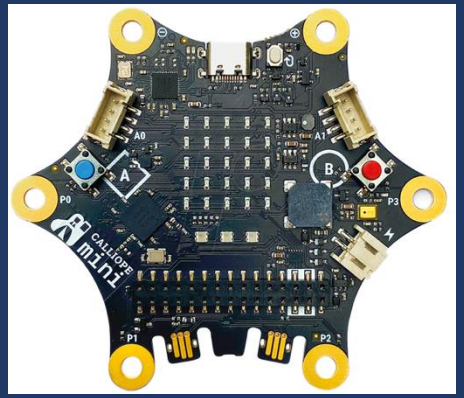


Programmierung des Calliope mini mit Python am Beispiel von SmartHome

Dr. Judith Boine – XLAB Göttingen
j.boine@xlab-goettingen.de

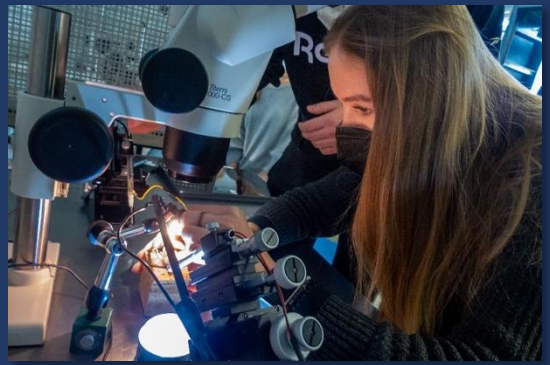
25 Jahre XLAB-Laborgebäude
www.xlab-goettingen.de

25 Jahre Schule trifft Forschung

Begeisterung für MINT

Experimentieren und Programmieren

Aktuelle Forschung im Fokus

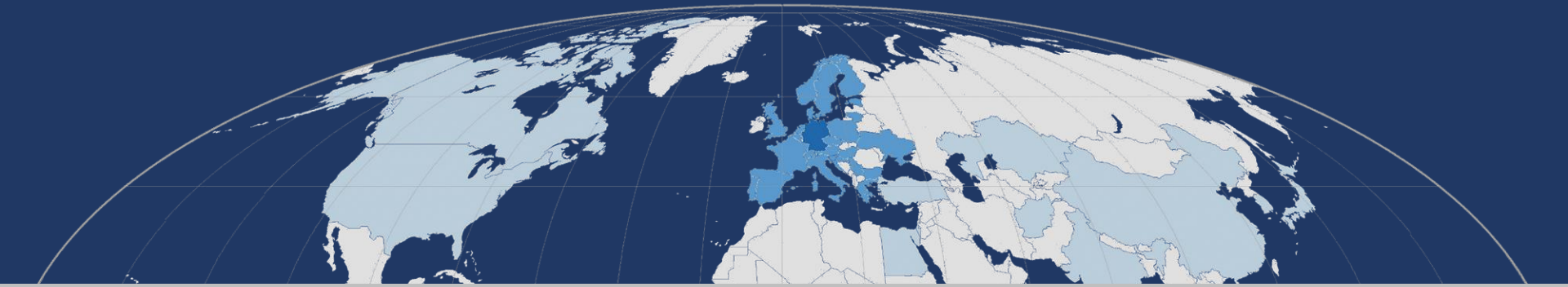


XLAB Göttinger Experimentallabor für junge Leute

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN IN PUBLICA COMMODA SEIT 1737

Reichweite: 10.000 Teilnehmende/Jahr

Physik, Chemie, Biologie, Informatik/Mathematik



Bundesländer

Niedersachsen	46 %
Übrige Bundesländer	42 %
Ausland	12 %

Jahrgänge

Jg. 11-13	75 %
Jg. U9-10	25 %

XLAB Göttinger Experimentallabor für junge Leute

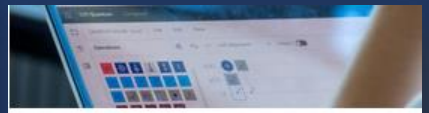
GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

Kurse für Schülergruppen

- **Klassen 8-13, Schwerpunkt Sek. II**
- **Mehr als 200 Experimente**
- **1 Tag (9-17 Uhr)**
- **BNE-, KC-Bezug**
- **Onlineangebot**
- **Individuelle Absprachen**

Einstieg in Python & Künstliche Intelligenz

Wie gelingt es einem Computer, handgeschriebene Zahlen zu erkennen? Bei der Bilderkennung kommt Künstliche Intelligenz zum Einsatz. Mit ...



