Tutorial Third Session

Lecturer :
Shahab Mahmoudi Sadaghiani

Email : shahab.mahmoudisadaghiani@mail.mcgill.ca

ECSE 444 - Microprocessor -Winter 2023

McGill

Agenda:

1. Review GPIO Pin Use
2. Reference Voltage
3. Temperature Sensor
4. Analog to Digital Converter (ADC)

Links :

STM32L4+ Discovery kit IoT node, low-power wireless

https://www.st.com/en/evaluation-tools/b-l4s5i-iot01a.html

STM32L4S5VI

https://www.st.com/en/microcontrollers-microprocessors/stm32l4s5vi.html

Schematic

en.MB1297-L4S5VI-E03_Schematic

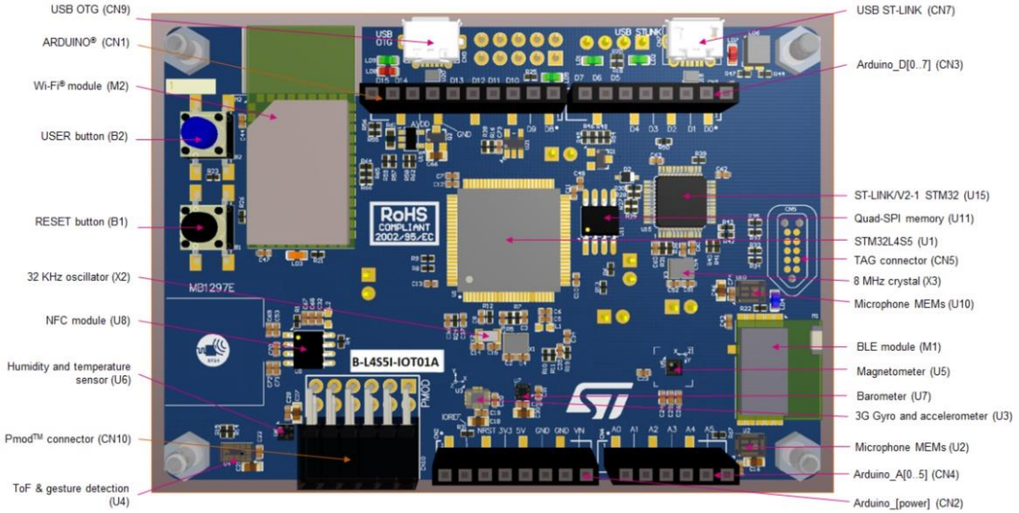
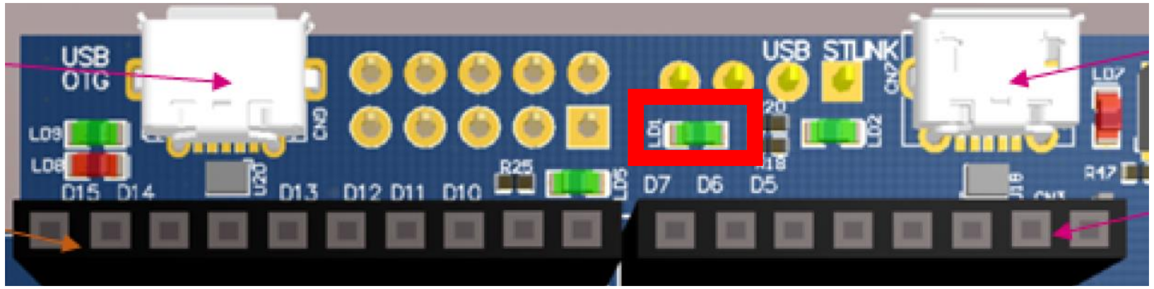
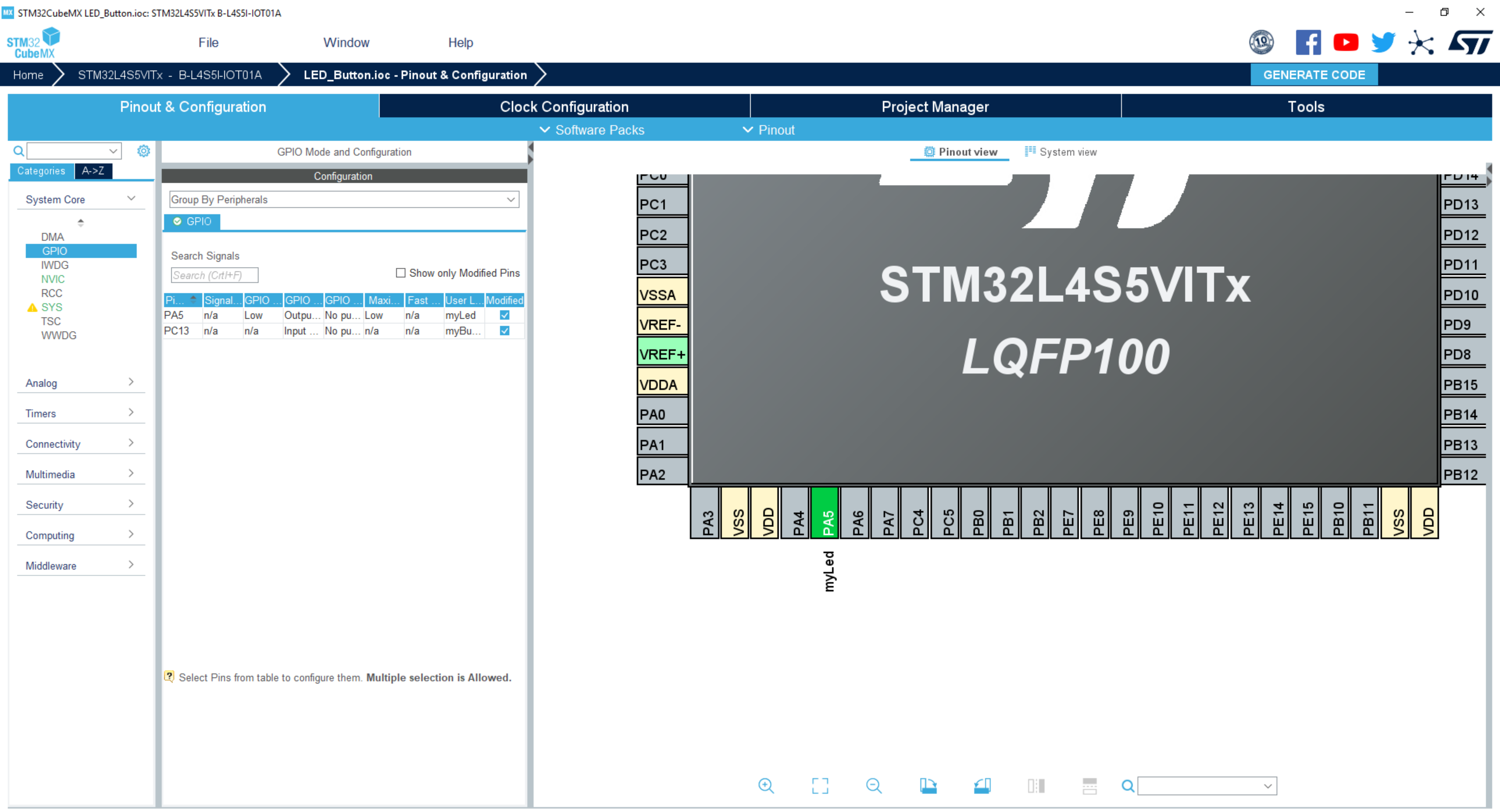


