GPIO features

- Up to 140 multifunction bi-directional I/O ports available on 176 pin package
- Almost standard I/Os are 5V tolerant
- All Standard I/Os are shared in 9 ports (GPIOA..GPIOI)
- Atomic Bit Set and Bit Reset using BSRR register
- GPIO connected to AHB bus: max toggling frequency = \( f_{\text{AHB}}/2 = 84 \text{ MHz} \)
- Configurable Output Speed up to 100 MHz (2MHz,25MHz,50MHz,100MHz)
- Locking mechanism (GPIOx_LCKR) provided to freeze the I/O configuration
- Up to 140 GPIOs can be set-up as external interrupt (up to 16 lines at time) able to wake-up the MCU from low power modes
- Most of the I/O pins are shared with Alternate Functions pins connected to onboard peripherals through a multiplexer that allows only one peripheral’s alternate function to be connected to an I/O pin at a time
## GPIO Configuration Modes

<table>
<thead>
<tr>
<th>MODER(i) [1:0]</th>
<th>OTYPER(i) [1:0]</th>
<th>PUPDR(i) [1:0]</th>
<th>I/O configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>0</td>
<td>0</td>
<td>Output Push Pull</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>Output Push Pull with Pull-up</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>Output Push Pull with Pull-down</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>Output Open Drain</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>Output Open Drain with Pull-up</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>Output Open Drain with Pull-down</td>
</tr>
<tr>
<td>00</td>
<td>x</td>
<td>0</td>
<td>Input floating</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>Input with Pull-up</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>Input with Pull-down</td>
</tr>
<tr>
<td>11</td>
<td>x</td>
<td>x</td>
<td>Analog mode</td>
</tr>
</tbody>
</table>

* In output mode, the I/O speed is configurable through OSPEEDR register: 2MHz, 25MHz, 50MHz or 100 MHz

---

*Alternate Function Input To On-chip Peripherals*

*Alternate Function Output*

*Input Data Register*

*Output Data Register*

*Bit Set/Reset Register*

*OnOff*

*Push-Pull Open Drain*

*Input Driver*

---

(1) VDD_FT is a potential specific to five-volt tolerant I/Os and different from VDD.
Alternate Functions features

- Most of the peripherals shares the same pin (like USARTx_Tx, TIMx_CH2, I2Cx_SCL, SPIx_MISO, EVENTOUT…)

- Alternate functions multiplexers prevent to have several peripheral’s function pin to be connected to a specific I/O at a time.

- Some Alternate function pins are remapped to give the possibility to optimize the number of peripherals used in parallel.
System Configuration

- Configure (2 bits) the type of memory accessible at address 0x00000000. These bits are used to select the physical remap by software and so, bypass the BOOT pins.
  - 00: Main Flash memory mapped at 0x0000 0000
  - 01: System Flash memory mapped at 0x0000 0000
  - 10: FSMC (NOR/SRAM bank1) mapped at 0x0000 0000
  - 11: Embedded SRAM (112kB) mapped at 0x0000 0000

- Select the Ethernet PHY interface (MII or RMII)

- Manage the external interrupt line connection to the GPIOs:

* x can be 0 to 15 for all ports]
EXTI module: from pin to NVIC

GPIOA_0
GPIOB_0
GPIOI_0
GPIOA_1
GPIOB_1
GPIOI_1
GPIOA_15
GPIOB_15
GPIOI_15
Channel 0
Channel 1
Channel 15

Input floating

EXTI
Event
Interrupt
Enable/Disable
Wakeup

Exti_0
Exti_1
Exti_2
Exti_3
Exti_4
Exti_9-5
Exti_15-10
PVD_IRQ
RTC_IRQ

PVD
RTC_Alarm
USB OTG HS Wkup
ETH Wkup
USB OTG FS Wkup
RTC Tamper
RTC Wkup

NVIC

CORTEX M4

STM32 F4
**ADC Features (1/2)**

- **3 ADCs**: ADC1 (master), ADC2 and ADC3 (slaves)
- Maximum frequency of the ADC analog clock is 36MHz.
- 12-bits, 10-bits, 8-bits or 6-bits configurable resolution.
- ADC conversion rate with 12 bit resolution is up to:
  - 2.4 M.sample/s in single ADC mode,
  - 4.5 M.sample/s in dual interleaved ADC mode,
  - 7.2 M.sample/s in triple interleaved ADC mode.
- Conversion range: 0 to 3.6 V.
- ADC supply requirement: VDDA = 2.4V to 3.6V at full speed and down to 1.65V at lower speed.
- **Up to 24 external channels**.
- **3 ADC1 internal channels** connected to:
  - Temperature sensor,
  - Internal voltage reference: VREFINT (1.2V typ),
  - VBAT for internal battery monitoring.
ADC Features (2/2)

- External trigger option for both regular and injected conversion.
- Single and continuous conversion modes.
- Scan mode for automatic conversion of channel 0 to channel ‘n’.
- Left or right data alignment with in-built data coherency.
- Channel by channel programmable sampling time.
- Discontinuous mode.
- Dual/Triple mode (with ADC1 and ADC2 or all 3 ADCs).
- DMA capability
- Analog Watchdog on high and low thresholds.
- Interrupt generation on:
  - End of Conversion
  - End of Injected conversion
  - Analog watchdog
  - Overrun
## ADC speed performances

<table>
<thead>
<tr>
<th>AHBCLK</th>
<th>APB2CLK</th>
<th>ADC_CLK</th>
<th>ADC speed (15 cycles)</th>
</tr>
</thead>
<tbody>
<tr>
<td>168MHz</td>
<td>(a) 84MHz</td>
<td>(2) 21MHz</td>
<td>0.714μs 1.4 Msample/s</td>
</tr>
<tr>
<td>144MHz</td>
<td>(a) 72MHz</td>
<td>(1) 36MHz</td>
<td>0.416μs 2.4 Msample/s</td>
</tr>
<tr>
<td>120MHz</td>
<td>(a) 60MHz</td>
<td>(1) 30MHz</td>
<td>0.5μs 2 Msample/s</td>
</tr>
<tr>
<td>96MHz</td>
<td>(a) 48MHz</td>
<td>(1) 24MHz</td>
<td>0.625μs 1.6 Msample/s</td>
</tr>
<tr>
<td>72MHz</td>
<td>(b) 72MHz</td>
<td>(1) 36MHz</td>
<td>0.416μs 2.4 Msample/s</td>
</tr>
</tbody>
</table>

