

# Ziel

Durch Ultraschallsensoren und Raspberry Pi Pico wird ein Ton abgespielt, wenn sich eine Person auf 50 cm nähert.

# Mehr Möglichkeit

Zuerst habe ich versucht, diese Funktion einfach von Thonny aus zu implementieren, aber nach mehr als einem Dutzend Fehlschlägen kam mir die Idee, die gleiche Funktion anders zu implementieren — über Touchdesigner. Als ich den Raspberry Pi mit der TouchDesigner–Software verband, entdeckte ich noch mehr Möglichkeiten. Ich kann über die Sensorhardware nicht nur den Ton steuern, sondern auch eine Vielzahl visueller Effekte. Dadurch wird die Grenze zwischen der realen und der virtuellen Welt aufgehoben und ich als Designer entdecke umfassendere Möglichkeiten.



# Probieren\_01

1. Raspberry Pi Pico + DFplayer–Musikplayermodul + HCSR04–Ultraschallsensor + kleiner Lautsprecher:

```
from machine import UART, Pin
import time
```

```
# Initialize UART (using Pico's GPIO TX and RX pins)
uart1 = UART(1, baudrate=9600, tx=Pin(4), rx=Pin(5)) # Change Pin(4) and Pin(5) to match
your actual connections
```

```
# Function to send a command to the DFPlayer Mini
def send_command(cmd, param1=0, param2=0):
    high_byte = param1 >> 8
    low_byte = param1 & 0xFF
    checksum = 0xFFFF - (0xFFFF & (0x7E + 0xFF + 6 + cmd + high_byte + low_byte)) + 1
    checksum_high = checksum >> 8
    checksum_low = checksum & 0xFF

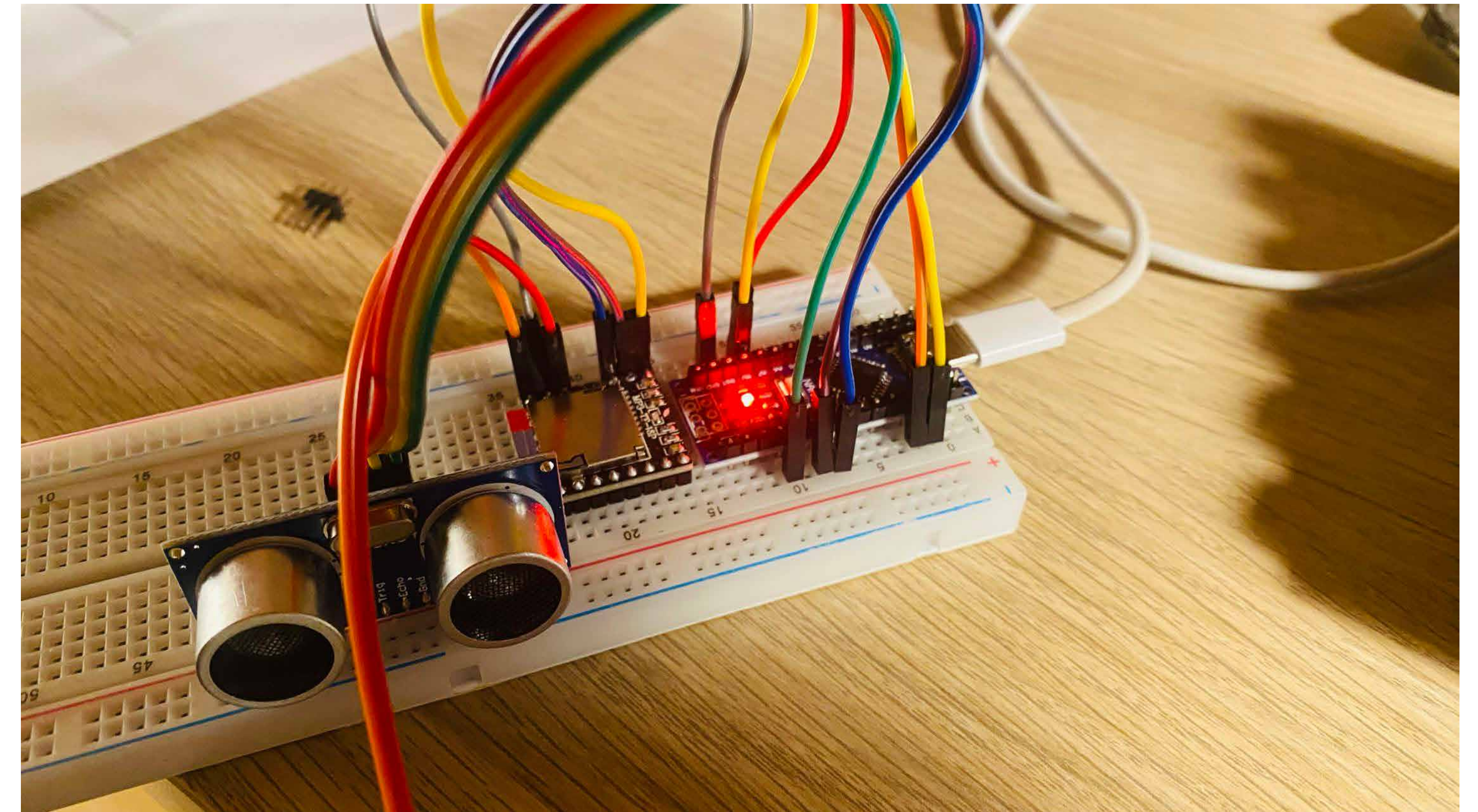
    command = bytearray([
        0x7E, 0xFF, 0x06, cmd, 0x00, high_byte, low_byte, checksum_high, checksum_low, 0xEF
    ])
    uart1.write(command)
```

```
# Initialize DFPlayer Mini
def initialize_player():
    # Set volume to 30 (range: 0–30)
    send_command(0x06, 0, 30)
    time.sleep(1)
```

```
# Main function for looping music playback
def loop_play():
    # Set loop playback mode for all files
    send_command(0x11, 0, 1) # Playback mode: 1 means loop playback for all files
    print("Starting looped music playback...")
```

```
# Main program
initialize_player()
loop_play()
```

```
# Run continuously (keep the program running to maintain playback state)
while True:
    # DFPlayer supports loop playback by itself, so there's no need to resend commands
    time.sleep(1) # Delay to reduce CPU usage of the main program
```



