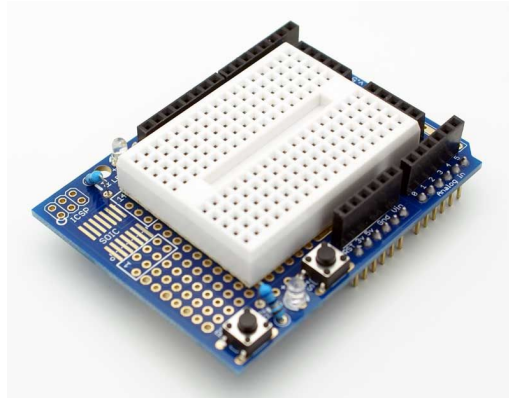


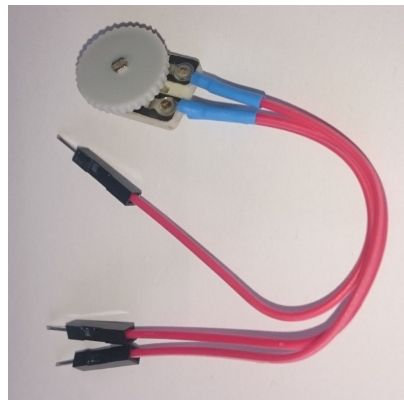
Téma : AD prevodník

1. Příprava na práci na cvičení (max 40 minút)

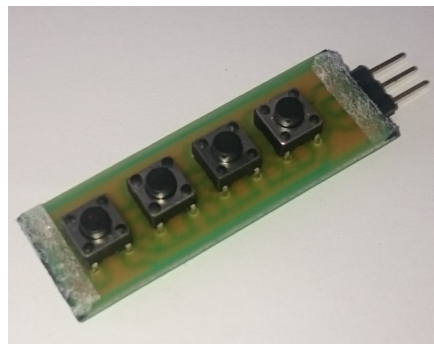
- Vytvoriť si embedded C projekt s názvom **vrs_cv4**, pričom ako HW bude zvolená nie vývojová doska ale MCU STM32L152RE a interfejs na debugovanie bude STLink
- Pre zjednodušenie pripájania potenciometrov a odporových klávesníc bude potrebné pripojiť k NUCLEO doske shield s breakout doskou.



