

# STM32WB – Interconnect Matrix

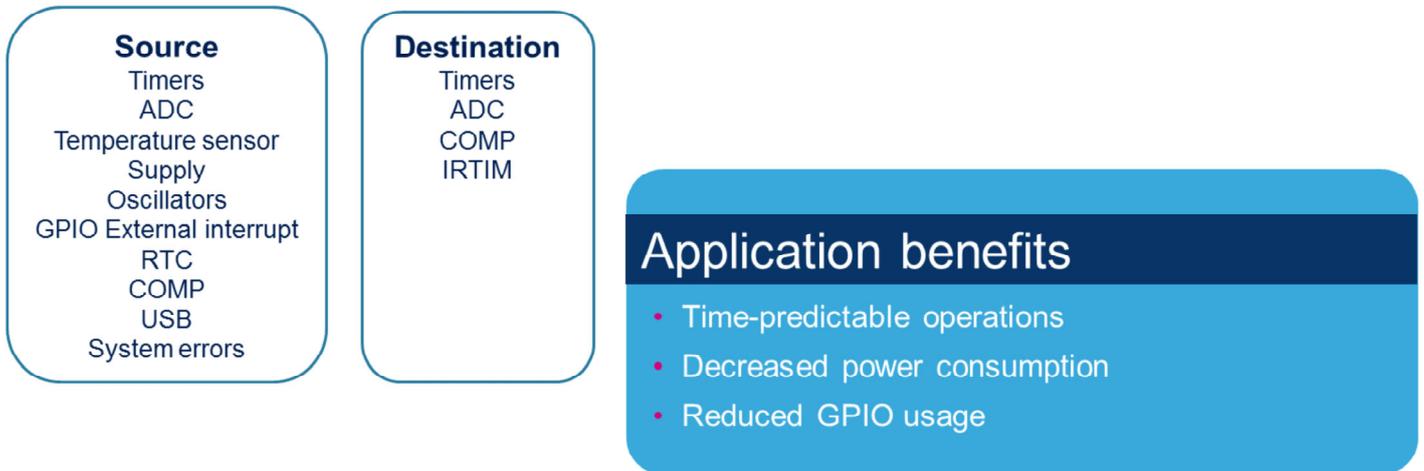
Interconnect Matrix

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.

- Provides direct connections between peripherals.



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

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

- Direct, autonomous connections between peripherals
  - Removes latency in regards to software handling
  - Saves CPU resources
  - Removes the need for looping signals through a dedicated GPIO
- Can operate during Low-power mode (depending on peripheral)



The Interconnect Matrix offers two main features. First, it ensures direct and autonomous connections between peripherals, allowing to remove latency in regards to software handling, thus saving GPIO and CPU resources.

Second, the interconnection between certain peripherals can even operate during low-power modes.

## • Plenty of interconnect possibilities available

- Source peripherals
  - Timers: TIMx, LPTIMx, RTC
  - Analog IPs: ADCx, COMPx, VrefInt, VBAT, Temp Sensor
  - Clocks: HSE, LSE, HSI16, MSI, LSI, MCO
  - EXTI, USB
  - System error
- Destination peripherals
  - Timers: TIMx, LPTIMx, IRTIM
  - Analog IPs: ADCx, COMPx



The main peripherals having direct, autonomous interconnections are timers, analog IPs, clocks, extended interrupt/event controller, USB and System Error for the connection sources.

And timers, and analog IPs for the connection destinations.

- **Most interconnections are able to work in low-power modes**

- All interconnections (except USB to TIM2) work in the following power modes:
  - Run, Sleep, Low-power run and Low-power sleep modes
- Connections from RTC, COMP1, COMP2 to low-power timer (LPTIM1, LPTIM2)
  - Also work in Stop0, Stop 1 and Stop 2 (LPTIM1 only) modes
- Connection from USB to timer TIM2
  - Only works in Run and Sleep modes



Peripherals can be interconnected using the Interconnect Matrix even when the circuit is in a low-power mode.

The operating modes that can be used are: Run, Low-power run, Sleep and Low-power sleep modes; except for the USB to Timer 2 connection, which can only be used in Run and Sleep modes.

The connections from the real-time clock or comparators to low-power timers can also be used in Stop0, Stop 1 and Stop 2 modes for Low-power Timer 1.

- Timer synchronization or chaining
- Triggering of ADC or COMP (by Timer or EXTI)
- Triggering of timers (by ADC, RTC, COMP or USB)
- Calibration of HSI16, MSI, LSI clocks
- Temperature and voltage monitoring
- Protection of timer-driven power switches (System Error to Timer)
- Infrared signal generation (Timer to IRTIM)



The Interconnect Matrix is mostly used for:

- Synchronizing or chaining timers, for example allowing a master timer to reset or trigger a second slave timer
- Triggering an ADC or comparator through a timer event or an external interrupt
- Triggering a timer through an ADC when a predefined threshold value is crossed by the analog input
- Timers can also be triggered based on a comparator output value, or when a USB Start Of Frame is detected
- Calibrating HSI16, MSI, LSI clocks, for example measuring the external oscillator LSE frequency by a timer clocked by the calibrated internal oscillator
- Monitoring the temperature of a connected internal temperature sensor or the VBAT to ADC voltage
- Protecting timer-driven power switches through the

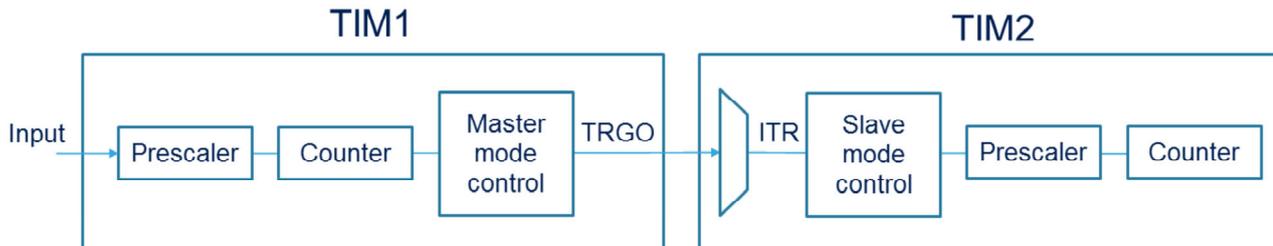
direct connection of System Error signals to the timer  
break input

- Infrared pulse modulation signal waveform generation  
using 2 timers

# Timer synchronization example

7

- Timer 1 can act as a pre-scaler for Timer 2



This slide shows a simple example of timer synchronization.

The Timer 1 is used as the Master Timer and can reset, start, stop or clock the Timer 2 configured in Slave mode. In this example, Timer 1 is clocking the Timer 2 so that it acts as a pre-scaler for Timer 2.

- For more details, please refer to:
  - Reference manuals for STM32WB microcontrollers



For more details about the Interconnect Matrix, refer to reference manual for STM32WB microcontrollers.