STLUX - STNRG
Exercise the SMED

Four simple examples using the SMEDs

Agrate 06 April 2016

life.augmented

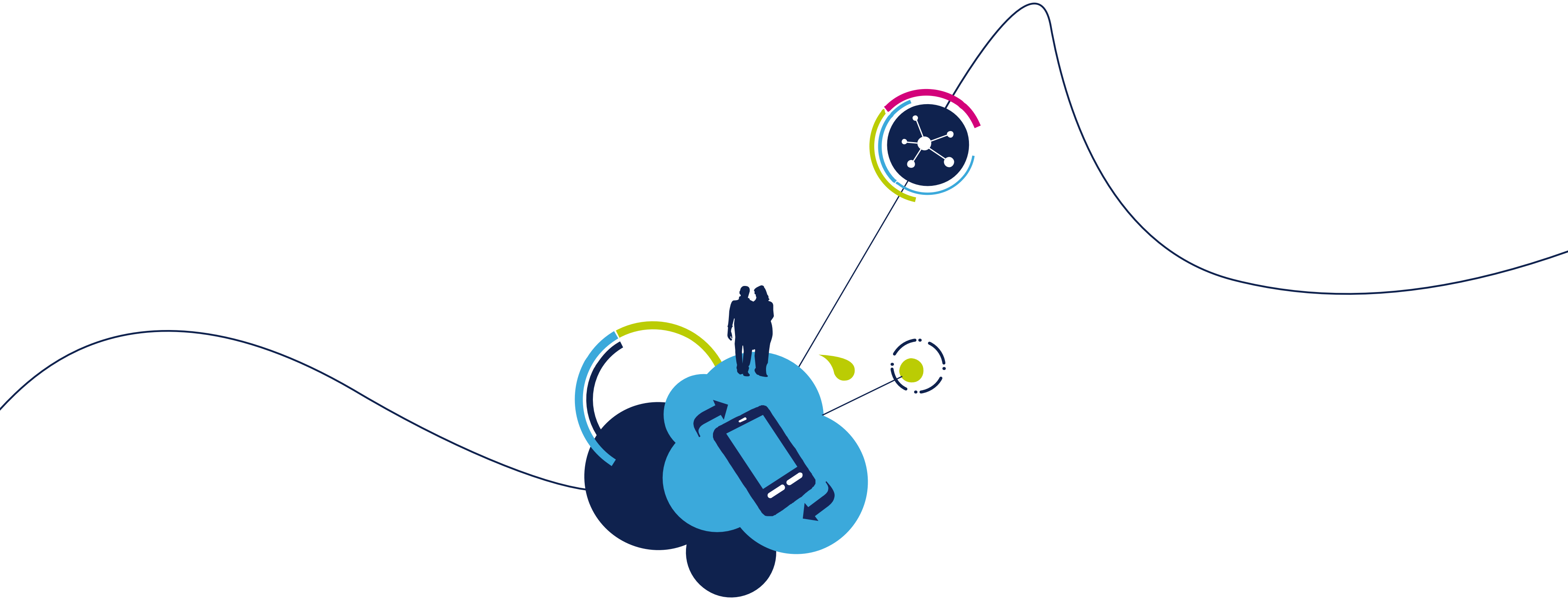
ST Confidential

Purpose

- Creating two examples to practice with the SMEDs and STLUX385A
- Creating two examples to practice with the SMEDs and STNRG388A
- Use the STLUX-STNRG Library
- Use the STEVAL-ILL075V1 and STEVAL-ISA164V1 dev board
- Use the SMEDs configurator
- Use the Raisonance/IAR IDE toolset and R-LINK/ST-LINK

- Two STLUX examples:
  - Using an ADC line control the duty of a PWM line (SMED0)
  - Generate a variable PWM line (SMED5) with CPP/DAC level

- Two STNRG examples:
  - Generate a variable PWM line (SMED5) with CPP/DAC level with hysteresis
  - Generate two variable PWM line (SMED2-SMED3) out of a variable PWM (SMED5) which keeps the voltage of a circuit within two fixed CPP/DAC levels

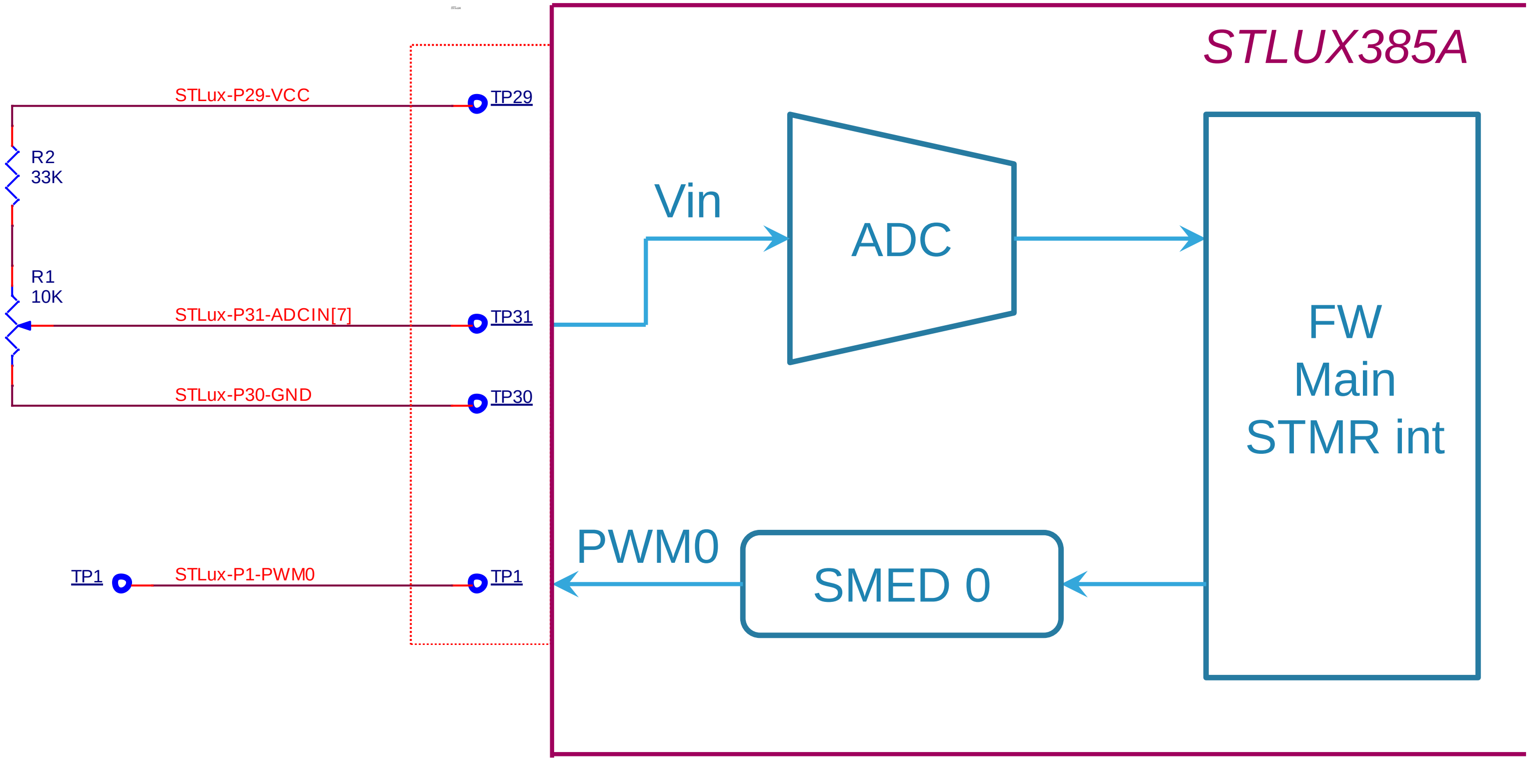


STLUX Example IV

STEVAL-ILL075V1 STLUX385A Dev board

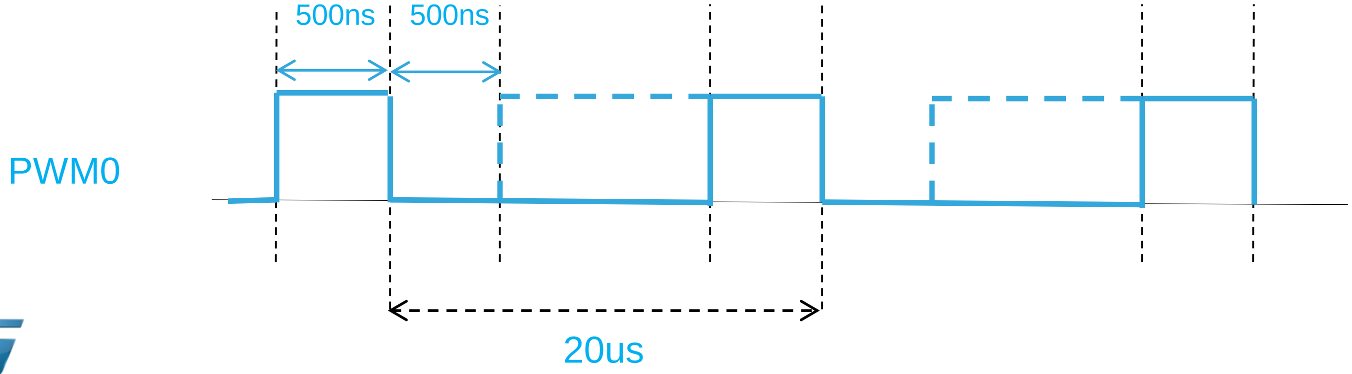
Example IV - hardware

- Creating a PWM signal as a function of an analog input.
- PWM0 output has fixed frequency (50KHz) but variable duty cycle