Table 4. Button and LED control port

Reference	Color	Name	Comment
B1	Black	Reset	-
B2	Blue	Wake-up	Wake-up alternate function
LD1	Green	LED1	PA5 (Alternate with ARD.D13)
LD2	Green	LED2	PB14
LD3	Yellow	LED3 (Wi-Fi®)	PC9, Wi-Fi® activity
LD4	Blue	LED4 (BLE)	PC9, Bluetooth® activity
LD5	Green	5V Power	5 V available
LD6	Bicolor (Red and green)	ST-LINK COM	Green during communication
LD7	Red	Fault Power	Current higher than 750 mA
LD8	Red	V_{BUS} OCRCR	PE3
LD9	Green	V_{BUS} OK	5 V USB available




STM32CubeMX LED_Button.ioc: STM32L4S5VITx B-L4S5I-IOT01A
STM32 CubeMX
File
Window
Help
Home
STM32L4S5VITx - B-L4S5I-IOT01A
LED_Button.ioc - Pinout & Configuration
GENERATE CODE
Pinout & Configuration
Clock Configuration
Project Manager
Tools
Software Packs
Pinout
Categories
A->Z
System Core
DMA
GPIO
IWDG
NVIC
RCC
SYS
TSC
WWDG
Analog
Timers
Connectivity
Multimedia
Security
Computing
Middleware
GPIO Mode and Configuration
Configuration
Group By Peripherals
GPIO
Search Signals
Search (Crtl+F)
Show only Modified Pins
Pi... Signal... GPIO ... GPIO ... GPIO ... Maxi... Fast ... User L... Modified
PA5 n/a Low Outpu... No pu... Low n/a myLed
PC13 n/a n/a Input ... No pu... n/a n/a myBu...
Select Pins from table to configure them. Multiple selection is Allowed.
Pinout view
System view
STM32L4S5VITx
LQFP100
PC1
PC2
PC3
VSSA
VREF-
VREF+
VDDA
PA0
PA1
PA2
PD13
PD12
PD11
PD10
PD9
PD8
PB15
PB14
PB13
PB12
PA3
VSS
VDD
PA4
PA5
PA6
PA7
PC4
PC5
PB0
PB1
PB2
PE7
PE8
PE9
PE10
PE11
PE12
PE13
PE14
PE15
PB10
PB11
VSS
VDD
myLed

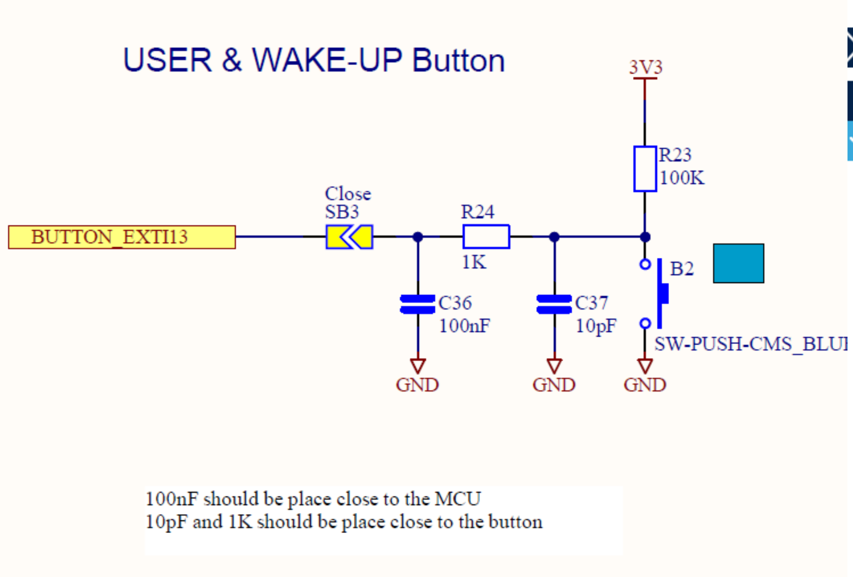
STM32CubeMX LED_Button.ioc: STM32L4S5VITx B-L4S5I-IOT01A

