

Компоненте и кола снаге

ЛАБОРАТОРИЈСКЕ ВЕЖБЕ, 2022/23.

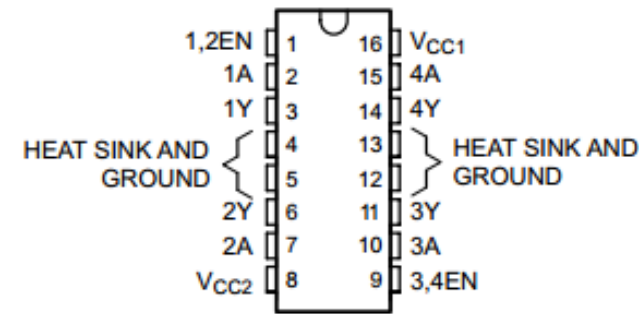
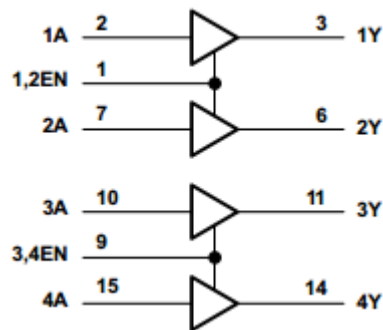
МИЛОШ МАРЈАНОВИЋ

МИЛАН СТОЈАНОВИЋ

ПЕТИ ТЕРМИН

L293D – *half-H driver* - карактеристике

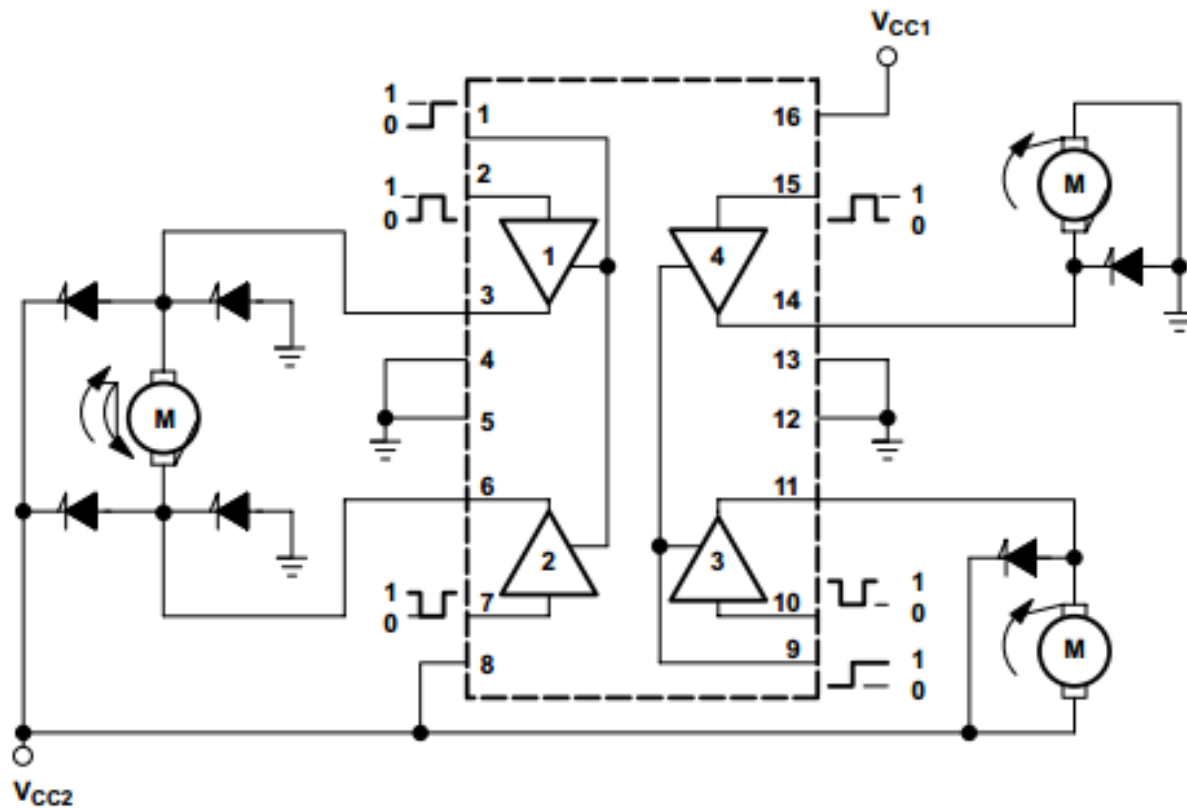
- Драјвер индуктивних оптерећења као што су релеји, соленоиди, DC и биполарни *stepper* мотор, итд.
- Сваки излаз је са интерним Дарлингтоновим паром транзистора (струја до 600 mA)
- Рад драјвера се омогућава у пару (1 и 2, 3 и 4)



