

General-Timer:

	TIM2			TIM3			TIM4			TIM5			TIM9			TIM10			TIM11		TIM12		TIM13			TIM14	
	APB1												APB2														
CH1	A0	A5	A15	A6	B4	C6	B6	D12	H10	A0	A2	E5	B8	F6	B9	F7	B14	H6	A6	F8	A7	F9					
CH2	A1	B3		A7	B5	C7	B7	D13	H11	A1	A3	E6					B15	H9									
CH3	A2	B10		B0	C8		B8	D14	H12	A2																	
CH4	A3	B11		B1	C9		B9	D15	I0	A3																	
ETR	A0	A5	A15	D2			E0																				

Codebeispiel:

```
#include <stdint.h>
#include <stdlib.h>
#include "hardware.h"
#include "mySTM32.h"

void initApplication()
{
    // u.a. nötig für waitMs(..) und waitUs(..)
    SysTick_Config(SystemCoreClock/100);
    // weitere Initialisierungen durchführen

    /* GPIOG Periph clock enable */
    RCC_AHB1PeriphClockCmd(
    RCC_AHB1Periph_GPIOD, ENABLE);
    /* Configure PG6 and PG8 in output pushpull mode */
    GPIO_InitTypeDef GPIO_InitStructure;
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_15;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;
    GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_100MHz;
    GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_NOPULL;
    GPIO_Init(GPIOD, &GPIO_InitStructure);
}

int main(void)
{
    SystemInit();
    initApplication();
    do{
        //Code(EVA)
        GPIO_ToggleBits(GPIOD,GPIO_Pin_15);
        waitMs(300);
    } while (true);
    return 0;
}

extern "C" void SysTick_Handler(void)
{ /* Application SysTick */ }
```

Interruptauswahl für den STM32F4: extern "C" void vektor () { ... }:

```
SysTick_Handler,
EXTIn_IRQHandler,
ADC_IRQHandler,
TIMn_IRQHandler,
TIMn_DAC_IRQHandler,
USARTn_IRQHandler,
SPIn_IRQHandler,
RTC_Alarm_IRQHandler,
SDIO_IRQHandler,
DMAn_StreamX_IRQHandler,
ETH_IRQHandler,
FPU_IRQHandler,
CRYP_IRQHandler,
DCML_IRQHandler,
HASH_RNG_IRQHandler,
WWWDG_IRQHandler,
...
```

Advanced-Timer:

	TIM1			TIM8		
	APB2					
CH1	E9	A8	I5	C6		
CH2	E11	A9	I6	C7		
CH3	E13	A10	I7	C8		
CH4	E14	A11	I2	C9		
ETR	E7	A12	I3	A0		

DAC:

	DAC
	APB1
CH1	A4
CH2	A5

Basic Timer:

TIM6	TIM7
APB1	

SDIO:

D0	D1	D2	D3	D4	D5	D6	D7	CK	CMD
APB2									
C8	C9	C10	C11	B8	B9	C6	C7	C12	D2

I²C:

	SCL		SDA		SMBA	
	APB1					
I2C1	B6	B8	B7	B9	B5	
I2C2	B10	F1	H4	B11	F0	H5
I2C3	A8	H7	C9	H8	A9	H9

CAN:

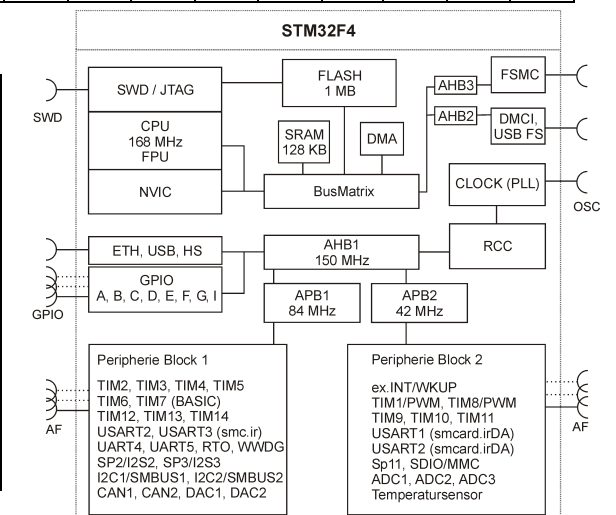
	CAN1			CAN2		
	APB1					
RX	A11	B8	D0	I9	B5	B12
TX	A12	B9	D1	H13	B6	B13

USART:

	USART1		USART2		USART3		USART4		USART5		USART6	
	APB1											
RX	A10	B7	A3	D6	B11	D9	C11	A1	C11	D2	C7	G9
TX	A9	B6	A2	D5	B10	D8	C10	A0	C10	C12	C6	G14

ADC:

ADC1	ADC2	ADC3	Channel
APB2			
A0			0
A1			1
A2			2
A3			3
A4	F6		4
A5	F7		5
A6	F8		6
A7	F9		7
B0	F10		8
B1	F3		9
C0			10
C1			11
C2			12
C3			13
C4	F4		14
C5	F5		15
Intern Temp.-Sensor			16
Intern VREF			17
Intern VBAT			18



Klassenübersicht mySTM32 C++ Framework 1.x (Auszug)

