

COOL CODING MIT OXOSCRIPT

Thomas Garaio



OOCARD

Büren an
der Aare

Aeschlen o. G.

Lauperswil

Niederuzwil

Wilderswil

Spiegel

Heimberg

Frutigen

Bern

Fribourg

Menzingen

Zürich

Kehrsatz

Rubigen

Adelboden

Konolfingen

Moosseedorf

Münsingen

Trimstein

Urtenen

Wangen an
der Aare

Baar

Brig

Chur

Luzern

Ittigen

Mirchel

Mumpf

Schaffhausen

Willisau

Windisch

Winterthur

Zug

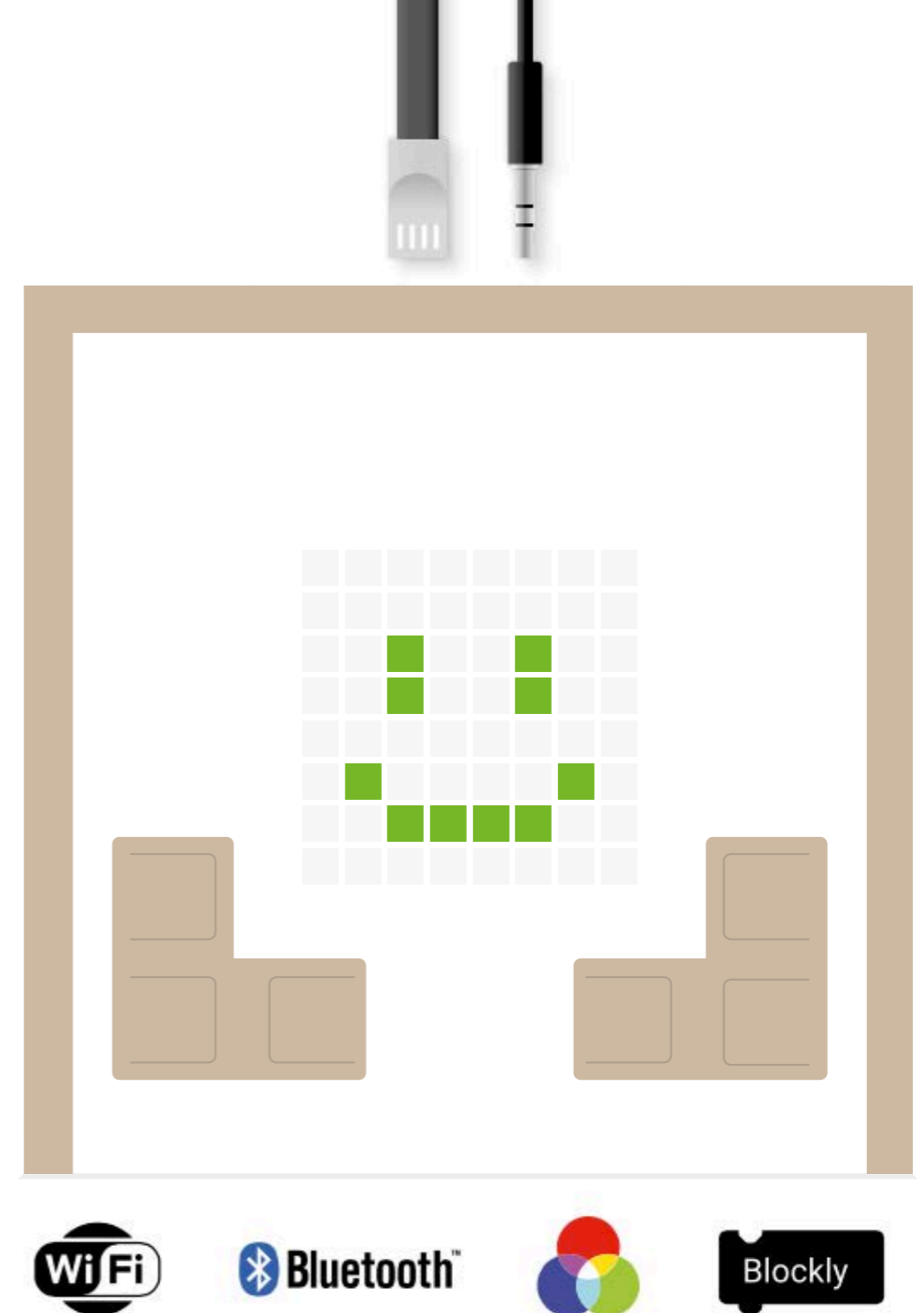
Battenberg

Brüttelen



OXOCARD

Spielend programmieren lernen mit der mehrfach ausgezeichneten Oxocard



worlddidac
A W A R D 2 0 1 8

B E L M A
Best European Learning Materials Awards

18th Worlddidac Award Ceremony

November 2018 Bern Switzerland

Worlddidac
AWARD 2018



BERNEXPO



BERN, SWITZERLAND

worlddidac
AWARD 2018



Gewinner des World Didac Awards 2018 und des Swisscom Innovation Awards 2018



Comenius Award 2019

Spielend Programmieren lernen.

Wie ist ein Computer aufgebaut? Wie wird ein Bild erzeugt?
Wie ist ein Spiel aufgebaut? Wie kommunizieren Geräte untereinander?
Wie kann ich Bewegung erkennen? Wie wird ein Ton erzeugt?
uvm..



