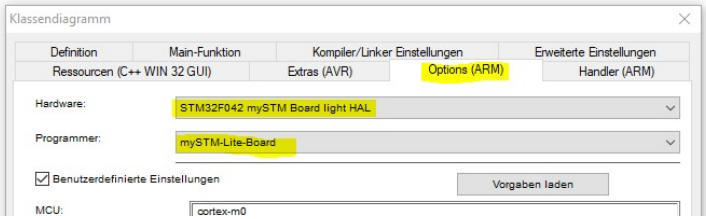
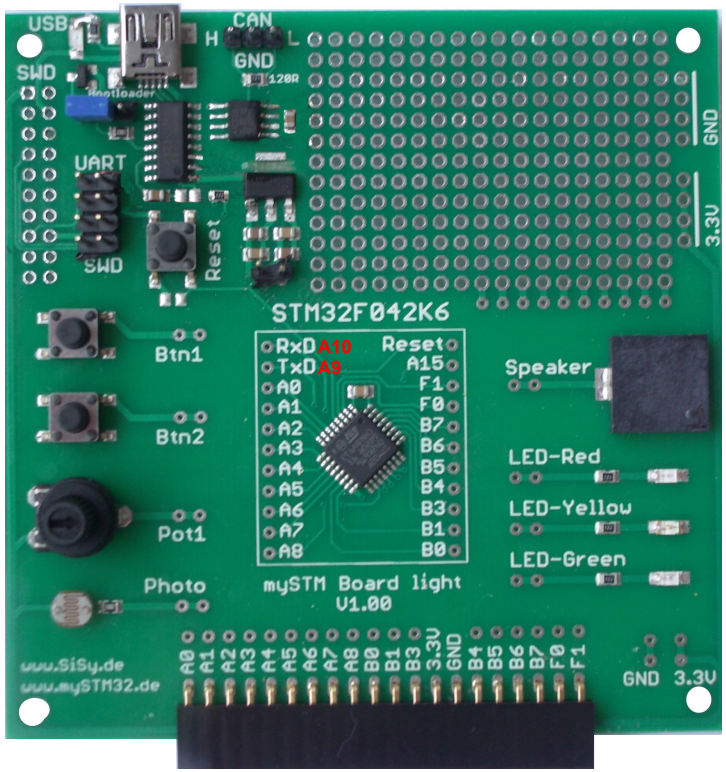




Pinbelegung der Erweiterungsbuchse
Pin assignments of the add-on-socket

1 = Port A.0
2 = Port A.1
3 = Port A.2
4 = Port A.3
5 = Port A.4
6 = Port A.5
7 = Port A.6
8 = Port A.7
9 = Port A.8
10 = Port B.0
11 = Port B.1
12 = Port B.3
13 = 3,3 V
14 = GND
15 = Port B.4
16 = Port B.5
17 = Port B.6
18 = Port B.7
19 = Port F.0
20 = Port F.1

Timer:

	TIM1	TIM2	TIM3	TIM14	TIM16	TIM17
	APB2	APB1 (max 48MHz)			APB2 (max 48MHz)	
CH1	A8 (AF2)	A0 (AF2)	A6 (AF1)	A4 (AF4)	A6 (AF5)	A7 (AF5)
CH2	A9 (AF2)	A1 (AF2)	A7 (AF1)			
CH3	A10 (AF2)	A2 (AF2)	B0 (AF1)			
CH4	A11 (AF2)	A3 (AF2)	B1 (AF1)			
ETR	A12 (AF2)	A5 (AF2)				

USART:

	USART1		USART2	
	APB2 (max 48MHz)		APB1 (max 48MHz)	
TX	**A9** (AF1)	B6 (AF0)	A2 (AF1)	A14 (AF1)
RX	**A10** (AF1)	B7 (AF0)	A3 (AF1)	A15 (AF1)

I²C:

	SDA		SCL	
	APB1 (8MHz, max 48MHz)			
I2C1	**F0** (AF1)	B7 (AF1)	**F1** (AF1)	B8 (AF1)

Simple C Codebeispiel/ C code example:

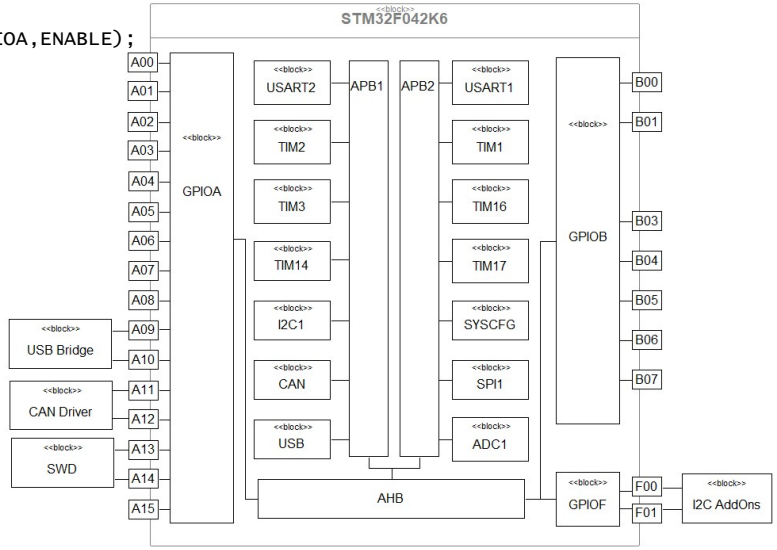
```
#include <stddef.h>
#include <stdlib.h>
#include "hardware.h"

void initApplication()
{
    // config 10ms SystemTimer
    SysTick_Config(SystemCoreClock/100);
    // user initializations here
    RCC_AHBPeriphClockCmd(RCC_AHBPeriph_GPIOA,ENABLE);
    GPIO_InitTypeDef led;
    led.GPIO_Mode  = GPIO_Mode_OUT;
    led.GPIO_OType = GPIO_OType_PP;
    led.GPIO_Pin   = GPIO_Pin_0;
    led.GPIO_PuPd  = GPIO_PuPd_NOPULL;
    led.GPIO_Speed = GPIO_Speed_2MHz;
    GPIO_Init(GPIOA,&led);
}
int main(void)
{
    SystemInit();
    initApplication();
    do{
        GPIO_ToggleBits(GPIOA,GPIO_Pin_0);
        waitMs(200);

    } while (true);
    return 0;
}
extern "C" void SysTick_Handler(void)
{
    // Application SysTick default 10ms
}
```

CAN:

	CAN	
	APB1 (max 48MHz)	
RX	**A11** (AF4)	B8 (AF4)
TX	**A12** (AF4)	

SPI:

	MOSI		MISO		SCK	
	APB2 (max 48MHz)					
SPI1	A7 (AF0)	B3 (AF0)	A6 (AF0)	B4 (AF0)	A5 (AF0)	B5 (AF0)



Interrupts STM32F042: extern "C" void vektor ( ) { … ClearITPendingBit(...); }

```
EXTI0_1_IRQn              //EXTI Line 0 and 1            extern "C" void EXTI0_1_IRQHandler()
EXTI2_3_IRQn              //EXTI Line 2 and 3            extern "C" void EXTI2_3_IRQHandler()
EXTI4_15_IRQn             //EXTI Line 4 to 15            extern "C" void EXTI4_15_IRQHandler()

DMA1_Channel1_IRQn        //DMA1 Channel 1               extern "C" void DMA1_Channel1_IRQHandler ()

ADC1_IRQn                 //ADC Interrupts               extern "C" void ADC1_IRQHandler()

TIM1_BRK_UP_TRG_COM_IRQn  //TIM1 Break, Update, Trigger  extern "C" void TIM1_BRK_UP_TRG_COM_IRQHandler()
TIM1_CC_IRQn              //TIM1 Capture Compare         extern "C" void TIM1_CC_IRQHandler()
TIM2_IRQn                 //TIM2 Interrupt               extern "C" void TIM2_IRQHandler()
TIM3_IRQn                 //TIM3 Interrupt               extern "C" void TIM3_IRQHandler()
TIM14_IRQn                //TIM14 Interrupt              extern "C" void TIM14_IRQHandler()
TIM16_IRQn                //TIM16 Interrupt              extern "C" void TIM16_IRQHandler()
TIM17_IRQn                //TIM17 Interrupt              extern "C" void TIM17_IRQHandler()

I2C1_IRQn                 //I2C1 Interrupt               extern "C" void I2C1_IRQHandler()
SPI1_IRQn                 //SPI1 Interrupt               extern "C" void SPI1_IRQHandler()

USART1_IRQn               //USART1 Interrupt             extern "C" void USART1_IRQHandler()
USART2_IRQn               //USART2 Interrupt             extern "C" void USART2_IRQHandler()

CEC_CAN_IRQn              //CEC and CAN Interrupts       extern "C" void CEC_CAN_IRQHandler()
```

ADC:

ADC	Channel
APB2 (max 14MHz)	
A0	0
A1	1
A2	2
A3	3
A4	4
A5	5
A6	6
A7	7
B0	8
B1	9
Intern Temp.-Sensor	10
Intern VREF	11
Intern VBAT	12

Kurzübersicht / short overview mySTM32 C++ PEC Portable Embedded Framework (Beispiele / Examples)