STM32F3 Analogue Peripherals

Microcontroller Division

Modified by Tomas DRESLER

life.augmented

STM32F3 vs. STM32F1 FW compatibility(1/3)

Peripheral	F1 series	F37x family	F30x family	Comment	FW compatibility
ADC	YES	YES	YES++	New design	Full for F37x Not compatible for F30x
COMP	NA	YES	YES	NA	NA
DAC	YES	YES	YES	Same feature	Full for F30x Not compatible for F37x
OPAMP	**NA**	**NA**	**YES**	**NA**	**NA**
SDADC	NA	YES++	NA	New peripheral	NA

Touch Sensing Controller (TSC)

life.augmented

TSC Features (1/2)

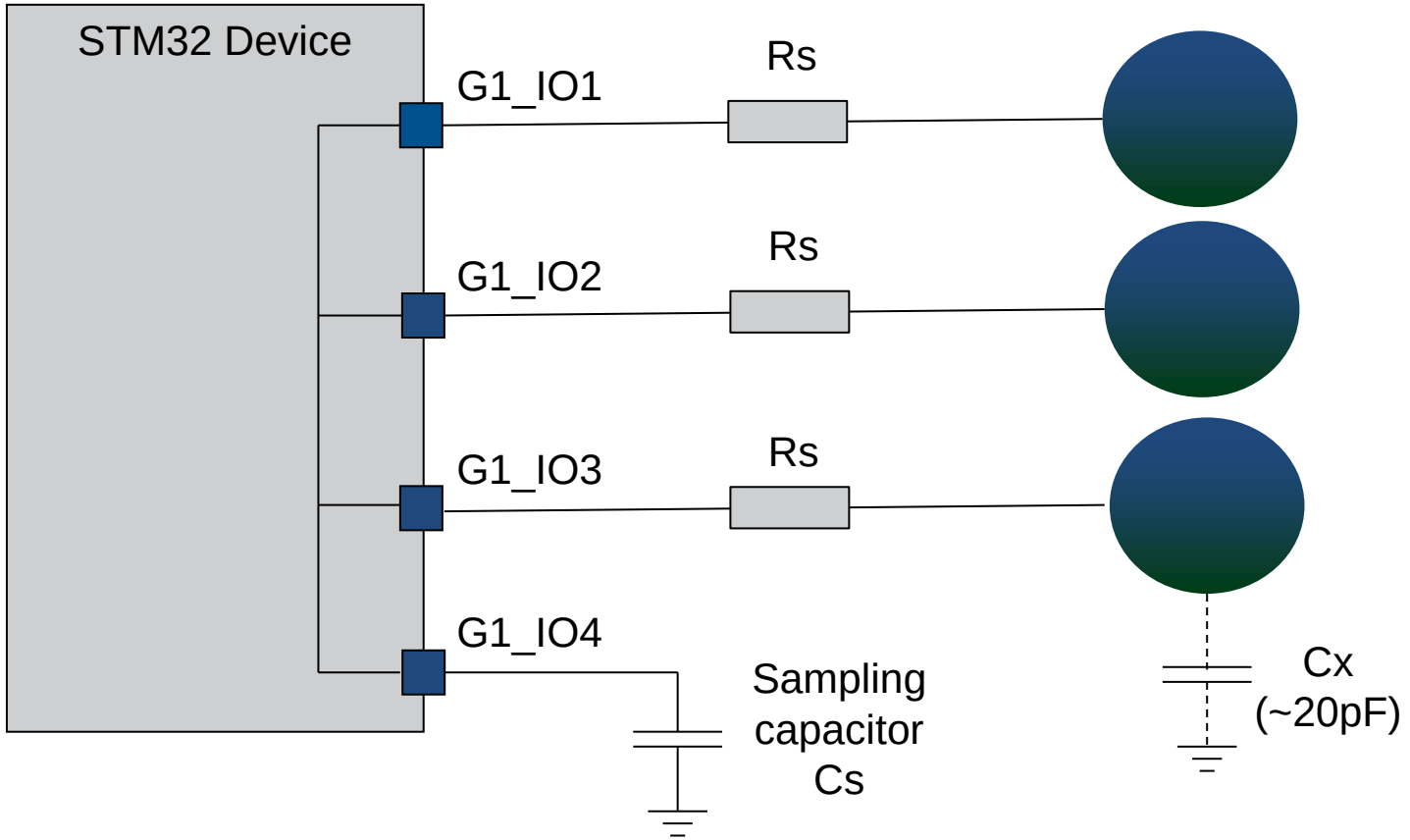
- Proven and robust surface charge transfer acquisition principle available on **STM32F05x**, **STM32F30x** and **STM32F37x** families
- Supports up to **24** capacitive sensing channels split over **8** analog I/O groups
  - Number of channels and analog I/O groups depend on the device used
- Up to **8** capacitive sensing channels can be acquired in parallel offering a very good response time
  - 1 counter per analog I/O group to store the current acquisition result
- One sampling capacitor for up to 3 capacitive sensing channels to reduce the system components
- Full hardware management of the charge transfer acquisition sequence
  - No CPU load during acquisition
- Spread spectrum feature to improve system robustness in noisy environment

TSC Features (2/2)

- Programmable charge transfer frequency
- Programmable sampling capacitor I/O pin
  - Any GPIO of an analog IO group can be used for the sampling capacitor
- Programmable channel I/O pin
  - Any GPIO of an analog IO group can be used for the channel
- Programmable max count value to avoid long acquisition when a channel is faulty
- Dedicated end of acquisition and max count error flags with interrupt capability
- Compatible with proximity, touchkey, linear and rotary touch sensors
- Designed to operate with STMTouch touch sensing firmware library

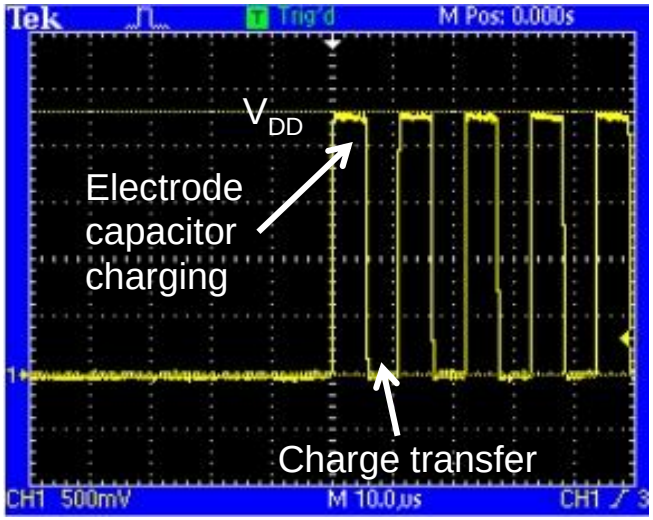
Charge Transfer Measuring Circuit

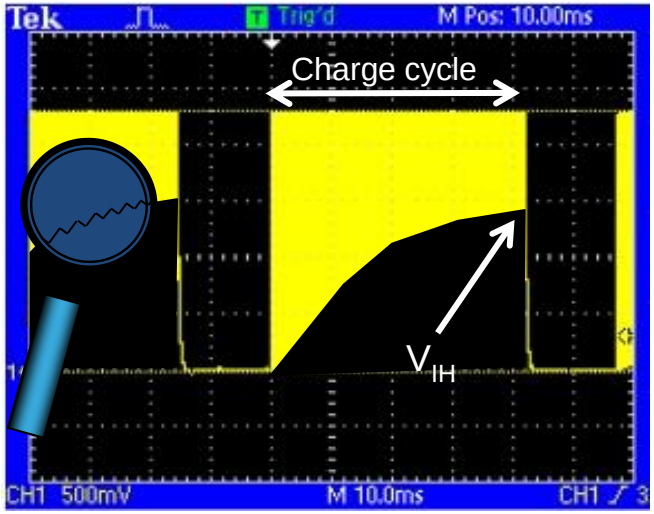
- Rs is used to improve ESD robustness (typically 10K)
- Cs sampling capacitor value depends on the required channels sensitivity
  - The higher the Cs value, the higher the sensitivity but longer acquisition time



Charge Transfer Acquisition Overview

- Charge transfer uses the electrical properties of the capacitor charge Q
- It uses a **sampling capacitor (C_S)** in which the **electrode (C_X)** charges are transferred to
- Charge Transfer is performed through analog switches directly embedded into the GPIO
- The charge transfer cycle is repeated N times until the voltage on the sampling capacitor reaches the **V_{IH} threshold** of the GPIO it is connected to
- The number N of transfer cycles required to reach the threshold represents the size of Cx
  - The number of transfer decreases when the electrode is touched.





life.augmented

STMTouch Touch Sensing Library

- Complete free C source code library with firmware examples
- Multifunction capability to combine capacitive sensing functions with traditional MCU features
- Enhanced processing features for optimized sensitivity and immunity
  - Calibration, environment control system (ECS), debounce filtering , detection exclusion system (DxS), …
- Complete and simple API for status reporting and application configuration
- Touchkey, proximity, linear and rotary touch sensors support
- Compliant with MISRA
- Compliant with all STM32 C compilers
- STM32F051 support planned for end Q2 2012, F3 easily adapted

STM32F30x Specific features/peripherals

life.augmented

Analog-to-digital converter (ADC) 5MSPS

life.augmented

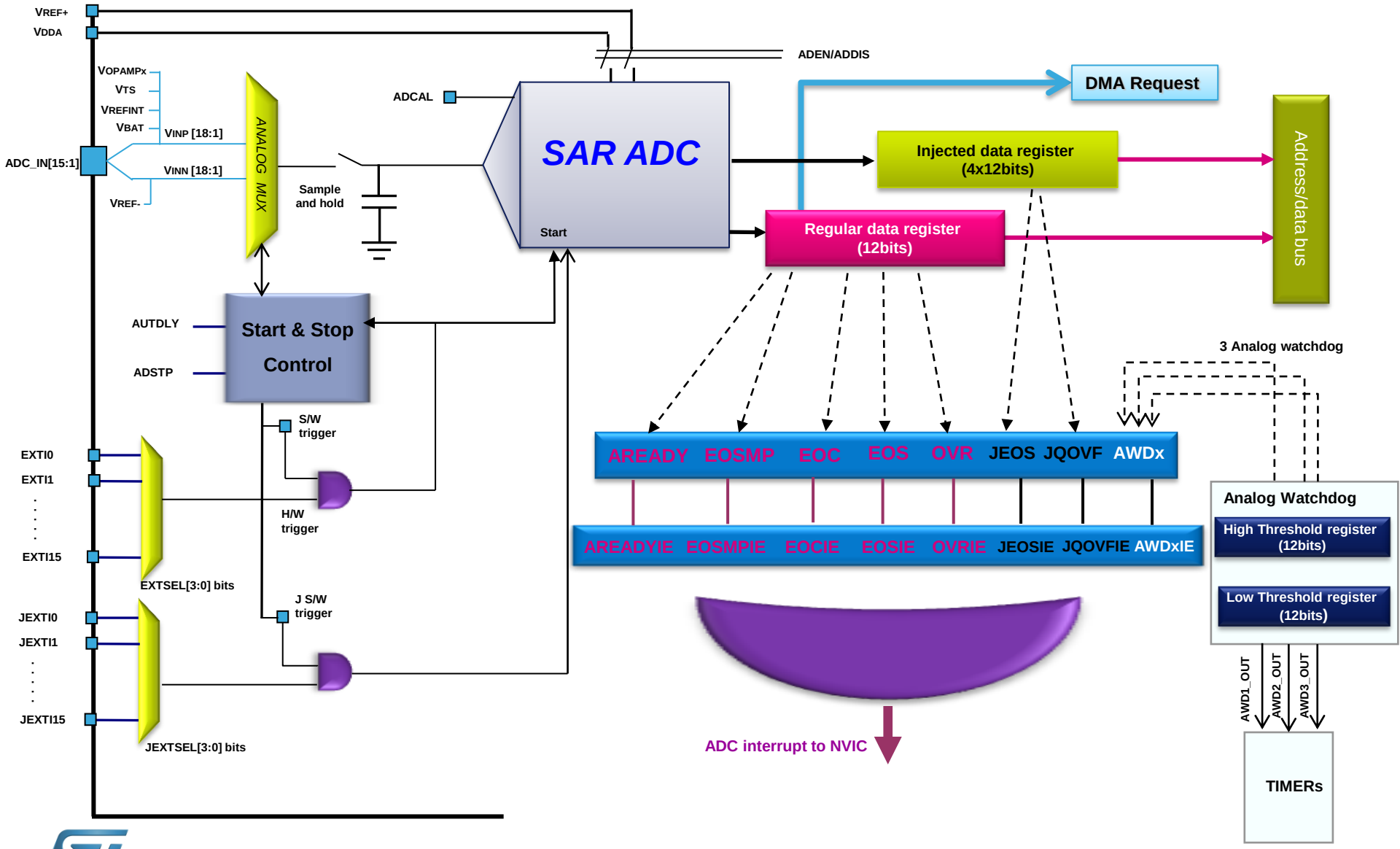
ADC Features (1/2)

