



#### Lektion 6: micro:bit und Python

#### Übung 1: Das Display

In dieser Einheit wird gezeigt, wie man das Display des micro:bit auf verschiedene Weise ansteuern kann. Die Einheit hat zwei Teile:

Teil 1: Alien.Begegnung

Teil 2: Darstellung von Bildern

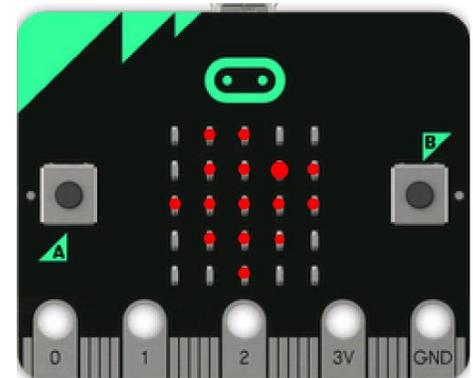
#### Lernziele:

- Ansteuern des Displays auf dem micro:bit durch die Befehle `.show( )`, `.scroll( )` und `.show(image)`
- Steuern der Geschwindigkeit des Displays durch `sleep(ms)`

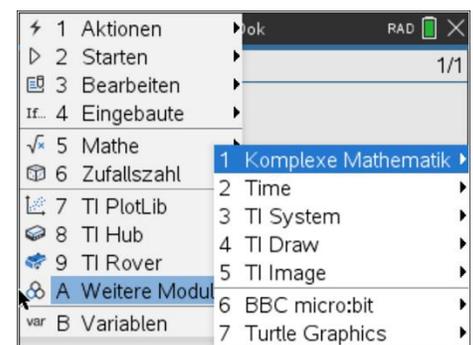
1. Bevor man anfängt, sollte man überprüfen
  - dass der TI-Nspire CX II das OS **5.3** oder höher hat
  - dass man die Übungen 1 – 6 der Programmierung in Python erfolgreich absolviert hat.
  - dass das **micro:bit** mit dem TI-Nspire CX II verbunden ist.
  - dass das in der Anleitung zum micro:bit beschriebene SetUp einschließlich der Übertragung des dort beschriebenen Programmes erfolgt ist: <https://education.ti.com/en/teachers/microbit>  
*Das SetUp muss nur einmal durchgeführt werden, allerdings sollte man sich über Updates informieren, um das micro:bit stets aktuell zu halten.*



2. Wenn alle Bedingungen erfüllt sind, sollte das micro:bit so aussehen, wenn es vom TI-Nspire mit Strom versorgt wird: Das Display auf dem micro:bit zeigt das TI-Logo, ein Icon des Bundesstaates Texas mit einem hellen Fleck in der Nähe von Dallas, der Heimat von **Texas Instruments, Inc.**



3. Das micro:bit – Modul ist jetzt in der Python-Bibliothek installiert. Drückt man im Python-Editor **[menu] > Weitere Module**, so sieht man, dass das **BBC micro:bit** unterhalb der TI-Module gelistet ist. Die Module werden alphabetisch nach ihrem Dateinamen aufgelistet, deshalb erscheint das BBC micro:bit nicht unter den B's, sondern unter den m's.



**Lehrtipp: Erste Schritte mit micro:bit**

Dieses Gerät geht davon aus, dass der micro:bit und der TI-Nspire Handheld (oder die Computersoftware) "ready to go" sind. Stellen Sie sicher, dass Sie (oder Ihre Schüler) die Einrichtungsschritte auf ihren einzelnen Taschenrechnern und micro:bits abgeschlossen haben. Lesen und führen Sie die Einrichtungsanweisungen im Handbuch "Erste Schritte" aus, das im Download-Ordner der TI Education-Website enthalten ist.

<https://education.ti.com/en/teachers/microbit>.

Wie bei jeder neuen Software und Hardware sollten Sie über Upgrades und Updates auf dem Laufenden bleiben.