(1) ADC_PRESC = /2
(2) ADC_PRESC = /4
(a) APB_PRESC = /2
(b) APB_PRESC = /1

SYSCLK (168MHz max) → AHBCLK (168MHz max) → APB2CLK (84MHz max) → ADC_CLK (36MHz max)

AHB_PRESC /1,2,...512 → APB2_PRESC /1, 2, 4, 8,16 → ADC_PRESC /2,4, 6, 8
**ADC Conversion Time**

- ADCCLK, up to 36MHz, taken from PCLK through a prescaler (Div2, Div4, Div6 and Div8).
- Programmable sample time for each channel (from 4 to 480 clock cycles)
- Total conversion Time: \( T_{\text{Sampling}} + T_{\text{conversion}} \)

With Sample time = 3 cycles @ ADC_CLK = 36MHz \( \Rightarrow \) total conversion time is equal to:

<table>
<thead>
<tr>
<th>Resolution</th>
<th>( T_{\text{Conversion}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>12 bits</td>
<td>12 Cycles</td>
</tr>
<tr>
<td>10 bits</td>
<td>10 Cycles</td>
</tr>
<tr>
<td>8 bits</td>
<td>8 Cycles</td>
</tr>
<tr>
<td>6 bits</td>
<td>6 Cycles</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>resolution</th>
<th>Total conversion Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>12 bits</td>
<td>12 + 3 = 15 cycles</td>
</tr>
<tr>
<td>10 bits</td>
<td>10 + 3 = 13 cycles</td>
</tr>
<tr>
<td>8 bits</td>
<td>8 + 3 = 11 cycles</td>
</tr>
<tr>
<td>6 bits</td>
<td>6 + 3 = 9 cycles</td>
</tr>
</tbody>
</table>
ADC Analog Watchdog

- 12-bit programmable analog watchdog low and high thresholds
- Enabled on one or all converted channels: one regular or/and injected channel, all injected or/and regular channels.
- Interrupt generation on low or high thresholds detection
ADC dual modes

- ADCs: ADC1 master and ADC2 slave, ADC3 is independently.
- The start of conversion is triggered alternately or simultaneously by the ADC1 master to the ADC2 slave depending on the mode selected.
- 6 ADC dual modes
ADC Triple modes

- ADCs: ADC1 master, ADC2 and ADC3 slaves.
- The start of conversion is triggered alternately or simultaneously by the ADC1 master to the ADC2 and ADC3 slaves depending on the mode selected.
- 6 ADC Triple modes.
DAC Features

- Two DAC converters: one output channel for each one
- 8-bit or 12-bit monotonic output
- Left or right data alignment in 12-bit mode
- Synchronized update capability
- Noise-wave or Triangular-wave generation
- Dual DAC channel independent or simultaneous conversions
- 11 dual channel modes
- DMA capability for each channel
- External triggers for conversion
- DAC supply requirement: 1.8V to 3.6 V
- Conversion range: 0 to 3.6 V
- DAC outputs range: 0 ≤ DAC_OUTx ≤ VREF+ (VREF+ is available only in 100, 144 and 176 pins package)
On board there are following timers available:

- **2x advanced 16bit** timers (TIM1,8)
- **2x general purpose 32bit** timers (TIM2,5)
- **8x general purpose 16bit** timers (TIM3,4,9,10..14)
- **2x simple 16bit** timers for DAC (TIM6,7)
- **1x 24bit system timer** (SysTick)
General Purpose timer Features overview

- TIM2, 3, 4 and 5 on Low Speed APB (APB1)
- Internal clock up to **84 MHz** (if AHB/APB1 prescaler distinct from 1)
- 16-bit Counter for TIM3 and 4
- 32-bit Counter for TIM2 and 5
  - Up, down and centered counting modes
  - Auto Reload
- 4 x 16 High resolution Capture Compare Channels
  - Programmable direction of the channel: input/output
  - Output Compare
  - PWM
  - Input Capture, PWM Input Capture
  - One Pulse Mode
- Synchronization
  - Timer Master/Slave
  - Synchronisation with external trigger
  - Triggered or gated mode
- Encoder interface
- 6 Independent IRQ/DMA Requests generation
  - At each Update Event
  - At each Capture Compare Events
  - At each Input Trigger
Advanced timer Features overview

- TIM1 and TIM8 on High Speed APB (APB2)
- Internal clock up to **168 MHz** (if AHB/APB2 prescaler distinct from 1)
- 16-bit Counter
  - Up, down and centered counting modes
  - Auto Reload
- 4 x 16 High resolution Capture Channels
  - Output Compare
  - PWM
  - Input Capture, PWM input Capture
  - One Pulse Mode
- 6 Complementary outputs: Channel 1, 2 and 3
- Output Idle state selection independently for each output
- Polarity selection independently for each output
- Programmable PWM repetition counter
- Hall sensor interface
- Encoder interface
- 8 Independent IRQ/DMA Requests Generation
  - At each Update Event
  - At each Capture Compare Events
  - At each Trigger Input Event
  - At each Break Event
  - At each Capture Compare Update
- Embedded Safety features
  - Break input
  - Lockable unit configuration: 3 possible Lock level.
General Purpose 2 Channels timer (TIM9 & TIM12) Features overview

- TIM9 on High speed APB (APB2) and TIM12 on Low Speed APB (APB1)
- Internal clock up to **168 MHz** and **84 MHz** respectively
- 16-bit Counter
  - Up counting mode
  - Auto Reload
- 2 x 16 High resolution Capture Compare Channels
  - Programmable direction of the channel: input/output
  - Output Compare
  - PWM
  - Input Capture, PWM Input Capture
  - One Pulse Mode
- Synchronization Timer Master/Slave
  - Synchronization with external trigger
  - Triggered or gated mode
- Independent IRQ Requests generation
  - At each Update Event
  - At each Capture Compare Events
  - At each Input Trigger
General Purpose 1 Channels timer (TIM10..11 & TIM13..14) Features overview

- TIM10..11 on High speed APB (APB2) and TIM13..14 on Low Speed APB (APB1)
- Internal clock up to 168 MHz for TIM10/11
- Internal clock up to 84 MHz for TIM13/14
- 16-bit Counter
  - Up counting mode
  - Auto Reload
- 2 x 16 High resolution Capture Compare Channels
  - Programmable direction of the channel: input/output
  - Output Compare
  - PWM
  - Input Capture
- Independent IRQ Requests generation
  - At each Update Event
  - At each Capture Compare Events
Synchronization – Configuration examples (1/3)

- **Cascade mode:**
  - TIM_A used as master timer for TIM_B, TIM_B configured as TIM_A slave and master for TIM_C.
One Master several slaves: TIM_A used as master for TIM_B, TIM_C and TIM_D.
Synchronization – Configuration examples (3/3)

- Timers and external trigger synchronization
  - TIM_A, TIM_B and TIM_C are slaves for an external signal connected to respective Timers inputs.

