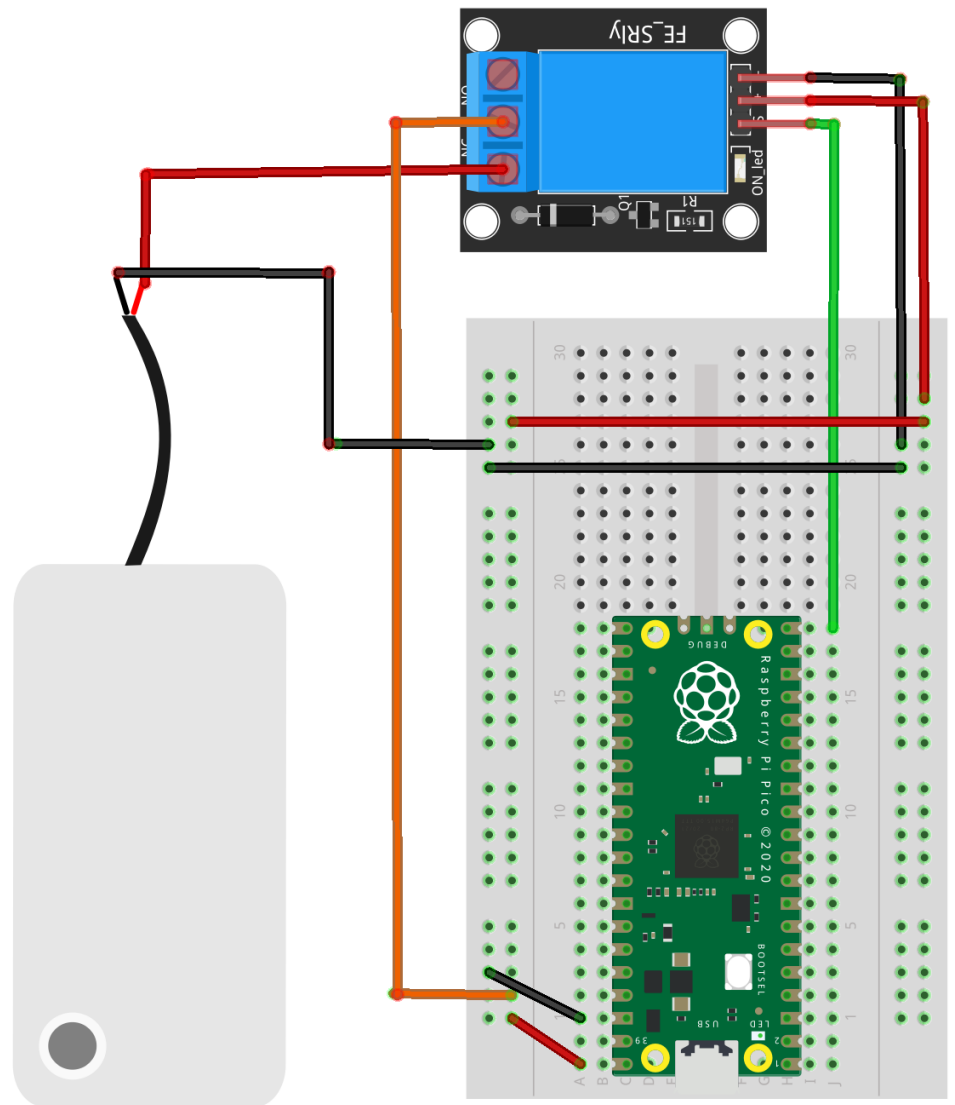


fritzing

2



fritzing

2

```

# Bibliothek
#-----
from machine import Pin
import time

# Initialisiere den GPIO-Pin für das Relais (hier GPIO 15)
#-----
relay = Pin(15, Pin.OUT)

# Funktion zum Einschalten der Pumpe
#-----
def turn_on_pump():
    relay.value(1)
    print("Pumpe eingeschaltet")

# Funktion zum Ausschalten der Pumpe
#-----
def turn_off_pump():
    relay.value(0)
    print("Pumpe ausgeschaltet")

```

```

# Hauptprogramm
#-----
try:
    while True:
        # Schalte die Pumpe für 5 Sekunden ein
        #-----
        turn_on_pump()
        time.sleep(5)

        # Schalte die Pumpe für 10 Sekunden aus
        #-----
        turn_off_pump()
        time.sleep(10)

```

```

#-----
# Warte 7 Tage (7 * 24 * 60 * 60 Sekunden)
#time.sleep(7 * 24 * 60 * 60)
#-----

```

```

# Schalte die Pumpe aus,
#wenn das Programm beendet wird
#-----
except KeyboardInterrupt:
    turn_off_pump()
    print("Programm beendet")

```

2

```

# Bibliothek
#-----
from machine import Pin
import time

# Initialisiere den GPIO-Pin für das Relais (hier GPIO 15)
#-----
relay = Pin(15, Pin.OUT)

# Funktion zum Einschalten der Pumpe
#-----
def turn_on_pump():
    relay.value(1)
    print("Pumpe eingeschaltet")

# Funktion zum Ausschalten der Pumpe
#-----
def turn_off_pump():
    relay.value(0)
    print("Pumpe ausgeschaltet")

```

```

# Hauptprogramm
#-----
try:
    while True:
        # Schalte die Pumpe für 5 Sekunden ein
        #-----
        turn_on_pump()
        time.sleep(5)

```

```

# Schalte die Pumpe für 10 Sekunden aus
#-----
        turn_off_pump()
        time.sleep(10)

```

```

#-----
# Warte 7 Tage (7 * 24 * 60 * 60 Sekunden)
#time.sleep(7 * 24 * 60 * 60)
#-----

```

```

# Schalte die Pumpe aus,
#wenn das Programm beendet wird
#-----
except KeyboardInterrupt:
    turn_off_pump()
    print("Programm beendet")

```

2