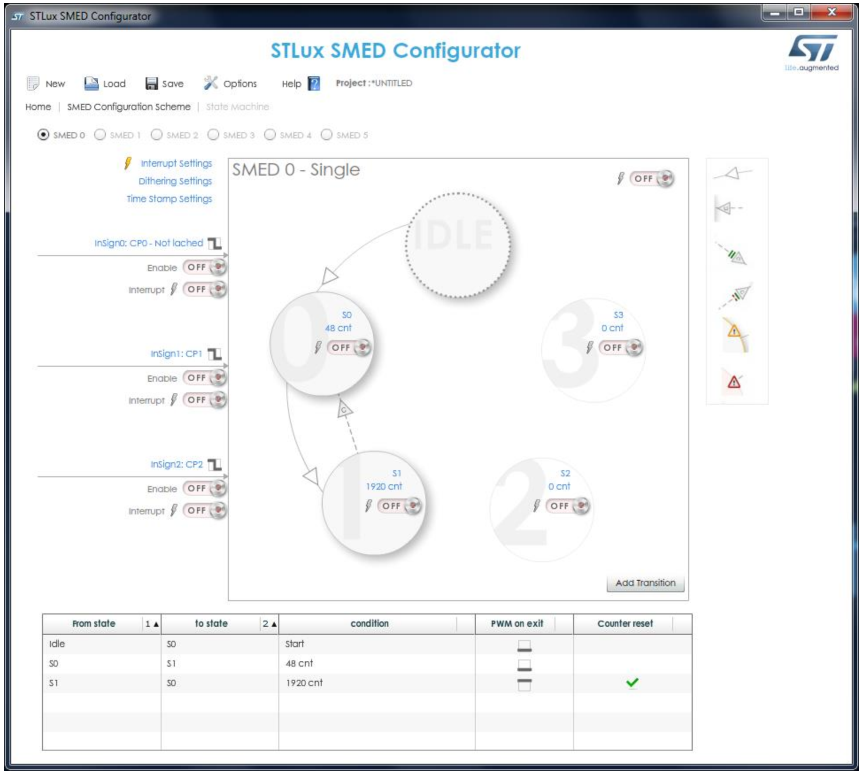
Example IV - specification

- Creating a PWM signal controlled by a voltage level:
  - When Vin is below 200mV:
    - PWM0 is OFF at low voltage.
  - When Vin is from 200mV to 1V:
    - PWM0 freq. is fixed at 50KHz but duty cycle is from 2.5% to 97.5% (step 10nS)
  - When Vin is higher then 1V:
    - PWM0 is OFF at low voltage
  - Minimum on/off time:
    - 500ns



Example IV - SMED

- SMED 0 behavior: Two states
  - S0 set the PWM ON time
  - S1 set the PWM OFF time
  - SMED update mode: counter reset
- S0 is the variable time derived by the ADC
- Save as “*SMED_Init.c*” file
- Exercise:
  - Modify SMED0 behavior
    - Use S0 for min-on time 500nS
    - Use S1 as the variable time
    - Use S2 for min-off time 500nS



From state	to state	condition	PWM on exit	Counter reset
Idle	S0	Start	_	
S0	S1	48 cnt	_	
S1	S0	1920 cnt	‾	✓

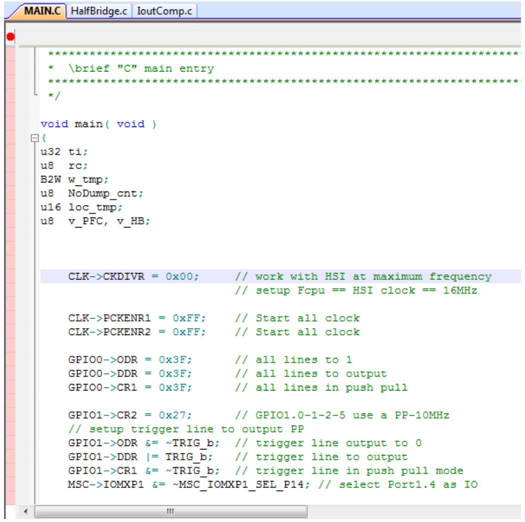
Example IV - firmware

- In the main:
  - Initialize IO pin, PLL, SMED, ADC, STMR
  - Wait interrupt flag

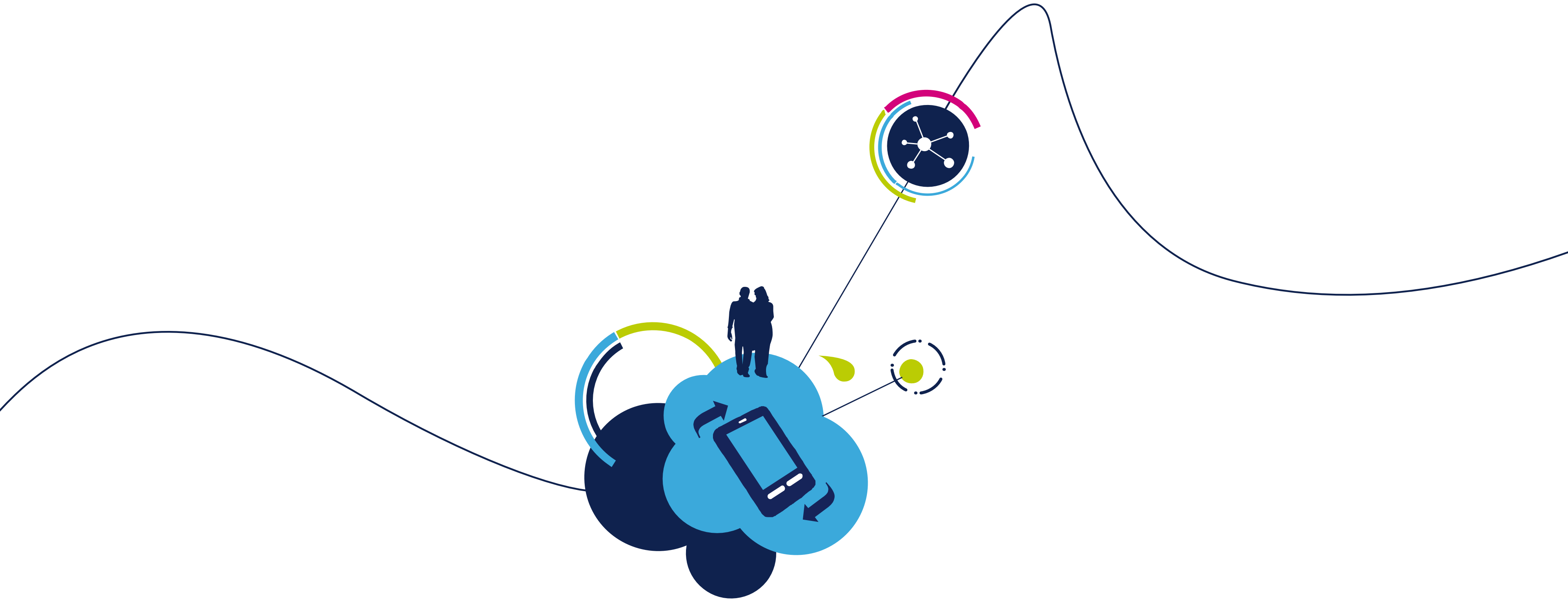
- In the STMR interrupt:
  - Acquire the ADC data value
  - Start/Stop the SMED
  - Compute the S0 CMP timer value
  - Update and validate the S0 time

Example IV – compile and debug

- Compile, download and debug the application using SWIM interface:
  - Look at the SMED0 output PWM while changing the ADC[7] input value
  - Set Vin=0.6V → PWM duty at 50%
  - Set Vin=0.1V → PWM fixed low
  - Set Vin=0.95V → PWM duty near 95%



```
 **********************************************************************
 *  \brief "C" main entry
 **********************************************************************
 */

void main( void )
{
u32 ti;
u8  rc;
B2W w_tmp;
u8  NoDump_cnt;
u16 loc_tmp;
u8  v_PFC, v_HB;



    CLK->CKDIVR = 0x00;       // work with HSI at maximum frequency
                              // setup Fcpu == HSI clock == 16MHz

    CLK->PCKENR1 = 0xFF;      // Start all clock
    CLK->PCKENR2 = 0xFF;      // Start all clock

    GPIO0->ODR = 0x3F;        // all lines to 1
    GPIO0->DDR = 0x3F;        // all lines to output
    GPIO0->CR1 = 0x3F;        // all lines in push pull

    GPIO1->CR2 = 0x27;        // GPIO1.0-1-2-5 use a PP-10MHz
    // setup trigger line to output PP
    GPIO1->ODR &= ~TRIG_b;  // trigger line output to 0
    GPIO1->DDR |= TRIG_b;   // trigger line to output
    GPIO1->CR1 &= ~TRIG_b;  // trigger line in push pull mode
    MSC->IOMXP1 &= ~MSC_IOMXP1_SEL_P14; // select Port1.4 as IO
```

