|  |
| --- |
| <https://github.com/Smadas/vrs_cv4.git> |
| **Úloha 2:**  **Pre klávesnicu:** Vytvorte aplikáciu, ktorá bude meniť periódu blikania LED pripojenej na PA5 podľa toho ktoré tlačidlo bolo stlačené.  void inicializaciaLED(void) |
| { | |
| //vytvorenie struktury GPIO | |
| GPIO\_InitTypeDef gpioInitStruc; | |
| gpioInitStruc.GPIO\_Mode = GPIO\_Mode\_OUT; | |
| gpioInitStruc.GPIO\_OType = GPIO\_OType\_PP; | |
| gpioInitStruc.GPIO\_Pin = GPIO\_Pin\_5; | |
| gpioInitStruc.GPIO\_Speed = GPIO\_Speed\_400KHz; | |
| //zapisanie inicializacnej struktury | |
| GPIO\_Init(GPIOA, &gpioInitStruc); | |
| }  **Funkcia inicializaciaLED() slúži na vytvorenie štruktúry pre inicializáciu pinu na ktorom je pripojená LED-ka.** | |
|  | |
| void inicializaciaADCpin(void) | |
| { | |
| GPIO\_InitTypeDef GPIO\_InitStructure; | |
| /\* Configure ADCx Channel 4 as analog input \*/ | |
| GPIO\_InitStructure.GPIO\_Pin = GPIO\_Pin\_4 ; | |
| GPIO\_InitStructure.GPIO\_Mode = GPIO\_Mode\_AN; | |
| GPIO\_InitStructure.GPIO\_PuPd = GPIO\_PuPd\_NOPULL ; | |
| GPIO\_Init(GPIOA, &GPIO\_InitStructure); | |
| }  **Funkcia inicializaciaADCpin slúži na vytvorenie štruktúry pre inicializáciu pinu na ktorom je pripojený výstup z klávesnice.** | |
|  | |
| void inicializaciaADC(void) | |
| { | |
| ADC\_InitTypeDef ADC\_InitStructure; | |
| /\* Enable the HSI oscillator \*/ | |
| RCC\_HSICmd(ENABLE); | |
| /\* Check that HSI oscillator is ready \*/ | |
| while(RCC\_GetFlagStatus(RCC\_FLAG\_HSIRDY) == RESET); | |
| /\* Enable ADC clock \*/ | |
| RCC\_APB2PeriphClockCmd(RCC\_APB2Periph\_ADC1, ENABLE); | |
| /\* Initialize ADC structure \*/ | |
| ADC\_StructInit(&ADC\_InitStructure); | |
| /\* ADC1 configuration \*/ | |
| ADC\_InitStructure.ADC\_Resolution = ADC\_Resolution\_12b; | |
| ADC\_InitStructure.ADC\_ContinuousConvMode = DISABLE; | |
| ADC\_InitStructure.ADC\_ExternalTrigConvEdge = ADC\_ExternalTrigConvEdge\_None; | |
| ADC\_InitStructure.ADC\_DataAlign = ADC\_DataAlign\_Right; | |
| ADC\_InitStructure.ADC\_NbrOfConversion = 1; | |
| ADC\_Init(ADC1, &ADC\_InitStructure); | |
| /\* ADCx regular channel4 configuration \*/ | |
| ADC\_RegularChannelConfig(ADC1, ADC\_Channel\_4, 1, ADC\_SampleTime\_16Cycles); | |
| /\* Enable the ADC \*/ | |
| ADC\_Cmd(ADC1, ENABLE); | |
| /\* Wait until the ADC1 is ready \*/ | |
| while(ADC\_GetFlagStatus(ADC1, ADC\_FLAG\_ADONS) == RESET) | |
| { | |
| } | |
| }  **Funkcia inicializaciaADC() slúži na spustenie a skontrolovanie oscilátora či je spustený, spustenie hodín, nakonfigurovanie ADC, nakonfigurovanie kanálu 4, spustenie a kontrolu ADC.** | |
|  | |
| int blikanieLED(int blikac, int blikacRychlost){ | |
| blikac++; | |
| if (blikac > blikacRychlost) | |
| { | |
| blikac = 0; | |
| GPIO\_ToggleBits(GPIOA, GPIO\_Pin\_5); | |
| } | |
| return blikac; | |
| }  **Funkcia blikanieLED spôsobuje blikanie LED-ky pomocou funkcie GPIO\_ToggleBits. Rýchlosť blikania je nastavená podľa premennej blikacRychlost, ktorá vstupuje do funkcie. Výstupom je premenná blikac, ktorá počíta cykly a keď dosiahne hodnotu blikacRychlost, zmení stav LED.** | |
|  | |
| uint16\_t citanieHodnotyADC(void) | |
| { | |
| ADC\_SoftwareStartConv(ADC1); | |
| while(!ADC\_GetFlagStatus(ADC1, ADC\_FLAG\_EOC)){} | |
| return ADC\_GetConversionValue(ADC1); | |
| }  **Funkcia citanieHodnotyADC slúži na čítanie hodnoty z prevodníka ADC pomocou premeny napätia na digitálnu hodnotu (funkcia ADC\_GetConversionValue()), pričom maximálna digitálna hodnota = 4096, t.j. 3,3V.** | |
|  | |
| int rychlostBlikaniaLED(int blikacRychlost,uint16\_t value) | |
| { | |
| if(value >= 3550 && value <= 3800) | |
| { | |
| return 200000; | |
| } | |
| else if(value > 3200 && value < 3550) | |
| { | |
| return 50000; | |
|  | |
| } | |
| else if(value >= 2600 && value <= 3200) | |
| { | |
| return 20000; | |
|  | |
| } | |
| else if(value >= 0 && value < 2600) | |
| { | |
| return 5000; | |
| } | |
| else | |
| { | |
| return blikacRychlost; | |
| } | |
| }  **Funkcia rychlostBlikaniaLED() slúži na identifikáciu stlačeného tlačidla, pričom vďaka nameraným hodnotám vieme, že prvé tlačidlo nadobúda digitálnu hodnotu približne 3650, druhé 3450, tretie 2900 a štvrté 2000. Na základe týchto hodnôt sme si zvolili rýchlosť blikania.** | |
|  | |
|  | |
| int main(void) | |
| { | |
| /\* Enable GPIO clock \*/ | |
| RCC\_AHBPeriphClockCmd(RCC\_AHBPeriph\_GPIOA, ENABLE); | |
|  | |
| int value = 0; | |
| int blikac = 0; | |
| int blikacRychlost = 200000; | |
|  | |
| //inicializacia periferii | |
| inicializaciaADCpin(); | |
| inicializaciaADC(); | |
| inicializaciaLED(); | |
|  | |
| while (1) | |
| { | |
|  | |
| blikac = blikanieLED(blikac, blikacRychlost); | |
| value = citanieHodnotyADC(); | |
| blikacRychlost = rychlostBlikaniaLED(blikacRychlost,value); | |
|  | |
| } | |
| return 0; | |
| } | |
|  |  | |

**Záver:**   
Vytvorili sme program, ktorý prepínaním tlačidiel menil rýchlosť blikania LED-ky. Tlačidlá boli identifikované digitálnou hodnotou, ktorá je úmerná nameranému napätiu z výstupu klávesnice. V zadaní sme sa nestretli s komplikáciami.