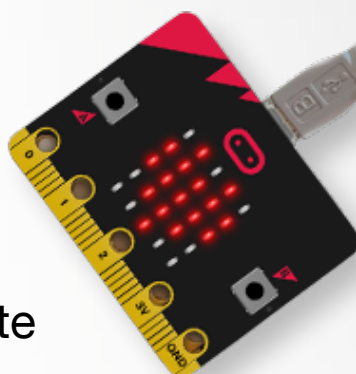


Programmieren in Python mit dem TI-84 Plus CE-T Python Edition Graphikrechner und BBC micro:bit



- ▶ Anleitung:
Erste Schritte



Dieses und weiteres Material steht Ihnen auf der TI Materialdatenbank zum Download bereit:
www.ti-unterrichtsmaterialien.net

© 2021 Texas Instruments

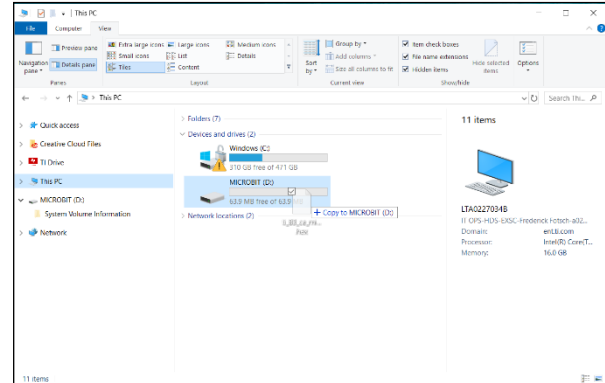
Dieses Werk wurde in der Absicht erarbeitet, Lehrerinnen und Lehrern geeignete Materialien für den Unterricht an die Hand zu geben. Die Anfertigung einer notwendigen Anzahl von Fotokopien für den Einsatz in der Klasse, einer Lehrerfortbildung oder einem Seminar ist daher gestattet. Hierbei ist auf das Copyright von Texas Instruments hinzuweisen. Jede Verwertung in anderen als den genannten oder den gesetzlich zugelassenen Fällen ist ohne schriftliche Genehmigung von Texas Instruments nicht zulässig. Alle Warenzeichen sind Eigentum ihrer Inhaber.

Anleitung: Erste Schritte

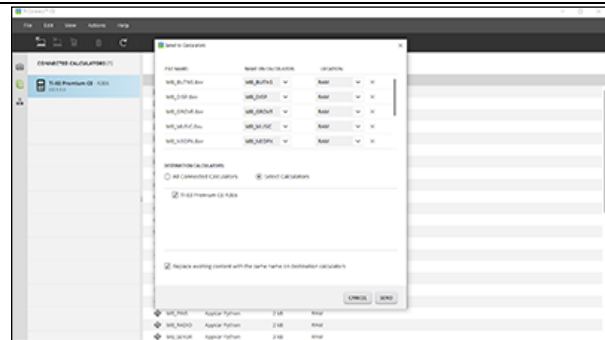
Installation

1. Verwenden Sie ein USB-Kabel, um das micro:bit an einen Computer anzuschließen. Das micro:bit wird als Laufwerk auf Ihrem Computer angezeigt. Ziehen Sie die Datei TI_Runtime_2.6.hex auf das micro:bit-Laufwerk. Die Datei wird in das micro:bit kopiert und ermöglicht eine verbesserte Kommunikation zwischen dem TI-84 Plus CE-T Python Edition und dem micro:bit.

Die Datei wird nicht dauerhaft auf micro:bit gespeichert. Wenn micro:bit wieder mit dem PC verbunden und in einer anderen Sprache als Python programmiert wird wie z.B. MakeCode, JavaScript, C++ oder Scratch, muss TI_Runtime_2.6.hex erneut installiert werden. Weitere Informationen finden Sie am Ende des Dokuments.