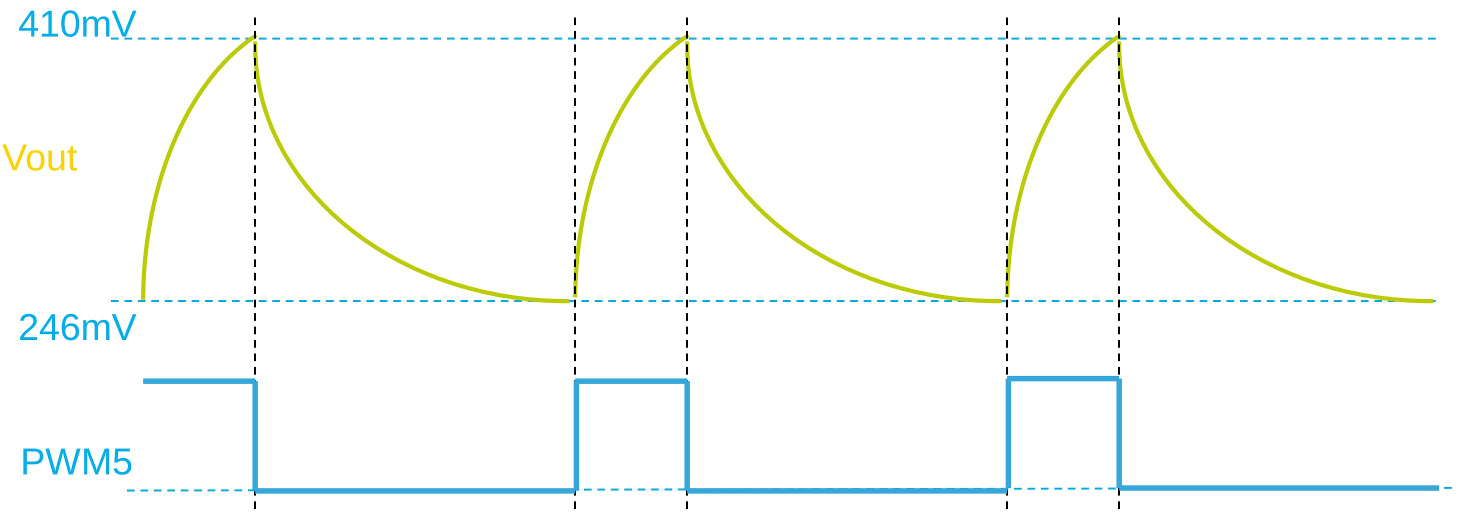


STLUX Example V

STEVAL-ILL075V1 STLUX385A Dev board

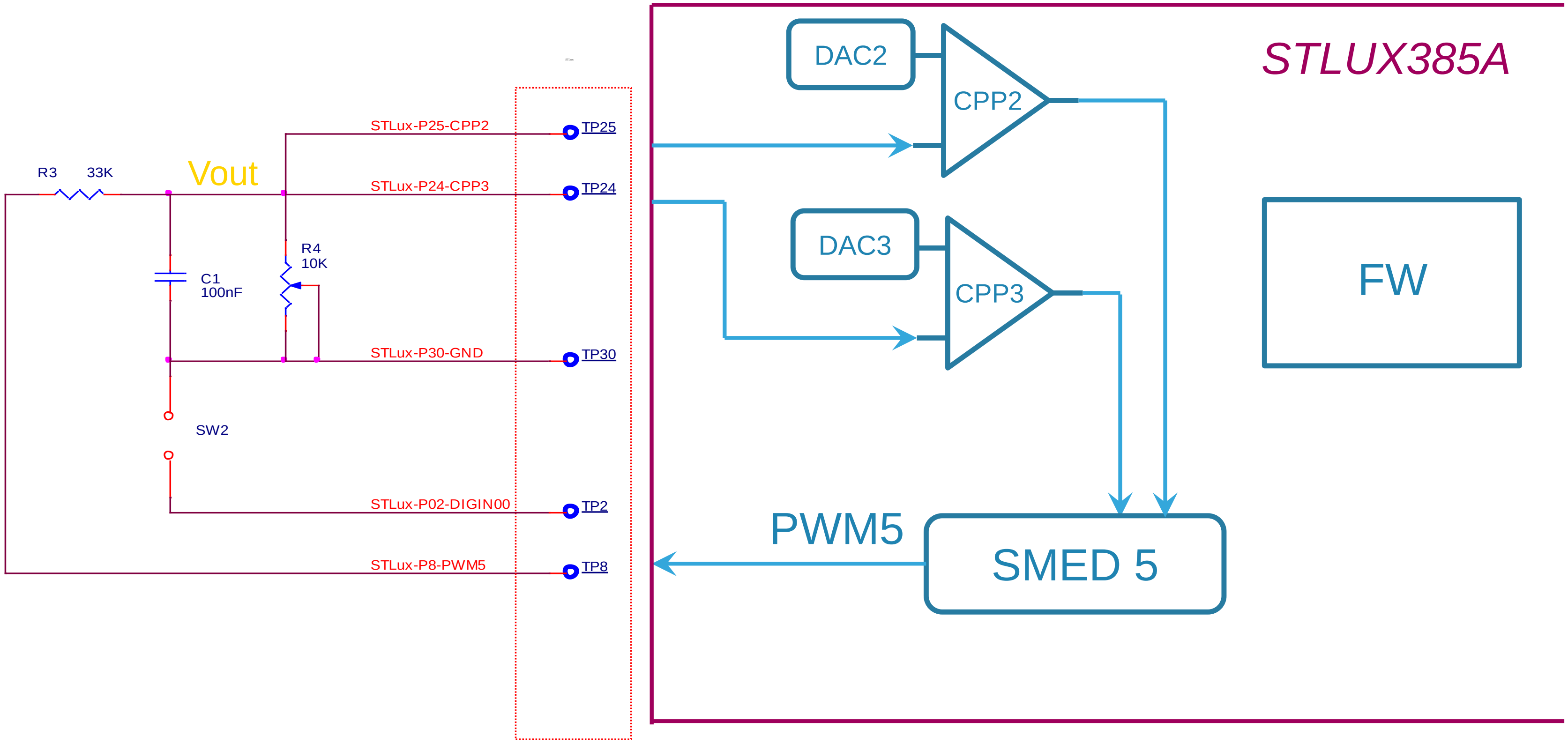
Example V - specification

- Fix the PWM frequency to generate a fixed level:
  - Vout must be kept over 246mV:
    - PWM5 is turned ON when voltage reaches the low threshold -> DAC3/CPP3
  - Vout must be kept under 410mV:
    - PWM5 is turned OFF when voltage hits the high threshold -> DAC2/CPP2
  - No firmware interaction, automatically handled by SMED



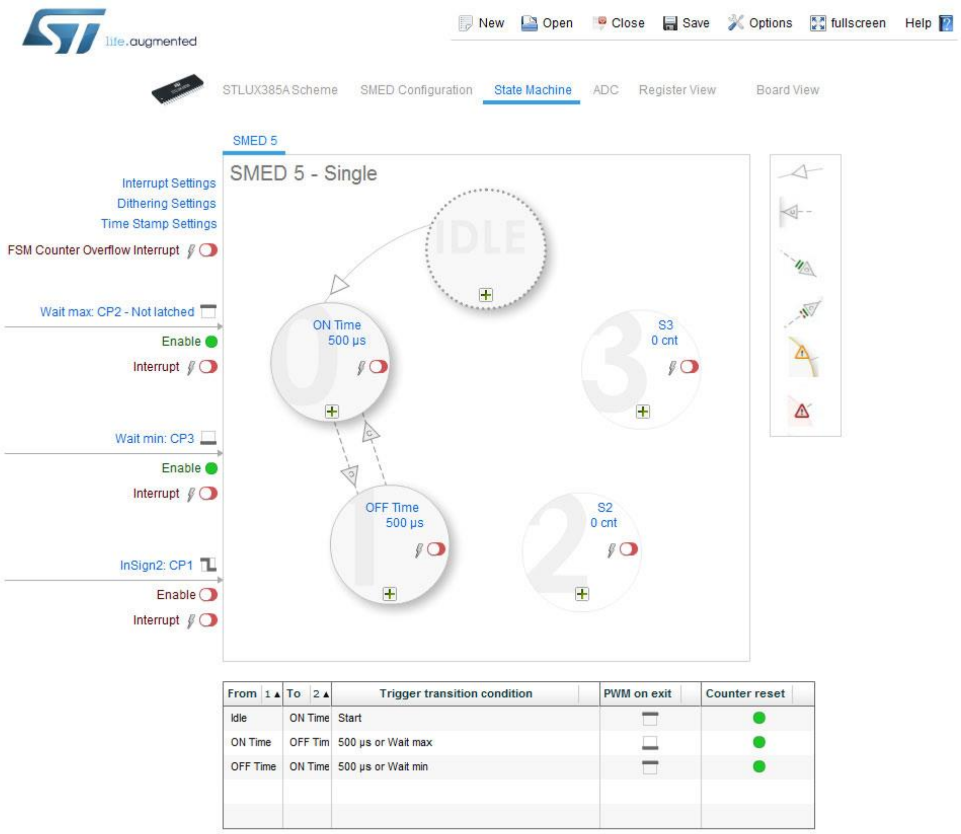
Example V - hardware

- Creating a PWM signal as a function of two comparators
- PWM5 variable frequency to get an analog output voltage between CPP2 and CPP3 references



Example V - SMED

- SMED 5 behavior: Two states
  - S0 set Vout Max level
  - S1 set Vout Min level
  - SMED update mode: event
- Save as “*SMED_Init.c*” file
- Exercise:
  - Modify SMED5 behavior
    - When DIG0 low, halt SMED5 output low



SMED 5 - Single

FSM Counter Overflow Interrupt

Wait max: CP2 - Not latched

Enable ●

Interrupt

Wait min: CP3

Enable ●

Interrupt

InSign2: CP1

Enable ○

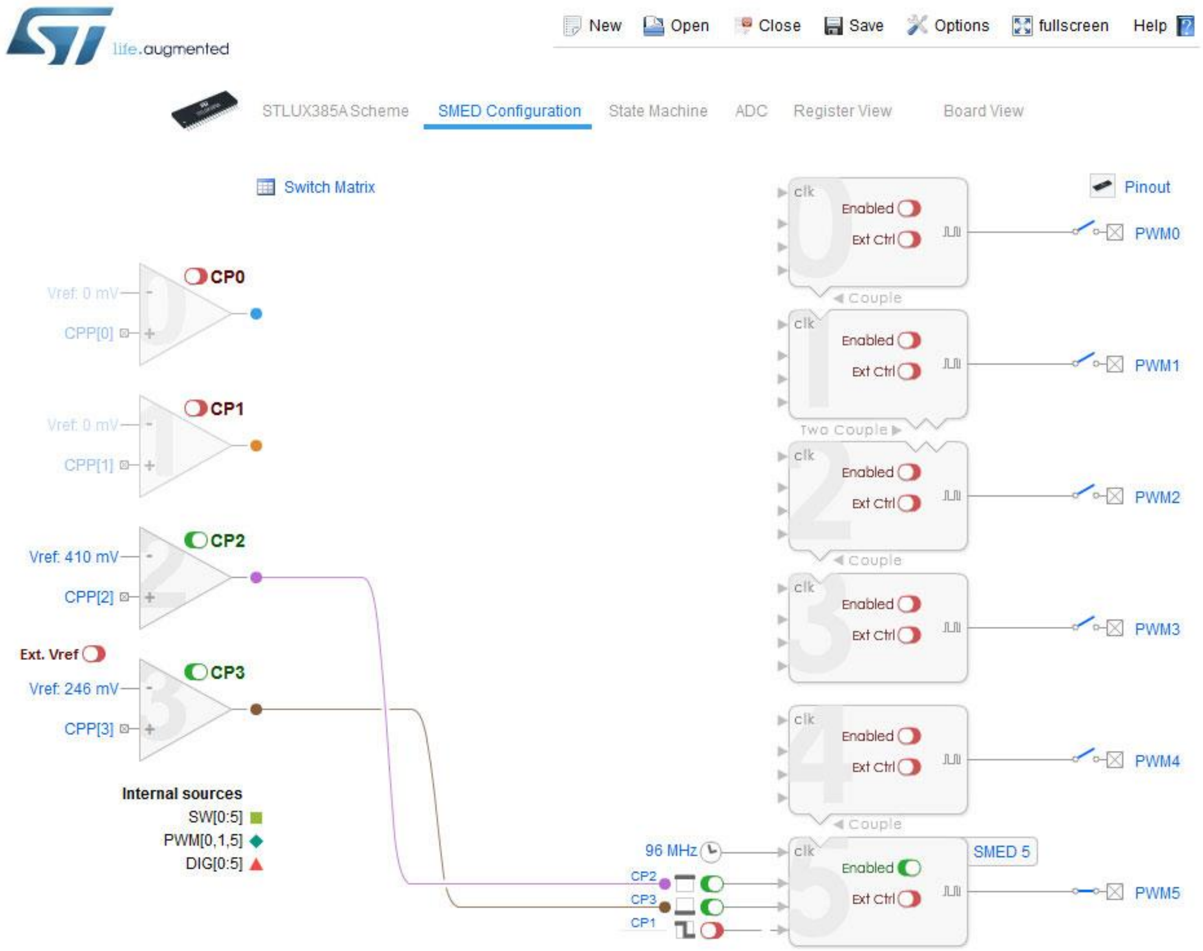
Interrupt

ON Time
500 µs

OFF Time
500 µs

From 1▲	To 2▲	Trigger transition condition	PWM on exit	Counter reset
Idle	ON Time	Start	▔	●
ON Time	OFF Tim	500 µs or Wait max	▁	●
OFF Time	ON Time	500 µs or Wait min	▔	●

