

## So funktioniert es

- Sie werden in einer Gruppe von mindestens zwei Teammitgliedern arbeiten. Jeder hat einen TI-Nspire CX II CAS mit angeschlossenem USB-Kabel und eine *micro:bit* Mikrocontroller Platine.
- Das Projekt *CyberSecurity* besteht aus sechs Modulen mit jeweils einer zugehörigen tns-Datei.
- In der Datei sind drei verschiedene Rollen der Teammitglieder integriert: „Sender“, „Empfänger“ und „Hacker“. Diese Rollen sollten während der Aktivitäten getauscht werden.
- Befolgen Sie die Anweisungen im Arbeitsblatt, während Sie die Programme auf Ihrem Rechner ausführen.
- Sie können ein Programm jederzeit erneut ausführen.

### Rolle des Senders

```

student_sender.py 6/11
from microbit_radio import *
# Der geheime Kanal und die geheime Gruppe
# müssen identisch mit den vom Empfänger
# verwendeten Kanal und Gruppe sein
channel = 1
group = 1
message = "Das Gold ist in der Keksdose versteckt"
clear_history()
print("\nmessage=",message)
tx(message,channel,group)
    
```

### Rolle des Empfängers

```

student_receiver.py erfolgreich gespeichert
from microbit_radio import *
# Der geheime Kanal und die geheime Gruppe
# müssen identisch mit den vom Sender
# verwendeten Kanal und Gruppe sein
channel = 1
group = 1
clear_history()
message = rx(channel,group)
print("\nmessage=",message)
    
```

### Rolle des Hackers

```

student_hacker.py 11/11
from microbit_radio import *
# Der geheime Kanal und die geheime Gruppe
# müssen identisch mit den vom Empfänger
# verwendeten Kanal und Gruppe sein
channel = 1
group = 1
clear_history()
message = rx(channel,group)
print("\nmessage=",message)
    
```

## Die Programme

In jeder tns-Datei gibt es drei verschiedene Arten von Seiten, die den drei Rollentypen zugeordnet sind. In der Tabelle sind die drei Arten in einer kurzen Beschreibung aufgeführt:

Das **Python-Programm** enthält den Code zur Lösung der Aufgabe. Beachten Sie die Modul-Import-Anweisung und die Code-Kommentare innerhalb des Programms.

Die **Python-Shell** ist die Seite, wo das Programm der vorherigen Seite ausgeführt wird. Die Programmeingabe und -ausgabe erfolgt auf dieser Seite.

Die **Moduleseite** enthält Python-Code, der im Studentenprogramm verwendet wird. Dieser Code wird mit der Anweisung *import* importiert. Dieser Code enthält nützliche Funktionen.

```

student_receiver.py erfolgreich gespeichert
from microbit_radio import *
# Der geheime Kanal und die geheime Gruppe
# müssen identisch mit den vom Sender
# verwendeten Kanal und Gruppe sein
channel = 1
group = 1
clear_history()
message = rx(channel,group)
print("\nmessage=",message)
    
```

```

Python-Shell 9/9
>>>#Running student_receiver.py
>>>from student_receiver import *
>>>
Connecting to micro:bit...
Micro:bit USB connection error:
- Unplug/replug micro:bit.
- Ensure TI-runtime is loaded on micro:bit.
- Correct, and rerun your program.
>>>|
    
```

```

microbit_radio.py erfolgreich gespeichert
# Dieser code ist ein Modul und enthält die
# tx() Sende- und rx() Empfang-Funktionen.
# Ändern oder löschen Sie diese Seite nicht.
from microbit import *
def tx(msg,ch,gp):
    radio.on()
    radio.config(length=250, channel=ch,power=6)
    radio.send(msg)
    print("Botschaft gesendet!")
    
```

## Was werden Sie tun?

- Zu jeder Aktivität gibt es ein ähnliches Handout mit Anweisungen für den geeigneten Ablauf.
- Ihr Lehrer wird Ihrem Team eine Gruppennummer (0..255) zuweisen, die Sie bei allen Aktivitäten verwenden.
- Öffnen Sie die tns-Datei auf Ihrem Rechner und wählen Ihre Rolle:
  - Der *Empfänger* wählt in der Datei die Seite "student\_receiver.py" aus, ändert die Gruppennummer entsprechend der Vorgabe des Lehrers und führt dann das Programm aus, bevor der Sender sein eigenes ausgeführt hat.
  - Der *Hacker* wählt in der Datei die Seite "student\_hacker.py" aus, ändert die Gruppennummer entsprechend der Vorgabe des Lehrers und führt dann das Programm aus, bevor der *Sender* sein eigenes ausgeführt hat.
  - Der *Sender* wählt in der Datei die Seite "student\_sender.py" aus, ändert die Gruppennummer entsprechend der Vorgabe des Lehrers, bearbeitet dann die Nachrichtenzeichenfolge und führt anschließend das Programm aus, nachdem der *Empfänger* und der *Hacker* ihre Programme gestartet haben.

## Weitere Übungen

Unter dieser Überschrift finden Sie in jeder Aktivität Vorschläge, wie Sie Python erforschen, spielen und mehr über das Programmieren lernen können. Am besten lernt man, indem man mit dem Code spielt, um zu sehen, was jede Zeile im Programm bewirkt.

## Prüfen Sie Ihr Verständnis

In diesem Abschnitt der Aktivität finden Sie eine Zusammenfassung dessen, was die Aktivität zu demonstrieren versuchte. Wenn Sie einen Punkt nicht verstanden haben, gehen Sie bitte zurück und wiederholen Sie die Aktivität, denn jede Aktivität baut auf dem zuvor Gelernten auf.

## Hilfe

- Wenn Sie den Hinweis "*micro:bit* USB-Verbindungsfehler" erhalten, ziehen Sie das USB-Kabel des Rechners ab und stecken Sie es erneut ein.
- Wenn der Code nicht ordnungsgemäß ausgeführt wird, laden Sie die ursprünglichen Cybersicherheitsdateien erneut auf Ihren Rechner.
- Überprüfen Sie, ob das Texas-Symbol auf dem *micro:bit* erscheint. Falls nicht, laden Sie die Datei ti\_runtime.hex neu. Quelle: <https://education.ti.com/en/product-resources/microbit>