Quanteninformatik

Wie werden Daten in der Quanteninformatik codiert und verarbeitet? Dieser Kurs vermittelt die Grundlagen der Quanteninformatik. Er zeigt, ...

Wellenphysik mit Laserlicht

Wie lässt sich durch Beugung und Interferenz die Wellennatur von Licht nachweisen? Wie können Interferometer zur Messung kleiner ...

Grenzwerte in der Mathematik

Neugierig auf Oberstufen-Mathematik? Das Konzept der Grenzwerte wird mit interaktiven Experimenten in Kleingruppen erarbeitet und das ...

XLAB Göttinger Experimentallabor für junge Leute

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

Angebote für Einzelteilnehmende

Forscherferien
- **ab 10 Jahre in 3 Altersgruppen**
- **1 Woche in den Schulferien**

Science Camps
- **ab 16 Jahre**
- **1-3 Wochen in den Schulferien**

20.-24.10.2025 Girls only! Forscherferien „Mein schönes smartes Zuhause“

Wie sieht das Zuhause der Zukunft aus und wie kann man es selbst gestalten? In der Projektwoche entdecken Schülerinnen, wie moderne ...

20.-23.10.2025 Polymere aus natürlichen und synthetischen Bausteinen

Nachhaltigkeit und Kunststoffe: wie geht das? Wie nutzen Chemiker*innen natürliche und synthetische Bausteine für die Herstellung von ...

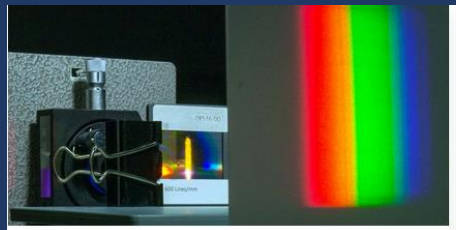
Ausgebucht!! 20.-24.10.2025 Physik realer Systeme

Wie lassen sich reale Systeme mit physikalischen Grundgleichungen beschreiben? Wie helfen Simulationen dabei, das Verhalten dieser Systeme ...

XLAB Göttinger Experimentallabor für junge Leute

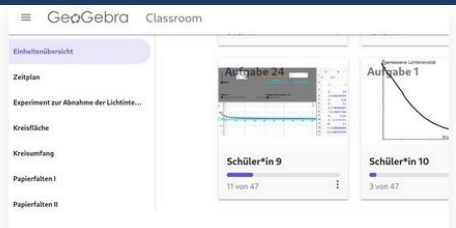
GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN IN PUBLICA COMMODA SEIT 1737

Aktuelle Lehrerfortbildungen – Präsenz oder online



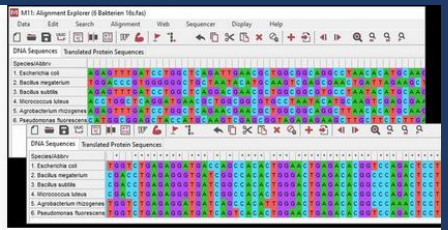
04.11.2025 Atomphysik am Lasermedium – online

Sie möchten die Grundlagen der Atomphysik an einem für Schüler*innen attraktiven System erarbeiten? In dieser Fortbildung erfahren Sie, wie ...



10. und 17.11.2025 Applets in GeoGebra Classroom – online

Der Einsatz von Applets innerhalb einer GeoGebra-Classroom-Umgebung bietet die Möglichkeit, den Funktionsumfang von GeoGebra für die ...

