

# **Introduction to Nucleo-64 platform**







## The company



- Intecs Italian company with activities in:
  - Defense
  - Railway
  - Aerospace
  - Traffic Control & Surveillance
  - Automotive
  - Telecom
- Approx. 500 employees over 6 cities in Italy (not only)
- Purpose of these classes: getting familiar with the world of embedded systems and microcontrollers.





- PhD in computer engineering @diag
- Focus on wireless sensor networks and low power devices.
- Since 2012 partner of Wsense (university spin-off): hw + microcontroller software development.
- In Intecs since October 2016: head of HW Lab in Rome, embedded sw developer/hw designer.









DIL24 socket

### **Sensor expansion board:**

# X-Nucleo-IKS01A2



Intecs Solutions 2018 – Roma-Genova-Milano-Napoli-Pisa-Torino





### **Bluetooth expansion board:**

# X-Nucleo-IDB05A1







#### Framework, IDE & tools



intecs Solutions

the Brainware company

STM CubeMX



Stm32CubeF4

Law Street					It # 20335
Charlenne () - () (	A mean of a second seco			-	
	I have \$1.5 to \$1.000 Character	-			19.00
	beater -	Banara 1	-	-	tur.

System Workbench 4



### **The Microcontroller**











Intecs Solutions 2018–Roma-Genova-Milano-Napoli-Pisa-Torino

VOCINEF, ADC

16 analog inputs

H-00

- 200

ADC1 #





DM ID, VIBUE, BOP

100 - 12 to 34 V VSS POR ON

VODA VISA

VALUE AND PARTY

OBCID IN

FLAME OUT STAMP1



Intecs Solutions 2018–Roma-Genova-Milano-Napoli-Pisa-Torino

VOCINEF, ADC

16 analog inputs

re sensor -342 A2C1 #

- 200







VOCINEF, ADC

16 analog inputs

H-00

- 200

ADC1 #







Intecs Solutions 2018–Roma-Genova-Milano-Napoli-Pisa-Torino

VOCINEF, ADC

16 analog inputs

H-00

- 200

ADC1 #







VOCINEF, ADC

16 analog inputs

re sensor -545 ADC1 #

- 200

















DM ID, VIBUE, BOP

100 - 12 to 34 V VSS POR ON

VODA VISA

VALUE AND A DAY OF TAXABLE PARTY.

MODULO, MECHO AK, SOUCK MODULO, MECHO AK, SOUCK MODULO, MECHO AK, SOUCK MODULO, MECHO AK, SOUCK MODULO, SOL MECHO

10. 104. 5MBA = 47

SCI. SCA. SMEA = A

Decia N OBCIR OUT

FLAME OUT STAMP1

10 + Carroll, 1 - Carroll, 1 -

TIMS 100 + charvels, ETR as AF TIME 100 + charvels, ETR as AF TIME 120 + charvels

USANT CTL ATT MA

II.

USB OTG FS 8

PV0

4.1040-0

W06 104

of the laws

RTC

TIM2

TIME

SP2(252

SP3/(253

OC1/SMOUS

I2C2/5MBUS

QC3/SMBUS



Intecs Solutions 2018–Roma-Genova-Milano-Napoli-Pisa-Torino

VOCINEF, ADC

16 analog inputs

H-00

- 201

ADC1 #







Intecs Solutions 2018–Roma-Genova-Milano-Napoli-Pisa-Torino

VOCINEF, ADC

16 analog inputs

H-00

- 201

ADC1 #







Intecs Solutions 2018–Roma-Genova-Milano-Napoli-Pisa-Torino

VECKEF ADD

16 analog inputs

ADC1 #

10





- Cortex M-4
  - Armv7-m architecture: Harvard architecture, 32-bit architecture (internal registers, data path, bus interface)
  - Thumb-2 instruction set (16/32 instructions)
  - Unified memory space 4GB
  - On-chip bus interfaces based on ARM AMBA
  - NVIC controller with priority levels (12 clock cycles)
  - Systick timer
  - Optimized for power consumption (alternatives: Cortex R or Cortex A)
  - Optional advanced debug features and MPU





### Address space: 4GB, little/big endian







- Systick Timer
  - Part of the NVIC, 24-bit decrement timer
  - Sourced from a reference clock source (typ. on-chip)
  - Has its own exception hanlder
  - Can be used as system clock for an OS (task management, context switch)
  - Used for portability





- Power consumption:
  - Various sleep modes available
  - Commands: Wait For Event (WFE) / Wait For Interrupt (WFI)
  - Code stops running
  - Based on the sleep mode, clock signals can selectively be turned off:
    - Deeper sleep mode -> less peripherals running
    - Deeper sleep mode -> higher wakeup time
    - Deeper sleep mode -> less wake-up sources





- Clock Sources:
  - External 4-26 Mhz crystal osc. (HSE)
  - Internal 16Mhz factory-trimmed RC (HSI16)
  - Internal 32 Khz low power RC (LSI)
  - External 32 Khz crystal for RTC (LSE)
  - System PLL (uses HSE,HSI16) up to 84Mhz
- At startup, the MCU uses HSI at 16Mhz
- Clock sources managed by Reset and Clock Control (RCC) module



# Stm32L476 Lookup



- Peripherals:
  - 11 x Timers
    - 6 x 16bit low power
    - 2 x 32bit
    - 2 x Watchdogs
    - 1 x Systick timer
  - 1 x RTC
  - 1 x ADC 12 bit

- •2 x SAI Interfaces
- •3 x I2C
- •3 x USART
- •4 x SPI (+ I2S)
- •1 x DMA 16 ch.
- •1 x SDIO
- •1 x USB OTG FS
- •81 x GPIO





# **Getting Started with CubeMX**





- Configuration tool:
  - Clock sources
  - Peripherals
  - Pinout
  - Middlewares
- Code generation:
  - IDE support







## **Usage Example: Clock and Timer 1 configuration**

- Step 1:
  - Launch CubeMX
  - Select "New Project"
  - Choose "Board Selector"
  - Vendor "ST Microelectronics"
  - Type of Board "Nucleo 64'
  - MCU Series "Stm32F4"
  - Select "Nucleo-F401RE"
  - Double click on it