- Up to 4 ADCs:
  - ADC1 & ADC2 are tightly coupled and can operate in dual mode (ADC1 is master)
  - ADC3 & ADC4 are tightly coupled and can operate in dual mode (ADC3 is master)
- Programmable Conversion resolution : 12, 10, 8 or 6 bit
- External Analog Input Channels for each of the 4 ADCs:
  - 5 fast channels from dedicated GPIOs pads
  - Up to 11 slow channels from dedicated GPIOs pads
- ADC conversion time:
  - Fast channels : up to 5.1Ms/s with 12 bit resolution in single mode
  - Slow channels: up to 4,8Ms/s with 12 bit resolution in single mode
- AHB Slave Bus interface
- Channel-wise programmable sampling time
- Self-calibration
- Configurable regular and injected channels
- Hardware assistant to prepare the context of the injected channels to allow fast context switching
- Can manage Single-ended or differential inputs

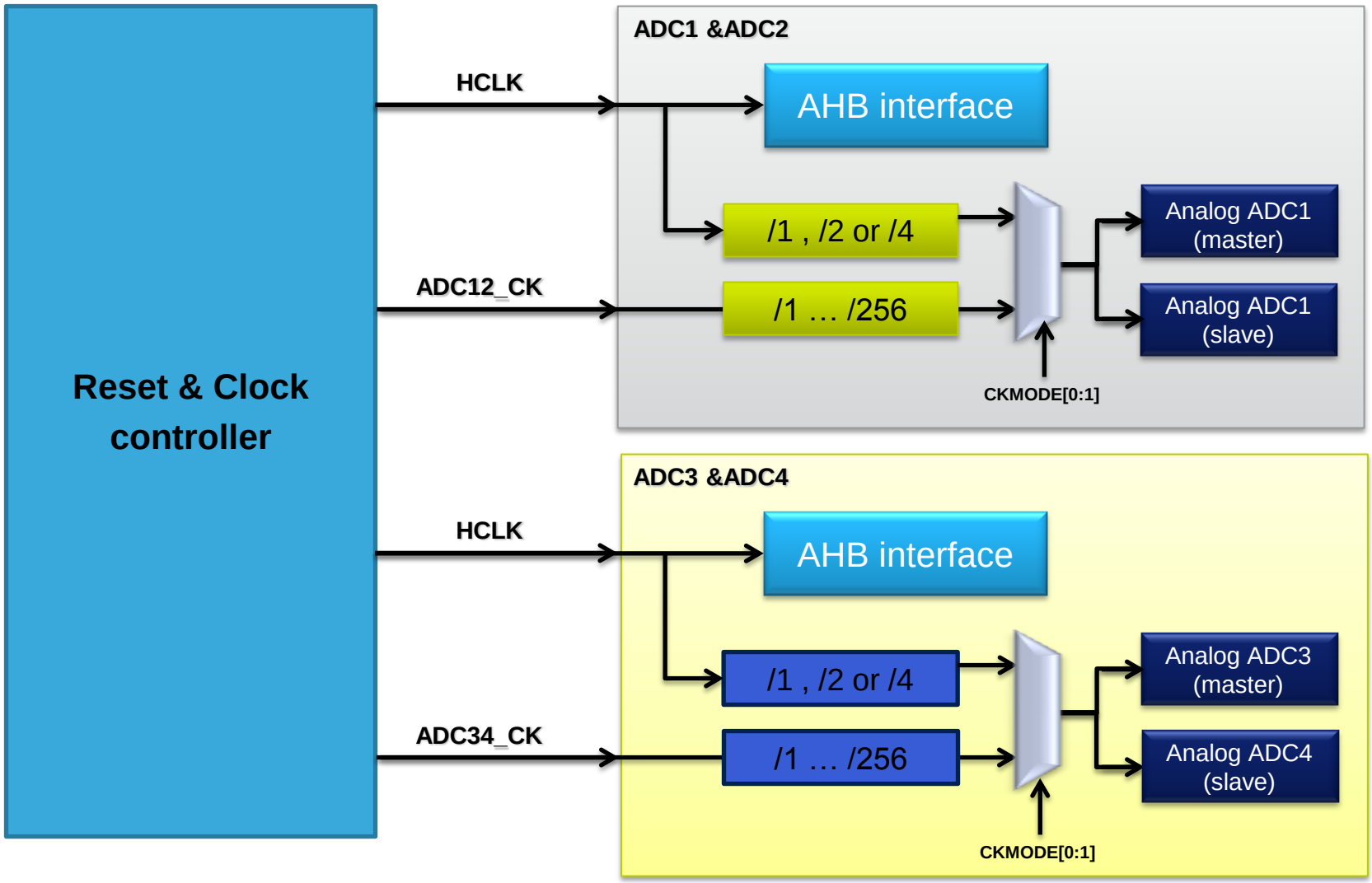
ADC Features (2/2)

- 3 internal channels connected to :
  - Temperature sensor Vsense connected to ADC1
  - Internal voltage reference VREFINT connected to all ADCs
  - VBAT/2 power supply connected to ADC1
- Programmable sampling time
- Single, continuous and discontinuous conversion modes
- Dual ADC mode
- Left or right Data alignment with inbuilt data coherency
- Software or Hardware start of conversion
- 3 Analog Watchdog per ADC
- DMA capability
- Auto Delay insertion between conversions
- Interrupt generation

ADC Block Diagram



ADC Clocks

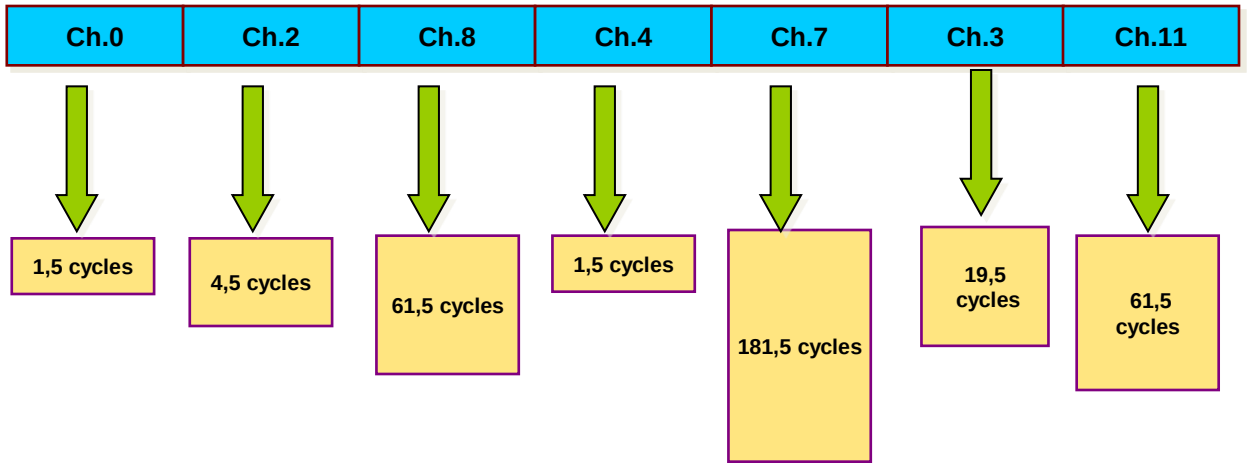

Reset & Clock controller
HCLK
ADC12_CK
ADC1 &ADC2
AHB interface
/1 , /2 or /4
/1 … /256
Analog ADC1 (master)
Analog ADC1 (slave)
CKMODE[0:1]
HCLK
ADC34_CK
ADC3 &ADC4
AHB interface
/1 , /2 or /4
/1 … /256
Analog ADC3 (master)
Analog ADC4 (slave)
CKMODE[0:1]

ADC Channel selection

- Up to 16 regular and 4 injected conversions with programmable order and programmable sampling time,

Example: - Conversion of channels: 0, 2, 8, 4, 7, 3 and 11

- Different sampling time.



Total Conversion Time

- Total conversion Time = $T_{Sampling}$ + $T_{Conversion}$

Resolution	$T_{Conversion}$
12 bits	12,5 Cycles
10 bits	10,5 Cycles
8 bits	8,5 Cycles
6 bits	6,5 Cycles

Resolution	Total conversion Time (When FADC = 72MHz)	
12 bits	12,5 + 1,5 = 14cycles	19.4 us → 5,1 Msps
10 bits	10,5 + 1,5 = 12 cycles	16,6 us → 6 Msps
8 bits	8,5 + 1,5 = 10 cycles	13,8 us → 7,2 Msps
6 bits	6,5 + 1,5 = 8 cycles	11,1 us → 9 Msps

Single-ended & Differential input channels

- Channels can be configured to be either single-ended or differential input by writing ADC_DIFSEL register:
  - In single ended input mode, the analog voltage to be converted for channel “i” is the difference between the external voltage ADC_INi (positive input) and VREF- (negative input)
  - In differential input mode, the analog voltage to be converted for channel “i” is the difference between the external voltage ADC_INi (positive input) and ADC_Ini+1 (negative input)

Note 1: When configuring the channel “i” in differential input mode, channel “i+1” is no longer usable in single-ended mode or in differential mode and must never be configured to be converted.

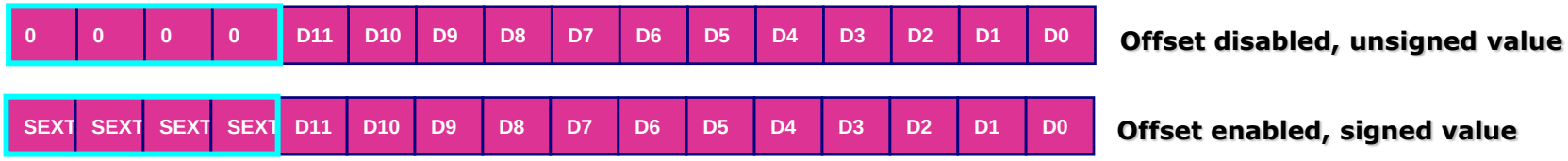
What is the queue of context

- It is a hardware assistant to prepare the context of the injected channels to allow fast context switching
- A queue of context is implemented to anticipate up to 2 contexts for the next injected sequences of conversions,
- The context consist of:
  - Configuration of the injected triggers (JEXTEN[1:0] and JEXTSEL[3:0]),
  - Definition of the injected sequence (JSQx[4:0] and JL[1:0]),
- Context parameters are defined in ADC_JSQR register which implements a queue of 2 buffers,

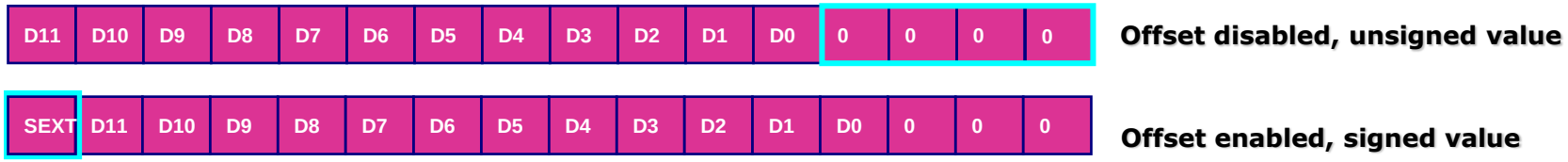
ADC Channel offset

- An offset x (x=1,2,3,4) can be applied to a channel by setting the OFFSETx_EN of ADC_OFRx register.
- The channel to which the offset will be applied is programmed into the bits OFFSETx_CH of ADC_OFRx register.
- In this case, the converted value is decreased by the user-defined offset written in the OFFSETx bits.
- The result may be a negative value so the read data is signed and the SEXT bit represents the extended sign value.

Right alignment

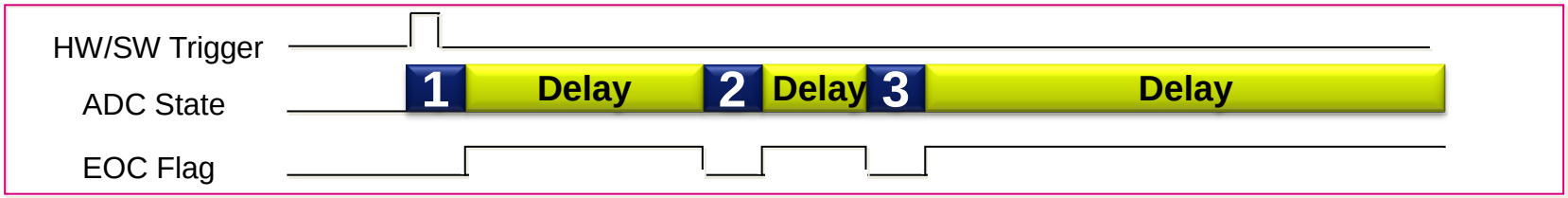


Left alignment

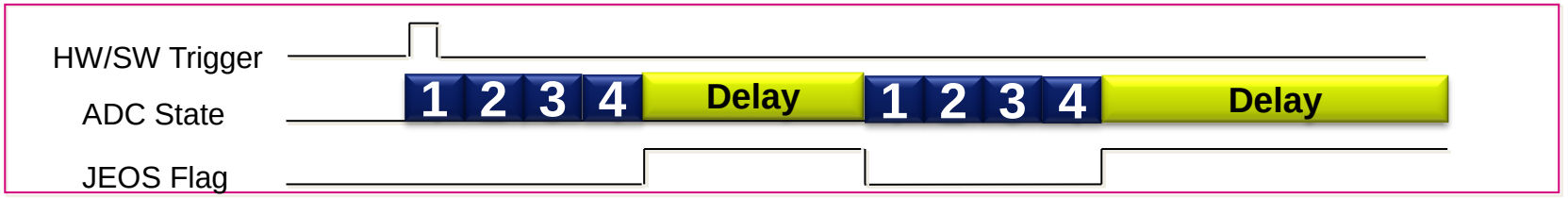


Auto delayed conversion (1/2)

- **Auto Delay Mode:** when AUTDLYbits = 1, a new conversion can start only if the previous data has been treated:
  - For regular conversions: once the ADC_DR register has been read or if the EOC bit has been cleared.



