

KHÓA HỌC LẬP TRÌNH VI ĐIỀU KHIỂN

Giảng viên

**NGUYỄN HUỲNH NHẬT
THƯỜNG**

LỊCH HỌC:

Tại Đà Nẵng: 19h30 - 22h30 thứ 2 và thứ 6

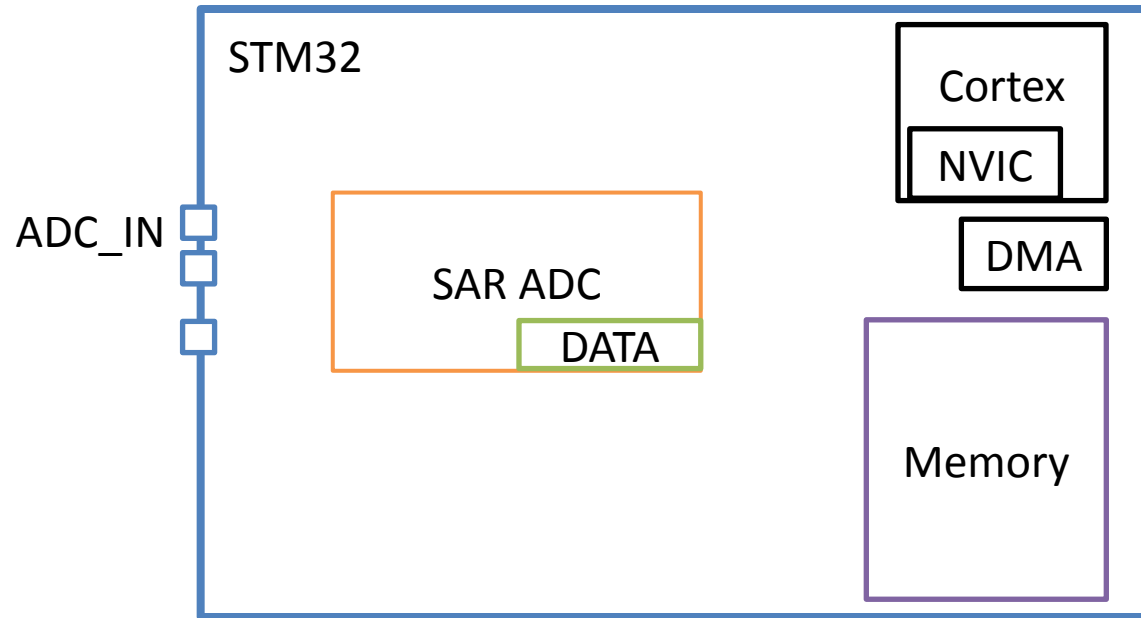
ĐỊA ĐIỂM:

Online qua nền tảng Zoom

MODULE 10

- ADC
- ADC POLLING
- ADC INTERRUPT, DMA
- PHÂN TÍCH DỰ ÁN THỰC TẾ





Analog to Digital Converter

Tổng quan về cảm biến

Cảm biến có ngõ ra tương tự analog

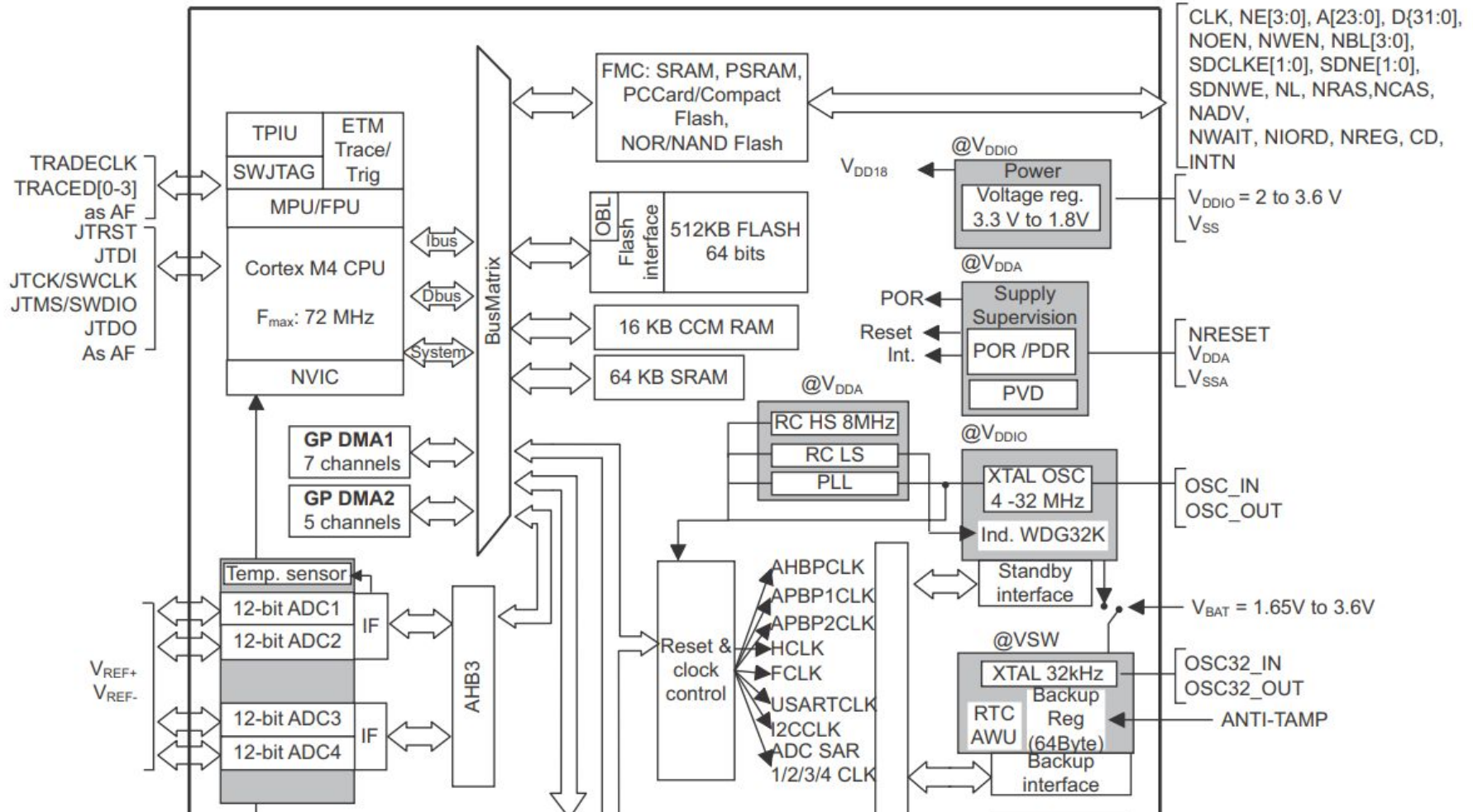
- + Điện áp (0v-5v, 0v-10v)
- + Dòng điện (0/4-20/24mA)

Cảm biến có ngõ ra số digital

- + Logic High, Logic Low (mạch so sánh)
- + Các chuẩn giao tiếp UART/ I2C/ SPI
- + Xung (Pulse)

STM32F303RE ADC

Figure 1. STM32F303xD/E block diagram



Ref Datasheet: <https://www.st.com/resource/en/datasheet/stm32f303re.pdf>

ADC CHANNEL

Each ADC has up to 19 multiplexed channels

External Channel

ADC external channels mapping:

Device	ADC1	ADC2	ADC3	ADC4
STM32F303xD/E	11	13	15	13

Internal Channel

Table 86. ADC internal channels summary

Product	ADC1	ADC2	ADC3	ADC4	Total of internal ADC channels
STM32F303xB/C/D/E, STM32F358 and STM32F398xE	<ul style="list-style-type: none">– 1 channel connected to temperature sensor.– 1 channel connected to VBAT/2– 1 channel connected to VREFINT– 1 channel connected to OPAMP1 reference voltage output (VREFOPAMP1).	<ul style="list-style-type: none">– 1 channel connected to VREFINT.– 1 channel connected to OPAMP2 reference voltage output (VREFOPAMP2).	<ul style="list-style-type: none">– 1 channel connected to VREFINT.– 1 channel connected to OPAMP3 reference voltage output (VREFOPAMP3).	<ul style="list-style-type: none">– 1 channel connected to VREFINT.– 1 channel connected to OPAMP4 reference voltage output (VREFOPAMP4).	7

Reference Manual:

https://www.st.com/resource/en/reference_manual/dm00043574-stm32f303xb-c-d-e-stm32f303xc-6-8-stm32f328x8-stm32f358xc-stm32f398xe-advanced-arm-based-mcus-stmicroelectronics.pdf

ADC CHANNEL INPUT MODE

ADC channel inputs can be configured in single-ended or differential mode

Single-ended mode (thường sử dụng)

A single-ended ADC designed to be connected to the same ground level as the microcontroller and to provide their measurement result as an analog voltage signal on a single wire referenced to this common ground

Differential mode

A differential ADC measures the voltage difference two inputs between .

Ref Differential and Single-Ended ADC:

<https://ww1.microchip.com/downloads/en/DeviceDoc/Differential-and-Single-Ended-ADC-WhitePaper-DS00003197A.pdf>

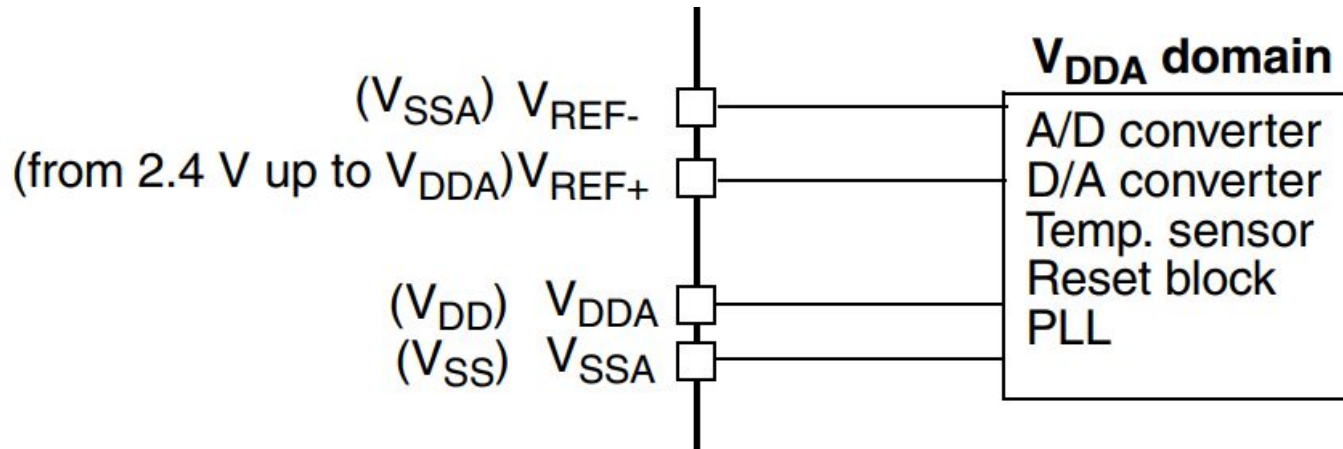
Ref STM32F30x ADC modes and application:

https://www.st.com/resource/en/application_note/dm00069390-stm32f30x-adc-modes-and-application-stmicroelectronics.pdf

https://www.st.com/resource/en/application_note/cd00211314-how-to-get-the-best-adc-accuracy-in-stm32-microcontrollers-stmicroelectronics.pdf

ADC RESOLUTION

12, 10, 8 or 6-bit configurable resolution



$$\begin{aligned}
 V_{IN} = V_{REF-} &\longrightarrow N_{ADC} = 0 \\
 V_{IN} = V_{REF+} &\longrightarrow N_{ADC} = 2^{12} - 1 \\
 V_{REF-} \leq V_{IN} \leq V_{REF+} &\longrightarrow 0 \leq N_{ADC} \leq 2^{12} - 1
 \end{aligned}$$

$$N_{ADC} = (2^{12} - 1) \times \frac{V_{IN} - V_{REF-}}{V_{REF+} - V_{REF-}} \quad \longrightarrow \quad V_{IN} = \frac{N_{ADC} \times V_{DD}}{4095}$$

*16bit data register -> DataAlignment (right alignment)

Bài 1:

ADC: 12 bit

$V_{ref+} = V_{DDA} = V_{DD} = 3.3v$

$V_{ref-} = V_{SSA} = V_{SS} = 0v$

Biết kết quả chuyển đổi nADC = 0x1FF

Tính điện áp ngõ vào trên ADC Input channel?