Example V - SMED



Example V - firmware

- On the main:
  - Initialize IO pin, PLL, SMED, DAC, CPP
  - Start the SMED5
  - HALT the STM8 processor

- SMED run autonomy, without FW modification

- Possible to use the SMED configurator and serial connection without compilation

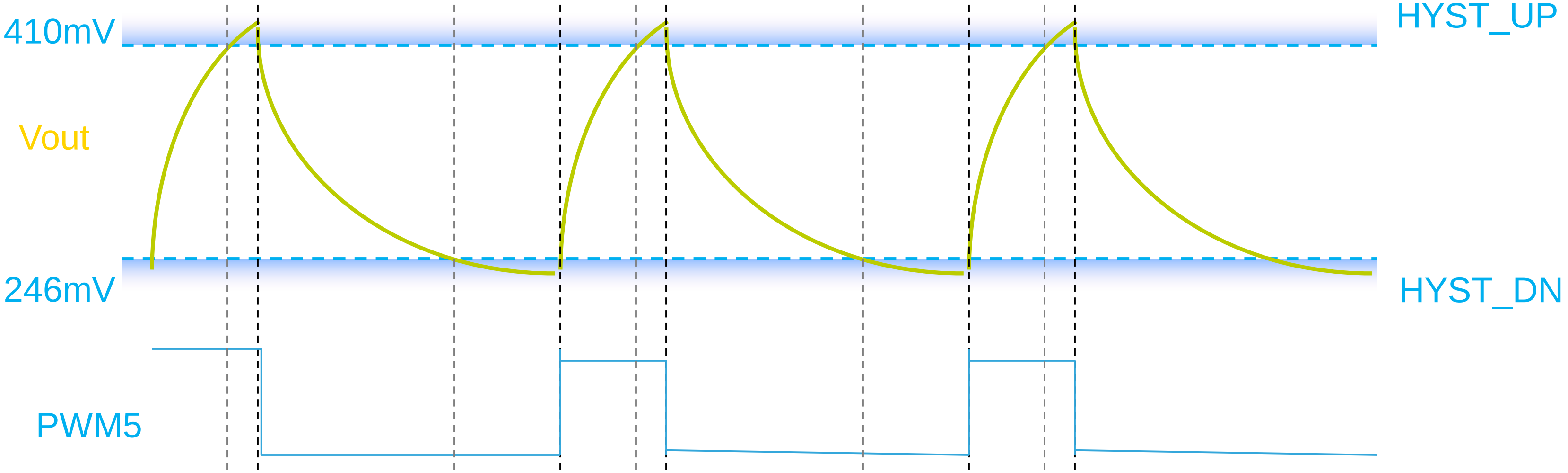
- Exercise
  1. SMED5 is configured in a way that feedback from comparators makes the PWM5 toggle. If the comparator feedback is missed, it gets stuck. Try modifying the state machine so that it restarts after a max_time it gets stuck in a state.
  2. SMED5: Stop the SMED5 behavior when GPIO0-2 is at ground
  3. Using the register write on SMED Configurator, modify the CPM 3 DAC level

STNRG Example VI

STEVAL-ISA164V1 STNRG388A Dev board

STNRG Example VI - specification

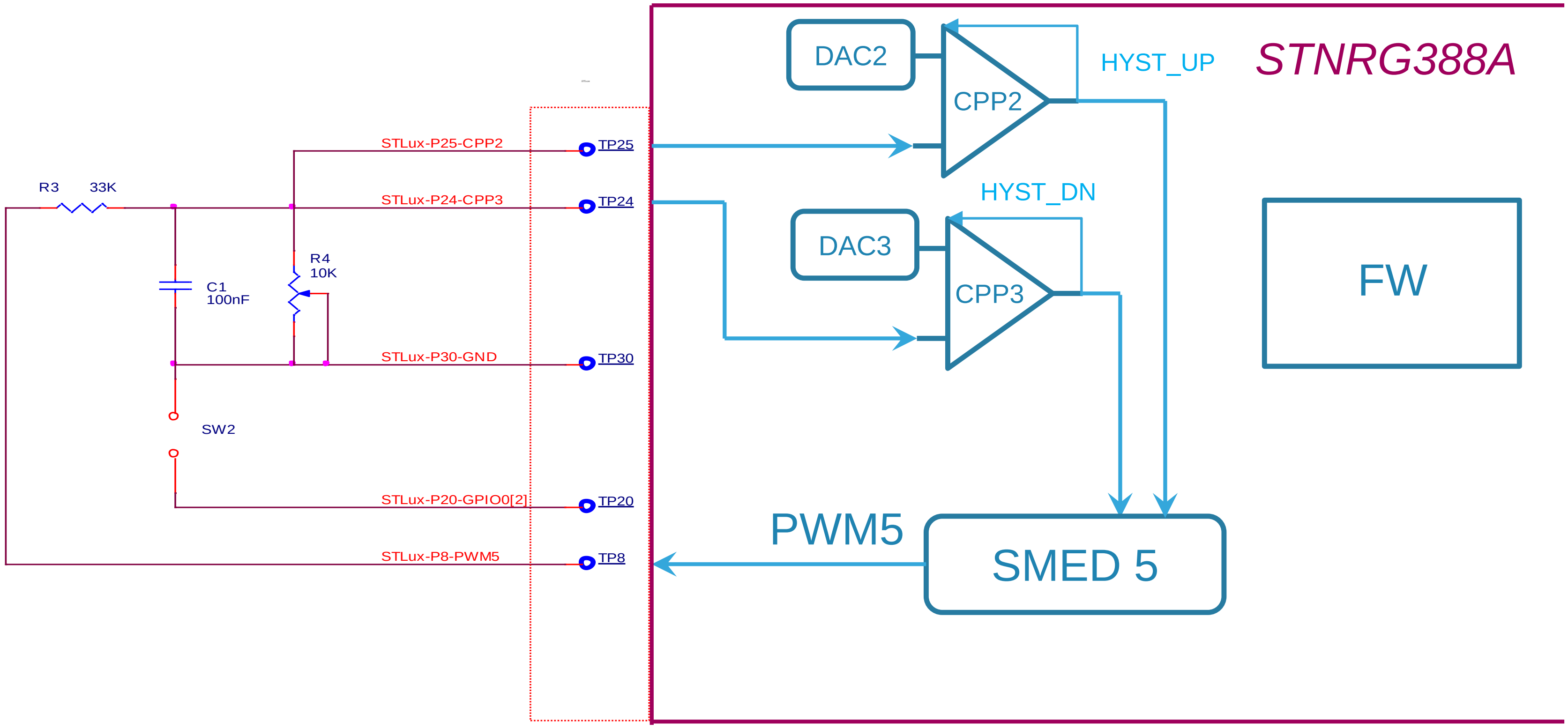
- Vout must be kept over 246mV:
  - A hysteresis down is set on CPP3 so Vout can fall below the threshold
  - PWM5 is turned ON when voltage reaches the low threshold -> DAC3/CPP3-HYST_DN
- Vout must be kept under 410mV:
  - A hysteresis up is set on CPP2 so Vout can rise above the threshold
  - PWM5 is turned OFF when voltage hits the high threshold -> DAC2/CPP2+HYST_UP
- No firmware interaction, automatically handled by SMED



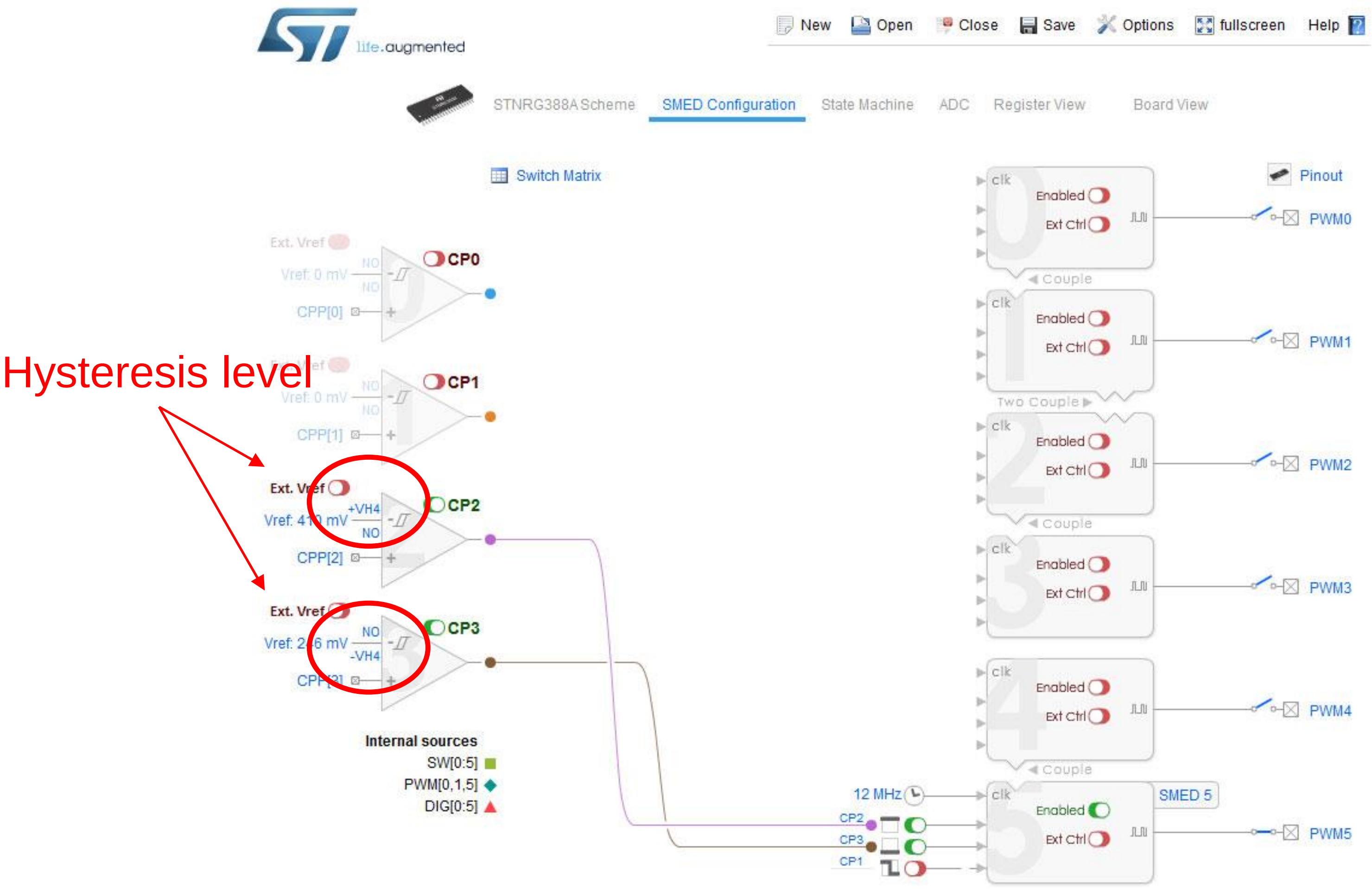
STNRG Example VI - hardware

- Creating a PWM signal as a function of two comparators

- PWM5 variable frequency to get an analog output voltage between CPP2 and CPP3 references