File Window Help

Home

Configuration

Software Packs



Analog

Timers

Connectivity

Multimedia

Security

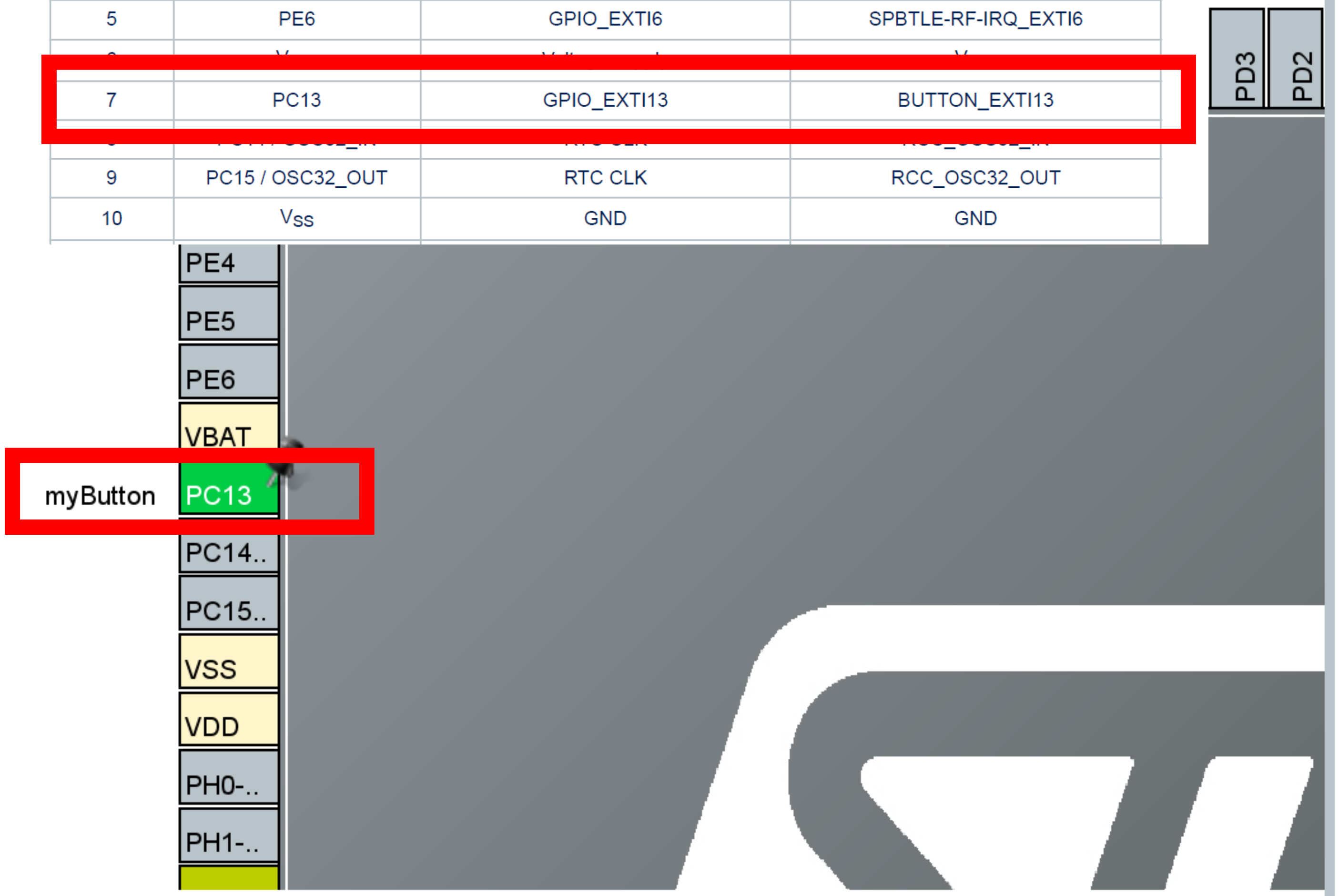
Computing

Middleware

Select Pins from table to configure them. **Multiple selection is Allowed.**

Table 13. B-L4S5I-IOT01A Discovery kit for the IoT node I/O assignment

Pin number	Pin name	Feature / comment	Signal / label
1	PE2	GPIO_Output	ST25DV04K RF_DISABLE
2	PE3	GPIO_EXTI3	USB_OTG_OVRCR_EXTI3
3	PE4	GPIO_EXTI4	ST25DV04K GPO
4	PE5	GPIO_EXTI5	SPSGRF-915-GPIO3_EXTI5
5	PE6	GPIO_EXTI6	SPBTLE-RF-IRQ_EXTI6
[illegible]	[illegible]	[illegible]	[illegible]
7	PC13	GPIO_EXTI13	BUTTON_EXTI13
[illegible]	[illegible]	[illegible]	[illegible]
9	PC15 / OSC32_OUT	RTC CLK	RCC_OSC32_OUT
10	V_{SS}	GND	GND



McGill

```c
/* USER CODE BEGIN 2 */
  char status = 0;
  /* USER CODE END 2 */
  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
  while (1)
  {
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
  status = HAL_GPIO_ReadPin(myButton_GPIO_Port, myButton_Pin);
  if (status == 0)
  HAL_GPIO_WritePin(myLed_GPIO_Port, myLed_Pin, GPIO_PIN_SET);
  else
  HAL_GPIO_WritePin(myLed_GPIO_Port, myLed_Pin, GPIO_PIN_RESET);
  }
  /* USER CODE END 3 */
}
```

Agenda:

1. Review GPIO Pin Use
2. Reference Voltage
3. Temperature Sensor
4. Analog to Digital Converter (ADC)

Links :