Bài 2:

ADC: 10 bit

$V_{ref+} = V_{DDA} = V_{DD} = 3.3v$

$V_{ref-} = V_{SSA} = V_{SS} = 0v$

Biết kết quả chuyển đổi nADC = 511

Tính điện áp ngõ vào trên ADC Input channel?

Independent modes

Regular conversion mode

Injected conversion mode

Dual modes

Dual injected simultaneous mode

Dual regular simultaneous mode

Dual interleaved mode

Dual alternate trigger mode

Ref STM32F30x ADC modes and application:

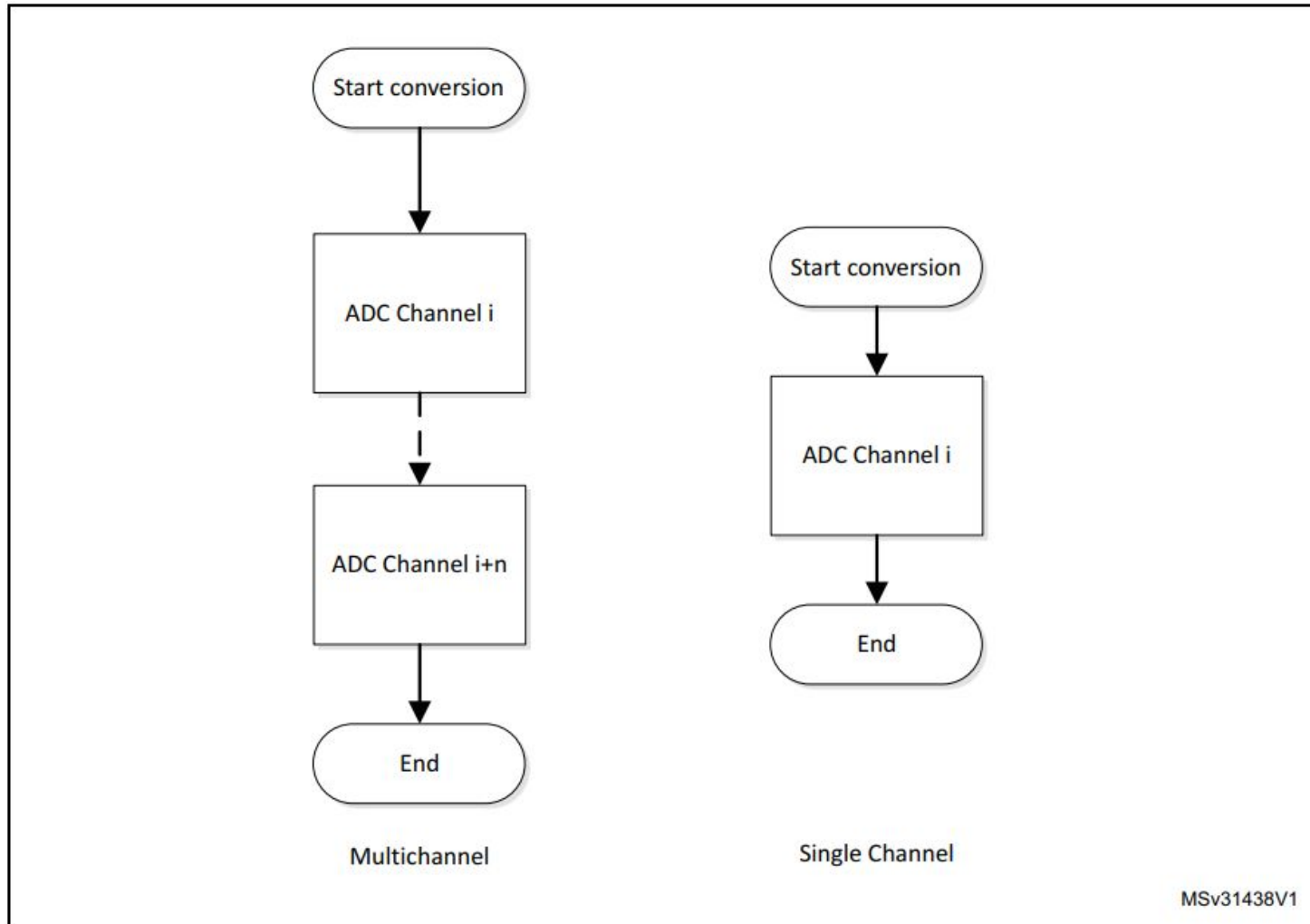
https://www.st.com/resource/en/application_note/dm00069390-stm32f30x-adc-modes-and-application-stmicroelectronics.pdf