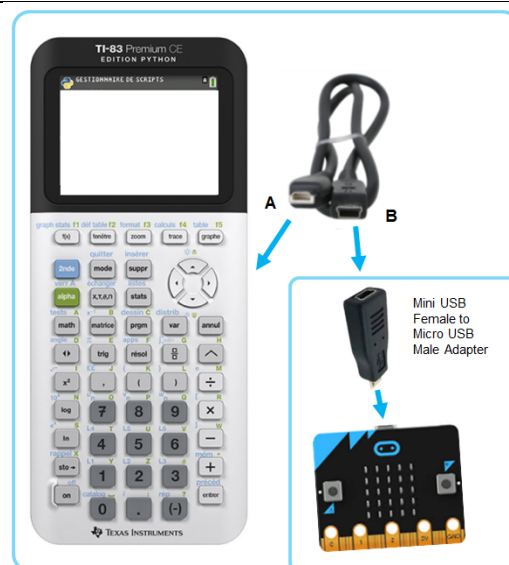


2. Verwenden Sie TI-Connect, um alle micro:bit-Module von Ihrem Computer auf den TI-84 Plus CE-T zu übertragen.



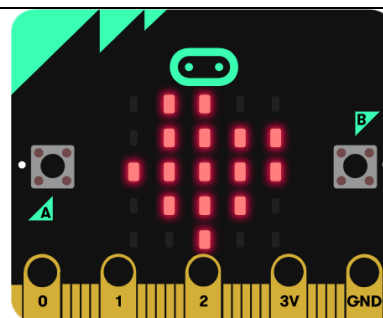
3. Schließen Sie das micro:bit mit dem TI-84 Plus CE-T-Rechner über das Kabel von Einheit zu Einheit und einen Mini-USB-Buchsen-zu-Micro-USB-Stecker-Adapter an.

Schließen Sie den Adapter an das B-Ende des Kabels an, und schließen Sie das A-Ende an den USB-Anschluss des Rechners an.

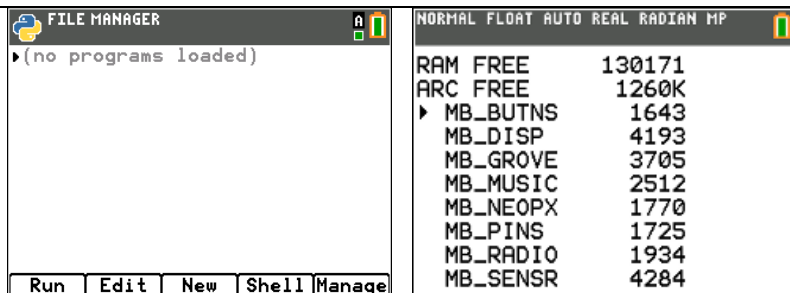


Anleitung: Erste Schritte

4. Starten Sie dann die Python-App.
Drücken Sie die Reset-Taste auf dem mikro:bit.
Wenn das Laufzeitprogramm erfolgreich geladen wurde, wird der TI-Symbol mit hell leuchtendem i-Punkt auf dem mikro:bit-LED-Display angezeigt.

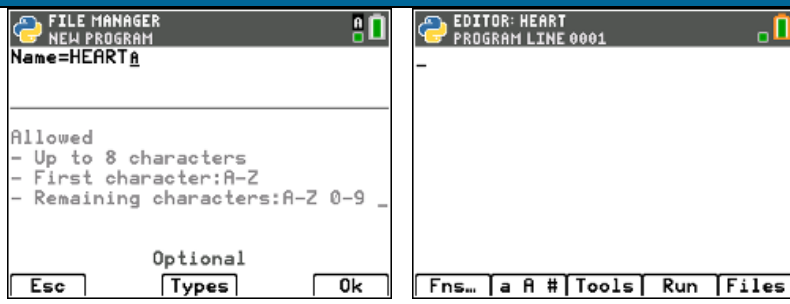


5. In der Python-App werden die mikro:bit-Module nicht als Programme angezeigt, denn sie sind im Archivspeicher als App-Variablen gespeichert.

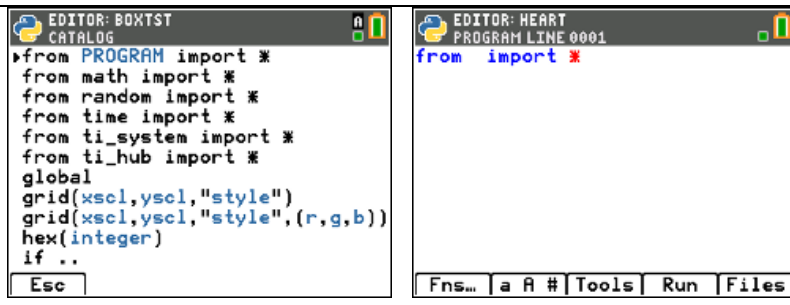


Erstellen Ihres ersten mikro:bit-Programmes

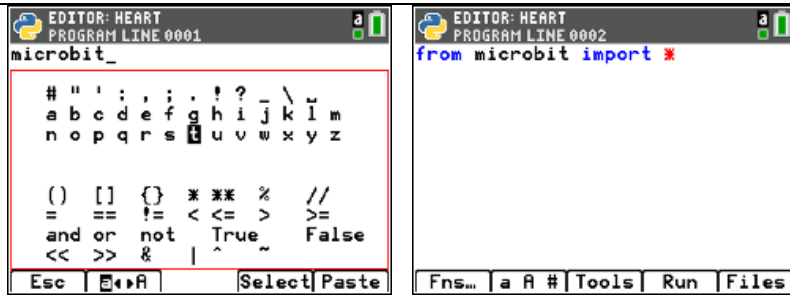
1. Wählen Sie **[New]**, um ein neues Programm (hier mit dem Namen **HEART**) zu erstellen.