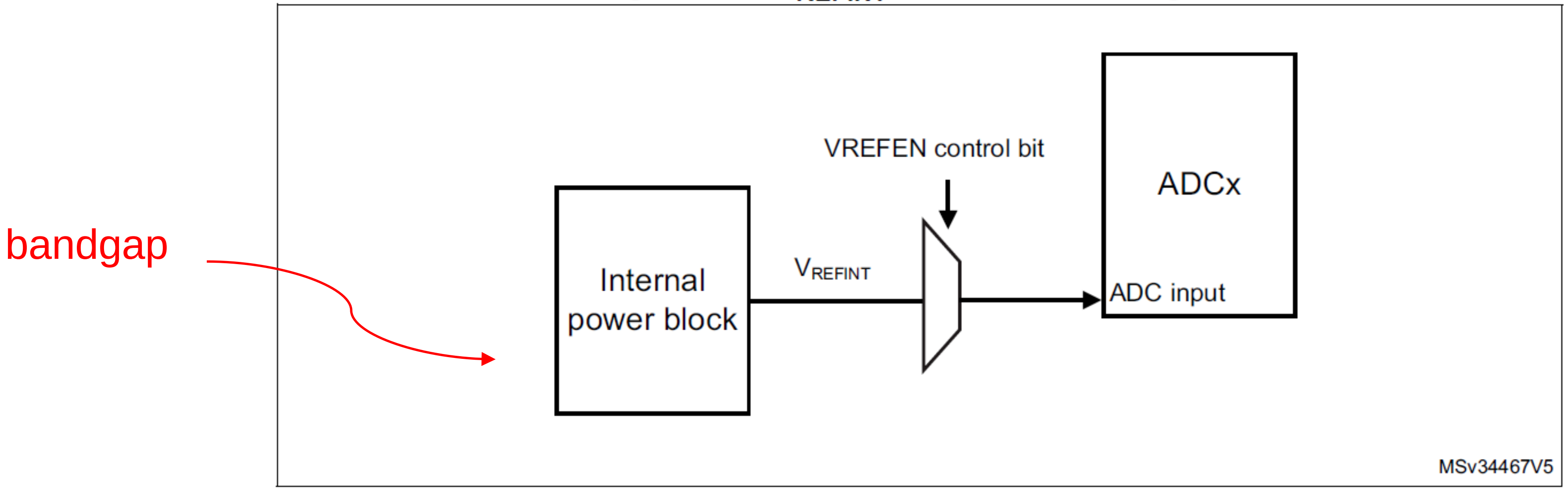
STM32L4S5VI
https://www.st.com/en/microcontrollers-microprocessors/stm32l4s5vi.html

Ultra-low-power Arm® Cortex®-M4 32-bit MCU+FPU, 100DMIPS, up to 1MB Flash, 128 KB SRAM, USB OTG FS, analog, audio

https://www.st.com/resource/en/datasheet/stm32l476je.pdf

21.4.34 Monitoring the internal voltage reference

Figure 156. V_{REFINT} channel block diagram



1. The VREFEN bit into ADCx_CCR register must be set to enable the conversion of internal channels (V_{REFINT}).

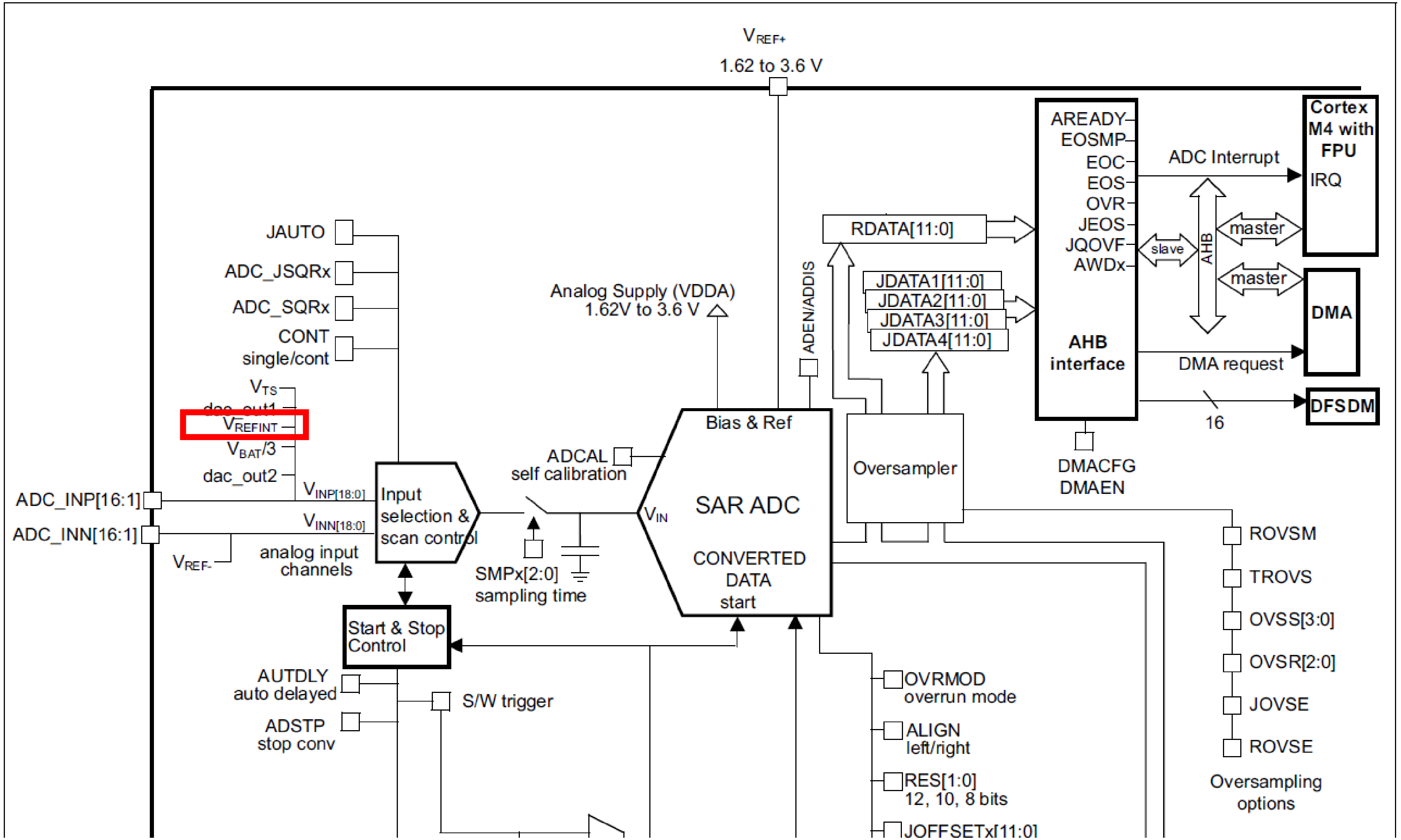
3.15.2 Internal voltage reference (V_{REFINT})

Table 9. Internal voltage reference calibration values

Calibration value name	Description	Memory address
VREFINT	Raw data acquired at a temperature of 30 °C (± 5 °C), $V_{DDA} = V_{REF+} = 3.0$ V (± 10 mV)	0x1FFF 75AA - 0x1FFF 75AB

