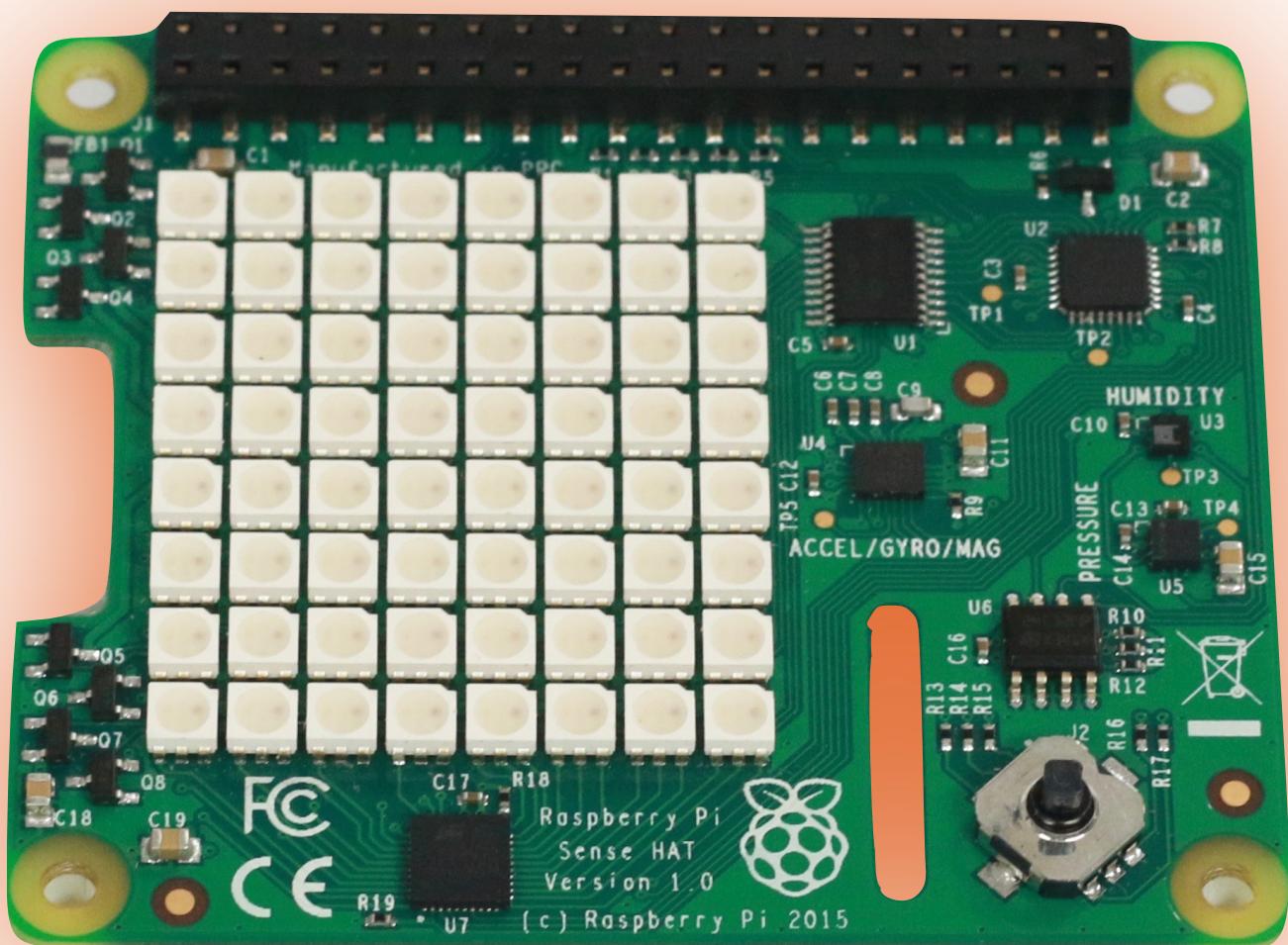
```
from mcpi.minecraft import Minecraft
from gpiozero import DigitalInputDevice
from time import sleep
mc = Minecraft.create()
mc.postToChat("Micromine bitcraft")
pin0 = DigitalInputDevice(17)
pin1 = DigitalInputDevice(27)
while True:
    sleep(0.1)
    pos = mc.player.getPos()
    if pin0.value == 1:
        pos.y = pos.y + 0.5
        mc.player.setPos(pos)
    if pin1.value == 1:
        pos.y = pos.y - 1
        mc.setBlock(pos, 0)
```

Dieses Programm bewirkt, dass ein kleines Erdbeben in Minecraft ausgelöst wird, wenn Du den micro:bit schüttelst. Drückst Du Knopf A, dann verschwindet außerdem der Block unter Dir.

