

LCD DISPLEJ

Pro práci s LCD třeba využít knihovny: „`config_kit_ver3.c`“

Využít funkcí:

- LCD_config(); a LCD_ini();
- puts_lcd(1,“TEXT”);
- LCD_ctrlWR(); a LCD_dataWR();
- Delay();

Program pro výpis jména:

```
1  #include "stm32f4xx.h"                // Device header
2  void LCD_config(void);
3  void LCD_ini (void);
4  void puts_LCD (int radek, char* ukaz);
5
6  int main()
7  {
8      SystemCoreClockUpdate();           // do SystemCoreClock se nahraze frekvence jadra
9      SysTick_Config(SystemCoreClock/ 10000); // konfigurace SysTick timeru na periodu 0.1ms
10
11     LCD_config();
12     LCD_ini();
13
14     while (1){
15         puts_LCD(1," DENYYYS"); //vlozeni radku [radek+text]
16         puts_LCD(2," MYNARI");
17     }
18
19
20
21 }
22
```

- psaní do řádků skrze funkci puts_LCD

Program pro výpis abecedy:

```
1  #include "stm32f4xx.h"                // Device header
2  void LCD_config(void);
3  void LCD_ini (void);
4  void puts_LCD (int radek, char* ukaz);
5  void LCD_ctrlWR(uint16_t cmd);
6  void LCD_dataWR(uint16_t cmd);
7  void Delay (uint32_t ms);
8
9  int main()
10 {
11     int i;
12     SystemCoreClockUpdate();           // do SystemCoreClock se nahraze frekvence jadra
13     SysTick_Config(SystemCoreClock/ 10000); // konfigurace SysTick timeru na periodu 0.1ms
14
15     LCD_config();
16     LCD_ini();
17
18     while (1){
19         for(i='A';i<='Z';i++){
20             LCD_ctrlWR(0xc3);
21             LCD_dataWR(i);
22             Delay(5000);
23         }
24     }
25 }
```

Program pro zobrazení čísel <1-1000>:

```
1  #include "stm32f4xx.h"           // Device header
2  #include "stdio.h"
3  void LCD_config(void);
4  void LCD_ini (void);
5  void puts_LCD (int radek, char* ukaz);
6  void LCD_ctrlWR(uint16_t cmd);
7  void LCD_dataWR(uint16_t cmd);
8  void Delay (uint32_t ms);
9
10 int main()
11 {
12     int i;
13     char x[8]; // z duvodu omezeni na 8 znaku
14     SystemCoreClockUpdate(); // do SystemCoreClock se nahraje frekvence jadra
15     SysTick_Config(SystemCoreClock/ 10000); // konfigurace SysTick timeru na periodu 0.1ms
16
17     LCD_config();
18     LCD_ini();
19
20     while (1){
21         for(i=1;i<=1000;i++){
22             sprintf(x,"%d",i); // prevedeni cisla (i) na char (x)
23             puts_LCD(1,"vypis ");
24             puts_LCD(2,x);
25             Delay(2000);
26         }
27     }
28 }
```

- Použití cyklu **for**, **sprintf** načte číslo (i) a převede na **char** (x)

Program pro posun textu:

```
1  #include "stm32f4xx.h"           // Device header
2  #include "stdio.h"
3  void LCD_config(void);
4  void LCD_ini (void);
5  void puts_LCD (int radek, char* ukaz);
6  void LCD_ctrlWR(uint16_t cmd);
7  void LCD_dataWR(uint16_t cmd);
8  void Delay (uint32_t ms);
9
10 int main()
11 {
12     int i;
13     char retezec[] = "Dollar Prync, Smichov, Praha";
14     SystemCoreClockUpdate(); // do SystemCoreClock se nahraje frekvence jadra
15     SysTick_Config(SystemCoreClock/ 10000); // konfigurace SysTick timeru na periodu 0.1ms
16
17     LCD_config();
18     LCD_ini();
19
20     while (1){
21         LCD_ctrlWR(0x18); // posun doleva (posun doleva = 0x1c)
22         puts_LCD(1,retezec); // treba v configu zmenit hodnotu n, jinak vezme pouze 9 znaku
23         Delay(3500);
24     }
25 }
```

- **LCD_ctrWR** řídí směr posunu

Instrukce LCD (ctrWR)

Instrukce	Řídící bity a kód instrukce RS RW 7 6 5 4 3 2 1 0	Popis	Doba provedení instrukce při fint = 250kHz
Vymaž displej	0 0 0 0 0 0 0 0 1	vymaže celý displej a nastaví adresu DD RAM na 0	1,64ms
Návrat	0 0 0 0 0 0 0 1 *	nastaví adresu DD RAM na 0, obsah beze změny, posune displej do původní pozice	1,64ms
Nastavení módu vstupu dat	0 0 0 0 0 0 1 ID S	nastavení pohybu kurzoru bit ID nastavení posuvu displeje bit S	40us
Nastavení módu displeje	0 0 0 0 0 1 D C B	zapnutí / vypnutí displeje bit D zapnutí / vypnutí kurzoru bit C blikání znaku na kurzoru bit B	40us
Posuv kurzoru nebo displeje	0 0 0 0 1 SC RL * *	posun kurzoru či displeje bit SC směr posuvu bit RL	40us
Funkce displeje	0 0 0 1 DL N F * *	nastavení 4/8 sběrnice bit DL nastavení počtu řádek bit N nastavení fontu znaků bit F	40us
Nastavení adresy CG RAM	0 0 0 1 A A A A A	nastavení adresy CG RAM (6 bitů)	40us
Nastavení adresy DD RAM	0 0 1 A A A A A A	nastavení adresy DD RAM (7 bitů)	40us
Čtení adresy a bitu BF	0 1 B F A A A A A A	BF = 1 indikuje probíhající instrukci (BF a T bitů adresy)	40us
Zápis dat do CG nebo do DD RAM	1 0 D D D D D D D D	zápis dat ze sběrnice do CG RAM nebo DD RAM	40us
Čtení dat z CG nebo z DD RAM	1 1 D D D D D D D D	čtení dat z paměti CG RAM nebo DD RAM	40us

Vysvětlivky:

CG RAM je paměť znakového generátoru (tedy paměť fontů)

DD RAM jen paměť displeje z ní jsou dle příslušné mapy zobrazeny příslušné adresy

ID=1 po zápisu na určitou adresu v DD RAM (nebo CG RAM) je ukazatel adresy inkrementován

ID=0 po zápisu na určitou adresu v DD RAM (nebo CG RAM) je ukazatel adresy dekrementován

S=1 nastaví posun celého displeje po zapsání (směr posuvu je určen bitem ID)

S=0 nastaví posun kurzoru po zapsání (směr posuvu je určen bitem ID)

SC=1 posunutí displeje (bez zápisu)

SC=0 posunutí kurzoru (bez zápisu)

RL=1 nastavení směru posuvu doprava

RL=0 nastavení směru posuvu doleva

DL=1 nastavení datové sběrnice na 8-bitů (tj. DB7,.....,DB0)

DL=0 nastavení datové sběrnice na 4-bity (tj. DB7,.....,DB4)

F=1 nastaví výběr znaků z fontu 5x10 bodů

F=0 nastaví výběr znaků z fontu 5x7 bodů, u většiny běžných displejů

N=1 nastavení počtu řádek na 2 nebo 4

N=0 nastavení počtu řádek na 1