Computer



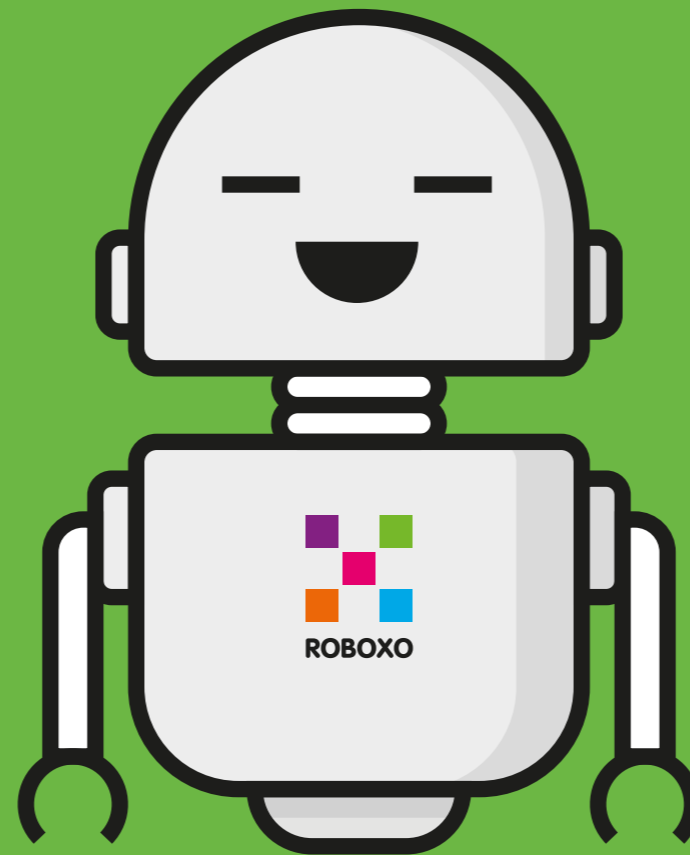
Physik



Algorithmen

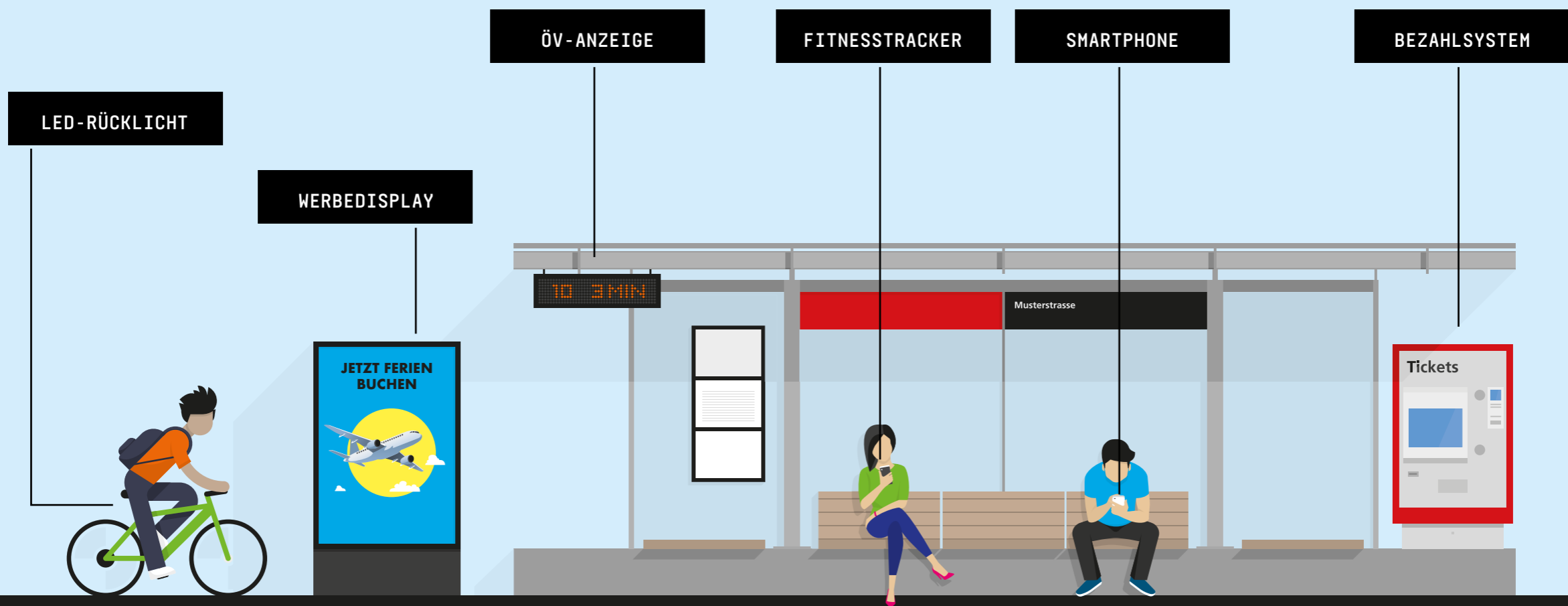


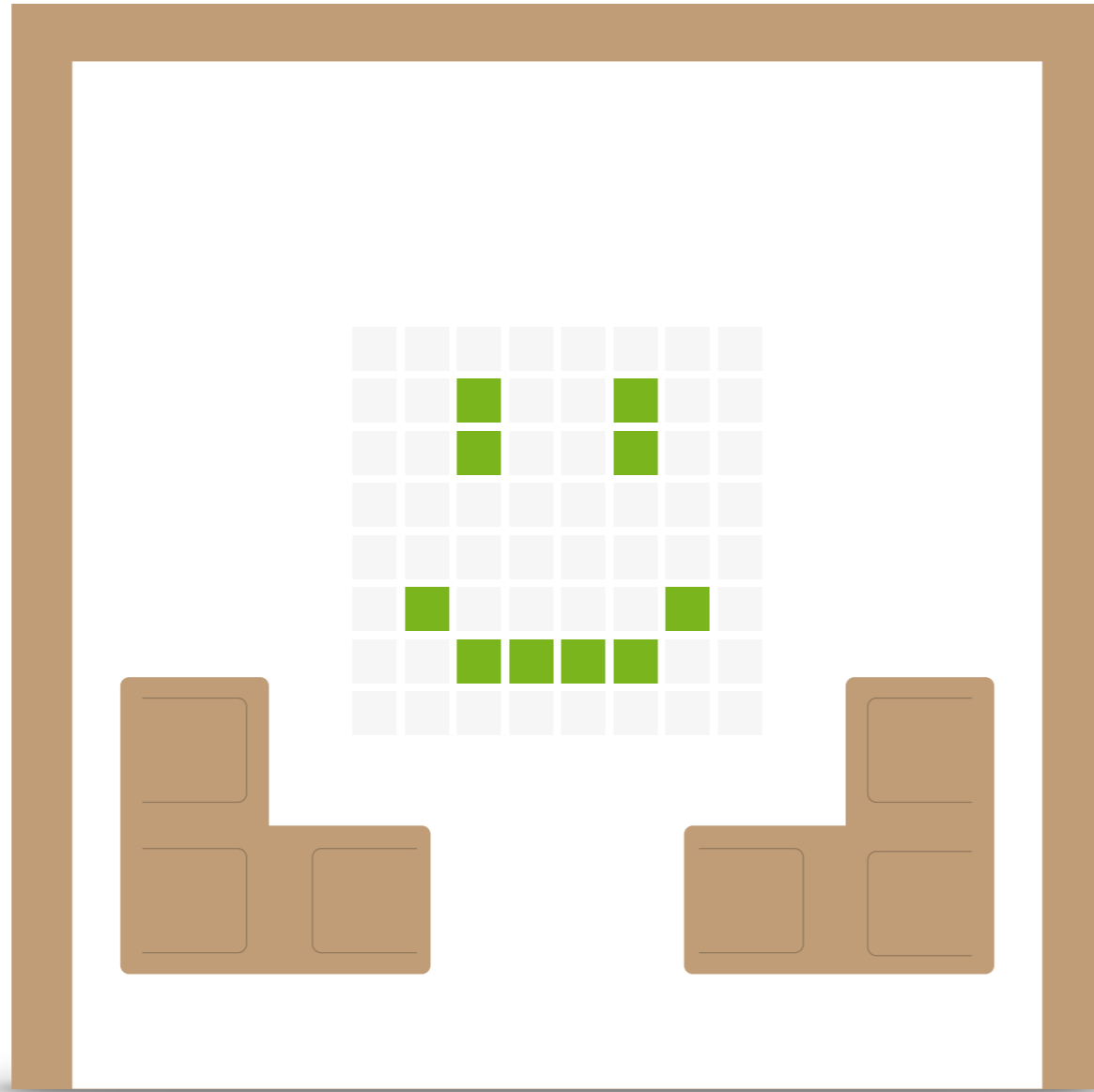
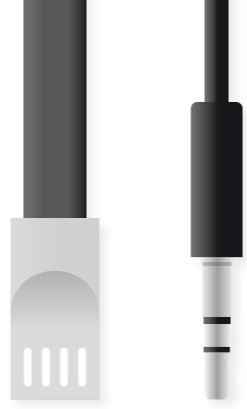
Kreativität