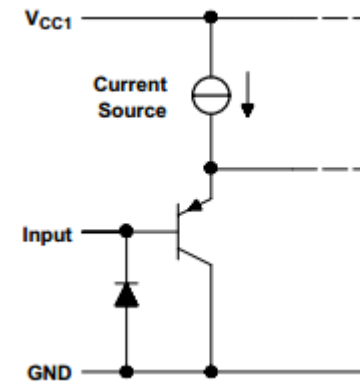
Pin Functions

PIN		TYPE	DESCRIPTION
NAME	NO.		
1,2EN	1	I	Enable driver channels 1 and 2 (active high input)
<1:4>A	2, 7, 10, 15	I	Driver inputs, noninverting
<1:4>Y	3, 6, 11, 14	O	Driver outputs
3,4EN	9	I	Enable driver channels 3 and 4 (active high input)
GROUND	4, 5, 12, 13	—	Device ground and heat sink pin. Connect to printed-circuit-board ground plane with multiple solid vias
V _{CC1}	16	—	5-V supply for internal logic translation
V _{CC2}	8	—	Power VCC for drivers 4.5 V to 36 V

L293D – *half-H driver* - структура

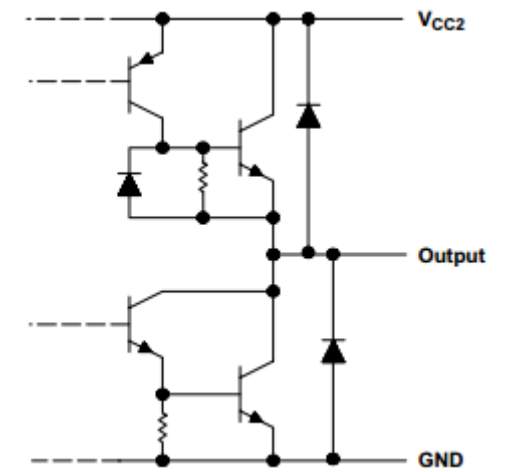


Output diodes are internal in L293D.