Das **BBC micro:bit module (microbit.tns)** ist ein spezielles Modul von Texas Instruments, das im Pylib-Ordner jedes Handhelds und in der TI-Nspire CX II Computer Software installiert ist und daher unter **[menu] > Weitere Modules >** als **BBC micro:bit** erscheint.

Die ersten sechs Einheiten von Python: **10 Minuten Coding** sollten vor dieser Einheit abgeschlossen sein. Die hier vorgestellte Codierung ist nicht kompliziert, aber man sollte sicher in Python auf dem TI-Nspire CX II codieren können.

Diese Lektion ist für die Verwendung mit den micro:bit-Versionen 1 und 2 konzipiert, verwendet jedoch keine Funktionen, die nur in Version 2 enthalten sind. Das BBC micro:bit-Modul unterstützt allerdings diese Funktionen.

**Alle online gefundenen micro:bit-Projekte (einschließlich micro:bit Version 2-Projekte) können auf dem TI-Nspire CX II mit Python und einem angehängten micro:bit entwickelt werden. Aber es gibt zwei wichtige Unterscheidungen:**

1. Sie können das micro:bit nicht vom Taschenrechner trennen, während das Programm läuft, auch wenn eine Batterie an den micro:bit angeschlossen ist. Die micro:bit-Laufzeit (HEX-Datei) ist so konfiguriert, dass sie auf die Anweisungen des Rechners hört. Der Taschenrechner schaltet den micro:bit ein und steuert ihn.
1. Viele der online gefundenen Demo-Programme verwenden While True: um eine Endlosschleife zu erstellen. Diese Schleife läuft direkt auf dem micro:bit-Board, bis das Programm mit einer heruntergeladenen .hex-Datei durch ein anderes ersetzt wird. Bei Verwendung des TI-Nspire steuert der Taschenrechner das micro:bit, so dass die am häufigsten verwendete Schleife lautet

**While get\_key() != 'esc':**

Dadurch kann der Benutzer die Taste [ESC] drücken, um das Programm zu beenden.

**Wichtig: Wenn die Meldung 'micro:bit not connected' erscheint, trennen Sie einfach das micro:bit und stecken Sie es wieder ein (Reset).**



4. **Teil 1: Alien-Begegnung:** Natürlich geht es los, wie bei jeder anderen ersten Programmiererfahrung, mit der Anzeige einer Meldung auf dem micro:bit-Display.

Starten Sie ein neues TI-Nspire-Dokument und wählen **Sie Python > Neu**, um ein neues Programm (leeres Programm) mit dem Namen "alien" hinzuzufügen. Verwenden Sie im Python-Editor **[menu] > Weitere Module > BBC micro:bit**, um die Importanweisung oben in den Menüpunkten auszuwählen:

```
from microbit import *
```

*Tipp: Wenn jemals die Meldung 'micro:bit not connected' erscheint, trennen Sie einfach das micro:bit und stecken Sie es wieder ein (reset).*

**Lehrtipp:** Wenn kein micro:bit angeschlossen ist, meldet die Importanweisung selbst einen Fehler, wenn das Programm ausgeführt wird. Da Python eine "modulare" Sprache ist, werden bei Bedarf zusätzliche Funktionen hinzugefügt. Diese Importanweisung lädt bestimmte Funktionen/Methoden, die für den Betrieb des micro:bit mit dem TI-Nspire CX II erforderlich sind. Es lädt auch einige nützliche Methoden aus anderen Standardmodulen und ti-Modulen.

5. Um eine Textnachricht auf dem micro:bit-Display anzuzeigen, verwenden Sie die Anweisung:

```
display.show(image or text)
```

Diese Anweisung finden Sie auf:

**[menu] > Weitere Module > BBC micro:bit > Display > Methods**

Die Anweisung wird als **display.show(images or text)** eingefügt, aber **images or text** ist nur ein Platzhalter, der durch etwas ersetzt werden muss. Ersetzen Sie es durch ihre Nachricht in Anführungszeichen:

```
"greetings, earthlings"
```

Wenn Sie dieses Programm ausführen, indem Sie **[ctrl] [R]** drücken, werden die Buchstaben Ihrer Nachricht einer nach dem anderen sehr schnell auf dem Display angezeigt. Die Kleinbuchstaben "e" erscheinen zwar zweimal, aber Sie können sie nicht unterscheiden.

