main.py ● | diagram.json ● ▾ | **Simulation**

```python
from time import sleep

print("Ready to go!\n")
print("Loop starting\n")

for i in range(10):
    print("Loop number:", i, "\n")
    sleep(0.3)

print("Loop finished")
```

⏱ 00:56.660 ⚡100%

```
Ready to go!

Loop starting

Loop number: 0

Loop number: 1

Loop number: 2

Loop number: 3

Loop number: 4

Loop number: 5

Loop number: 6

Loop number: 7

Loop number: 8

Loop number: 9

Loop finished
MicroPython v1.24.1 on 2024-11-29; Raspberry Pi Pico W with RP2040
```
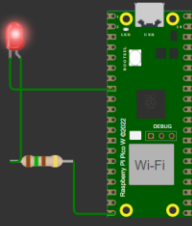
---

main.py ● | diagram.json ● ▾ | **Simulation**

```python
from machine import Pin
from time import sleep

print("Ready to go!\n")

# LEDs
onboard_led = Pin(25, Pin.OUT)
external_led = Pin(15, Pin.OUT)

# Ask name
name = input("What is your name? ")

# Logic
if name == "Clark Kent":
    print("\nYou are the Superman!\n")
else:
    print("\nYou are an ordinary person.\n")

    # Optimized NON-blocking blinking
    for i in range(5):
        onboard_led.toggle()
        external_led.toggle()
        sleep(0.3)
```

⏱ 00:37.896 ⚡99%

Wi-Fi

```
Ready to go!

What is your name? nikita

You are an ordinary person.

MicroPython v1.24.1 on 2024-11-29; Raspberry Pi Pico W with RP2040
Type "help()" for more information.
>>>
```

---

main.py ● | diagram.json ● ▾ | **Simulation**

```python
from machine import Pin
from time import sleep

print("Ready to go!\n")

# LEDs
onboard_led = Pin(25, Pin.OUT)
external_led = Pin(15, Pin.OUT)

# Ask name
name = input("What is your name? ")

# Logic
if name == "Clark Kent":
    print("\nYou are the Superman!\n")
else:
    print("\nYou are an ordinary person.\n")

    # Optimized NON-blocking blinking
    for i in range(5):
        onboard_led.toggle()
        external_led.toggle()
        sleep(0.3)
```
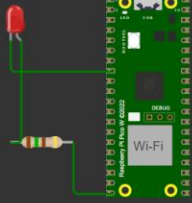
⏱ 00:16.915 ⚡100%

Wi-Fi

```
Ready to go!

What is your name? Clark Kent

You are the Superman!
```

bink led (3)

main.py

```python
from machine import Pin
from time import sleep

# Setup LEDs
led_onboard = Pin(25, Pin.OUT)   # Onboard LED
led_external = Pin(15, Pin.OUT)  # External LED (with resistor in series)

# Optimized blink function
def blink(led, delay=0.3):
    led.toggle()
    sleep(delay)

# Main loop - continuous blinking
while True:
    blink(led_onboard)
    blink(led_external)
```


hold button (4)

main.py

```python
from machine import Pin
from time import sleep

# Pin setup
button = Pin(14, Pin.IN, Pin.PULL_DOWN)  # Button input with pull-down resistor
led = Pin(15, Pin.OUT)                   # External LED output

# Main loop
while True:
    if button.value() == 1:   # Button pressed
        led.value(1)          # LED ON
    else:                     # Button released
        led.value(0)          # LED OFF
    sleep(0.05)               # Small delay to avoid bouncing
```


Traffic lights(5)

main.py

```python
from machine import Pin
from time import sleep

# Pins setup
red_led = Pin(15, Pin.OUT)
yellow_led = Pin(16, Pin.OUT)
green_led = Pin(17, Pin.OUT)
button = Pin(14, Pin.IN, Pin.PULL_DOWN)
buzzer = Pin(18, Pin.OUT)

# Function to turn off all LEDs
def all_off():
    red_led.value(0)
    yellow_led.value(0)
    green_led.value(0)

# Main loop
while true:
    # Check if button is pressed
    if button.value() == 1:
        all_off()
        red_led.value(1)       # Red ON
        buzzer.value(1)        # Buzzer ON
        sleep(2)               # Keep buzzer on for 2 seconds
        buzzer.value(0)        # Buzzer OFF
    else:
        # Normal traffic light sequence
        all_off()
        red_led.value(1)
        sleep(2)
        all_off()
        yellow_led.value(1)
        sleep(1)
        all_off()
        green_led.value(1)
        sleep(2)
```

```python
from machine import Pin
from time import sleep, ticks_ms
import urandom

# Pins setup
led = Pin(15, Pin.OUT)    # External LED
button = Pin(14, Pin.IN, Pin.PULL_DOWN)  # Button input

while True:
    print("Get ready...")
    led.value(1)  # LED ON
    # Wait random time between 2 and 5 seconds
    sleep(urandom.uniform(2,5))

    led.value(0)  # LED OFF
    start_time = ticks_ms()  # Start timing

    # Wait for button press
    while button.value() == 0:
        pass

    reaction_time = ticks_ms() - start_time
    print("Your reaction time:", reaction_time, "ms")

    sleep(2)  # Small pause before next round
```


Burglary alarm (7)

```python
from machine import Pin
from time import sleep

# PIR sensor connected to GP14
pir = Pin(14, Pin.IN)

print("Burglary alarm system started...")

while True:
    if pir.value() == 1:       # Motion detected
        print("⚠ INTRUDER DETECTED!")
    else:
        print("No movement")
    sleep(0.5)                 # Check every 0.5 seconds
```