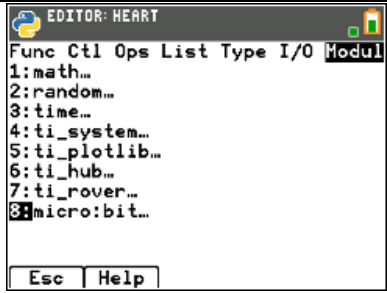
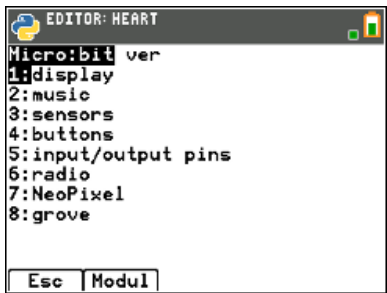
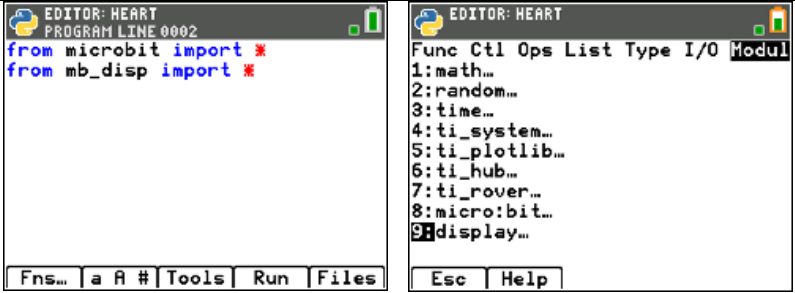
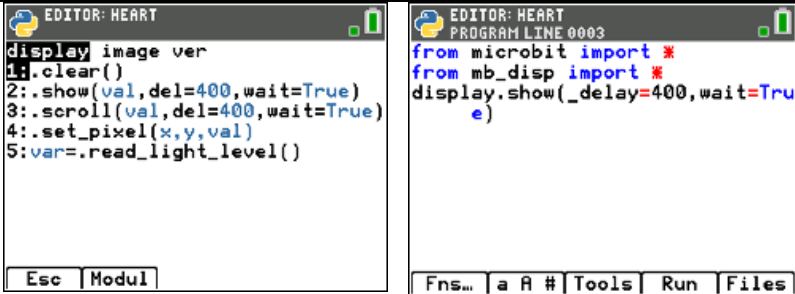
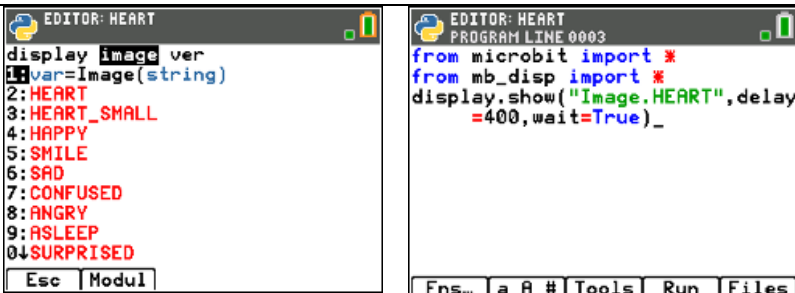
2. Als nächstes muss das mikro:bit eingebunden werden. Wählen Sie dazu **[2nd][catalog]** und dann **from PROGAM import ***.



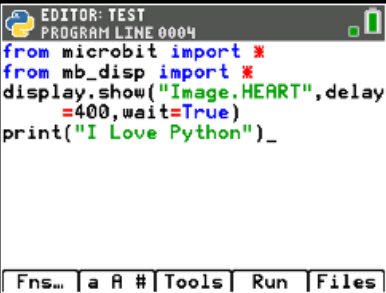

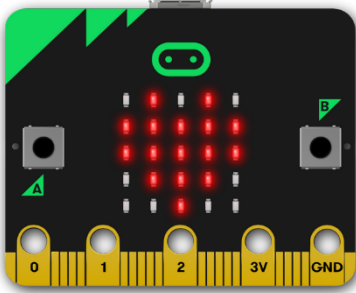
3. Setzen Sie den Cursor hinter **from** und wählen Sie **[a A #]**. Geben Sie dann **microbit** buchstabenweise über das Steuerkreuz oder mit den alpha-Tasten ein. Mit **Paste** wird das Wort dann in die **from import *** Anweisung eingefügt.