No bear for block	and the ball	ri tirkari			G101100
Constant profession	that before	1.00			
Anglesis/Senadors brinds		Bardinist ( Bara			and the second second
Arghesis/Greaters 8		Tate	Advena	80	a
kan			NC 8-MIN		Example        1000000000000000000000000000000000000





## **Usage Example: Clock and LPTimer 1 configuration**

- Step 2:
  - From "Pinout" tab
  - Expand "RCC"
  - Select "Crystal/Ceramic resonator" in Low Speed Clock (LSE)
  - This will enable external 32Khz crystal of the Nucleo Board





## **Usage Example: Clock and Timer 1 configuration**

œ

- Step 3:
  - From "Pinout" tab
  - Expand "TIM1"
  - Select "Internal Clock" as clock source

SP13				
SYS				
🖲 TIM1				
Slave Mod	le Dis	able		$\sim$
Trigger So	ource	Disab	e	~
Clock Sou	rce	Intern	al Clock	~
Channel1	Disat	ole		~
-Channel2	Disable		~	
-Channel3	Disable		~	
Channel4	Disat	ole		~
Combined	Chan	nels (	isable	$\sim$
- Activa	te-Bri	eak-Ing	out	
- Use E	TR as	Cleari	ng Source	
- XOR a	octivat	tion		

USART_TX	PC1 PC2 VSS PA0 PA1 PA2	USART_RX PA3 - 4





## **Usage Example: Clock and Timer 1 configuration**

- Step 4:
  - From "Clock Configuration" tab
  - Leave HSI@84Mhz in System Clock Mux



Intecs Solutions 2018–Roma-Genova-Milano-Napoli-Pisa-Torino







## **Usage Example: Clock and Timer 1 configuration**

- Step 5:
  - Q: Which bus is connected to TIM1?
  - Annotate its frequency





## **Usage Example: Clock and Timer 1 configuration**

### Step 7:

•From "Configuration" tab

•Check that peripherals and clocks are set correctly

•Double click on TIM1, select counter period to be 65535

•Q: What prescaler and division should we set for 1ms tick timer?

•NVIC settings enable TIM1 update interrupt

Multimedia	Connectivity	Analog	System	Control
	USART2			TIM1





### **Usage Example: Code Generation**

- Step 1:
  - Click on "Project" -> "Settings"
  - In "Project" tab
  - Set a project name
  - Select SW4STM32 IDE
  - Check that MCU and
    Firmware package are correct

rioject betongs			
oject Code Generator Advanced Set	tings		
Project Settings			
Project Name			
StartupTest2			
Project Location			
C:\Users\ucole\STMicro\Projects			
Toolchain Folder Location			
C:\Users\ucole\STMicro\Projects\Star	tupTest2\		
Toolchain / IDE			
SW4STM32	~	Generate Under Root	
Minimum Stack Size 0x400	=		
Mcu and Firmware Package			
Mcu Reference			
STM32F401RETx			
Firmware Package Name and Version			
STM32Cube FW_F4 V1.19.0			
Use Default Firmware Location			
C:/Users/ucole/STM32Cube/Reposito	ry/STM32Cube	FW_F4_V1.19.0	Browse





### **Usage Example: Code Generation**

- Step 2:
  - In "Code Generator" tab
  - Select "Generate peripheral initialization..."
  - Keep other options unchanged
  - Click on "OK"

Project Settings		>
Project Code Generator Advanced Settings		
STM32Cube Firmware Library Package		
O Copy all used libraries into the project folder		
Copy only the necessary library files		
Add necessary library files as reference in the toolchain project configuration.	iguration file	
Generated files		
Generate peripheral initialization as a pair of '.c/.h' files per peripher	ral	
Backup previously generated files when re-generating		
Keep User Code when re-generating		
Delete previously generated files when not re-generated		
HAL Settings		
Set all free pins as analog (to optimize the power consumption)		
Enable Full Assert		
Template Settings		
Select a template to generate customized code	Settings	
	Ok Can	el





### **Usage Example: Code Generation**

- Step 3:
  - Click on "Project" -> "Generate Code"
  - Wait the end of the execution
  - You can now import the project on System Workbench
    4

Project Settings		×
Project Code Generator Adva	nced Settings	
STM32Cube Firmware Library	/ Package	
Copy all used libraries int	to the project folder	
Copy only the necessary	library files	
Add necessary library file	s as reference in the toolchain project configuration	file
Generated files		
Generate peripheral initia	lization as a pair of '.c/.h' files per peripheral	
Backup previously genera	ted files when re-generating	
Keep User Code when re	generating	
Delete previously generat	ted files when not re-generated	
HAL Settings		
Set all free pins as analo	g (to optimize the power consumption)	
Enable Full Assert		
Template Settings		
Select a template to generate	e customized code	Settings
	04	Cancel





# System Workbench 4

## Importing project and debugging




### Importing project generated with CubeMX

- Step 1:
  - Launch SW4STM32
  - In "File" menu click on "import..."
  - In "General", select "Existing Project into Workspace"
  - Select the root folder generated with CubeMx
  - Keep default options and click finish

Import					o x
Import Projects Select a directory to sea	rch for existing	Eclipse projects.			
Select root directory:     Select archive file:     Projects:	C:\Users\ucole	\STMicro\Projec	ts\StartupTest	× (	Browse_ Browse_
Options Options Search for nested pro Copy projects into w Hide projects that all Working sets Add project to work Working sets:	ers\ucole\STMi ojects orkspace ready exist in th ing sets	e workspace	tupTest)	>	Select All Deselect All Refresh
0	< Back	Next >	Finish		Cancel







### Importing project generated with CubeMX

- Step 3:
  - Plug the nucleo
  - Right click on the project and select "Debug as"
  - When prompted to switch in debug view click yes (check the "keep option" if you don't want to repeat this step each time)
  - The code will halt on HAL\_Init()
  - Click on "step over" or "step into" to get familiar with the IDE in debugging mode
  - You can click on "Resume" if you want your code to freely run (but it won't do anything since it's empty <sup>(2)</sup>)