*Wenn Sie einen Fehler machen.... Gehen Sie zurück zur ersten Seite, um Ihr Programm zu bearbeiten, und führen Sie das Programm dann erneut aus. Haben Sie vergessen, den Text, den Sie anzeigen möchten, mit Anführungszeichen zu ergänzen?*

```
1.1 *alien.py 3/3
from microbit import *
```

```
1.1 1.2 *Dok RAD
alien.py erfolgreich gespeichert
from microbit import *
display.show("greetings, earthlings")
```



6. Eine bessere Methode zum Anzeigen von Meldungen ist:  
**display.scroll("greetings, earthlings")**

die auch auf

**[menu] > Weitere Module > BBC micro:bit > Display > Methods**  
zu finden ist.

Um die **.scroll()**-Anweisung abzuschließen, können Sie die Zeichenfolge aus der **.show-Anweisung** kopieren/einfügen.

Machen Sie die vorherige **.show()-Anweisung** zu einem Kommentar (**#comment**), indem Sie den Cursor in diese Zeile platzieren und drücken Sie dann **[ctrl] [T]**, um sie zu deaktivieren. Führen Sie das Programm dann erneut aus.

*Ja, Sie können auch einfach **.show** zu **.scroll** durch überschreiben ändern*

7. Die **.scroll()-Methode** bewirkt, dass die Nachricht wie ein Banner über das Display von rechts nach links scrollt (sich bewegt), was das Lesen erleichtert.

Sie können die Geschwindigkeit des Bildlaufs steuern, indem Sie den **Delay=** - Parameter hinzufügen:

**display.scroll("greetings, earthlings", delay = 200)**

der eine Verzögerung von 200 Millisekunden (0,2 Sekunden) beim Scrollen hinzufügt. Versuchen Sie es auch mit anderen Verzögerungswerten.

8. **Teil 2: Schlagendes Herz (Beating.Heart)**

Drücken Sie **[ctrl] [doc]**, um eine neue Seite dem Dokument hinzuzufügen, und wählen Sie **Python hinzufügen > Neu**, um ein neues Python-Programm zu erstellen. Im Beispiel heißt es **'beat'**.

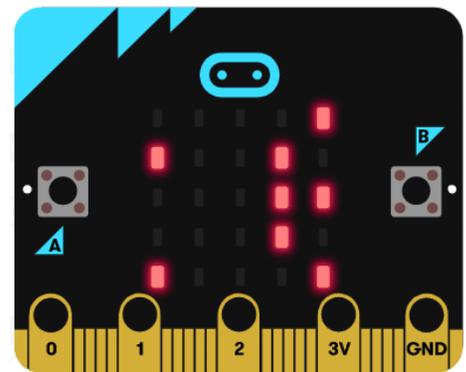
Über **[menu] > Weitere Module > BBC micro:bit** fügen Sie nun hinzu:  
**from microbit import \***

```

1.1 1.2 *Dok RAD
alien.py erfolgreich gespeichert
from microbit import *

#display.show("greetings, earthlings")
display.scroll("greetings, earthlings")

```



<greetings, earthlings.gif>

```

1.1 1.2 *Dok RAD
beat.py 1/1

```



9. Mit der Anweisung

**display.show( )**

kann das Bild eines Herzens dargestellt werden. Diese Anweisung findet man unter:

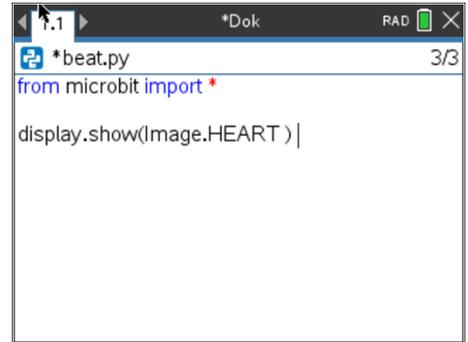
**[menu] > Weitere Module > BBC micro:bit > Display > Methods**

