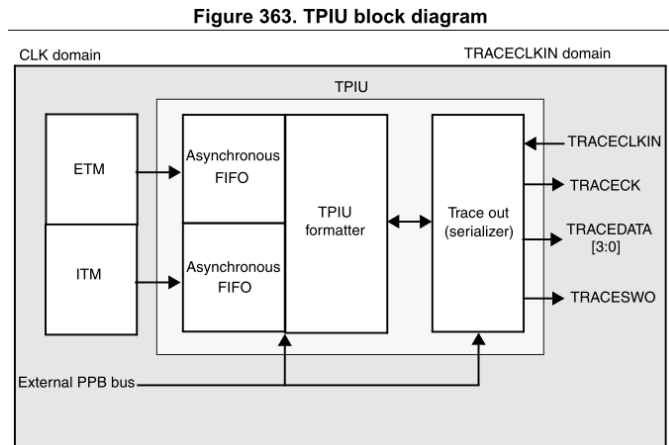


Erweiterte Debug Möglichkeiten beim Cortex-M3/M4

1 Trace -Data

Die TPIU (Trace Port Interface Unit) ist die Verbindung zwischen Host und ITM (Instrumentation Trace Model) bzw. Host und ETM (Embedded Trace Model).



Target mit einfachster Trace Option:
1-Pin, asynchron

Das ITM kann Trace Informationen als Pakete erzeugen:

- Software Trace, Software schreibt direkt in die ITM Register
- Hardware Trace, von DWT (data watchpoint trigger) erzeugte Pakete
- Time stamping, 21-bit Register innerhalb des ITM

Die DBGMCU (MCU debug component) stellt ein Register bereit mit dem die Zuordnung der Pins für Trace eingestellt und die TPIU aktiviert werden kann.

DBGMCU_CR register

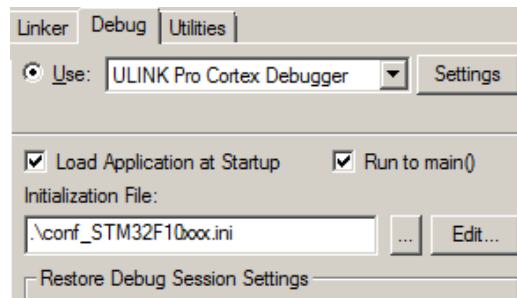
Address: 0xE004 2004

Only 32-bit access supported

POR Reset: 0x0000 0000 (not reset by system reset)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	DBG_TIM11 S_TÖP	DBG_TIM10 S_TÖP	DBG_TIM9 S_TÖP	DBG_TIM14 S_TÖP	DBG_TIM13 S_TÖP	DBG_TIM12 S_TÖP	Reserved			DGB_CAN2 S_TÖP	DBG_TIM7 S_TÖP	DBG_TIM6 S_TÖP	DBG_TIM5 S_TÖP	DBG_TIM8 S_TÖP	DBG_I2C2 SMBUS TIMEOUT
	rw	rw	rw	rw	rw	rw				rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBG_I2C1 SMBUS TIMEOUT	DBG_CAN1 S_TÖP	DBG_TIM4 S_TÖP	DBG_TIM3 S_TÖP	DBG_TIM2 S_TÖP	DBG_TIM1 S_TÖP	DBG_WWDG S_TÖP	DBG_IWDG S_TÖP	TRACE_MODE [1:0]		TRACE IOEN	Reserved		DBG_STANDBY	DBG_STOP	DBG_SLEEP
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw

Für die Aktivierung der TPIU muss TRACE_IOEN = 1 sein. Das kann mittels eines Init-Files automatisiert erfolgen, oder muss in der Debug-Session händisch im Systemviewer/Debug erfolgen.



```

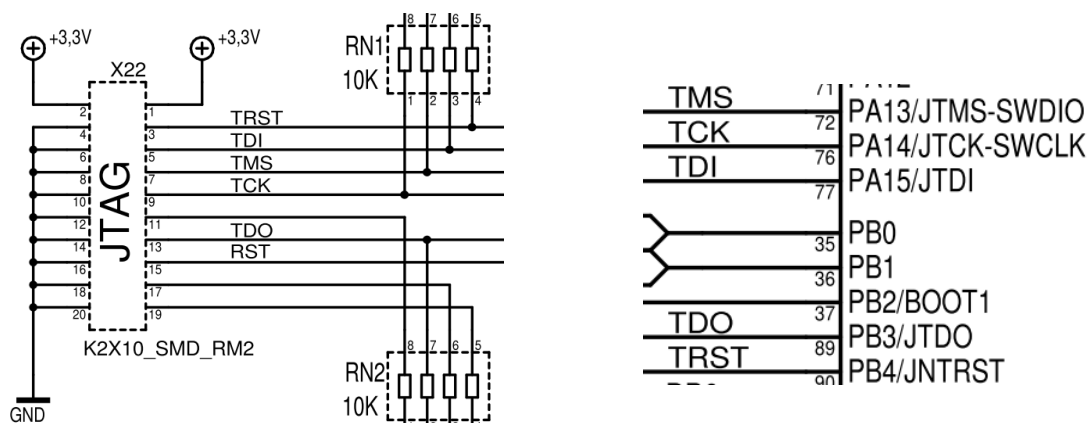
in.c  stm32f10x_spi.c  wait.c  conf_STM32F10xxx.ini
/*
 * function to en-/disable the trace port
 */
FUNC void EnableTPIU( unsigned int bEnable )
{
    unsigned long DBGMCU_CR;
    unsigned long nValue;

    DBGMCU_CR = 0xE0042004;
    nValue = _RDWORD( DBGMCU_CR );
    if( bEnable == 1 )
        nValue |= 0x00000020; /* Set asynchronous communication via DBGMCU_CR */
    else
        nValue &= ~0x00000020; /* no trace */
    _WDWORD( DBGMCU_CR, nValue );
}

```

1.1 Hardware

Bei der einfachsten Verdrahtung der JTAG-Schnittstelle wird nur TDO↔JTDO für Trace-Zwecke bereitgestellt.



Trace kann deshalb nur im asynchronen Modus erfolgen, entsprechend können einige Optionen des Traceoutputs nicht genutzt werden. Hierzu gehören u.a.:

- Execution Profiling, Show Time
- Execution Profiling, Show Calls
- Logic Analyzer
- Performance Analyzer
- Code Coverage

TRACE pin assignment

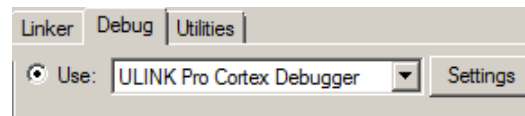
- Asynchronous mode
The asynchronous mode requires 1 extra pin and is available on all packages. It is only available if using Serial Wire mode (not in JTAG mode).

Table 230. Asynchronous TRACE pin assignment

TPUI pin name	Trace synchronous mode		STM32F10xxx pin assignment
	Type	Description	
TRACESWO	O	TRACE Async Data Output	PB3

1.2 Konfiguration der Entwicklungsumgebung

Debug Tool auswählen und *Settings* betätigen.

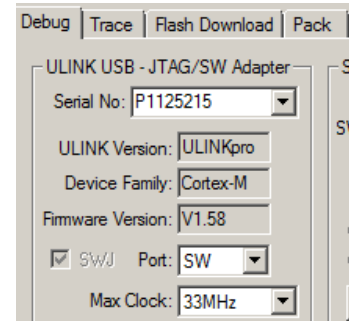


In der Registerkarte *Debug* muss

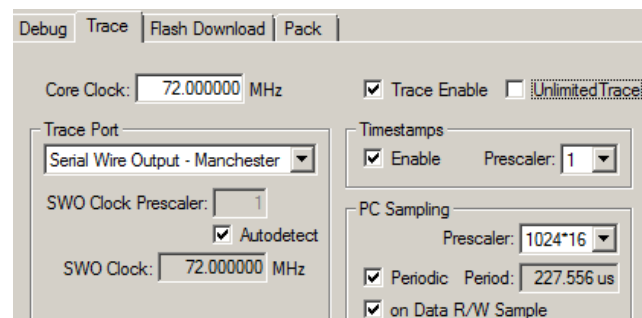
- Port *SW* (Single Wire) selektiert werden