#### **Important files: the linker script**

- In project explorer: STM32F401RETx\_FLASH.ld
- Where to find program and data memory (RAM,FLASH) w.r.t. the linear memory map of the MCU
- What to put inside each area (e.g., .isr\_vector, .text and constant data in flash, .data and .bss in ram etc...)

```
32/* Entry Point */
33 ENTRY(Reset Handler)
34
35 /* Highest address of the user mode stack */
36_estack = 0x20018000; /* end of RAM */
37/* Generate a link error if heap and stack don't fit into RAM */
38 Min Heap Size = 0x200;
                              /* required amount of heap */
39 Min Stack Size = 0x400; /* required amount of stack */
40
41/* Specify the memory areas */
42 MEMORY
43 {
                 : ORIGIN = 0x20000000, LENGTH = 96K
44 RAM (xrw)
45 FLASH (rx)
                 : ORIGIN = 0x8000000, LENGTH = 512K
46 }
47
48 /* Define output sections */
49 SECTIONS
50 {
51 /* The startup code goes first into FLASH */
    .isr_vector :
52
53 {
54
      = ALIGN(4);
      KEEP(*(.isr_vector)) /* Startup code */
55
56
      . = ALIGN(4);
57
    } >FLASH
```





#### Important files: the linker script

• Quick recall on memory segments:



Automatic variables, returned address ...

```
int dummy_func(int arg1){
    int arg2;
    if(arg2 > arg1) return arg2;
    else return arg1;
}
```

Dynamic memory allocation (e.g., malloc,...)

Global or static variables initialized to 0 or not explicitly initialized Global or static variable with pre-defined value and that can be modified Read only data (e.g., code)





#### Important files: the linker script

• Quick recall on memory segments:







#### Important files: the linker script

• Quick recall on memory segments:







#### **Important files: the linker script**

• Quick recall on memory segments:







#### Important files: the linker script

• Quick recall on memory segments:







#### Important files: the startup file

- startup/startup\_stm32f401xe.s
- Written in assembly, it holds the reset handler (first code to be executed) and the vector table

```
76
77
      .section
                  .text.Reset Handler
              Reset Handler
78
      .weak
      .type Reset Handler, %function
79
80 Reset Handler:
    ldr sp, = estack /* Atollic update: set stack pointer */
81
82
83/* Copy the data segment initializers from flash to SRAM */
    movs r1, #0
84
   b LoopCopyDataInit
85
86
87 CopyDataInit:
    ldr r3, = sidata
88
   ldr r3, [r3, r1]
89
      str r3, [r0, r1]
90
            r1, r1, #4
91
      adds
92
93 LoopCopyDataInit:
94
      ldr r0, = sdata
95
      ldr r3, = edata
              r2, r0, r1
96
      adds
97
      cmp r2, r3
0.0
      has ConvDataInit
```





#### Important files: the system file

- Src/system\_stm32f4xx.c
- SystemInit function for clock and vector table initialization
- Other clock utilities...

g main.c	Is startup_stm32I4/bx	I stm32Hxx_hal.c	Istm32Hox_hal_msp.c	G STM32L4/6RGTX_FLAS_	Lsi system_stm32Hxx.c ≈
198 <b>0 vo</b>	id SystemInit(void)				
199 {					
200 /	/* FPU settings				*/
201 4	if (FPU_PRESENT	== 1) && (FPU	_USED == 1)		
202	SCB->CPACR  = ((3	BUL << 10*2) (3U	L << 11*2)); /* se	t CP10 and CP11 Full	Access */
203	lendif				
204 /	/* Reset the RCC cl	lock configuration	on to the default r	eset state	*/
205 /	/* Set MSION bit */	/			
206 1	RCC->CR  = RCC_CR_M	(SION;			
207					
208 /	/* Reset CFGR regis	ster */			
209 1	$RCC \rightarrow CFGR = 0x00000$	;0000			
210					
211 /	/* Reset HSEON, CSS	SON , HSION, and	PLLON bits */		
212 1	RCC->CR 6= (uint32	t) 0xEAF6FFFF;			
213					
214 /	/* Reset PLLCFGR re	gister */			
215 1	RCC->PLLCFGR = 0x00	0001000;			
216					
217	/* Reset HSEBYP bit	: */			
218 1	RCC->CR 6= (uint32	t) 0xFFFBFFFF;			
219					
220	/* Disable all inte	errupts */			
221 1	$RCC \rightarrow CIER = 0 \times 00000$	0000;			
222					
223	/* Configure the Ve	ector Table loca	tion add offset add	iress	*/
224 11	fdef VECT TAB SRAM				
225 3	SCB->VTOR = SRAM BA	SE   VECT TAB O	FFSET; /* Vector Ta	ble Relocation in In	ternal SRAM */
226 🚛	lse				
227	SCB->VTOR = FLASH I	ASE   VECT TAB	OFFSET: /* Vector T	able Relocation in I	nternal FLASH */
228 #er	ndif				
229 }					





### Important files: peripherals initialization

- Src/tim.c
- Src/gpio.c
- MX\_[peripheral]\_init: high-level init
- HAL\_[peripheral]\_init: low-level init

#### **Important files: Hardware Abstraction Layer drivers**

• Drivers/STM32L4xx\_HAL\_Driver/stm32l4xx\_hal\_[peripheral].c





#### Important files: interrupt management

- Src/stm32f4xx\_it.c
- Glue code between HAL peripheral and your code:
  - Interrupt handlers in the vector table are not directly defined in the HAL layer.
  - User can define them in the stm32f4xx\_it.c file and call the HAL\_Handler in it.







#### Launching and debugging

- Plug the Stm32F401 board
- Right click on the project and select "Debug as..." and select "Ac6 Stm32 C/C++ Application"
- This will trigger a recompilation (but it's already done)
- Then the board will be programmed and the mcu reset
- The program will halt by default at the beginning of the main function