ADC block diagram : P617

Figure 87. ADC block diagram



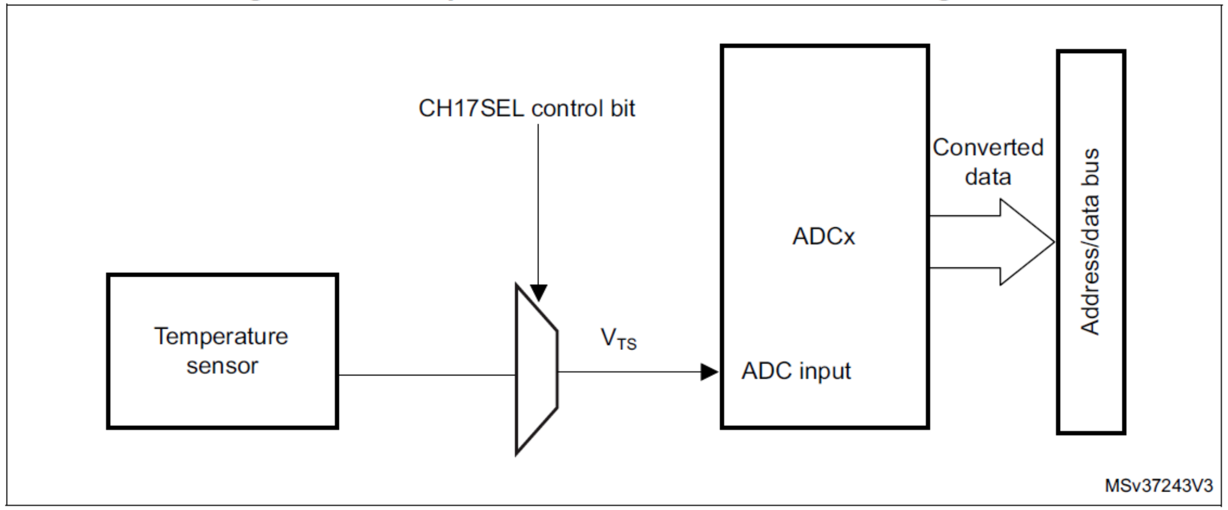
https://www.st.com/en/microcontrollers-microprocessors/stm32l4s5vi.html

Agenda:

1. Review GPIO Pin Use
2. Reference Voltage
3. Temperature Sensor
4. Analog to Digital Converter (ADC)

21.4.32 Temperature sensor

Figure 154. Temperature sensor channel block diagram



6.3.23 Temperature sensor characteristics

Table 77. TS characteristics

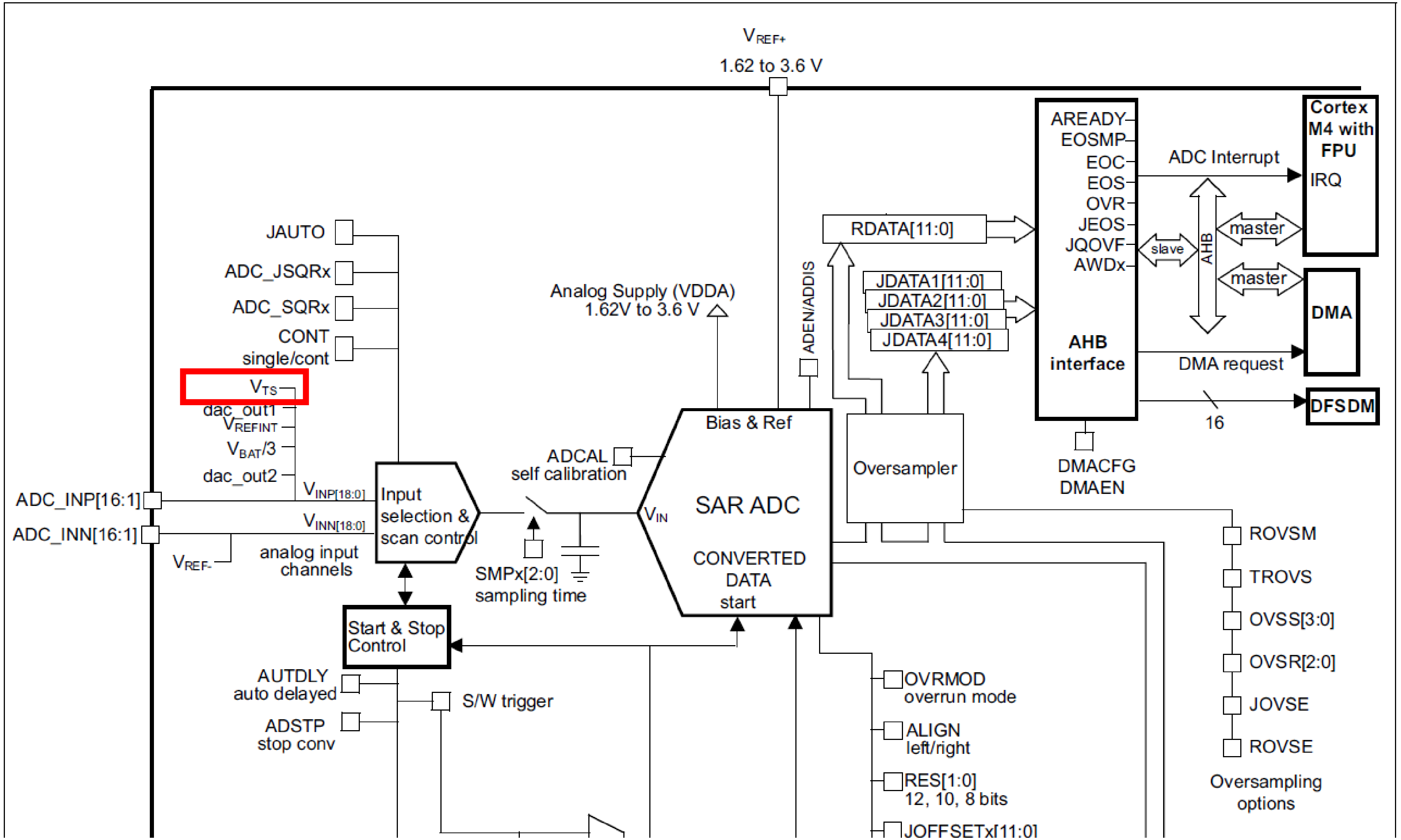
Symbol	Parameter	Min	Typ	Max	Unit
T_L (1)	V_{TS} linearity with temperature	-	±1	±2	°C
Avg_Slope (2)	Average slope	2.3	2.5	2.7	mV/°C
V_{30}	Voltage at 30°C (±5 °C) (3)	0.742	0.76	0.785	V
t_{START} (TS_BUF) (1)	Sensor Buffer Start-up time in continuous mode (4)	-	8	15	µs
t_{START} (1)	Start-up time when entering in continuous mode (4)	-	70	120	µs
t_{S_temp} (1)	ADC sampling time when reading the temperature	5	-	-	µs
I_{DD}(TS) (1)	Temperature sensor consumption from V_{DD}, when selected by ADC	-	4.7	7	µA

