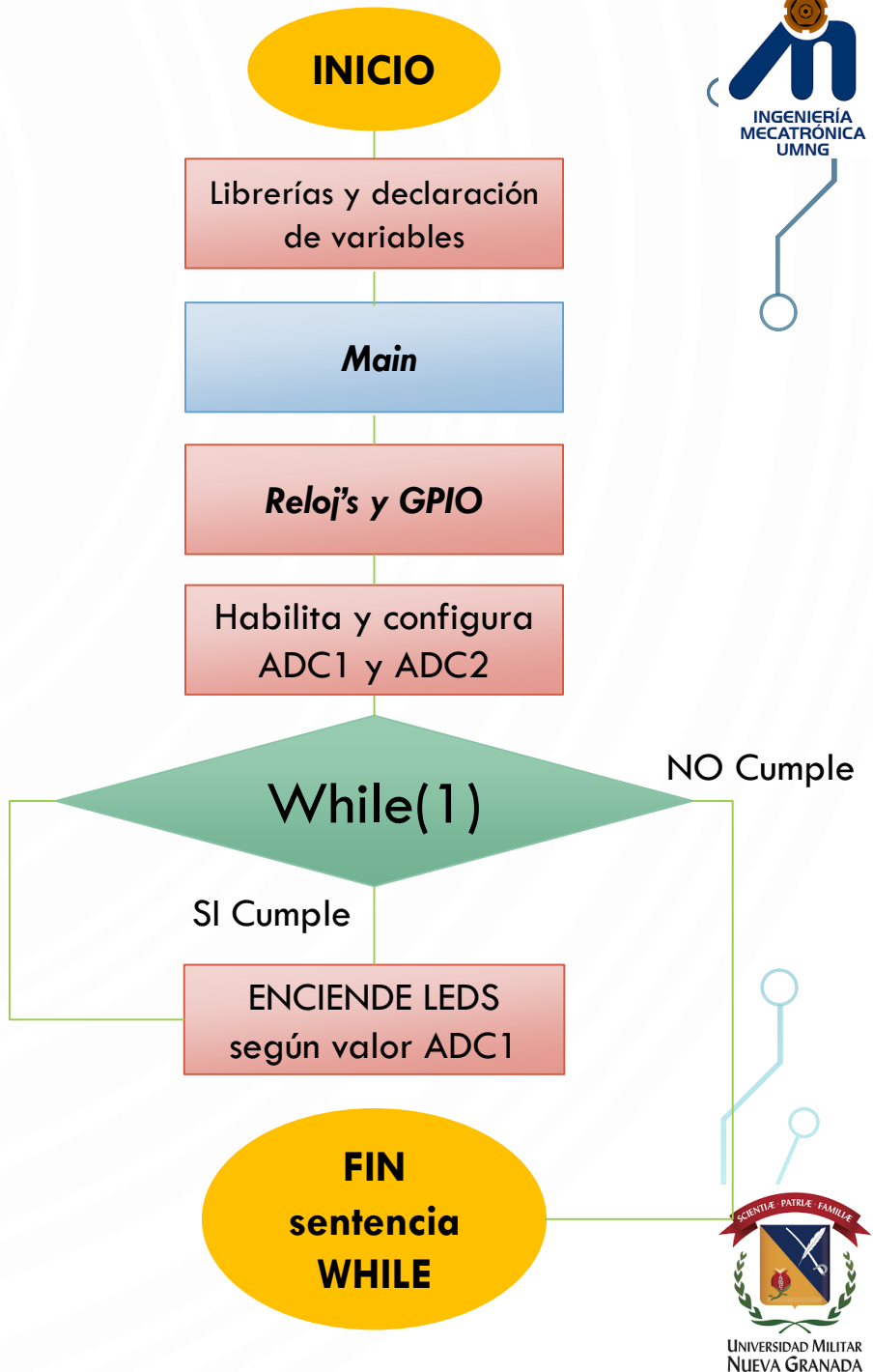
# MICROS 32 BITS
# STM - GPIO

ROBINSON JIMENEZ MORENO


UNIVERSIDAD MILITAR NUEVA GRANADA

INGENIERÍA MECATRÓNICA UMNG

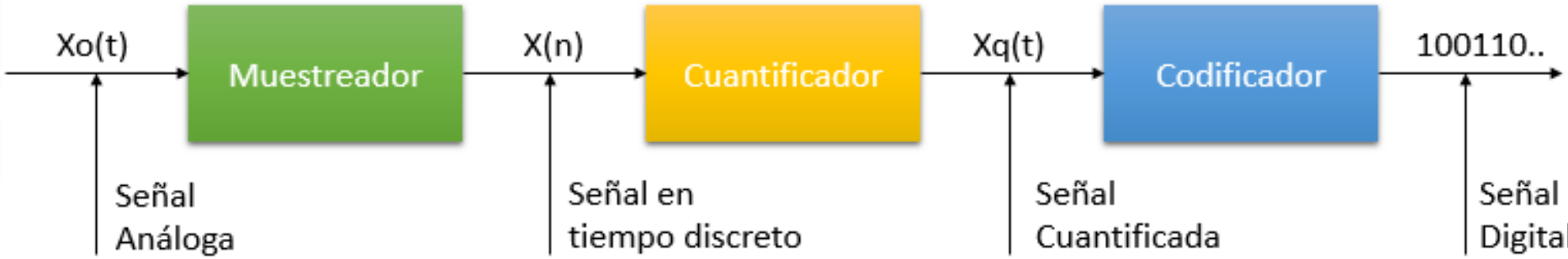
```c
#include <stdio.h>
#include "stm32f7xx.h"

int main(void){

  RCC -> AHB1ENR = 0X6; //PUERTO B Y C
  RCC -> APB2ENR = 0X100; //HABILITAR EL ADC 1
//  RCC -> APB2ENR = 0X200; //HABILITAR EL ADC 2

  GPIOB -> MODER = 0X10004001; //COLOCAR EN SALIDA PARA ENCENDER LOS LEDS
  GPIOB -> OTYPER = 0X0; //PUSH PULL
  GPIOB -> OSPEEDR = 0X10004001; //VELOCIDAD MEDIA
  GPIOB -> PUPDR = 0X10004001; //PULL UP

  GPIOC -> MODER = 0X3; //ANALOGO PARA EL PIN 0

  ADC1 -> CR1 = 0X0; //RESOLUCION DE 12 BITS (POR DEFECTO)
  ADC1 -> CR2 = 0X1; //ENCENDER EL ADC
  ADC1 -> SMPR1 = 0X7FFFFFF; //TIEMPO DE MUESTREO EN CICLOS
  ADC1 -> SMPR2 = 0X37777777; //TIEMPO DE MUESTREO EN CICLOS
  ADC1 -> SQR3 = 10; //CANAL 10 DEL ADC
  while(1){
    ADC1 -> CR2 |= 0X40000000; //INICIAR LA CONVERSION EN CANALES REGULARES
    if(ADC1 -> DR == 2048){
      GPIOB -> ODR = 0X4000;
    }
    if(ADC1 -> DR == 4095){
      GPIOB -> ODR = 0X1;
    }
    if(ADC1 -> DR == 0){
      GPIOB -> ODR = 0X80;
    }
```



# ADC- STM32F4

## Módulo ADC:

Permite la conversión de una señal analógica a digital, mediante el siguiente procedimiento:




La STM32F4VG cuenta con tres canales de conversión y guarda el valor análogo en un código binario de 12 bits. Además, posee hasta 19 entradas multiplexadas que le permiten medir señales de 16 fuentes externas, dos fuentes internas, y una del canal VBAT (voltaje de batería)

# ADC- STM32F4

**Módulo ADC:**

La conversión DAC de los canales se puede realizar en diferentes modos: modo único, continuo, escaneo o discontinuo. El resultado de la conversión ADC se almacena en un registro de 16 bits, en donde el código de 12 bits puede ser justificado a la derecha o a la izquierda del registro.

# ADC- STM32F4

## Módulo ADC:

Las características principales del módulo ADC son:

- Resolución configurable a 12 bits, 10 bits, 8 bits o 6 bits.
- Generación de una interrupción al final de la conversión.
- Modos de conversión único y continuo.
- Opción de activación externa con polaridad configurable
- Modo discontinuo.
- Modo Doble / Triple (en dispositivos con 2 ADCs o más).
- Retardo configurable entre conversiones en modo entrelazado Dual / Triple
- Suministro de energía al módulo ADC: 2,4 V a 3,6 V a toda velocidad y hacia abajo a 1,8 V a velocidad más lenta
- Rango de entrada ADC: VREF- ≤ ≤ VIN VREF +

