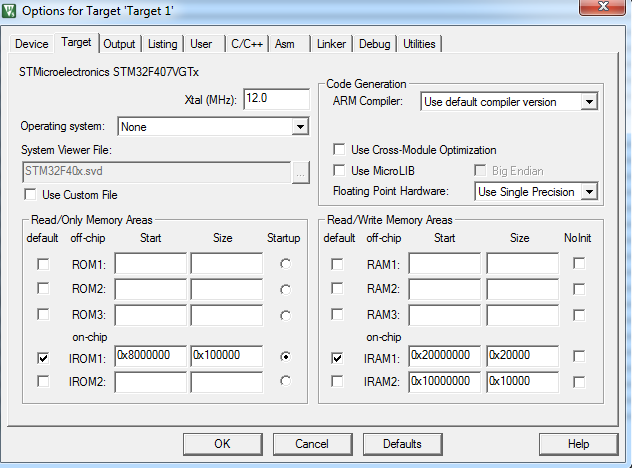
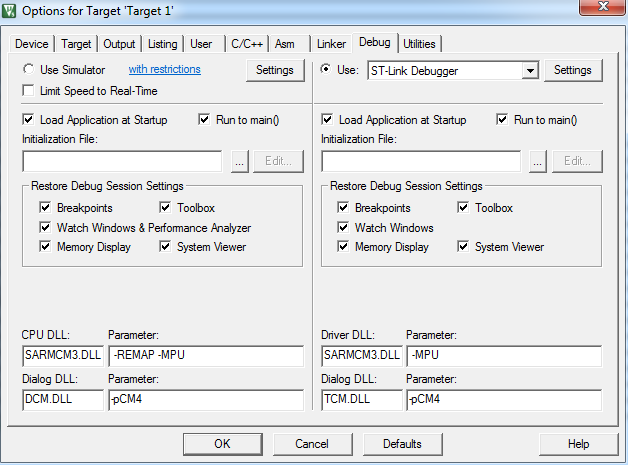
**Teoretický rozbor**

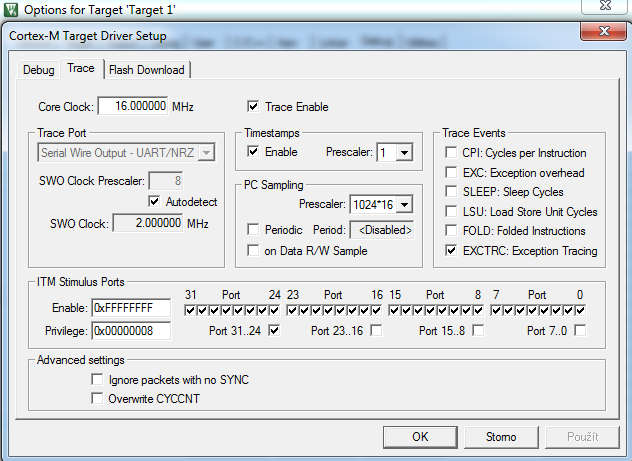
**ARM STM32F4 –** Je programovatelný 32-bitovy mikroprocesor, který jsme použili pro vypracovaní této úlohy. Programujeme v jazyce C, ve vývojovém prostředí Keil5. K mikroprocesoru se dá připojit řadu komponentu, například klávesnice, LCD displej atd.

Nastavení procesoru:



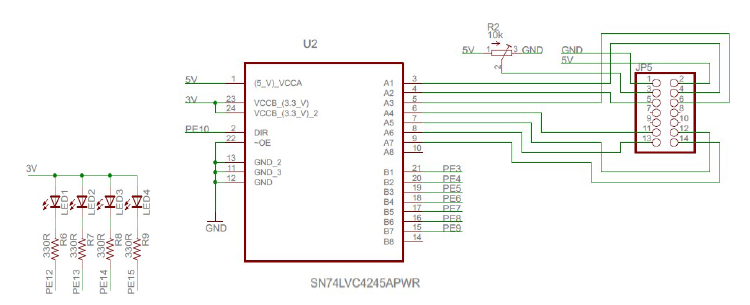
Nastavení Debugeru:





**LCD displej –** K dispozici jsme měli LCD displej s možnosti zobrazení 8x2 znaku v jeden okamžik. V programu konfigurujeme LCD displej pomoci funkce LCD\_config(); a inicializujme funkci LCD\_ini();. Pro správnou funkčnost displeje v programu, je potřeba prvně nakonfigurovat časovaní jádra procesoru a poté se až konfiguruje a inicializuje displej. Jinak časovaní procesoru nestíhá a znaky se nestačí na LCD displeji zobrazovat.