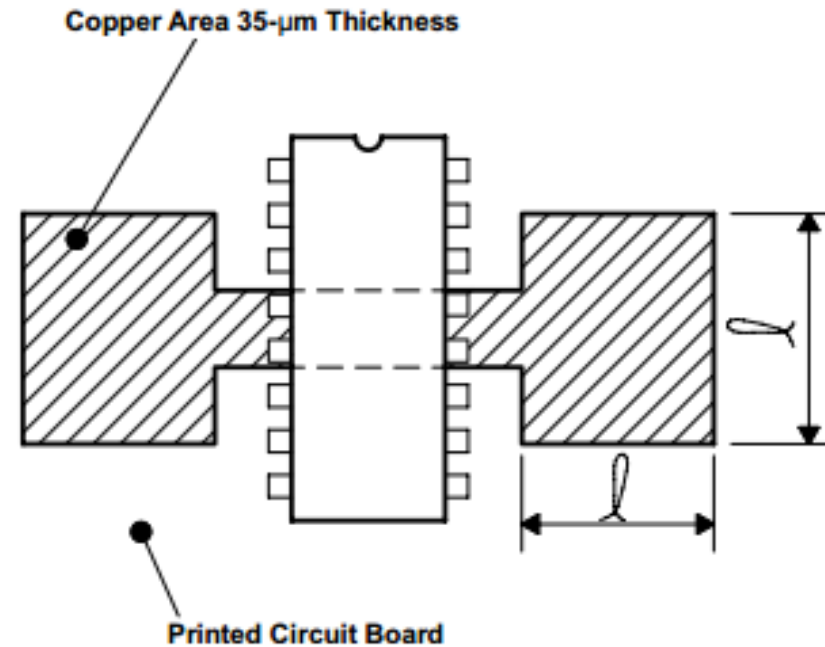
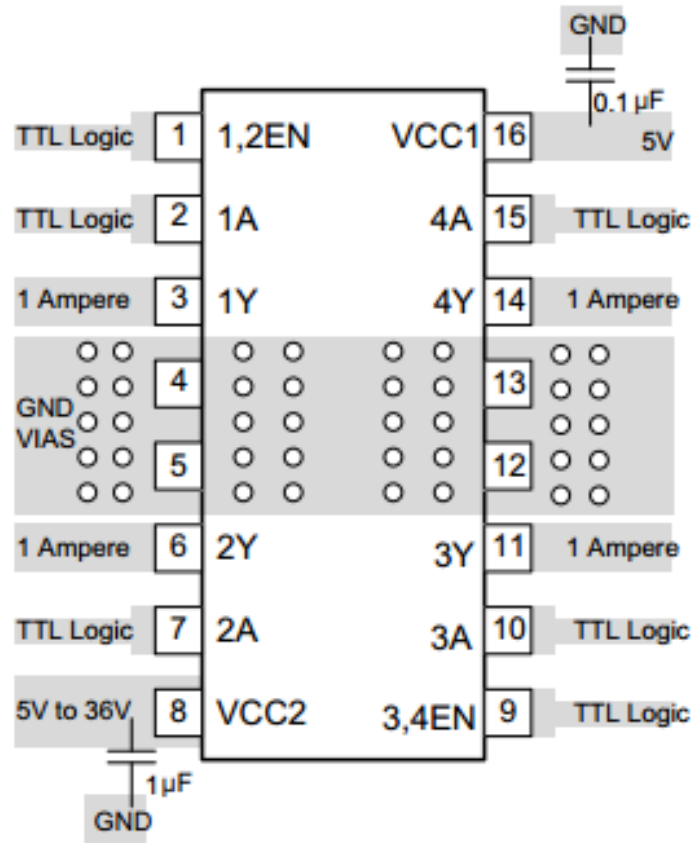


Улазно коло L293

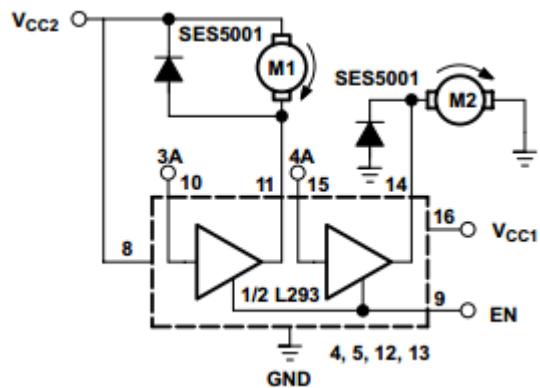
Излазно коло L293



L293D на PCB-у

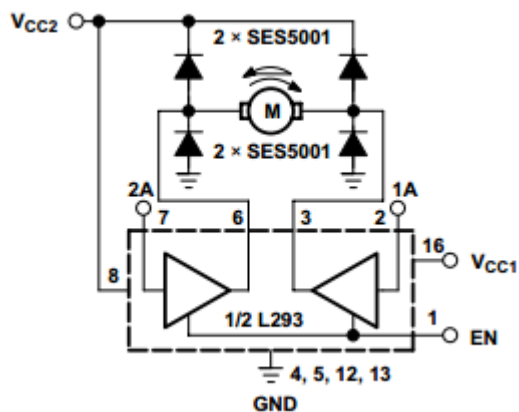


L293D - примена



Контрола унидирекционих DC мотора

EN	3A	M1 ⁽¹⁾	4A	M2
H	H	Fast motor stop	H	Run
H	L	run	L	Fast motor stop
L	X	Free-running motor stop	X	Free-running motor stop