# ADC- STM32F4

## **Módulo ADC:** Piner principales del módulo ADC

| NOMBRE | TIPO DE SEÑAL | DESCRIPCION |
|---|---|---|
| $V_{REF+}$ | Entrada, Referencia Análoga Positiva | El mayor voltaje (positivo) para ADC 1.8 v ≤ $V_{REF+}$ ≤ $V_{DDA}$ |
| $V_{DDA}$ | Entrada, Suministro Análogo | Suministro de voltaje análogo igual a VDD:<br>• 2.4 v ≤ $V_{DDA}$ ≤ $V_{DD}$ (3.6 v) para máxima velocidad<br>• 1.8 v ≤ $V_{DDA}$ ≤ $V_{DD}$ (3.6 v) para velocidad reducida |
| $V_{REF-}$ | Entrada, Referencia Análoga Negativa | El menor voltaje (negativo) para ADC 1.8 $V_{REF-}$ = $V_{SSA}$ |
| $V_{SSA}$ | Entrada, Tierra Análoga | Tierra para suministro de voltaje análogo igual a $V_{SS}$ |
| ADCX_IN[15:0] | Señales de Entrada Análogas | 16 Canales Análogos de entrada |

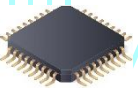
# ADC introduction

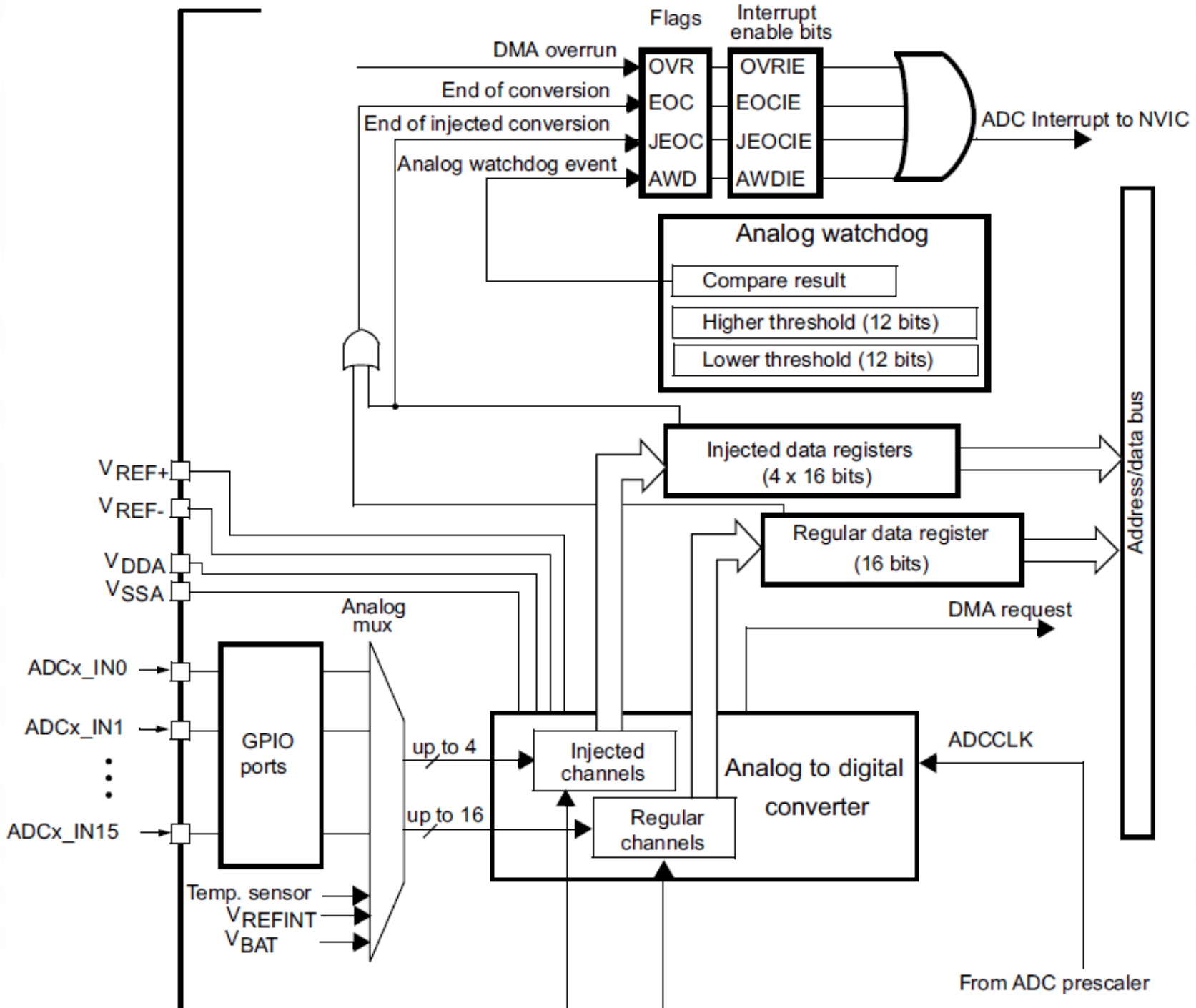
The 12-bit ADC is a successive approximation analog-to-digital converter. It has up to 19 multiplexed channels allowing it to measure signals from 16 external sources, two internal sources, and the $V_{BAT}$ channel. The A/D conversion of the channels can be performed in single, continuous, scan or discontinuous mode. The result of the ADC is stored into a left- or right-aligned 16-bit data register.