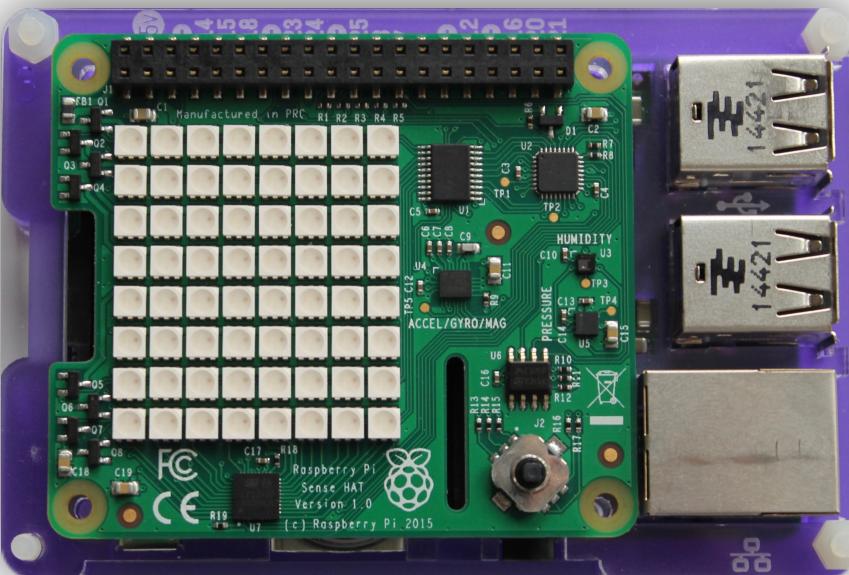
Versuche nun, das Programm noch zu ergänzen. Du kannst z.B. unter Deiner Spielfigur einen TNT-Block entstehen lassen - der Code hierfür lautet 46,1.

STATION NR. 8

DER SENSE-HAT



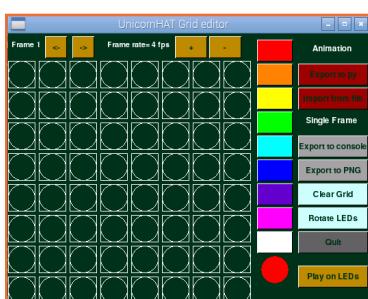
STATION NR. 8: SENSE HAT



Der Sense Hat wurde von der Raspberry Pi Foundation für den Einsatz in der internationalen Raumstation entworfen. Er enthält neben einem kleinen Joystick und einem 8x8 LED-Display auch viele Sensoren, mit denen sich z. B. die Temperatur, die Luftfeuchtigkeit oder die Lage des Geräts im Raum messen lassen.

Um das Display auszuprobieren, gibt man einfach folgende Befehle ein:

```
cd RPi-8x8GridDraw
sudo python 8x8grid.py
```



Arbeitsanregung:

Probiere aus, wie man mit dem Programm Animationen erstellt und erstelle eine kleine Bewegtbild-Sequenz. Mit dem Knopf „Export to py“ lässt sich die Animation als Python-Programm speichern.

