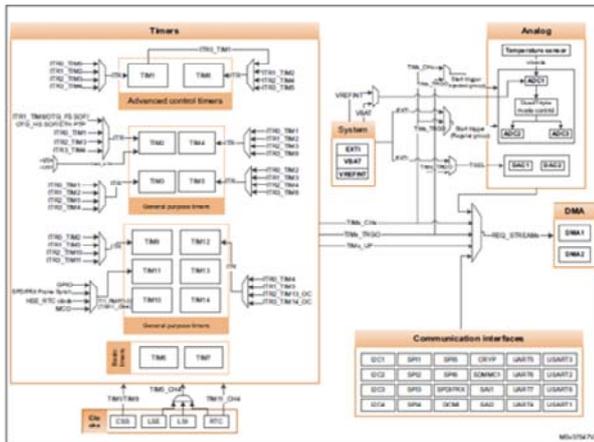## STM32F7 – Interconnect Matrix

Interconnect Matrix

Revision 1.0

Hello and welcome to this presentation of the STM32 Interconnect Matrix. It covers the main features of this matrix, which is widely used to connect various internal peripherals between each other.

- The Interconnect Matrix provides direct connections between peripherals.

Application benefits

- Time-predictable operations
- Decreased power consumption
- Reduced GPIO use

The Interconnect Matrix integrated inside STM32 microcontrollers provides direct connections between peripherals.

Applications benefit from these interconnections to ensure time-predictable operations and decrease power consumption. This matrix avoids complex management of peripheral communications through reading/writing registers using CPU instructions. In some cases, it reduces the need to loop a signal from a source to a destination through a dedicated GPIO.

- Direct, autonomous connections between peripherals
    - Remove latency in comparison to software event handling
    - Save CPU resources
    - Remove the need for looping signals through dedicated GPIOs
    - Allow automating various processes and regulation loops
    - Improve security of MCU-controlled applications

- Can operate during low-power modes (peripheral dependent)

The Interconnect Matrix offers two main features. First, it ensures direct and autonomous connections between peripherals, allowing to remove latency in comparison to software handling.
Second, the interconnection between certain peripherals can operate during low-power modes, avoiding having to use the CPU.

Interconnect sources and destinations

Plenty of interconnect possibilities available

- **Source peripherals**
  - Timers: TIMx and RTC
  - Analog IPs: ADCx, DACx, VrefInt, VBAT, and temperature sensor
  - Clocks: HSE, LSE, LSI, and HSI
  - EXTI, DFSDM, USB, and ETH
  - System errors (CSS, parity, ECC, and Lock-up)

- **Destination peripherals**
  - Timers: TIMx and LPTIM
  - Analog IPs: ADCx, and DACx
  - DFSDM, DMA, and EXTI

		Users																	
		Timers												DMA		Analog			
																ADC			DAC
Generators		TIM1	TIM2	TIM3	TIM4	TIM5	TIM6	TIM7	TIM8	TIM9	TIM10	TIM11	TIM12	DMA1	DMA2	1	2	3	1	2
TIMERS	TIM1	-	X	X	X	-	-	-	X	-	-	-	-	X	-	X	X	X	-	-
	TIM2	X	-	X	X	X	-	-	X	X	-	-	-	X	-	X	X	X	X	X
	TIM3	X	X	-	X	X	-	-	X	-	-	-	-	-	-	X	X	X	-	-
	TIM4	X	X	X	-	X	-	-	X	-	-	-	-	-	-	X	X	X	X	X
	TIM5	X	-	-	-	-	-	-	-	-	-	-	-	-	-	X	X	X	X	X
	TIM6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	X	X	X	X
	TIM7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	X	X
	TIM8	-	X	-	X	X	-	-	-	-	-	-	-	-	-	X	X	X	X	X
	TIM10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	TIM11	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	TIM13	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-
	TIM14	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-
	TIM15	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Analog	ADC 1	-	-	-	-	-	-	-	-	-	-	-	-	X	-	X	X	-	-	-
	ADC 2	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-
	ADC 3	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-
	DAC 1	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-

STM32F7 Series peripheral interconnections (AN4676)

The main peripherals having direct, autonomous interconnections are:
- connection sources: timers, analog IPs, clocks, extended interrupt/event controllers, digital filters for sigma-delta modulators, USB interfaces and system errors
- connection targets: timers, analog IPs, digital filters for sigma-delta modulators and direct memory access controllers

Several interconnections are able to work in low-power modes

- All interconnections work in the following power modes:
  - Run
  - Sleep

- Some interconnections work in Stop mode:
  - Ethernet MAC, USB, LPTIM, MDIOS, RTC and PVD connection to EXTI
 - Connections from RTC, GPIO to low-power timer (LPTIM1)