0+0.012 N.H.0.014 N.N.010 P.0.10 P.0.	Sec. 20 (1997) 1997 1997	0.00				GARAGES B BOCH S Debug
b Detrop 11  • Consequences (Sold SPACE Descripting) d - demonstrated, est value fringerood d - Consequences (Ad SPACE Descripting) • Consequences (Ad SPACE Descripting)  • Consequences (Ad SPACE Descripting)  • Consequences (Ad SPACE Descripting)  • Consequences (Ad SPACE Descripting) • Consequences (Ad SPACE Descripting)  • Consequences (Ad SPACE Descripting) • Conseque	$ \pi_{ij}  =  \Psi_{ij}  = 0$ water the Property MyBerger (Dev.) water the Property MyBerg (Dev.) (Dev.)	Vanis No	exeptions 12 in a	e lengorary	O fagunes Tá Spain Mitholais	x <b>% 2</b> 0 <b>x   x = 6</b> 0
4	3					
2 Panel II 3 monta_productions 21 /* Unit Color Ray 0 */ 22 23 milest main/rotal 24 [ 25 /* Unit Color Ray 0 */ 25 /* Unit Color Ray 1 */ 26 /* Unit Color Ray 1 */ 27 /* Unit Color Ray 1 */ 28 /* Ray 1 de la peripherala, faitision 29 /* Unit Color Ray 1 */ 21 /* Color Laws 1 de la peripherala, faitision 21 /* Color Laws 1 de la peripherala, faitision 23 /* Color Laws 1 de la peripherala, faitision 24 /* Color Laws 1 de la peripherala, faitision 24 /* Color Laws 1 de la peripherala, faitision 29 /* Color Laws 1 de la peripherala, faitision 20 /* Color Laws 1 de la peripherala, faitision 21 /* Color Laws 1 de la peripherala, faitision 24 /* Color Laws 1 de la peripherala, faitision 25 /* Color Laws 1 de la peripherala, faitision 26 /* Color Laws 1 de la peripherala, faitision 27 /* Color Laws 1 de la peripherala, faitision 28 /* Color Laws 1 de la peripherala, faitision 29 /* Color Laws 1 de la peripherala, faitision 20 /* Color Laws 1 de la peripherala, faitision 29 /* Color Laws 1 de la peripherala, faitision 20 /* Color Laws 1 de la peripherala, faitision 20 /* Color Laws 1 de la peripherala, faitision 29 /* Color Laws 1 de la peripherala, faitision 20 /* Color La	o the Fisch interface and the	tentisk.	Convertience     C	Bill Bill Bill Bill Bill Bill Bill Bill	D Manony Report Desconting (MAL) Dealth Desconting (MAL) Dealth Desconting (MAL) Dealth Desconting (MAL) Dealth Desconting (MAL) Desconting (MAL) Des	(040*0000040* 0)(0 0 0)(0 0)
©Consule II divis 2 relations O lancation B throady Secularized Phili Designed (American relations Marching) the current language does not match t Sequency breakpoint 1, main 0 at ,/Bro/main The breakpoint matter 2.	ipepinton,452,7479044,and2,and2, Nis fram. .ci73	dit, fek (pingring fin	ef.nc, energino	0.411.004	and2_1128.2014TO454T7Aoo64complection	K & B B D D D D D D D D D D D D D D D D D





#### **Useful views during a breakpoint**

•Variables: shows (some?) global and automatic variables with their values

•			in a creat stored
🕬 Variables 🕄 💁 Breakpoints 🖽 Re	gisters 🔳 I/O Registers 👼 Signals 🛋 Modules		셀 ≪ 등   위 복 %   13 번 ▼ □ □
Name	Туре	Value	
60- tmpreg	volatile uint32_t	0	
			~
1			~
·			/





#### Useful views during a breakpoint

•Registers: shows the values of all the MCU registers

$\Leftrightarrow \bullet \bullet \bullet$		Quick Access 🔡 🗟 C/C++ 🎄 Debug	
🕬 Variables 💁 Breakpoints 🕮 Registers 😂 📾 I/O Reg	isters 🛼 Signals 🛋 Modules	2 4 B B V P B	
Name	Value	Description	•
✓ ₩ General Registers		General Purpose and FPU Register Group	
1111 r0	0x0 (Hex)		
200 r1	0x0 (Hex)		
1111 r2	0x0 (Hex)		
IIII r3	0xe000ed14 (Hex)		
3212 r4	0	·	1
<		>	
		· · · · · · · · · · · · · · · · · · ·	
		· · · · · · · · · · · · · · · · · · ·	1
<		>	





#### **Useful views during a breakpoint**

•I/O Registers: detailed view of all peripheral registers with their offsets and values

Quick Access     B     B     C/C++     ☆ Debug						Debug	
🕬 Variables 💁 Breakpoints 💷 Registers 🖾 I/O Registers 🕮 🐱 Signals 🛋 Modules 💭 🗖							
① Double-click on register to fetch	value						÷
Register	Hex value	Binary value	Reset val	Access	Address	Description	^
✓ ₩ RCC					0x400210	Reset and clock control	
> IIII CR			0x000000		0x400210	Clock control register	
> 調評 ICSCR			0x100000		0x400210	Internal clock sources calibration	regi:
> ## CFGR			0x000000		0x400210	Clock configuration register	
> ## PLLCFGR			0x000010	READ-W	0x400210	PLL configuration register	
> 期 PLLSAI1CFGR			0x000010	READ-W	0x400210	PLLSAI1 configuration register	
> 調練 PLLSAI2CFGR			0x000010	READ-W	0x400210	PLLSAI2 configuration register	
> ## CIER			0x000000	READ-W	0x400210	Clock interrupt enable register	
<							>





#### **Useful views during a breakpoint**

•Disassembly: disassembled code on the go (not always reliable)

🗄 Outline 🔤 D	isassembly 🕸	Memory Browser	Enter location here	V 🕄 🖞 😫 🖳 📑 🖻	~
08001691:	b.n	0x8001690 <main+18></main+18>			^
	Error_Han	iler:			
08001693:	b.n	0x8001692 <error_handler></error_handler>			
	HAL_MspIn	Lt:			
• 08001694:	ldr	r3, [pc, #132] ; (0x800171c <hal_msp)< td=""><td>Init+136&gt;)</td><td></td><td></td></hal_msp)<>	Init+136>)		
08001696:	push	{r0, r1, r2, lr}			
08001698:	ldr	r2, [r3, #96] ; 0x60			
0800169a:	orr.w	r2, r2, #1			
0800169e:	str	r2, [r3, #96] ; 0x60			
080016a0:	ldr	r2, [r3, #96] ; 0x60			
080016a2:	and.w	r2, r2, #1			
080016a6:	str	r2, [sp, #0]			
080016a8:	ldr	r2, [sp, #0]			
080016aa:	ldr	r2, [r3, #88] ; 0x58			
080016ac:	orr.w	r2, r2, #268435456 ; 0x10000000			~
	<				>