STM32F30x ADC modes

Independent modes

Regular conversion mode

Multichannel/single, single conversion mode

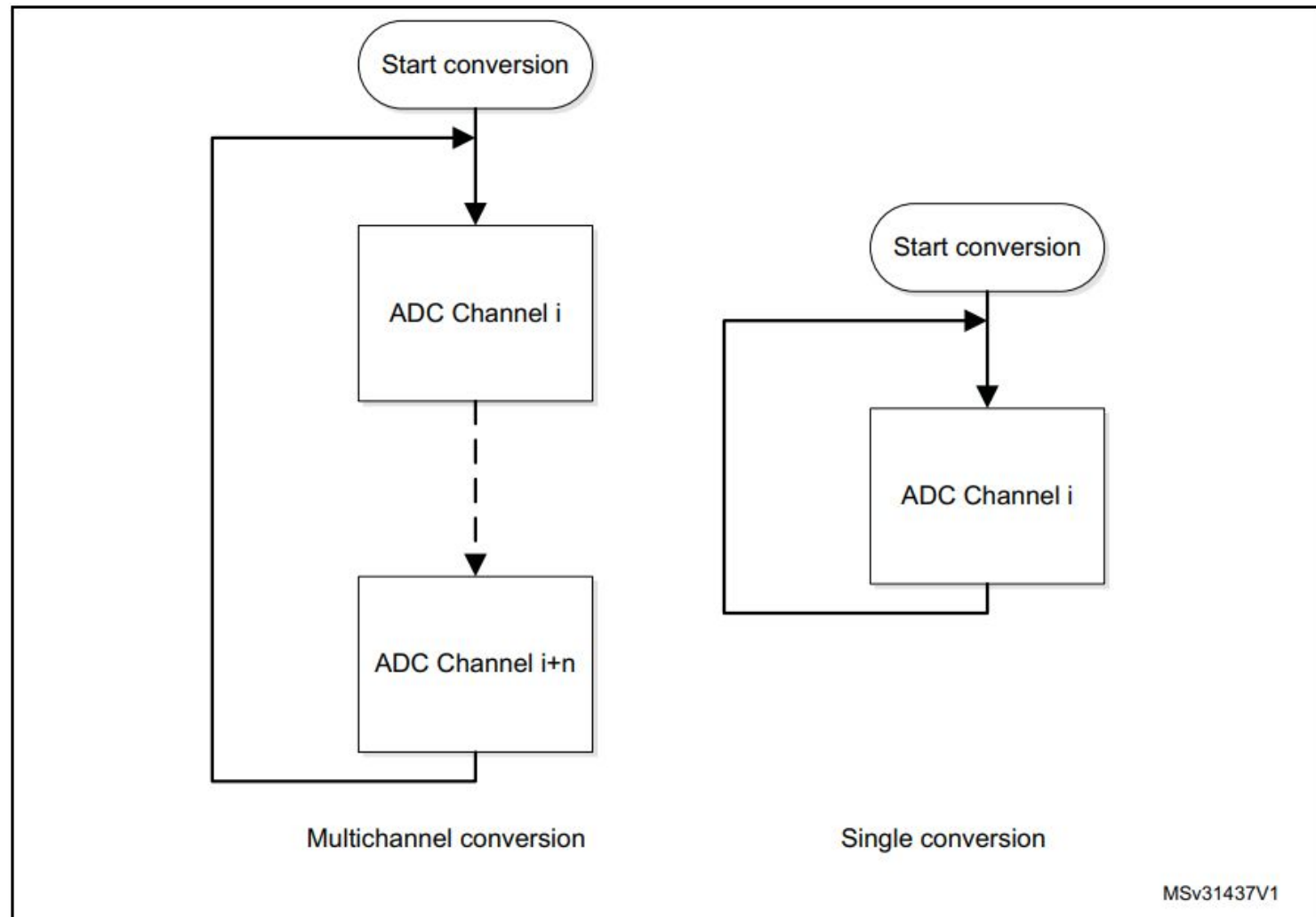


STM32F30x ADC modes

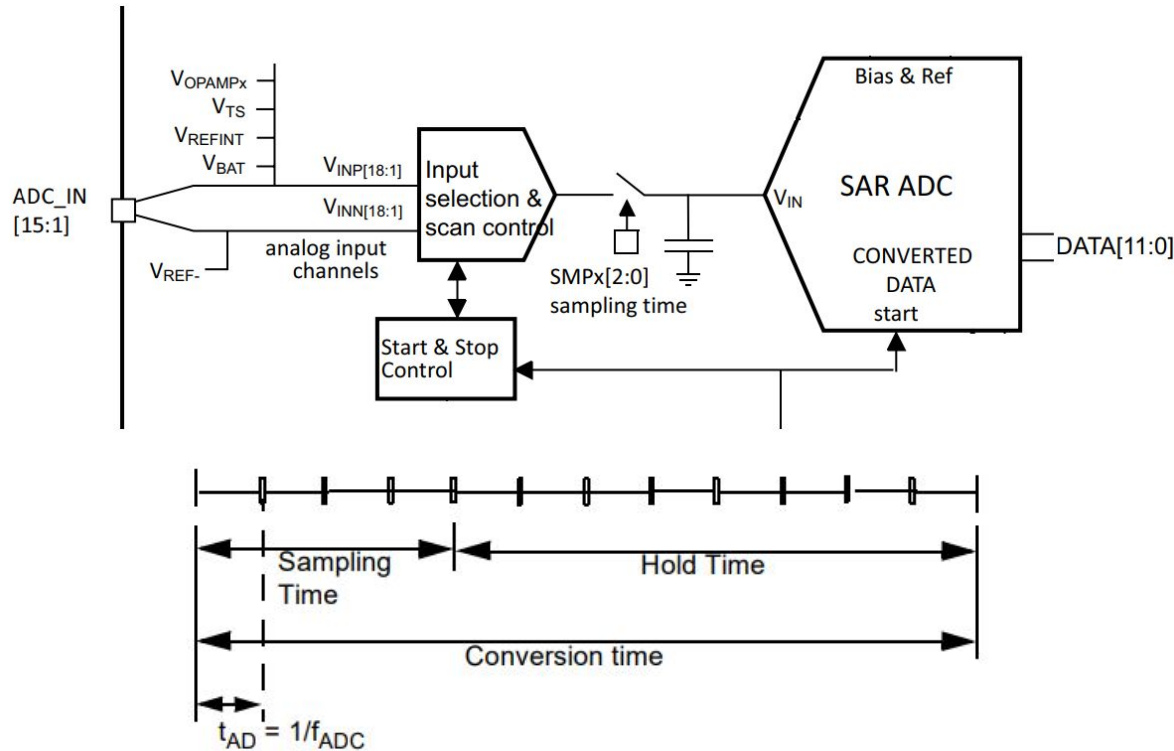
Independent modes

Regular conversion mode

Multichannel/Single, continuous mode



ADC CONVERSION TIME



The elapsed time between the start of a conversion and the end of conversion is the sum of the configured sampling time plus the successive approximation time depending on data resolution:

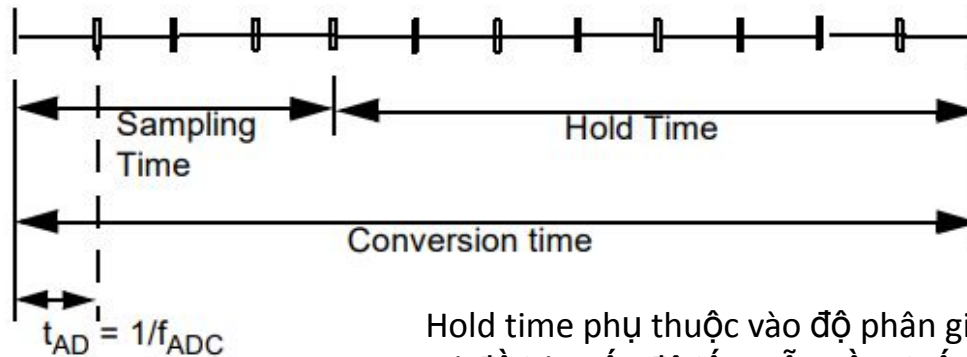
$$T_{ADC} = T_{SMPL} + T_{SAR} = [1.5 |_{min} + 12.5 |_{12bit}] \times T_{ADC_CLK}$$

$$T_{ADC} = T_{SMPL} + T_{SAR} = 20.83 \text{ ns } |_{min} + 173.6 \text{ ns } |_{12bit} = 194.4 \text{ ns (for } F_{ADC_CLK} = 72 \text{ MHz)}$$

Ref manual:

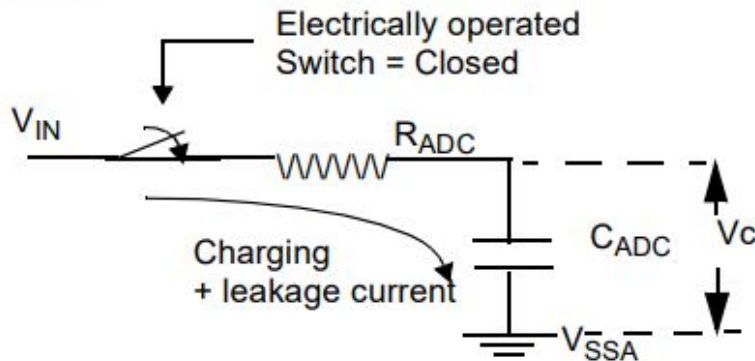
https://www.st.com/resource/en/reference_manual/dm00043574-stm32f303xb-c-d-e-stm32f303x6-8-stm32f328x8-stm32f358xc-stm32f398xe-advanced-arm-based-mcus-stmicroelectronics.pdf

ADC SAMPLING TIME

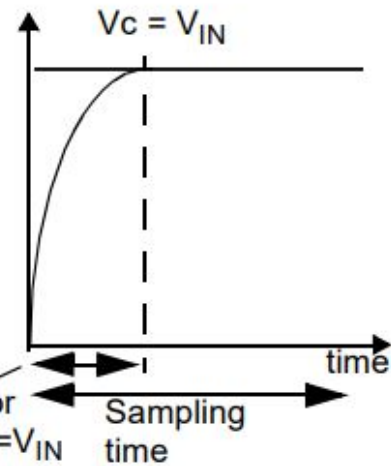


Hold time phụ thuộc vào độ phân giải.
Có đề bài: tốc độ lấy mẫu cần thiết

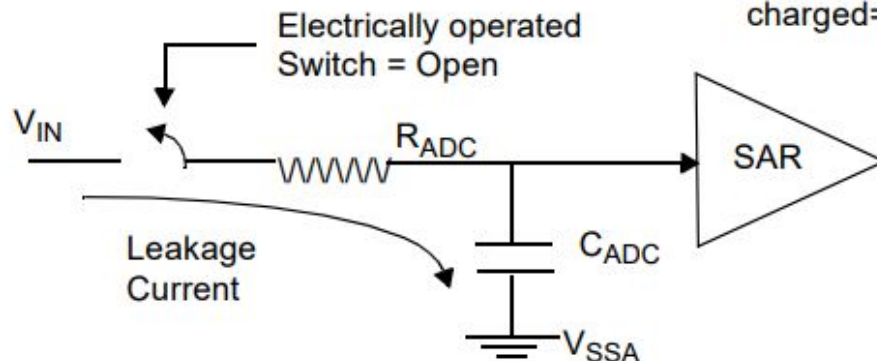
Sampling



V_C = Voltage developed across capacitor.



Hold and Conversion



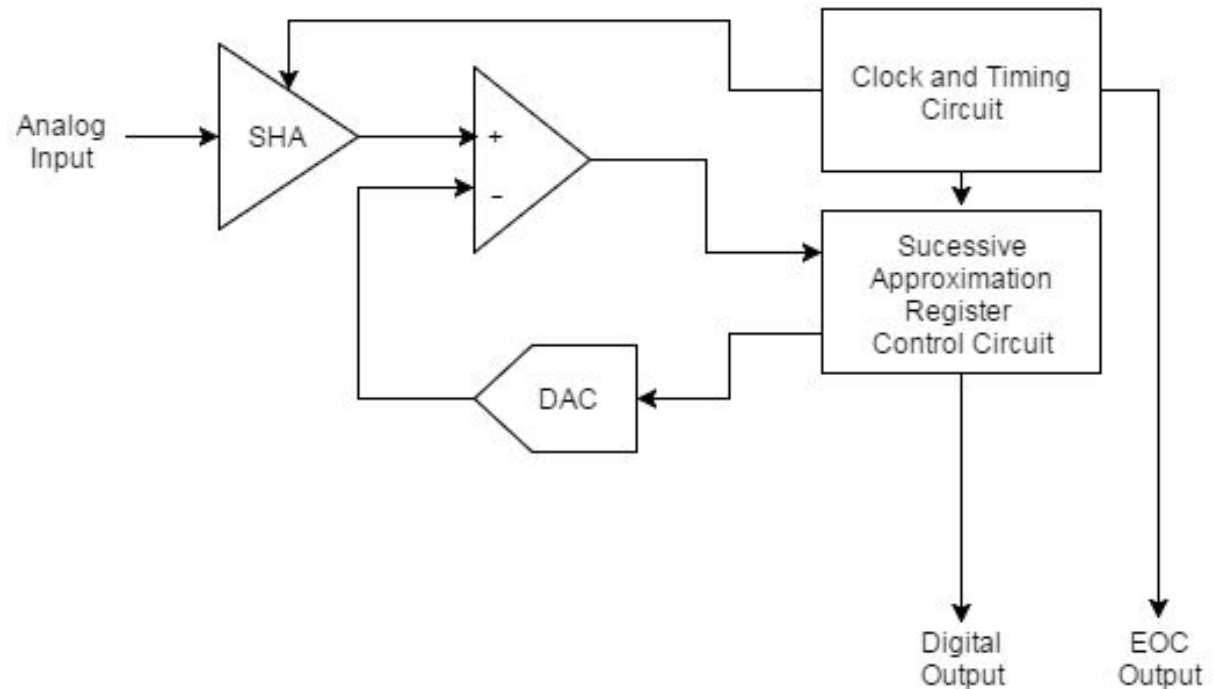
ADC SAR

<https://www.allaboutcircuits.com/technical-articles/understanding-analog-to-digital-converters-the-successive-approximation-reg/>

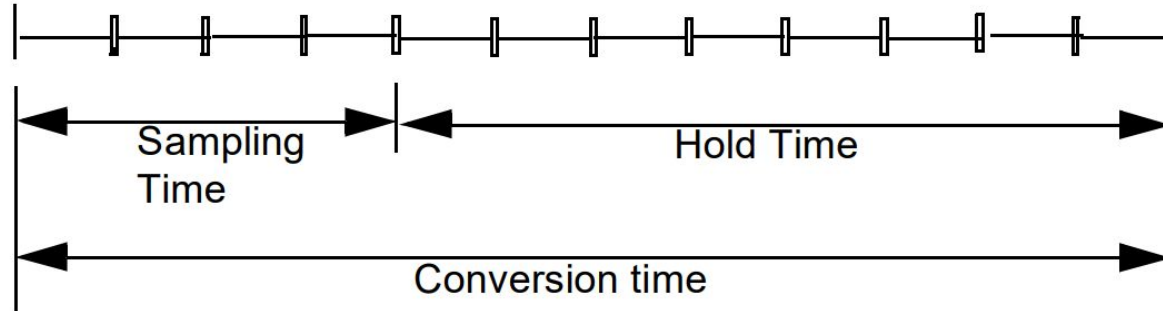
$V_{ref} = 3.3V$

6 bit

1 V.



Get ADC Value



- Polling

```
HAL_ADC_Start(ADC_HandleTypeDef* hadc)
```

```
HAL_ADC_PollForConversion(ADC_HandleTypeDef* hadc, uint32_t Timeout) uint32_t
```

```
HAL_ADC_GetValue(ADC_HandleTypeDef* hadc)
```

```
HAL_ADC_Stop(ADC_HandleTypeDef* hadc)
```