Den Platzhalter innerhalb der Klammern ersetzt man durch das Bild HEART aus **[menu] > Weitere Module > BBC micro:bit > Display > Images > Set 1 > Heart:**

:

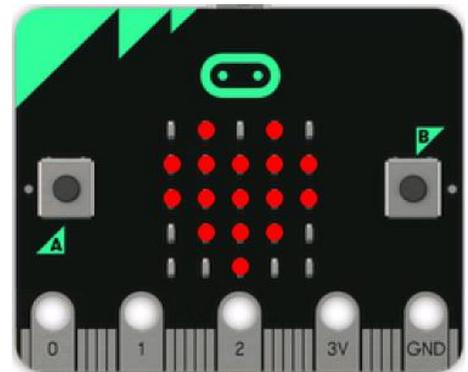
**Image.HEART**

10. Lässt man nun das Programm laufen, indem man **[ctrl] [R]** drückt, so sieht man das nebenstehend abgebildete HEART-Icon auf der 5x5 LED-Matrix. Das Bild bleibt bestehen, solange es nicht durch etwas anderes ersetzt wird, selbst wenn das Programm beendet ist.



```
*beat.py 3/3
from microbit import *

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



11. Es gibt auch noch ein kleines Herz, das ebenfalls abgebildet werden soll:

**display.show(Image.HEART\_SMALL)**

Die Anweisung wird auf dieselbe Art eingefügt wie die darüber stehende.

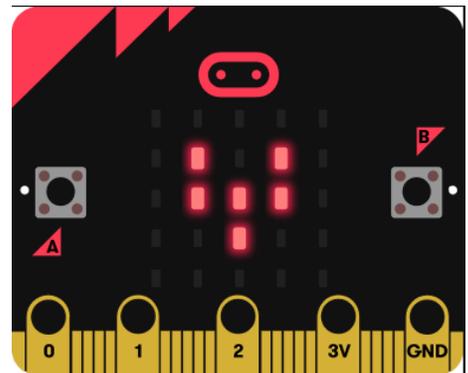
*Hinweis: Man kann natürlich auch die erste display-Anweisung durch copy/paste einfügen und weiter editieren. Für SMALL. benötigt man den Unterstrich und man muss Großbuchstaben verwenden.*



```
beat.py 5/5
from microbit import *

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

12. Führen Sie das Programm erneut aus. Es zeigt schnell das große Herz an und dann das kleine Herz, das nebenstehend abgebildet ist.





13. **Fügen Sie eine Schleife ein:** Um die beiden Herzen dazu zu bringen, wiederholt zu blinken ("Schlagen"), schließen Sie die beiden Anzeigeanweisungen in eine Schleife ein. Vor den beiden Anzeigeanweisungen muss dabei die Anweisung eingefügt werden:  
**while get\_key() != "esc":**

Man findet sie unter

**[menu] > Weitere Module > BBC micro:bit > Commands**

**Wichtig:** Rücken Sie die beiden Anzeigeanweisungen so ein, dass sie den Schleifenkörper bilden.

Das Programm kann nun jederzeit durch Drücken von **[esc]** angehalten werden.

*Wichtiger Hinweis: Einrückungen sind in Python-Programmen von entscheidender Bedeutung, denn so identifiziert Python Schleifenblöcke und If-Blöcke. Wenn die beiden Anzeigeanweisungen nicht mit der gleichen Anzahl von Leerzeichen eingerückt sind, wird ein Syntaxfehler angezeigt. Verwenden Sie die **[Leerzeichen]**- oder **[Tabulatortaste]**, um die beiden Zeilen in gleicher Höhe einzurücken. Einzugsräume werden in diesem Editor als hellgraue Diamantsymbole (◆◆) angezeigt, um die korrekte Einrückung zu unterstützen.*

14. Führen Sie Ihr Programm noch einmal aus und beobachten Sie das schlagende Herz! Drücken Sie die **[Esc]-Taste**, um das Programm zu beenden.