1. Guaranteed by design.
2. Guaranteed by characterization results.
3. Measured at V_{DDA} = 3.0 V ±10 mV. The V_{30} ADC conversion result is stored in the TS_CAL1 byte. Refer to *Table 8: Temperature sensor calibration values*.
4. Continuous mode means Run/Sleep modes, or temperature sensor enable in Low-power run/Low-power sleep modes.

You need to study document for setting ADC parameters.

ADC block diagram : P617

Figure 87. ADC block diagram



Steps :

1. Download documents :
   1. Chip document
   2. HAL driver document
2. Study section 21.4.32 from chip document
3. Follow steps for reading the temperature
   1. Consider the “NOTE” section.
4. Using HAL functions for writing your program
5. Calibration coefficient is on:

https://www.st.com/resource/en/datasheet/stm32l476je.pdf

These values are stored in memory, you need to read them.

Agenda:

1. Review GPIO Pin Use
2. Reference Voltage
3. Temperature Sensor
4. Analog to Digital Converter (ADC)

McGill

Figure 87. ADC block diagram

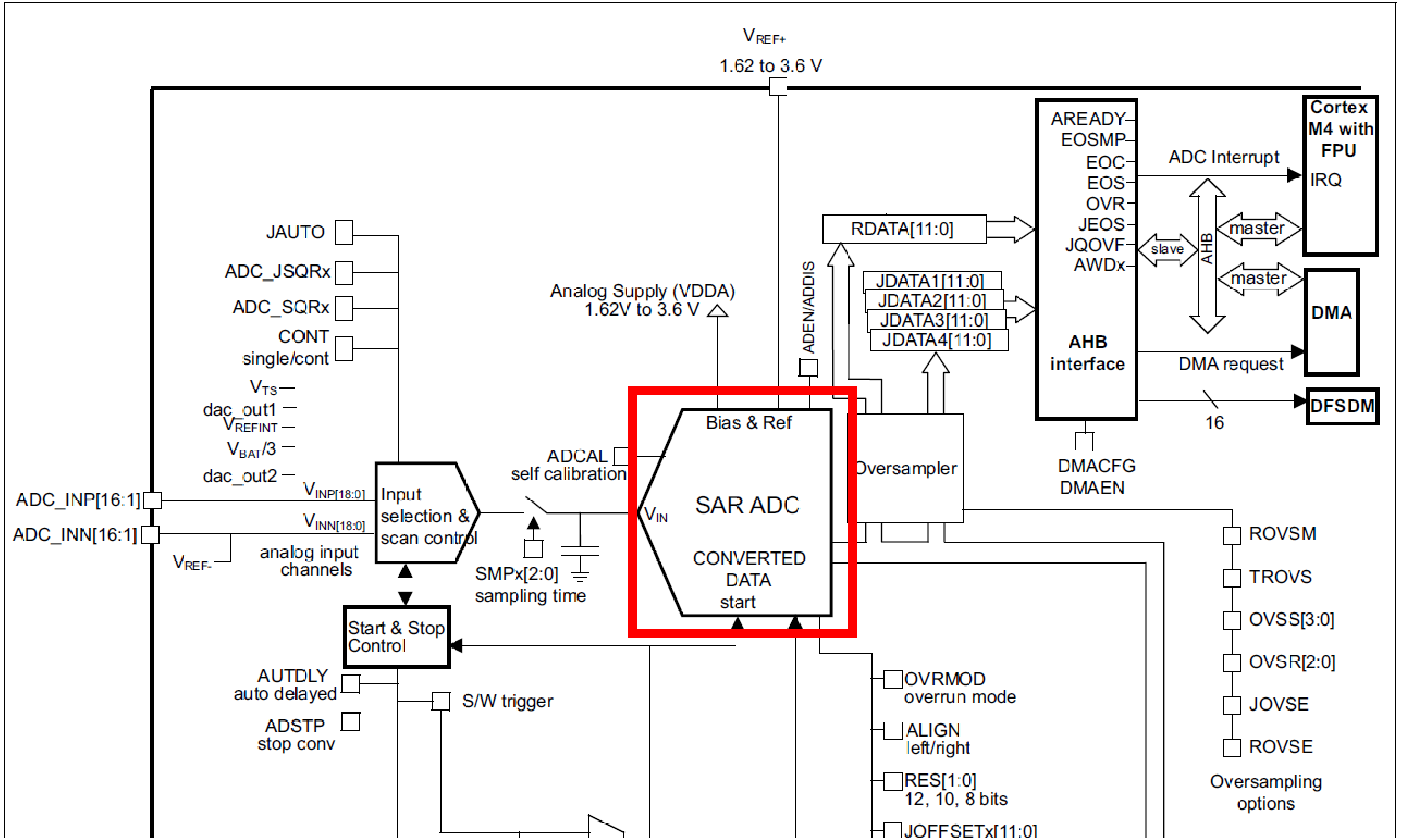


Table 127. ADC internal input/output signals

Internal signal name	Signal type	Description
EXT[15:0]	Inputs	Up to 16 external trigger inputs for the regular conversions (can be connected to on-chip timers). These inputs are shared between the ADC master and the ADC slave.
JEXT[15:0]	Inputs	Up to 16 external trigger inputs for the injected conversions (can be connected to on-chip timers). These inputs are shared between the ADC master and the ADC slave.
ADC_AWDx_OUT	Output	Internal analog watchdog output signal connected to on-chip timers (x = Analog watchdog number 1,2,3)
V_{TS}	Input	Output voltage from internal temperature sensor
dac_out1	Input	DAC internal channel 1
V_{REFINT}	Input	Output voltage from internal reference voltage
dac_out2	Input	DAC internal channel 2
V_{BAT}	Input supply	External battery voltage supply