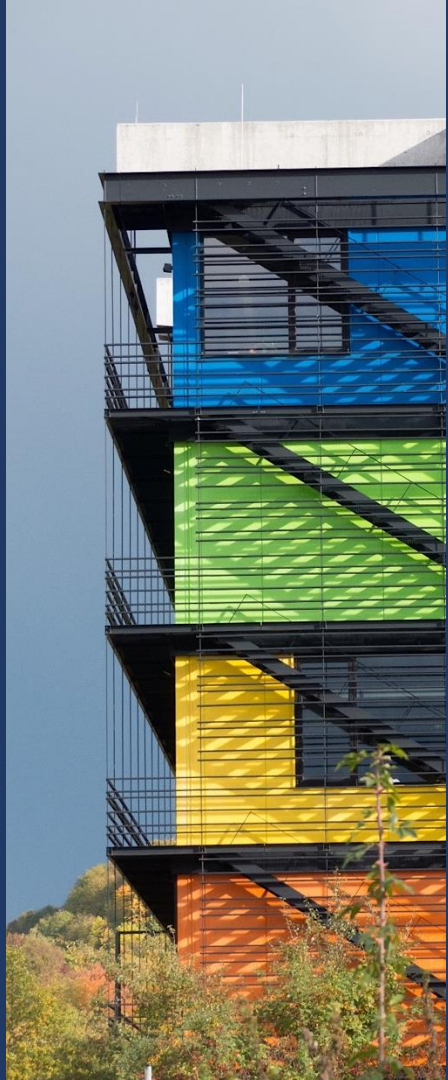


13.11.2025 DNA und Evolution – DNA-Daten auswerten und evolutionäre Zusammenhänge verstehen

Wie lassen sich genetische Daten nutzen, um evolutionäre Verwandtschaftsverhältnisse sichtbar zu machen? Diese Fortbildung bietet ...

25.11.2025 Künstliche Intelligenz in Schule und Unterricht – online

Wie funktioniert KI, was sind neuronale Netze und was sind die Chancen und Risiken von KI in der Schule. Welche Tools können für die ...



Ablauf 1. Teil

- Vorstellung Python-Editor
- Vorstellung Programmierung (intern):
  - LED-Matrix
  - RGB-LEDs
  - Taster
  - Sensoren
- Vorstellung Programmierung (extern):
  - LED-Lichtleiste, LED-Matrix, Ampel
  - Ultraschalsensor, Lichtsensor
  - Temperatur- und Feuchtigkeitssensor (BME680)
  - Servo-Motoren, Gleichstrommotoren
  - 4-Ziffern-Display
  - Drucksensor, Taster, Tastermodul

Ablauf 2. Teil

Jeder bekommt einen Calliope und kann verschiedene Sensoren und Aktoren selber in Python programmieren

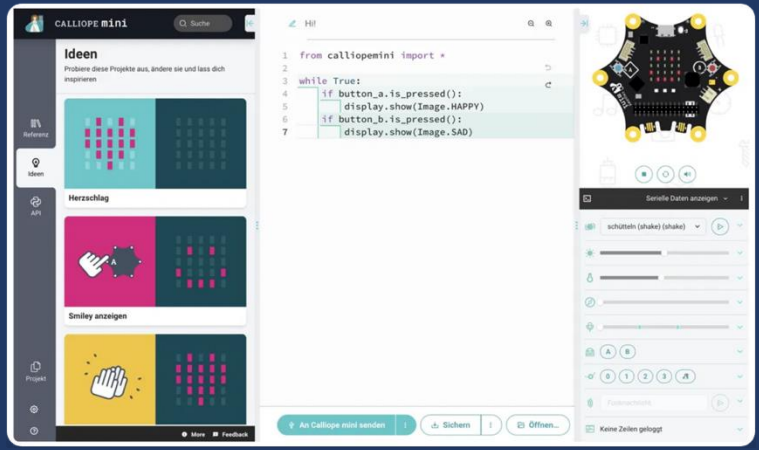
XLAB Göttinger Experimentallabor für junge Leute

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN IN PUBLICA COMMODA SEIT 1737

Der Python Editor

Der **Python Editor** für den Calliope mini ist eine webbasierte Entwicklungsumgebung, die speziell für die Programmierung des Mikrocontrollers konzipiert wurde.

- **Webbasiert:** Erreichbar unter **python.calliope.cc**, keine Installation notwendig
- **Code-Editor:** Syntax-Highlighting, Autovervollständigung und Fehlerprüfung
- **Simulator:** Testen von Programmen ohne physischen Calliope mini
- **API-Dokumentation:** Integrierte Hilfe zu allen verfügbaren Funktionen
- **Direktes Übertragen:** Programme können direkt auf den Calliope mini übertragen werden



XLAB Göttinger Experimentallabor für junge Leute

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

Grundstruktur eines Python-Programms

Python-Programme für den Calliope mini folgen einer einfachen Grundstruktur:

1. Imports

Am Anfang des Programms werden die benötigten Module importiert.

```
from calliopemini import *
```

2. Hauptschleife

Die meisten Programme laufen in einer Endlosschleife, um kontinuierlich zu arbeiten.

```
while True: # Dein Code hier
```