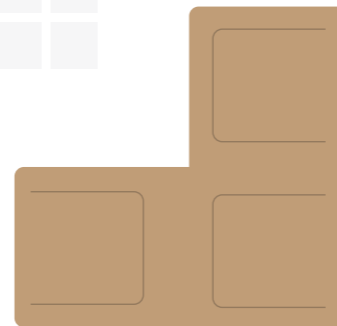
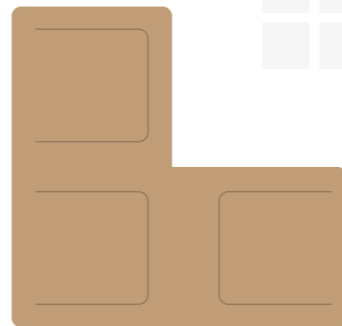
**Programmieren wird so wichtig werden,
wie das Lesen und Schreiben.**

Du verstehst besser, wie die Geräte um uns funktionieren.



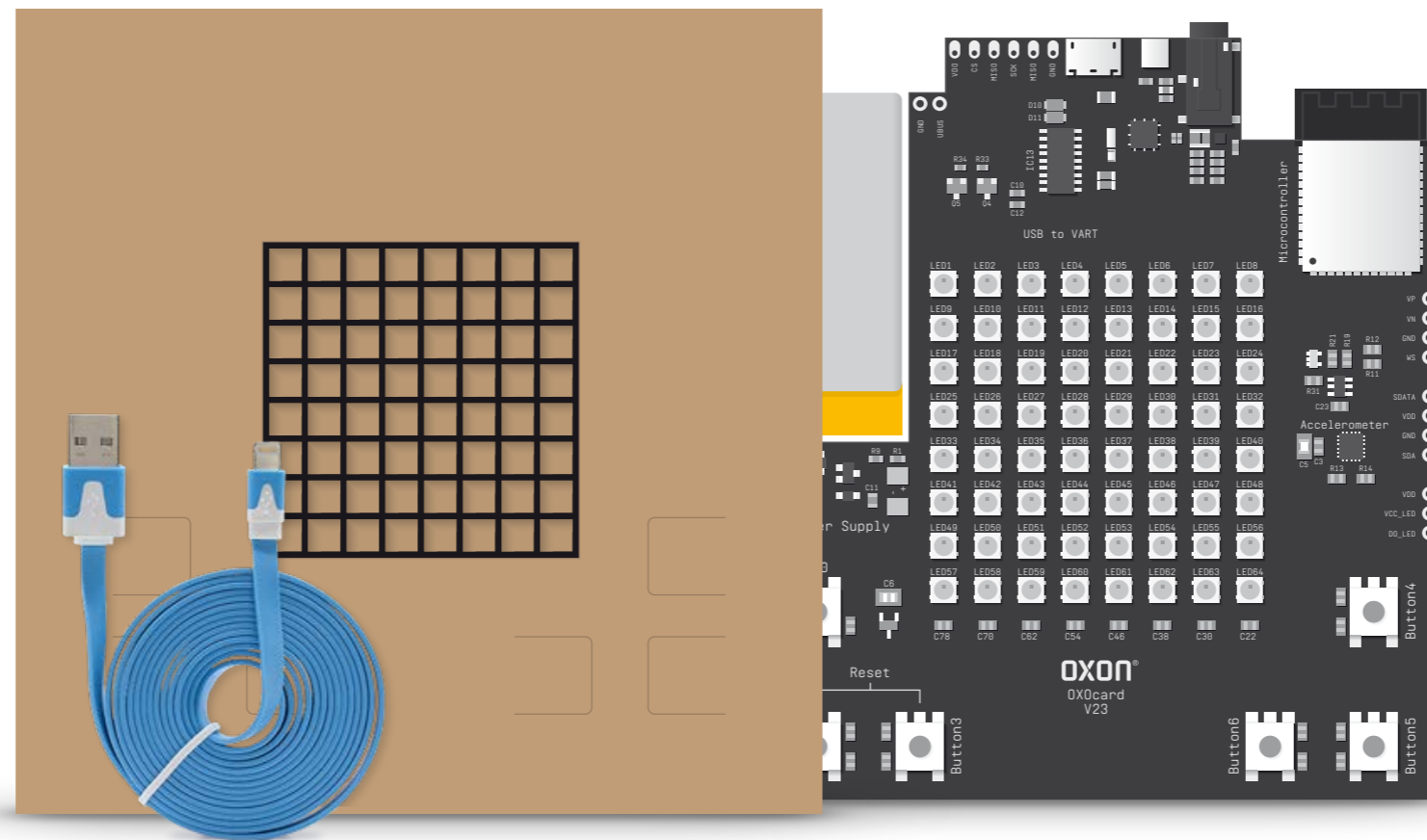


NACHHALTIG
ENTWICKELT



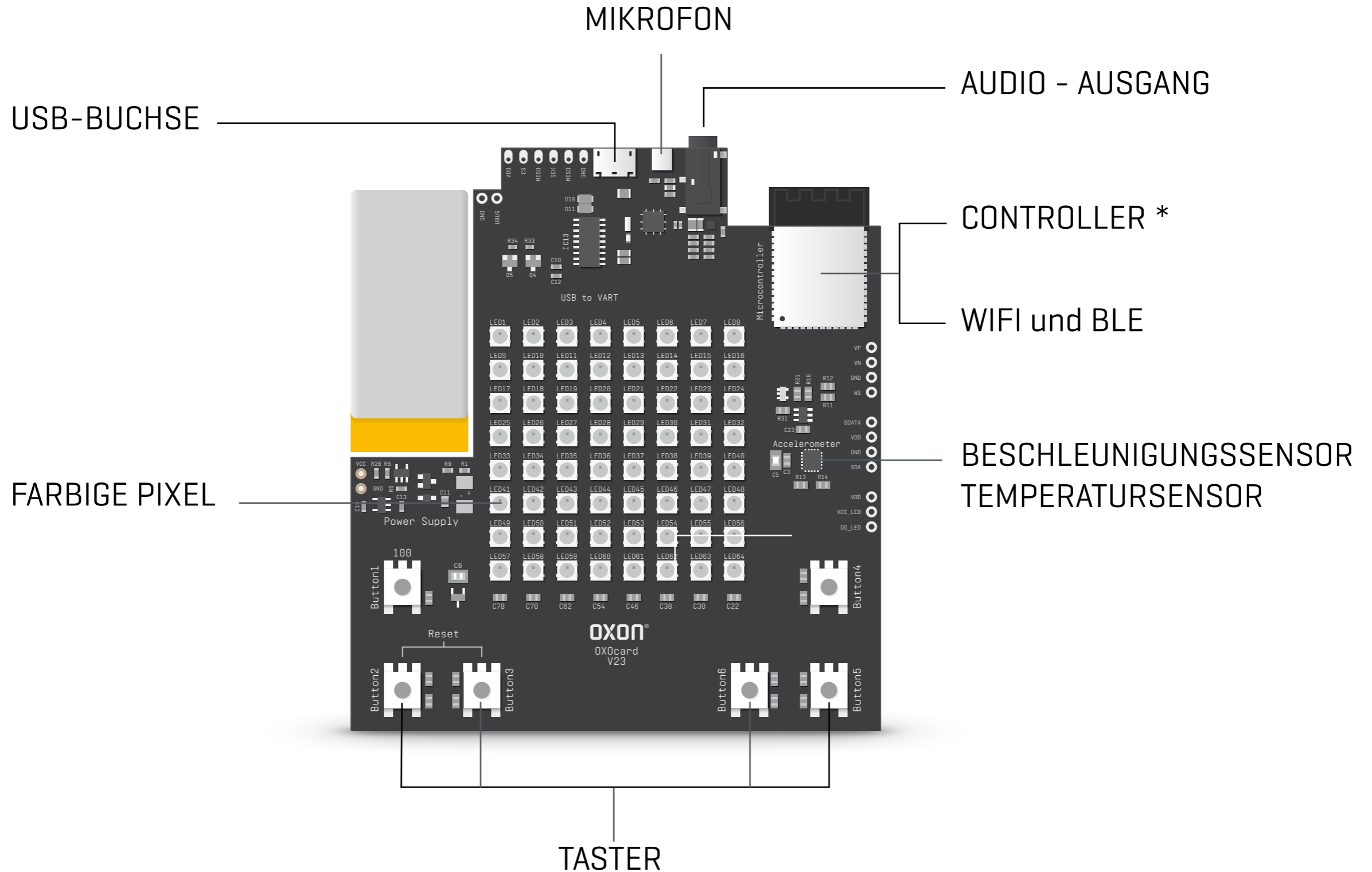
Karton-Gehäuse

- ✓ Rezyklierbar
- ✓ für mehrere Klassen verwendbar*
- ✓ separat erhältlich

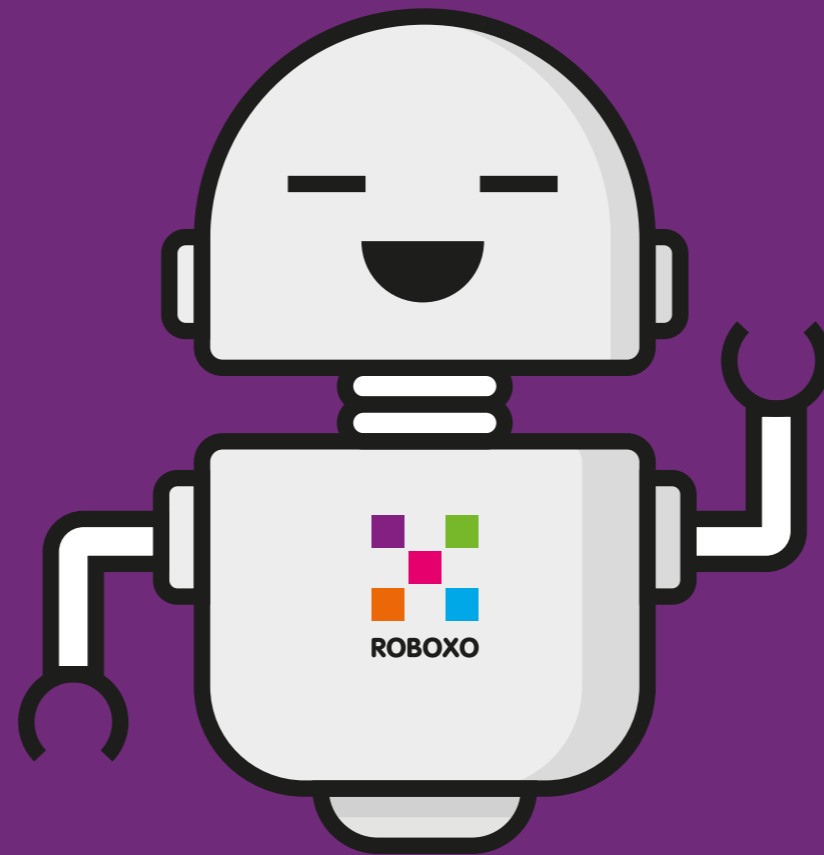


* wie bei einem Leihbuch

ELEKTRONIK

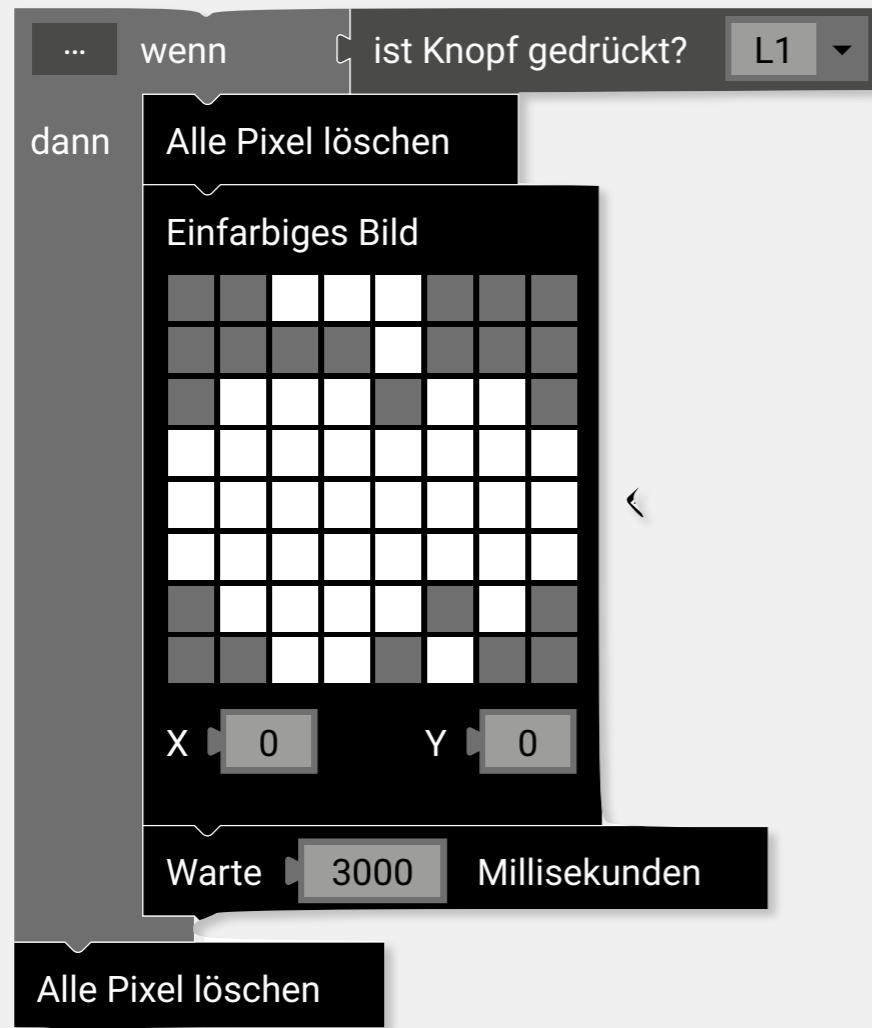


*DUAL-CORE TENSILICA-CHIP-DESIGN, 4 MB FLASH



Programmierumgebungen

Blockly



Python (Tigerjython)

```
from oxocard import *

while True:
    for i in range(8):
        for k in range(8):
            r = randint(0, 50)
            g = randint(0, 50)
            b = randint(0, 50)
            dot(i, k, (r, g, b))
```

Oxoscript

```
def main():
    connect(WIFI)
    startMicrophone()
    while true:
        db = readDecibel()
        if db > 90:
            print("Es ist laut!")

    delay(1000)
```


Blockly

Blockly bietet einen sehr niederschweligen Einstieg und eignet sich daher besonders für Anfänger mit wenig Vorkenntnissen. Die Umgebung ist intuitiv verständlich und kann auch von Lehrpersonen bedient werden, die wenig Informatikgrundwissen mitbringen.

Die SuS arbeiten selbstständig oder in kleinen Gruppen. Als Basis dienen Arbeitsblätter oder das Arbeitsbuch „Achtung, fertig, Code!“ des hep verlags.

- ▶ Einfacher Einstieg
- ▶ Fokussierung auf Algorithmen
- ▶ Spielerische Herangehensweise
- ▶ Sehr fehlertolerant

Python / Tigerjython

Python ist eine einfach zu nutzende, dynamisch typisierte Scripting-Sprache. Sie ist aktuell sehr beliebt an Hochschulen und Universitäten.

Tigerjython besticht durch besonders gute Fehlermeldungen, einem umfangreichen Funktionskatalog und vielen Arbeitsaufgaben.

Das verwendete „Micropython“ ist leider sehr rechner- und speicherintensiv, wodurch die Oxocard am Limit läuft und nicht alle Funktionen zur Verfügung stellen kann [kein FOTA, kein Mikrofon, eingeschränkte Audioausgabe].

- ▶ Fördert exaktes Arbeiten
- ▶ Prozedurale und objektorientierte API's
- ▶ Nahezu beliebig komplexe Aufgaben, bis auf Universitätsniveau

Oxoscript

Oxoscript ist eine für Microcontroller optimierte Scriptingsprache mit Python-Syntax. Das Designziel war, eine pädagogische Programmiersprache zu schaffen, die so einfach wie Python funktioniert, dabei aber so effizient bleibt, damit auch Microcontroller aus Lerncomputern, Haushaltsgeräten oder Steuerungen damit programmiert werden können.

Auf der Oxocard können wir damit die komplette Hardware steuern, wobei via Browserfenster programmiert wird.

- ▶ Moderne Programmierumgebung mit Autocomplete etc.
- ▶ Kompletter Hardwaresupport + FOTA
- ▶ Vereinfachte Sprache, die sich stark an Python orientiert.