In der Registerkarte *Trace* muss

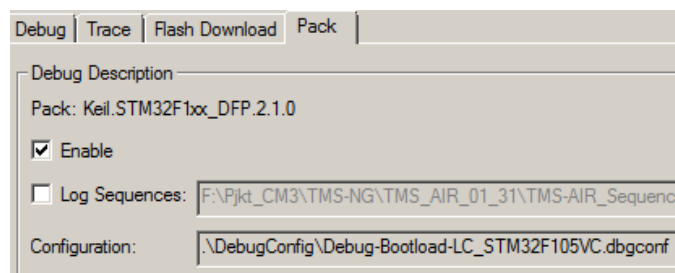
- Core-Clock eingegeben werden
- Serial Wire Output - Manchester ausgewählt werden
- Trace enable gesetzt sein



Bei Verwendung der Packs kann auf dieser Registerkarte das Konfigurationsfile aktiviert werden. Im Konfigurationsfile ist es **nicht** möglich, die TPIU zu *enablen*. Dieses **muss** zusätzlich, wie oben beschrieben erfolgen.



Die Logdateien werden entsprechend groß und sollten nicht unnötig aktiviert werden.



1.3 Debug Session

In der Debug Session kann das Window *Trace Data* angezeigt werden in dem die gewünschten Ausgaben dann erfolgen.

In *Display* kann ausgewählt werden, welche Pakete zur Anzeige kommen sollen.

In der Toolbar ist auch eine Suchfunktion, um in der Aufzeichnung einen bestimmten Datensatz zu finden.
(Beispiel: SPI2_Sendbyte)

Trace Data				
Display: ITM - PC Samples		SPI2_Sendbyte		
Time	ETM - Code Exec HLL	Action / Data	Src C...	Function
35.474 46	ITM - All	r1, r0		HCD_IsDeviceConnected
35.474 68	ITM - SW Stimuli	r3, {r2, #0x08}		SPI_I2S_GetFlagStatus
35.474 91	ITM - Event Counters	0x08018206		main
35.475 14	ITM - Exceptions	lr		SPI_I2S_GetFlagStatus
35.475 37	ITM - PC Samples	{r4-r7, pc}		ADC_RegularChannelConf..
35.475 59	ITM - Data Read	BLW USBH_Process (0x080...		CallUsbhProcess
35.475 82	ITM - Data Write	LDR r0, [pc, #16] ; @0x080...		SPI2_SendByte
35.476 05	ITM - Data Read/Write	ADD sp, sp, #0x14		CheckForTanklevelAlarm
35.476 28	X: 0x08009AFA	BX lr		SPI_I2S_GetFlagStatus
35.476 50	X: 0x0800F67A	B 0x0800F63A		SPI2_ReceiveByte
35.476 73	X: 0x0800F8A0	CBZ r3, 0x0800F8A6		SPI_I2S_GetFlagStatus
35.476 96	X: 0x0800F67C	BLW SPI_I2S_GetFlagStatu...		SPI2_SendByte
35.477 19	X: 0x08009DA8	POP {r4-r11, pc}		CheckForTanklevelAlarm
35.477 42	X: 0x0800F65E	MOVS r1, #0x02		SPI2_SendByte
35.477 64	X: 0x0800F644	BEQ 0x0800F636		SPI2_ReceiveByte
35.477 87	X: 0x0800F6AA	LDR r0, [pc, #8] ; @0x0800...		SPI2_SetCsLow
35.478 10	X: 0x0800F89A	MOVS r0, #0x00		SPI_I2S_GetFlagStatus
35.478 33	X: 0x08009D94	ADD r0, r1, r0, LSL #2		CheckForTanklevelAlarm
35.478 57	X: 0x0800F8A0	CBZ r3, 0x0800F8A6		SPI_I2S_GetFlagStatus
35.478 78	X: 0x08009566	POP {r4, pc}		AutoReset
35.479 01	X: 0x0800F674	BLW AutoReset (0x080095...		SPI2_SendByte
35.479 24	X: 0x08009566	POP {r4, pc}		AutoReset
35.479 46	X: 0x080182F2	BLW ProcessTank (0x0800...		main
35.479 69	X: 0x0800F89C	LDRH r3, {r2, #0x08}		SPI_I2S_GetFlagStatus
35.479 92	X: 0x0800F636	BLW AutoReset (0x080095...		SPI2_ReceiveByte
35.480 15	X: 0x0800F89C	LDRH r3, {r2, #0x08}		SPI_I2S_GetFlagStatus
35.480 37	X: 0x0800BFF4	LDR r1, loc #41 ; @0x0800...		IWDG_ReloadCounter