Peripherals can be interconnected using the Interconnect Matrix even when the circuit is in a low-power mode.
The fully supported power modes are Run and Sleep.
The connections from the real-time clock or GPIO to low-power timers can also be used in Stop mode, as well as Ethernet MAC, USB, low-power timers, management data input/output (MDIO) slaves, real-time clocks and the programmable voltage detector (PVD) connection to the external interrupt/event controller.
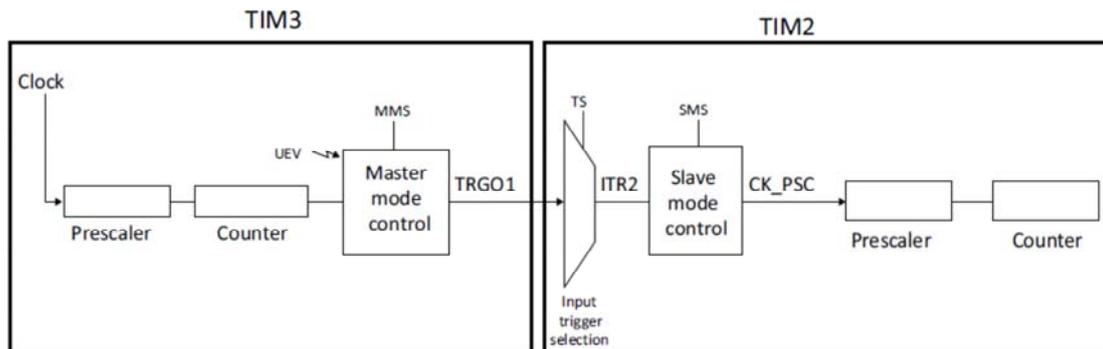
The Interconnect Matrix can be used for synchronizing or chaining timers allowing, for example, a master timer to reset or trigger a second slave timer.

This slide shows a simple example of timer synchronization.
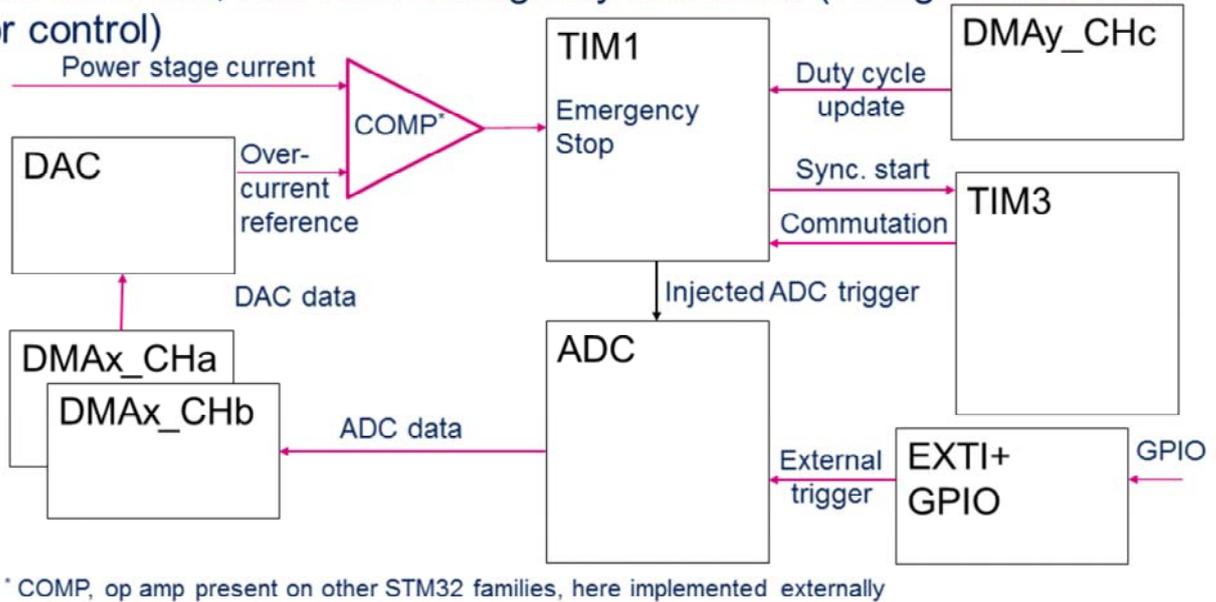Timer 3 is used as the Master Timer and can reset, start, stop or clock Timer 2 configured in Slave mode.
Timer 3 is clocking Timer 2 so that it acts as a prescaler for Timer 2.