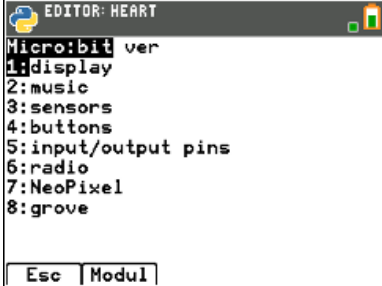
Anleitung: Erste Schritte

<p>4. Setzen Sie den Cursor dann an das Zeilenende. Mit [enter] wird das micro:bit-Modul importiert und eine neue Zeile erzeugt. Wählen Sie nun [Fns] und die Modul-Registerkarte, so sehen Sie micro:bit... als neue Option am Ende der Liste. Dieses Modul enthält eine Sammlung verschiedener Module für das micro:bit.</p>	
<p>5. Wählen Sie das micro:bit-Modul durch [enter] aus. Sie erhalten eine Liste aller Module, die im micro:bit-Modul enthalten sind. Für das Programm soll das display verwendet werden. Mit [enter] wird das Modul dem Programm im Editor hinzugefügt. Es können auch weitere Module hinzugefügt werden, aber: Wählen Sie nur Module aus, die für das Skript unerlässlich sind, um RAM zu sparen. Sie können alle Module mit Ausnahme des Hub- und Rover-Moduls verwenden.</p>	
<p>6. Das Modul mit dem Namen mb_disp wurde dem Programm hinzugefügt. Wählen Sie die [Fns]-Taste und dann Modul. Beachten Sie den neuen display...-Menüpunkt, der sich nun am Ende der Liste befindet. Weitere Microbit-Module würden hier ebenfalls hinzugefügt.</p>	
<p>7. Wählen Sie display... und dann show(val,del=400,wait=true) und [enter]. Im Editor blinkt der Cursor und erwartet die Eingabe der Variablen.</p>	
<p>8. Wählen Sie über [Fns] Modul erst display... und dann image. Wählen Sie HEART aus, um "Image.HEART" in das Programm einzufügen. delay und wait sind optionale Angaben und können auch entfallen.</p>	

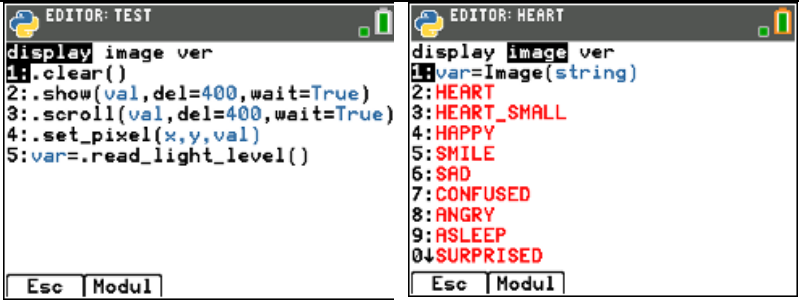

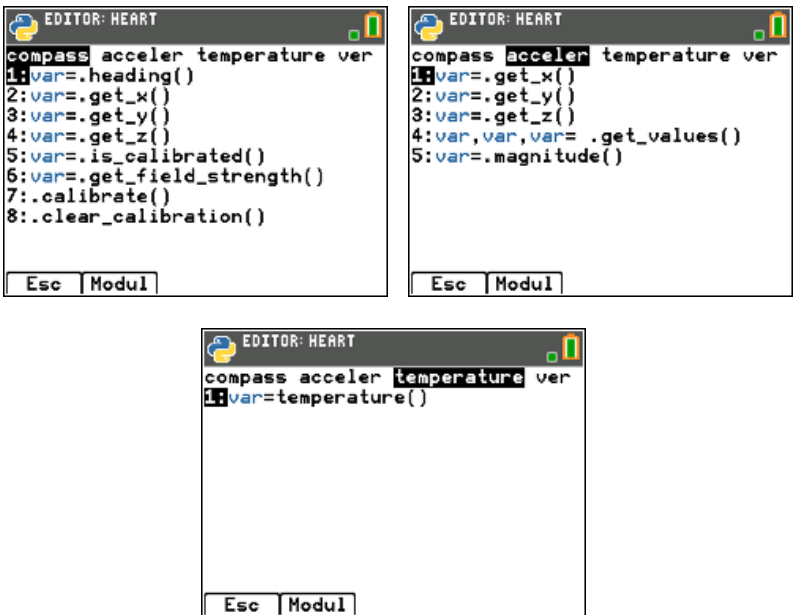
Anleitung: Erste Schritte

<p>9. Wählen Sie [Fns...] und I/O und dann print(). Geben Sie in die Klammern "I Love Python" ein.</p> <p>Die Anweisung "print" veranschaulicht, wie micro:bit-Anweisungen zusammen mit Standard- oder TI-Modul-Python-Anweisungen innerhalb desselben Programmes verwendet werden können.</p>	
<p>10. Stellen Sie sicher, dass das micro:bit über das Kabel und den Mini->Micro-Adapter mit dem Rechner verbunden ist, und wählen Sie dann [Run].</p> <p>Herzlichen Glückwunsch! Sie haben das micro:bit mit dem TI-84 Plus CE-T programmiert.</p>	 

