**VRS 2. Cvičenie – Zoltán Czinege, Tomáš Nagy**

1. **Úloha**

Na pine PA5 máme pripojenú zelenú LED. Na to aby sme túto LED mohli ovládať potrebujeme nastaviť príslušnú GPIO perifériu a jej príslušný port následovne:

GPIO mode => OUT

Output type => PushPull

PullUp PullDown => UP

GPIO speed => 40MHz

Riešenie:

/\* uloha 1 \*/

RCC\_AHBPeriphClockCmd(RCC\_AHBPeriph\_GPIOA, *ENABLE*);

GPIO\_InitTypeDef gpioInitStruc;

gpioInitStruc.GPIO\_Mode = *GPIO\_Mode\_OUT*;

gpioInitStruc.GPIO\_OType = *GPIO\_OType\_PP*;

gpioInitStruc.GPIO\_PuPd = *GPIO\_PuPd\_UP*;

gpioInitStruc.GPIO\_Pin = GPIO\_Pin\_5;

gpioInitStruc.GPIO\_Speed = *GPIO\_Speed\_40MHz*;

GPIO\_Init(GPIOA, &gpioInitStruc);

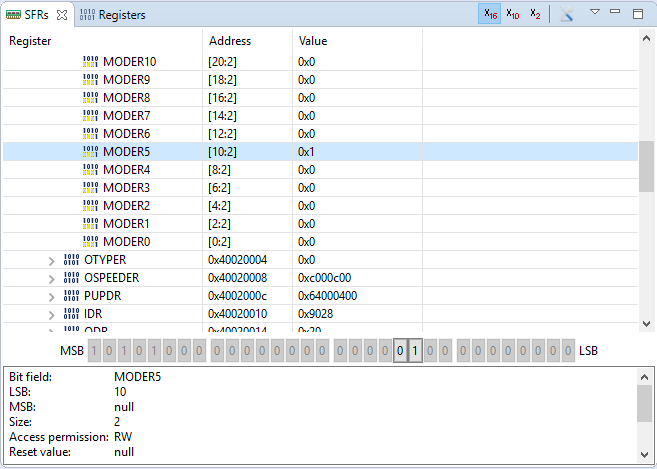
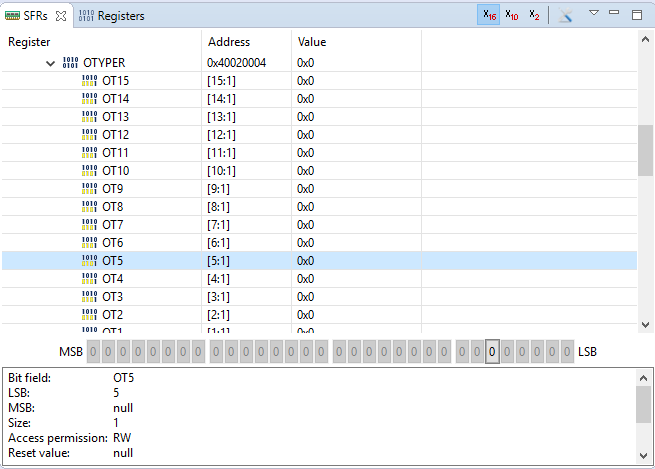
Samozrejme, že LED budeme chcieť aj zapnúť, vypnúť alebo prepnuť jej stav.

/\* zapinanie a vypinanie \*/

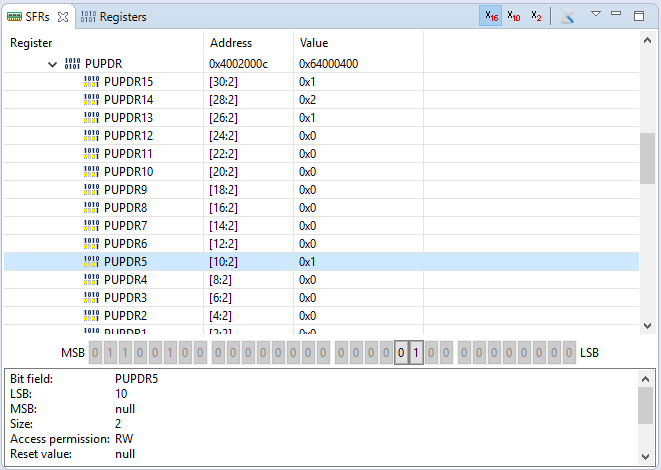
GPIO\_SetBits(GPIOA, GPIO\_Pin\_5);

GPIO\_ResetBits(GPIOA, GPIO\_Pin\_5);

GPIOA->MODER



GPIOA->OTYPER



GPIOA->PUPDR

1. **Úloha**

Na pine PC13 máme pripojené tlačidlo, ktorého stavy chceme snímať. Nastavenia periférie bude v tomto prípade následovné:

GPIO mode => IN

Output type => PushPull

PullUp PullDown => NOPULL

Riešenie:

/\* uloha 2 \*/

RCC\_AHBPeriphClockCmd(RCC\_AHBPeriph\_GPIOC, *ENABLE*);

gpioInitStruc.GPIO\_Mode = *GPIO\_Mode\_IN*;

gpioInitStruc.GPIO\_OType = *GPIO\_OType\_PP*;

gpioInitStruc.GPIO\_PuPd = *GPIO\_PuPd\_NOPULL*;

gpioInitStruc.GPIO\_Pin = GPIO\_Pin\_13;

GPIO\_Init(GPIOC, &gpioInitStruc);