  - For injected conversions: when the JEOS bit has been cleared,



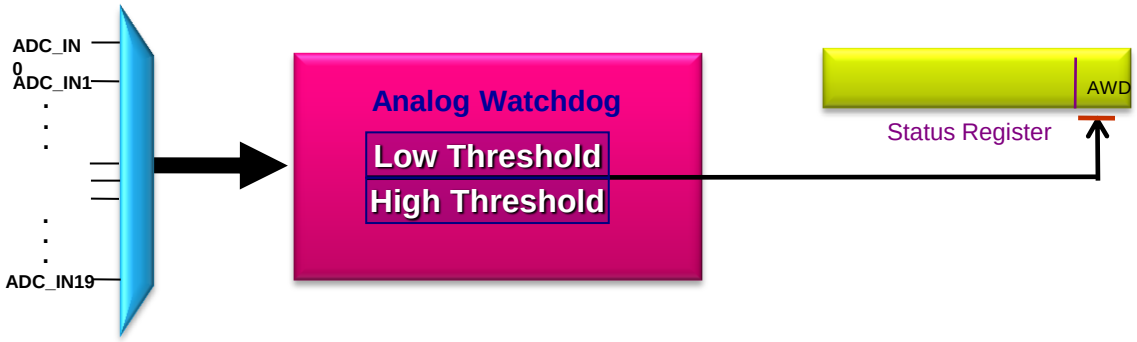
i Regular channel conversion

Note : A trigger event (**for the same group of conversions**) occurring during an already ongoing sequence or during this delay is ignored.

➔ This is a way to automatically adapt the speed of the ADC to the speed of the system that reads the data.

ADC Analog Watchdogs

- ADC Analog Watchdog 1
  - 12-bit programmable analog watchdog low and high thresholds
  - Enabled on one or all converted channels
  - Interrupt generation on low or high thresholds detection
- ADC Analog Watchdog 2&3
  - Enabled on some selected channels by programming bits in AWDCHx[19:0],
  - Resolution Limited to 8 bits and only the 8 MSBs of the thresholds can be programmed into HTx[7:0] and LTx[7:0]

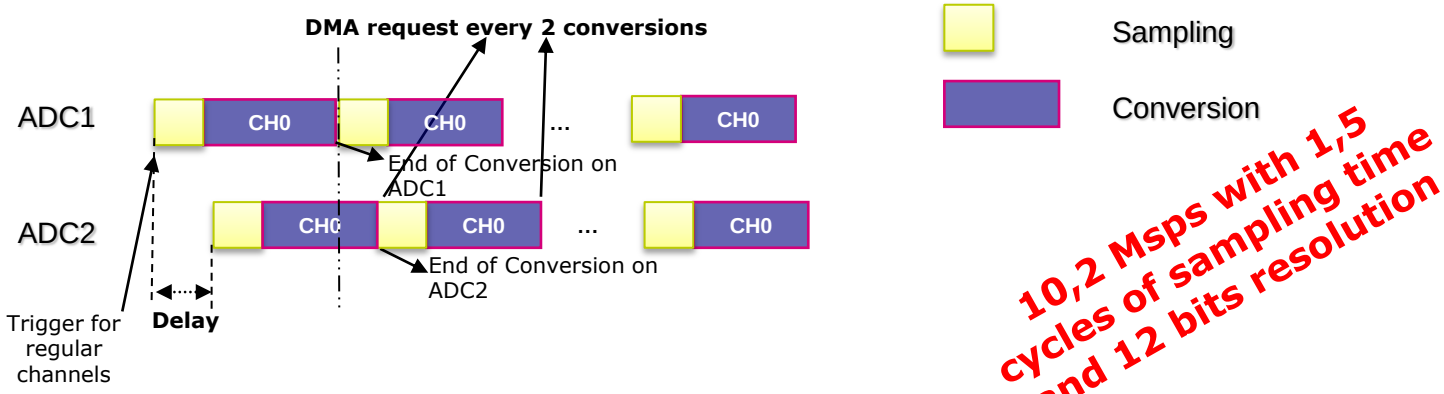


Note: *The watchdog comparison is performed on the raw converted data before any alignment calculation and before applying any offsets.*

Interleaved mode

- Converts a regular channel group (usually one channel).
- The external trigger source, which start the conversion, comes from ADC1:
  - ADC1 starts immediately,
  - ADC2 starts after a configurable delay,
- An EOC is generated at the end of each channel conversion,
- Results stored on the common data register ADC_CDR and on the each ADCx_DR,

18 Msps with 1,5 cycles of sampling time and 6 bits resolution

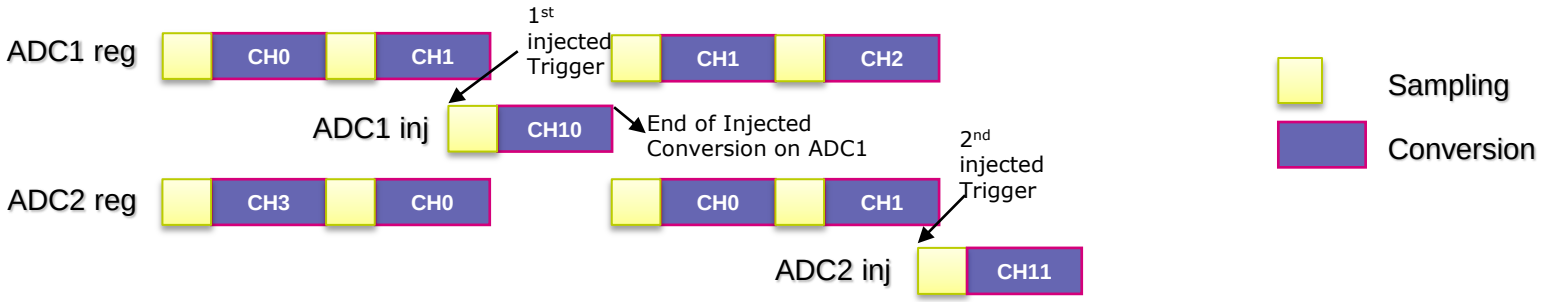


10,2 Msps with 1,5 cycles of sampling time and 12 bits resolution

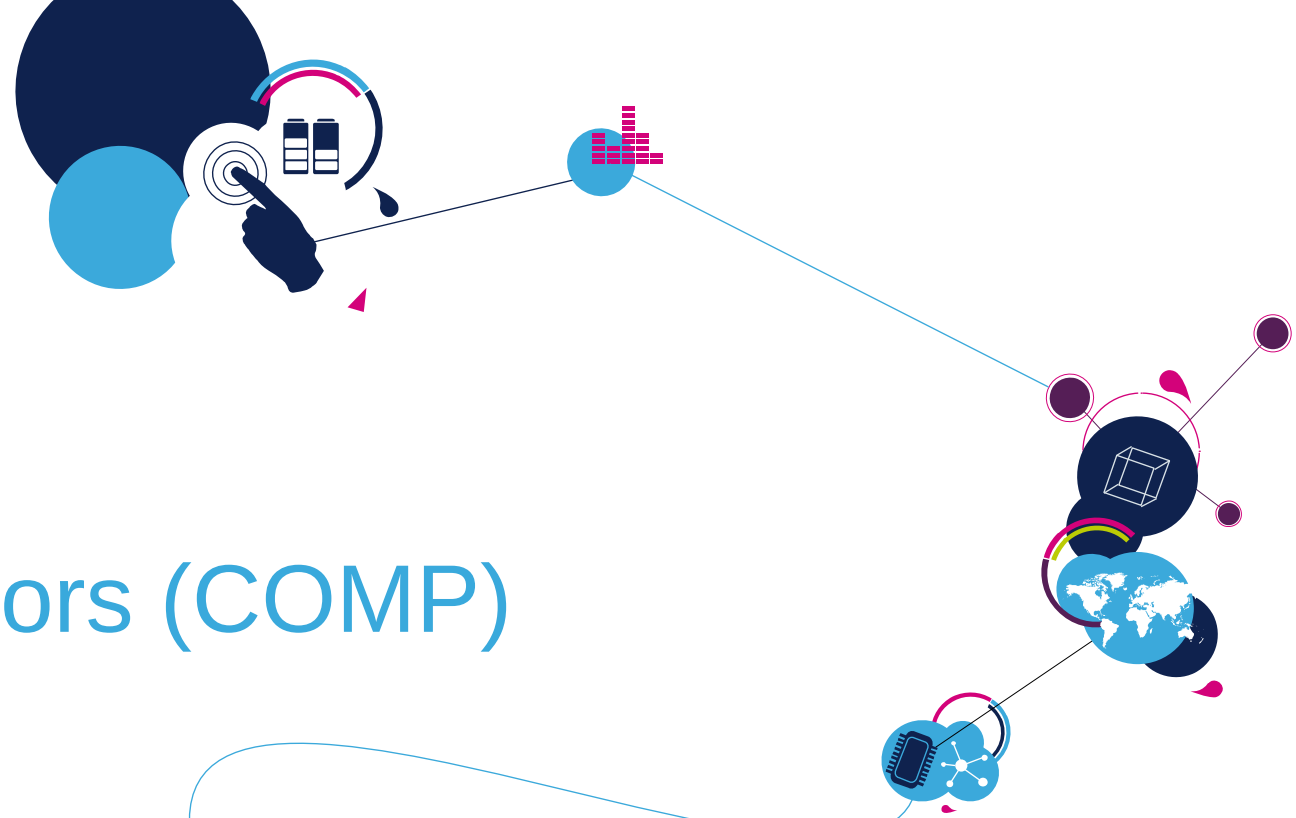
- This mode can not be combined with auto-delayed mode,
- Once SW set ADSTART or ADSTP bits of the master ADC, the corresponding bits of the slave ADC are also automatically set,

Regular simultaneous + Alternate trigger

- Converts an injected and regular channel groups,
- The external trigger source comes from the master ADC,
- Results of injected channels stored on ADCx_JDRy registers, and regular channels on each ADCx_DR register and on ADC_CDR register.



- This mode can not be combined with auto-delayed mode,



Comparators (COMP)

life.augmented

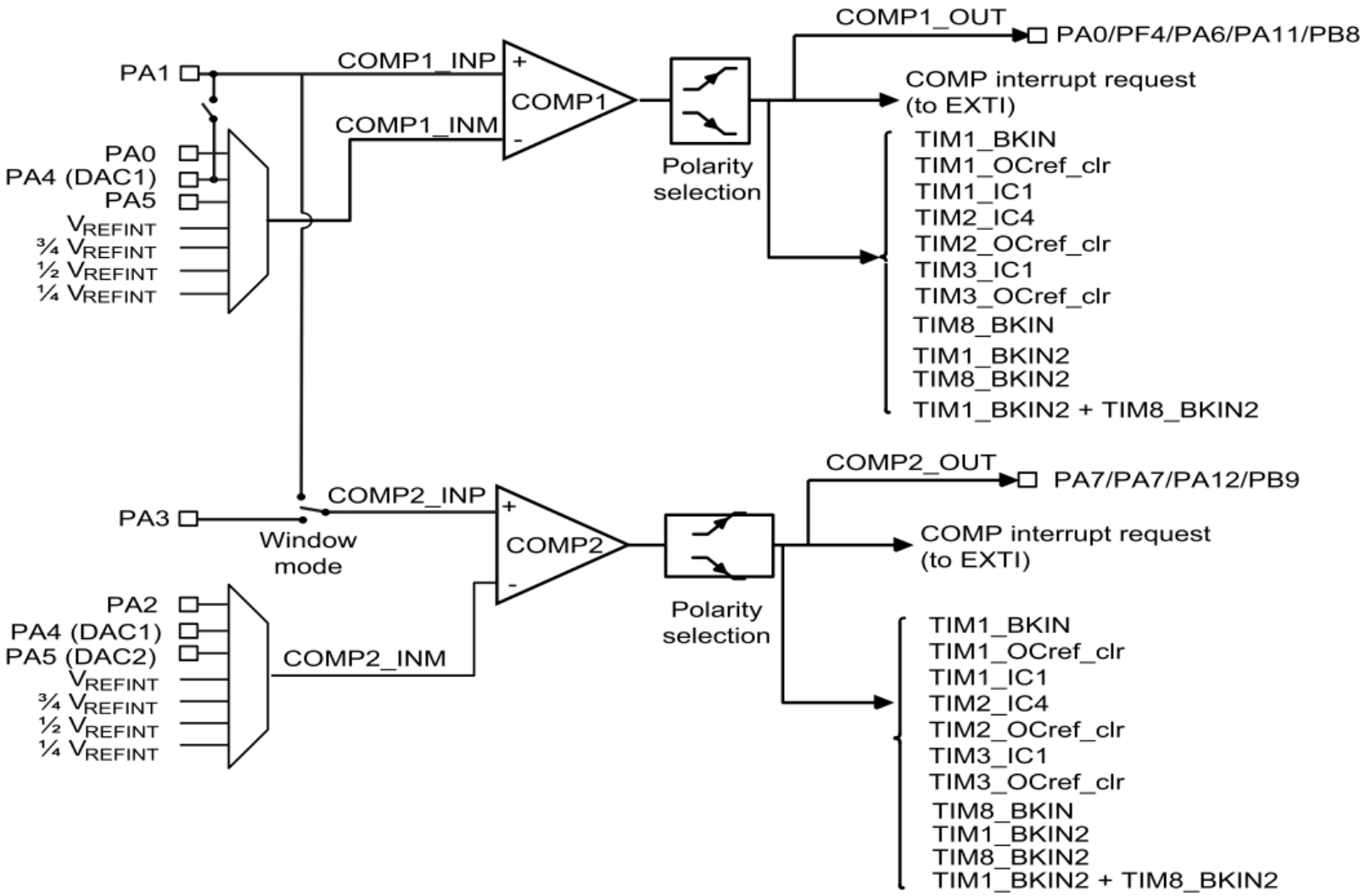
COMP features (1/2)