- Interrupt generation at End of Conversion, End of Injected

```
HAL_ADC_Start_IT(ADC_HandleTypeDef* hadc)
```

```
HAL_ADC_Stop_IT(ADC_HandleTypeDef* hadc)
```

- DMA request generation during regular channel conversion

```
HAL_ADC_Start_DMA(ADC_HandleTypeDef* hadc, uint32_t* pData, uint32_t Length)
```

```
HAL_ADC_Stop_DMA(ADC_HandleTypeDef* hadc)
```

```
void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef* hadc)
```

```
{
```

```
..... HAL_ADC_GetValue...
```

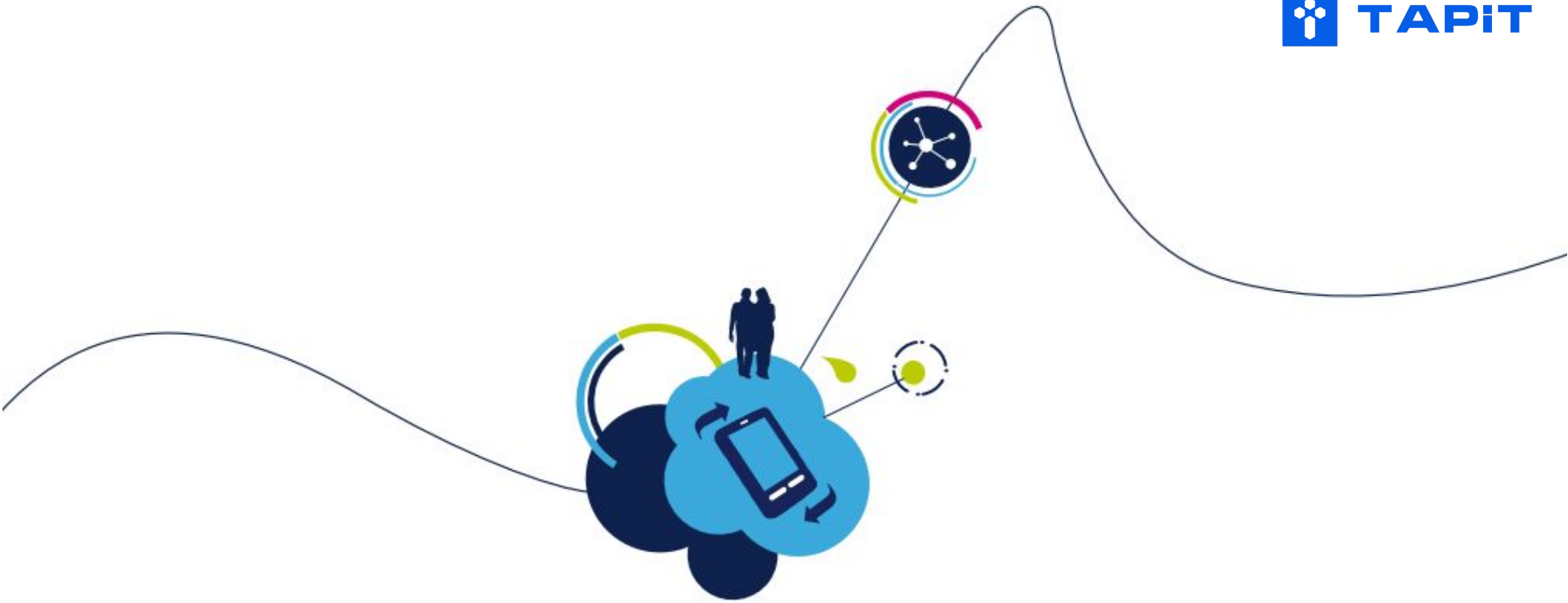
```
}
```

Tính toán số mẫu / 1 giây ADC

1. Xác định tần số cấp cho ngoại vi ADC
block diagram datasheet, xác định bus?
clock config ở giao diện cubemx, xem tần số bus
bao nhiêu?
1. Xác định thời gian lấy mẫu T_s
Xem cấu hình adc, chỗ rank, sampling time.
1. Xác định thời gian giữ và chuyển đổi mẫu T_c
(12bit)
2. $T_1 = T_s + T_c$
3. $1/T_1$

Tính toán số mẫu / 1 giây ADC

1. Xác định tần số cấp cho ngoại vi ADC 16MHz
2. Xác định thời gian lấy mẫu T_s 601,5 C
3. Xác định thời gian giữ và chuyển đổi mẫu T_c 12.5
4. $T_1 = T_s + T_c \quad (601.5 + 12.5) * 1 / (16.10^6)$
5. $1/T_1 \sim 26\text{ksp/s}$



ADC Poll lab

How to setup ADC in poll in CubeMX and Generate Code
How to Generate Code in CubeMX and use HAL functions

Use ADC in polling mode

```
HAL_ADC_Start(DAC_HandleTypeDef* hdac, uint32_t Channel)  
HAL_ADC_PollForConversion(ADC_HandleTypeDef* hadc, uint32_t Timeout)  
HAL_ADC_GetValue(ADC_HandleTypeDef* hadc)
```

Use ADC in polling mode

```
/* USER CODE BEGIN PV */  
    uint32_t value_adc;  
/* USER CODE END PV */
```

```
/* USER CODE BEGIN 3 */  
    /* Infinite loop */  
    while (1)  
    {  
        HAL_ADC_Start(&hadc1);  
        HAL_ADC_PollForConversion(&hadc1,10);  
        value_adc=HAL_ADC_GetValue(&hadc1);  
        HAL_Delay(1000);  
    }  
/* USER CODE END 3 */
```

=>Debug

Tính toán giá trị điện áp trên chân ngõ vào ADC

In giá trị nADC

In giá trị Vin

lên serial UART.

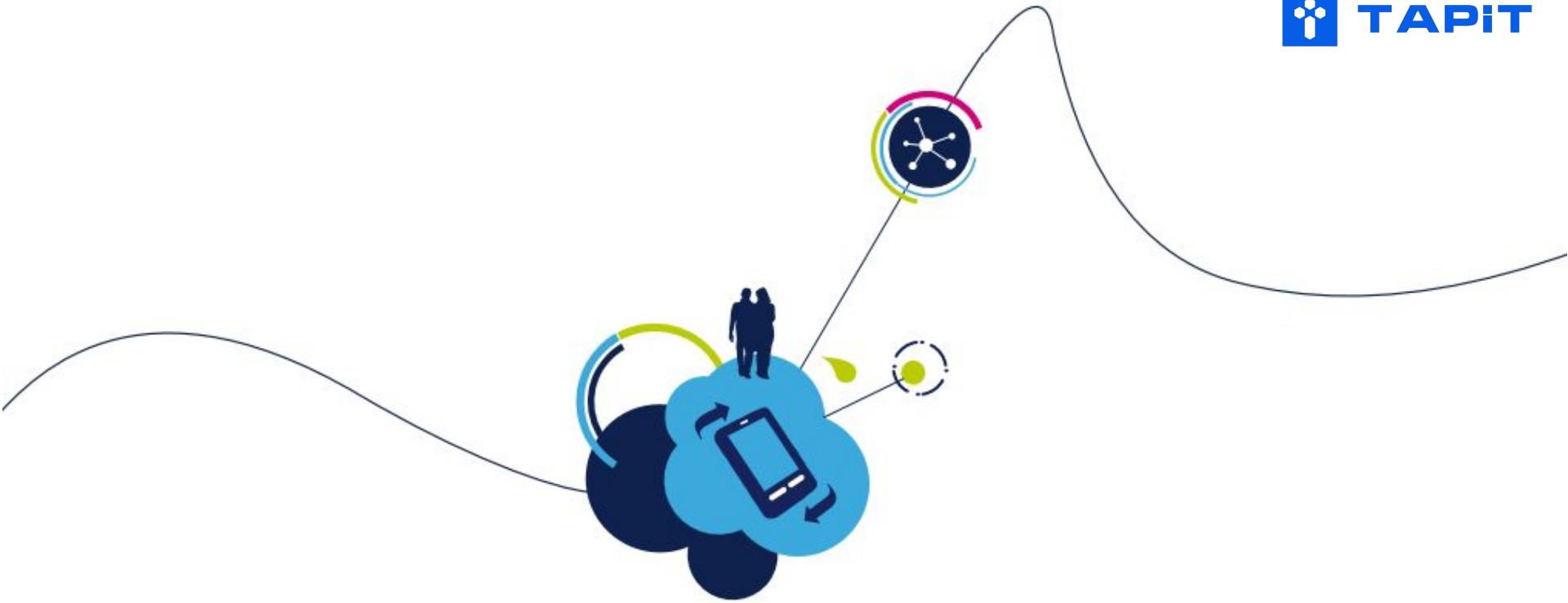
Và nếu điện áp $< 1V$: Tắt đèn trên mạch

Nếu điện áp $> 2V$: Bật đèn

Trong vi điều khiển stm32 có cảm biến nhiệt độ tích hợp sẵn. Thực hiện:

1. Xác định ngoại vi ADC nào (ADC1, ADC2, ...) có cảm biến nhiệt độ.
2. Cảm biến nhiệt độ được thiết kế ở Input channel mấy?
3. Xác định công thức tính nhiệt độ từ nADC.
4. Cấu hình, viết code, debug để xem mcu đang có nhiệt độ là bao nhiêu.