3. Funktionen und APIs

Innerhalb der Hauptschleife werden Funktionen aus den importierten Modulen verwendet, um mit der Hardware zu interagieren.

```
display.show(Image.HEART)
sleep(1000) # Pause für 1 Sekunde
display.scroll('Hallo!')
```

```
1 from calliopemini import *
2
3 while True:
4     if button_a.is_pressed():
5         display.show(Image.HAPPY)
6     if button_b.is_pressed():
7         display.show(Image.SAD)
```

```
from calliopemini import *

# Endlosschleife
while True:
    display.show(Image.HEART)
    sleep(1000)
    display.scroll("Hello!")
```

XLAB Göttinger Experimentallabor für junge Leute

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN IN PUBLICA COMMODA SEIT 1737

Programmierung der LED-Matrix

Die 5x5 LED-Matrix ist eines der zentralen Elemente des Calliope mini und kann auf verschiedene Arten programmiert warden (Befehle unter Referenz – Grundlagen):

Vordefinierte Bilder oder Text anzeigen:

```
display.show(Image.HEART)
```

Text scrollen lassen:

```
display.scroll('Hallo Welt!')
```

Eigene Muster erstellen mit Helligkeitswerten (0-9):

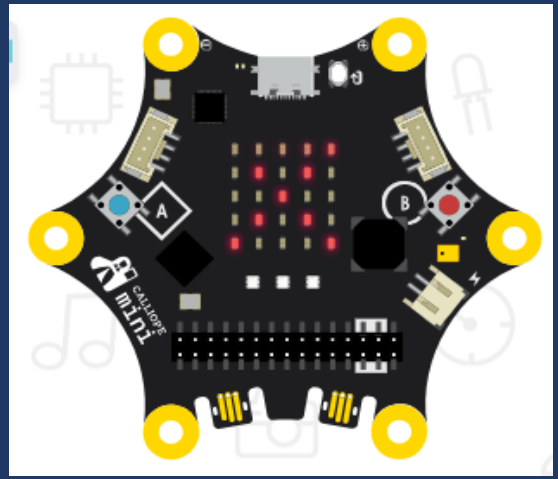
```
display.show(Image( '90009:' '09090:' '00900:' '09090:' '90009'))
```

Einzelne LEDs steuern:

```
display.set_pixel(2, 2, 9) # x, y, Helligkeit
```

Display löschen:

```
display.clear()
```



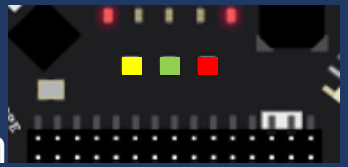

```
# Imports go at the top
from calliopemini import *


while True:
    display.show(Image( '123579:' '09090:' '00900:' '09090:' '90009'))

```

XLAB Göttinger Experimentallabor für junge Leute

UNIVERSITÄT

RGB-LEDs Calliope mini

1. Modul neopixel importieren
2. Anzahl der LEDs festlegen
3. Pin festlegen
4. Befehle unter: Referenz – RGB-LED



```
while True:
    np.fill((255,0,0))
    np.show()
    sleep(500)
    np.clear()
    sleep(500)
```

Blinken (Display ausschalten)

```
from calliopemini import*
import neopixel

n = 3
np = neopixel.NeoPixel(pin_RGB, n)

np[0] = (255, 0, 0) # erste RGB-LED auf rot, vo
np[1] = (0, 128, 0) # zweite  RGB-LED auf grün,
np[2] = (0, 0, 64)  # dritte RGB-LED auf blau,
np.show()
```

```
from calliopemini import*
import neopixel

n = 3
np = neopixel.NeoPixel(pin_RGB, n)

np.fill((255,0,0)) # füllt alle Pixel mit der Farbe rot
np.show()
```

Farben (mit ColorPicker o.ä. RGB-Farbe bestimmen)

RGB-Werte von 0-255 für jede Farbe:

- Rot: (255, 0, 0)
- Grün: (0, 255, 0)
- Blau: (0, 0, 255)
- Weiß: (255, 255, 255)

XLAB Göttinger Experimentallabor für junge Leute

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

Module importieren

Es gibt zwei Möglichkeiten, externe Methoden zu benutzen

Import neopixel - läd das Modul

From neopixel import Neopixel - läd die Klasse Neopixel aus dem Modul neopixel

```
# Imports go at the top
from calliopemini import*
import neopixel

np = neopixel.NeoPixel(pin_RGB, 3)


while True:
    np[0] = (255, 0, 0) # erste RGB
    np[1] = (0, 128, 0) # zweite  R
    np[2] = (0, 0, 64)  # dritte RG

    np.show()
```

```
# Imports go at the top
from calliopemini import*
from neopixel import NeoPixel

np = NeoPixel(pin_RGB, 3)


while True:
    np[0] = (255, 0, 0) # erste
    np[1] = (0, 128, 0) # zweit
    np[2] = (0, 0, 64)  # dritt
    np.show()
```