- 7 comparator pairs COMPx, x = 1..7
  - Rail-to-rail inputs
  - Programmable speed / consumption: 4 modes
  - Programmable hysteresis: 4 levels
  - Inputs and outputs available externally - can be used as a standalone device without MCU interaction
  - Comparator pairs can be combined into a window comparators
  - Multiple choices for output redirection
  - Comparator blanking – The blanking time period is defined by TIM OC – multiple timer OC events available
    - to avoid reaction of the regulation loop on the current spikes at the beginning of the PWM period caused by the recovery current in power switches
- Can be used for:
  - Exiting low power modes
  - Signal conditioning
  - Cycle-by-cycle current control with blanking (w/ DAC and TIM)

COMP features (2/2)

- Comparator characteristics at a glance
  - Full operating voltage range 2V < VDDA < 3.6V
  - Propagation time vs consumption
    High speed / full power
    - Medium speed / medium power
    - Low speed / Low power
    - Very low speed / Ultra-low power
    - Input offset: +/-4mV typ, +/- 20mV max
  - Programmable hysteresis: 0, 8, 15, 31 mV
- Fully asynchronous operation
  - Comparators working in STOP mode
  - No clock related propagation delay
- Functional safety (Class B)
  - The comparator configuration can be locked with a write-once bit

Block diagram for STM32F30x

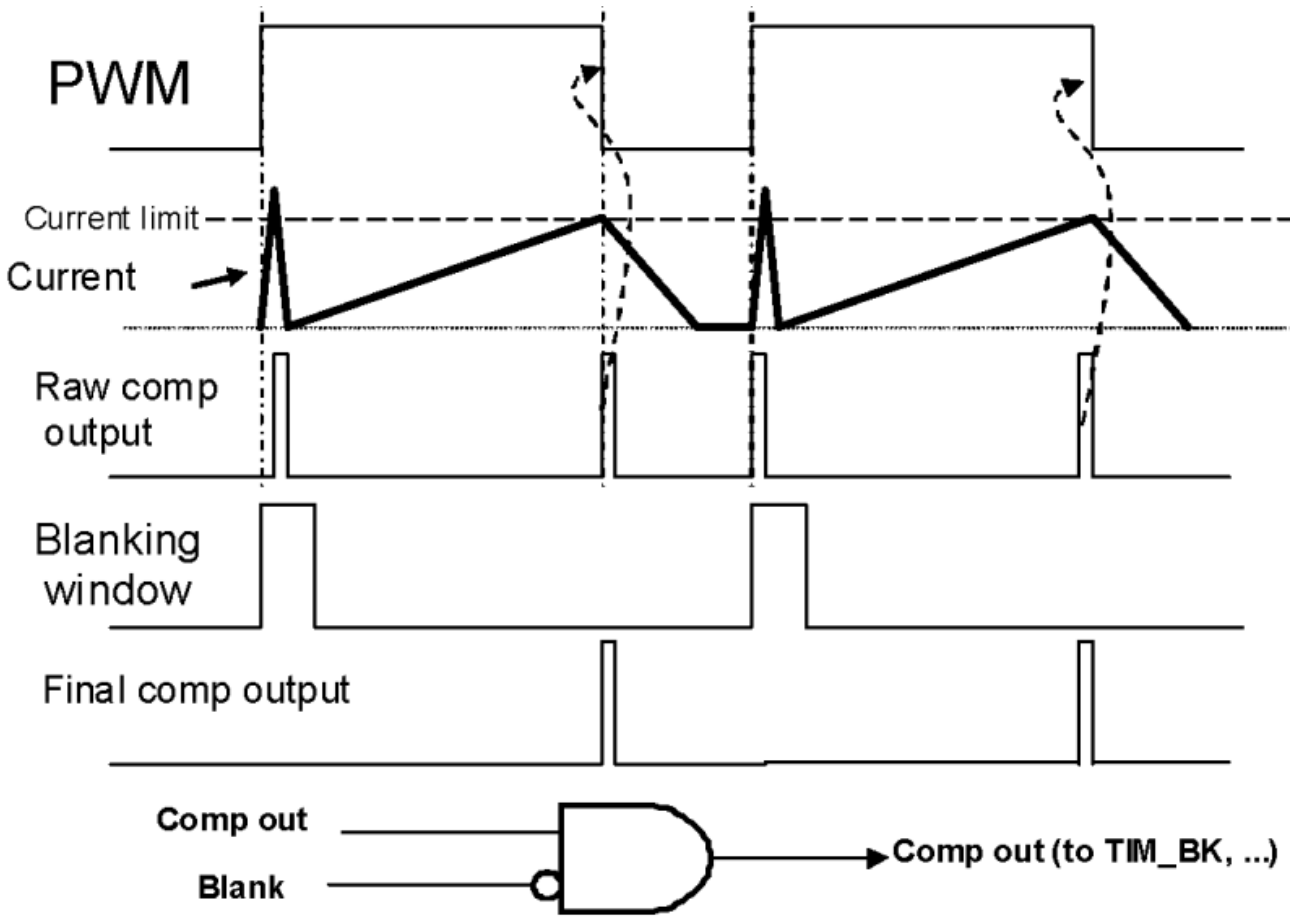


BKIN: PWM's "Emergency stop" input
OCRefClear: PWM clear for cycle-by-cycle current controller

life.augmented

Blanking function

- Purpose: prevent the current regulation to trip upon short current spikes at the beginning of the PWM period (typically the recovery current in power switches anti parallel diodes).



Operational Amplifier

life.augmented

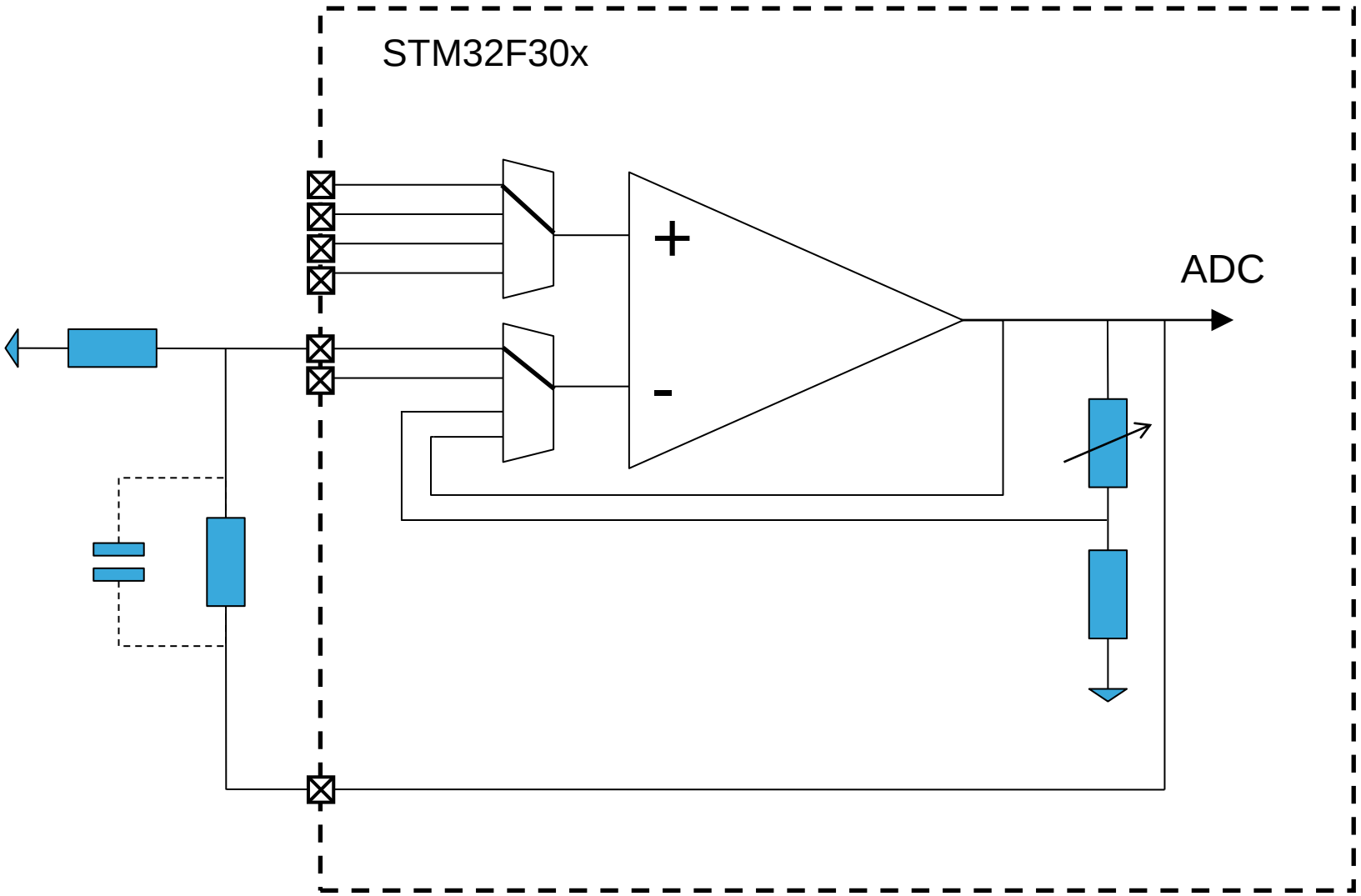
Features (1/2)

- Up to 4 operational amplifiers
- Rail to Rail input/output
- Low Offset voltage
- Access to all terminals
- Input multiplexer on inverting and non inverting inputs
- Input multiplexer can be triggered by a timer and synchronized with a PWM signal.
- 4 operating modes:
  - Standalone mode: External gain setting
  - Follower mode
  - PGA mode: internal gain setting (x2, x4, x8, x16)
  - PGA mode: internal gain setting (x2, x4, x8, x16) with inverting input used for filtering.

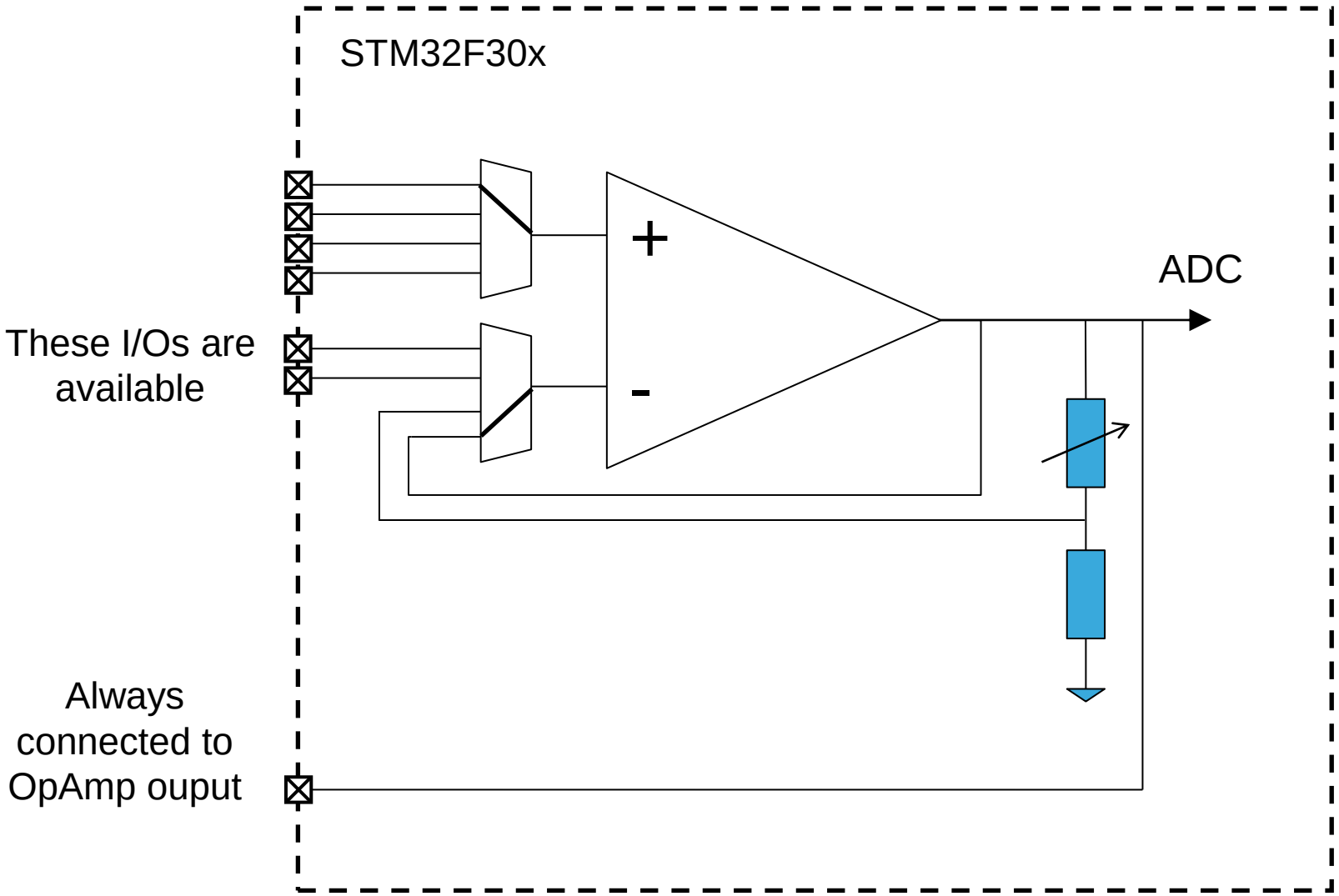
Features (2/2)