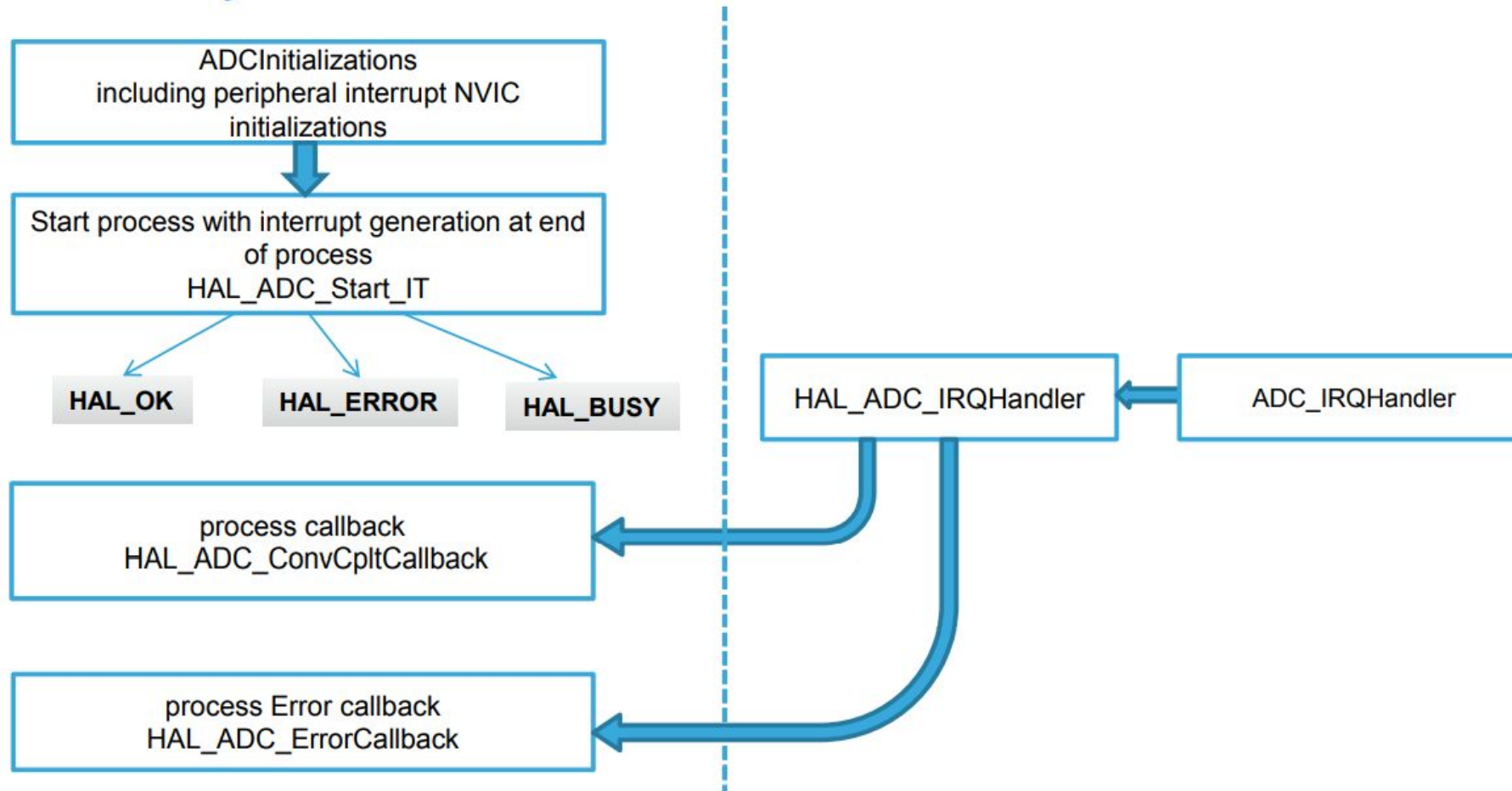
Tài liệu cần tham khảo:
STM32... reference manual
STM32... datasheet.



ADC Interrupt lab

How to setup ADC in interrupt in CubeMX and Generate Code
How to Generate Code in CubeMX and use HAL functions

HAL Library ADC with IT flow



For ADC start use function

- `HAL_ADC_Start_IT(ADC_HandleTypeDef* hadc, uint32_t Channel)`

ADC complete callback function

- `HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef* hadc)`
 - `HAL_ADC_GetValue(ADC_HandleTypeDef* hadc)`

```
/* USER CODE BEGIN PV */
    uint32_t value_adc;
    uint32_t value_dac=0;
/* USER CODE END PV */

/* USER CODE BEGIN 2 */
    HAL_ADC_Start_IT(&hadc1);
/* USER CODE END 2 */

/* USER CODE BEGIN 4 */
void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef* hadc)
{
    value_adc=HAL_ADC_GetValue(&hadc1);
    HAL_ADC_Start_IT(&hadc1);
}
/* USER CODE END 4 */
```

=>Debug

- B1. Xác định ADC nào có cảm biến nhiệt độ nội (internal input channel, temperature sensor)
- B2. TempSensor nằm ở INput channel nào? Cấu hình chọn làm ngõ vào của ADC
- B3. Cài đặt ADC liên tục, ngắt để đọc giá trị nadc từ kênh cảm biến nhiệt độ
- B4. Tính toán giá trị nhiệt độ từ giá trị nadc. (Dựa vào tài liệu của hãng)
- B5. Debug xem kết quả.
- B6. Nhấn kết quả vào phần khung chat.

Nhiều kênh

BTVN: Có 3 kênh ngõ vào ADC.

Thực hiện 2 project chuyển đổi 3 kênh liên tục (scanmode - cont..) ở chế độ polling và chế độ Interrupt. Lưu kết quả vào 1 mảng có 3 phần tử (hoặc 3 biến khác nhau), tương ứng với 3 kênh.

Thử kết quả bằng cách lần lượt để 2 kênh hở mạch và 1 kênh nối GND/VCC?

Hướng dẫn cấu hình:

Chọn 3 kênh ngõ vào (single ended), cấu hình number of conversion:3 , cấu hình cho từng rank thứ tự chuyển đổi các kênh khác nhau, sampling time max.)

KHÓA HỌC LẬP TRÌNH VI ĐIỀU KHIỂN

Giảng viên

**NGUYỄN HUỲNH NHẬT
TRẦN THỤY NGỌC HẠNG**

LỊCH HỌC:

Tại Đà Nẵng: 18h30 - 22h00 thứ 2 và thứ 4

ĐỊA ĐIỂM:

Tại Đà Nẵng: Số 32 Hòa Minh 16, Q. Liên Chiểu

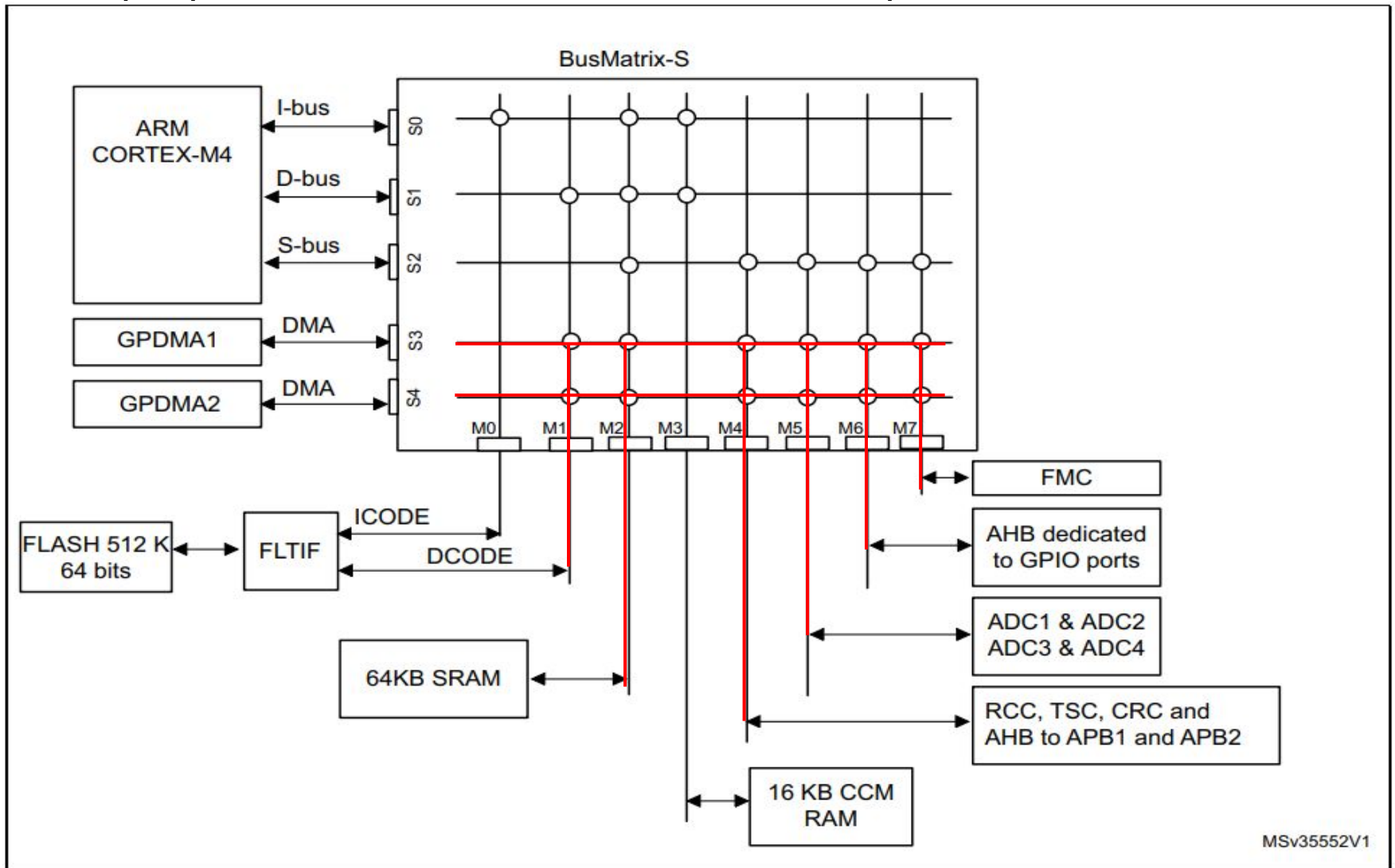
MODULE 7

- DMA
- ADC DMA
- PHÂN TÍCH DỰ ÁN THỰC TẾ



DMA

- Direct Memory Access



DMA

- Các thông số:
 - ☐ Địa chỉ nguồn, địa chỉ đích
 - ☐ Kiểu dữ liệu của mỗi lần truyền (data width): byte, half-word, word
 - ☐ Lượng dữ liệu truyền (transfer size)
 - ☐ Kiểu truyền dữ liệu (transfer types): Normal mode hoặc Circular mode (ring buffer)
 - ☐ Tự động tăng địa chỉ nguồn/địa chỉ đích hoặc cả 2 sau mỗi lần truyền dữ liệu
 - ☐ Yêu cầu ngắt: Half-transfer hoặc Transfer complete (double buffering)