XLAB Göttinger Experimentallabor für junge Leute

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN IN PUBLICA COMMODA SEIT 1737

Die Tasten des Calliope mini

Der Calliope mini verfügt über zwei programmierbare Tasten, A und B, sowie über vier TouchPins, die mit Python gesteuert werden können :
Befehle unter: Referenz - Eingaben

Taste wird gedrückt

Wähle Taste: A

```
while True:
    if button_a.is_pressed():
        display.scroll('A')
```

Touchpins Mehr

Die Pins P0, P1, P2 und P3 können als Berührungssensoren...

Wähle Pin: Pin P0

```
while True:
    if pin0.is_touched():
        display.set_pixel(2,2,9)
        sleep(1500)
        display.clear()
```

XLAB Göttinger Experimentallabor für junge Leute

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

Sensoren nutzen

Der Calliope mini verfügt über verschiedene interne Sensoren, die mit Python einfach ausgelesen werden können:
Befehle unter: Referenz - Eingaben

Temperatursensor:	temperature()
Beschleunigungssensor:	accelerometer.get_x()
Lichtsensor:	read_light_level()
Lautstärkesensor:	microphone.sound_level()
Kompass:	compass.heading()
Gesten:	accelerometer.was_gesture('shake')

Zur Verfügung stehende externe Sensoren und Aktoren

1. Ultraschallsensor (eB, 1,5€-5€)
2. BME 680 Temperatur- und Feuchtigkeitssensor (eB, 15€ - 22€)
3. Externer Taster (P, 2€)
4. Drucksensor (P, 5€ - 15€)
5. Tastermodul (P, 2€)
6. Lichtsensor(P, 2€)

1. 180° Servo-Motor (FT90B) (eB, 5€)
2. 360° Servo-Motor (FT90R) (eB, 5€)
3. Gleichstrommotor mit Propeller (M, 2€ - 3€)
4. 4Ziffern-Display (eB, 3€ - 8€)
5. LED-Lichtleiste (30 LED) (B, 5€)
6. LED-Matrix WS2812 (8x8 LED) (B, 3€- 5€)
7. Ampel (P, 1€ - 5€)

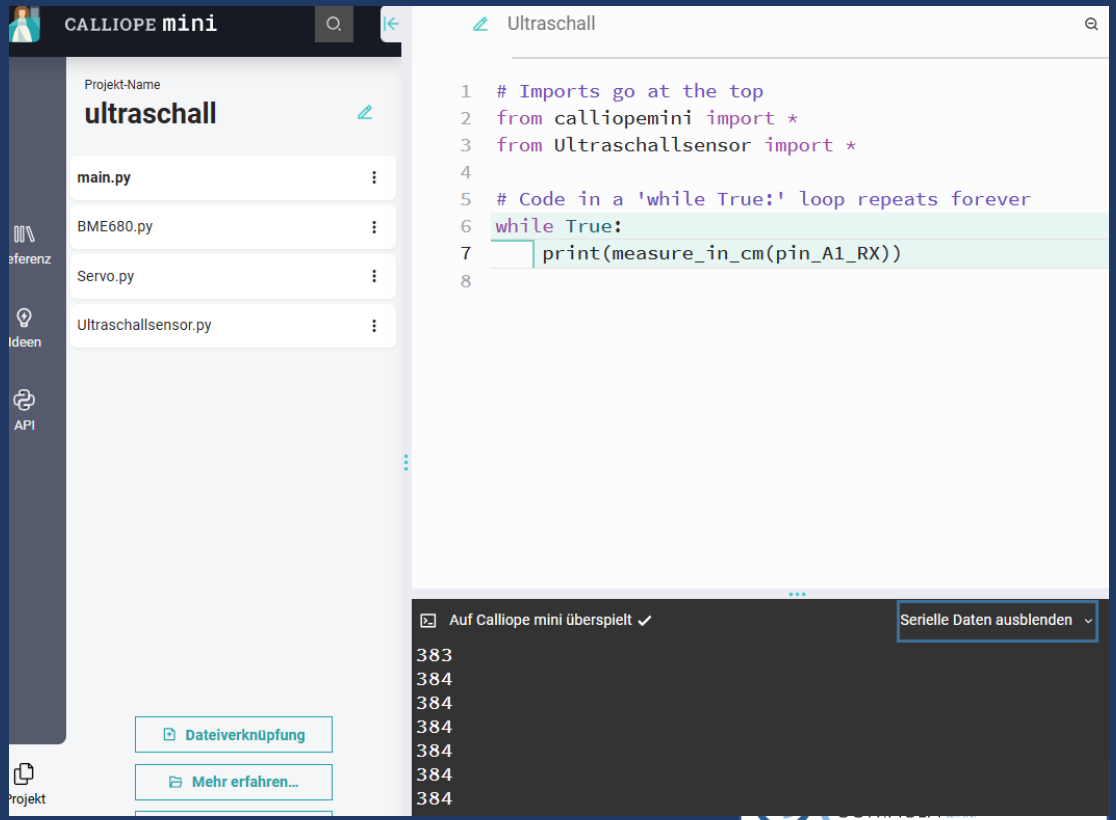
B	Modul importieren
eB	Projekt-Modul hinzufügen – Modul importieren
P	PIN-Befehle unter API – Calliope mini
M	Befehle unter Referenz - Motor