- Operating conditions
  - 2.4V < VDDA < 3.6V
  - -40 C < Temp < 105 C
- Input stage
  - Input: rail to rail
  - Offset: 10mV max
  - Ibias < +/-1µA max (mostly I/O leakage)
- Output stage
  - Output: rail to rail
  - Iload < 500µA (sink and source)
  - Capacitive load < 50pF (stable when connected internally on ADC input)
  - GNDA + 100mV < Vout < VDDA – 100mV (Max)
- Speed
  - GBW: 8MHz
  - Slew rate 4.5V/µs
  - unity gain stable

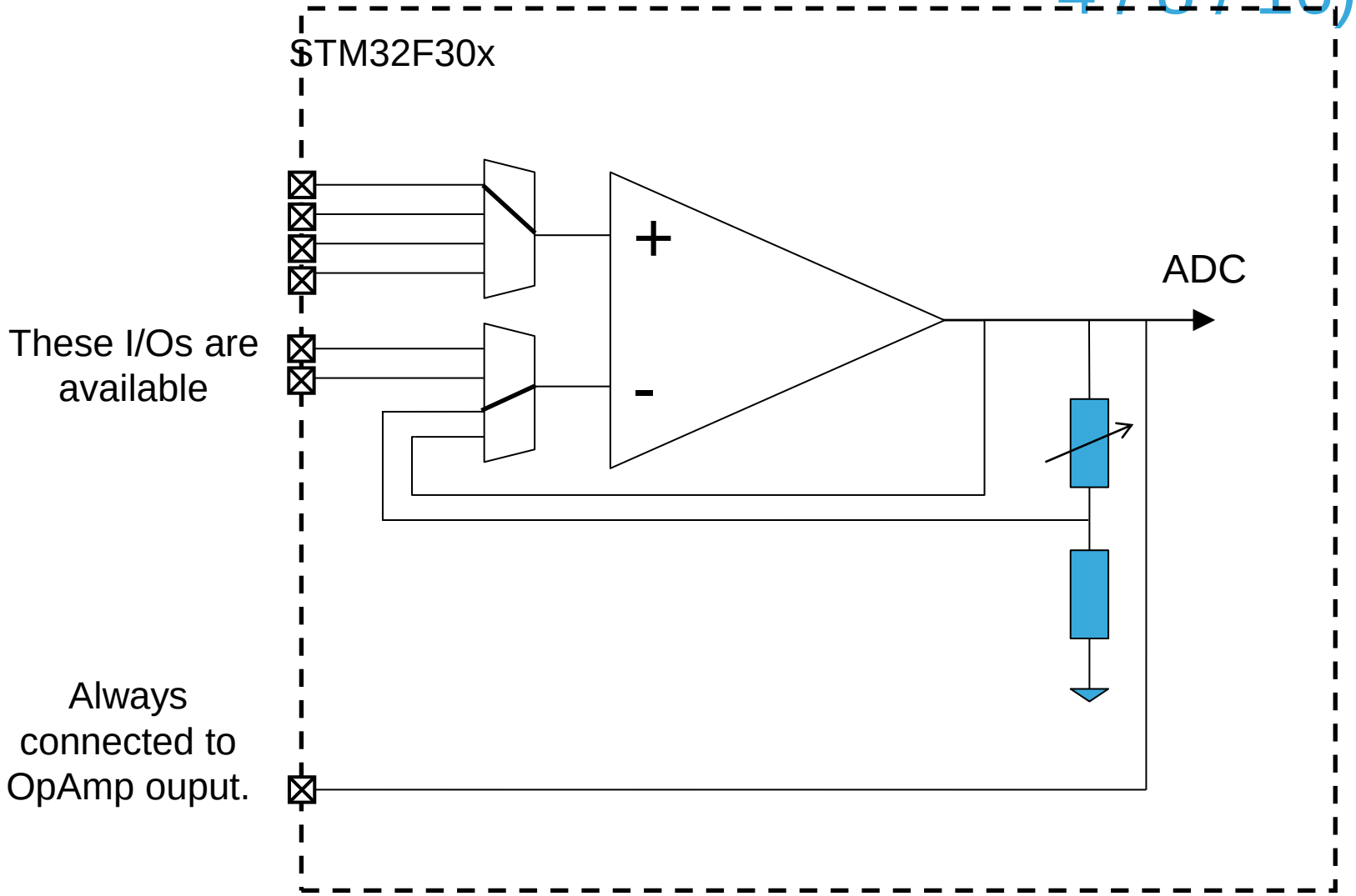
Standalone mode, External Gain setting



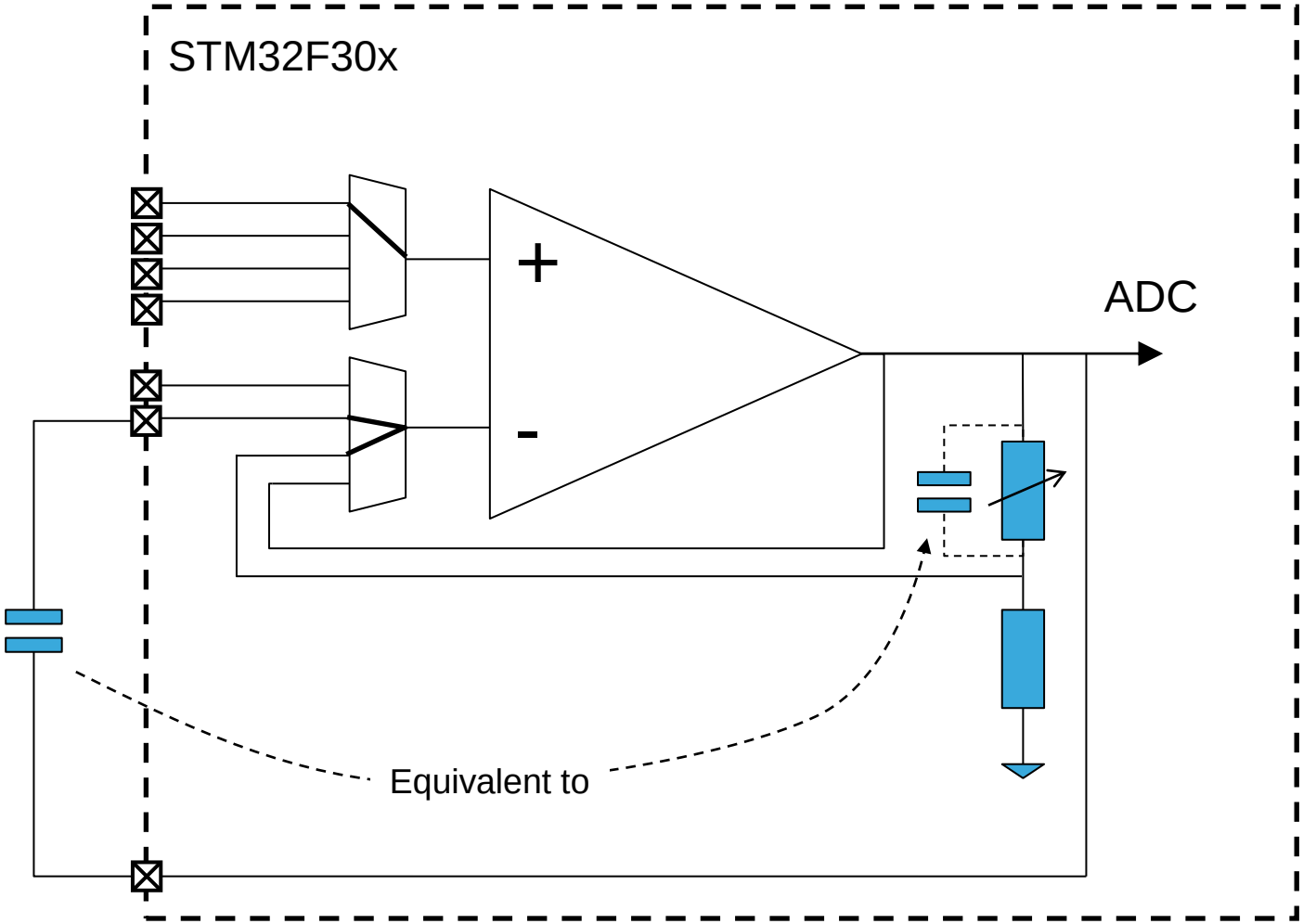
Follower mode



PGA Mode, Internal Gain setting (Gain = 2 / 4 / 8 / 16)



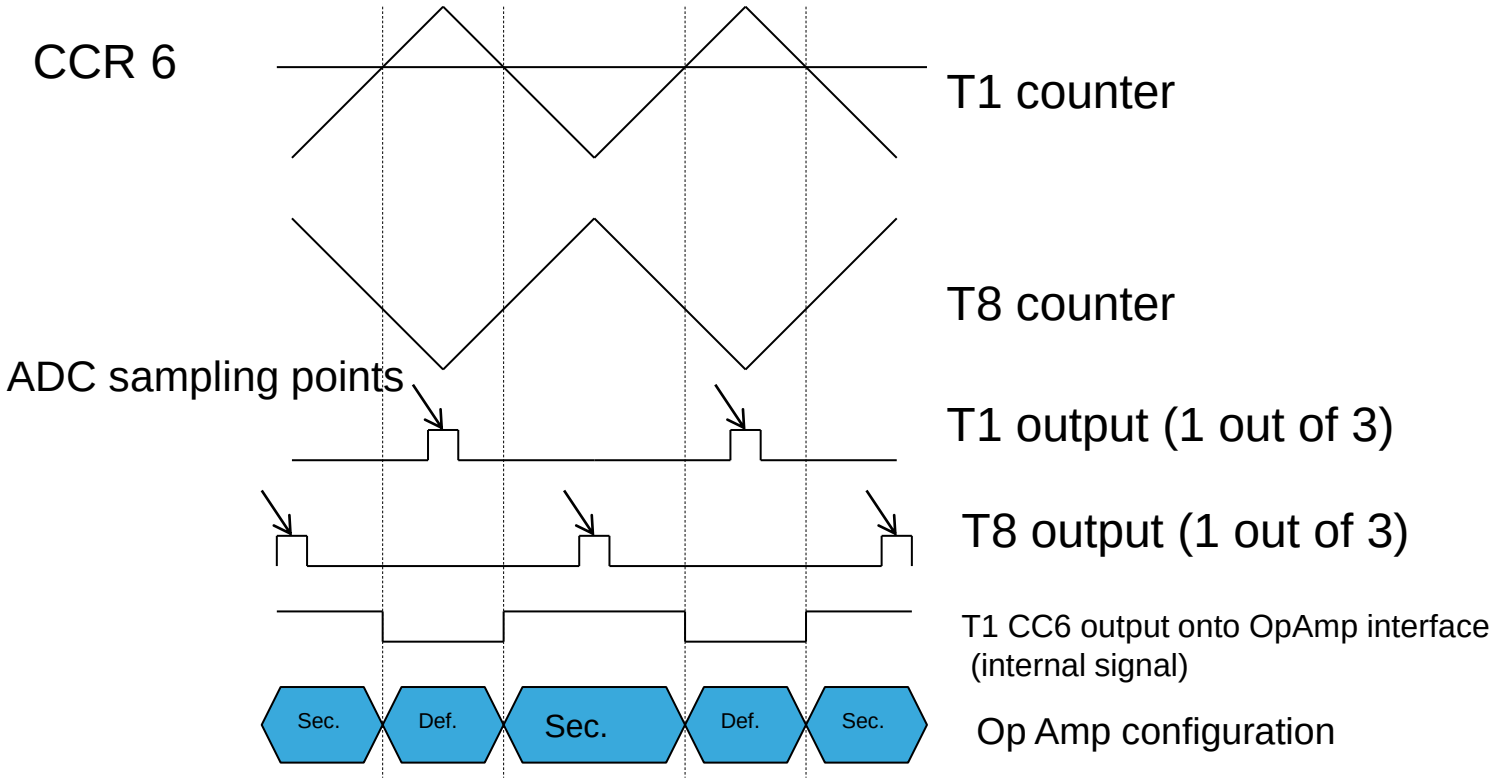
PGA Mode, Internal Gain setting (Gain = 2 / 4 / 8 / 16) with Inverting input used for filtering.