![Diagram with TIM_A, TIM_B, and TIM_C block diagrams connected with External Trigger signal](image-url)
### STM32F4xx Timer features overview (1/2)

<table>
<thead>
<tr>
<th></th>
<th>Counter resolution</th>
<th>Counter type</th>
<th>Prescaler factor</th>
<th>DMA</th>
<th>Capture Compare Channels</th>
<th>Complementary output</th>
<th>Synchronization</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Advanced TIM1 and TIM8</strong></td>
<td>16 bit</td>
<td>up, down and up/down</td>
<td>1..65536</td>
<td>YES</td>
<td>4</td>
<td>3</td>
<td>YES</td>
</tr>
<tr>
<td><strong>General purpose (1) TIM2 and TIM5</strong></td>
<td>32 bit</td>
<td>up, down and up/down</td>
<td>1..65536</td>
<td>YES</td>
<td>4</td>
<td>0</td>
<td>YES</td>
</tr>
<tr>
<td><strong>General purpose TIM3 and TIM4</strong></td>
<td>16 bit</td>
<td>up, down and up/down</td>
<td>1..65536</td>
<td>YES</td>
<td>4</td>
<td>0</td>
<td>YES</td>
</tr>
<tr>
<td><strong>Basics TIM6 and TIM7</strong></td>
<td>16 bit</td>
<td>up</td>
<td>1..65536</td>
<td>YES</td>
<td>0</td>
<td>0</td>
<td>YES</td>
</tr>
<tr>
<td><strong>1 Channel (2) TIM10..11 and TIM13..14 (2)</strong></td>
<td>16 bit</td>
<td>up</td>
<td>1..65536</td>
<td>NO</td>
<td>1</td>
<td>0</td>
<td>YES (OC signal)</td>
</tr>
<tr>
<td><strong>2 Channel(2) TIM9 and TIM12</strong></td>
<td>16 bit</td>
<td>up</td>
<td>1..65536</td>
<td>NO</td>
<td>2</td>
<td>0</td>
<td>NO</td>
</tr>
</tbody>
</table>

(1) Same as STM32F2xx 32-Bit Timers
(2) These Timers are identical to STM32F2xx and STM32F1 XL Timers
## STM32F4xx Timer features overview 2/2

<table>
<thead>
<tr>
<th>Counter clock source</th>
<th>Output Compare</th>
<th>PWM</th>
<th>Input Capture</th>
<th>PWMI</th>
<th>OPM</th>
<th>Encoder interface</th>
<th>Hall sensor interface</th>
<th>XOR Input</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Advanced</strong>&lt;br&gt;<strong>TIM1 and TIM8</strong>&lt;br&gt;- Internal clock APB2&lt;br&gt;- External clock: ETR/TI1/TI2/TI3/TI4 pins&lt;br&gt;- Internal Trigger: ITR1/ITR2/ITR3/ITR4&lt;br&gt;- Slave mode</td>
<td>7</td>
<td>7</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>General Purpose</strong>&lt;br&gt;<strong>TIM2 and TIM5</strong>&lt;br&gt;- Internal clock APB1&lt;br&gt;- External clock: ETR/TI1/TI2/TI3/TI4 pins&lt;br&gt;- Internal Trigger: ITR1/ITR2/ITR3/ITR4&lt;br&gt;- Slave mode</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>General Purpose</strong>&lt;br&gt;<strong>TIM3 and TIM4</strong>&lt;br&gt;- Internal clock APB1&lt;br&gt;- External clock: ETR/TI1/TI2/TI3/TI4 pins&lt;br&gt;- Internal Trigger: ITR1/ITR2/ITR3/ITR4&lt;br&gt;- Slave mode</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Basics</strong>&lt;br&gt;<strong>TIM6 and TIM7</strong>&lt;br&gt;- Internal clock APB1</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td><strong>1 Channel</strong>&lt;br&gt;<strong>TIM10/T1 and 13/14</strong>&lt;br&gt;- Internal clock APB1/APB2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td><strong>2 Channel</strong>&lt;br&gt;<strong>TIM9 and TIM12</strong>&lt;br&gt;- Internal clock APB1/APB2&lt;br&gt;- External clock: TI1/TI2/TI3/TI4 pins&lt;br&gt;- Internal Trigger: ITR1/ITR2/ITR3/ITR4&lt;br&gt;- Slave mode</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>
RTC Features

- Ultra-low power battery supply current < 1uA with RTC ON.
- Calendar with **sub seconds**, seconds, minutes, hours, week day, date, month, and year.
- Daylight saving compensation programmable by software
- Two programmable alarms with interrupt function. The alarms can be triggered by any combination of the calendar fields.
- A periodic flag triggering an automatic wakeup interrupt. This flag is issued by a 16-bit auto-reload timer with programmable resolution. This timer is also called ‘wakeup timer’.
- A second clock source (50 or 60Hz) can be used to update the calendar.
- Maskable interrupts/events:
  - Alarm A, Alarm B, Wakeup interrupt, Time-stamp, Tamper detection
- Digital calibration circuit (periodic counter correction) to achieve 5 ppm accuracy
- Time-stamp function for event saving with sub second precision (1 event)
- 20 backup registers (80 bytes) which are reset when an tamper detection event occurs.
STM32F4
Communication peripherals
Communication Peripherals