The analog watchdog feature allows the application to detect if the input voltage goes beyond the user-defined, higher or lower thresholds.

# ADC main features

- 12-bit, 10-bit, 8-bit or 6-bit configurable resolution
- Interrupt generation at the end of conversion, end of injected conversion, and in case of analog watchdog or overrun events
- Single and continuous conversion modes
- Scan mode for automatic conversion of channel 0 to channel ‘n’
- Data alignment with in-built data coherency
- Channel-wise programmable sampling time
- External trigger option with configurable polarity for both regular and injected conversions
- Discontinuous mode
- Dual/Triple mode (on devices with 2 ADCs or more)
- Configurable DMA data storage in Dual/Triple ADC mode
- Configurable delay between conversions in Dual/Triple interleaved mode
- ADC supply requirements: 2.4 V to 3.6 V at full speed and down to 1.8 V at slower speed
- ADC input range: $V_{REF-} \leq V_{IN} \leq V_{REF+}$
- DMA request generation during regular channel conversion




Flags
Interrupt enable bits
DMA overrun
End of conversion
End of injected conversion
Analog watchdog event
OVR
EOC
JEOC
AWD
OVRIE
EOCIE
JEOCIE
AWDIE
ADC Interrupt to NVIC
Analog watchdog
Compare result
Higher threshold (12 bits)
Lower threshold (12 bits)
Injected data registers (4 x 16 bits)
Regular data register (16 bits)
Address/data bus
DMA request
VREF+
VREF-
VDDA
VSSA
Analog mux
ADCx_IN0
ADCx_IN1
ADCx_IN15
GPIO ports
up to 4
up to 16
Injected channels
Regular channels
Analog to digital converter
ADCCLK
Temp. sensor
VREFINT
VBAT
From ADC prescaler

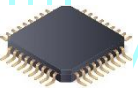
# ADC on-off control

The ADC is powered on by setting the ADON bit in the ADC_CR2 register. When the ADON bit is set for the first time, it wakes up the ADC from the Power-down mode.

The conversion starts when either the SWSTART or the JSWSTART bit is set.

The user can stop conversion and put the ADC in power down mode by clearing the ADON bit. In this mode the ADC consumes almost no power (only a few µA).

| Name | Signal type | Remarks |
|---|---|---|
| $V_{REF+}$ | Input, analog reference positive | The higher/positive reference voltage for the ADC, 1.8 V ≤ $V_{REF+}$ ≤ $V_{DDA}$ |
| $V_{DDA}$ | Input, analog supply | Analog power supply equal to $V_{DD}$ and 2.4 V ≤$V_{DDA}$ ≤$V_{DD}$ (3.6 V) for full speed 1.8 V ≤$V_{DDA}$ ≤$V_{DD}$ (3.6 V) for reduced speed |
| $V_{REF-}$ | Input, analog reference negative | The lower/negative reference voltage for the ADC, $V_{REF-}$ = $V_{SSA}$ |
| $V_{SSA}$ | Input, analog supply ground | Ground for analog power supply equal to $V_{SS}$ |
| ADCx_IN[15:0] | Analog input signals | 16 analog input channels |



## ADC clock

The ADC features two clock schemes:

- Clock for the analog circuitry: ADCCLK, common to all ADCs

  This clock is generated from the APB2 clock divided by a programmable prescaler that allows the ADC to work at $f_{PCLK2}$/2, /4, /6 or /8. Refer to the datasheets for the maximum value of ADCCLK.

- Clock for the digital interface (used for registers read/write access)

  This clock is equal to the APB2 clock. The digital interface clock can be enabled/disabled individually for each ADC through the RCC APB2 peripheral clock enable register (RCC_APB2ENR).

