

Обласи вежби: Linux руковаоци

КОРИШЋЕЊЕ ИЗ КОРИСНИЧКОГ ПРОСТОРА

Предуслови:

- RPi2 повезан на роботску руку

Увод

У овој вежби изучаваћемо како се Linux руковаоци користе из корисничког простора, док ћемо у наредним ући у детаље везане за сам руковаоц, тј. применићемо приступ од-горе-на-доле (**top-down**). У овој вежби је дат руковалац за управљање серво моторима у зглобовима роботске руке. Серво моторима се управља **PWM** дигиталним сигналом. Овај сигнал има периоду 20ms. Фактор испуне (енгл. Duty Cycle) сигнала је однос времена у ком је сигнал на логичкој јединици и периоде сигнала, и он се користи за постављање угла серво мотора. У конкретној поставки у овој вежби, фактор испуне у интервалу [2.5%, 12.5%] се мапира на углове од [-90°, 90°]. Руковалац генерише овај PWM сигнал на ножицама RPi плоче, а преко датотеке руковаоца кориснички процес поставља (тј. командује) жељени фактор испуне.

Из корисничког простора, осим тога што је преко датотеке овог руковаоца могуће постављати вредности углова серво мотора, могуће је и овај руковаоц користити у оквиру ROS (енгл. Robot Operating System) окружења и тиме управљачом (енгл. joypad) контролисати робота у једном од 3 режима:

- подешавање (енгл. jog) где се сваки зглоб робота засебно покреће,
- серво (енгл. servo) где се управља позицијом шаке у простору,
- путање (енгл. traj, trajectory) где се реализује кретање руке по задатој путањи.

Задатак

У пројекту су присутна 2 директорија: *SW* и *ROS*. У *SW* имамо руковаоц *SW/Driver/servo_ctrl* и тестну апликацију *SW/Test/test_app_servo_ctrl*. У *ROS* директорију имамо окружење за рад са роботом. У оквиру тог окружења је потребно реализовати спрегу између роботског окружења и руковаоца у датотеци *ROS/small_4dof_arm_ws/src/s4a_main/src/robot_hardware_interface.cpp* по узору како је то урађено у горе наведеној тестној апликацији.

Даље следи упутство како да се дође до решења овог задатка.

Приликом рада са руковаоцима, погодно је у једном језичку терминала (енгл. tab) покренути исписе (енгл. log) ради дебаговања коришћењем команде:

```
dmesg -w
```

Ради превођења руковаоца, у другом језичку терминала потребно је отићи у директориј `SW/Driver/servo_ctrl` и покренути команду:

```
make
```

А ради убацивања модула у језгро покренути:

```
make start
```

У пак трећем језичку отићи у директориј `SW/Test/test_app_servo_ctrl` и тамо конфигурирати и превести тестну апликацију:

```
./waf configure
```

```
./waf build
```

Као што је већ напоменуто, угао серво мотора $[-90^\circ, 90^\circ]$, односно $[-\pi/2, \pi/2]$, је мапиран на фактор испуне $[2.5\%, 12.5\%]$, односно у промилима (енгл. permille) као јединици фактора испуне руковаоца $[25\text{‰}, 125\text{‰}]$. У почетку су мотори постављени на 0° , а могу се поставити (командовати) нови углови путем команди:

```
./build/test_app_servo_ctrl w 0 25
```

```
./build/test_app_servo_ctrl w 0 125
```

Како је реализован приступ датотеци руковаоца може се видети у датотеци тестне апликације `test_app_servo_ctrl.c`, где се користе 4 функције односно 4 системска позива: `open()`, `read()`, `write()` и `close()`. Више детаља о свакој од ових функција могуће је пронаћи на интернету, нпр. коришћењем претраживача Google.

Након тестираног руковаоца и изучене тестне апликације потребно је приступити изменама прилагодног слоја између ROS окружења и руковаоца у горе поменутој датотеци `ROS/small_4dof_arm_ws/src/s4a_main/src/robot_hardware_interface.cpp`. У истој датотеци испратити `TODO` ознаке и реализовати потребан код за рад са руковаоцем. ROS команда према серво моторима и повратна спрега од серво мотора се преноси преко низова `pos_cmd` и `pos_fb`, респективно. Испратити у коду ове низове и реализовати одговарајућа прерачунавања ради прилагођења према руковаоцу.