INTER-INTEGRATED CIRCUIT INTERFACE (I²C)

Same as STM32F-2
I²C Features (1/2)

- Multi Master and slave capability
- Controls all I²C bus specific sequencing, protocol, arbitration and timing
- Standard and fast I²C mode (up to 400kHz)
- 7-bit and 10-bit addressing modes
- Dual Addressing Capability to acknowledge 2 slave addresses
- Status flags:
  - Transmitter/Receiver mode flag
  - End-of-Byte transmission flag
  - I²C busy flag
- Configurable PEC (Packet Error Checking) Generation or Verification:
  - PEC value can be transmitted as last byte in Tx mode
  - PEC error checking for last received byte
I²C Features (2/2)

- Error flags:
  - Arbitration lost condition for master mode
  - Acknowledgement failure after address/data transmission
  - Detection of misplaced start or stop condition
  - Overrun/Underrun if clock stretching is disabled

- 2 Interrupt vectors:
  - 1 Interrupt for successful address/data communication
  - 1 Interrupt for error condition

- 1-byte buffer with DMA capability
- SMBus 2.0 Compatibility
- PMBus Compatibility
Communication Peripherals

UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (USART)

Same as STM32F-2
### USART Features (1/2)

- **6 USARTs**: USART1 & USART6 on APB2 and USART2,3,4,5 on APB1
- Fully-programmable serial interface characteristics:
  - Data can be 8 or 9 bits
  - Even, odd or no-parity bit generation and detection
  - 0.5, 1, 1.5 or 2 stop bit generation
  - **Oversampling by 16 (default) or by 8**
  - Programmable baud rate generator
    - Integer part (12 bits)
    - Fractional part (4 bits)
    - Baud rate for standard USART (SPI mode included)
      \[
      \text{Tx/Rx baud} = \frac{fck}{8 \times (2 - \text{OVR8}) \times \text{USARTDIV}}
      \]
    - Where:
      - **Tx/Rx baud**: desired baudrate
      - **OVR8**: oversampling by 8 (1 if enabled, 0 if disabled)
      - **fck**: APB frequency
      - **USARTDIV**: value to be programmed to the BRR register

Up to 10.5 Mbps
USART Features (2/2)

- Support **hardware flow control** (CTS and RTS)
- Dedicated transmission and reception flags (TxE and RxNE) with interrupt capability
- Support for DMA
  - Receive DMA request and Transmit DMA request
- 10 interrupt sources to ease software implementation
- LIN Master/Slave compatible
- Synchronous Mode: Master mode only
- IrDA SIR Encoder Decoder
- Smartcard Capability
- Single wire Half Duplex Communication
- Multi-Processor communication
  - USART can enter Mute mode
  - Mute mode: disable receive interrupts until next header detected
  - Wake up from mute mode (by idle line detection or address mark detection)
- **Support One Sample Bit method**: allows to disable noise detection (for noise-free applications) in order to increase the receiver’s tolerance to clock deviations.

Clock deviation tolerance up to 4.375%
SERIAL PERIPHERAL INTERFACE (SPI)

Same as STM32F-2

Communication Peripherals
SPI Features (1/2)

- **Up to 3 SPIs**: SPI1 on high speed APB2 and SPI2, SPI3 on low speed APB1
- **SPI2, SPI3 can work as SPI or I²S interface**
- **Full duplex synchronous transfers on 3 lines**
- **Simplex synchronous transfers on 2 lines with or without a bi-directional data line**
- **Programmable data frame size**: 8- or 16-bit transfer frame format selection
- **Programmable data order with MSB-first or LSB-first shifting**
- **Master or slave operation**
- **Programmable bit rate**: up to 37.5 MHz in Master/Slave mode
- **NSS management by hardware or software for both master and slave**: Dynamic change of Master/Slave operations
- **Motorola / TI mode** (master and slave operations).
SPI Features (2/2)

- Programmable clock polarity and phase
- Dedicated transmission and reception flags (Tx buffer Empty and Rx buffer Not Empty) with interrupt capability
- SPI bus busy status flag
- Master mode fault and overrun flags with interrupt capability
- Hardware CRC feature for reliable communication (CRC8, CRC16)
- Support for DMA
- Each SPI has a DMA Tx and Rx requests
- Each of the SPIs requests is mapped on a different DMA Stream: possibility to use DMA for all SPIs transfer direction at the same time
- Calculated CRC value is automatically transmitted at the end of data transfer
I²S Features (1/2)

- Two I²Ss: Available on SPI2 and SPI3 peripherals.
- Two I²Ss extension added for Full-Duplex communication.
- Dedicated PLL for high quality audio clock generation.
- Simplex/or Full duplex communication (transmitter and receiver)
- Can operate in master or slave configuration.
- 8-bit programmable linear prescaler to support all standard audio sample frequencies up to 192KHz.
- Programmable data format (16-, 24- or 32-bit data formats)
- Programmable packet frame (16-bit and 32-bit packet frames).
- Underrun flag in slave transmit mode, Overrun flag in receive mode and new de-synchronization flag in slave transmit/receive mode.
- 16-bit register for transmission and reception.
- Support for DMA (16-bit wide).
I²S Features (2/2)

- I²S protocols supported:
  - I²S Phillips standard.
  - MSB Justified standard (Left Justified).
  - LSB Justified standard (Right Justified).
  - PCM standard (with short and long frame synchronization on 16-bit channel frame or 16-bit data frame extended to 32-bit channel frame)

- Master clock may be output to drive an external audio component. Ratio is fixed at 256xFs (where Fs is the audio sampling frequency).