Der Lagesensor lässt sich am besten mit dieser 3D-Darstellung der Soyuz-Rakete ausprobieren:

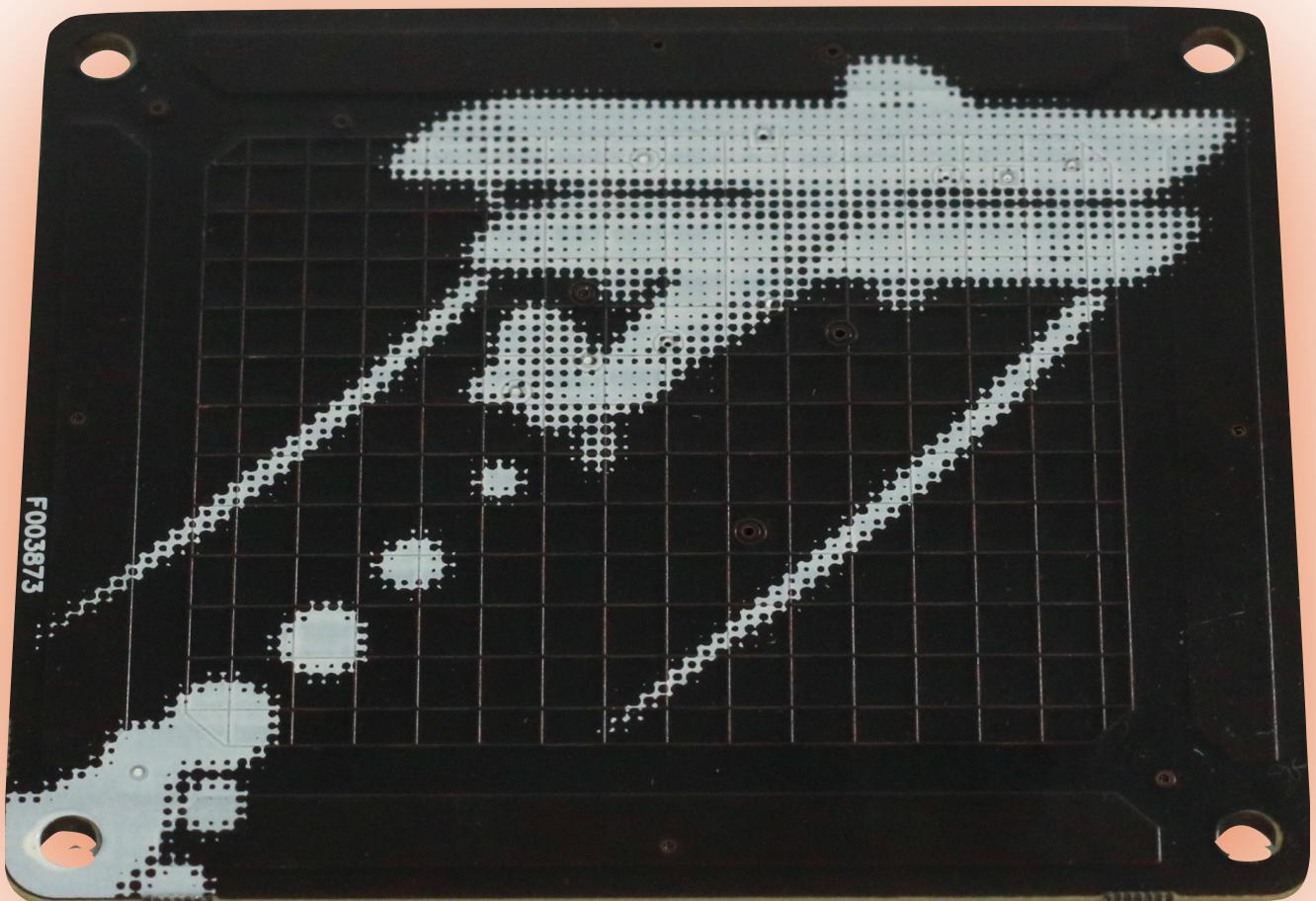
```
cd apollo-soyuz
./soyuz.py
```

Im „sense-hat-examples/python“-Ordner befinden sich auch noch andere Programme, z.B.:

jokes.py - zeigt Programmierer-Witze an
humidity.py - zeigt die Luftfeuchtigkeit an
minecraft_color.py - zeigt die Farbe des aktuellen Felds an (Minecraft erforderlich)
minecraft_map.py - zeigt die Farbe der Blöcke im Umfeld an (Minecraft erforderlich)