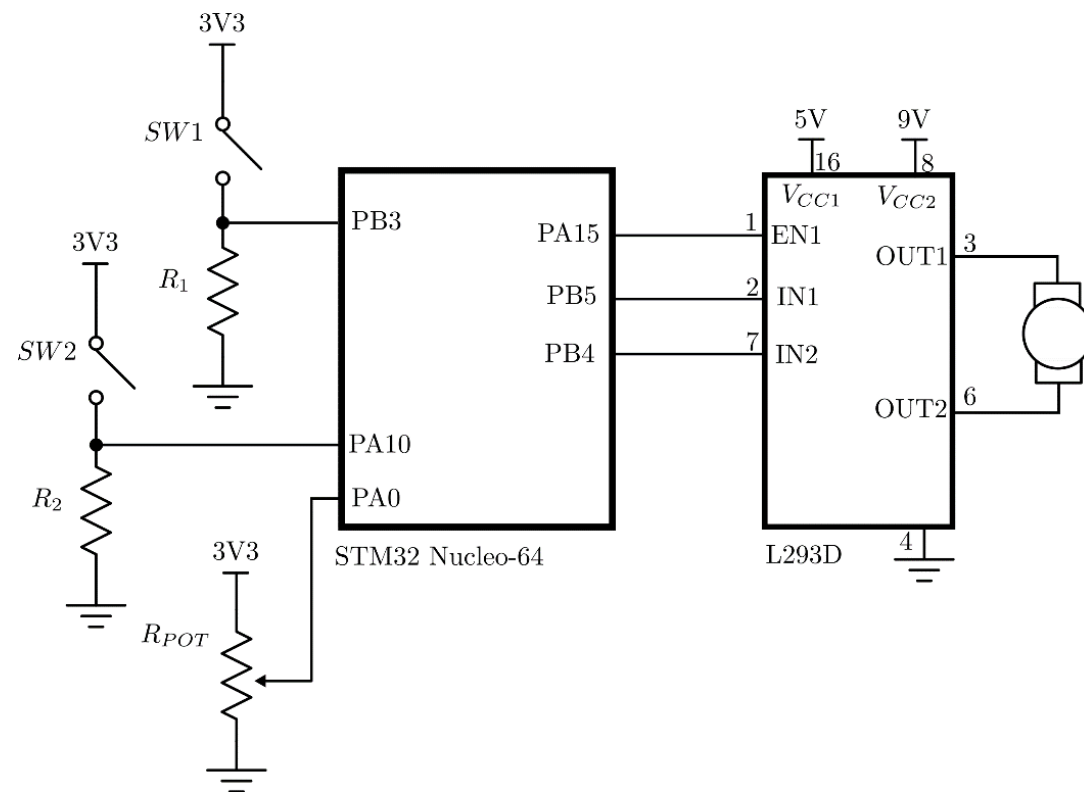
Контрола бидирекционих DC мотора

EN	1A	2A	FUNCTION ⁽¹⁾
H	L	H	Turn right
H	H	L	Turn left

EN	1A	2A	FUNCTION ⁽¹⁾
H	L	L	Fast motor stop
H	H	H	Fast motor stop
L	X	X	Free-running motor stop