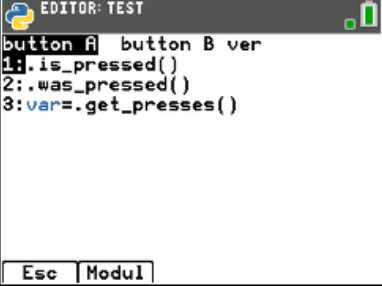
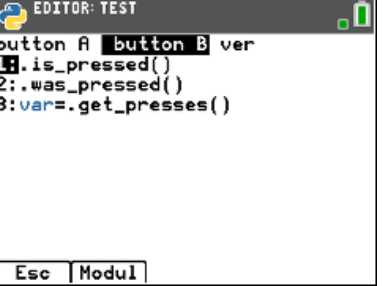
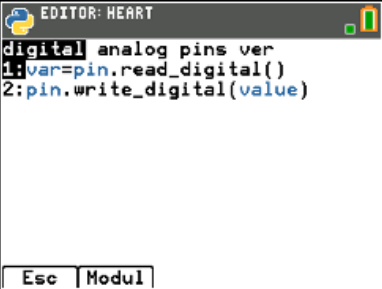
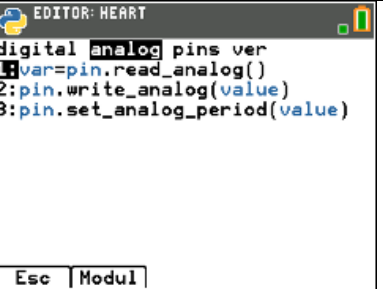
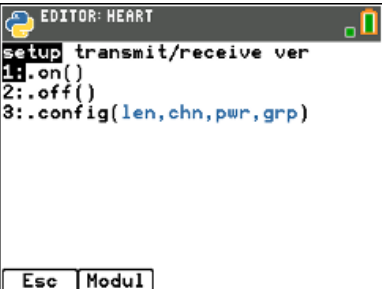
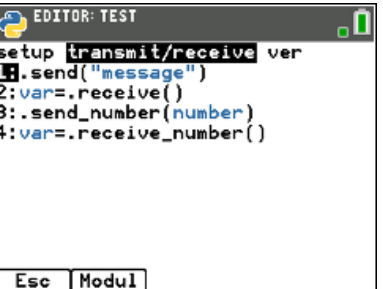
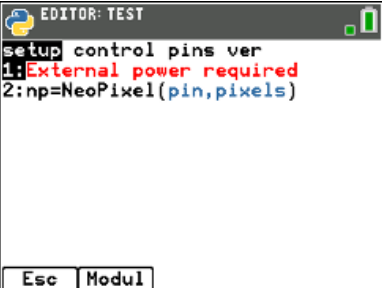
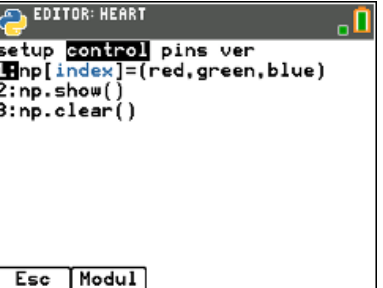
Module und Methoden

<p style="text-align: center;">Micro:bit - Module</p> <p>display – importiert als mb_disp music – importiert als mb_music sensors (eingebaute Sensoren) – importiert als mb_sensr buttons (A- und B-Tasten) – importiert als mb_butns input/output pins – importiert als mb_pins radio (2.4 MHz Funk) – importiert als mb_radio NeoPixel - importiert als mb_neopx grove (Grove Sensoren) – importiert als mb_grove</p>	<div style="display: flex; justify-content: space-around;"> <div data-bbox="715 1279 1098 1563">  </div> <div data-bbox="1118 1279 1501 1563">  </div> </div>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Anleitung: Erste Schritte