## Gründe für das Scheitern

Mutmaßliche Ursache: Es ist nicht sicher, ob das DFplayer–Modul mit 5 V oder 3,3 V betrieben wird. Während des Experiments könnte das DFplayer–Modul durch die Verwendung einer 5–V–Stromversorgung durchgebrannt sein.



# Probieren\_02

## 2. Steuerung von TouchDesigner über Raspberry Pi Pico

Code in Thonny zum Raspberry Pi importieren

```
import machine
import time
```

```
# Configure ultrasonic sensor pins
TRIG = machine.Pin(2, machine.Pin.OUT) # Ultrasonic sensor TRIG pin connected to GPIO2
ECHO = machine.Pin(3, machine.Pin.IN)   # Ultrasonic sensor ECHO pin connected to GPIO3
```

```
# Initialize the TRIG pin to a low level
TRIG.low()
time.sleep(2) # Wait for the sensor to stabilize
```

```
# Function to measure distance
```

```
def measure_distance():
```

```
    # Send trigger signal
```

```
    TRIG.low()
```

```
    time.sleep_us(2)
```

```
    TRIG.high()
```

```
    time.sleep_us(10)
```

```
    TRIG.low()
```

```
    # Wait for the echo signal to return
```

```
    while ECHO.value() == 0:
```

```
        pass
```

```
    start_time = time.ticks_us() # Echo start time
```

```
    while ECHO.value() == 1:
```

```
        pass
```

```
    end_time = time.ticks_us() # Echo end time
```

```
    # Calculate distance (unit: cm)
```

```
    duration = time.ticks_diff(end_time, start_time)
```

```
    distance = (duration * 0.0343) / 2 # Speed of sound = 343 m/s
```

```
    return distance
```

