

Улази-излаз из корисничког простора

верзија 1.0

Милош Суботић

5. децембар 2025.

1 Увод

У овој вежби биће приказано како се на Линукс оперативном систему из корисничког простора управља улаз-излаз опште намене енгл. GPIO. GPIO омогућује да било која ножица енгл. pin чипа на нашој RPi платформи се може исконфигурисати да ли је улаз или излаз, прочитати или уписати логичка вредност са њега, и још по нешто. То се постиже извршавањем датотечких операција (`open()`, `read()`, `write()`) над датотеком `/dev/gpio_stream` GPIO руковаоца енгл. Driver. Помоћу GPIO руковаоца ћемо руковати DC мотором брисача енгл. Wiper motor и корачним мотором енгл. Stepper motor уписом логичке 0 или 1 на одређене ножице што ће резултовати напонским сигнаlima 0 V односно 3.3 V.

2 Радни процес

С обзиром да имамо ограничен број поставки енгл. Setup, као и лоше искуство са пребацивањем хардвера са једног RPi на други, постојаће један RPi са повезаном поставком на којем ћете извршавати своје програме, које сте искомпajлирали на личном RPi. Подсетимо се типичних корака ради провере рада мреже и интернета на RPi:

```
ifconfig
ping 8.8.8.8
ping google.com
```

Проверимо да ли је мрежа проходна према поставци:

```
ping rpi-setup-1.local
```

Конектујемо се на поставку:

```
ssh pi@rpi-setup-1.local
```

Лабораторијска вежба се може отпаковати или клонирати са

```
git clone https://github.com/MilosSubotic/OSuRV_Labs
```

и позиционирамо се у њу:

```
cd OSuRV_Labs/L5_User_Space/
```

Један судент или асистент треба да иницијално покрене драјвер на поставци:

```
cd SW/Driver/gpio_ctrl/  
make  
make start  
dmesg
```

3 DC мотор

На следећој слици је приказана хардверска поставка са DC мотором брисача.¹



Слика 1: Хардверска поставка са DC мотором брисача

На локалном RPi искомпајлирати следећи тестни програм.

```
cd SW/Test/test_gpio/  
./waf configure  
./waf build
```

Копирати искомпајлиран програм на setup RPi

```
scp ./build/test_gpio pi@rpi-setup-1.local:/home/pi/Public
```

¹TODO новија слика

На setup RPi преко ssh конекције извршити:

```
cd /home/pi/Public
./test_gpio w 2 0
./test_gpio w 3 0
./test_gpio w 4 1
./test_gpio w 2 1
```

Мотор би требао да се окреће. За окретање у другу страну:

```
./test_gpio w 2 0
./test_gpio w 3 1
./test_gpio w 4 0
./test_gpio w 2 1
```

за заустављање:

```
./test_gpio w 2 0
```

Са асистентима искоментаришите шта се десило и прођите кроз код `test_gpio.c` од 130–150 линије. `open()` отвара GPIO руковаоца. Припрема се пакет од 3 бајта за писање на GPIO са следећим вредностима:

1. 'w' за упис
2. број пина
3. вредност (0 или 1)

Позива се `write()` функција са уписом садржаја пакета и провером успешности.

4 Joypad

USB Joypad се тестира на setup RPi:

```
lsusb
ls -l /dev/input/js0
groups
jstest /dev/input/js0
```

Компајлирање у локали:

```
cd SW/Test/test_joypad/
./waf configure
./waf build
scp ./build/test_joypad pi@rpi-setup-1.local:/home/pi/Public
```

Тест ssh тестирати

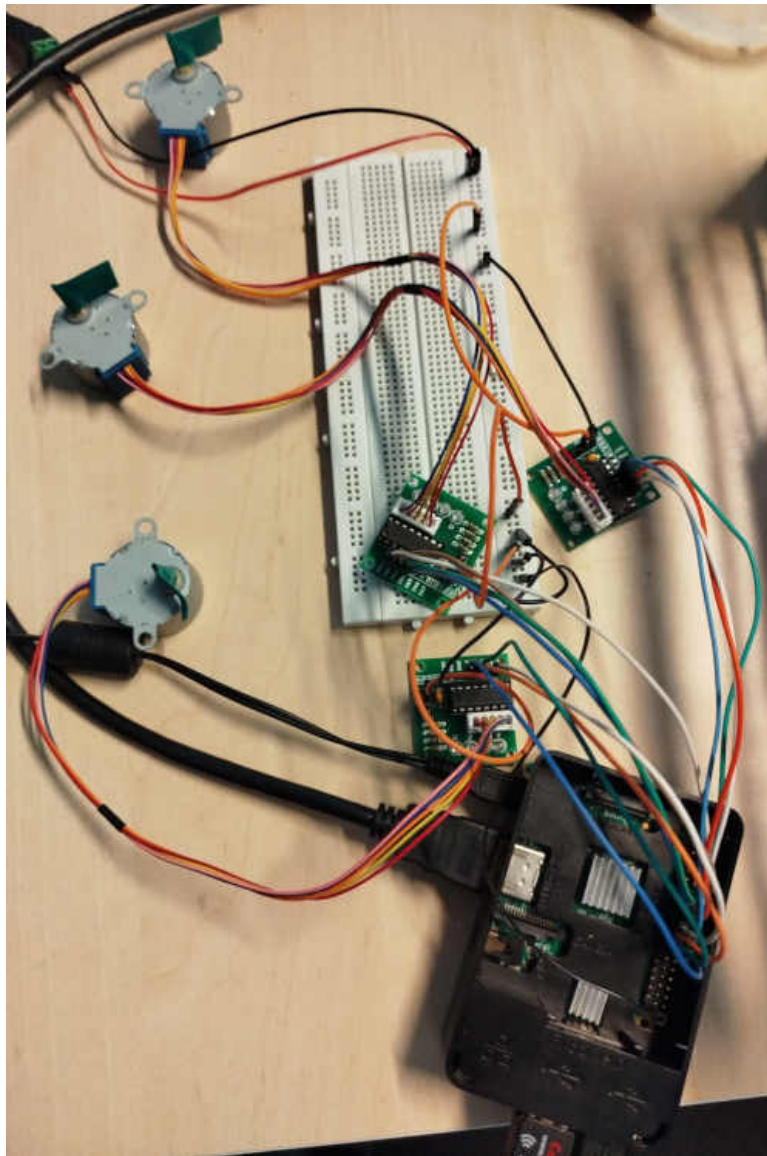
```
./test_joypad
```

Притискати дугмиће. Проћи кроз код. Новитет је коришћење `read()` функције. Joypad има предефинисан `struct js_event` као пакет за читање статуса дугмића и joystick оса.

Ако има времена пробати апликацију `SW/App/1_Joy_Wiper` која обједињује joypad и мотор и проћи кроз код.

5 Задатак: корачни мотор

На следећој слици је хардверска поставка са корачним моторима.²



Слика 2: Хардверска поставка са корачним моторима

Пример за тестирање хардверске поставке мотора на setup RPi:

```
cd SW/Test/test_gpio/  
./waf configure  
./waf build  
./test_steppers.sh
```

Мотори се непрестано врте. Поиграти се са `STEP_DURATION`. Проћи кроз код скрипте и индуктивно закључити поступак корачања мотора.

Исто што `test_steppers.sh` скрипта ради реализовати у С апликацији `SW/App/4_Stepper` водећи се примером `test_gpio.c`.

Ако има времена, може се урадити и задатак `SW/App/5_Joy_Stepper` који спаја јоурад и корачни мотор, водећи се примером `SW/App/1_Joy_Wiper`.

²TODO новија слика