#### Useful views during a breakpoint

•Memory Browser: allows to browse over the entire linear memory of the MCU

0x20000000		~	Go New Tab
)x20000000 <tr< th=""><th>aditional&gt; 🛙</th><th></th><th></th></tr<>	aditional> 🛙		
0x20000000	003D0900 0000000 0000000 0000000 0000000 00000	=	
x20000018	<b>00000000</b> 0000000 <b>0000000</b> 0000000 <b>0000000</b> 00000000		
x20000030	<b>00000000</b> 0000000 <b>0000000</b> 0000000 <b>0000000</b> 00000000		
x20000048	<b>00000000</b> 0000000 <b>0000000</b> 0000000 <b>0000000</b> 00000000		
x20000060	40022014 40022010 ED52BE00 200018B4 20018B4 20018000	@@. *Ri' '	
x20000078	080001E5 08000235 08000235 08000235 08000235 08000235	å5555	
x20000090	00000000 0000000 0000000 0000000 08000235 08000235		
x200000A8	00000000 08000235 08001721 08000235 08000235 08000235	5555	
x200000c0	08000235 08000235 08000235 08000235 08000235 08000235	55555	
x200000D8	08000235 08000235 08000235 08000235 08000235 08000235	55555	
x200000F0	08000235 08000235 08000235 08000235 08000235 08000235	55555	





#### Hands-on: Le'ts put a breakpoint on the Reset Handler...

•The first steps of the cortex-m during power on are:

- Load address 0x0 (address of the stack pointer) in the stack pointer register (MSP)
- Load address 0x4 (address of the reset handler) in the program counter (PC)

#### •After re-running "debug as":

• Follow the boot sequence up to the main

Intecs Solutions 2017–Roma-Genova-Milano-Napoli-Pisa-Torino

# Hands on: let's switch on the led (LD2)

•Just one line of code in the main.c...

User LD2: the green LED is a user LED connected to Arduino signal D13 corresponding to STM32 I/O PA5 (pin 21) or PB13 (pin 34) depending on the STM32 target. Refer to Table 11 to Table 23 when:

the I/O is HIGH value, the LED is on

/\* Configure the system clock \*/

the I/O is LOW, the LED is off

LD3 PWR: the red LED indicates that the STM32 part is powered and +5V power is available.

```
SystemClock_Config();
/* Initialize all configured peripherals */
MX_GPIO_Init();
MX_LPTIM1_Init();
/* USER CODE BEGIN 2 */
HAL_GPIO_WritePin(GPIOA,GPIO_PIN_5,GPIO_PIN_SET);
/* USER CODE_END 2 */
```



57



intecs Solutions

## System Workbench 4





#### Hands on: let's switch on the led (LD2)

•Let's have a closer look to what it is happening through the debugger...



⊖/\*\*



#### Hands on: let's toggle the led (LD2) each 500ms

•Using the HAL library and systick timer:



@brief This function handles System tick timer. void SysTick Handler(void) /\* USER CODE BEGIN SysTick IRQn 0 \*/ /\* USER CODE END SysTick\_IRQn 0 \*/ HAL IncTick(); HAL SYSTICK IRQHandler(); /\* USER CODE BEGIN SysTick IRQn 1 \*/ /\* USER CODE END SysTick IRQn 1 \*/

#### stm32l4xx\_it.c

Polls <u>uwTick</u> which increases by one at each systick interrupt (check with the debugger)

main.c





#### SW4 – CubeMX integration

Intecs Solutions 2018 – Roma-Genova-Milano-Napoli-Pisa-Torino

60





### Create a plain project

- Step 1:
  - Launch SW4STM32
  - In "File" menu click on "new"
  - Select "C project"
  - Enter the project name
  - Make sure you select "Empty Project" with "AC6" toolchain
  - Click on "Next"

•	C Project - + ×
C Project Create C project of selected type	
Project name: PervasiveClass	pace/PervasiveClass Browse
Project type:	Toolchains:
<ul> <li>CNU Autotools</li> <li>Executable</li> <li>Empty Project</li> <li>Ac6 STM32 MCU Project</li> <li>Hello World ANSI C Project</li> <li>Chared Library</li> <li>Static Library</li> <li>Makefile project</li> </ul>	Ac6 STM32 MCU GCC Cross GCC Linux GCC
Show project types and toolchai	ins only if they are supported on the platform           Next >         Cancel         Finish





### **Create a plain project**

- Step 2:
  - Select "Debug" and "Release"
  - Click on "Next"
  - Select STM32F4 as "Series"
  - Select Nucleo-F401RE as "Board"
  - Click on "Next"

eet ertri	er the n	ncu or the board ta	arget and configurations	
cu Boa	rd			
Shov Shov	v ST Dis v ST NU	covery boards ICLEO boards	Show ST EVAL boards	
Series :	STM32	2F4		~
Board :	NUCL	EO-F401RE		~
Cre	ate a ne	w custom board	Kemove this custom board	1
Mcu		STM32F401RETx		
Cara		LOEP64		
Core Package		Size 0v19000 (@)	0x20000000)	
Core Package Memory	'RAM'	2176 OX 10000 (GA		
Core		LQFP64	0x20000000)	





#### **Create a plain project**

- Step 3:
  - Select "Hardware Abstraction Layer" and keep the rest unchanged
  - Download the framework if you do not have it
  - Click on "Finish"

O No firmware	
	Don't generate startup file
Standard Peripheral Library (StdPeriph)	
Hardware Abstraction Layer (Cube HAL)	
Firmware <u>STM32Cube_FW_F4_V1.18.0</u> has bee	m found.
Download target firm	ware
See 'Firmware Installation' for settings related to	firmware installation
Extract all firmware in separate folder ()	
Add low level drivers in the project	
As sources in the application project ①	
O As static external libraries	
Additional drivers	
STM32_USB_Host_Library	
STemWin	
STM32_Audio	
STM32_USB_Device_Library	
Additional utilities and third-party utilities:	
LibJPEG	
FreeRTOS	
LwP	
FatFs	





### **Create a plain project**

- Pros:
  - All HAL drivers available
  - Board Support Packages (BSP) drivers for external peripherals (button,led on nucleo-64)
- Cons:
  - Missing peripheral and HAL setup
  - Compiles all the files (even HAL drivers that are not used)





### Merging with CubeMX

- Step 1:
  - Create a plain project in SW4
  - Create a project in CubeMX with Clock and peripheral initialized
  - Copy the peripheral configuration files (e.g. gpio.c) from src in CubeMX to src in SW4
  - Do the same with header files (e.g. gpio.h) in the inc folder