*Hinweis: Wenn Sie denken, dass Ihr Programm in einer Endlosschleife steckt, drücken und halten Sie die Taste **[home/on]** auf Ihrem TI-Nspire gedrückt, um das Programm zu "unterbrechen". Dies kann passieren, wenn Sie **while** aus dem Menü "Befehle" nicht ordnungsgemäß verwenden.*

15. Das Menü **micro:bit Commands** enthält einige nützliche Python-Befehle, die auch in anderen Menüs zu finden sind. Das **micro:bit-Modul** importiert diese Python-Befehle für Sie.

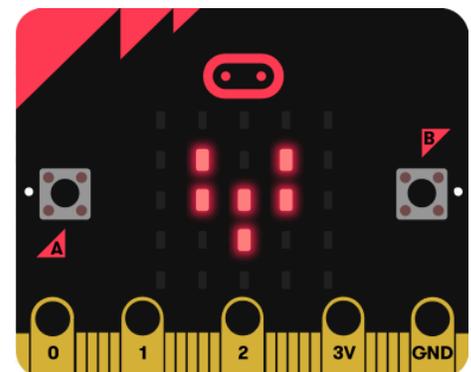
*Darüber hinaus können Sie natürlich auch alle anderen Python-Anweisungen verwenden, wenn sie zur Ausführung des Programmes notwendig sind.*

```

1.1 1.2 *Dok RAD
*beat.py 7/7
from microbit import *

while get_key() != "esc":
    display.show(Image.HEART)
    display.show(Image.HEART_SMALL)

```



<beating\_heart.gif>

```

1 Actions 1 from microbit import *
2 Command 2 while get_key() != "esc":
3 Display 3 store_list("name",list)
4 Music 4 clear_history()
5 Audio and 5 while True:
6 Buttons ar 6 if <Boolean Expression>: break
7 Sensors 7 module version()
8 Radio 8 Help
9 I/O Pins
A Grove Devices

```



16. Um die schlagende Herzfrequenz zu steuern, fügen Sie zwei **sleep()-Anweisungen** hinzu, eine nach jeder Anzeigeanweisung :

**sleep(1000)**

bedeutet 1000 Millisekunden oder 1 Sekunde Verzögerung.

Die Anweisung findet man ebenfalls in **[menu] > Weitere Module > BBC micro:bit > Commands**

*Hinweis: Beachten Sie die Einrückung!*

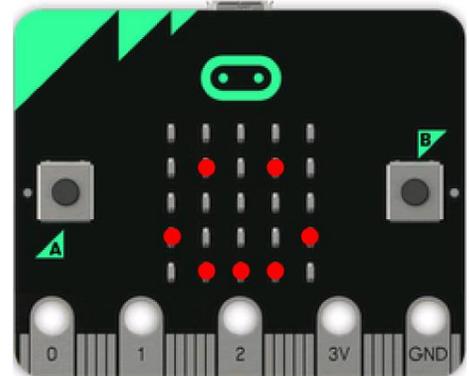
```

1.1 1.2 *Dok RAD 9/9
*beat.py
from microbit import *

while get_key() != "esc":
    display.show(Image.HEART)
    sleep(1000)
    display.show(Image.HEART_SMALL)
    sleep(1000)

```

17. **Erweiterung:** Versuchen Sie **'Gesichtermachen'**. Verwenden Sie eine ähnliche Programmstruktur wie **"Beating Heart"**, aber verwenden Sie stattdessen die Gesichtsbilder in Set 1.



**Hinweis:** Die Online-micro:bit-Lektionen verwenden normalerweise eine **While True:-Schleife** in Python und eine Forever:-Schleife in MakeCode. Diese "Endlosschleifen" laufen direkt auf dem micro:bit, bis es ausgeschaltet oder durch ein anderes Programm (.hex-Datei) ersetzt wird. Wenn Sie den TI-Nspire zur Steuerung des micro:bit verwenden, ist die Endlosschleife nicht erforderlich, da der Taschenrechner die volle Kontrolle hat.