DMA

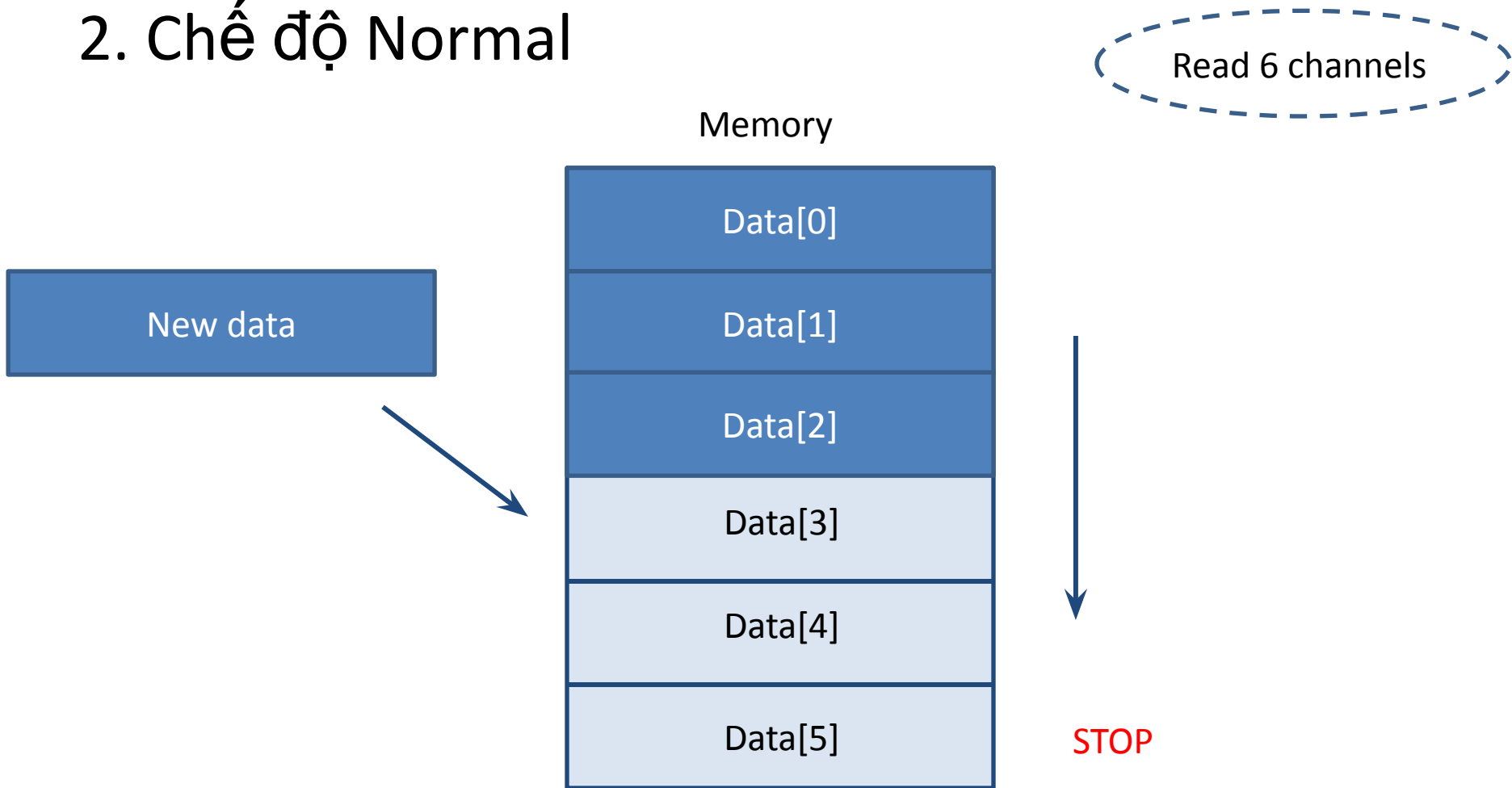
- Ưu điểm
 - ☐ DMA truyền dữ liệu nhanh hơn CPU
 - ☐ Nâng cao hiệu suất của vi điều khiển
 - ☐ DMA có khả năng tự động tăng địa chỉ lưu trữ dữ liệu
 - ☐ Tiết kiệm năng lượng tiêu thụ với các ứng dụng truyền nhận một lượng lớn dữ liệu
- Nhược điểm
 - ☐ Phần cứng VĐK phải hỗ trợ DMA (STM32 all)
 - ☐ Sử dụng DMA không phù hợp thì có khả năng tăng năng lượng tiêu thụ

https://www.st.com/resource/en/application_note/dm00046011-using-the-stm32f2-stm32f4-and-stm32f7-series-dma-controller-stmicroelectronics.pdf

Round-robin priority scheme

CÁC CHẾ ĐỘ HOẠT ĐỘNG CỦA DMA

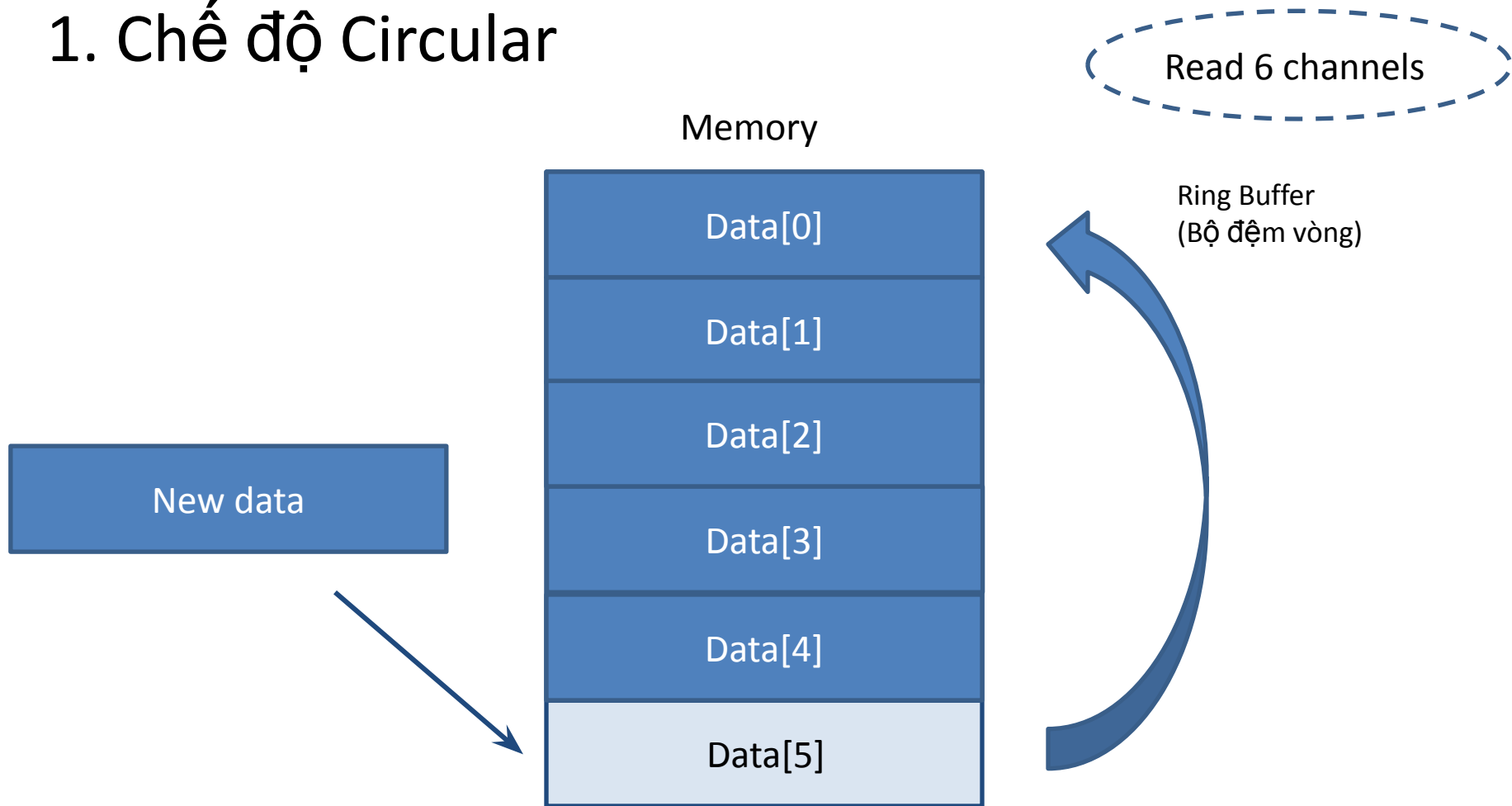
2. Chế độ Normal



Khi DMA dừng hoạt động, cần gọi lại lệnh bắt đầu DMA để tiếp tục truyền dữ liệu lại từ đầu

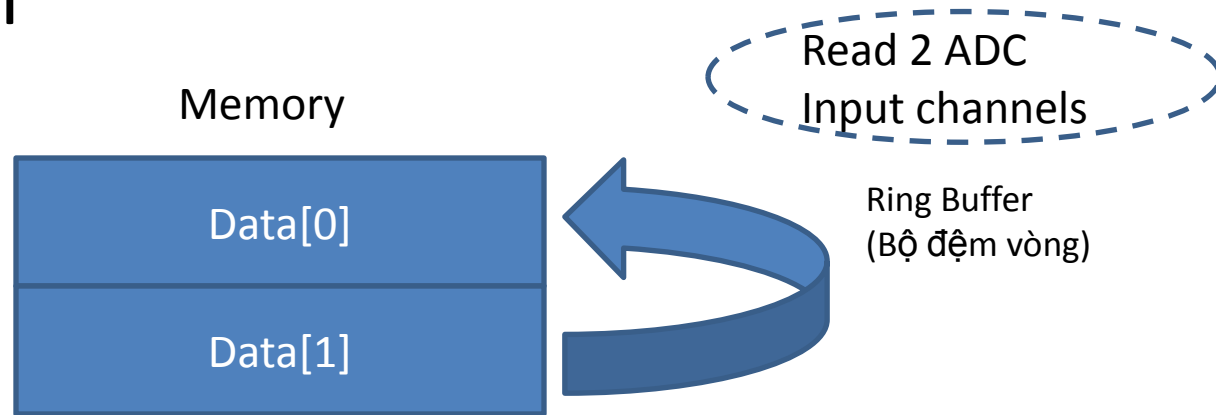
CÁC CHẾ ĐỘ HOẠT ĐỘNG CỦA DMA

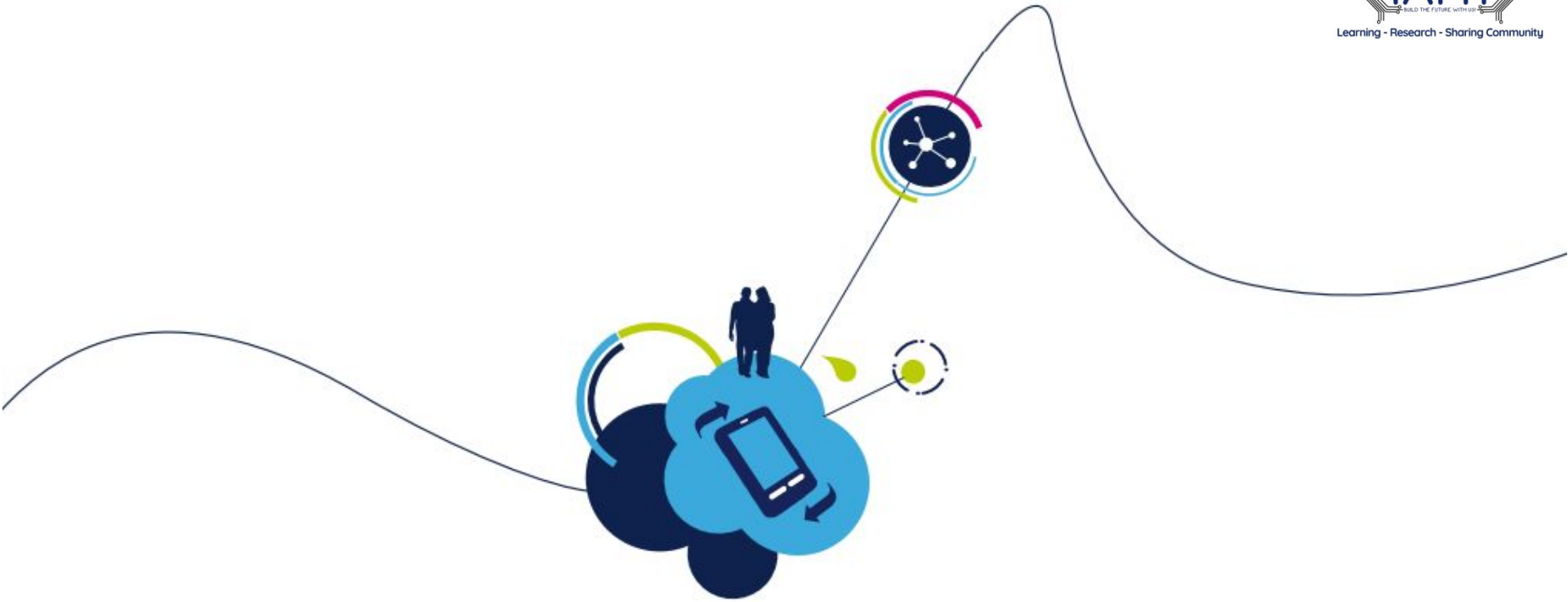
1. Chế độ Circular



CÁC CHẾ ĐỘ HOẠT ĐỘNG CỦA DMA

1. Chế độ Circular





ADC DMA Poll lab

How to setup ADC DMA in CubeMX and Generate Code
How to Generate Code in CubeMX and use HAL functions

CẤU HÌNH DMA QUA GIAO DIỆN CUBEMX

- ADC: Multichannel (2 channel) , continuous mode
- Lưu mẫu mới nhất vào địa chỉ một cách tự động bằng DMA

The screenshot displays the STM32CubeMX software interface, specifically the 'Pinout & Configuration' tab. The left sidebar shows the 'Categories' list with 'ADC1' selected under the 'Analog' category. The main area is divided into 'ADC1 Mode and Configuration' and 'Configuration' sections.