XLAB Göttinger Experimentallabor für junge Leute

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN IN PUBLICA COMMODA SEIT 1737

Ultraschallsensor

1. Anschluss an GROVE A1 oder Pin
2. Unter Projekt Modul Ultraschallsensor auswählen
3. Modul importieren
4. Folgende Methoden sind vorhanden
   measure_in_cm(pin)
   measure_in_inch(pin)



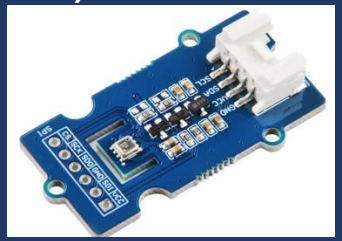
Feuchte-, Temperatur- und Luftdrucksensor

1. Anschluss an GROVE A0
2. Unter Projekt Modul BME680 auswählen
3. Modul importieren
   import BME680 oder
   from BME680 import *
4. Folgende Methoden sind vorhanden
   get_temperature(calib)
   get_pressure(calib)
   get_humidity(calib)

```
3  from  BME680 import *
4
5  chip_id = read_register(0xD0)[0]
6  if chip_id != 0x61:
7      raise RuntimeError("Unerwartete Chip ID!")
8
9  calib = load_calibration()
10 configure_sensor()
11 write_register(0xE0, 0xB6)
12 sleep(10)
13 configure_sensor()
14
15 while True:
16     print("Temperatur:", get_temperature(calib), "°C")
17     print("Luftdruck:", get_pressure(calib), "hPa")
18     print("Luftfeuchtigkeit:", get_humidity(calib), "%")
19     sleep(2000)
```

Auf Calliope mini überspielt ✓ Serielle Daten ausblenden

```
Luftfeuchtigkeit: 0.0 %
Temperatur: 27.3 °C
Luftdruck: 981.34 hPa
Luftfeuchtigkeit: 0.0 %
```



XLAB Göttinger Experimentallabor für junge Leute

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

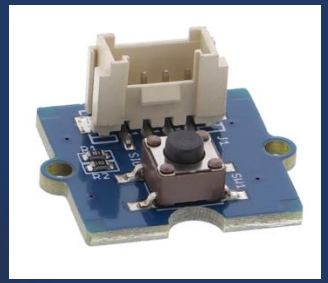
Drucksensor und Taster

1. Anschluss an GROVE A1 oder Pin
2. Abfrage der Sensorwerte über pin_A1_RX
3. Taster liefert digitale Werte 0/1
4. Drucksensor liefert analoge Werte (0-1024)

```
while True:
    display.scroll(pin_A1_RX.read_digital())
```

```
while True:
    display.scroll(pin_A1_RX.read_analog())
```

Befehle unter: API-Calliope mini



XLAB Göttinger Experimentallabor für junge Leute

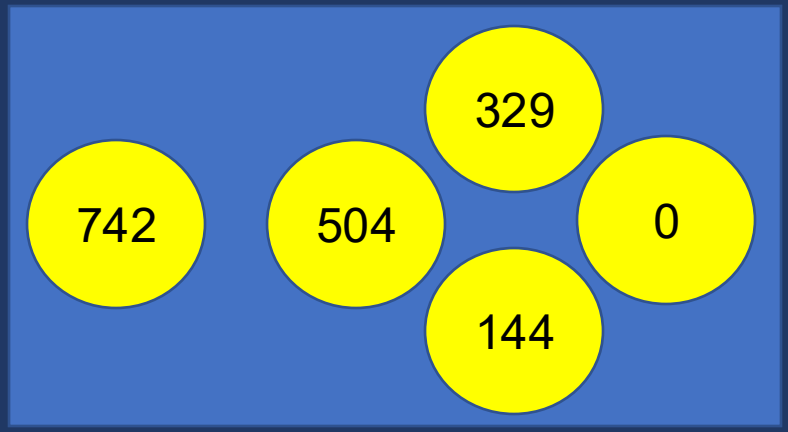
GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