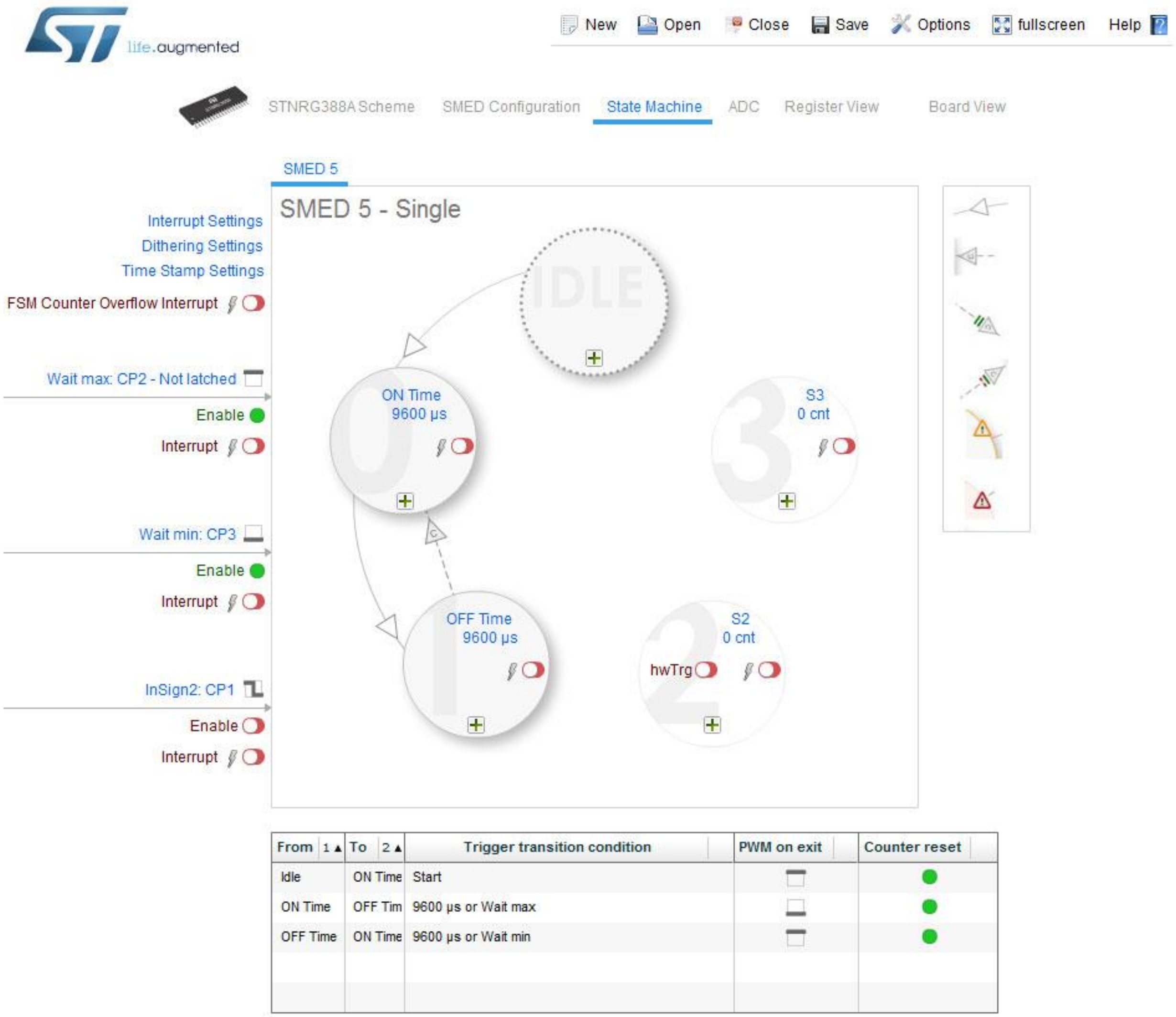


STNRG Example VI - SMED



Hysteresis level

STNRG Example VI - SMED



SMED 5 - Single

Interrupt Settings

Dithering Settings

Time Stamp Settings

FSM Counter Overflow Interrupt ○

Wait max: CP2 - Not latched

Enable ●

Interrupt ○

Wait min: CP3

Enable ●

Interrupt ○

InSign2: CP1

Enable ○

Interrupt ○

ON Time 9600 µs ○

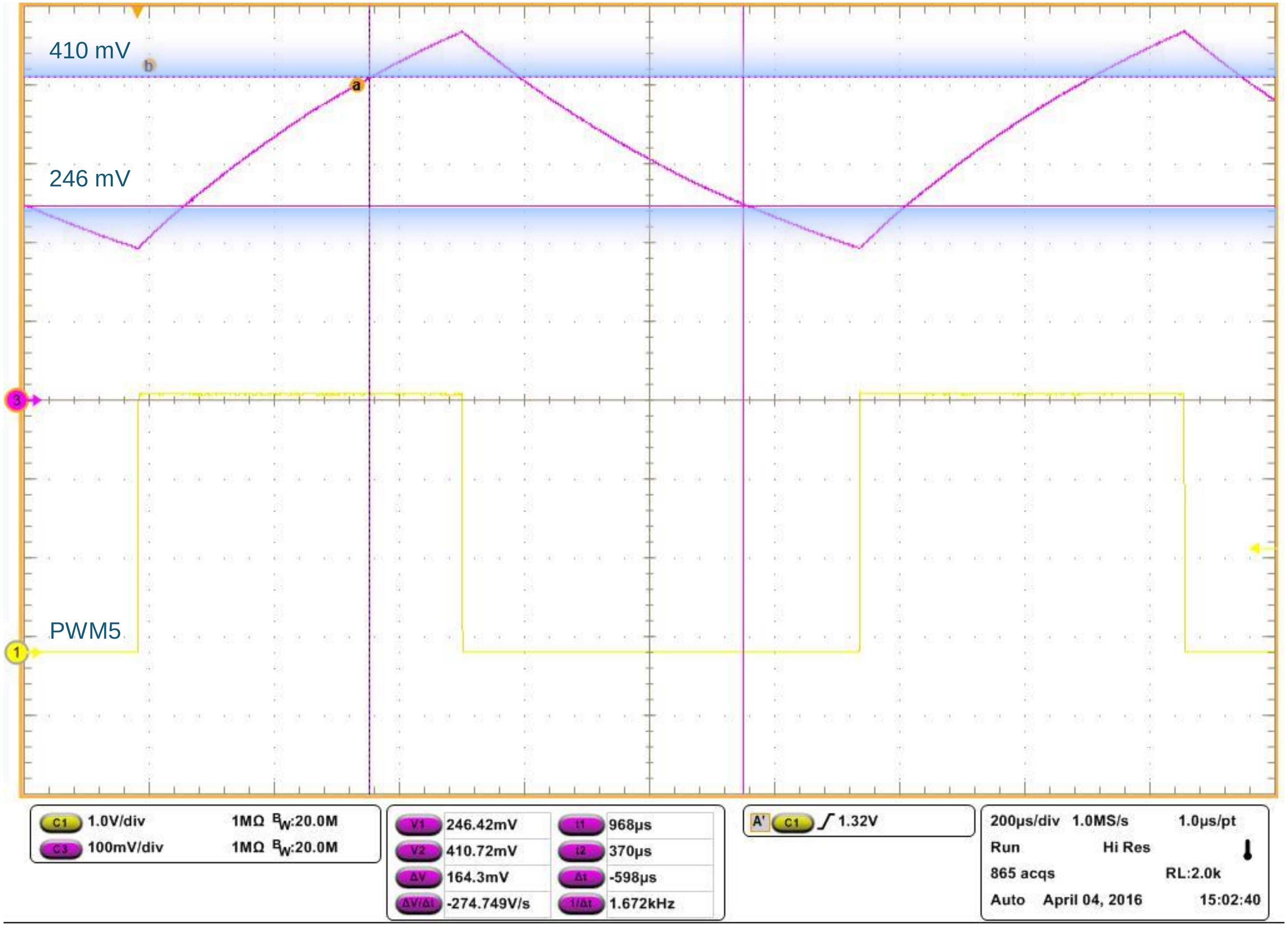
OFF Time 9600 µs ○

S3 0 cnt ○

S2 0 cnt hwTrg ○ ○

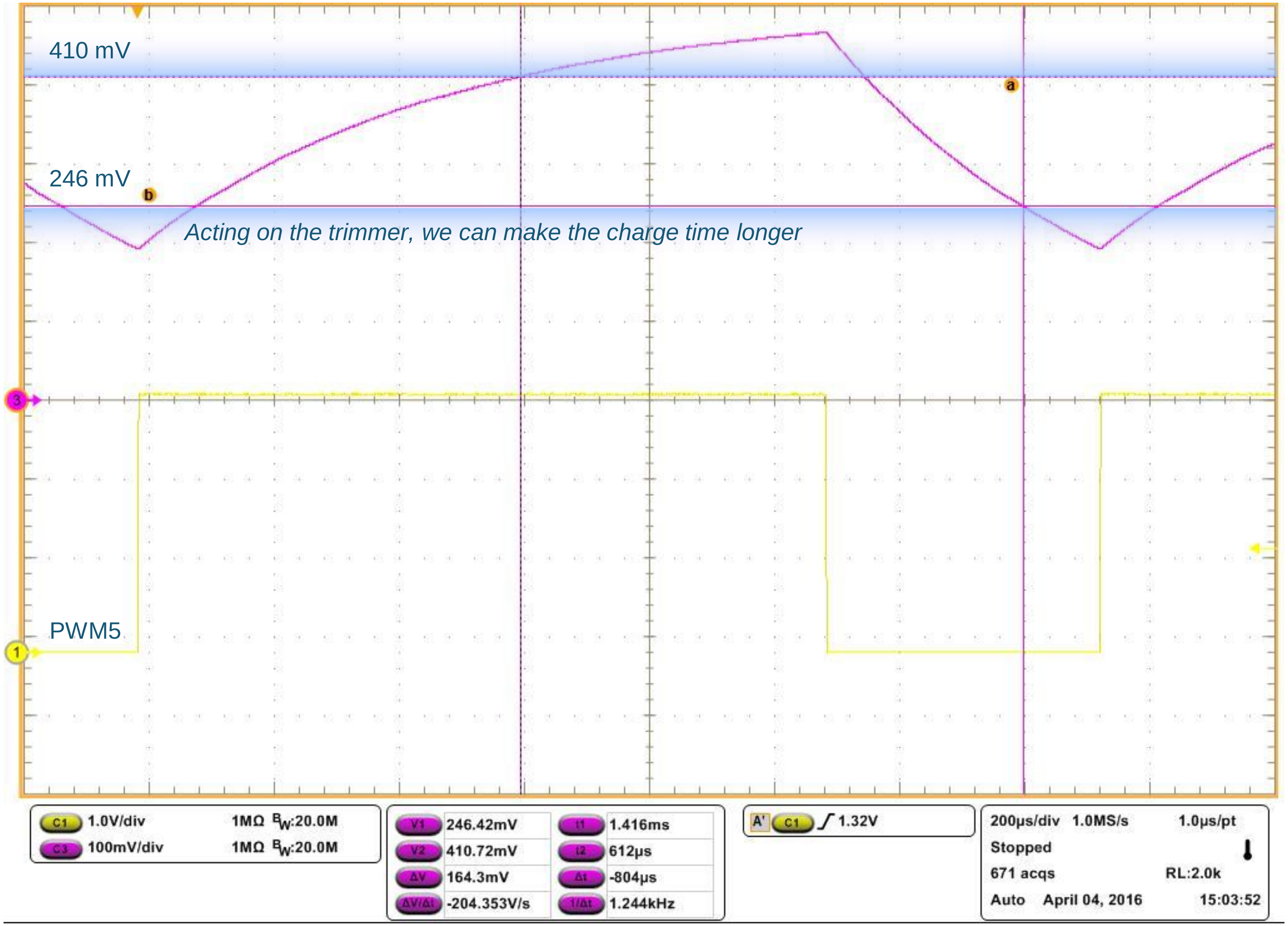
From 1▲	To 2▲	Trigger transition condition	PWM on exit	Counter reset
Idle	ON Time	Start	▔	●
ON Time	OFF Tim	9600 µs or Wait max	▁	●
OFF Time	ON Time	9600 µs or Wait min	▔	●

STNRG Example VI - waveforms



```
C1  1.0V/div     1MΩ  BW:20.0M     V1     246.42mV     t1   968μs     A' C1 ⨍ 1.32V     200μs/div  1.0MS/s     1.0μs/pt
C3  100mV/div    1MΩ  BW:20.0M     V2     410.72mV     t2   370μs                       Run              Hi Res
                                   ΔV     164.3mV      Δt   -598μs                      865 acqs                RL:2.0k
                                   ΔV/Δt  -274.749V/s  1/Δt 1.672kHz                    Auto   April 04, 2016   15:02:40