<ul> <li>         Includes     </li> </ul>	
HAL_Driver	
Utilities	
- 😅 inc	
🕨 🖻 gpio.h	
🗈 lptim.h	
Image:	
* 😂 src	
🗈 gpio.c	
lptim.c	
<ul> <li>Imain.c</li> </ul>	
Istm32l4xx_it.c	
<ul> <li>Isyscalls.c</li> </ul>	
• system_stm32l4xx.c	
Startup	
CMSIS	
🗟 LinkerScript.ld	
NUCLEO-L476RG.xml	

PervasiveClass



### Merging with CubeMX

- Step 2:
  - Copy the main.h header file from inc folder of CubeMX to inc folder of SW4
  - Replace the stm32f4xx\_it.c (and .h) file in SW4 with the one in CubeMX
  - Copy the stm32f4xx\_hal\_msp.c file from CubeMX to SW4
  - Remove the stm32f4xx\_hal\_msp\_template.c file from HAL\_Driver/Src (or rightclick on it and select Resoure Configurations->exclude from build and select Debug and Release)

<ul> <li>PervasiveClass</li> </ul>			
<ul> <li> <i>S</i> includes     </li> </ul>			
• A HAL Driver			
• 68 LBilities			
T (Bloc			
L P opio b			
- In gproun			
· ja ipum.n			
E main.h			
<ul> <li>stm32l4xx_it.h</li> </ul>			
- 😁 src			
<ul> <li>gpio.c</li> </ul>			
<ul> <li>Iptim.c</li> </ul>			
<ul> <li>Imain.c</li> </ul>			
stm32l4xx hal msp.c			
stm32l4xx, it.c			
<ul> <li>Respective</li> </ul>			
t Deustern stm32lday c			
System_sunsziekke			
· 🛃 scartup			
<ul> <li>CMSIS</li> </ul>			
LinkerScript.ld			
NUCLEO-L476RG.xml			
Stm32i4xx_hal_dfsdn Bessme		12	
Stm32l4xx_hal_dma.c			•
a stm32Hxx_hal_dma2 to import			ilobal MSP.
2 stm3dbox hal firely to opport.			
Refresh		15	
a stm32Hxx hal flash			MspInit 0 */
Stm30iexx hal gpio.e Make Targets			
Stm32Hox_hal_hash_	ions		Exclude mon build
Stm32l4xx_hal_hash.c Build Selected File(s)	)		NABLE ( ) ;
Stm32Haxx_hal_hcd.c Clean Selected Pile(s	)		developing and
Stm32Hxx_hal_i2c_er Profiling Tools			Soundhuidtause Lutions
Stm32Axx_hal_i2c.c Stmac/C++ Code Anal	lysis		1 init"/
2 stm3294xx_hal_irda.c Team			y RenoryRanagement_IR
G startifier hal lot c Basica with			sterrupt configuration
D checkflider had lotin			interrupt configuration
Properties		Alt+Enter	yluxapeFault_IRDn, 0,
a startility hil and c	68	/* SWCall IROn int	errupt configuration "
O startility hal nor c		WUMIC SECURITY	Chromenic turns, 4, 411
D startitizz hal onamo es c			
Statility hal one of	di Prob	iems 🖅 Tasks 💟 Con	sole # Properties
stm32i4xx hel opemp.c	- termin	stade Banahas all face	CTURNS Disk services I serve

66







### Merging with CubeMX

- Step 3:
  - Edit main.c of SW4 by including the imported header files (main.h, gpio.h etc...)
  - If a peripheral configuration file has a handle variable (e.g. TIM1\_HandleTypeDef), declare it as extern in the main file

🔎 m	ain.c 🖻 usart.c 🖻 main.c 🖻 stm32l4xx_hal_msp.c
38	*@file main.c
10	<pre>#include "main.h"</pre>
11	#include "lptim.h"
12	<pre>#include "gpio.h"</pre>
13	
14	<pre>#include "stm32l4xx.h"</pre>
15	<pre>#include "stm32l4xx_nucleo.h"</pre>
16	
\$17	extern LPTIM_HandleTypeDef hlptim1;
18	
19	1.1.1.1.1.1.1
200	int main(void)
21	1
22	for to be
23	TOF(;;);
24	}



the Brainware company



### Merging with CubeMX

- Step 4:
  - Copy the SystemClock\_Config and the ErrorHandler functions from the main.c of CubeMX to the main.c of SW4
  - Forward declare the copied functions
  - Copy the inner code of the main function from CubeMX to SW4

```
main.c 🛱 💽 main.c
18
   void SystemClock Config(void);
19
    void _Error_Handler(char * file, int line);
20
21
220 int main(void)
23 {
24
          /* MCU Configuration------
25
          /* Reset of all peripherals, Initializes the Flash in
26
27
          HAL Init();
28
          /* Configure the system clock */
29
          SystemClock Config();
30
31
32
         /* Initialize all configured peripherals */
33
          MX GPIO Init();
34
          MX LPTIM1 Init();
35
36
          /* USER CODE BEGIN 2 */
37
          HAL GPIO WritePin(GPIOA, GPIO PIN 5, GPIO PIN SET);
38
          /* USER CODE END 2 */
39
40
          /* Infinite loop */
41
          /* USER CODE BEGIN WHILE */
42
          while (1)
43
          /* USER CODE END WHILE */
44
45
              HAL Delay(500);
46
              HAL GPIO TogglePin(GPIOA,GPIO PIN 5);
47
          /* USER CODE BEGIN 3 */
```





### Using the BSP:

- Check the file "stm32f4xx\_nucleo.c" in Utilities/STM32F4XX\_Nucleo
- Modify the main as follows and run the code (right click->run as->Ac6):

```
/* Initialize all configured peripherals */
MX GPIO Init();
MX USART2 UART Init();
MX TIM1 Init();
/* USER CODE BEGIN 2 */
//HAL_GPIO_WritePin(GPIOA,GPIO_PIN_5,GPIO_PIN_SET);
BSP_PB_Init(BUTTON_USER, BUTTON_MODE_GPIO);
BSP_LED Init(LED2);
/* USER CODE END 2 */
/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
    if(BSP_PB_GetState(BUTTON_USER) == 1){
        BSP_LED_On(LED2);
    }
    else{
        BSP_LED_Off(LED2);
    //HAL Delay(500);
    //HAL_GPI0_TogglePin(GPI0A,GPI0_PIN_5);
/* USER CODE END WHILE */
```