<p>display (5x5 LED Display Modul)</p> <p>display display.clear() display.show(val,del=400,wait=True) display.scroll(val,del=400,wait=True) display.setpixel(x,y,val) var=.read_light_level()</p> <p>image var=Image(String) Image.NAME (bis zum Buchstaben Z) ver (vorhanden in allen Modulen) module version (Kommentar)</p>	 <p>EDITOR: TEST display image ver 1: .clear() 2: .show(val,del=400,wait=True) 3: .scroll(val,del=400,wait=True) 4: .set_pixel(x,y,val) 5: var=.read_light_level()</p> <p>EDITOR: HEART display image ver 1: var=Image(string) 2: HEART 3: HEART_SMALL 4: HAPPY 5: SMILE 6: SAD 7: CONFUSED 8: ANGRY 9: ASLEEP 0: SURPRISED</p>
<p>music (eingebauter Speaker)</p> <p>play music.play(song,wait=True) music.pitch(frequency,t,wait=True) t: Dauer in s music.set_tempo(Ticks,BPM)</p> <p>songs songs.NAME (bis zum Buchstaben K)</p>	 <p>EDITOR: TEST play songs ver 1: play(song,wait=True) 2: pitch(frequency,t,wait=True) 3: set_tempo(ticks,BPM)</p> <p>EDITOR: HEART play songs ver 1: DADADADUM 2: ENTERTAINER 3: PRELUDE 4: ODE 5: NYAN 6: RINGTONE 7: FUNK 8: BLUES 9: BIRTHDAY 0: WEDDING</p>
<p>sensors (integriertes Sensormodul)</p> <p>compass var = compass.heading() var = compass.get_x() var = compass.get_y() var = compass.get_z() var = compass.is_calibrated() var = compass.get_field_strength() compass.calibrate() compass.clear_calibration()</p> <p>Befolgen Sie die Anweisungen auf micro:bit-Bildschirm, um die Kalibrierung abzuschließen.</p> <p>acceler var = accelerometer.get_x() var = accelerometer.get_y() var = accelerometer.get_z() var,var,var = accelerometer.get_values() var = accelerometer.magnitude()</p> <p>temperature var = temperature()</p> <p>Sensor an der Vorderseite des Boards</p>	 <p>EDITOR: HEART compass acceler temperature ver 1: var=.heading() 2: var=.get_x() 3: var=.get_y() 4: var=.get_z() 5: var=.is_calibrated() 6: var=.get_field_strength() 7: .calibrate() 8: .clear_calibration()</p> <p>EDITOR: HEART compass acceler temperature ver 1: var=.get_x() 2: var=.get_y() 3: var=.get_z() 4: var,var,var= .get_values() 5: var=.magnitude()</p> <p>EDITOR: HEART compass acceler temperature ver 1: var=temperature()</p>

Anleitung: Erste Schritte

<p style="text-align: center;">A- und B-Tastenmodul</p> <p>Button A <code>button_a.is_pressed()</code> <code>button_a.was_pressed()</code> <code>var = button_a.get_presses()</code></p> <p>Button B <code>button_b.is_pressed()</code> <code>button_b.was_pressed()</code> <code>var = button_b.get_presses()</code></p>		
<p style="text-align: center;">input/output pins (Eingangs- und Ausgangsanschlüsse)</p> <p>digital <code>var = pin.read_digital()</code> <code>pin.write_digital(value)</code></p> <p>analog <code>var = pin.read_analog()</code> <code>pin.write_analog(value)</code> <code>pin.set_analog_period(value)</code></p> <p>pins <code>pin0, pin1, pin2, pin8</code> <code>pin14, pin15,</code> <code>pin 16</code> <code>var = pin(number)</code></p>		
<p style="text-align: center;">radio (2,4 MHz Funkmodul)</p> <p>setup <code>radio.on()</code> <code>radio.off()</code> <code>radio.config(len,chn,pwr,grp)</code> Zwei Funkgeräte müssen Kanal und Gruppe teilen, um zu kommunizieren.</p> <p>transmit/receive <code>radio.send(„message“)</code> <code>var = radio.receive()</code> <code>radio.send_number(number)</code> <code>var = radio.receive_number()</code></p>		
<p style="text-align: center;">NeoPixel-Modul</p> <p>setup <code>np = NeoPixel(pin,pixels)</code> Mit dieser Anweisung können die Pixels (maximal 16) einem Pin zugeordnet werden. Der Strom ist auf 90mA begrenzt.</p> <p>control <code>np[index] = (rot, grün,blau)</code> <code>np.show()</code> <code>np.clear()</code></p>		

Anleitung: Erste Schritte

pins

pin0, pin1, pin2, pin8, pin13

```
EDITOR: TEST
setup control pins ver
1:pin0
2:pin1
3:pin2
4:pin8
5:pin13
Esc Modul
```