Allows optional low-pass filtering

NB: gain dependant cut-off frequency

Timer Controlled Multiplexer mode (2/2)



STM32F37x Specific Features/ peripherals

life.augmented

Sigma delta analog to digital converter (SDADC)

life.augmented

SDADC introduction (1/2)

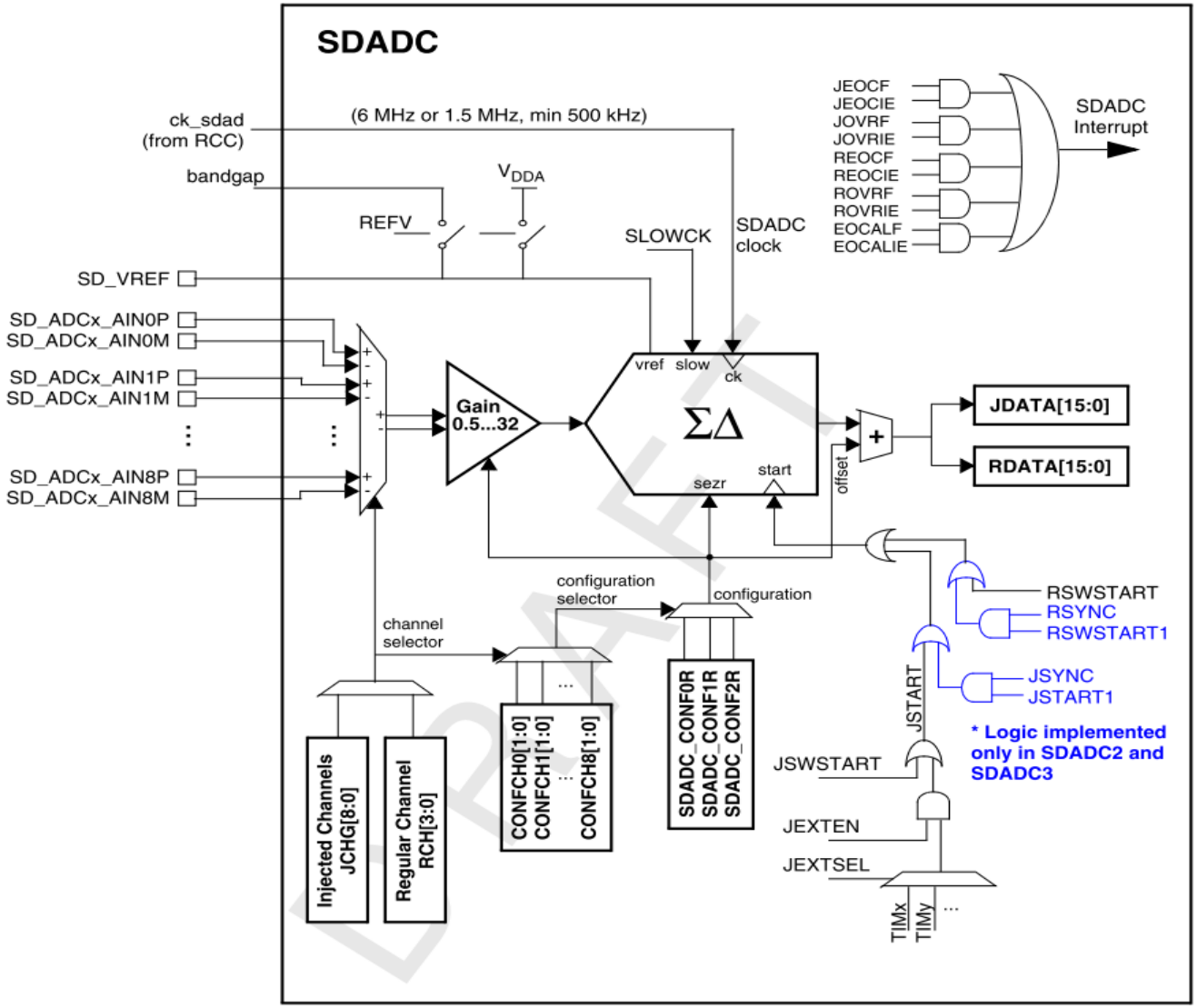
- Sigma delta principle inside STM32:
  - High precision (new applications: medical, metering, gaming)
  - Excellent linearity (simplifies calibration)
  - No sample & hold
- Main properties:
  - 3 Σ-Δ ADCs in all packages (19 single ended and 10 differential inputs max.)
  - 16-bit resolution, ENOB = 14 bits (SNR = 89dB)
  - Low power modes:
    - Slow (speed reduced 4x): up to 600uA (instead of 1200uA in run mode)
    - Standby: up to 200uA, wakeup time 50us
    - Power down: up to 10uA, wake up time 100us
  - Internal or external reference voltage usage
  - Independent power supply pins: SDADCx_VDD
  - Conversion rates:
    - Up to 50ksps in fast mode (single channel)
    - Up to 16.6ksps in normal mode (multiple channels)
  - 7 programmable gains: ½, 1, 2, 4, 8, 16*, 32* (* = digital gains)

SDADC introduction (2/2)

- Next features:
  - 9 single ended inputs or 5 differential inputs per one SDADC (or combination)
  - DMA capability to transfer data to RAM (conversion when CPU in sleep mode)
  - Triggers:
    - Software
    - Timer
    - External pin
    - Synchronization to first SDADC (SDADC1)
  - Signed output data format (16-bit signed number)
  - Zero offset calibration
  - 3 measuring modes – per analog channel selection:
    - Single ended referenced to zero
    - Single ended offset mode
    - Differential mode
  - Interrupts and flags:
    - Interrupts: EOCAL, REOC, JEOC, ROVR, JOVR
    - Flags: STABIP, CALIBIP, RCIP, JCIP

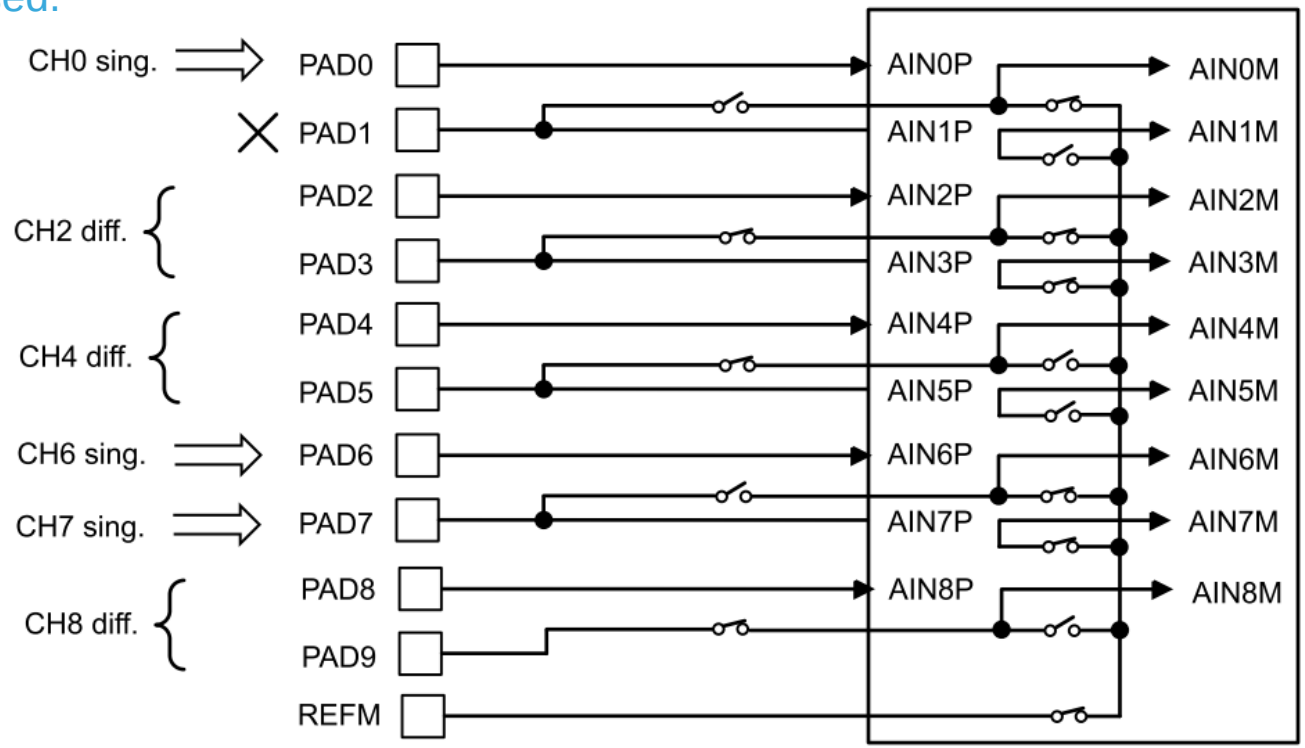
SDADC block diagram

- One SDADC configuration:



Channels configuration example

- Mixed configurations – example of input pins connection:
  - CH2, CH4 and CH8 are used as differential.
  - CH0, CH6 and CH7 are used in single-ended mode.
  - REFM is used – VSSA.
  - PAD 1 is not used.



Regular and injected conversions

- Injected conversions
  - Injected group is defined as bitfield in register – each one bit corresponds to one channel
  - Selected channels in the injected group are always converted sequentially (from lowest selected channel) – scan mode
  - Triggers:
    - Software (writing ‘1’ to the JSWSTART bit)
    - External pin
    - Timers
    - Synchronous with SDADC1
- Regular conversions
  - Channel selection is defined as channel number in register
  - Cannot run in scan mode
  - Triggers:
    - Software (writing ‘1’ to the RSWSTART bit)
    - Synchronous with SDADC1

Standard, slow, low power conversion modes

- Standard mode:
  - Normal:
    - Multiplexing more channels
    - One conversion takes 360 cycles (16.6ksps @ 6MHz)
  - Fast continuous (FAST = 1):
    - On one channel only in continuous mode – regular channel or one injected channel selected
    - One conversion takes 120 cycles (50ksps @ 6MHz)
- Slow mode (SLOWCK = 1):
  - Reduced power consumption (~600uA consumption), operation from 2.2V
  - Limited clock speed – up to 1.5MHz (so 4x reduced also conversion rate)
- Standby when idle (SBI = 1):
  - SDADC goes to standby when no conversion (~200uA consumption)
  - Needed time for wakeup from power down – 50us
- Power down when idle (PDI = 1):
  - SDADC goes to power down when no conversion (~10uA consumption)
  - Needed time for wakeup from power down – 100us

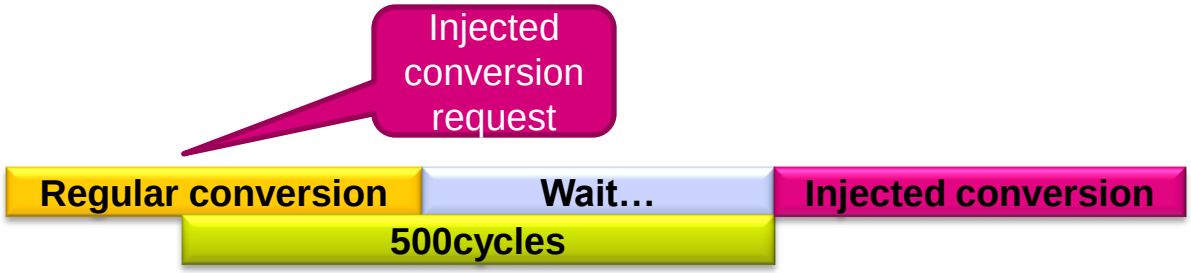
SDADC calibration

- General properties for sigma delta converters:
  - Perfect linearity (due to 1-bit converter and oversampling)
  - Resolution increases with decreasing data rate
  - But large offset and gain error (need calibration)

- Offset calibration:
  - Principle:
    - Short internally both channel inputs (positive and negative)
    - Perform conversion and store result to configuration register(s)
    - During standard conversion subtract from result the calibrated value
  - Implementation in STM32F37x:
    - Set in configuration registers:
      - required gain (1/2 .. 32)
      - common mode for calibration (VSSA, VDDA, VDDA/2)
    - Set how many configurations to calibrate (CALIBCNT[1:0] bits)
    - Start calibration by setting bit STARTCALIB
    - Calibration sequence then executes on given gain(s) :
      - Calibration values are stored into configuration registers (OFFSETx[11:0] bits)
      - 30720 cycles (5.12 ms at 6 MHz) for one configuration register
    - Calibration data are automatically subtracted from each conversion data