## Channel selection

There are 16 multiplexed channels. It is possible to organize the conversions in two groups: regular and injected. A group consists of a sequence of conversions that can be done on any channel and in any order. For instance, it is possible to implement the conversion sequence in the following order: ADC_IN3, ADC_IN8, ADC_IN2, ADC_IN2, ADC_IN0, ADC_IN2, ADC_IN2, ADC_IN15.

- A **regular group** is composed of up to 16 conversions. The regular channels and their order in the conversion sequence must be selected in the ADC_SQRx registers. The total number of conversions in the regular group must be written in the L[3:0] bits in the ADC_SQR1 register.
- An **injected group** is composed of up to 4 conversions. The injected channels and their order in the conversion sequence must be selected in the ADC_JSQR register. The total number of conversions in the injected group must be written in the L[1:0] bits in the ADC_JSQR register.

# Single conversion mode

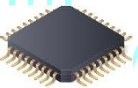
In Single conversion mode the ADC does one conversion. This mode is started with the CONT bit at 0 by either:

- setting the SWSTART bit in the ADC_CR2 register (for a regular channel only)
- setting the JSWSTART bit (for an injected channel)
- external trigger (for a regular or injected channel)

Once the conversion of the selected channel is complete:

- If a regular channel was converted:
  - The converted data are stored into the 16-bit ADC_DR register
  - The EOC (end of conversion) flag is set
  - An interrupt is generated if the EOCIE bit is set
- If an injected channel was converted:
  - The converted data are stored into the 16-bit ADC_JDR1 register
  - The JEOC (end of conversion injected) flag is set
  - An interrupt is generated if the JEOCIE bit is set

Then the ADC stops.



# Data alignment

The ALIGN bit in the ADC_CR2 register selects the alignment of the data stored after conversion. Data can be right- or left-aligned as shown in *Figure 77* and *Figure 78*.

The converted data value from the injected group of channels is decreased by the user-defined offset written in the ADC_JOFRx registers so the result can be a negative value. The SEXT bit represents the extended sign value.

For channels in a regular group, no offset is subtracted so only twelve bits are significant.

**Figure 77. Right alignment of 12-bit data**

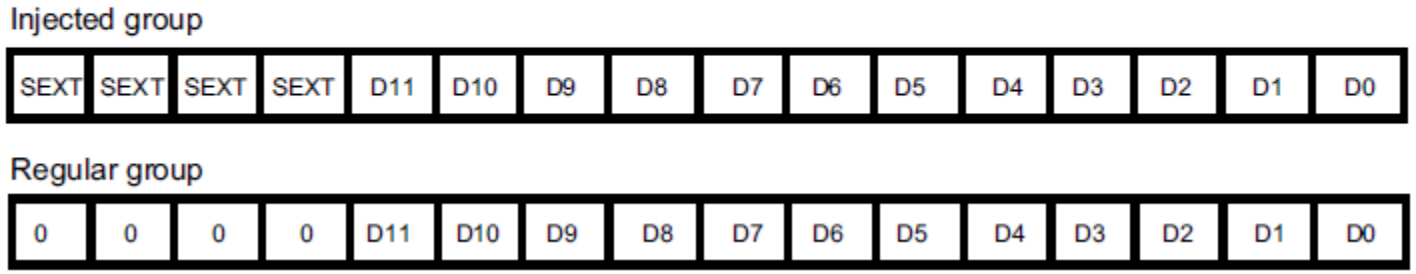
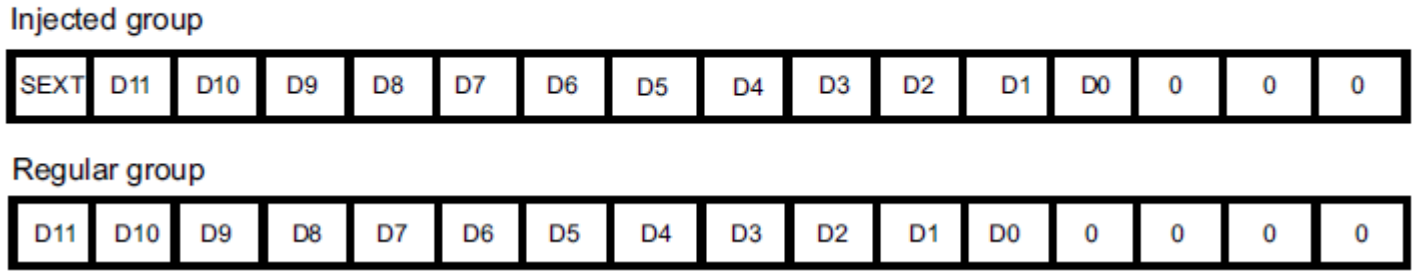



