

STM32 세미나

ADC

STM32 ADC Read Methods 3

1. Polling - 구현이 쉽다.
2. Interrupts - 구현이 복잡하다, 대신에 시스템 부하가 적다.
3. DMA - CPU 개입 없이 메모리로 ADC결과 도출

1.Polling

HAL_ADC_Start

Function name	HAL_StatusTypeDef HAL_ADC_Start (ADC_HandleTypeDef * hadc)
Function description	Enables ADC, starts conversion of regular group.
Parameters	<ul style="list-style-type: none">• hadc: ADC handle
Return values	<ul style="list-style-type: none">• HAL: status

HAL_ADC_GetValue

Function name	uint32_t HAL_ADC_GetValue (ADC_HandleTypeDef * hadc)
Function description	Get ADC regular group conversion result.
Parameters	<ul style="list-style-type: none">• hadc: ADC handle
Return values	<ul style="list-style-type: none">• ADC: group regular conversion data
Notes	<ul style="list-style-type: none">• Reading register DR automatically clears ADC flag EOC (ADC group regular end of unitary conversion).• This function does not clear ADC flag EOS (ADC group regular end of sequence conversion). Occurrence of flag EOS rising: If sequencer is composed of 1 rank, flag EOS is equivalent to flag EOC.If sequencer is composed of several ranks, during the scan sequence flag EOC only is raised, at the end of the scan sequence both flags EOC and EOS are raised. To clear this flag, either use function: in programming model IT: HAL_ADC_IRQHandler(), in programming model polling: HAL_ADC_PollForConversion() or __HAL_ADC_CLEAR_FLAG(&hadc, ADC_FLAG_EOS).

HAL_ADC_PollForConversion

Function name	HAL_StatusTypeDef HAL_ADC_PollForConversion (ADC_HandleTypeDef * hadc, uint32_t Timeout)
Function description	Wait for regular group conversion to be completed.
Parameters	<ul style="list-style-type: none">• hadc: ADC handle• Timeout: Timeout value in millisecond.
Return values	<ul style="list-style-type: none">• HAL: status
Notes	<ul style="list-style-type: none">• This function cannot be used in a particular setup: ADC configured in DMA mode. In this case, DMA resets the flag EOC and polling cannot be performed on each conversion.• On STM32F1 devices, limitation in case of sequencer enabled (several ranks selected): polling cannot be done on each conversion inside the sequence. In this case, polling is replaced by wait for maximum conversion time.

1. Polling-single

Pinout & Configuration

Clock Configuration

▼ Software Packs

Search

Categories A->Z

System Core >

Analog ▼

ADC1 ✓
ADC2

Timers ▼

RTC
TIM1
TIM2 ✓
TIM3
TIM4

Connectivity >

Computing >

Middleware and Software Packs >

ADC1 Mode and Configuration

Mode

☐ IN0
☒ IN1
☐ IN2

Configuration

Reset Configuration

NVIC SettingsDMA SettingsGPIO Settings

Parameter SettingsUser Constants

Configure the below parameters :

Search (Ctrl+F)

modeindependent mode

ADC_Settings

Data AlignmentRight alignment

Scan Conversion ModeDisabled

Continuous Conversion Mo...Disabled

Discontinuous Conversion ...Disabled

ADC_Regular_ConversionMode

Enable Regular ConversionsEnable

Number Of Conversion1

External Trigger Conversio...Regular Conversion launched by soft...

Rank1

ChannelChannel 1

Sampling Time239.5 Cycles

ADC_Injected_ConversionMode

1. Polling-Multi Channel

The screenshot shows the STM32CubeMX Pinout & Configuration tab. On the left, the 'Categories' list includes System Core, Analog, ADC1, ADC2, Timers, RTC, TIM1, TIM2 (selected), TIM3, TIM4, Connectivity, Computing, and Middleware and Software Packs. The main area is titled 'ADC1 Mode and Configuration'. Under the 'Mode' section, checkboxes for IN0, IN1, and IN2 are shown, with IN1 and IN2 checked and highlighted by red boxes. Under the 'Configuration' section, there is a 'Reset Configuration' button and tabs for NVIC Settings, DMA Settings, GPIO Settings, Parameter Settings (selected), and User Constants. Below these tabs, a search bar and a list of parameters are visible. The parameters are organized into sections: ADCs_Common_Settings (Mode: Independent mode), ADC_Settings (Data Alignment: Right alignment, Scan Conversion Mode: Disabled, Continuous Conversion Mo...: Disabled, Discontinuous Conversion ...: Disabled), and ADC_Regular_ConversionMode (Enable Regular Conversions: Enable, Number Of Conversion: 1, External Trigger Conversio...: Regular Conversion launched by soft..., Rank: 1, Channel: Channel 1).