Napojení LCD displeje na ARM:



Konfigurace LCD displeje:

void LCD\_config(void){

RCC->AHB1ENR |= ((1UL << 4) ); // PE povolit clock

GPIOE->MODER |= ( (1UL << 2\*3) | // RS(1UL << 2\*4) | // R/W (1UL << 2\*5) | // E (1UL << 2\*6) | // vystup DB4..DB7 data (1UL << 2\*7) | (1UL << 2\*8) | (1UL << 2\*9) | (1UL << 2\*10)) ;

GPIOE->OTYPER &= ~( (1UL << 3) |(1UL << 4) |(1UL << 5) | (1UL << 6) |

(1UL << 7) | (1UL << 8) | (1UL << 9) |(1UL << 10)); // Push - Pull

GPIOE->OSPEEDR &= ~( (3UL << 2\*3) | (3UL << 2\*4) | (3UL << 2\*5 )|(3UL << 2\*6) |(3UL << 2\*7) | (3UL << 2\*8) | (3UL << 2\*9) |(3UL << 2\*10) ) ;

GPIOE->OSPEEDR |= ( (2UL << 2\*3) |(2UL << 2\*4) | (2UL << 2\*5) | // 50 Mhz out

(2UL << 2\*6) | (2UL << 2\*7) | (2UL << 2\*8) | (2UL << 2\*9) |(2UL << 2\*10) );

GPIOE->PUPDR &= ~((3UL << 2\*3) |(3UL << 2\*4) |(3UL << 2\*5) |(3UL << 2\*6) |

(3UL << 2\*7) | (3UL << 2\*8) | (3UL << 2\*9) |(3UL << 2\*10) ) ;}

Inicializace LCD displeje:

void LCD\_ini (void){

GPIOE->BSRR = (1UL << (10+16) ); //DIR=0

Delay(800); // 80 ms

write\_nibble\_res(0x3); // 3

Delay(50); // 5 ms

write\_nibble\_res(0x3);

Delay(10); // 1 ms

write\_nibble\_res(0x3); //

Delay(10); // 1 ms

write\_nibble\_res(0x2);

LCD\_ctrlWR(0x28);

LCD\_ctrlWR(0x28);

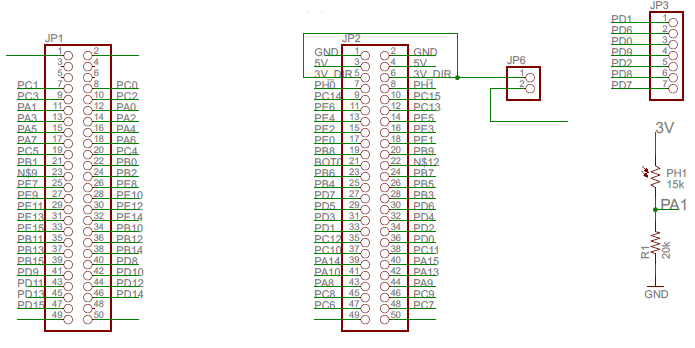
LCD\_ctrlWR(0x0C);

LCD\_ctrlWR(0x06);

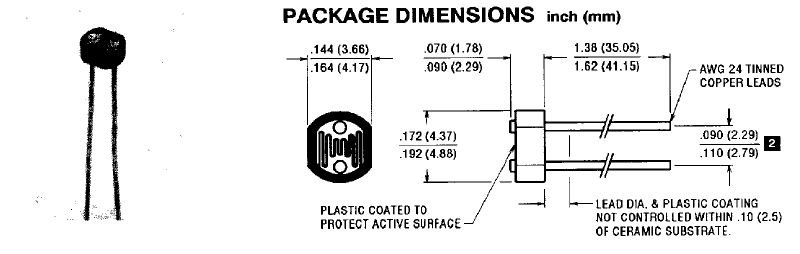
LCD\_ctrlWR(0x01);