**Figure 78. Left alignment of 12-bit data**



## 15.13.2 ADC control register 1 (ADC_CR1)

Address offset: 0x04

Reset value: 0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | Res. | Res. | Res. | Res. | OVRIE | RES | | AWDEN | JAWDEN | Res. | Res. | Res. | Res. | Res. | Res. |
| | | | | | rw | rw | rw | rw | rw | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DISCNUM[2:0] | | | JDISCEN | DISCEN | JAUTO | AWDSGL | SCAN | JEOCIE | AWDIE | EOCIE | AWDCH[4:0] | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

Bit 5 **EOCIE:** Interrupt enable for EOC

This bit is set and cleared by software to enable/disable the end of conversion interrupt.

0: EOC interrupt disabled

1: EOC interrupt enabled. An interrupt is generated when the EOC bit is set.

## 15.13.2 ADC control register 1 (ADC_CR1)

Address offset: 0x04

Reset value: 0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | Res. | Res. | Res. | Res. | OVRIE | RES | | AWDEN | JAWDEN | Res. | Res. | Res. | Res. | Res. | Res. |
| | | | | | rw | rw | rw | rw | rw | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DISCNUM[2:0] | | | JDISCEN | DISCEN | JAUTO | AWDSGL | SCAN | JEOCIE | AWDIE | EOCIE | AWDCH[4:0] | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

Bits 25:24 **RES[1:0]**: Resolution

These bits are written by software to select the resolution of the conversion.

00: 12-bit (minimum 15 ADCCLK cycles)

01: 10-bit (minimum 13 ADCCLK cycles)

10: 8-bit (minimum 11 ADCCLK cycles)

11: 6-bit (minimum 9 ADCCLK cycles)

| Hexa | Decimal | Voltaje |
|---|---|---|
| 0 | 0 | 0V |
| FFF | 4095 | 3,3V |

## 15.13.3 ADC control register 2 (ADC_CR2)

Address offset: 0x08

Reset value: 0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | SWSTART | EXTEN | | EXTSEL[3:0] | | | | Res. | JSWSTART | JEXTEN | | JEXTSEL[3:0] | | | |
| | rw | rw | rw | rw | rw | rw | rw | | rw | rw | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | Res. | Res. | Res. | ALIGN | EOCS | DDS | DMA | Res. | Res. | Res. | Res. | Res. | Res. | CONT | ADON |
| | | | | rw | rw | rw | rw | | | | | | | rw | rw |

Bit 10 **EOCS:** End of conversion selection

This bit is set and cleared by software.

0: The EOC bit is set at the end of each sequence of regular conversions. Overrun detection is enabled only if DMA=1.

1: The EOC bit is set at the end of each regular conversion. Overrun detection is enabled.

Bit 0 **ADON:** A/D Converter ON / OFF

This bit is set and cleared by software.

Note: 0: Disable ADC conversion and go to power down mode

1: Enable ADC

## 15.13.9 ADC regular sequence register 1 (ADC_SQR1)

Address offset: 0x2C

Reset value: 0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | L[3:0] | | | | SQ16[4:1] | | | |
| | | | | | | | | rw | rw | rw | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SQ16_0 | SQ15[4:0] | | | | | SQ14[4:0] | | | | | SQ13[4:0] | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

# 15.13.11 ADC regular sequence register 3 (ADC_SQR3)

Address offset: 0x34

Reset value: 0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | Res. | SQ6[4:0] | | | | | SQ5[4:0] | | | | | SQ4[4:1] | | | |
| | | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SQ4_0 | SQ3[4:0] | | | | | SQ2[4:0] | | | | | SQ1[4:0] | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

Bits 31:30 Reserved, must be kept at reset value.

Bits 29:25 **SQ6[4:0]:** 6th conversion in regular sequence

These bits are written by software with the channel number (0..18) assigned as the 6th in the sequence to be converted.