Tastermodul

Anschluss mit Lüsterklemmen:
- Rot von VCC an VCC
- Schwarz von GND an GND
- Gelb von OUT z.B. an P0 (PIN 0)

1. Jede Taste hat einen anderen Wert
2. Alle Werte werden über einen Pin ausgelesen
3. Zu Begin: für jede Taste den Wert bestimmen

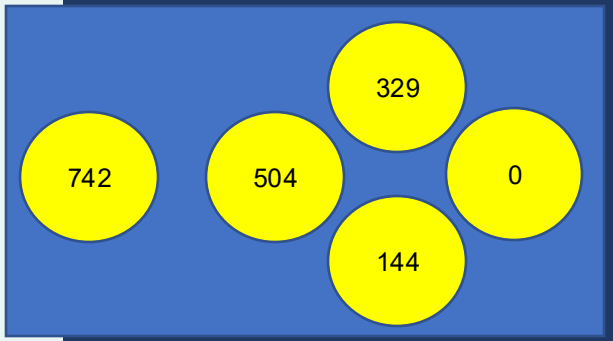
```
display.scroll(pin0.read_analog())
```



XLAB Göttinger Experimentallabor für junge Leute

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

```
while True:
    #display.scroll(pin0.read_analog())
    if pin0.read_analog()>100 and pin0.read_analog()<200:
        display.show(4)
        sleep(100)
    if pin0.read_analog()>300 and pin0.read_analog()<400:
        display.show(2)
        sleep(100)
    if pin0.read_analog() >400 and pin0.read_analog()<600:
        display.show(1)
        sleep(100)
    if pin0.read_analog()<100:
        display.show(3)
        sleep(100)
    if pin0.read_analog()>600 and pin0.read_analog()<800:
        display.show(0)
        sleep(100)
```



XLAB Göttinger Experimentallabor für junge Leute

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

Lichtsensor

1. Anschluss an GROVE A1 oder Pin
2. Abfrage der Sensorwerte über z.B. pin_A1_RX
3. Lichtsensor liefert analoge Werte (0-1024)


```
while True:
    display.scroll(pin_A1_RX.read_analog())
```

XLAB Göttinger Experimentallabor für junge Leute

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

180° & 360°- Servomotor

1. Anschluss an Pin
2. Unter Projekt servo auswählen
3. Alle (*) Funktionen und Klassen des Moduls Servo importieren
   from Servo import *
4. Folgende Methoden sind vorhanden
   set_servo_angle(pin14, 0) für 180°
   set_servo_speed(pin9, 100) für 360°

```
from calliopemini import*
from Servo import *

while True:
    if button_a.is_pressed(): #180° Servo an C14
        set_servo_angle(pin14,0)
        sleep(1000)
        set_servo_angle(pin14, 180)
        sleep(1000)

    if button_b.is_pressed(): #360° Servo an C9
        set_servo_speed(pin9, 100)
        sleep(2000)
        set_servo_speed(pin9, 0)
```

XLAB Göttinger Experimentallabor für junge Leute

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

Gleichstrommotor

Anschluss mit Lüsterklemmen:
- Rot an M0+
- Schwarz an M0-

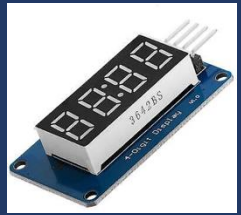
Befehle unter
Referenz - Motoren

```
from calliopemini import *
pin_M_MODE.write_digital(1) # Motoren einschalten
pin_M0_DIR.write_digital(1) # Laufrichtung für M0 festlege

while True:
    pin_M0_SPEED.write_analog(1023) # M0 wird auf volle Ge
    sleep(500)
    pin_M0_SPEED.write_analog(500) # M0 wird auf halbe Ges
    sleep(500)
    pin_M0_SPEED.write_analog(0) # M0 wird auf 0 Geschwind
    sleep(500)
    pin_M0_SPEED.write_analog(500) # M0 wird auf halbe Ges
    sleep(500)
```