Timers can also be triggered by a short-circuit detection on a digital filter for sigma-delta modulators, when a USB Start Of Frame is detected, or by a real-time clock interrupt at a given time or at a regular interval. All these use cases are enabled thanks to the propagation of these triggering signals through the Interconnect Matrix.

Application examples (2)

- Timers with ADC, DMA and emergency shutdown (voltage inverters or motor control)

Power stage current → COMP* → TIM1 Emergency Stop

DAC — Over-current reference → COMP*

DMAy_CHc → Duty cycle update → TIM1

TIM1 → Sync. start → TIM3

TIM3 → Commutation → TIM1

DAC → DAC data → DMAx_CHa / DMAx_CHb

TIM1 → Injected ADC trigger → ADC

ADC → ADC data → DMAx_CHb

EXTI+ GPIO → External trigger → ADC

GPIO → EXTI+ GPIO

* COMP, op amp present on other STM32 families, here implemented externally

This example generates 3-phase PWM signals in Timer 1. The commutation and duty cycle update are generated via the DMA, respectively Timer 3.

The overcurrent is guarded by a comparator (external in STM32F7 family), whose reference is driven by the digital-to-analog converter (DAC) and changed via the DMA controller.

Phase voltage and currents are measured by an analog-to-digital converter (ADC), which samples synchronously with event generated by Timer 1, channel 4. Other voltage sources are measured on external events, detected by the GPIO and routed via the EXTI unit to the ADC regular trigger.

The whole mechanism runs autonomously just after proper setup, so the CPU can safely execute speed or voltage regulation algorithms without the need for low-level control.

Other application examples

- Triggering of ADC, DAC or DFSDM (by Timers or EXTI)
- Triggering of timers (by DFSDM, RTC or USB)
- Triggering of DMA data transfer from memory to DAC (by Timer)
- Calibration of HSI/LSI clocks by Timer
- Dual-ADC or Triple-ADC modes (synchronization and load balancing)
- Temperature and supply + battery voltage monitoring
- Analog IP interconnect (DAC, op amp to ADC, COMP)
- Protection of timer-driven power switches (System Error to Timer)

The Interconnect Matrix is mostly used for:
- Triggering an ADC, DAC, digital filter for sigma-delta modulator through a timer event or an external interrupt,
- Triggering a timer through an ADC or DFSDM watchdog signal when a predefined threshold value is crossed by the analog input,
- Triggering a DMA data transfer from memory to the DAC by a timer to allow a frequency-controlled conversion,
- Calibrating HSI/LSI clocks, for example measuring the external oscillator LSE frequency by a timer clocked by the calibrated internal oscillator,
- Dual-ADC or Triple-ADC mode, using ADC1 as the master to trigger a start of conversion for the ADC2 and ADC3 slave,
- Monitoring the temperature of a connected internal

temperature sensor or the VBAT to ADC voltage,
- Analog IP interconnect, for example, DAC to an ADC,
- Protecting timer-driven power switches through the direct connection of System Error signals to the timer break input.

Interconnection in details

- Both peripherals need to be active and configured
- Their inputs and outputs need to be active and input multiplexer configured (e.g. TIM1 TRGO enabled, TIM3 slave mode enabled and its TRGI configured as input ITR0).
  - ➤ There is no specific interconnection peripheral to be configured, details for each peripheral can be found in Reference manual separately!
- Some peripheral inputs share connections with other peripherals and the associated GPIO pin (e.g. Comparator and Timer CHx inputs). In such cases, the GPIO Alternate Function must be deactivated to avoid interference.
- ADCx and COMP inputs can be shared, but two ADCs must not sample the same input at the same time.
- Peripherals with analogue outputs, when active, override the associated GPIO functionality (e.g. DAC)

life.augmented

These recommendations should be followed to ensure a proper interconnection in between the peripherals and avoid unexpected behaviors:
- Both peripherals need to be active and their inputs and outputs configured.
- When peripheral inputs share connections with other peripherals, associated GPIO alternate functions must be deactivated to avoid interference.
- Peripherals with active analogue outputs override the associated GPIO functionality.

- For more details, please refer to:
  - STM32F7 Series peripheral interconnections application note (AN4676)
  - STM32F7 Reference manuals (RM0385, RM0410)
  - STM32F7 Datasheets

For more details about the Interconnect Matrix, refer to the *STM32F7 Series peripheral interconnections* application note (AN4676).
Specific details can be found in the specific sections for each peripheral in the Reference Manual as well as in the Datasheet.