- **Note:** Since some SPI3/I²S3 pins are shared with JTAG pins, they are not controlled by the I/O controller and are reserved for JTAG usage (after each Reset). Prior to configure these pins, the user has to disable the JTAG and use the SWD interface (when debugging the application), or disable both JTAG/SWD interfaces (for standalone application).
Communication Peripherals

SD/SDIO MMC CARD HOST INTERFACE (SDIO)

Same as STM32F-2
SDIO Features

- Cards Clock Management: Rising and Falling edge, 8-bit prescaler, bypass, power save..
- Hardware Flow Control: to avoid FIFO underrun (TX mode) and overrun (RX mode) errors.
- A 32-bit wide, 32-word FIFO for Transmit and Receive
- DMA Transfer Capability
- Data Transfer: Configurable mode (Block or Stream), configurable data block size from 1 to 16384 bytes, configurable TimeOut
- 24 interrupt sources to ease software implementation
- CRC Check and generation
- SD I/O mode: SD I/O Interrupt, suspend/resume and Read Wait
- Data transfer up to 48 MHz
The SDIO consists of two parts:

- The SDIO adapter block provides all functions specific to the MMC/SD/SD I/O card such as the clock generation unit, command and data transfer.
- The APB2 interface accesses the SDIO adapter registers, and generates interrupt and DMA request signals.
SD/SDIO & MMC Cards

- The SDIO has 10 pins to control different kinds of memory cards
  - Only 6 pins (SDIO_CMD, SDIO_CK, SDIO_D[3:0]) at most for SD cards (SD full size, miniSD, microSD)
  - Only 6 pins (SDIO_CMD, SDIO_CK, SDIO_D[3:0]) at most for SDIO cards (SD full size, miniSD, microSD)
  - 10 pins (SDIO_CMD, SDIO_CK, SDIO_D[7:0]) at most for MMC cards (MMC full size, RS-MMC, MMC+ and MMCMobile)
FLEXIBLE STATIC MEMORY CONTROLLER (FSMC)

Same as STM32F-2
FSMC Features

- The Flexible Static Memory Controller has the following main features:
  - **4 Banks** to support External memory
  - FSMC external access frequency is **60MHz when HCLK is at 168Hz**
  - Independent chip select control for each memory bank
  - Independent configuration for each memory bank
  - Interfaces with static memory-mapped devices including:
    - static random access memory (SRAM)
    - read-only memory (ROM)
    - NOR/ OneNAND Flash memory
    - PSRAM
  - Interfaces parallel LCD modules: Intel 8080 and Motorola 6800
  - Supports burst mode access to synchronous devices (NOR Flash and PSRAM)
  - NAND Flash and 16-bit PC Cards
    - With ECC hardware up to 8 Kbyte for NAND memory
    - 3 possible interrupt sources (Level, Rising edge and falling edge)
  - Programmable timings to support a wide range of devices
  - External asynchronous wait control
  - Enhanced performance vs. STM32F10x
The FSMC consists of four main blocks:

- The AHB interface (including the IP configuration registers)
- The NOR Flash/PSRAM controller
- The NAND Flash/PC Card controller
- The external devices interface
FSMC Bank memory mapping

- For the FSMC, the external memory is divided into 4 fixed size banks of 4x64 MB each:
  - Bank 1 can be used to address NOR Flash, OneNAND or PSRAM memory devices.
  - Banks 2 and 3 can be used to address NAND Flash devices.
  - Bank 4 can be used to address a PC Card device.

<table>
<thead>
<tr>
<th>Bank</th>
<th>Capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bank 1</td>
<td>4x64 MB</td>
</tr>
<tr>
<td>Bank 2</td>
<td>256 MB</td>
</tr>
<tr>
<td>Bank 3</td>
<td>256 MB</td>
</tr>
<tr>
<td>Bank 4</td>
<td>256 MB</td>
</tr>
</tbody>
</table>

Supported Memory Type

- NOR/PSRAM/ SRAM/ CRAM/ OneNAND
- NAND Flash
- PC Card
USB 2.0 ON-THE-GO FULL SPEED (OTG FS)

Same as STM32F-2

Communication Peripherals
General Features

- Fully compliant with **Universal Serial Bus Revision 2.0** specification
- Dual Role Device (DRD) controller that supports both device and host functions compliant with **On-The-Go (OTG) Supplement Revision 1.3**
- Can be configured as host-only or device-only controller
- **Integrated PHY** with full support of the OTG mode
- Full-speed (12 Mbits/s) and low-speed (1.5 Mbits/s) operation (only full speed for device)
- **Dedicated RAM of 1.25 kB** with advanced FIFO management and dynamic memory allocation
Device Mode Features

- 1 bidirectional control endpoint0
- Up to 3 IN and 3 OUT endpoints configurable to support Bulk, Interrupt or Isochronous mode
- Shared RxFIFO for OUT endpoints
- Dedicated TxFIFO for each IN endpoint
- FIFO management with multi-packet transfer support
- Soft disconnection feature (removing internal D+ pull-up)
- USB suspend/resume with exit from STOP mode
Host Mode Features

- Up to 8 host channels (pipes) dynamically reconfigurable to any type of USB transfer

- Shared RxFIFO for IN channels

- Shared periodic TxFIFO for interrupt and isochronous OUT channels

- Shared non-periodic TxFIFO for bulk and control OUT channels

- Separate queue management for periodic and non-periodic transfer requests with up to 8 requests for each queue

- Built-in hardware scheduler for giving priority to periodic transfers request over non-periodic transfers requests

- FIFO management with Multi-packet transfer support
Embedded Full-speed OTG PHY Features

- FS/LS transceiver module used for Host/Device operation
- ID line detection for A/B device identification in OTG mode
- DP/DM integrated pull-up/pull-down resistors controlled by the USB core for device/host operation
- Vbus sensing and pulsing used for Session Request Protocol (SRP) in OTG mode
Hardware connections

Host-only Operation

Device-only Operation

STM32F4xx

Dual role OTG (Host/Device) Operation
Communication Peripherals

**USB 2.0 ON-THE-GO HIGH SPEED (OTG HS)**

Same as STM32F-2
Main Features

- Fully compatible (@ register level) with the full-speed USB OTG peripheral

- High-speed (480 Mbit/s), full-speed and low speed operation in host mode and High-speed/Full-speed in device mode

- Three PHY interfacing options
  - Internal full-speed PHY (as for FS peripheral)
  - I2C interface for full-speed I2C PHY
  - ULPI bus interface for high-speed PHY

- DMA support with a dedicated FIFO of 4Kbytes
Device mode Features

- Same as Full-speed mode with some extended/new features:
  - Up to 5 IN bulk, interrupt or isochronous endpoints (Vs 3 in FS)
  - Up to 5 OUT bulk, interrupt or isochronous endpoints (Vs 3 in FS)
  - Separate NVIC interrupt vector for EP1_IN
  - Separate NVIC interrupt vector for EP1_OUT