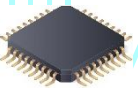
Bits 24:20 **SQ5[4:0]:** 5th conversion in regular sequence

Bits 19:15 **SQ4[4:0]:** 4th conversion in regular sequence

Bits 14:10 **SQ3[4:0]:** 3rd conversion in regular sequence

Bits 9:5 **SQ2[4:0]:** 2nd conversion in regular sequence

Bits 4:0 **SQ1[4:0]:** 1st conversion in regular sequence



# 15.13.14 ADC regular data register (ADC_DR)

Address offset: 0x4C

Reset value: 0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. |
| | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DATA[15:0] | | | | | | | | | | | | | | | |
| r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r |

Bits 31:16 Reserved, must be kept at reset value.

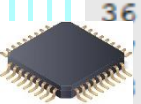
Bits 15:0 **DATA[15:0]:** Regular data

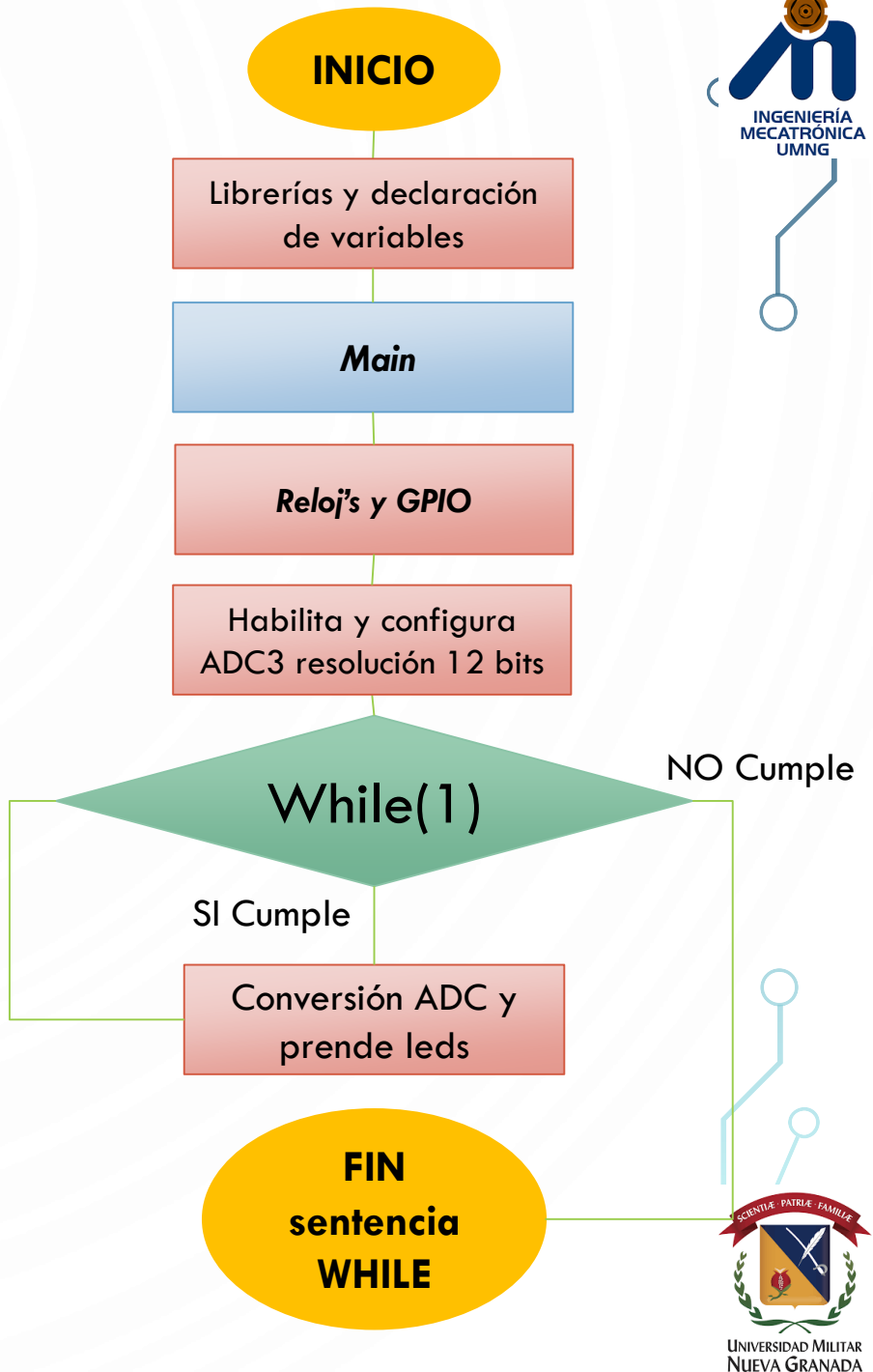
These bits are read-only. They contain the conversion result from the regular channels. The data are left- or right-aligned as shown in *Figure 77* and *Figure 78*.