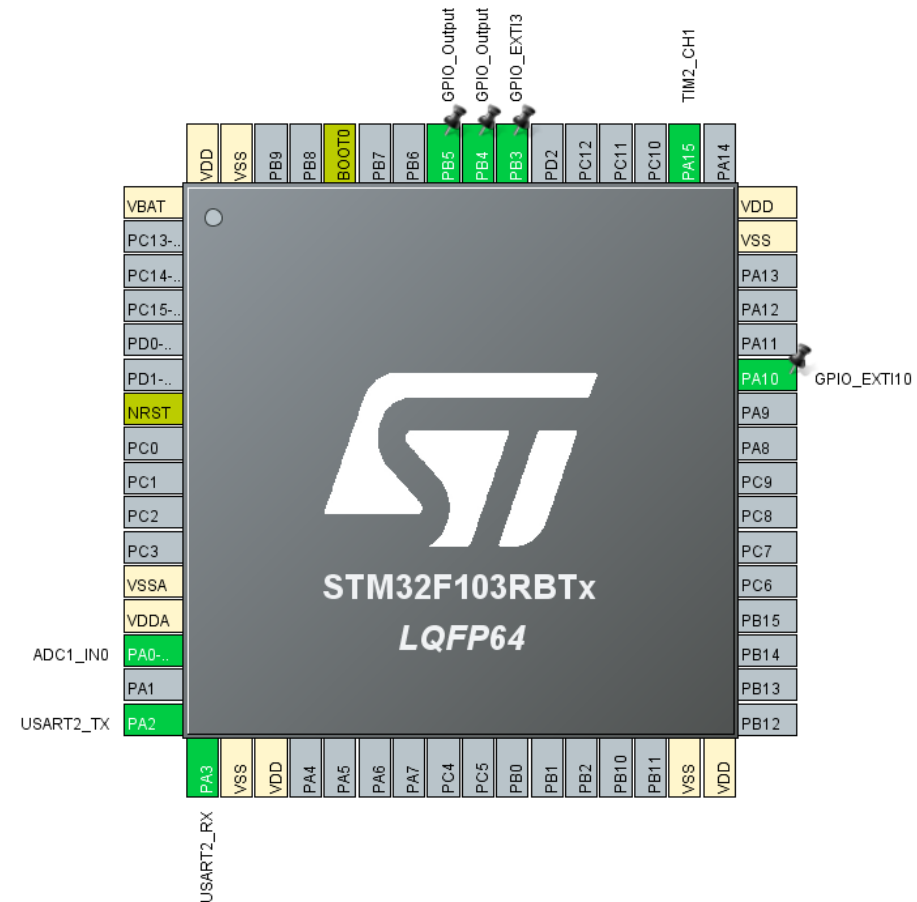
Пример 1 - Nucleo + L293D

Реализовати систем за управљање радом DC мотора коришћењем драјвера L293D. Систем се састоји из Nucleo развојног окружења на које су повезана два тастера у *pull-down* конфигурацији, потенциометар и драјвер L293D. Један тастер (SW1) служи за дозволу рада – на сваки притисак се мења стање *on/off*. Други тастер (SW2) служи за одређивање смера окретања – на сваки притисак мења се смер окретања мотора. Брзина окретања мотора одређена је положајем потенциометра. Написати програм у C-у који имплементира описане захтеве.



Пример 1 - Nucleo + L293D

- Тастер за дозволу рада (SW1) читава се преко пина PB3, коришћењем интерапта – GPIO_EXTI3
- Тастер за промену смера мотора (SW2) читава се преко пина PA10, коришћењем интерапта – GPIO_EXTI10
- Потенциометар за контролу брзине мотора повезује се на пин PA0, користи се ADC1 – ADC1_IN0
- Дозвола рада L293D врши се преко PA15, који треба да се понаша као PWM (реализује се помоћу тајмера) – TIM2_CH1
- Контролни сигнали L293D шаљу се на пинове PB5 и PB4 – GPIO_Output
- UART комуникација преко PA2 (Tx) и PA3 (Rx)