Breakout board

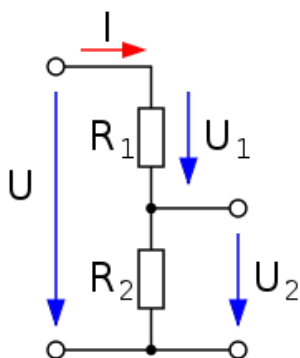


Potenciometer



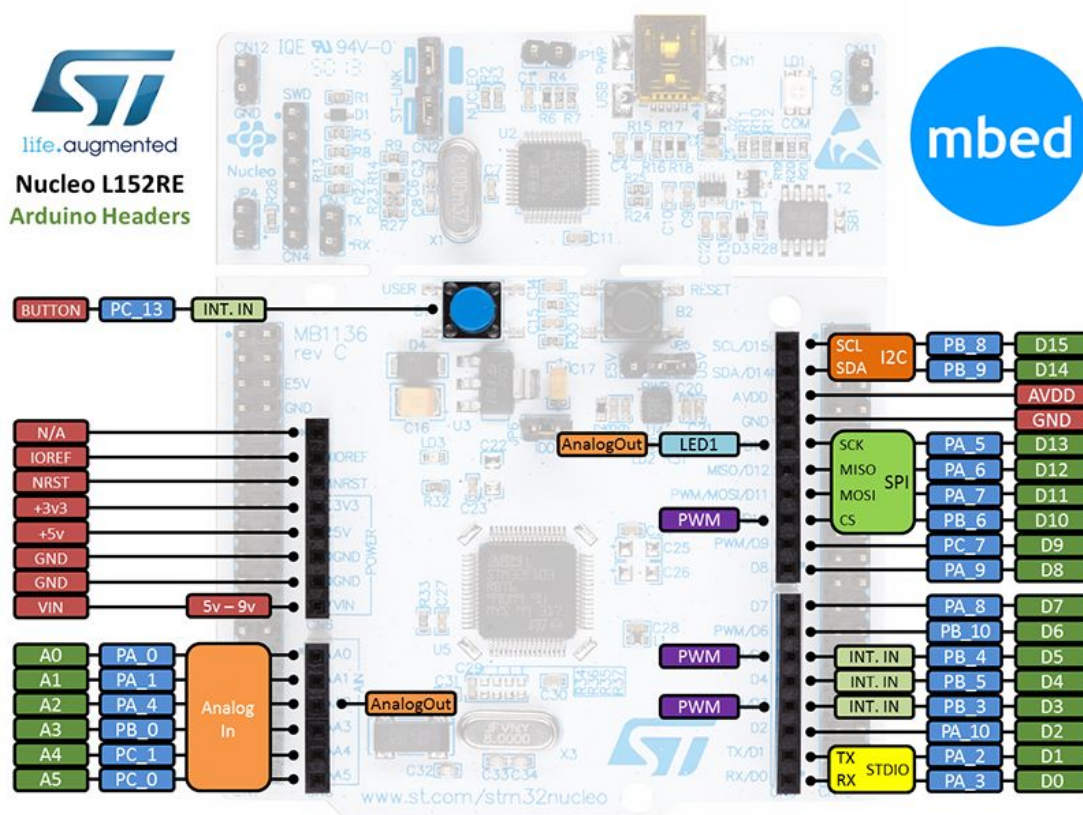
Odporová klávesnica

- Potenciometer a klávesnicu budete využívať ako odporový delič. Klávesnica sa správa ako odpor (medzi krajnými vodičmi) ktorý sa mení v závislosti od stlačenia tlačidiel alebo skupiny tlačidiel. Klávesnicu bude treba zapojiť s ďalším rezistorom ako odporový delič, pričom využijete rezistor 8.2kOhm. Odporové deliče napájajte 3.3V!



Odporový delič, viď. Tiež prednáška č.1

- Ako analógový vstup môžete využívať vstupy na Arduino shielde A0 - A5



Popis vývodov NUCLEO L152RE

- Vzorový kód pre nastavenie ADC

```

void adc_init(void)
{
    GPIO_InitTypeDef      GPIO_InitStructure;
    ADC_InitTypeDef       ADC_InitStructure;

    /* Enable GPIO clock */
    RCC_AHBPeriphClockCmd(RCC_AHBPeriph_GPIOA, ENABLE); //Opravít a upravít

    /* Configure ADCx Channel 2 as analog input */
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_2 ;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AN;
    GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_NOPULL ;
    GPIO_Init(GPIOA, &GPIO_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 channel8 configuration */
    ADC_RegularChannelConfig(ADC1, ADC_Channel_2, 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)
    {
    }
    /* Start ADC Software Conversion */
    ADC_SoftwareStartConv(ADC1);
}

```

Spustenie prevodu:

```

/* Start ADC Software Conversion */
ADC_SoftwareStartConv(ADC1);
while(!ADC_GetFlagStatus(ADC1, ADC_FLAG_EOC)){ }
AD_value=ADC_GetConversionValue(ADC1);

```

2. Úloha (max 60 minút)

Pre potenciometer:

Vytvorte aplikáciu, ktorá podľa zmeny odporu potenciometra bude meniť periódu blikania LED pripojenej na PA5 lineárne alebo skokovo. Zmena musí byť zjavná.

Pre klávesnicu:

Vytvorte aplikáciu, ktorá bude meniť periódu blikania LED pripojenej na PA5 podľa toho ktoré tlačidlo bolo stlačené.

Nastavenia jednotlivých periférii dajte do separátnych funkcií so zrozumiteľným a logickým názvom. Taktiež sa snažte funkčnosť aplikácie rozumne rozdeliť do funkcií tak, aby bol Váš zdrojový kód zrozumiteľný a čitateľný.

Aplikáciu dajte na Váš Github a pravidelne komitujte zmeny.

Odovzdajte vypracovanú správu spolu s linkom na projekt na Github obvyklým spôsobom na zdieľaný Google Drive priečinok do 16.10.2016 23.00. Vývojové dosky bude možné v ojedinelých prípadoch zapožičať, v stredu (12.10.2016) výlučne v čase 14:00 až 14:30 a odovzdanie v piatok (14.10.2016) výlučne v čase 10:00 až 10:30.