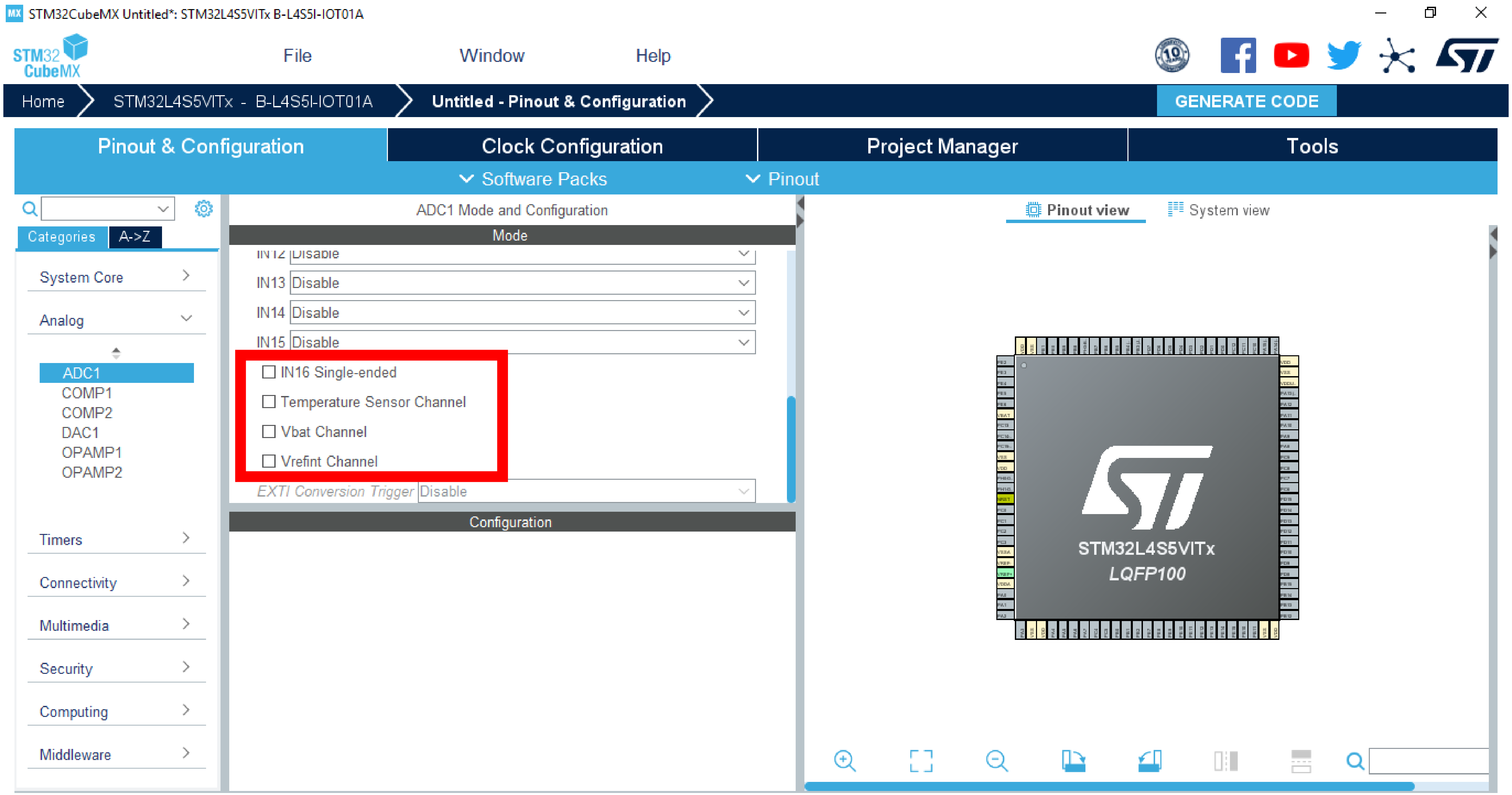
21.2 ADC main features

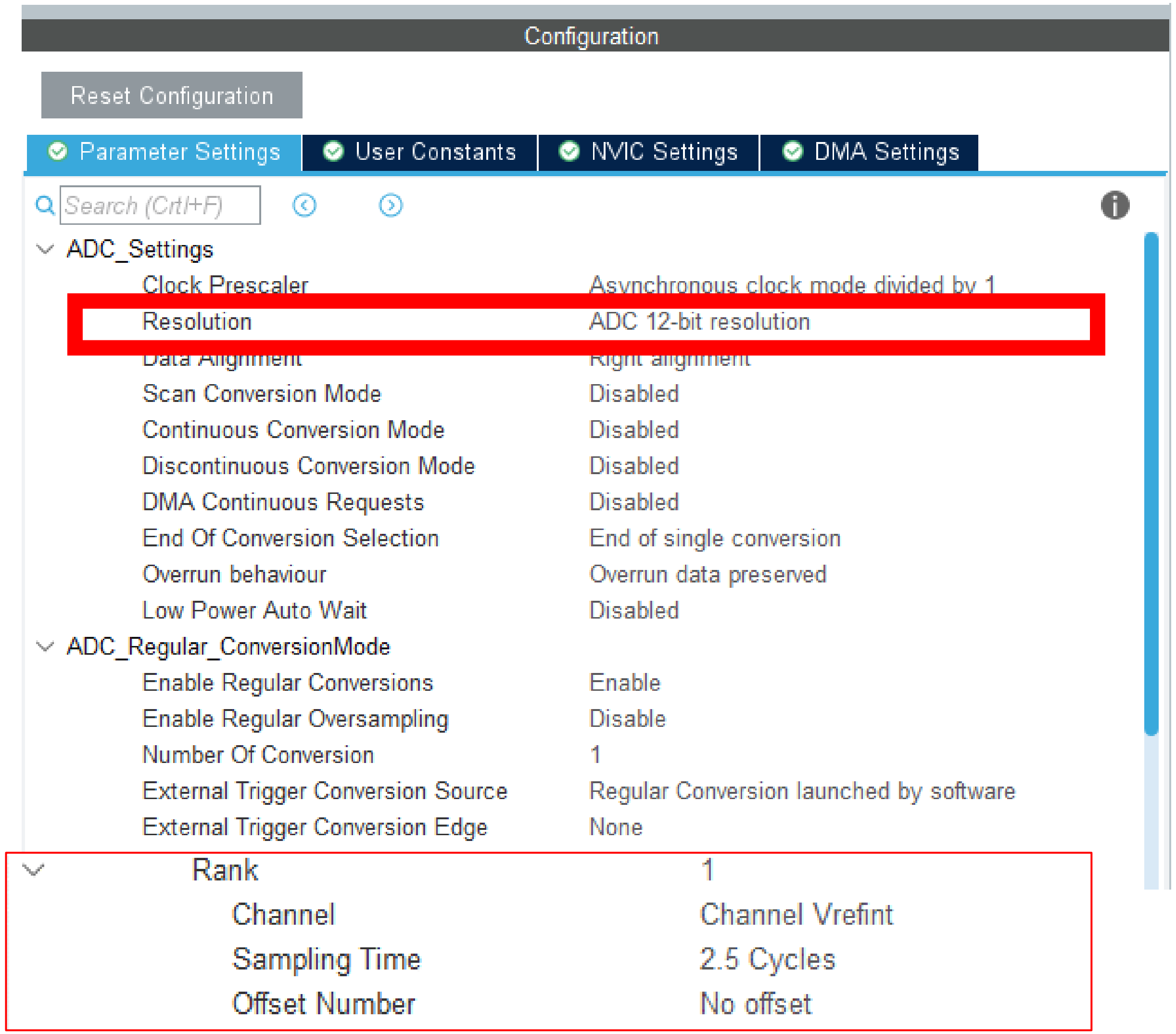
- High-performance features
  - Up to 2 ADCs which can operate in dual mode:
    ADC1 is connected to 16 external channels + 3 internal channels
    ADC2 is connected to 16 external channels + 2 internal channels
  - 12, 10, 8 or 6-bit configurable resolution
  - ADC conversion time is independent from the AHB bus clock frequency
  - Faster conversion time by lowering resolution
  - Manage single-ended or differential inputs
  - AHB slave bus interface to allow fast data handling
  - Self-calibration
  - Channel-wise programmable sampling time
  - Up to four injected channels (analog inputs assignment to regular or injected channels is fully configurable)
  - Hardware assistant to prepare the context of the injected channels to allow fast context switching
  - Data alignment with in-built data coherency
  - Data can be managed by DMA for regular channel conversions
  - Data can be routed to DFSDM for post processing
  - 4 dedicated data registers for the injected channels

21.4.11 Channel selection (SQRx, JSQRx)

There are up to 19 multiplexed channels per ADC:

- 5 fast analog inputs coming from GPIO pads (ADCx_INP/INN[1:5])
- Up to 11 slow analog inputs coming from GPIO pads (ADCx_INP/INN[6:16])
- The ADCs are connected to the following internal analog inputs:
  - The internal reference voltage (V_{REFINT}) is connected to ADC1_INP0.
  - The internal temperature sensor (V_{TS}) is connected to ADC1_INP17.
  - The V_{BAT} monitoring channel ($V_{BAT}/3$) is connected to ADC1_INP18.
  - The DAC1 internal channel 1 is connected to ADC2_INP17.
  - The DAC1 internal channel 2 is connected to ADC2_INP18.


STM32CubeMX Untitled*: STM32L4S5VITx B-L4S5I-IOT01A
File
Window
Help
Home
STM32L4S5VITx - B-L4S5I-IOT01A
Untitled - Pinout & Configuration
GENERATE CODE
Pinout & Configuration
Clock Configuration
Project Manager
Tools
Software Packs
Pinout
Categories
A->Z
System Core
Analog
ADC1
COMP1
COMP2
DAC1
OPAMP1
OPAMP2
Timers
Connectivity
Multimedia
Security
Computing