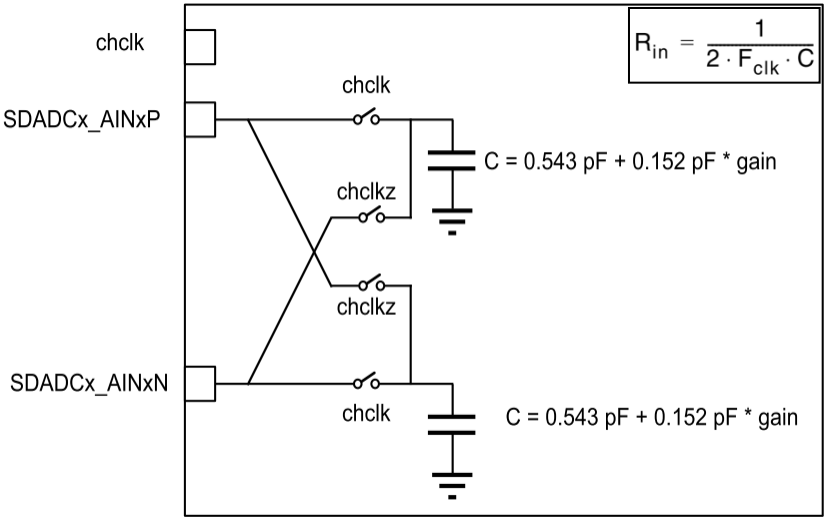
Deterministic timing

- Application requirements:
  - Launching conversion in precise intervals (e.g. FFT sampling by timer trigger)
  - Problem: waiting for some ongoing (regular) conversion
- Solution in SDADC:
  - Start of each injected conversion with delay during which cannot be started regular conversion
  - When bit JDS = 1 (Injected Delay Start) the start of each injected conversion is delayed:
    - by 500 cycles if PDI = 0 (power down when idle)
    - by 600 cycles if PDI = 1, SLOWCK = 0 (because wakeup from power down takes 600 cycles)

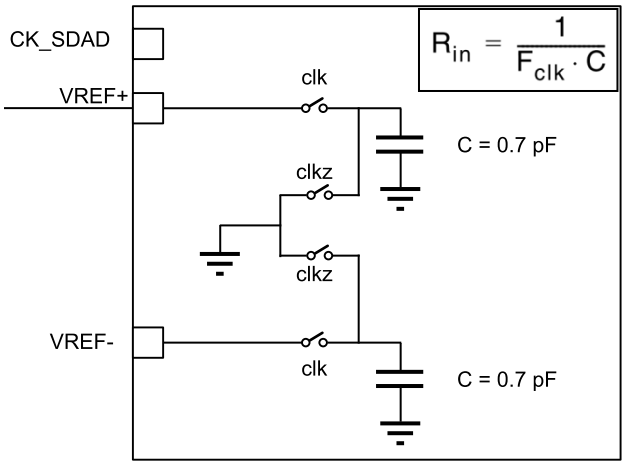


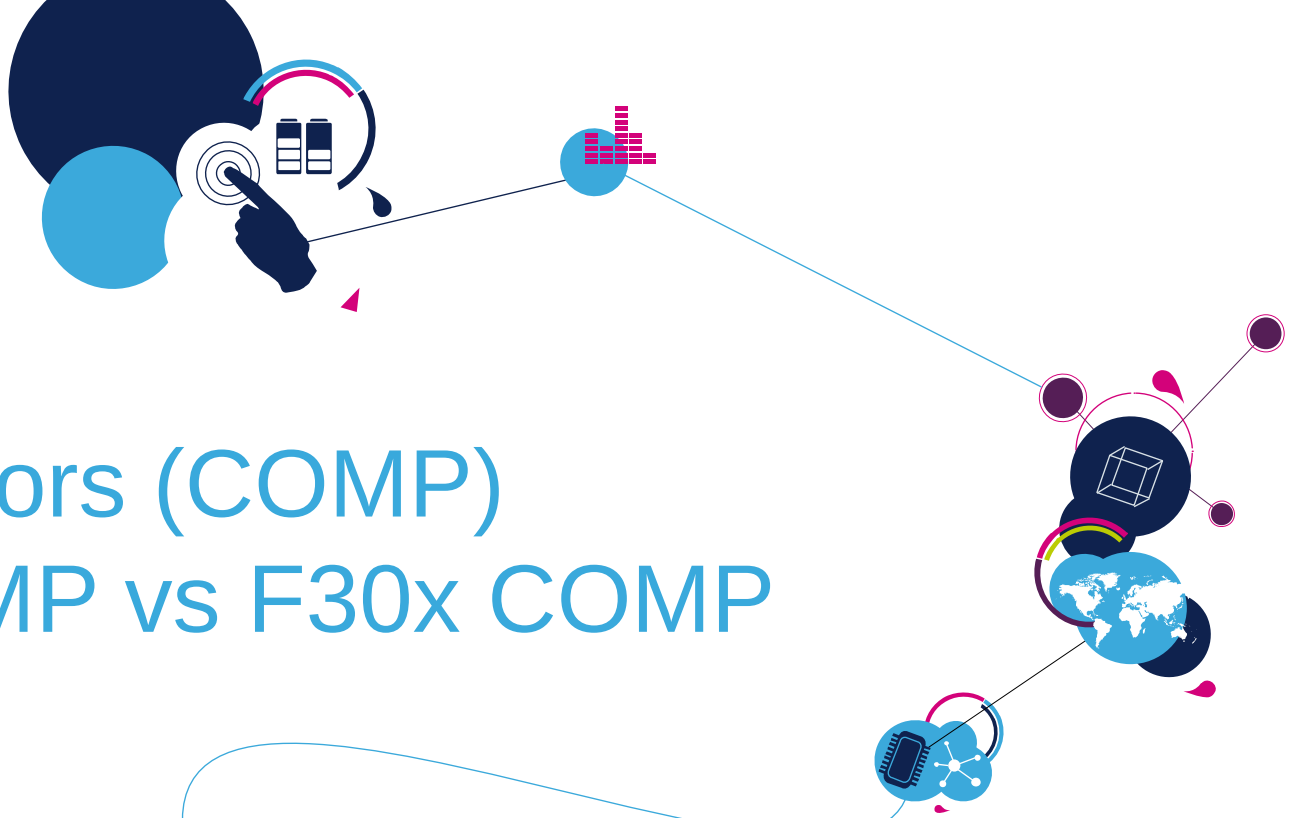
Inputs impedances

- Analog inputs impedance:
  - Depends from:
    - selected SDADC clock
    - analog gain (0.5 – 8)
    - conversion is in progress
  - Switching capacitance character
  - Range (examples):
    - 540kΩ @ 1.5MHz, gain = 0.5
    - 135kΩ @ 6MHz, gain = 1
    - 47kΩ @ 6MHz, gain = 8



- Reference voltage input impedance:
  - Depends only from selected SDADC clock
  - Switching capacitance character
  - Range (6MHz – 1.5MHz):
    - ~ 230kΩ – 1000kΩ





Comparators (COMP)
F37x COMP vs F30x COMP

life.augmented

F37x COMP vs F30x COMP

- 2 comparators (7 in STM32F30x)
- A single register manages both comparators (in STM32F30x: one regsiter per comparator).
- No mux on the non inverting input
- No blanking feature

ADC 1 MSPS

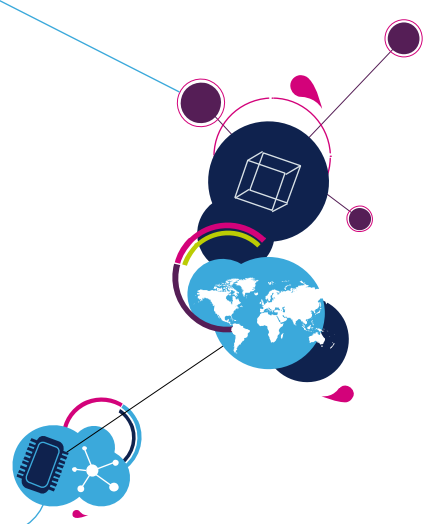
life.augmented

ADC Features

- Same like in STM32F1 family:
  - 12-bit, 1Msps
  - Triggers, self-calibration
  - Up to 18 input analog channels
  - Analog watchdog, interrupts, DMA
  - Programmable sampling time, Vref+ input range
  - Injected, regular channels, alignment
  - Continuous, single, scan conversion modes
  - Temperature sensor, Vrefint measuring
- Added feature:
  - VBAT measuring

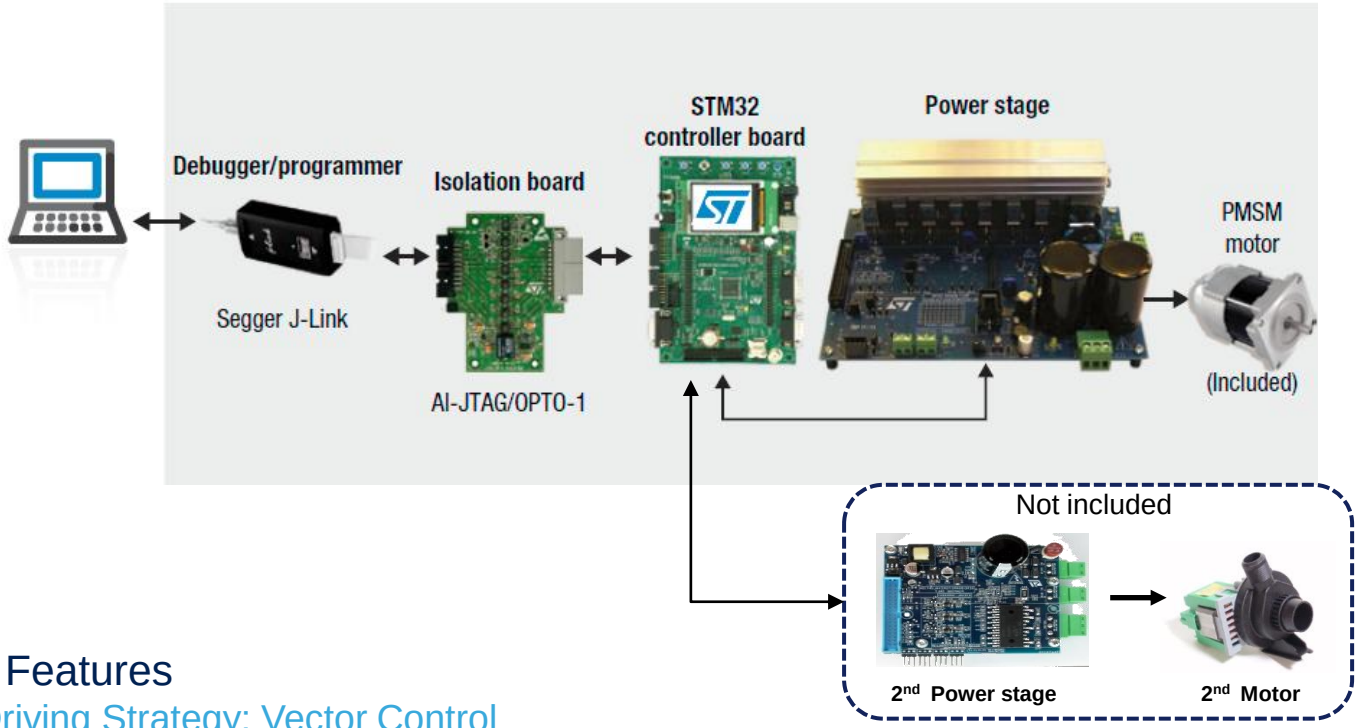
STM32F30x Motor Control Features

Gianluigi FORTE (SystemLab)



life.augmented

STM32F3 MC kit

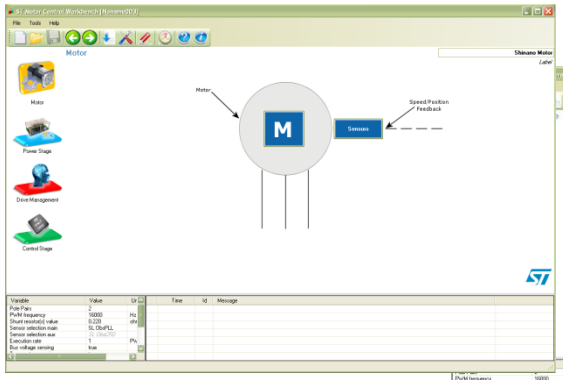


- Main Features
  - Driving Strategy: Vector Control
  - PMSM motor sensored and sensorless
  - Two (34-pin) dedicated motor control connectors
  - Encoder sensor input
  - Hall sensor input
  - Tachometer sensor input
  - Current sensing mode:
    - 3 shunt resistors
    - Single shunt

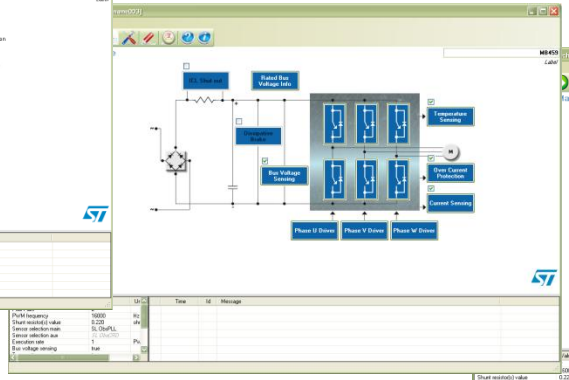
- Key Component
  - STM32F3xx (32-bit MCU ARM M4 with motor control dedicated IPs)
  - L6390D (Gate Drivers)
  - VIPer16LD (Power Supply down converter)
  - L7815ABV, L78M05CDT, LD1117S33TR (Voltage regulators)
  - STGP10NC60KD (IGBT)
  - TS391ILT, (Comparator)
  - M74HC14TTR (Logic)