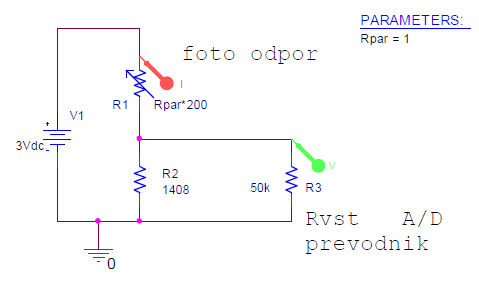
Delay(20); // 2 ms }

**Klávesnice –** Propojení klávesnice s mikroprocesorem ARM:

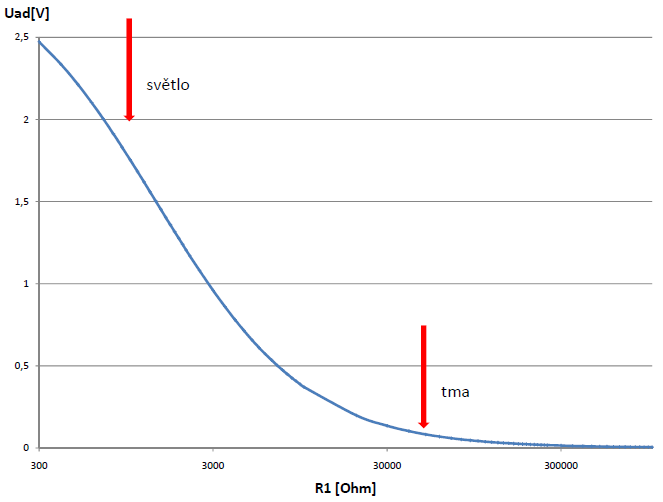


**Vstupní obvod –** Jako vstupní obvod používáme obvod pro měření světla VT93N2 s foto odporem.





Obvod funguje na základě toho, že s nižším světlem dopadající na obvod, se zvyšuje jeho vnitřní odpor a tím klesá jeho napětí.



**Funkce**

**Vývojový diagram –**

+

+

+

Výpis y na displej v Luxech

Výpis y na displej v Luxech

Výpis y na displej v Cd

prum = prum/129;

y = prum\*3000/212;

y = 28,428 \* expm1(0,0019\*y);

y = y\*1,11;

prum = prum/129;

y = prum\*3000/212;

y = 28,428 \* expm1(0,0019\*y);

y = y\*0,95;

prum = prum/129;

y = prum\*3000/212;

y = 28,428 \* expm1(0,0019\*y);

y = y\*1,11;

2

1

-

-

i == 129

-

i == 129

Vzorek ze vstupního obvodu;

i = 0;

prum = 0 ,y=0; prum = prum + vzorek; i++;

Vzorek ze vstupního obvodu;

i = 0;

prum = 0 ,y=0; prum = prum + vzorek; i++;

Vzorek ze vstupního obvodu;

i = 0;

prum = 0 ,y=0; prum = prum + vzorek; i++;

1: Lampa

2: Den. sv

1: Luxy

2: Cd

Konfigurace

1: Lampa

2: Den. sv

1

2

i == 129

1

2

+

Vzorek ze vstupního obvodu;

i = 0;

prum = 0 ,y=0; prum = prum + vzorek; i++;

i == 129

prum = prum/129;

y = prum\*3000/212;

y = 28,428 \* expm1(0,0019\*y);

y = y\*0,95;

Výpis y na displej v Cd

-

**Kalibrace luxmetru –** Program odstartujeme odebráním 129 vzorku ze vstupního obvodu, a vypočítáme z nich průměr. Vypočítaný průměr vložíme do vzorečku: y=průměr\*U(mV)/2(počet bitu). Spustíme program a budeme měřit y a luxy na luxmetru, výsledky zapíšeme do Excelu a vytvoříme exponenciální graf.

Pokud máme spolehlivost (R) vyšší jak 0,97, můžeme zkopírovat vzorec y do našeho programu a dopočítat převod mV na Luxy.

**Výběrové menu –** Pomoci funkci puts\_LCD(); a getkey(); jsme vytvořili jednoduché menu kde budeme mít na výběr zda chceme měření v Luxech nebo v Candelech.

**Ledky –** Pokud naměřené hodnoty neodpovídají podmínkám pro osvětlení v daném prostředí (viz. Zadaní), ledka pro dane prostředí zhasne.  
Krytá nástupiště a chodby pro cestující – Oranžova

Psaní, čtení, práce na PC – Červena

Tabule ve školách – Modrá

**Závěr**

Na procesoru ARM zezačátku působí zmateně a složitě, ovšem po naučení a odzkoušení několika programu se s ním začíná dobře pracovat. Program funguje dobře, a při tvorbě nedošlo k žádným větším komplikacím. Při porovnání naměřených hodnot digitálním lux metrem a našim vyrobeným, odchylka se může rovnat až 100lx. Jediné zlepšení by tedy mohlo být větší přesnost měření. Program by mohl mít využiti ke kontrole, zda je v daném prostředí dostatečné osvětlení.