Grove (Grove-Sensormodul, Erweiterung)

input

```
var(t),var(h)=grove.read_sht()
var = grove.read_temperature(pin)
var = grove.read_lightlevel(pin)
var = grove.read_temperature(pin)
var = grove.read_moisture(pin)
var = grove.read_pressure(pin)
calibrate_pressure(m,b)
var = grove.read_ranger_time(pin)
var = grove.read_ranger_cm(pin)
```

output

```
grove.power(pin,value)
grove.relay(pin,value)
grove.set_servo(pin,deg,min,max)
```

pins

pin0, pin1, pin2, pin8, pin14, pin 15, pin16

```
EDITOR: HEART
input output pins ver
1:var(t),var(h)=.read_sht()
2:var=.read_temperature(pin)
3:var=.read_lightlevel(pin)
4:var=.read_moisture(pin)
5:var=.read_pressure(pin)
6:calibrate_pressure(m,b)
7:var=.read_ranger_time(pin)
8:var=.read_ranger_cm(pin)
Esc Modul
```

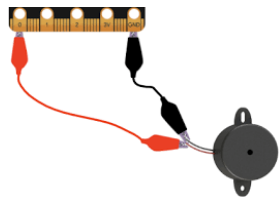
```
EDITOR: HEART
input output pins ver
1:.power(pin,value)
2:.relay(pin,value)
3:.set_servo(pin,deg,min,max)
Esc Modul
```

```
EDITOR: TEST
input output pins ver
1:pin0
2:pin1
3:pin2
4:pin8
5:pin14
6:pin15
7:pin16
Esc Modul
```

Testprogramme

DISPTTEST.PY: Veranschaulicht alle Befehle der 5x5-LED-Matrix.

MUSCTEST.PY: Demonstriert alle Musik- und Tonbefehle. Ein Lautsprecher kann an Pin0 und Masse angeschlossen werden.



SNSRTEST.PY: Demonstriert die Sensoren Kompass, Beschleunigungsmesser und Temperatur. Folgen Sie den Anweisungen auf dem micro:bit, um den Kompass zu kalibrieren.

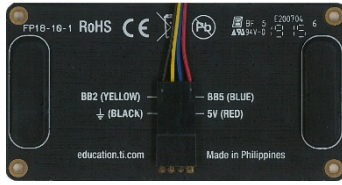
BUTNTEST.PY: Demonstriert alle A- und B-Tastenbefehle.

PINSTEST.PY: Demonstriert die analogen und digitalen Eingänge. Verwenden Sie eine Krokodilklemme, um 3V oder Gnd abwechselnd mit den Pins 0, 1 und 2 zu verbinden. Schließen Sie eine LED an Gnd und abwechselnd an die Pins 0, 1 oder 2 für analoge und digitale Ausgänge an.

RADITEST.PY: Demonstriert die Funkbefehle. Dazu muss dieses Programm auf zwei oder mehr Rechner mit angeschlossenen micro:bits laufen.

Anleitung: Erste Schritte

NPTEST.PY: Demonstriert den NeoPixel-Befehl. Um das TI-RGB-Array mit diesem Test zu verwenden, stellen Sie die folgenden Verbindungen her:



Mikrobit
3v
Gnd
pin0
pin1

TI-RGB-Array
5V (Rot)
⏏ (SCHWARZ)
BB2(GELB)
BB5(BLAU)

GROVTEST.PY: Die Demonstration erfordert eine Erweiterungsplatine mit Grove-Buchsen und den passenden Grove-Sensoren. Der Temperatur- und Feuchtigkeitssensor SHT35 muss an einen I2C-Anschluss angeschlossen werden. Überprüfen Sie auch die anderen Sensoranschlüsse.

Material



Erweiterungsplatine mit Grove-Ports

https://www.reichelt.de/micro-bit-grove-shield-v2-0-fuer-micro-bit-bbcz-grove-v2-0-p255699.html?&trstct=pos_4&nbc=1



Lautsprecher mit Leiterplatten-Befestigungsstiften

https://www.reichelt.de/kleinlautsprecher-al-23p-0-1w-8ohm-al-23p-p145879.html?&trstct=pol_8&nbc=1



USB Mini auf USB Micro Adapter

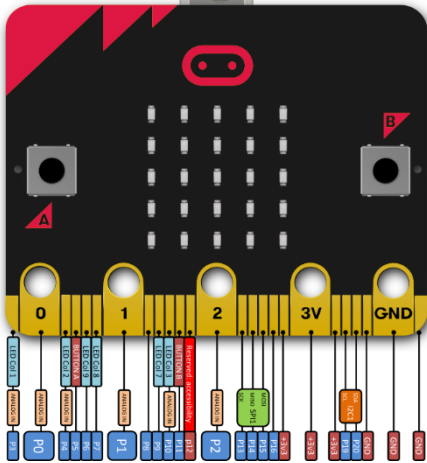
<https://www.reichelt.de/index.html?ACTION=446&LA=446&nbc=1&q=usb%20mini%20auf%20usb%20micro>



Krokodilklemmen

<https://www.reichelt.de/verbindungsleitungs-satz-10-teilig-mk-612s-p12242.html?search=krokodilklemmen>

Anleitung: Erste Schritte



micro:bit Application Programming Interface (API)

Das TI-micro:bitmodul ist auf die Standard-micro:bit-API ausgerichtet. Bitte verwenden Sie diese Referenz, wenn Sie micro:bit-Anweisungen in Ihre TI-Python-Programme einschließen.