OXOSCRIPT

DEMO





Das Programm wurde kompiliert und kann nun auf die Karte geladen werden


```
1 def main():
2     connect(WIFI)
3
4     drawText("connected", false)
5     delay(500)
6
7     setColor(0, 150, 40)
8     drawText("Hello World", false)
9
10    # this prints the Hello World in the output screen of your browser
11
12    print("Hello World from the Oxocard!!")
13
14    delay(2000)
15
16    disconnect()
17
```



```
12     print("Hello World from the Oxocard!!")
13
14     delay(2000)
15
16     disconnect()
17
```

OUTPUT



 Clear

```
15:55:09.802 Hello World from the Oxocard!!
```



Neu: die Karte kommuniziert direkt mit dem Browser.
Hierzu wird eine sichere Kommunikation über einen Server
aufgebaut.

Kaleidoskop

```
def main():  
    disableAutoUpdate()  
    while true:  
        x = random(0,3)  
        y = random(0,3)  
        setColor(0, 0, random(1,10)*25)  
  
        drawPixel(x, y)  
        drawPixel(x, 7 - y)  
        drawPixel(7 - x, y)  
        drawPixel(7 - x, 7 - y)  
  
        update()  
  
        delay(50)
```

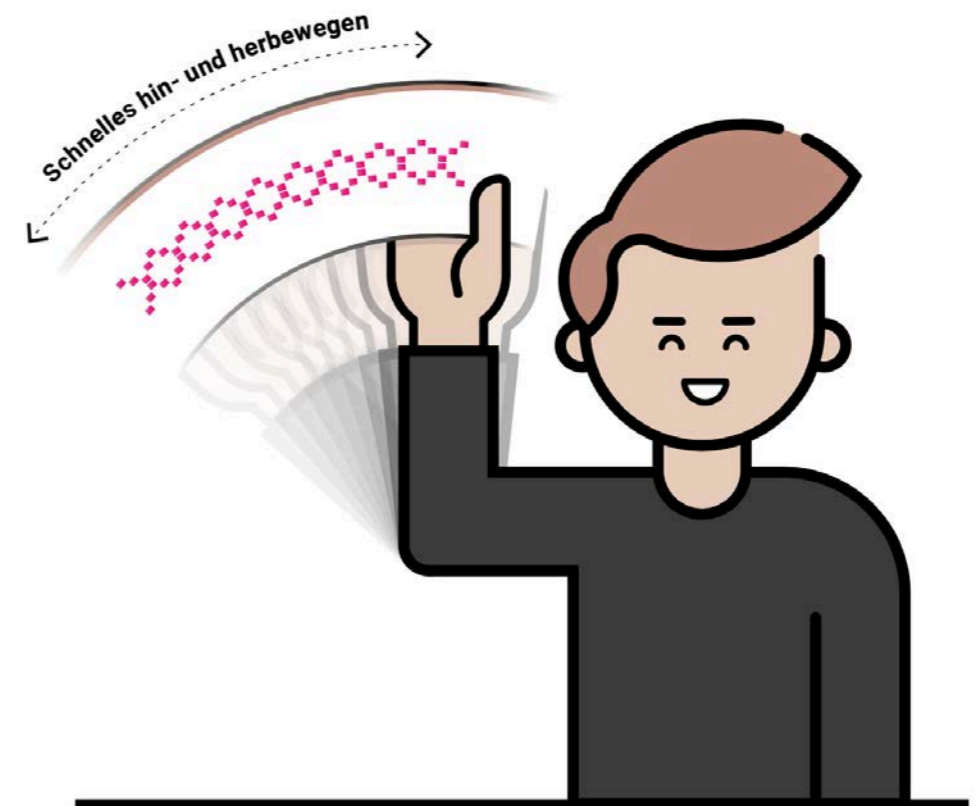


POV-Display

```
# this is a persistence of vision example (POV)
# - switch to a darken room
# - then move the cards very fast from left to right.
# you will see a nice wave painted in the air!!!
```

```
def main():
    disableAutoUpdate()
    y_list = [0,1,2,3,4,4,3,2,1,0]

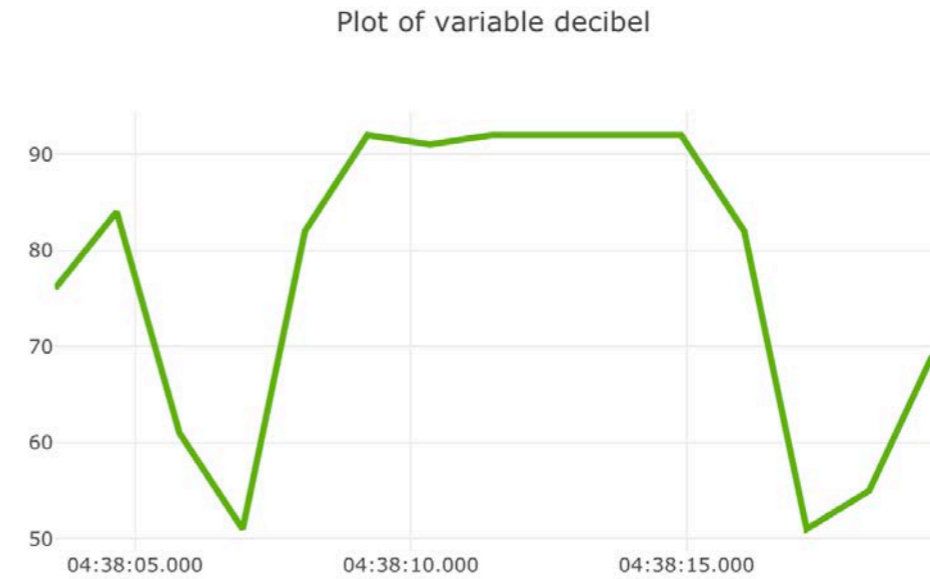
    while(1):
        for y in y_list:
            clearScreen()
            setColor(255,204,51)
            drawPixel(0,y)
            setColor(255,154,31)
            drawPixel(0,y+1)
            setColor(255,124,5)
            drawPixel(0,y+2)
            update()
```



Dezibel-Sensor

```
def main():  
    connect(WIFI)  
    drawText("OK", false)  
  
    startMicrophone()  
  
    while true:  
        db = 0  
        repeat 10:  
            db = db + readDecibel()  
        db = db / 10  
  
        pushValue("decibel", db)  
  
        delay(1000)
```