За рад са ROS окружењем позиционирати се у корен ROS пројекта што је овде `ROS/small_4dof_arm_ws`. Ту је потребно отворити неколико језичака терминала. У једном је потребно извршити превођење путем команде:

`catkin_make`

Након тога у сваком од терминала у којима се ради нешто на ROS пројекту потребно је подесити окружење командом:


`source devel/setup.bash`

За детаље коју команду треба покренути у којем језичку терминала погледати у `flow.sh` под секцијом *Run*.

Ножице повезати према Табели 1, где је *GND* маса заједничка за све серво моторе, а *S0-S3* су PWM сигнали за сваки од серво мотора. Слика 1 приказује просторни положај ножица.

GND	S0	S1	S2	S3
	GPIO18	GPIO19	GPIO20	GPIO21
PIN6	PIN12	PIN35	PIN38	PIN40

Табела 1 Повезивање ножица за роботску руку



		Physical Pins			
GPIO#	2nd func	pin#	pin#	2nd func	GPIO#
N/A	+3V3	1	2	+5V	N/A
GPIO2	SDA1 (I2C)	3	4	+5V	N/A
GPIO3	SCL1 (I2C)	5	6	GND	N/A
GPIO4	GCLK	7	8	TXD0 (UART)	GPIO14
N/A	GND	9	10	RXD0 (UART)	GPIO15
GPIO17	GEN0	11	12	GEN1	GPIO18
GPIO27	GEN2	13	14	GND	N/A
GPIO22	GEN3	15	16	GEN4	GPIO23
N/A	+3V3	17	18	GEN5	GPIO24
GPIO10	MOSI (SPI)	19	20	GND	N/A
GPIO9	MISO (SPI)	21	22	GEN6	GPIO25
GPIO11	SCLK (SPI)	23	24	CE0_N (SPI)	GPIO8
N/A	GND	25	26	CE1_N (SPI)	GPIO7
EEPROM	ID_SD	27	28	ID_SC	EEPROM
GPIO5	N/A	29	30	GND	N/A
GPIO6	N/A	31	32	-	GPIO12
GPIO13	N/A	33	34	GND	N/A
GPIO19	N/A	35	36	N/A	GPIO16
GPIO26	N/A	37	38	N/A	GPIO20
N/A	GND	39	40	N/A	GPIO21

Слика 1 Мапа ножица за RPi2

Додатни Задатак

ROS окружење је базирано на чворовима (енгл. node) који су засебни процеси који обављају различите послове. Чворови међусобно комуницирају путем порука (енгл. message) на одређеној теми (енгл. topic), тј. тема представља канал путем којег се поруке преносе. Могуће је да постоји више чворова који објављују поруке на теми (енгл. publisher) као и више чворова претплатника на поруке на некој теми (енгл. subscriber).

У оквиру датог ROS окружења је од интереса тема под називом */s4a/motors_en*, преко које се управља дозволама за моторе. У директорију *ROS/small_4dof_arm_ws/src/s4a_teleop/src* се налази код чвора који генерише поруку на локалну тему чвора приликом притиска тастера управљача „Start“, док је у једној датотеци за покретање у директорију *ROS/small_4dof_arm_ws/src/s4a_teleop/launch* направљено мапирање поменуте локалне теме на глобалну тему */s4a/motors_en*. С друге стране, у већ поменутом модулу *robot_hardware_interface.cpp* се претплаћује на ову тему, и обављају акције које иницирају поруке примљене на тој теми.

Ради праћења порука на теми покренути команду:

```
rostopic echo /s4a/motors_en
```

Порука се може такође ручно објавити путем *rostopic pub* команде (погледати примере за тему */s4a/motors_en* у *flow.sh*)

За **додатни задатак** потребно је реализовати постављање серво мотора у режим слободног хода (енгл. freewheeling), што се постиже постављањем фактора испуне на 0. У ROS окружењу овај режим се мења на тастер 2 управљача и објављује се одговарајућа порука. Урадите реализацију претплатника у *robot_hardware_interface.cpp* по узору на тему *motor_en*.