- NYET handshake sending
  - In High Speed mode, after receiving a packet, the core sends NYET handshake if it does not find threshold amount of free space available in the RxFIFO
Host mode features

- Same as Full-speed mode features
  - Up to 12 channels (Vs 8 channels in FS peripheral)
- High-speed protocol specific features
  - **PING protocol**: when a HS device is not ready to accept new packet, it sends the NYET or NAK handshake, in this case the host shouldn’t continue data sending, but it should start the PING protocol to check periodically if device is ready to resume operation
  - **SPLIT protocol**: when the host is connected to a HS HUB, on which is connected a full/low speed device, the host will not wait response from the device, but it can do other HS transactions then after a period of time return to check if HS HUB has received any response from device
  - **Multi-transaction** during one micro-frame (125us) on isochronous transfers using (DATA0, DATA1, DATA2 and MDATA data PIDs)
ULPI High Speed PHY connection
USB High-Speed DMA features overview

- User can enable or disable the DMA mode

- The USB High-Speed DMA is connected to system bus matrix, it can support Burst transfers to/from SRAM or FSMC

- DMA can be enabled with or without FIFO thresholding
  - When FIFO thresholding is enabled a FIFO Rx/Tx levels can be configured to trigger DMA data transfer from RxFIFO to application buffer or from FIFO to USB bus
  - When FIFO thresholding is disabled, the trigger level is fixed to max packet size

- In DMA mode, as part of the transfer configuration parameters, application should program:
  - The application buffer destination address for OUT endpoints/IN channels
  - The application buffer source address for IN endpoints/OUT channels

- When DMA is enabled, a full transfer will be handled by DMA without CPU intervention for copying data to/from FIFOs
  - No more need for interrupts needed for FIFO management (TXFIFO empty interrupt, RxFIFO level interrupt)
Communication Peripherals

ETHERNET MAC 10/100

Same as STM32F-2
Main Features

- Supports 10/100Mbits Half/Full-duplex operations modes
- MII/RMII PHY interface
- Several options for MAC address filtering
- IPv4 checksum offload during receive and transmit operation
- Dedicated DMA controller with two FIFOs (Rx/Tx) of 2KBytes each
  - Connected as AHB master to system bus matrix
- Ethernet Time Stamping support - IEEE1588 version 2
- Power management: Wake on LAN with Magic Packet or Wakeup frame
- MAC management Counters for statistics
- MII loopback mode for debug purpose
**Ethernet Block Diagram**

- **AHB Bus**
  - AHB Slave Interface
  - AHB Master Interface

- **BusMatrix**
  - SRAM 112K
  - SRAM 16K
  - Ext SRA M

- **Ethernet DMA**
  - DMA Control Registers
  - Operation Mode Register
  - 2KB RX FIFO
  - 2KB TX FIFO

- **Media Access Control**
  - MAC 802.3
  - MAC Control Registers
  - Checksum Offload
  - PTP IEEE1588
  - PMT
  - MMC

- **RMII interface**
  - Select
  - MII
  - MDC
  - MDIO

- **External PHY**

**Key Terms**
- MMC: MAC Management Counters
- PMT: Power Management
- PTP: Precision Time Protocol
- RMII: Reduced MII
- MII: Media Independent Interface
Physical Layer Interface

- Supports both Media Independent Interface (MII) and Reduced Media Independent Interface (RMII)
- RMII is a lower pin count alternative, which targets multi-port applications and low cost design
  - **MII** = 16 pins (8 data and 8 control)
  - **RMII** = 7 pins (4 data and 3 control)

### Diagram

**RMII mode**

- STM32F4x7
- 802.3 MAC
- TXD[1:0] → External PHY
- TX_EN → 
- RXD[1:0] → 
- RX_ER → 
- CRS_DV → 
- MDC → 
- MDIO → 
- REF_CLK → PHY_CLK

**MII mode**

- STM32F4x7
- 802.3 MAC
- TX_CLK → External PHY
- TXD[3:0] → 
- TX_EN → 
- TX_ER → 
- RX_CLK → 
- RXD[3:0] → 
- RX_ER → 
- RX_DV → 
- CRS → 
- COL → 
- MDC → 
- MDIO → PHY_CLK
MAC FIFOs

- Two modes for data transfer between FIFOs and SRAM:
  - Threshold mode
  - Store and forward mode

- Threshold mode
  - During frame transmission as soon as the TXFIFO level crosses a defined FIFO level (default is 64 bytes), the data start to be pushed to MAC for frame transmission
  - During frame reception as soon as the RXFIFO level crosses a defined FIFO level (default is 64 bytes), the DMA start to transmit the data to SRAM

- Store and forward mode
  - A full frame should be available in TX or RXFIFO before transfer to MAC or SRAM
CONTROLLER AREA NETWORK (BXCAN)

Communication Peripherals

Same as STM32F-2
CAN Features

- **Dual CAN** 2.0 A, B Active w/ Bit rates up to 1Mbit/s, mapped on APB1
- Support time Triggered Communication
- Three transmit mailboxes w/ configurable transmit priority
- Two receive FIFOs with three stages and 28 filter banks shared between CAN1 and CAN2
- Time Stamp on SOF reception and transmission
- Maskable interrupts for easy software management
- Software efficient mailbox mapping at a unique address space
- 4 dedicated interrupt vectors: transmit interrupt, FIFO0 interrupt, FIFO1 interrupt and status change error interrupt
- The two CAN cells share a dedicated 512-byte SRAM memory and capable to work simultaneously with USB OTG FS peripheral
**CAN1 (Master) with 512 bytes SRAM**

- **Control/Status/Configuration Registers**
  - Master Control
  - Tx Status
  - Interrupt Enable
  - Bit Timing
  - Filter Master
  - Filter Mode
  - Filter Activation
  - Master Status
  - Rx FIFO 0 Status
  - Rx FIFO 1 Status
  - Error Status
  - Filter Scale
  - Filter FIFO Assignment
  - Filter Bank x[27:0]