**While True:** befindet sich im Menü Befehle, sollte aber in Verbindung mit **if (Boolescher Ausdruck): break** verwendet werden, um aus der Endlosschleife herauszukommen.

**Wenn ein Schüler in einer Endlosschleife auf dem Handheld stecken bleibt, drücken und halten Sie die [On]-Taste gedrückt, um das Programm zu "unterbrechen".**

Informationen zum Erstellen von Python-Programmen auf Computern finden Sie im Handbuch "Erste Schritte" (PDF), das in der micro:bit-Software von TI enthalten ist.

Das Menü **"Commands"** enthält eine Sammlung gängiger, häufig verwendeter Python-Befehle, die sich beim Codieren für micro:bit als nützlich erweisen. Das micro:bit-Modul importiert diese Befehle bei Bedarf aus anderen Modulen. Aber alle Python-Befehle sind aus anderen Menüs verfügbar und können in micro:bit-Projekten verwendet werden.

**Optional:** Um die Anzeige am Ende eines Programms wieder in ihren "TI"-Zustand (das TI-Logo von Texas) zu versetzen, fügen Sie einfach die Anweisung

**display.show( ti )** (ti in Kleinbuchstaben eingeben)

nach dem Ende der While-Schleife hinzu.



Dies ist nicht notwendig, aber es hilft anzuzeigen, dass das Programm sowohl auf dem micro:bit-Display als auch auf dem Bildschirm beendet ist.

Zu Beginn eines micro:bit-Programms (bevor die while-Schleife beginnt) kann es hilfreich sein, etwas auf dem Taschenrechnerbildschirm zu drucken, z. B. print('running...').

**Über sleep( ):** Die einzige Python-Funktion, die für das micro:bit *signifikant* modifiziert wurde, ist die sleep( )-Anweisung im Time-Modul. Normalerweise stellt der Argumentwert "Sekunden" dar, aber das micro:bit-Modul hat die Sleep-Methode so überarbeitet, dass der Wert Millisekunden (*Tausendstelsekunden*) darstellt. Nach dem Import des micro:bit-Moduls stellt sleep(1000) eine Verzögerung von einer Sekunde dar. Beachten Sie, dass sleep( ) jetzt die Geschwindigkeit des Taschenrechnerprogramms steuert, auch wenn sleep( ) in einem anderen Zusammenhang verwendet wird (z.B. für eine Anzeige auf dem Display des TI-nspire).

Die Methode **.show()** hat zwei optionale Parameter: **delay=** und **wait=**  
**display.show(Wert, delay=xxx , wait=False/True)**

**delay** Der Wert **xxx** ist in Millisekunden und verlangsamt die Anzeige.

**wait=True** blockiert die Verarbeitung, bis die Animation abgeschlossen ist, andernfalls erfolgt die Animation im Hintergrund.

Dies ist die vollständige Liste der Bilder:

Set 1	Set 2	Set 3	Set 4
1:HEART	1:YES	1:MUSIC_CROCHET	1:BUTTERFLY
2:HEART_SMALL	2:NO	2:MUSIC_QUAVER	2:STICKFIGURE
3:HAPPY	3:TRIANGLE	3:MUSIC_QUAVERS	3:GHOST
4:SMILE	4:TRIANGLE_LEFT	4:PITCHFORK	4:SWORD
5:SAD	5:CHESSBOARD	5:XMAS	5:GIRAFFE
6:CONFUSED	6:DIAMOND	6:PACMAN	6:SKULL
7:ANGRY	7:DIAMOND_SMALL	7:TARGET	7:UMBRELLA
8:ASLEEP	8:SQUARE	8:TSHIRT	8:SNAKE
9:SURPRISED	9:SQUARE_SMALL	9:ROLLERSKATE	
A:SILLY	A:RABBIT	A:DUCK	
B:FABULOUS	B:COW	B:HOUSE	
C:MEH	C:TI LOGO	C:TORTOISE	