Пример 1 - Nucleo + L293D

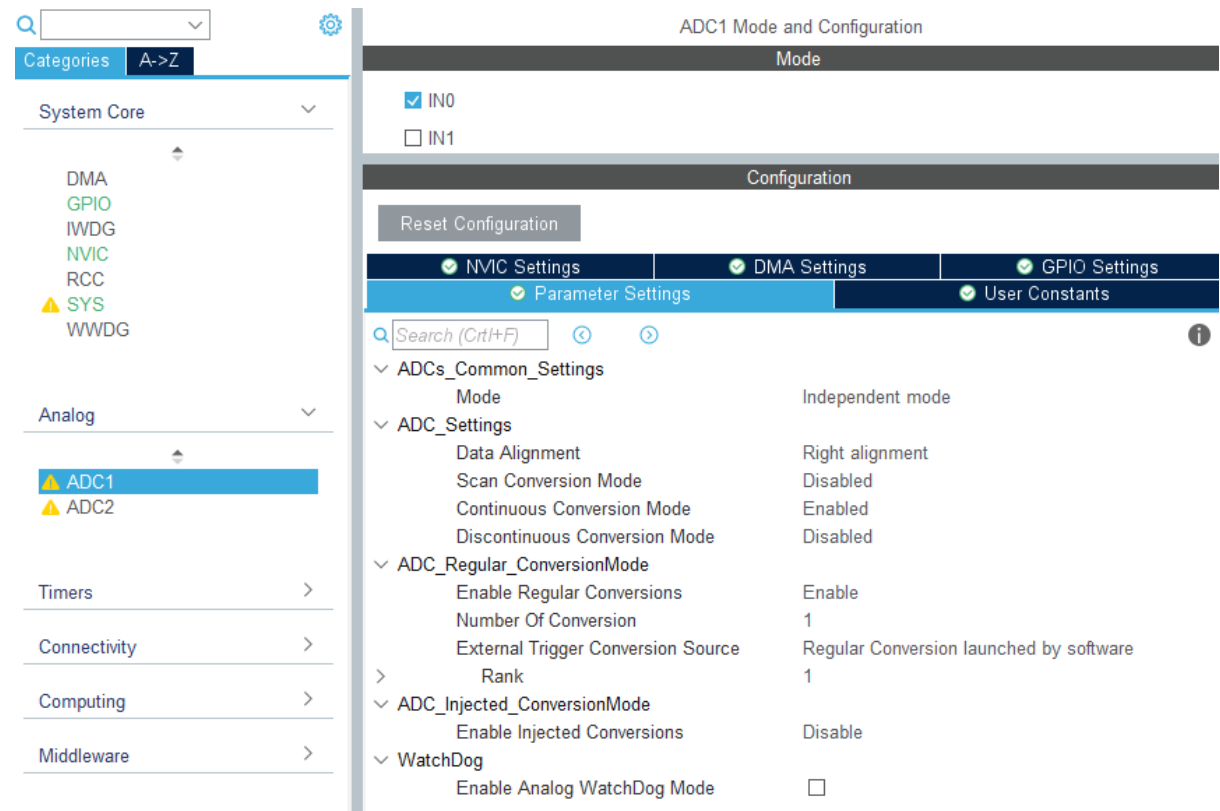
- Улазни пинови у GPIO моду – *external interrupt mode with rising edge trigger detection*
- Излазни пинови у GPIO моду – *output push pull*

The screenshot shows the STM32CubeMX software interface. On the left, the 'System Core' tree is expanded to 'GPIO'. On the right, the 'Group By Peripherals' dropdown is set to 'GPIO'. Below this, there are tabs for 'GPIO', 'ADC', 'TIM', 'USART', and 'NVIC', with 'GPIO' selected. A search bar labeled 'Search Signals' is present. A checkbox 'Show only Modified Pins' is unchecked. The main table displays the configuration for four pins:

Pin Na...	Signal on ...	GPI...	GPIO mode	U...	...
PA10	n/a	n/a	External Interrupt Mode with Rising edge trigger detection	...	n/a		<input type="checkbox"/>
PB3	n/a	n/a	External Interrupt Mode with Rising edge trigger detection	...	n/a		<input type="checkbox"/>
PB4	n/a	Low	Output Push Pull		<input type="checkbox"/>
PB5	n/a	Low	Output Push Pull		<input type="checkbox"/>

Пример 1 - Nucleo + L293D

- Подешавање ADC-а: селектовањем канала 0 (IN0) означава се пин A0 где треба повезати потенциометар.
- Укључује се континуални мод конверзије (*continuous conversion mode*). Подесити жељени *sampling time*: на пример, 239,5 циклуса.



