

Обласи вежби: *Linux* руковаоци

РАД У ЈЕЗГРУ

Предуслови:

- RPi2 повезан на аутомобилску шасију и роботску руку

Увод

У овој вежби изучаваћемо рад у језгру тј. које могућности пружа, у првом реду GPIO (енгл. General Purpose Input Output), прекиде (енгл. Interrupt) и тајмере (енгл. Timer). Ова вежба се наставља на претходну, тако што ћемо реализовати додатну функционалност која омогућава извршавање процедуре (енгл. routine) где се реализује кретање шасије према задатој процедури (тј. путањи). Процедура подразумева извршавање команди које задају брзину погона, угао скретања на управљању као и пут за прећи. На овај начин могуће је самостално кретање шасије по испланираној путањи. Ова вежба ће бити усмерена на код руковаоца који се налази у директорију *SW/Driver/motor_ctrl*.

GPIO је решен са *gpio.h* библиотеком у оквиру нашег руковаоца, који скрива специфичности RPi платформе. Пре било које употребе, усмерити мултиплексер ножица (енгл. Pinmux) да буде улаз, излаз, или неку другу алтернативну функционалност иза које стоји хардверска периферија (нпр. PWM, UART...). Након тога је могуће поставити или обрисати излаз, или пак прочитати улаз. Пример рада са GPIO се налази у датотеци *servo_fb.c*.

Прекиди омогућавају да наш руковаоц буде обавештен о некој промени на некој улазној ножици. Пример рада са прекидом се налази у датотеци *servo_fb.c*. При иницијализацији потребно је

- прво поставити ножицу да буде улаз путем *gpio__steer_pinmux()* функције,
- онда затражити GPIO улаз од стране језгра функцијом *gpio_request_one()*,
- извршити конверзију GPIO броја у број прекида путем *gpio_to_irq()*
- и подесити прекид *request_irq()*, где му се задаје
 - функција повратног позива (енгл. Callback) тј. прекидна рутина (енгл. ISR - Interrupt Service Routine), која ће се позвати у случају промене улазне ножице, што је *fb_isr()* у случају *servo_fb.c*,
 - Показивач на податак који ће бити прослеђен прекидној рутини као аргумент *data*, који се после може конвертовати и користити за приступ

пољима структуре специфичне за ножицу (пример објектно-оријентисаног програмирања у C-у).

У самој прекидној рутини потребно је:

- обавити потребан задатак и
- вратити `IRQ_HANDLED` на крају рутине.

Тајмер омогућава да се неки задатак изврши након одређеног времена. Пример рада са тајмером се налази у `sw_pwm.c` датотеци. При раду са тајмерима користи се `hrtimer` библиотека из језгра. Тајмер се иницијализује тако што се:

- изврши сама иницијализација структуре типа `struct hrtimer` путем функције `hrtimer_init()`.
- Постави функција повратног позива (енгл. Callback) у горепоменућу структуру у поље `function`, коју тајмер позва након истека задатог интервала времена, што је `timer_callback()` у случају датотеке `sw_pwm.c`.
- покрене тајмер са задатим интервалом `hrtimer_start()`,

У самој функцији повратног позива могуће је:

- обавити задатак који је већ потребан,
- обновити тајмер на нови интервал и тиме постићи периодичност путем функције `hrtimer_forward_now()`.

Задаци

Први задатак је коришћењем GPIO библиотеке реализовати функцију `bldc_set_dir()`, тако да се у зависности од параметра `dir` постави (CW) или обриси (CCW) ножица за правац окретања BLDC мотора. Такође сачувати `dir` у променљиву `dir` низа структура типа `bldc_t` за касније задатке. У `методи за иницијализацију`, поставити горепоменућу ножицу за излаз путем функције `gpio_steer_pinmux()`.

Други задатак је направити софтверски PWM за BLDC погонски мотор помоћу тајмера. Ове функционалности треба реализовати у оквиру датотеке `SW/Driver/motor_ctrl/bldc.c`. Водити се кодом из `sw_pwm.c` датотеке. У тој реализацији, тајмер наизменично чека између укљученог и искљученог временског интервала, чији се однос мења фактором испуне, док је збир та два интервала једнак периоди PWM сигнала.

У случају BLDC-а, поједноставити функционалност коришћењем фиксне периоде од 200 микро секунди. Водити рачуна да је из функције `main.c` потребно позвати одговарајућу функцију декларисану у `bldc.h`, како би се поставио фактор испуне, као и превезати PWM сигнал BLDC мотора на PWM (SW) ножицу.

Практично, целокупан код за генерисање прекида из `sw_pwm.c` се може преузети, с том разликом да се у функцији `set_intervals()` `on` и `off` времена не рачунају на основу параметара `modulo` и `threshold`, који се постављају у функцијама `sw_pwm__set_modulo()` или `sw_pwm__set_threshold()`, већ на основу `duty_per mille` који се поставља у `bl dc__set_duty()`.

Трећи задатак је избројати импулсе са енкодера BLDC мотора путем прекида. Користити 64-битни бројач `pulse_cnt` који је `atomic64_t` типа јер му се приступа у функцији `bl dc__get_pulse_cnt()` што је други контекст од сервисне рутине где се мења. Овај атомски бројач увећавати путем функције `atomic64_add()`. Водити рачуна о правцу обртаја мотора (променљива `dir`) који утиче да ли ће се бројач импулса увећавати или умањивати.

За детаље око **покретања ROS** окружења погледати команде у `ROS/arm_and_chassis_ws/flow.sh` под секцијом *Run chassis*.


Ножице повезати према Табели 1 и Табели 2. Слика 1 приказује просторни положај ножица.

Функција	5V	GND	S0	S1	S2	S3
Боја	Црвена	Браон	Наранџаста	Наранџаста	Наранџаста	Наранџаста
SW			GPIO 18	GPIO 19	GPIO 20	GPIO 21
Ножица		PIN 6	PIN 12	PIN 35	PIN 38	PIN 40

Табела 1 Повезивање ножица за роботску руку

	Погон - BLDC							Управљање - Серво			
Функција	24V	GND	PG	DIR	PWM (S0)	PWM (S4)	PWM (SW)	5V	GND	S1	S5
Боја	Црвена	Црна	Жута	Бела	Плава	Плава	Плава	Црвена	Браон	Наранџаста	Наранџаста
SW			GPIO 16	GPIO 17	GPIO 18	GPIO 22	GPIO 24			GPIO 19	GPIO 23
Ножица		PIN 6	PIN 36	PIN 11	PIN 12	PIN 15	PIN 18		PIN 6	PIN 35	PIN 16

Табела 2 Повезивање ножица за шасију



		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 Маја ножица за RPi2