<https://www.reichelt.de/bbc-micro-bit-v2-bbc-micro-bit-v2-p289797.html?search=bbc>

micro:bit Let's Code

Wird das micro:bit mit einer anderen Sprache wie MakeCode-Blöcken, JavaScript oder C++ programmiert, wird eine andere Laufzeit -hex-Datei auf das micro:bit geladen. Um die Kommunikationsfunktionalität des Rechners wiederherzustellen, muss die TI-84 Plus CE-T-Laufzeitdatei neu installiert werden wie im ersten Schritt dieses Dokuments angegeben. Das Ändern der Laufzeitbibliotheken schadet dem micro:bit nicht.

<https://microbit.org/code/>

micro:bit C/C++ - Laufzeitprogramm

<https://lancaster-university.github.io/microbit-docs/>

micro:bit Mikropython

<https://tech.microbit.org/software/micropython/>

TI-84 Plus CE-T Python Edition Betriebssystem

<https://education.ti.com/de/software/update/84-ce-software-update/84ce-download?q1=84-cet-py-os&count=1>

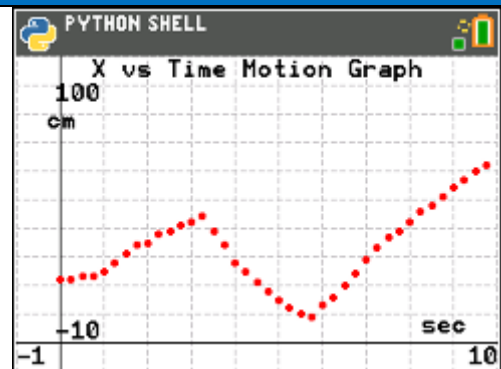
Programmierbeispiele

```
import ti_plotlib as plt
from ti_system import *
from microbit import *
from mb_grove import *

plt.cls()
plt.window(-1,10,-10,100)
plt.grid(1,10,"dash")
plt.axes("on")
plt.labels("sec "," cm",11,3)
plt.title("X vs Time Motion Graph")
plt.color(255,0,0)

initilize=grove.read_ranger_cm(pin0)

for i in range(40):
    echo=grove.read_ranger_time(pin0)
    d=echo/2*34000
    t=i*.25
    sleep(250)
    plt.plot(t,d,"o")
plt.show_plot()
```



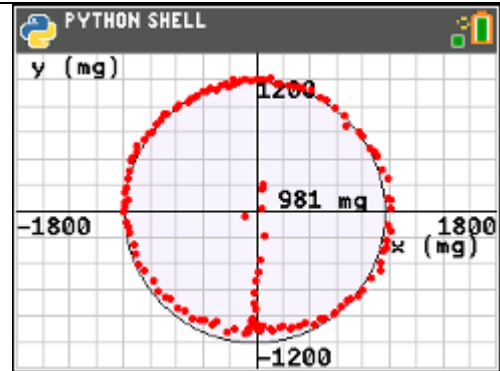
Anleitung: Erste Schritte



```
import ti_plotlib as plt
from ti_system import *
from microbit import *
from mb_sensr import *

plt.window(-1800,1800,-1200,1200)
x = 318/2
y = 30+212/2
r = 981*318/(plt.xmax-plt.xmin)
plt.color(240,240,240)
plt.cls()
plt.gr.fillCircle(x,y,r)
plt.grid(200,200,"solid")
plt.color(0,0,0)
plt.gr.drawArc(x-r,y-r,2*r,2*r,0,3600)
plt.axes("on")
plt.labels("x (mg) "," y (mg)",8,1)
plt.gr.drawString("981 mg",x+15,y-15)
plt.color(255,0,0)

while not escape():
    accx = accelerometer.get_x()
    accy = accelerometer.get_y()
    plt.plot(accx,accy,"o")
plt.show_plot()
```



Haben Sie Fragen zu Produkten von Texas Instruments? Oder sind Sie an weiteren Unterrichtsmaterialien oder einer Lehrerfortbildung interessiert? Gerne steht Ihnen auch unser Customer Service Center mit Rat und Tat zu Seite. Nehmen Sie mit uns Kontakt auf:



Customer Service Center
TEXAS INSTRUMENTS
education.ti.com/csc

education.ti.com/deutschland

education.ti.com/oesterreich

education.ti.com/schweiz