```

STNRG Example VI - waveforms



410 mV

246 mV

Acting on the trimmer, we can make the charge time longer

PWM5

```
C1  1.0V/div      1MΩ  BW:20.0M      V1    246.42mV     t1    1.416ms     A' C1 ⨍ 1.32V     200µs/div  1.0MS/s    1.0µs/pt
C3  100mV/div     1MΩ  BW:20.0M      V2    410.72mV     t2    612µs                          Stopped
                                     ΔV    164.3mV      Δt    -804µs                         671 acqs               RL:2.0k
                                     ΔV/Δt -204.353V/s  1/Δt  1.244kHz                       Auto    April 04, 2016  15:03:52
```

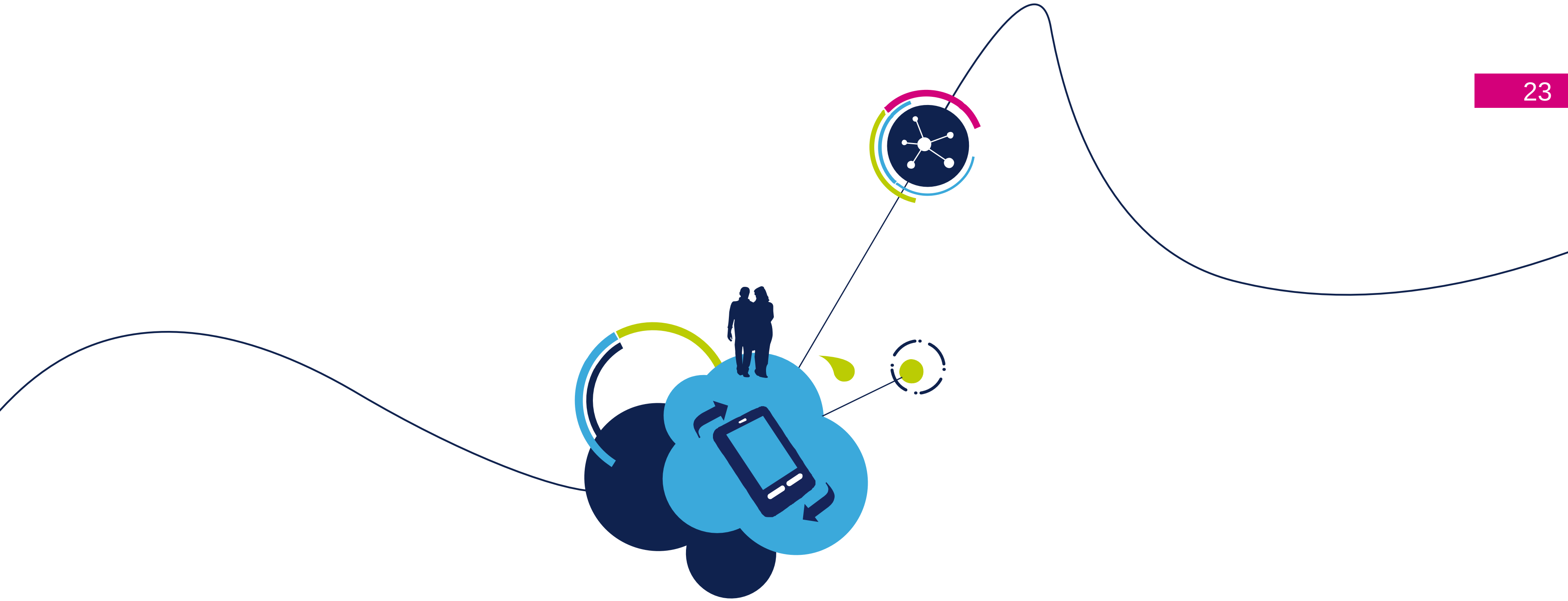
STNRG Example VI - firmware

- On the main:
  - Initialize CLK, PLL, SMED, DAC, CPP, HYST
  - Start the SMED5
  - HALT the STM8 processor

- SMED run autonomy, without FW modification

- Possible to use the SMED configurator and serial connection without compilation

- Exercise
  - SMED5 is configured in a way that feedback from comparators makes the PWM5 toggle. If the comparator feedback is missed, it gets stuck. Try modifying the state machine so that it restarts after a max_time it gets stuck in a state.