XLAB Göttinger Experimentallabor für junge Leute

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

4 Ziffern-Display



1. Anschluss an A1 oder PIN
2. Unter Projekt DigitDisplay auswählen
3. Modul importieren
4. Folgende Methoden sind vorhanden
   digit_display.clear_display()
   digit_display.show_letter_at('2', 1)
   digit_display.show_number(1234)

```
from calliopemini import*
from DigitDisplay import *

# Initialize the 4-digit display
digit_display = TM1637(pin_A1_RX, pin_A1_TX)

# Beispiel zur Verwendung
while True:
    digit_display.clear_display()
    digit_display.show_letter_at('C', 0)
    digit_display.show_letter_at('O', 1)
    digit_display.show_letter_at('2', 2)
    digit_display.show_letter_at('-', 3)
    sleep(1000)
    digit_display.clear_display()
    digit_display.show_number(1234)
    sleep(1000)
```

XLAB Göttinger Experimentallabor für junge Leute

SITÄT

LED-Leiste (30 LEDs) Calliope mini

Anschluss mit Lüsterklemmen:
- Rot an VCC
- Schwarz an GND
- Gelb (Grün) z.B. an P0 (PIN 0)

1. Modul importieren
   import neopixel
2. Anzahl der LEDs festlegen
3. Pin festlegen

```
from calliopemini import*
import neopixel

n = 30
np = neopixel.NeoPixel(pin0, n)

np.fill((255,0,0)) # füllt alle Pixel mit der Farbe rot
np.show()
```

Einzelne LEDs anschalten

```
n = 30
np = neopixel.NeoPixel(pin0, n)

np[0] = (255, 0, 0) # 1.RGB-LED auf rot, vol
np[10] = (0, 128, 0) # 11. RGB-LED auf grün,
np[15] = (0, 0, 64)  # 16. RGB-LED auf blau,
np.show()
```

XLAB Göttinger Experimentallabor für junge Leute

IST-UNIVERSITÄT

LED-Matrix (8x8 LEDs) Calliope mini

Anschluss mit Lüsterklemmen:
- Rot an VCC
- Schwarz an GND
- Gelb (Grün) z.B. an P0 (PIN 0)

```
n = 64
np = neopixel.NeoPixel(pin0, n)

np.fill((255,0,0)) # füllt alle Pixel mit der Farbe rot
np.show()
```

1. Modul importieren
   import neopixel
2. Anzahl der LEDs festlegen
3. Pin festlegen

Einzelne LEDs anschalten

```
n = 64
np = neopixel.NeoPixel(pin0, n)

np[0] = (255, 0, 0) # 1. in 1. Zeile RGB-LED auf rot
np[10] = (0, 128, 0) # 3. LED in 2. Zeile auf grün
np[53] = (0, 0, 64)  # 5. LED 6. Zeile auf blau
np.show()
```

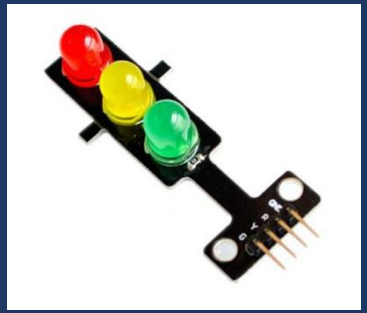
XLAB Göttinger Experimentallabor für junge Leute

GÖTTINGEN SEIT 1737

Ampel

Anschluss mit Krokodilklemmen oder Jumperkabel:

- 3V an VCC
- Rot an Pin0
- Gelb an Pin1
- Grün an Pin2




```
from calliopemini import *
pin0.write_digital(0)
pin1.write_digital(0)
pin2.write_digital(0)

while True:
    pin0.write_digital(1)
    sleep(1000)
    pin0.write_digital(0)
    sleep(500)
    pin1.write_digital(1)
    sleep(1000)
    pin1.write_digital(0)
    sleep(500)
    pin2.write_digital(1)
    sleep(1000)
    pin2.write_digital(0)
    sleep(500)
```

XLAB Göttinger Experimentallabor für junge Leute

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN IN PUBLICA COMMODA SEIT 1737

Praxisphase

Materialien:

- Präsentation mit Code-Beispielen mit QR-Code downloaden
- A4-Seite mit Übersicht der zur Verfügung stehenden Aktoren und Sensoren

XLAB Göttinger Experimentallabor für junge Leute

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN IN PUBLICA COMMODA SEIT 1737