01




02

Variable "decibel"

| | |
|--------------|----|
| 16:38:03.548 | 76 |
| 16:38:04.669 | 84 |
| 16:38:05.807 | 61 |
| 16:38:06.953 | 51 |
| 16:38:08.084 | 82 |
| 16:38:09.217 | 92 |
| 16:38:10.347 | 91 |
| 16:38:11.476 | 92 |
| 16:38:12.612 | 92 |
| 16:38:13.749 | 92 |
| 16:38:14.902 | 92 |
| 16:38:16.046 | 82 |
| 16:38:17.179 | 51 |
| 16:38:18.312 | 55 |
| 16:38:19.453 | 69 |

03

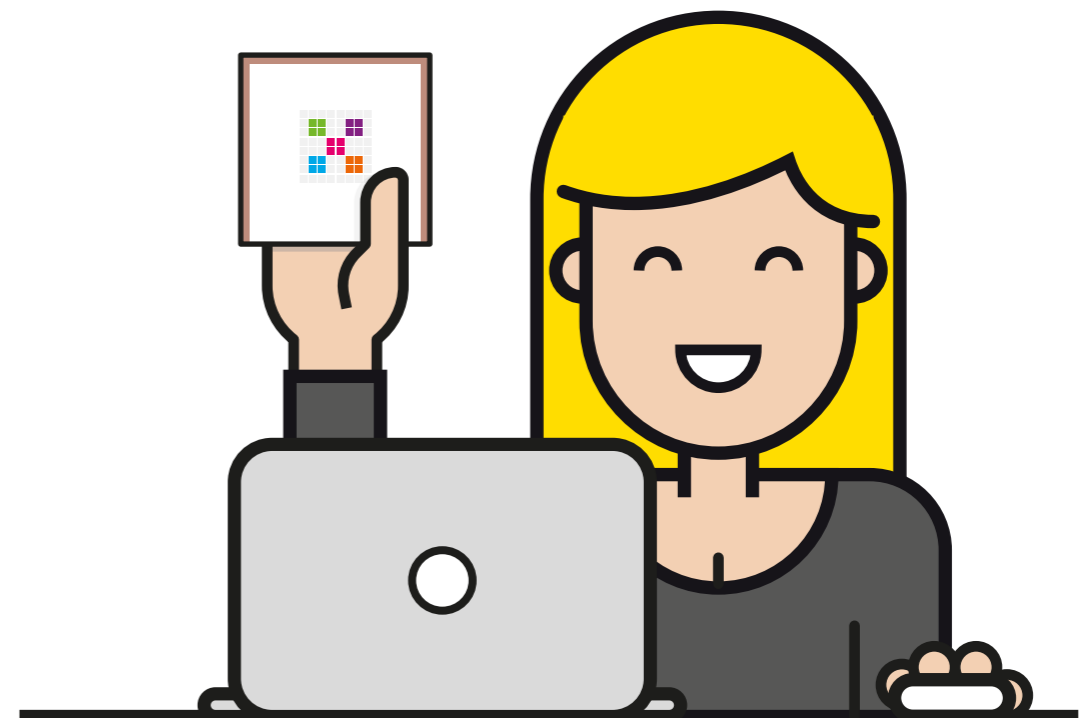
Features der Entwicklungsumgebung

- * Programmierung im Browser
- * Code-Completion
- * Codeprüfung im Hintergrund
- * Keine Installation
- * Mehrsprachig
- * Umfangreiche kostenlose Beispiele
- * Ausgabe von Meldungen
- * Daten speichern im Browser
- * Graphendarstellung
- * Sichere Kommunikation
- * Kein Login
- * Keine persönl. Datenspeicherung
- * Firmware-over-the-air
- * In der Schweiz entwickelt 



Features der Sprache „Oxoscript“

- * Einrückung und Grundsyntax wie bei Python!
- * Ganzzahlen von -32768 bis +32768 und Zeichen
- * Dynamische Listen und Zeichenketten
- * Grundrechen- und Vergleichsoperationen
- * Support für dezimal, binär, hex und ASCII
- * Komplexe Ausdrücke
- * Listenarithmetik: ergänzen, löschen, verbinden
- * Vergleichen von Listen und Zeichenketten
- * Gängige String-Operationen (Kombination mit Zahlen)
- * Globale und (funktions-)lokale Variablen
- * if/else, while, repeat
- * Deklaration von Funktionen („def“)
- * Schneller Single-Pass-Compiler
- * Effizienter Memory-Manager
- * Sehr schnell Ausführungsgeschwindigkeit



Features der Funktionsbibliothek

- * Matrix: Ausgabe von Text/Zahlen, Pixel lesen/schreiben, Farbkreise, Linie, Rechtecke, Kreise, Dreiecke zeichnen, Farbwahl im mittels RGB oder HUE [Farbkreis], Buffern des Bildspeichers, Ausgabe von Bildern
- * Math: Sinus, Cosinus, Min, Max, Avg, Random
- * System: Ausschalten, Aufwachgrund, Batteriestand, an USB abgeschlossen? Automatisches Ausschalten ein/ausschalten, Script runterladen, Hotspot-Modus, Pairing-Modus
- * Sensoren: Beschleunigung/Erschütterung x/y/z, Winkelmessung [mit Referenzvektor], Temperatur, Lärmpegel in Dezibel, Frequenzmessung
- * Tasten: Welche Taste ist gedrückt, Welche Tasten starten die Oxocard, bestimmen, mit welcher Taste die Oxocard gestartet wurde.
- * Zeit: Warten, Schlafen für n Sekunden [Timer], Zeit seit Start, aktuelle Zeit, Datum/Zeit setzen
- * Ton: Ton nach Frequenz oder Noten spielen, Oktave wählen, Wellenformen: Sinus, Rechteck, Sägezahn; Lautstärke lesen/setzen, Sample aufnehmen/abspielen, Interne Samples abspielen.
- * Kommunikation: Ausgabe von Text im Output-Fenster, Senden von numerischen Werten einzeln oder in einer Liste, Liste auf dem Server löschen, Liste runterladen
- * Hardware: Konfiguration, sowie setzen und lesen der freien Pins