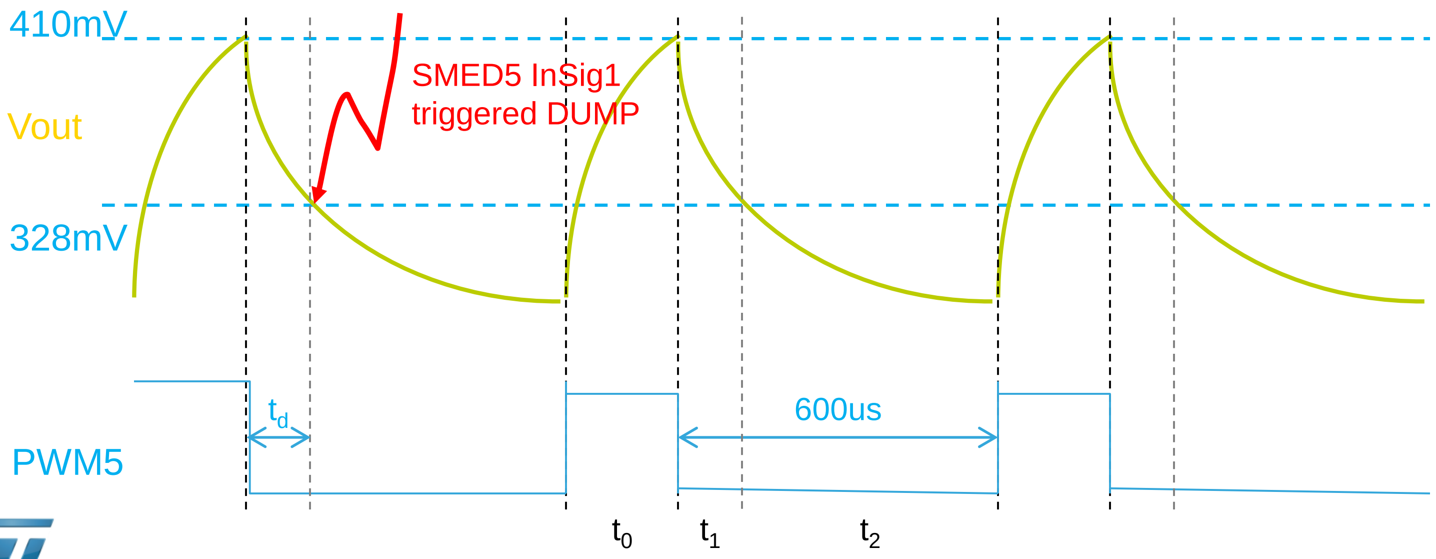


STNRG Example VII

STEVAL-ISA164V1 STNRG388A Dev board

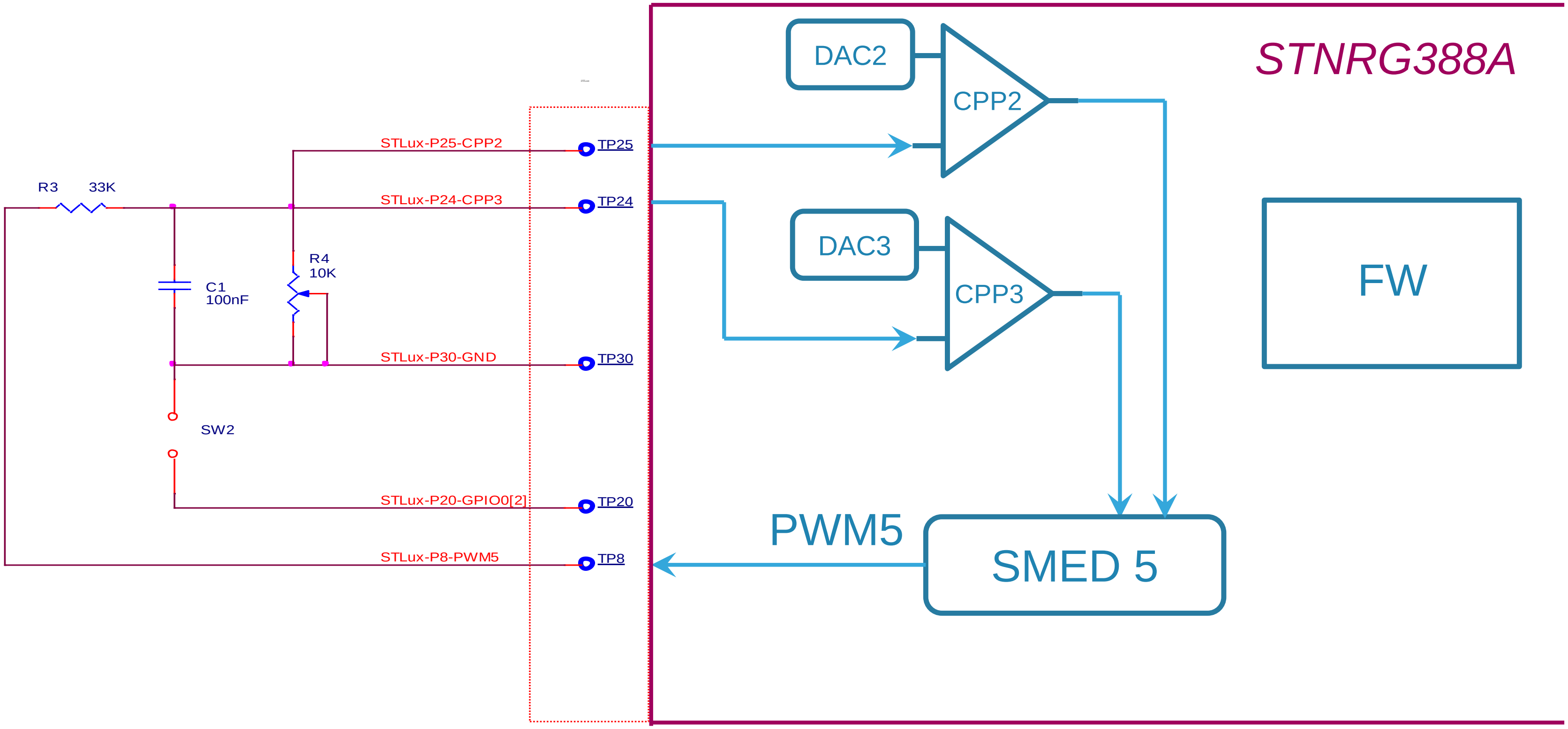
STNRG Example VII - specification

- Vout must be kept under 410mV:
  - PWM5 is turned OFF when voltage hits the high threshold -> DAC2/CPP2
- When Vout falls below 328mV a dump is triggered:
  - A dump is triggered when voltage reaches the low threshold -> DAC3/CPP3
  - PWM5 is turned ON after 600 us
- Orthogonal PWM2 and PWM3 must be generated:
  - Duty cycle must be symmetric with ON/OFF time equal to the sensed dump-time t_d

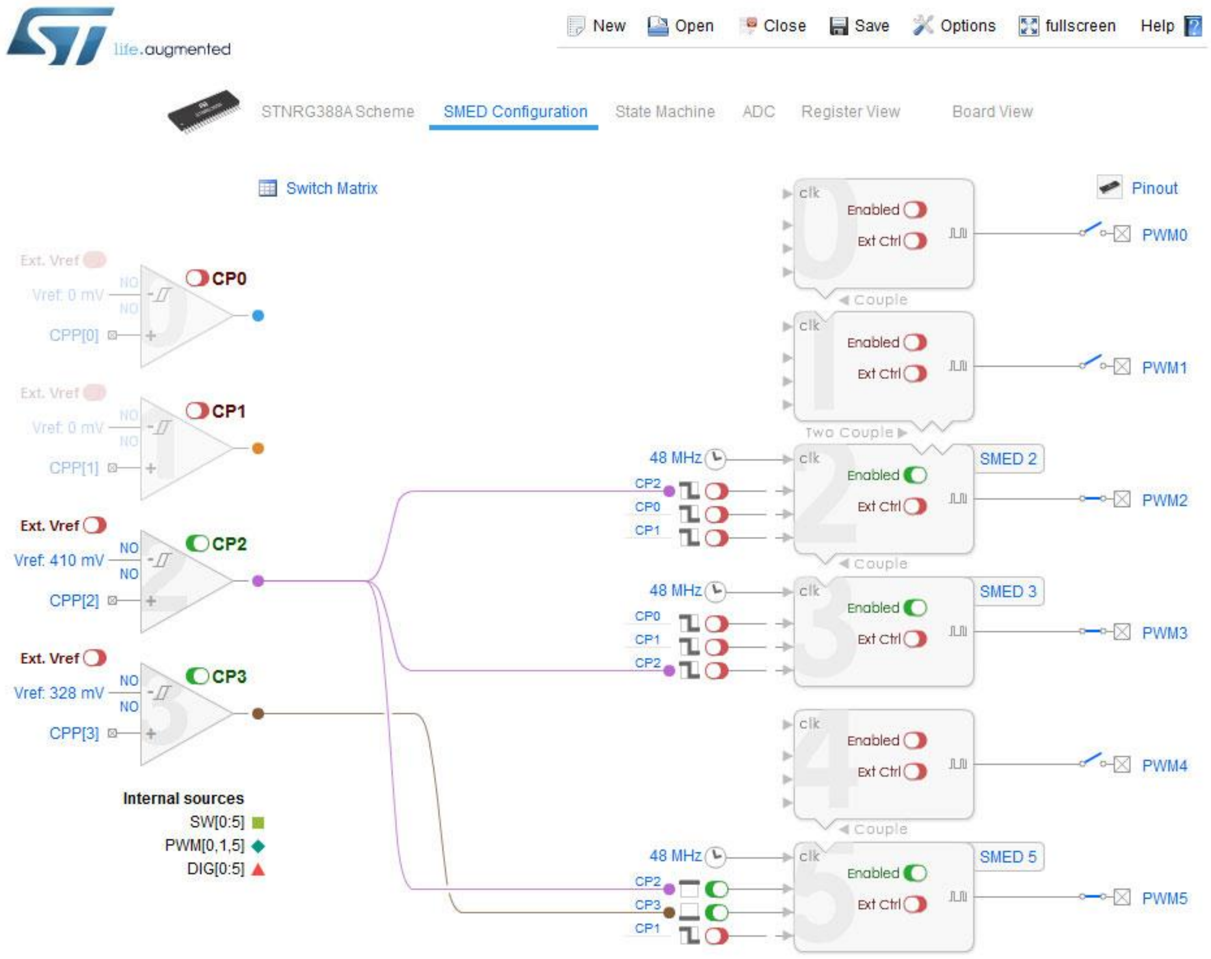


STNRG Example VII - hardware

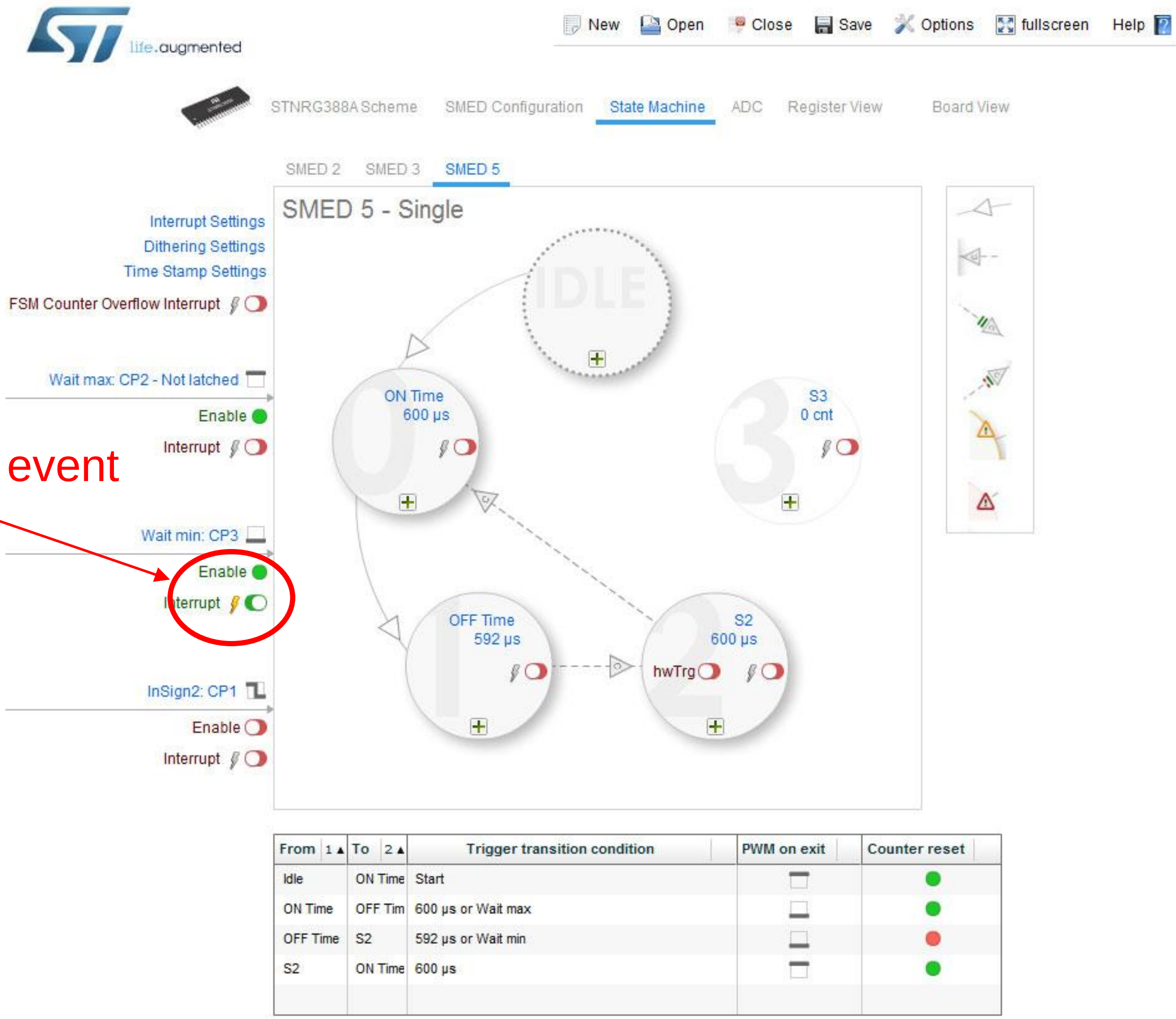
- Creating a PWM signal as a function of two comparators
- PWM5 variable frequency to get an analog output voltage between CPP2 and CPP3 references