ADC1 Mode and Configuration:

- Mode:** IN1 and IN2 are set to 'Single-ended' (highlighted with a red box). IN3 is 'Disable'. IN4, IN6, IN7, and IN8 are also 'Disable'. IN9 and IN14 are 'Single-ended'. Temperature Sensor Channel, Vrefint Channel, Vbat Channel, and VRefOpamp1 Channel are all 'Disable'. EXT1 Conversion Trigger is 'Disable'.

Configuration:

- Reset Configuration:** NVIC Settings, DMA Settings, and GPIO Settings are all checked.
- Parameter Settings:** Search (Ctrl+F) is available.
- ADCs_Common_Settings:** Mode is set to 'Independent mode'.
- ADC_Settings:**
 - Clock Prescaler: ADC Asynchronous clock mode
 - Resolution: ADC 12-bit resolution
 - Data Alignment: Right alignment
 - Scan Conversion Mode: Enabled (highlighted with a red box)
 - Continuous Conversion Mode: Enabled (highlighted with a red box)
 - Discontinuous Conversion Mode: Disabled
 - DMA Continuous Requests: Enabled (highlighted with a red box)
 - End Of Conversion Select...: End of single conversion
 - Overrun behaviour: Overrun data overwritten
 - Low Power Auto Wait: Disabled

Pinout view: The right side of the interface shows the pinout view of the STM32F303RETx LQFP64 package. The pins are labeled with their names and functions. The ADC1_IN1 and ADC1_IN2 pins are highlighted in green, corresponding to the configuration in the left panel.

CẤU HÌNH DMA QUA GIAO DIỆN CUBEMX

- ADC: Rank number, Sampling Time

The screenshot displays the STM32CubeMX software interface for configuring an STM32F303RETx LQFP64 microcontroller. The interface is divided into several panes:

- Pinout & Configuration:** The left sidebar shows a tree view of components. Under 'Analog', 'ADC1' is selected.
- ADC1 Mode and Configuration:** The central pane shows the configuration for ADC1. The 'Mode' section lists input channels (IN1 to IN8) and their settings. The 'Configuration' section includes a 'Reset Configuration' button and tabs for 'NVIC Settings', 'DMA Settings', and 'GPIO Settings'. The 'Parameter Settings' tab is active, showing a table of ADC parameters.
- Pinout view:** The right pane shows a pinout diagram of the STM32F303RETx LQFP64 package. Pins are color-coded: green for digital, yellow for analog, and blue for power. Pins PA0 and PA1 are highlighted in green, indicating they are configured as digital outputs.

The 'Parameter Settings' table for ADC1 is as follows:

Rank	Channel	Sampling Time	Offset Number	Offset
1	Channel 1	601.5 Cycles	No offset	0
2	Channel 2	601.5 Cycles	No offset	0

CẤU HÌNH DMA QUA GIAO DIỆN CUBEMX

- DMA: Mode Circular, Data width: Half-word

The screenshot displays the STM32CubeMX configuration tool. On the left, the 'Categories' list includes 'System Core', 'Analog', 'Timers', 'Connectivity', 'Multimedia', 'Computing', and 'Middleware'. Under 'Analog', 'ADC1' is selected. The main window shows the 'ADC1 Mode and Configuration' section with various input channels (IN1 to IN14) and conversion triggers. Below this, the 'Configuration' section is active, with the 'DMA Settings' tab highlighted. A table lists the DMA configuration:

DMA Request	Channel	Direction	Priority
ADC1	DMA1 Channel 1	Peripheral To Memory	Low

Below the table, the 'DMA Request Settings' section shows the 'Mode' set to 'Circular' and 'Data Width' set to 'Half Word'. The 'Increment Address' checkbox is unchecked, and the 'Memory' checkbox is checked.

Use ADC in DMA mode

```
HAL_ADC_Start_DMA(ADC_HandleTypeDef* hadc, uint32_t* pData, uint32_t Length)  
HAL_ADC_Stop_DMA(ADC_HandleTypeDef* hadc)
```

```
/* USER CODE BEGIN PV */  
    uint16_t adc_value[2] = {0};  
/* USER CODE END PV */
```

```
MX_DMA_Init();  
MX_ADC1_Init();
```

```
/* USER CODE BEGIN 2 */  
HAL_ADC_Start_DMA(&hadc1, (uint32_t*)adc_value, 2);  
    /* Infinite loop */  
while (1)  
{  
}  
/* USER CODE END 3 */
```

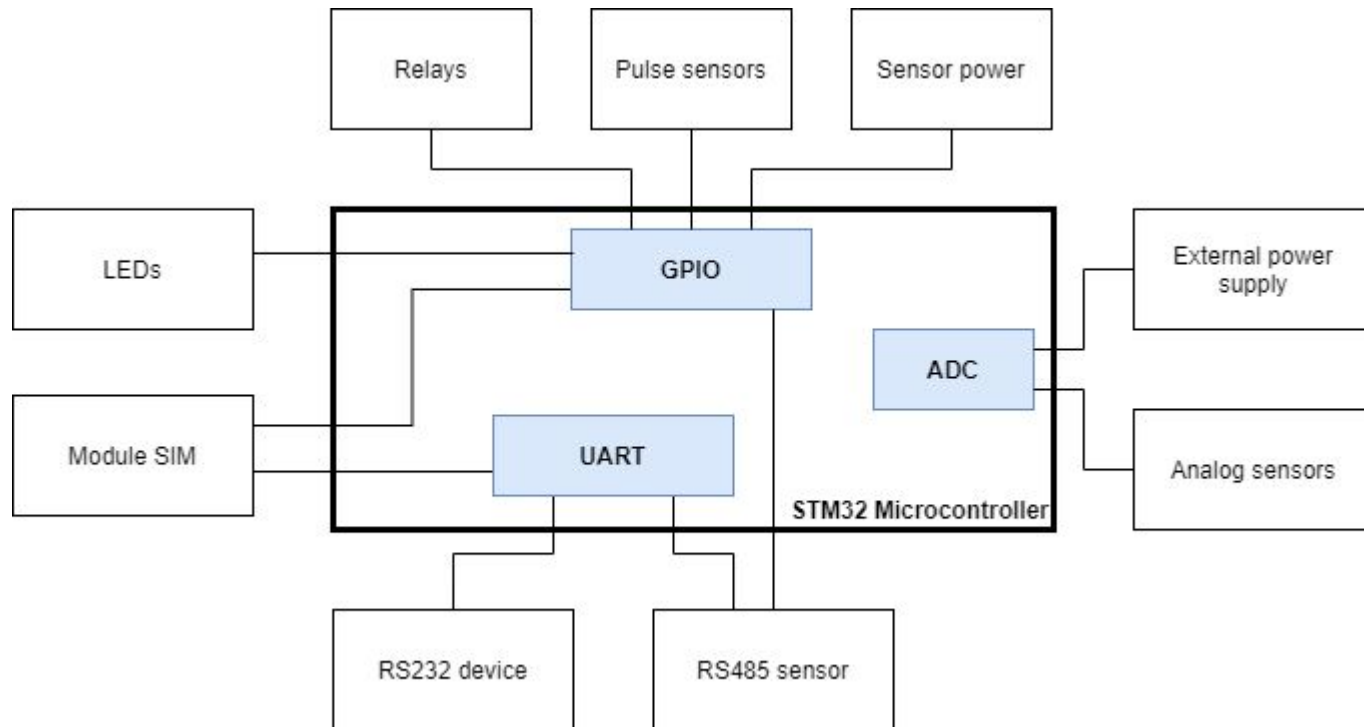
Tài liệu How to get the best ADC accuracy in STM32 microcontrollers

https://www.st.com/resource/en/application_note/cd00211314-how-to-get-the-best-adc-accuracy-in-stm32-microcontrollers-stmicroelectronics.pdf

PHÂN TÍCH DỰ ÁN THỰC TẾ

KTTV - DATALOGGER

Hardware Block Diagram



Yêu cầu dự án

- Cứ tròn 10 phút thiết bị sẽ thu thập và tổng hợp các số liệu sau và gửi lên Server:
 - Giá trị mực nước đo được từ cảm biến siêu âm, tín hiệu analog current
 - Giá trị lượng mưa đo được từ cảm biến đo mưa kiểu chao lật, tín hiệu xung
 - Giá trị điện áp cung cấp cho hệ thống, tín hiệu analog voltage
 - Trạng thái của cảm biến: OK hoặc ERROR

- Báo hiệu trạng thái của hệ thống thông qua LED STATUS trên thiết bị
 - LED nhấp nháy 10 giây 1 lần: Hệ thống hoạt động bình thường
 - LED nhấp nháy 3 giây 1 lần: Thiết bị không kết nối được Internet
 - LED nhấp nháy 1 giây 1 lần: Lỗi cảm biến

- Khởi động lại thiết bị từ xa thông qua tin nhắn SMS

- Khởi động lại thiết bị vào lúc 0h4p hằng ngày

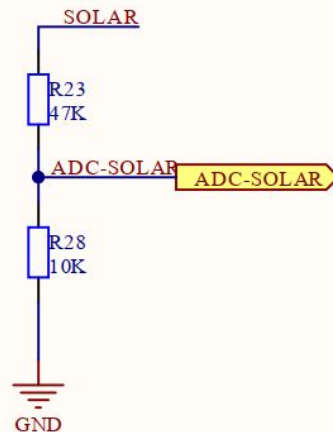
- Tiết kiệm năng lượng tiêu thụ của hệ thống:
 - Đưa Module SIM về trạng thái Sleep Mode
 - Tắt nguồn cung cấp cho các cảm biến (Relay, FET)