### **Using the Sensor Expansion Board**

Intecs Solutions 2018 – Roma-Genova-Milano-Napoli-Pisa-Torino

70





### Importing an existing SW4 project

- Step 1:
  - In SW4 click on File->import-> existing projects into workspace
  - Click "Next"
  - Browse to the MEMS1\_V4.3.0 folder and select Projects->Multi->Examples->IKS01A2 >DataLogTerminal->SW4STM32 >STM32F401RE-Nucleo
  - Click "OK"
  - Keep everything unchanged
  - Click "Finish"

•	Import		- + ×
Import Projects			12
Select a directory to searc	h for existing Eclipse projects.		
Calact co at disactors	An and the second se	_	
Select roog directory:	/nome/pervasive/Downloads/STM32Cul	-	B[owse
<ul> <li>Select archive file:</li> </ul>		*	Browse
Projects:			
STM32L4xx-Nucleo-	DataLogTerminal (/home/pervasive/Download		Select All
			Deselect All
			Refresh
Options			
Search for nested pro	jects		
□ <u>C</u> opy projects into wo	rkspace		
<ul> <li>Hide projects that alr</li> </ul>	eady exist in the workspace		
Working sets			
Add project to work	ng sets		
Working sets:		1	Sglect
0	<back next=""> Cancel</back>		Finish





#### Importing an existing SW4 project

		GtkTerm - /dev/ttyACM0 115200-8-N-1
• Ste	en 2.	File Edit Log Configuration Control signals View ODR[0]: 104.000 Hz
•	Compile and run Open a serial port terminal Configure ttyACM0 with 8n1 and 115200bps You should see the sensor output	FS[0]: 2.000 g GYR_X[0]: 2450, GYR_Y[0]: -2170, GYR_Z[0]: -980 WHO AM I address[0]: 0x6A ODR[0]: 104.000 Hz FS[0]: 2000.000 dps ACC_X[1]: 8, ACC_Y[1]: 4, ACC_Z[1]: 998 WHO AM I address[1]: 0x33 ODR[1]: 100.000 Hz FS[1]: 2.000 g
	•	MAG_X[0]: 253, MAG_Y[0]: -78, MAG_2[0]: -724 WHO AM I address[0]: 0x40 ODR[0]: 100.000 Hz FS[0]: 50.000 Gauss
		HUM[0]: 45.50 WHO AM I address[0]: 0xBC ODR[0]: 1.000 Hz
		TEMP[0]: 27.10 WHO AM I address[0]: 0xBC ODR[0]: 1.000 Hz
		TEMP[1]: 27.39 WHO AM I address[1]: 0xB1 ODR[1]: 25.000 Hz
		PRESS[0]: 1011.86 WHO AM I address[0]: 0xB1 ODR[0]: 25.000 Hz


## **Sensor Expansion Board**



• Step 1:

•Import the DataLogTerminal example from MEMS framework home/ps/Desktop/resources/STM32CubeExpansion\_MEMS1\_V4.3.0/Projects/Examples/IKS01A1/D ataLogTerminal/SW4STM32/STM32F401RE-Nucleo)

•Build the project

•Unplug the Nucleo and install the extension board

•Plug the Nucleo and program it





#### Importing an existing SW4 project

		GtkTerm - /dev/ttyACM0 115200-8-N-1
• Ste	en 2.	File Edit Log Configuration Control signals View ODR[0]: 104.000 Hz
•	Compile and run Open a serial port terminal Configure ttyACM0 with 8n1 and 115200bps You should see the sensor output	FS[0]: 2.000 g GYR_X[0]: 2450, GYR_Y[0]: -2170, GYR_Z[0]: -980 WHO AM I address[0]: 0x6A ODR[0]: 104.000 Hz FS[0]: 2000.000 dps ACC_X[1]: 8, ACC_Y[1]: 4, ACC_Z[1]: 998 WHO AM I address[1]: 0x33 ODR[1]: 100.000 Hz FS[1]: 2.000 g
	•	MAG_X[0]: 253, MAG_Y[0]: -78, MAG_2[0]: -724 WHO AM I address[0]: 0x40 ODR[0]: 100.000 Hz FS[0]: 50.000 Gauss
		HUM[0]: 45.50 WHO AM I address[0]: 0xBC ODR[0]: 1.000 Hz
		TEMP[0]: 27.10 WHO AM I address[0]: 0xBC ODR[0]: 1.000 Hz
		TEMP[1]: 27.39 WHO AM I address[1]: 0xB1 ODR[1]: 25.000 Hz
		PRESS[0]: 1011.86 WHO AM I address[0]: 0xB1 ODR[0]: 25.000 Hz





- Unplug the board and attach the sensor board on it
- Open the STM32CubeExpansion\_MEMS1\_V4.3.0
  - Drivers
    - BSP -> You need mems drivers (Components folder), Board adaptation files (X\_NUCLEO\_IKS01A2 folder). Don't need the generic L476 BSP file (you already have it)
    - CMSIS -> Don't need: already have your CMSIS library
    - STMXXX -> Don't need: already have your HAL driver
  - Projects -> several examples
  - Utilities -> GUI program for the PC (we won't use it)







# WARNING!!!

- The imported project has several files linked in the framework folder
- It is <u>highly unrecommended</u> to change this project as the changes might affect other projects based on the same linked files
- Import the necessary file in a plain SW4 project instead.