STLUX Example VII - firmware



STLUX Example VII - firmware



SMED 5 - Single

Interrupt Settings

Dithering Settings

Time Stamp Settings

FSM Counter Overflow Interrupt ○

Wait max: CP2 - Not latched

Enable ●

Interrupt ○

Wait min: CP3

Enable ●

Interrupt ●

InSign2: CP1

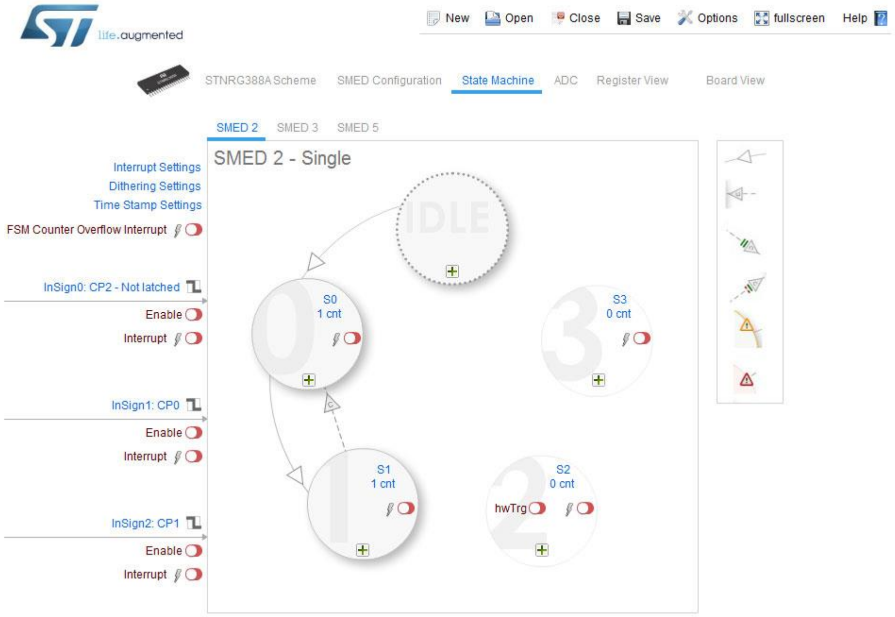
Enable ○

Interrupt ○

Interrupt on event

From 1▲	To 2▲	Trigger transition condition	PWM on exit	Counter reset
Idle	ON Time	Start	▔	●
ON Time	OFF Tim	600 µs or Wait max	▁	●
OFF Time	S2	592 µs or Wait min	▁	●
S2	ON Time	600 µs	▔	●

STLUX Example VII - firmware



SMED 2 - Single

Interrupt Settings
Dithering Settings
Time Stamp Settings
FSM Counter Overflow Interrupt

InSign0: CP2 - Not latched
Enable
Interrupt

InSign1: CP0
Enable
Interrupt

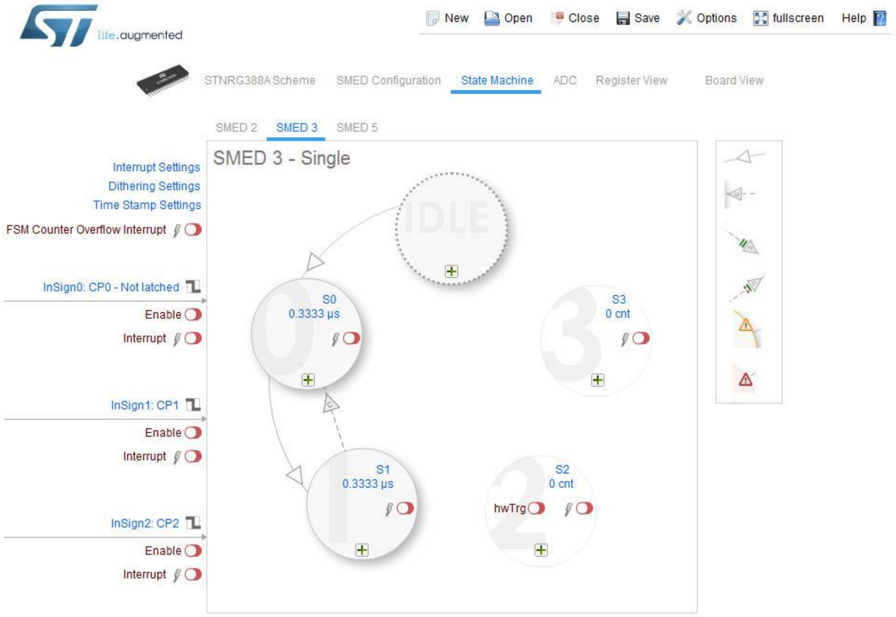
InSign2: CP1
Enable
Interrupt

S0
1 cnt

S1
1 cnt

S2
0 cnt
hwTrg

S3
0 cnt



SMED 3 - Single

Interrupt Settings
Dithering Settings
Time Stamp Settings
FSM Counter Overflow Interrupt

InSign0: CP0 - Not latched
Enable
Interrupt

InSign1: CP1
Enable
Interrupt

InSign2: CP2
Enable
Interrupt

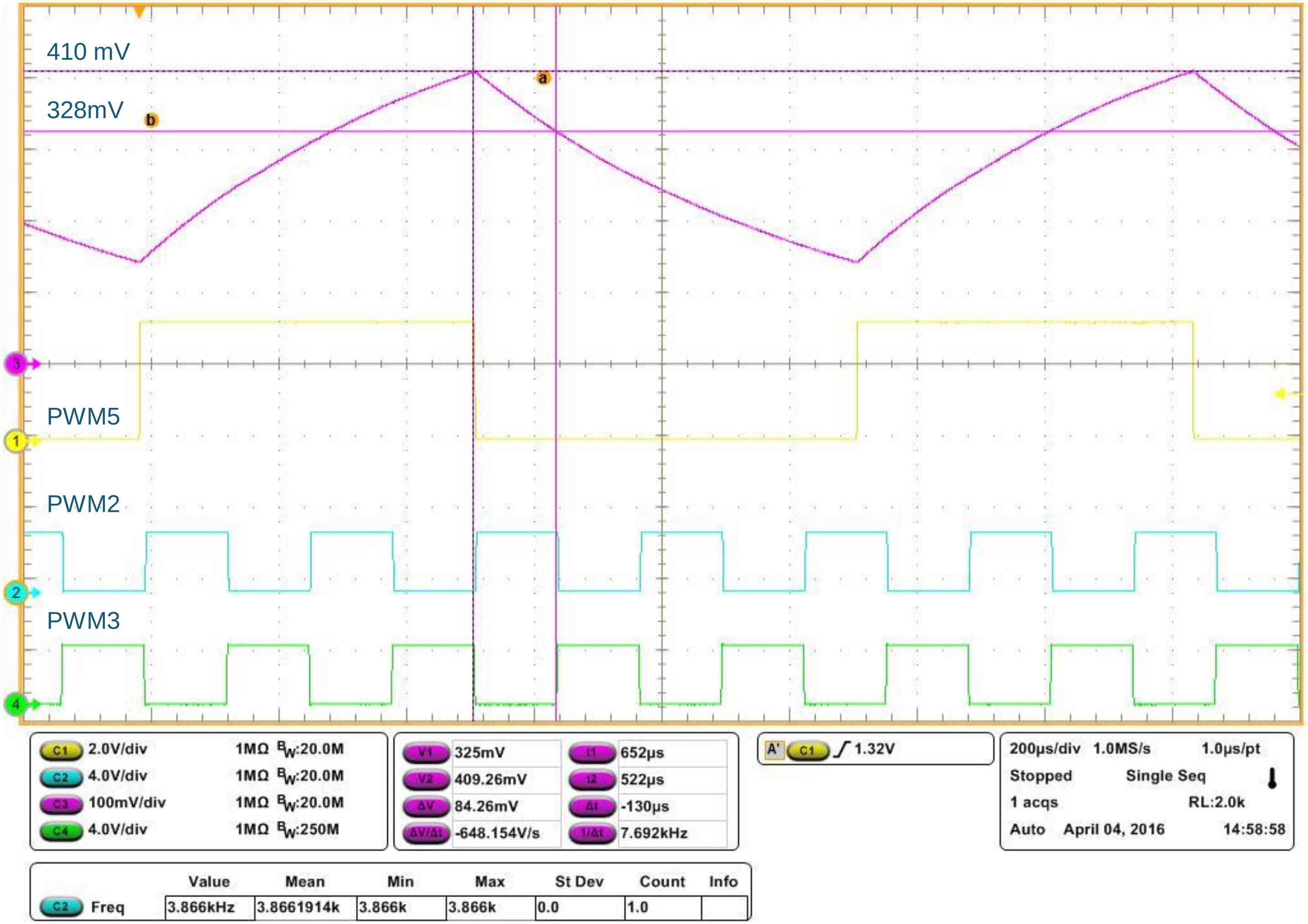
S0
0.3333 µs

S1
0.3333 µs

S2
0 cnt
hwTrg

S3
0 cnt

STLUX Example VII - waveforms



C1	2.0V/div	1MΩ	B_W:20.0M
C2	4.0V/div	1MΩ	B_W:20.0M
C3	100mV/div	1MΩ	B_W:20.0M
C4	4.0V/div	1MΩ	B_W:250M

V1	325mV	t1	652µs
V2	409.26mV	t2	522µs
ΔV	84.26mV	Δt	-130µs
ΔV/Δt	-648.154V/s	1/Δt	7.692kHz

A' C1 ⨍ 1.32V

200µs/div 1.0MS/s 1.0µs/pt
Stopped Single Seq
1 acqs RL:2.0k
Auto April 04, 2016 14:58:58

		Value	Mean	Min	Max	St Dev	Count	Info
C2	Freq	3.866kHz	3.8661914k	3.866k	3.866k	0.0	1.0	

STNRG Example VII - firmware

- On the main:
  - Initialize CLK, PLL, SMED, DAC, CPP, HYST
  - Start the SMED5, then wait for interrupt
  - Start the SMED2 and SMED3
  - HALT the STM8 processor

- In the SMED5 InSig1 (WaitMin) Input interrupt:
  - DUMP time t_d
  - Set SMED2 time $t_0 = t_1 = t_d$
  - Set SMED3 time $t_0 = t_1 = t_d$
  - Validate SMED2 time t_0 and t_1
  - Validate SMED3 time t_0 and t_1

- Necessary to compile and download using SWIM interfaces

Thank you very much for your attention