The screenshot shows the STM32CubeMX configuration tool. On the left, the 'Analog' section is expanded, and 'ADC1' is selected. The main window displays the 'ADC1 Mode and Configuration' settings. The 'Mode' section shows 'IN0' selected. The 'Configuration' section includes a 'Reset Configuration' button and tabs for 'NVIC Settings', 'DMA Settings', 'GPIO Settings', 'Parameter Settings', and 'User Constants'. The 'Parameter Settings' tab is active, showing a search bar and a list of settings:

Setting	Value
ADCs_Common_Settings	
Mode	Independent mode
ADC_Settings	
Data Alignment	Right alignment
Scan Conversion Mode	Disabled
Continuous Conversion Mode	Enabled
Discontinuous Conversion Mode	Disabled
ADC_Regular_ConversionMode	
Enable Regular Conversions	Enable
Number Of Conversion	1
External Trigger Conversion Source	Regular Conversion launched by software
Rank	1
ADC_Injected_ConversionMode	
Enable Injected Conversions	Disable
WatchDog	
Enable Analog WatchDog Mode	<input type="checkbox"/>

Пример 1 - Nucleo + L293D

- Подешавање TIM2 као PWM генератора.
 - Подесити интерни тактни сигнал (*Clock source: Internal Clock*) и поставити за канал 1 (*Channel 1: PWM Generation CH1*)
 - Подесити жељену фреквенцију PWM сигнала и иницијалну вредност *duty cycle*-а:
 - $Prescaler = 8000-1$
 - $CounterPeriod = 100-1$
 - Период PWM-а: 100 ms
 - Иницијални $duty\ cycle = pulse/CounterPeriod = 25\%$

NVIC Settings		DMA Settings	
Parameter Settings		User Constants	
NVIC Interrupt Table	Enabled		
TIM2 global interrupt	<input checked="" type="checkbox"/>	0	

The screenshot displays the STM32CubeMX configuration tool. On the left, the 'Timers' category is expanded, and 'TIM2' is selected. The main panel shows the 'TIM2 Mode and Configuration' settings. The 'Mode' section includes: Slave Mode (Disable), Trigger Source (Disable), Clock Source (Internal Clock), Channel1 (PWM Generation CH1), and Channel2 (Disable). The 'Configuration' section includes a 'Reset Configuration' button and tabs for 'NVIC Settings', 'DMA Settings', 'GPIO Settings', 'Parameter Settings', and 'User Constants'. The 'Parameter Settings' tab is active, showing the following parameters:

Configure the below parameters :	
Counter Settings	
Prescaler (PSC - 16 bits value)	8000-1
Counter Mode	Up
Counter Period (AutoReload Register - 1...	100-1
Internal Clock Division (CKD)	No Division
auto-reload preload	Disable
Trigger Output (TRGO) Parameters	
Master/Slave Mode (MSM bit)	Disable (Trigger input effect not delayed)
Trigger Event Selection	Reset (UG bit from TIMx_EGR)
PWM Generation Channel 1	
Mode	PWM mode 1
Pulse (16 bits value)	25
Output compare preload	Enable
Fast Mode	Disable
CH Polarity	High

Пример 1 - Nucleo + L293D

- Активирати UART2 у асинхронном моду.
- Омогућити екстерне интерапте на *line 3* и *line 15:10*

USART2 Mode and Configuration

Mode

Mode Asynchronous

Hardware Flow Control (RS232) Disable

Configuration

Reset Configuration

NVIC Settings DMA Settings GPIO Settings

Parameter Settings User Constants

Configure the below parameters :

Search (Ctrl+F)

Basic Parameters

Baud Rate 115200 Bits/s

Word Length 8 Bits (including Parity)