MC Workbench

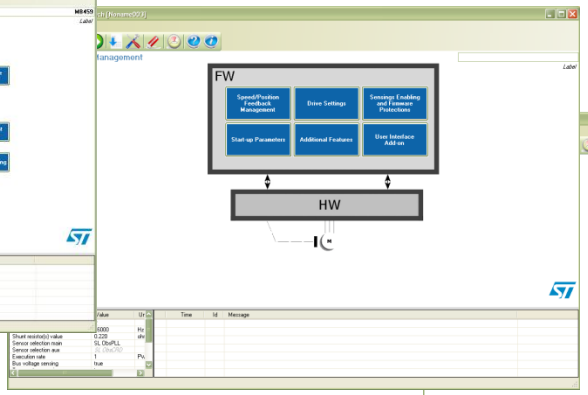
Quick setup of the library according customer needs



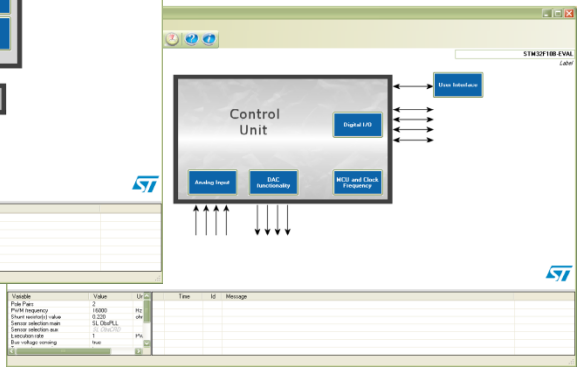
Motor



Power Stage



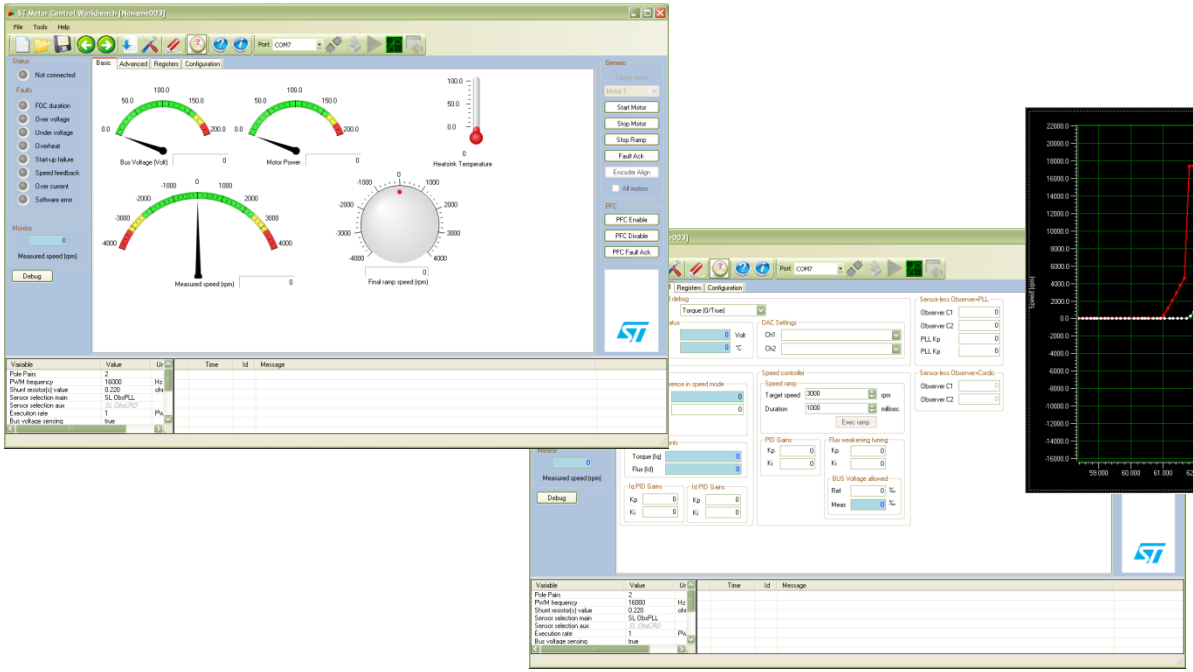
Drive Management

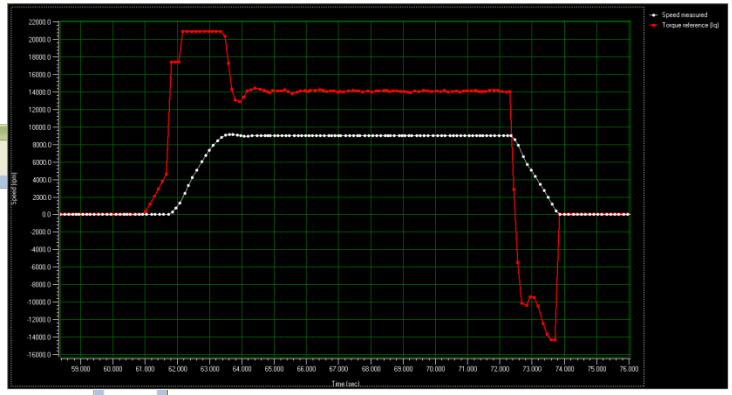


Control Stage

- ST Motor Control Workbench
  - PC software that reduces the design effort and time in the STM32 PMSM FOC firmware library configuration. The user through a graphical user interface (GUI) generate all parameter header files which configures the library according the application needs.

Serial communication



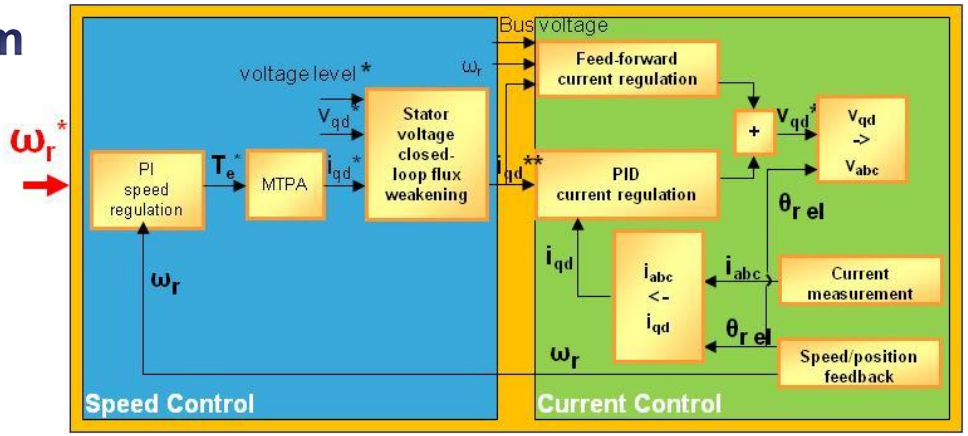


- RS232 (Available)
- SPI (T.B.I.)
- I2C (T.B.I.)

- Real time communication
  - Using the ST MC workbench is possible to instantiate a “real time communication” to send start/stop commands or to set a speed ramp.
  - Debug or fine tuning motor control variables (like speed PI parameters) can be assessed using the advanced tab.
  - Plotting significant motor control variables (virtual oscilloscope) like target or measured motor speed.

Cortex M4 + Floating point unit

FOC Algorithm



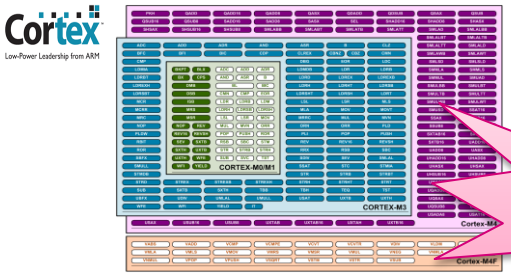
Execution rate 10-20kHz

Two times for dual drive

Flexible design

Cortex-M0	Cortex-M3	Cortex-M4
"8/16-bit" applications	"16/32-bit" applications	"32-bit/DSC" applications

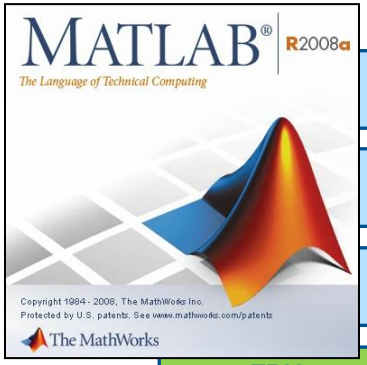
Binary and tool compatible



* *Expected*

-10% CPU Load*

+



High level approach
Matrix, mathematical equations

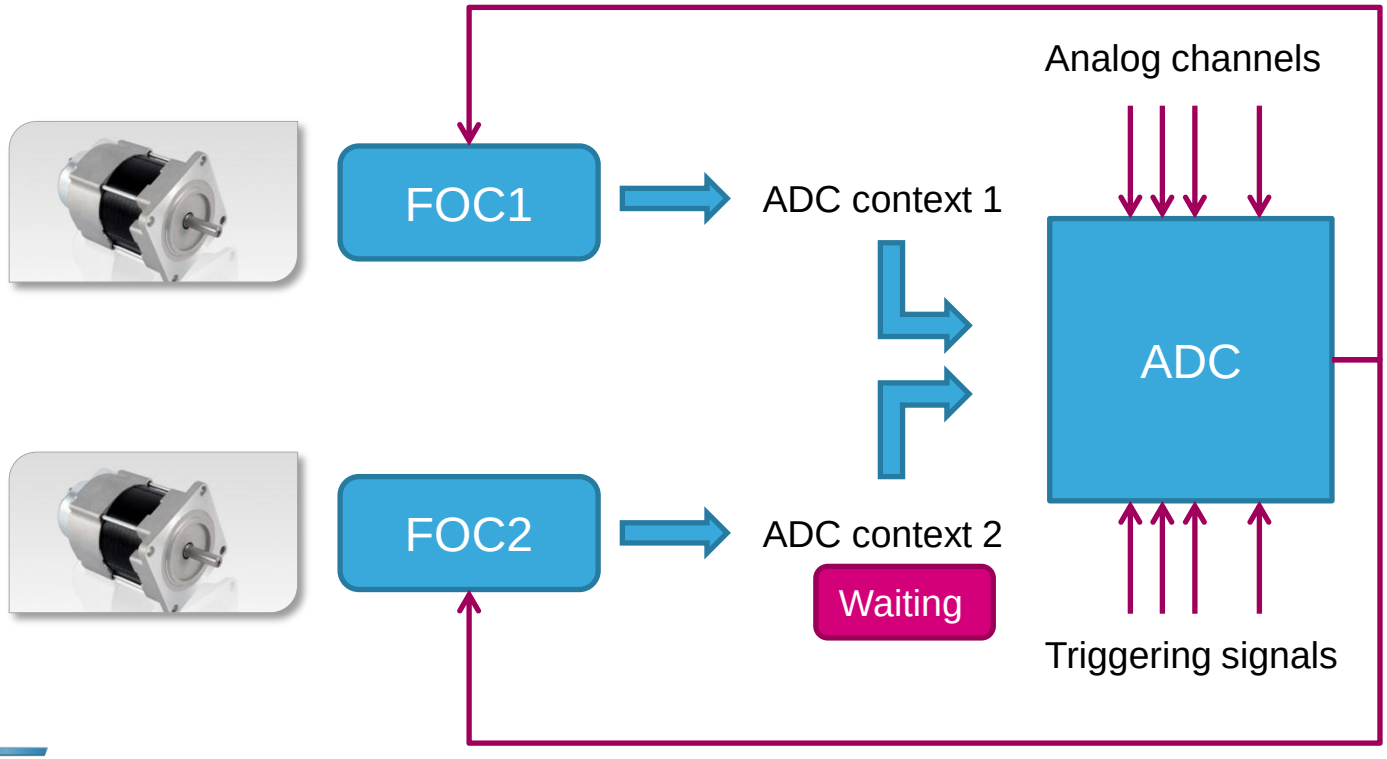
Meta language tools
Matlab ,Scilab…etc…

C code generation
Floating point numbers (`float`)

FPU	No FPU	No FPU
Direct mapping	**Usage of SW lib**	**Usage of integer based format**
No code modification	No code modification	Code modification
High performance	Low performance	Corner case behavior to be checked (saturation, scaling)
Optimal code efficiency	Medium code efficiency	Medium/high performance
		Medium code efficiency

ADC context FIFO

- FOC1 requires a conversion of channel x triggered by signal y (ADC context 1)
- FOC2 requires a conversion of channel n triggered by signal m (ADC context 2) but the ADC has been reserved so the context is stored in the FIFO
- Signal y triggers the conversion and the result is sent to FOC1. The FIFO go ahead programming the context 2
- Signal m triggers the conversion and the result is sent to FOC2.



For simplicity only one ADC is used in this Example

Thank you !