Systemy Komputerowe w Sterowaniu i Pomiarach

Laboratorium 2

Aleksander Kruk
Michał Sobiech

30 października 2023

Praca domowa

Uruchomienie i skonfigurowanie OpenWRT dla maszyny wirtualnej i uruchomienie na qemu

Użyte komendy

`wget https://downloads.openwrt.org/releases/22.03.5/targets/armvirt/64/openwrt-22.03.5-armvirt-64-Image -O openwrt_Image`	Pobranie najnowszej wersji maszyny z armvirt
`wget https://downloads.openwrt.org/releases/22.03.5/targets/armvirt/64/openwrt-22.03.5-rootfs-ext4.img.gz -O openwrt_rootfs-ext4.img.gz`	Pobranie systemu ext4
`gzip -d openwrt_rootfs-ext4.img.gz`	Rozpakowanie systemu
`qemu-system-aarch64 -M virt -cpu cortex-a57 -nographic -smp 4 -kernel openwrt_Image -append "root=/dev/vda console=ttyAMA0" -drive file=./openwrt_rootfs-ext4.img,if=none,format=raw,id=hd0 -device virtio-blk-device,drive=hd0`	Uruchomienie quemu na żądanych ustawieniach
`opkg update`	Aktualizacja menedżera pakietów
`opkg install python3`	Pobranie pythona 3

Implementacja w języku Python oraz uruchomienie w OpenWRT / qemu dwóch programów generujących sygnał PWM

```python
import sys

def pwm_with_alternating_frequencies(duty_cycle, frequency):
    period = 1/frequency
    on_time = (duty_cycle/100) * period
    off_time = period - on_time
    return (on_time, off_time)

duty_cycle = float(sys.argv[1])
for frequency_arg in sys.argv[2:]:
    frequency = float(frequency_arg)
    on_time, off_time = pwm_with_alternating_frequencies(duty_cycle, frequency)
    print(f'Frequency: {frequency} Hz, duty cycle: {duty_cycle}, 1: {on_time:.4f} s, 0: {off_time:.4f} s')
```

Listing 1: Pwm ze zmiennymi częstotliwościami

```python
import sys

def pwm_with_alternating_duty_cycles(duty_cycle, frequency):
    period = 1/float(frequency)
    on_time = (float(duty_cycle)/100) * period
    off_time = period - on_time
    return (on_time, off_time)

frequency = float(sys.argv[1])
for duty_cycle in sys.argv[2:]:
    on_time, off_time = pwm_with_alternating_duty_cycles(duty_cycle, frequency)
    print(f'Frequency: {frequency} Hz, duty cycle: {duty_cycle}%, 1: {on_time:.4f} s, 0: {off_time:.4f} s')
```

Listing 2: Pwm ze zmiennym wypełnieniem

Komendy uruchamiające programy

```
python3 pwm_with_alternating_frequencies.py 50 10 20 30
Frequency: 10.0 Hz, duty cycle: 50.0, 1: 0.0500 s, 0: 0.0500 s
Frequency: 20.0 Hz, duty cycle: 50.0, 1: 0.0250 s, 0: 0.0250 s
Frequency: 30.0 Hz, duty cycle: 50.0, 1: 0.0167 s, 0: 0.0167 s
python3 pwm_with_alternating_duty_cycles.py 10 25 50 75
Frequency: 10.0 Hz, duty cycle: 25%, 1: 0.0250 s, 0: 0.0750 s
Frequency: 10.0 Hz, duty cycle: 50%, 1: 0.0500 s, 0: 0.0500 s
Frequency: 10.0 Hz, duty cycle: 75%, 1: 0.0750 s, 0: 0.0250 s
```