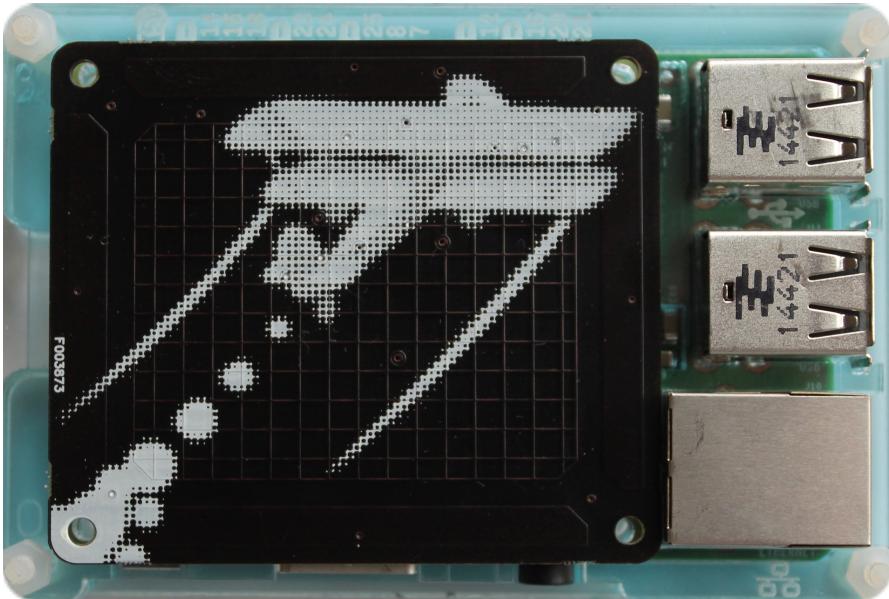
STATION NR. 9

DER
SKYWRITER-HAT





STATION NR. 9: SKYWRITER HAT



Der Skywriter Hat ist eine Art Touchpad für den Raspberry Pi. Er kann jedoch auch erkennen, wenn sich die Hand einige Zentimeter über der Fläche befindet. Dadurch ergeben sich kreative Möglichkeiten der Steuerung von Programmen. Schließe an dieser Station den kleinen Lautsprecher an den Pi an.

Um den Skywriter auszuprobieren, kannst Du die folgenden Programme aufrufen:

```
cd Pimoroni  
cd skywriter  
cd examples  
sudo python test.py  
sudo python mouse.py  
sudo python keyboard.py  
sudo python therermin.py  
sudo python synth.py
```

Arbeitsanregung:

Erstelle eine Präsentation mit LibreOffice Impress. Starte anschließend das Programm /home/pi/Pimoroni/skywriter/examples/magic_control.py. Nun kannst Du die Präsentation mit Gesten auf dem Skywriter steuern.

Der Skywriter lässt sich auch zur Steuerung in Minecraft verwenden. Öffne dazu Minecraft und starte dann folgendes Programm:

```
cd Minecraft  
sudo python x-wing.py
```

Findest Du heraus, wie du den X-Wing steuern kannst?

STATION NR. 10

DER PIANO-HAT



STATION NR. 10: PIANO HAT



Der Piano Hat geht zurück auf den Entwurf des 15jährigen Zachary Igelman. Mit Hilfe der Firma Pimoroni wurde es zu einem fertigen Produkt weiterentwickelt.

Der Piano Hat besitzt 13 Klaviertasten und drei weitere Tasten oben rechts. Zusätzlich befinden sich 16 LEDs auf dem Board. Alle Tasten und Lampen können einzeln programmiert und zur Steuerung von Programmen eingesetzt werden.

Um den Piano Hat auszuprobieren, tippe folgende Befehle ein:

```
cd Pimoroni
cd pianohat
cd examples
  sudo python buttons.py
  sudo python leds.py
  sudo python learn-to-play-py
  sudo python simple-piano.py
```