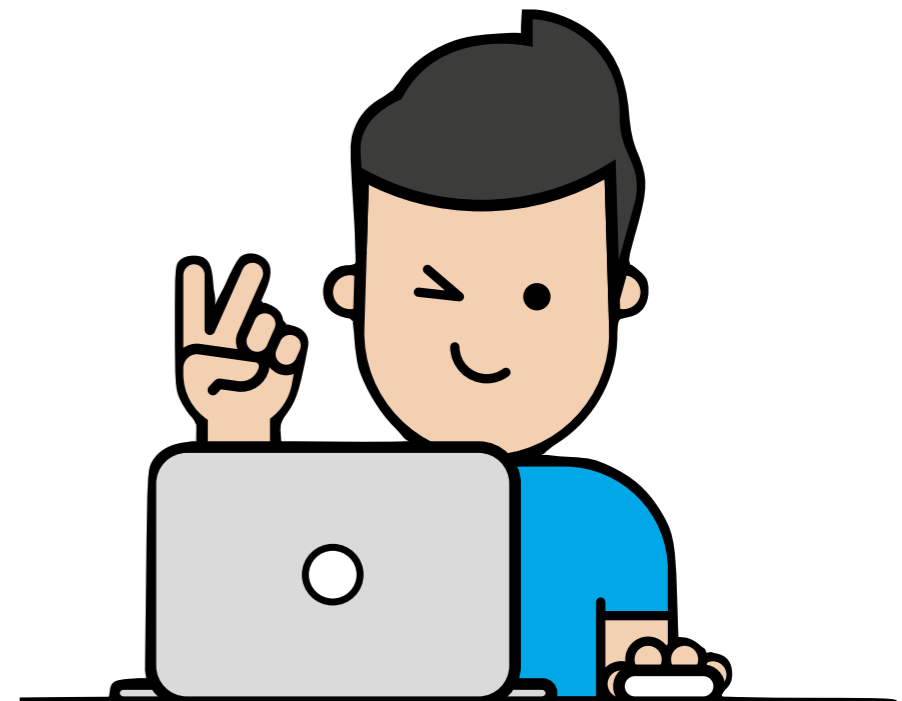


Aktuell in Arbeit

- * Internet-Kommunikationsprotokolle (MQTT, HTTP)
- * Kommunikation zwischen Oxocards
- * Laufzeitfehler-Management
- * Dokumentation und Lehrmittel
- * Arbeitsblätter

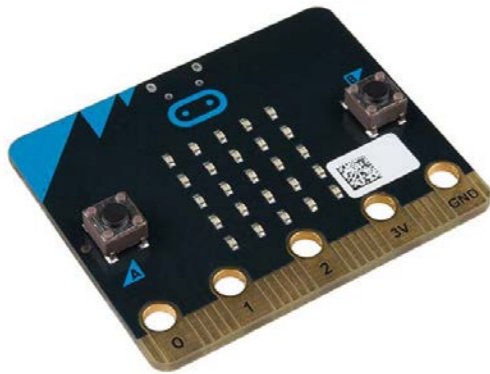
Wünsche oder weitere Idee?

E-Mail an Thomas Garaio (tg@oxon.ch)



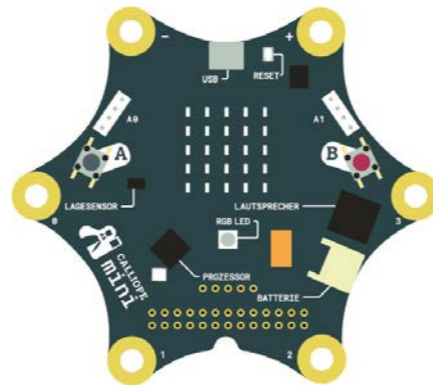
Positionierung

Micro:bit 29.90 CHF
England
2015



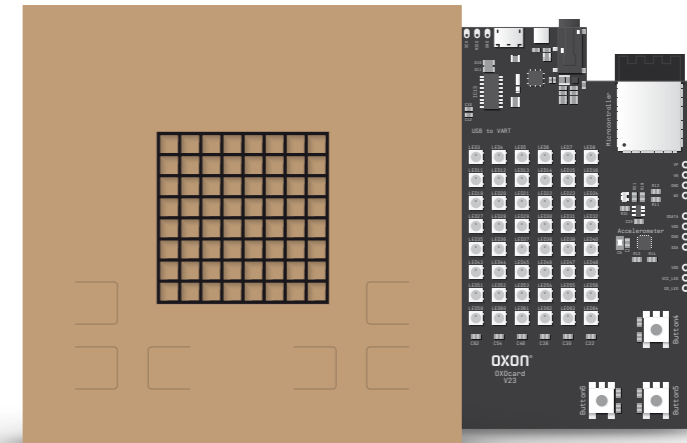
5x5 rote Pixel
Ohne Batterie
Ohne Zubehör
Bluetooth

Calliope 42.00 CHF
Deutschland
2017



5x5 rote Pixel
Ohne Batterie
Ohne Zubehör
Bluetooth

Oxocard 49.00 CHF
Schweiz
2018



8x8 FARBIGE Pixel
Mit Batterie
Inkl. Zubehör
Gehäuse
Bluetooth
WiFi

...

**Internet
der Dinge**

Zusammenfassung

Oxoscript ist eine für Kleinstrecher [MCU's] optimierte Programmiersprache, die geringen Ressourcenverbrauch, hohe Ausführungsgeschwindigkeit und einfache, pythonorientierte Syntax vereint.

Oxoscript erfüllt zwei Zwecke:

1. Sie dient als **pädagogische** Programmiersprache zur Programmierung der Oxocard.
2. Sie dient als **Scripting-Sprache für** unsere **IoT-Produkte** (Buttons, Sensoren)

Gegenüber Blockly können die Oxocards nun mit einer leistungsfähigen textbasierten Programmiersprache codiert werden, die u.a. dynamische Datenstrukturen, lokale Variablen, rekursive Funktionen und ein umfangreiches Set an Basisfunktionen bereitstellt.

Gegenüber Python bietet Oxoscript eine für Einsteiger und kleine Projekte genügend grosse Mächtigkeit, eine leistungsfähige IDE, hohe Ausführungsgeschwindigkeit, „Firmware-Over-the-Air“-Updates und einer kompletten Abdeckung der Hardwarefunktionen.

**BETA
PREVIEW**

JETZT TESTEN



editor.oxocard.ch

Infos

Promo-Code:

PH-Promo19

20% auf Einzelset / 15% auf Lehrmittel
[für Lehrerexemplare]

Beschaffung via Hasler-Stiftung:

Bedarf per E-Mail an tg@oxon.ch (Thomas Garaio)

Offerte / Rechnung weiterleiten an die Stiftung



**Vielen Dank für
Ihre Aufmerksamkeit**

WWW.OXOCARD.CH

OXON[®]
♥ DESIGNED
WITH PASSION