- Step 1:
  - Import an existing project in SW4
  - Create a plain SW4 project with CubeMX settings merged in it





- Step 2:
  - Copy the BSP folder from MEMS\_V3\_0\_0 to your main project folder
  - Include the BSP folder as source folder by right-clicking on the project->properties->C/C++ General->Paths and Symbols->Source Location->Add Folder







- Step 3:
  - Filter out every c file in this folder that it is not used in the imported project: right click on a file -> resource configure->exclude from build (check the files that are used by navigating the imported project first)







- Step 4:
  - Righ-click on the project->properties->C/C++ General->Paths and Symbols->includes and selecting GNU C as "Languages"
  - Add all the relevant folders that uses headers related to the new BSP folder (every used subfolder)







- Step 5:
  - Righ-click on the project->properties->C/C++ General->Paths and Symbols->symbols and add existing symbols that are present in the imported project

Builders Configuration: Debug			[Active]			Aanage Configurations.	
C/C++ Build C/C++ General							
Code Analysis	Bincludes	# Symbols	Libraries	Elbrary Paths	Source Location	n References	
Documentation	Languages	s	Symbol		Value		Add.
File Types	GNU C		# STM32				Edit
Indexer	Assembl	Assembly	# STM32L4				
Language Mappings Paths and Symbols Preprocessor Include			# STM32L47	SRGTX			Delet
		# STM32L47	5xx		_	Expo	
			# USE_HAL_ # USE_STM3	2L4XX_NUCLEO			
Linux Tools Path	"Preprocessor Include Paths, Macros etc." property page may define additional entries						
Project References	Show b	uilt-in values					
Run/Debug Settings Task Repository	and Import	Settings	🛞 Export Se	ttings			
and the second							





• Step 6:

intecs Solutions

the Brainware company

- Import interrupt handlers from the stm32l4xx\_it.c file
- Import MCU initialization functions from the stm32l4xx\_hal\_msp.c file
- Import BSP headers in your main.h and in any other file where it is required

3⊕ * File Name : main.h 38 /* Define to prevent recursive incl	usion
38 /* Define to prevent recursive incl	usion
and the state of the second seco	
39 #ifndef MAIN H	
40 #define MAIN H	
41 /* Includes	
42	
43 /* USER CODE BEGIN Includes */	
<pre>44 #include "stm32l4xx_hal.h"</pre>	
<pre>45 #include "stm32l4xx_nucleo.h"</pre>	
46	
<pre>47 #include "x_nucleo_iks01a2.h"</pre>	
48 #include "x_nucleo_iks01a2_accelero	).h⁼
<pre>49 #include "x_nucleo_iks01a2_gyro.h"</pre>	
50 #include "x_nucleo_iks01a2_magneto.	.h=
51 #include *x_nucleo_iks01a2_pressure	2.h*
52 #include "x_nucleo_iks01a2_humidity	/.h*
53 #include "x nucleo iks01a2 temperat	ure.h"
54 /* USER CODE END Includes */	
55	
56 /* Private define	
57	
58 #define B1_Pin_GPIO_PIN_13	
59 #define B1_GPI0_Port_GPI0C	
00 #define USART TX Pin GPIO PIN 2	
ol #define USART IX GPID Port GPIDA	
02 #define USAKI KX PIN GPIO PIN 3	
63 #define USAKI KX GPIO PORT GPIDA	
64 #define LD2 PIN GPIO PIN 5	
65 #define TVS Die CDTO DTN 32	
67 #define THE COLO Part COLO	
62 #define TCK Die CDTO DTN 14	





- Step 7:
  - Copy whatever you want from the main.c file
  - Fix the remaining minor issues (check for errors, check missing files/functions, optionally initialize peripherals with CubeMX etc...)

```
🙆 main.c 🖾 🕒 main.h
 17 #include <stdio.h> /* sprintf */
  18
    #include <math.h>
                        /* trunc */
  19
  20 #include "stm32l4xx.h"
  21
    #include "stm32l4xx nucleo.h"
 22
    extern LPTIM HandleTypeDef hlptim1;
 23
 24
     extern UART HandleTypeDef huart2
 25
 26
    void SystemClock Config(void);
  27
  28
     void Error Handler(char * file, int line);
 29
  38
  31 extern int use LSI;
  32 int RTC SYNCH PREDIV;
  33
  34⊖ /* Private typedef
    /* Private define
  36 /* Private macro
  37 /* Private variables .....
  38 static volatile uint8 t acquire data enable request = 1;
 39 static volatile uint8 t acquire data disable request = 0;
 40
  41 static uint8 t acquire data enabled = θ;
  42 static uint8 t verbose
                                         = 1; /* Verbose output
  43 static RTC HandleTypeDef RtcHandle;
  44 static char dataOut[256];
 45
  46 static void *LSM6DSL X 0 handle = NULL:
  47 static void *LSM6DSL G 0 handle = NULL;
  48 static void *LSM303AGR X 0 handle = NULL;
```





# **Bluetooth Expansion Board**





- Unplug the board and attach the Bluetooth board on top of the sensor board
- Open the STM32CubeExpansion\_BLE1\_V2.8.0
  - Drivers
    - BSP -> You need board adaptation files (X-NUCLEO-IDB0xA1 folder).
       Don't need the generic L476 BSP file (you already have it)
    - CMSIS -> Don't need: already have your CMSIS library
    - STMXXX -> Don't need: already have your HAL driver
  - Middlewares -> You need the whole folder
  - Projects -> several examples
  - Utilities -> GUI program for the PC (we won't use it)







#### Importing an existing SW4 project

- Step 1:
  - In SW4 click on File->import-> existing projects into workspace
  - Click "Next"
  - Browse to the MEMS1\_V3.0.0 folder and select Projects->Multi->Applications->SensorDemo->SW4STM32->STM32L476RG-Nucleo
  - Click "OK"
  - Keep everything unchanged
  - Click "Finish"

•	Import		- + ×	
Import Projects			12	
Select a directory to sear	ch for existing Eclipse projects.		4	
Select root directory:	/home/pervasive/Downloads/STM32Cul		Browse	
Celect archive file:	maneppermanepoonnouespannaceu		Browne	
Devlocities			Blowse	
Projects:				
SensorDemo (/home	Select All			
			Deselect All	
			Rgfresh	
Options				
Search for nested pro	ojects			
□ <u>C</u> opy projects into we	orkspace			
<ul> <li>Hide projects that alr</li> </ul>	eady exist in the workspace			
Working sets				
Add project to work	ing sets			
Working sets:			Sglect	
(?)	«Back Next» Cancel		Finish	
	concer		1111211	





#### Importing an existing SW4 project

- Step 2:
  - Compile and run the example application
  - On your Android or iOS smartphone install "blueNRG" from the store
  - Launch the phone application and bind the Bluetooth board
  - You should see a cube that rotates each time you press the user button on the board













#### **Assignment (if there is time)**

- Merge the Bluetooth framework in your Accelerometer project
- Send actual data from the accelerometer to the smartphone





# **Thank You!**

#### Master thesis: ugomaria.colesanti@intecs.it

www.intecs.it