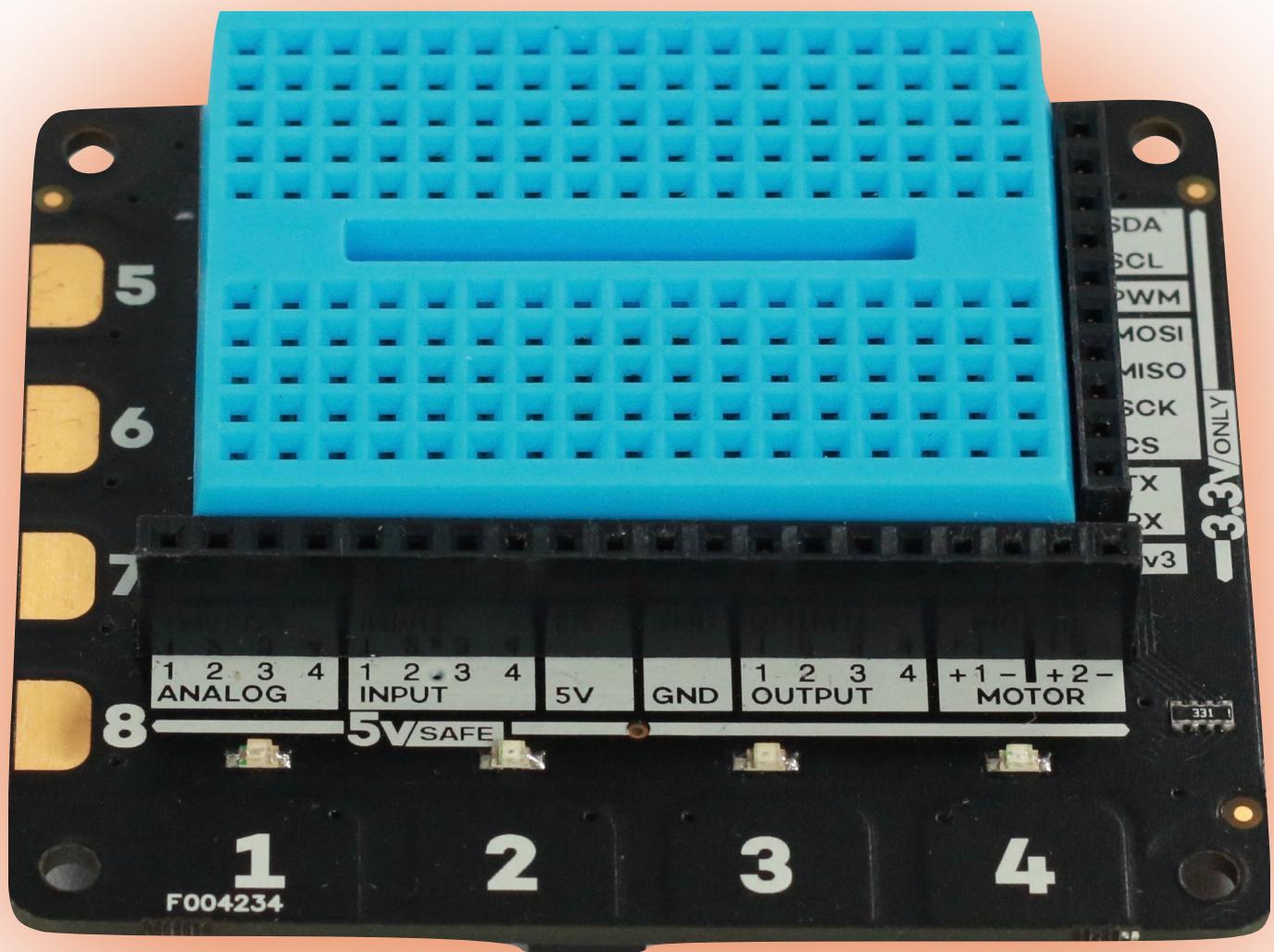
Das Programm „simple-piano.py“ spielt WAV-Sounds ab, die sich im Verzeichnis „/home/pi/Pimoroni/pianohat/examples/sounds“ befinden.

Arbeitsanregung:

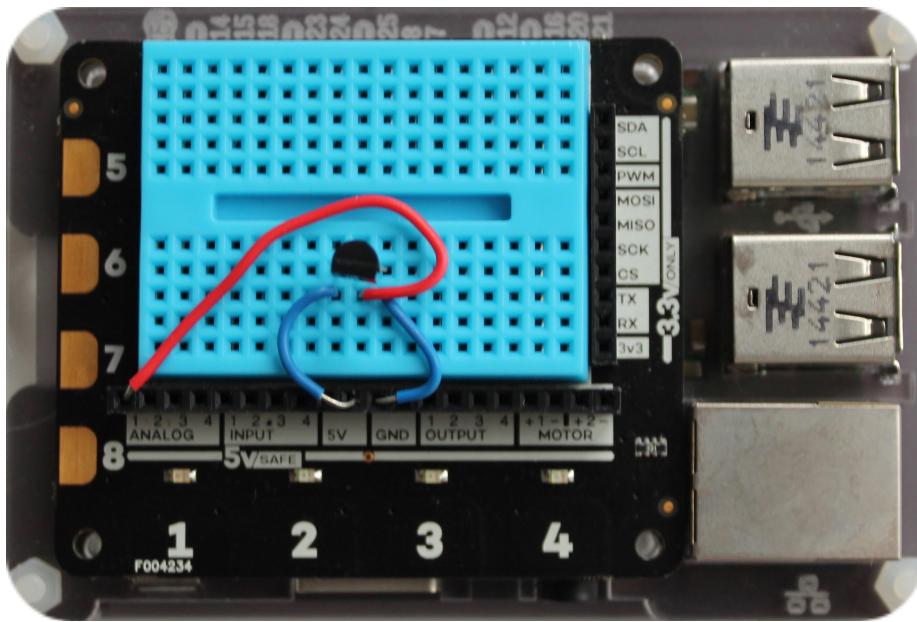
Schließe das USB-Mikrofon an, starte das Programm „Audacity“ und nimm eigene Samples auf, die Du dann mit dem Piano Hat abspielen kannst.

STATION NR. 11

DER *EXPLORER-HAT*



STATION NR. 11: EXPLORER HAT PRO



Der Explorer HAT Pro bietet ein kleines Steckbrett, vier kleine Touchpads, vier LEDs sowie analoge Eingänge und die Möglichkeit, zwei 5V-Motoren anzusteuern.

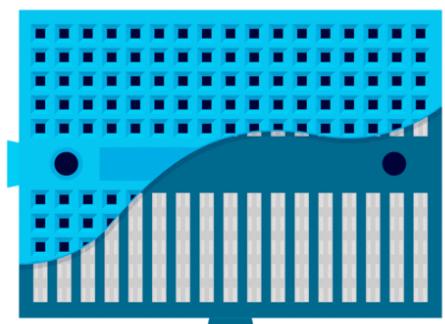
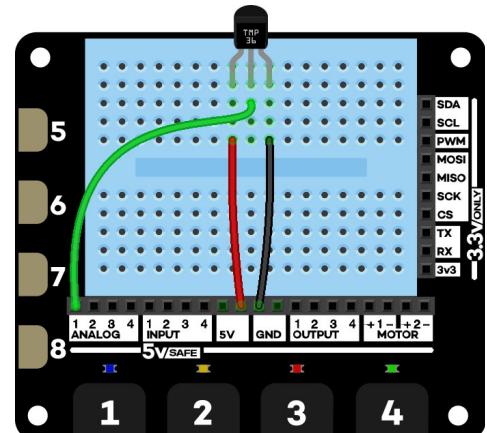
Arbeitsanregung:

Mit dem analogen Eingang kann man z. B. ein Thermometer auslesen. Starte dazu Minecraft und ein Terminal-Fenster. Starte den Editor mit „nano temp.py“ und tippe dann das folgende Programm in Python ab:

```
import explorerhat as eh
import explorercraft as ec
import time
```

```
X = 0
Y = 20
Z = 0
MAX_TEMP = 30
HEIGHT = 30
```

```
therm = ec.Thermometer(X,Y,Z,HEIGHT,MAX_TEMP,ec.WOOL_PURPLE)
therm.teleport_player()
while True:
    temp = eh.analog.one.read() * 100 - 50
    therm.update(temp) # Update our thermometer
    time.sleep(0.1)
```



STATION NR. 12

EIN SELBST- FAHRENDES AUTO BAUEN



► STATION NR. 12: SELBSTFAHRENDES AUTO

An dieser letzten (und schwersten) Station gibt es keine ausführliche Anleitung, nur eine Aufgabe: Setze das selbstfahrende Auto in Bewegung! Wie es funktioniert und wie man es programmiert, erfährst Du im Video-Training im Kapitel 10.6.

Viel Erfolg!