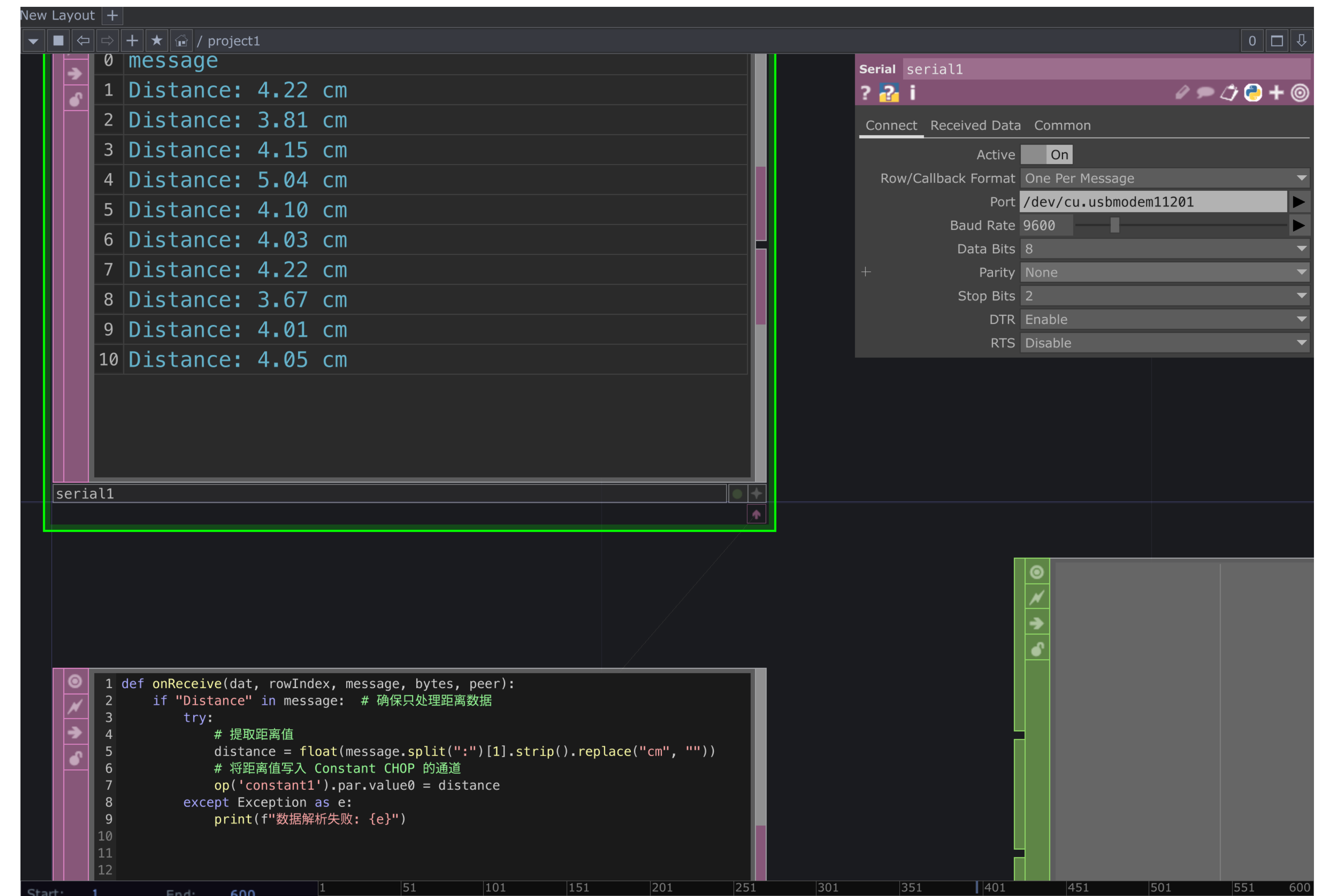
```
# Main loop: send data via the USB serial port
```

```
while True:
```

```
    distance = measure_distance() # Get distance measurement
```

```
    print(f"Distance: {distance:.2f} cm") # print() automatically adds a newline
```

```
    time.sleep(1) # Wait one second between measurements
```



Laden Sie pyserial–miniterm herunter und verwenden Sie es im Terminal.

Laden Sie pyserial–miniterm herunter, verwenden Sie es im Terminal und geben Sie den Port in touchdesigner ein.

Auf diese Weise kennt TouchDesigner die Entfernung des Ultraschallsensors und Sie können TouchDesigner verwenden, um basierend auf den Entfernungsdaten interessante Dinge zu tun.