- **CAN 2.0B Active Core**

- **Master Mailboxes**
  - Mailbox 2
  - Mailbox 1
  - Mailbox 0

- **Transmission Scheduler**

- **Memory Access Controller**

- **Slave Mailboxes**
  - Mailbox 2
  - Mailbox 1
  - Mailbox 0

**CAN2 (Slave)**

- **Control/Status/Configuration Registers**
  - Master Control
  - Tx Status
  - Interrupt Enable
  - Bit Timing
  - Master Status
  - Rx FIFO 0 Status
  - Rx FIFO 1 Status
  - Error Status

- **CAN 2.0B Active Core**

- **Master Mailboxes**
  - Mailbox 2
  - Mailbox 1
  - Mailbox 0

- **Receive FIFO 0**

- **Receive FIFO 1**

**Acceptance Filters**

- Master filter: 0 .. n-1
- Slave filter: n .. 27

- **Transmission Scheduler**

- **Slave Mailboxes**
  - Mailbox 2
  - Mailbox 1
  - Mailbox 0

**Features**

- **CAN1**: Master bxCAN manage the communication between a Slave bxCAN and the 512-byte SRAM memory.
- **CAN2**: Slave bxCAN start filter bank number n[27:1] is configurable by SW.
DIGITAL CAMERA INTERFACE (DCMI)

Same as STM32F-2
The Digital Camera Interface has the following main features:

- 8-, 10-, 12- or 14-bit parallel interface
- Continuous or snapshot mode
- Crop feature
- Supports the following data formats:
  - 8/10/12/14-bit progressive scan: either monochrome or raw bayer
  - YCbCr 4:2:2 progressive scan
  - RGB 565 progressive video
  - Compressed data: JPEG

With a 48MHz PIXCLK and 8-bit parallel input data interface it is possible to receive:

- up to 15fps uncompressed data stream in SXGA resolution (1280x1024) with 16-bit per pixel
- up to 30fps uncompressed data stream in VGA resolution (640x480) with 16-bit per pixel
The data are packed into a 32-bit data register (DCMI_DR) connected to the AHB bus.

8x32-bit FIFO with DMA handling.
DCMI CROP feature

- The DCMI interface supports two types of capture:
  - The DCMI can select a rectangular window from the received image
  - The start coordinates and size are specified using two 32-bit registers DCMI_CWSTRT and DCMI_CWSIZE.
- The size of the window is specified in number of pixel clocks (horizontal dimension) and in number of lines (vertical dimension)
CRYPTOGRAPHIC PROCESSOR (CRYP)

Same as STM32F-2
Definitions

- **AES**: Advanced Encryption Standard
- **DES**: Data Encryption Standard
- **TDES**: Triple Data Encryption Standard

Encryption/Decryption modes

- **ECB**: Electronic code book mode
- **CBC**: Cipher block chaining mode or chained encryption
- **CTR**: Counter mode (used for GCM: Galois Counter Mode)
  
  GCM is a combination of CTR and GHASH.
CRYP algorithms overview

<table>
<thead>
<tr>
<th></th>
<th>AES</th>
<th>DES</th>
<th>TDES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key sizes</td>
<td>128, 192 or 256 bits</td>
<td>64* bits</td>
<td>192***, 128** or 64* bits</td>
</tr>
<tr>
<td></td>
<td></td>
<td>* 8 parity bits</td>
<td>* 8 parity bits: Keying option 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>** 16 parity bits: Keying option 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>***24 parity bits: Keying option 3</td>
</tr>
<tr>
<td>Block sizes</td>
<td>128 bits</td>
<td>64 bits</td>
<td>64 bits</td>
</tr>
<tr>
<td>Time to process one</td>
<td>14 HCLK cycle for key = 128bits</td>
<td>16 HCLK cycle for key = 192bits</td>
<td>16 HCLK cycles</td>
</tr>
<tr>
<td>block</td>
<td>18 HCLK cycle for key = 256bits</td>
<td></td>
<td>48 HCLK cycles</td>
</tr>
<tr>
<td>Type</td>
<td>block cipher</td>
<td>block cipher</td>
<td>block cipher</td>
</tr>
<tr>
<td>Structure</td>
<td>Substitution-permutation</td>
<td>Feistel network</td>
<td>Feistel network</td>
</tr>
<tr>
<td></td>
<td>network</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- AES: Advanced Encryption Standard
- DES: Data Encryption Standard
- TDES: Triple Data Encryption Standard
CRYP Features (1/2)

- Suitable for AES, DES and TDES enciphering and deciphering operations
- **Runs at the same frequency as the CPU**, up to 168 MHz.
- DES/TDES
  - Direct implementation of simple DES algorithms (a single key, K1, is used)
  - Supports the ECB and CBC chaining algorithms
  - Supports 64-, 128- and 192-bit keys (including parity)
  - 64-bit initialization vectors (IV) used in the CBC mode
  - 16 HCLK cycles to process one 64-bit block in DES
  - 48 HCLK cycles to process one 64-bit block in TDES
CRYP Features (2/2)

- AES
  - Supports the ECB, CBC and CTR chaining algorithms
  - Supports 128-, 192- and 256-bit keys
  - 128-bit initialization vectors (IV) used in the CBC and CTR modes
  - 14, 16 or 18 HCLK cycles (depending on the key size) to transform one 128-bit block in AES
- Common to DES/TDES and AES
  - IN and OUT FIFO (each with an 8-word depth, a 32-bit width, corresponding to 4 DES blocks or 2 AES blocks)
  - Automatic data flow control with support of direct memory access (DMA) (using 2 channels, one for incoming data the other for processed data)
  - Data swapping logic to support 1-, 8-, 16- or 32-bit data
CRYP Block Diagram

### Flags

- INIM
- INMIS
- OUTIM
- OUTMIS
- INRIS
- IFEM
- IFNF
- BUSY
- OFFU
- OFNE
- OUTRIS
- DMA request for incoming data transfer
- DMA request for outgoing data transfer

### CRYPTO Processor

- AES
  - ECB
  - CBC
  - CTR
  - Key: 128-, 192- and 256-bit
- TDES
  - ECB
  - CBC
  - Key: 64-, 128- and 192-bit
- DES
  - ECB
  - CBC
  - Key: 64-bit