**int** button;

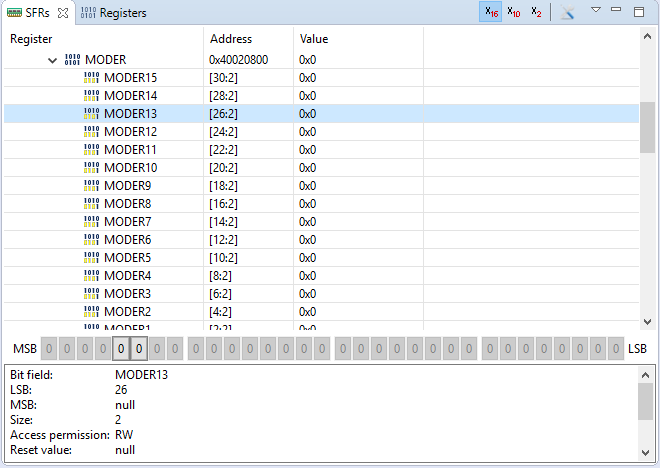
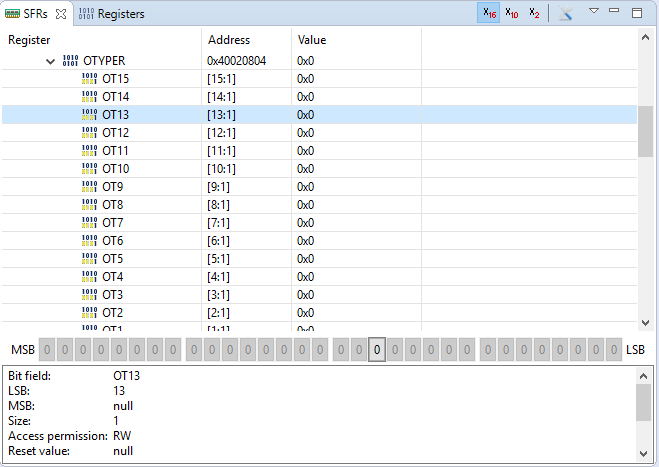
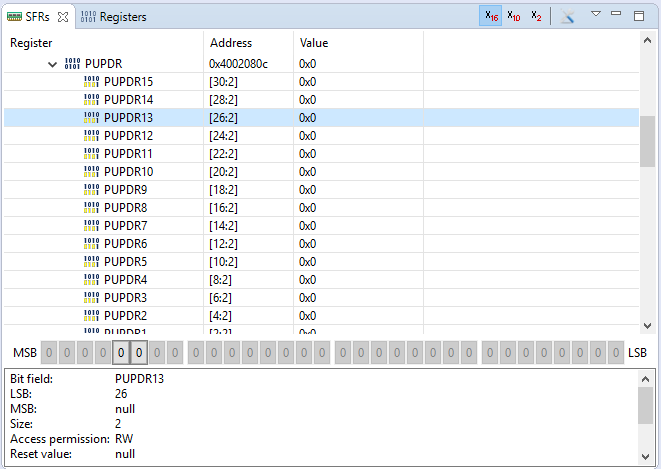
/\* uloha 2 v cykle while\*/

**if** ((GPIO\_ReadInputDataBit(GPIOC, GPIO\_Pin\_13)) == 0)

button = 1;

**else** **if** ((GPIO\_ReadInputDataBit(GPIOC, GPIO\_Pin\_13)) == 1)

button = 0;



GPIOC->MODER

GPIOC->OTYPER

GPIOC->PUPDR

1. **Úloha**

* Vytvorte program, v ktorom bude blikať LED vo Vami vybranom intervale, tak aby bolo blikanie viditeľné. Na časovanie môžete použiť for cyklus.

/\* uloha 3 prva cast \*/

GPIO\_SetBits(GPIOA, GPIO\_Pin\_5);

**for** (**int** c=1; c<= 1000; c++){

**for** (**int** d=1; d<= 1000; d++)

{}

}

GPIO\_ResetBits(GPIOA, GPIO\_Pin\_5);

**for** (**int** c=1; c<= 1000; c++){

**for** (**int** d=1; d<= 1000; d++)

{}

}

* Vytvorte program, ktorý bude sledovať stav tlačidla a tento stav bude zobrazovať na LED.

/\* uloha 3 druha cast \*/

**if** ((GPIO\_ReadInputDataBit(GPIOC, GPIO\_Pin\_13)) == 0){

GPIO\_SetBits(GPIOA, GPIO\_Pin\_5);

}

**else** **if** ((GPIO\_ReadInputDataBit(GPIOC, GPIO\_Pin\_13)) == 1) {

GPIO\_ResetBits(GPIOA, GPIO\_Pin\_5);

}

* Vytvorte program, ktorý po každom stlačení tlačidla zmení stav LED (zo svieti na nesvieti a naopak). Nezabudnite, že stlačenie tlačidla znamená, že tlačidlo prejde stavmi z 0 - 1 - 0. Taktiež si je dobre uvedomiť, že počas stlačenia tlačidla nastávajú prechodové deje zákmitu kontaktu a preto treba ošetriť aj tento problém. Google je tu pre Vás.

/\* uloha 3 tretia cast \*/

**for** (**int** cc=1; cc<= 5; cc++)

{

**if** ((GPIO\_ReadInputDataBit(GPIOC, GPIO\_Pin\_13)) == 0) {

temp\_inc++;

}

}

**if** (temp\_inc > 2)

{

inc++;

temp\_inc = 0;

}

**if** (inc == 1)

GPIO\_SetBits(GPIOA, GPIO\_Pin\_5);

**else**{

GPIO\_ResetBits(GPIOA, GPIO\_Pin\_5);

inc = 0;

}