| | | | |
|---|---|---|---|
| A0 | ADC | PA3 | ADC1_IN3 |
| A1 | ADC | PC0 | ADC1_IN10 |
| A2 | ADC | PC3 | ADC1_IN13 |
| A3 | ADC | PC1 | ADC1_IN11 |
| A4 | ADC | PC4 or PB9(1) | ADC1_IN14 (PC4) or I2C1_SDA (PB9) |
| A5 | ADC | PC5 or PB8(1) | ADC1_IN15 (PC5) or I2C1_SCL (PB8) |

```c
#include <stdio.h>
#include "stm32f7xx.h"
int temp=0;
int main(void){

  RCC -> AHB1ENR |= 0X6; //PUERTO B Y C
  RCC -> APB2ENR |= 0X400; //HABILITAR EL ADC 3

  GPIOB -> MODER |= 0X10004001; //COLOCAR EN SALIDA PARA ENCENDER LOS LEDS
  GPIOB -> OTYPER = 0X0; //PUSH PULL
  GPIOB -> OSPEEDR = 0X10004001; //VELOCIDAD MEDIA
  GPIOB -> PUPDR |= 0X10004001; //PULL UP

  GPIOC -> MODER |= 15; //ANALOGO PARA EL PIN 0 y 1

  ADC3 -> CR1 |= 0X0; //RESOLUCION DE 12 BITS (POR DEFECTO)
  ADC3 -> CR2 |= 0X201; //ENCENDER EL ADC

  while(1){
    ADC3 -> SQR3 =10; //CANAL 10 DEL ADC
    ADC3 -> CR2 |= 0X40000000; //INICIAR LA CONVERSION EN CANALES REGULARES
    while((ADC3->SR &=0X2)==1);
    temp=(ADC3 -> DR)*8058/100000;
    if(temp> 50){
      GPIOB -> ODR |= 0X1;
    }
    else {GPIOB -> ODR &= 0;}
      //
  }
}
```

```c
#include <stdio.h>
#include "stm32f7xx.h"
int temp=0;
int main(void){

  RCC -> AHB1ENR |= 0X6; //PUERTO B Y C
  RCC -> APB2ENR |= 0X400; //HABILITAR EL ADC 3

  GPIOB -> MODER |= 0X10004001; //COLOCAR EN SALIDA PARA ENCENDER LOS LEDS
  GPIOB -> OTYPER = 0X0; //PUSH PULL
  GPIOB -> OSPEEDR = 0X10004001; //VELOCIDAD MEDIA
  GPIOB -> PUPDR |= 0X10004001; //PULL UP

  GPIOC -> MODER |= 15; //ANALOGO PARA EL PIN 0 y 1

  ADC3 -> CR1 |= 0X0; //RESOLUCION DE 12 BITS (POR DEFECTO)
  ADC3 -> CR2 |= 0X201; //ENCENDER EL ADC

  while(1){
    ADC3 -> SQR3 =10; //CANAL 10 DEL ADC
    ADC3 -> CR2 |= 0X40000000; //INICIAR LA CONVERSION EN CANALES REGULARES
    while((ADC3->SR &=0X2)==1);
    temp=(ADC3 -> DR)*8058/100000;
    if(temp> 50){
      GPIOB -> ODR |= 0X01;
    }
    else {GPIOB -> ODR &= 0xFFFFFFF0;}
      //
    ADC3 -> SQR3 =11; //CANAL 11 DEL ADC
    ADC3 -> CR2 |= 0X40000000; //INICIAR LA CONVERSION EN CANALES REGULARES
    while((ADC3->SR &=0X2)==1);
    temp=(ADC3 -> DR)*8058/100000;
    if(temp > 40){
      GPIOB -> ODR |= 0X80;
    }
    else {GPIOB -> ODR &= 0XFFFFFF01;}
  }
}
```






INGENIERÍA MECATRÓNICA UMNG

UNIVERSIDAD MILITAR NUEVA GRANADA

# TAREA

Emplear un sensor acelerómetro para indicar el la dirección de inclinación de un panel solar con capacidad de seguimiento solar, emplear una LCD para visualizar dicho movimiento desde el centro de la misma con pixeles hacia arriba, abajo, izquierda o derecha según la información del acelerómetro.