Parity None

Stop Bits 1

Advanced Parameters

Data Direction Receive and Transmit

Over Sampling 16 Samples

NVIC Mode and Configuration

Configuration

NVIC Code generation

Priority Group 4 bits for pre... Sort by Preemption Priority and Sub Priority

Search Show only enabled interrupts Force DMA channels Interrupts

NVIC Interrupt Table	Enabled	Preemption Priority	Sub Priority
Non maskable interrupt	<input checked="" type="checkbox"/>	0	0
Hard fault interrupt	<input checked="" type="checkbox"/>	0	0
Memory management fault	<input checked="" type="checkbox"/>	0	0
Prefetch fault, memory access fault	<input checked="" type="checkbox"/>	0	0
Undefined instruction or illegal state	<input checked="" type="checkbox"/>	0	0
System service call via SWI instruction	<input checked="" type="checkbox"/>	0	0
Debug monitor	<input checked="" type="checkbox"/>	0	0
Pendable request for system service	<input checked="" type="checkbox"/>	0	0
Time base: System tick timer	<input checked="" type="checkbox"/>	0	0
PVD interrupt through EXTI line 16	<input type="checkbox"/>	0	0
Flash global interrupt	<input type="checkbox"/>	0	0
RCC global interrupt	<input type="checkbox"/>	0	0
EXTI line3 interrupt	<input checked="" type="checkbox"/>	0	0
ADC1 and ADC2 global interrupts	<input type="checkbox"/>	0	0
TIM2 global interrupt	<input checked="" type="checkbox"/>	0	0
USART2 global interrupt	<input type="checkbox"/>	0	0
EXTI line[15:10] interrupts	<input checked="" type="checkbox"/>	0	0

Пример 1 - Nucleo + L293D

```
/* USER CODE BEGIN Includes */
#include <stdbool.h>
/* USER CODE END Includes */
/* USER CODE BEGIN 0 */
volatile bool enableFleg=false, smerFleg=false;
int adcVrednost,duty;
/* USER CODE END 0 */
/* USER CODE BEGIN 2 */
    HAL_ADC_Start(&hadc1);    //стартовање ADC-а у континуалном моду
    HAL_TIM_PWM_Start(&htim2,TIM_CHANNEL_1);    //стартовање тајмера PWM
/* USER CODE END 2 */
```

Пример 1 - Nucleo + L293D

```
/* USER CODE BEGIN 3 */
    adcVrednost=HAL_ADC_GetValue(&hadc1); //читамо вредност ADC-а са потенциометра
    duty=adcVrednost*100/4095; //дефинишемо duty cycle PWM-а, тиме одређујемо брзину мотора*
    if(enableFleg) //ако је рад мотора дозвољен
    {
        htim2.Instance->CCR1=duty; //прослеђујемо "брзину мотора"* у CCR1 регистар
        if(smerFleg) //у зависности од изабраног смера, подешавамо контролне сигнале
        {
            HAL_GPIO_WritePin(GPIOB,GPIO_PIN_5,GPIO_PIN_SET);
            HAL_GPIO_WritePin(GPIOB,GPIO_PIN_4,GPIO_PIN_RESET);
            HAL_UART_Transmit(&huart2,"smer1\r\n",strlen("smer1\r\n"),10);
        }
        else
        {
            HAL_GPIO_WritePin(GPIOB,GPIO_PIN_5,GPIO_PIN_RESET);
            HAL_GPIO_WritePin(GPIOB,GPIO_PIN_4,GPIO_PIN_SET);
            HAL_UART_Transmit(&huart2,"smer2\r\n",strlen("smer2\r\n"),10);
        }
    }
}
```

Пример 1 - Nucleo + L293D

```
else
{
    htim2.Instance->CCR1=0;    //ако не постоји дозвола рада мотора, искључујемо га
    HAL_UART_Transmit(&huart2,"stop\r\n",strlen("stop\r\n"),10);
}
/* USER CODE BEGIN 4 */
void EXTI3_IRQHandler(void)    //обрисати ISR из stm32f1xx_it.c
{
    /* USER CODE BEGIN EXTI3_IRQn 0 */
    for(long i=0;i<100000;i++);    //debouncing
    enableFleg=!enableFleg;    //постављање flag-а за дозволу рада
    /* USER CODE END EXTI3_IRQn 0 */
    HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_3);
    /* USER CODE BEGIN EXTI3_IRQn 1 */

    /* USER CODE END EXTI3_IRQn 1 */
}
```

Пример 1 - Nucleo + L293D

```
void EXTI15_10_IRQHandler(void)    //обрисати ISR из stm32f1xx_it.c
{
    /* USER CODE BEGIN EXTI15_10_IRQn 0 */
        for(long i=0;i<100000;i++);        //debouncing
        smerFleg=!smerFleg;                //постављање флаг-а за смер мотора
    /* USER CODE END EXTI15_10_IRQn 0 */
    HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_10);
    /* USER CODE BEGIN EXTI15_10_IRQn 1 */

    /* USER CODE END EXTI15_10_IRQn 1 */
}
/* USER CODE END 4 */
```