The screenshot shows a configuration tree for the ADC. The 'ADC_Regular_ConversionMode' section is expanded, showing the following settings: 'Enable Regular Conversions' is set to 'Enable' (highlighted with a red box), 'Number Of Conversion' is set to '2' (highlighted with a red box), and 'External Trigger Conversio...' is set to 'Regular Conversion launched by soft...'. Below this, the 'Rank' is set to '1', 'Channel' is set to 'Channel 1', and 'Sampling Time' is set to '239.5 Cycles'. The 'Rank' is set to '2', 'Channel' is set to 'Channel 2' (highlighted with a red box), and 'Sampling Time' is set to '239.5 Cycles'. The 'ADC_Injected_ConversionMode' and 'WatchDog' sections are also visible but not expanded.

1.Polling-Multi Channel

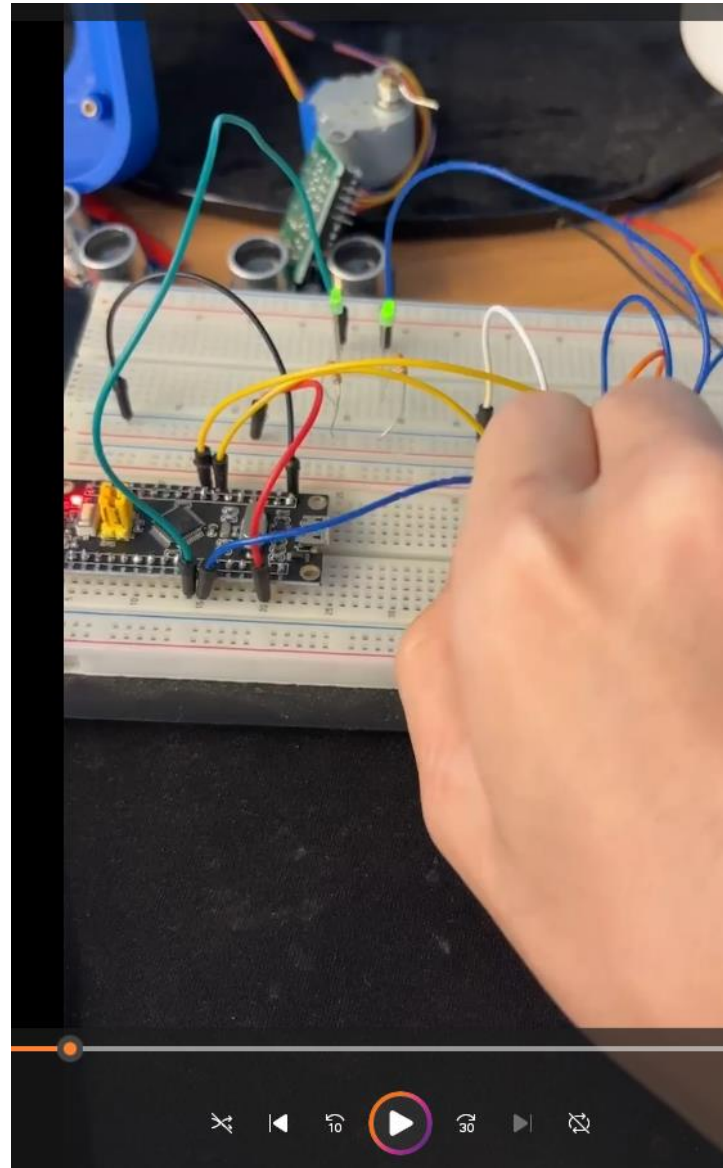
✓ NVIC Settings	✓ DMA Settings	✓ GPIO Settings
✓ Parameter Settings	✓ User Constants	
Configure the below parameters :		
<input type="text" value="Search (Ctrl+F)"/> ⏪ ⏩ ⓘ		
✓ ADCs_Common_Settings		
Mode	Independent mode	
✓ ADC_Settings		
Data Alignment	Right alignment	
Scan Conversion Mode	Enabled	
* Continuous Conversion Mode	Disabled	
Discontinuous Conversion ...	Enabled	
* Number Of Discontinuous ...	1	
✓ ADC_Regular_ConversionMode		
Enable Regular Conversions	Enable	
Number Of Conversion	2	
External Trigger Conversio...	Regular Conversion launched by soft...	
✓ Rank	1	

1.Polling-Multi Channel

```
uint16_t ADC_VAL[2];

void adcadc()
{
    HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_1);
    HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_2);
    while(1)
    {
        HAL_ADC_Start(&hadc1);           //ADC 시작
        HAL_ADC_PollForConversion(&hadc1,100); //1000ms 동안 데이터 받는다.
        ADC_VAL[0]=HAL_ADC_GetValue(&hadc1)/41;
        htim2.Instance->CCR1 = ADC_VAL[0];
        HAL_ADC_Start(&hadc1);           //ADC 시작
        HAL_ADC_PollForConversion(&hadc1,100); //1000ms 동안 데이터 받는다.
        ADC_VAL[1]=HAL_ADC_GetValue(&hadc1)/41;
        htim2.Instance->CCR2 = ADC_VAL[1];
    }
}
```

