

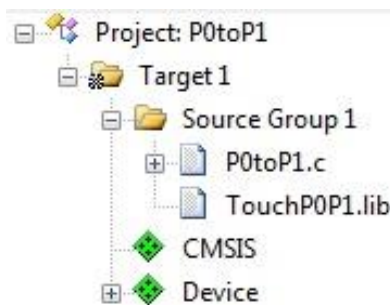
# Programmierkonzepte

mit uVision5

## a) 8Bit Programmierung

mit P0- und P1-Ports  
über Touchscreen

- TouchPOP1.h und .lib ins Projektverzeichnis legen
- TouchPOP1.lib ins Projekt aufnehmen
- TouchPOP1.h im Code einbinden
- Der Systemtakt beträgt 72MHz



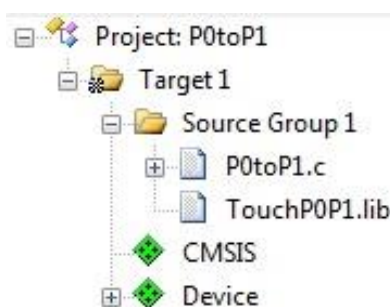
```
#include "TouchPOP1.h"

int main(void)
{
    InitTouchPOP1("1");
    while(1)
        P1 = P0;
}
```



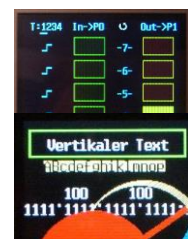
## b) Erweiterte 8Bit Programmierung

P0-, P1-Ports über Touchscreen sowie Grafik und integrierte Peripherie



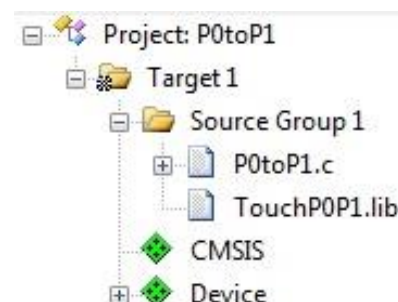
```
#include "TouchPOP1.h"

int main(void)
{
    InitTouchPOP1("1");
    ADCInit(1, "PC4");
    while(1)
        P1 = ADCGetVal(1);
}
```



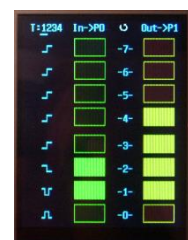
## c) 8Bit Programmierung in 32Bit Registern

mit P0-, P1-Ports und Touchscreen



```
#include "TouchPOP1.h"

int main(void)
{
    InitTouchPOP1("1");
    RCC->APB2ENR |= 1<<9;
    ADC1->SQR3 = 14;
    while(1)
        P1 = ADC1->DR>>4;
}
```



### d) Reine 32Bit Registerprogrammierung mit 8MHz

- SystemInit() leer definieren
- stm32f10x.h, Startup und NUR CMSIS einbinden

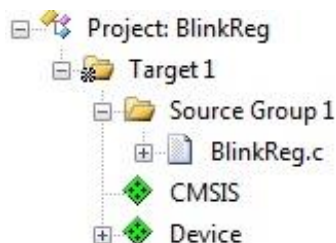


```
#include <stm32f10x.h>
void SystemInit(void) {}

int main(void)
{
    RCC->APB2ENR |= 1<<6;
    while(1)
        GPIOE->ODR = 0xAA00;
}
```

### e) 32Bit Registerprogrammierung mit SystemInit und 27MHz

- Für SystemInit() im Device startup



```
#include <stm32f10x.h>

int main(void)
{
    RCC->APB2ENR |= 1<<6;
    while(1)
        GPIOE->ODR = 0xAA00;
}
```

### f) CMSIS - Programmierung

- Der Systemtakt beträgt 72MHz

- use\_STD\_Periph\_Driver beachten
- stm.....conf und alle benötigten Periph...h und c einbinden, alle .c ins Projekt aufnehmen



```
#include "stm32f10x_gpio.h"

int main(void)
{
    GPIO_InitTypeDef GPIO_InitStructure;
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_12;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
    GPIO_SetBits(GPIOE, GPIO_Pin_14);
}
```

Professionelle hersteller- und controllerübergreifende CMSIS-Programmierung

Die benötigten StdPeriph\_Driver Header- und Sourcedateien sind aus

Keil\_v5/ARM/Pack/Keil/ STM32F1xx\_ LIB/1.0.5/Device/StdPeriph\_Driver/inc und /src ins Projektverzeichnis zu kopieren und die Sourcedateien ins Projekt einzubinden.

### alle) Hinweis zu Systemtakt und Verzögerungsschlaufen:

SystemInit() schaltet den SystemClock von 8 MHz auf 72 MHz und baut für den Flash-speicherzugriff WaitStates ein. Dadurch laufen Verzögerungsschlaufen nur 6x schneller. Der Compiler erlaubt sich, gleiche Verzögerungsschlaufen an verschiedenen Programmstellen verschieden zu übersetzen. Wichtige Zeiten also immer überprüfen!