NHỮNG TÍNH NĂNG LIÊN QUAN ĐẾN NGOẠI VI ADC

ĐỌC GIÁ TRỊ ĐIỆN ÁP CUNG CẤP

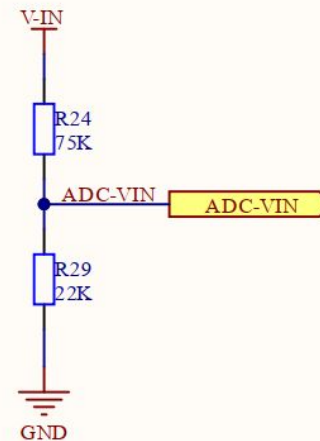
ADC SOLAR PANEL

V-SOLAR = 18V
V-ADC = V-SOLAR * R28 / (R28+R23)
= 3.16V



ADC-VIN

VIN=14.4V -> VOUT=3.26V
VIN=10.8V -> VOUT=2.45V



$$V_{ADC-VIN} = \frac{N_{ADC} \times VDD}{(2^{12} - 1)} \longrightarrow V_{IN} = \frac{V_{ADC-VIN} \times R29}{(R29 + R24)}$$

$$V_{ADC-SOLAR} = \frac{N_{ADC} \times VDD}{(2^{12} - 1)} \longrightarrow V_{SOLAR} = \frac{V_{ADC-SOLAR} \times R28}{(R28 + R23)}$$

ĐỌC GIÁ TRỊ CẢM BIẾN ANALOG 4-20mA

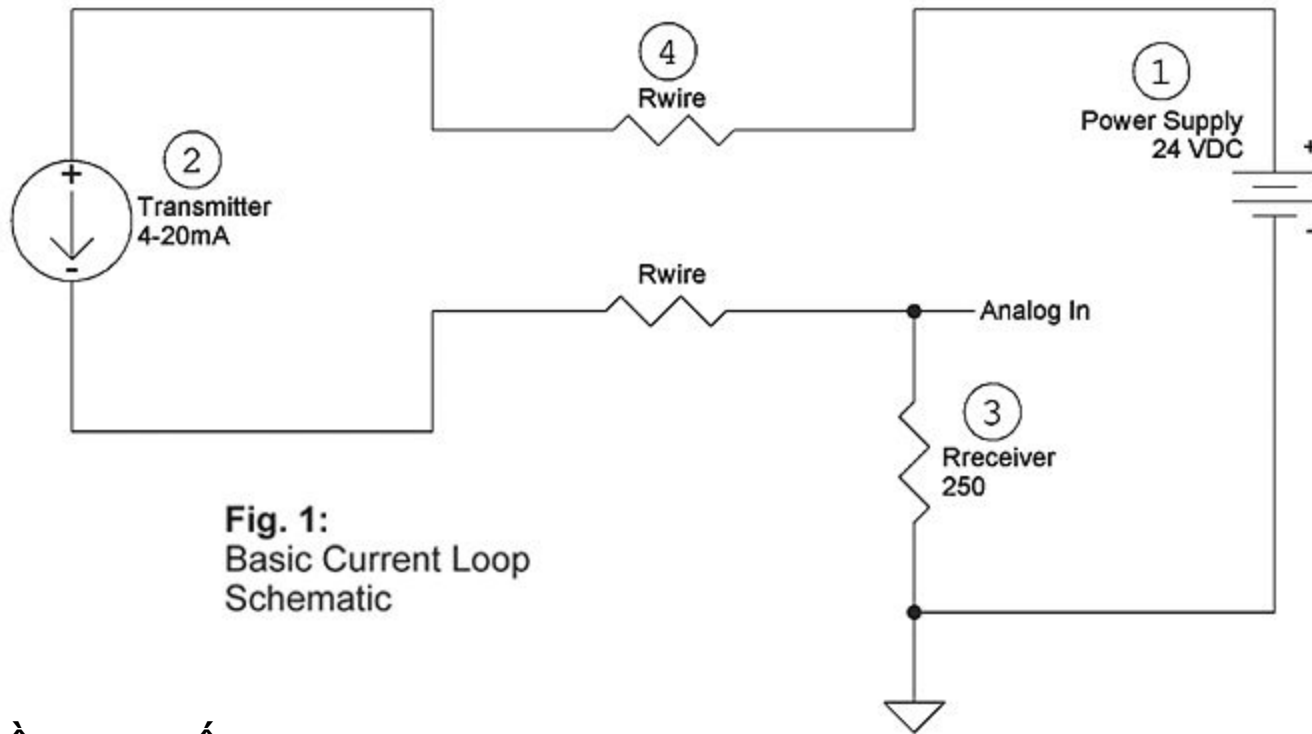


Fig. 1:
Basic Current Loop
Schematic

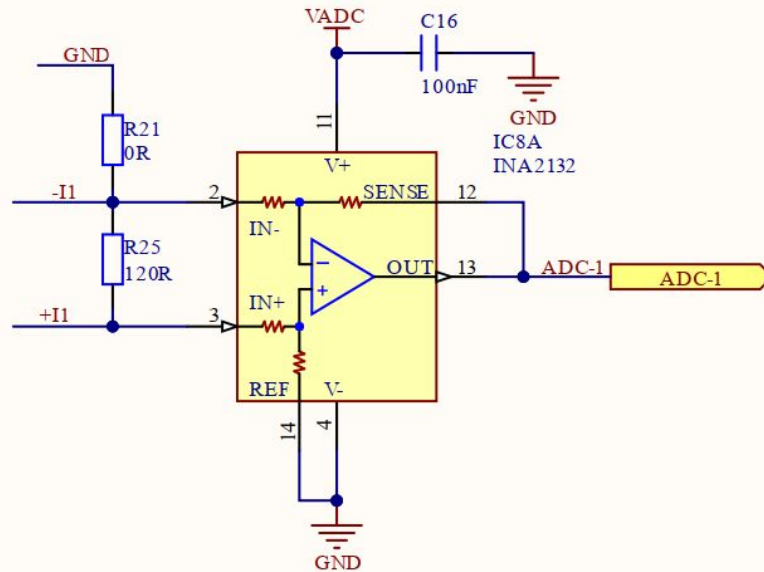
1. Nguồn cung cấp
2. Nguồn dòng 4-20mA
3. Trở đóng vai trò chuyển đổi dòng qua áp
4. Trở đóng vai trò là dây dẫn

ĐỌC GIÁ TRỊ CẢM BIẾN ANALOG 4-20mA

Giá trị mực nước đo được từ cảm biến siêu âm, tín hiệu analog current

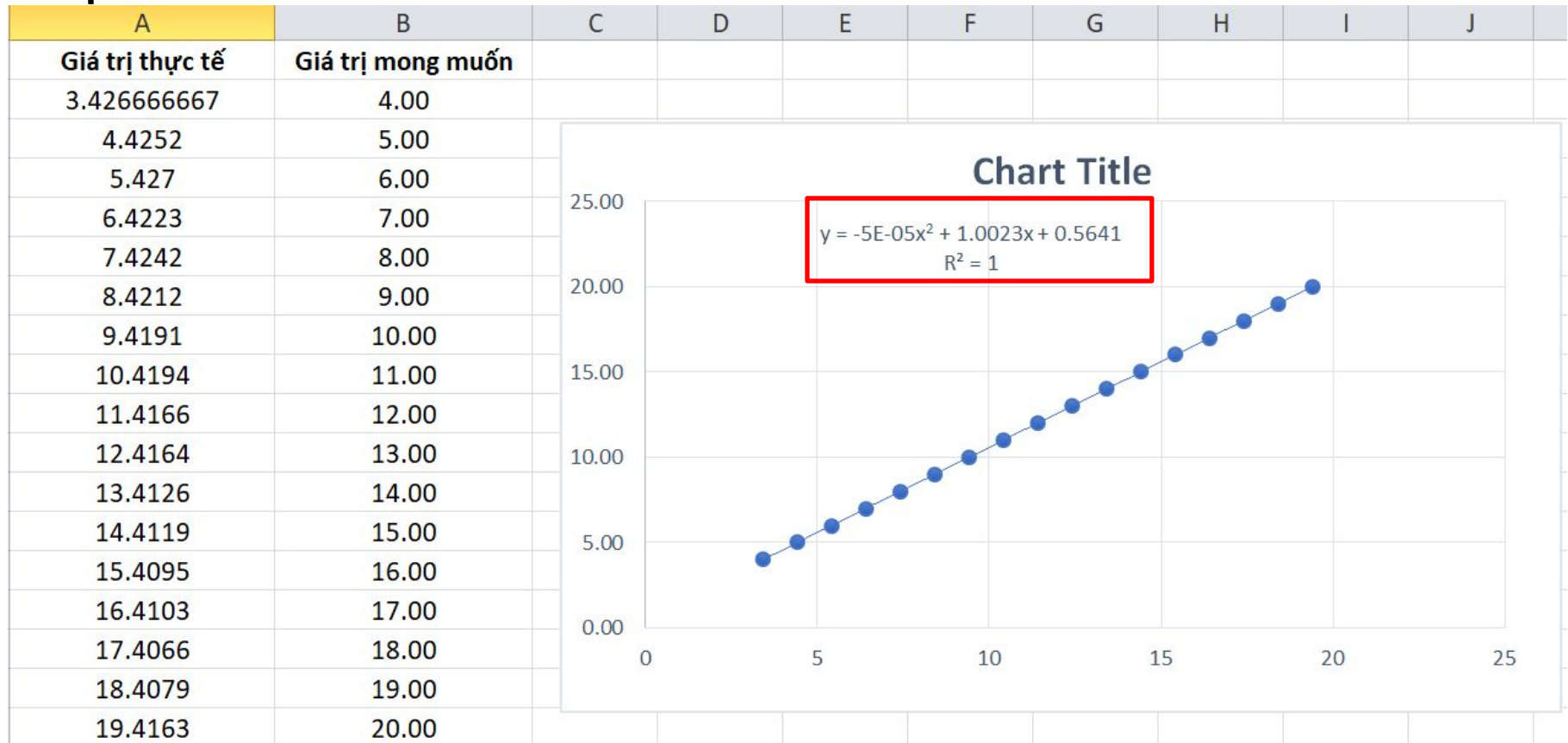
4-20mA CONVERTER

SIGNAL ADC IN = 4-20mA
SIGNAL ADC OUT = $(4-20) \times 120 = 0.48-2.4V$



PHƯƠNG PHÁP NỘI SUY

Sử dụng phương pháp nội suy để hiệu chỉnh giá trị Analog đo được





Instructor

Eng. Nguyen Huynh Nhat Thuong



Eng. Tran Thuy Ngoc Hang

Mảng 10 phần tử
Chuyển 10 dữ liệu
Chuyển 1 byte
Circular

Dùng Hercules:

Gửi từng phần tử. Đến phần tử $\frac{4}{5}$ thì sẽ có ngắt
Đến phần tử 9. thì sẽ có ngắt.

Ring buffer

Double buffer