### Data swapping

- Input FIFO
- Output FIFO

### CRYPTO Global interrupt (NVIC)
ECB Encryption

- The simplest of the encryption modes is the **Electronic codebook** (ECB) mode. The message is divided into blocks and each block is encrypted separately.
- The disadvantage of this method is that identical plaintext blocks are encrypted into identical cipher text blocks; thus, it does not hide data patterns well. To avoid this weakness, CBC or CTR modes can be used.
**Cipher block chaining mode (CBC)**

- CBC mode of operation was invented by IBM in 1976.
- In the CBC mode, each block of plaintext is XORed with the previous cipher text block before being encrypted.
- This way, each cipher text block is dependent on all plaintext blocks processed up to that point.
- To make each message unique, an initialization vector must be used in the first block.
Counter mode (CTR): AES only

- Counter mode turns a block cipher into a stream cipher. It generates the next key stream block by encrypting successive values of a "counter".
- The counter can be any function which produces a sequence which is guaranteed not to repeat for a long time, although an actual counter is the simplest and most popular.
- CTR mode is well suited to operation on a multi-processor machine where blocks can be encrypted in parallel.
- The IV/nonce and the counter can be concatenated, added, or XORed together to produce the actual unique counter block for encryption.

**Diagram:**

- **Encryption**
  - IV = Counter value
  - Cipher Text1
  - Cipher Text2

- **Decryption**
  - IV = Counter value
  - Plain Text1
  - Plain Text2
## CRYP throughput

- Throughput in MB/s at 168 MHz for the various algorithms and implementations

<table>
<thead>
<tr>
<th></th>
<th>AES-128</th>
<th>AES-192</th>
<th>AES-256</th>
<th>DES</th>
<th>TDES</th>
</tr>
</thead>
<tbody>
<tr>
<td>HW Theoretical</td>
<td>192.00</td>
<td>168.00</td>
<td>149.33</td>
<td>84.00</td>
<td>28.00</td>
</tr>
<tr>
<td>HW Without DMA</td>
<td>72.64</td>
<td>72.64</td>
<td>62.51</td>
<td>43.35</td>
<td>16.00</td>
</tr>
<tr>
<td>HW With DMA</td>
<td>168.00</td>
<td>168.00</td>
<td>149.33</td>
<td>84.00</td>
<td>28.00</td>
</tr>
<tr>
<td>Pure SW</td>
<td>1.38</td>
<td>1.14</td>
<td>0.96</td>
<td>0.74</td>
<td>0.25</td>
</tr>
</tbody>
</table>
RANDOM NUMBER GENERATOR (RNG)

Same as STM32F-2
RNG Features

- 32-bit random numbers, produced by an analog generator (based on a continuous analog noise)
- **Clocked by a dedicated clock** (PLL48CLK)
- **40 periods of the PLL48CLK** clock signal between two consecutive random numbers
- Can be disabled to reduce power-consumption
- Provide a success ratio of more than **85% to FIPS 140-2** (Federal Information Processing Standards Publication 140-2) tests for a sequence of 20 000 bits.
- 5 Flags
  - 1 flag occurs when Valid random Data is ready
  - 2 Flags to an abnormal sequence occurs on the seed.
  - 2 flags for frequency error (PLL48CLK clock is too low).
- 1 interrupt
  - To indicate an error (an abnormal sequence error or a frequency error)
RNG Block Diagram

RNG

RNG_CLK

32bit random data register

LFSR (Linear Feedback Shift register)

Analog Seed

Error management

Clock checker

Fault detector

Interrupt enable bit IM

DRDY SECS SEIS CECS CEIS

Flags

RNG interrupt to NVIC

STMicroelectronics
HASH PROCESSOR (HASH)

Same as STM32F-2
A cryptographic hash function is a deterministic procedure that takes an arbitrary block of data and returns a fixed-size bit string, the (cryptographic) hash value, such that an accidental or intentional change to the data will change the hash value. The data to be encoded is often called the "message", and the hash value is sometimes called the message digest or simply digest.
Definitions

- **SHA-1**: the Secure Hash algorithm
- **MD5**: Message-Digest algorithm 5 hash algorithm
- **HMAC**: (keyed-Hash Message Authentication Code) algorithm
- **HASH**: Computes a SHA-1 and MD5 message digest for messages of up to \((2^{64} - 1)\) bits

HMAC algorithms provide a way of authenticating messages by means of hash functions.

HMAC algorithms consist in calling the SHA-1 or MD5 hash function twice on message in combination with a secret value (key).
HASH Features

- Suitable for Integrity check and data authentication applications, compliant with:
  - FIPS PUB 180-2 (Federal Information Processing Standards Publication 180-2)
  - Secure Hash Standard specifications (SHA-1)
  - IETF RFC 1321 (Internet Engineering Task Force Request For Comments number 1321) specifications (MD5)

- AHB slave peripheral
- Fast computation of SHA-1 and MD5:
  - 66 HCLK clock cycles in SHA-1
  - 50 HCLK clock cycles in MD5

- 5 × (32-bit) words (H0, H1, H2, H3 and H4) for output message digest, reload able to continue interrupted message digest computation
- Automatic data flow control with support for direct memory access (DMA)

- 32-bit data words for input data, supporting word, half-word, byte and bit bit-string representations, with little-endian data representation only
HASH Block Diagram

- Input FIFO: 16 x 32bit
- Data swapping
- HASH Processor
  - MD5
  - SHA-1
  - HMAC
- Message Digest: H0..H4, 5x32bit
- Flags: DINIS, BUSY, DMAS, DCIS
- DINIM
- DCIM
- HASH Global interrupt (NVIC)
- DMA request
### HASH throughput

- Throughput in MB/s at 168 MHz for SHA-1 and MD5 algorithms with different implementations

<table>
<thead>
<tr>
<th></th>
<th>MD5</th>
<th>SHA1</th>
</tr>
</thead>
<tbody>
<tr>
<td>HW Theoretical</td>
<td>162.9</td>
<td>131.12</td>
</tr>
<tr>
<td>HW Without DMA</td>
<td>77.35</td>
<td>71.68</td>
</tr>
<tr>
<td>HW With DMA</td>
<td>105.40</td>
<td>91.11</td>
</tr>
<tr>
<td>Pure SW</td>
<td>11.52</td>
<td>5.15</td>
</tr>
</tbody>
</table>
Thank you

STM32 Releasing your creativity

STM32 F4

www.st.com/stm32f4