Middleware
ADC1 Mode and Configuration
Mode
IN13 Disable
IN14 Disable
IN15 Disable
IN16 Single-ended
Temperature Sensor Channel
Vbat Channel
Vrefint Channel
EXTI Conversion Trigger Disable
Configuration
Pinout view
System view
STM32L4S5VITx
LQFP100



Consider sampling time of ADC and input signal

UM1884

Description of STM32L4/L4+ HAL and low-layer drivers

Execution of ADC conversions

1. Optionally, perform an automatic ADC calibration to improve the conversion accuracy using function HAL_ADCEx_Calibration_Start().
2. ADC driver can be used among three modes: polling, interruption, transfer by DMA.
   - ADC conversion by polling:
     - Activate the ADC peripheral and start conversions using function HAL_ADC_Start()
     - Wait for ADC conversion completion using function HAL_ADC_PollForConversion()
     - Retrieve conversion results using function HAL_ADC_GetValue()
     - Stop conversion and disable the ADC peripheral using function HAL_ADC_Stop()
   - ADC conversion by interruption:
     - Activate the ADC peripheral and start conversions using function HAL_ADC_Start_IT()
     - Wait for ADC conversion completion by call of function HAL_ADC_ConvCpltCallback() (this function must be implemented in user program)
     - Retrieve conversion results using function HAL_ADC_GetValue()
     - Stop conversion and disable the ADC peripheral using function HAL_ADC_Stop_IT()
   - ADC conversion with transfer by DMA:
     - Activate the ADC peripheral and start conversions using function HAL_ADC_Start_DMA()
     - Wait for ADC conversion completion by call of function HAL_ADC_ConvCpltCallback() or HAL_ADC_ConvHalfCpltCallback() (these functions must be implemented in user program)
     - Conversion results are automatically transferred by DMA into destination variable address.
     - Stop conversion and disable the ADC peripheral using function HAL_ADC_Stop_DMA()

Lab 2 summary:

First part

Push button and LED

Second part

Read reference voltage

Third part

Read temperature sensor

Fourth part

Combining all three previous steps

Image source : www.mtlblog.com/