1. Polling-Multi Channel



1. Interrupt-Multi Channel

Pinout & Configuration

Clock Configuration

Software Packs

Search

Categories A->Z

System Core >

Analog >

ADC1

ADC2

ADC3

DAC

Timers >

Connectivity >

Multimedia >

Security >

Computing >

Middleware and Softwar... >

ADC1 Mode and Configuration

Mode

IN2

IN3

IN4

Configuration

Reset Configuration

NVIC Settings

DMA Settings

GPIO Settings

Parameter Settings

User Constants

NVIC Interrupt Table

Enabled

Preemption Priority

Sub Priority

ADC1, ADC2 and ADC3 global interrupts

1. Interrupt-Multi Channel

✓ NVIC Settings	✓ DMA Settings	✓ GPIO Settings
✓ Parameter Settings	✓ User Constants	
Configure the below parameters :		
<input type="text" value="Search (Ctrl+F)"/> ⏪ ⏩ ?		
✓ ADCs_Common_Settings		
Mode	Independent mode	
✓ ADC_Settings		
Clock Prescaler	PCLK2 divided by 4	
Resolution	12 bits (15 ADC Clock cycles)	
Data Alignment	Right alignment	
Scan Conversion Mode	Enabled	
Continuous Conversion Mode	Disabled	
Discontinuous Conversion ...	Enabled	
Number Of Discontinuous C...	2	
DMA Continuous Requests	Disabled	
End Of Conversion Selection	EOC flag at the end of single channel ...	
ADC_Regular_ConversionMode		
Number Of Conversion	2	
External Trigger Conversion...	Regular Conversion launched by softw...	
External Trigger Conversion...	None	

1.Interrupt-Multi Channel

```
void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef* hadc)
{
    adcVal[adcIndex] = HAL_ADC_GetValue(&hadc1);

    if(adcIndex == 1)
    {
        result1 = adcVal[0] / 41;
        result2 = adcVal[1] / 41;
    }

    adcIndex++;
    if(adcIndex > 1) adcIndex = 0;

    HAL_ADC_Start_IT(&hadc1);
}

int main(void)
{
    /* USER CODE BEGIN 2 */
    HAL_ADC_Start_IT(&hadc1);
    HAL_TIM_PWM_Start(&htim1, TIM_CHANNEL_2);
    HAL_TIM_PWM_Start(&htim1, TIM_CHANNEL_3);
    /* USER CODE END 2 */
}
```

3.DMA

Pinout & Configuration

Categories

A->Z

⚠ SYS

WWDG

Analog

Timers

LPTIM1

RTC

✓ TIM1

⚠ TIM2

TIM3

TIM4

⚠ TIM5

TIM6

TIM7

⚠ TIM8

TIM9

TIM10

Clock Configuration

▼ Software Packs

ADC1 Mode and Configuration

Mode

☐ IN2

☒ IN3

☒ IN4

Configuration

Reset Configuration

✓ NVIC Settings

✓ DMA Settings

✓ GPIO Settings

✓ Parameter Settings

✓ User Constants

Configure the below parameters :

Search (Ctrl+F)

▼ ADCs_Common_Settings

Mode Independent mode

▼ ADC_Settings

Clock Prescaler	PCLK2 divided by 4
Resolution	12 bits (15 ADC Clock cycles)
Data Alignment	Right alignment
Scan Conversion Mode	Enabled
Continuous Conversion Mode	Enabled
Discontinuous Conversion Mode	Disabled
DMA Continuous Requests	Enabled
End Of Conversion Selection	EOC flag at the end of single channel co...

▼ ADC_Regular_ConversionMode

Number Of Conversion	2
----------------------	---

3.DMA

Configuration

Reset Configuration

✓ NVIC Settings

✓ DMA Settings

✓ GPIO Settings

✓ Parameter Settings

✓ User Constants

Configure the below parameters :

Search (Ctrl+F)

◀▶

i

DMA Continuous Requests		Enabled
End Of Conversion Selection		EOC flag at the end of single channel co...
▼ ADC_Regular_ConversionMode		
Number Of Conversion		2
External Trigger Conversion S...		Regular Conversion launched by software
External Trigger Conversion E...		None
▼	Rank	1
	Channel	Channel 3
	Sampling Time	480 Cycles
▼	Rank	2
	Channel	Channel 4
	Sampling Time	480 Cycles

3.DMA

Reset Configuration

✔ Parameter Settings

✔ User Constants

✔ NVIC Settings

✔ DMA Settings

✔ GPIO Settings

DMA Request	Stream	Direction	Priority
ADC1	DMA2 Stream 0	Peripheral To Memory	Low

Add

Delete

DMA Request Settings

Peripheral		Memory	
Mode	<div>Circular</div>	Increment Address <input type="checkbox"/>	<input checked="" type="checkbox"/>
Use Fifo <input type="checkbox"/>	Threshold <div></div>	Data Width <div>Word</div>	<div>Word</div>
	Burst Size <div></div>	<div></div>	<div></div>

3.DMA

```
HAL_ADC_Start_DMA(&hadc1, (uint32_t*)Result, 2);
```