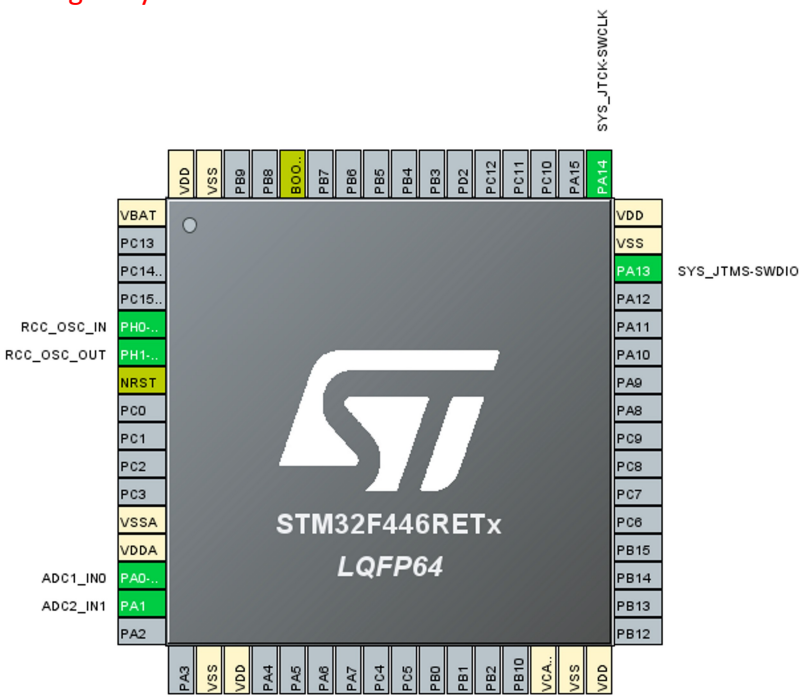
ADC Interrupt

25 Aralık 2021 Cumartesi 00:56

ADC Interrupt

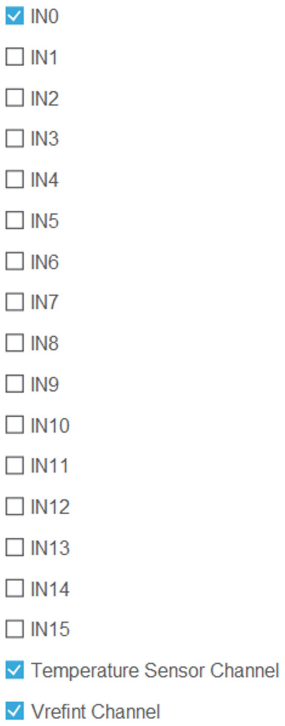
- HAL

Konfigürasyon Kısmı



Pin Name	Signal on Pin	GPIO output l...	GPIO mode	GPIO Pull-up/...	Maximum out...	User Label	Modified
PA0-WKUP	ADC1_IN0	n/a	Analog mode	No pull-up an...	n/a		☐
PA1	ADC2_IN1	n/a	Analog mode	No pull-up an...	n/a		☐

- ADC1 için IN0, Temperature Sensor Channel ve Vrefint Channel seçimi yaptık.



- Scan Mode ile Continuous Mode Enabled yapılır.

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	Disabled
End Of Conversion Selection	EOC flag at the end of single channel conversion

- ADC1 için 3 kanal olduğundan her biri için rank işlemi yapılır.

ADC_Regular_ConversionMode

Number Of Conversion	3
External Trigger Conversion Source	Regular Conversion launched by software
External Trigger Conversion Edge	None
Rank	1
Channel	Channel 0
Sampling Time	56 Cycles
Rank	2
Channel	Channel Vrefint
Sampling Time	56 Cycles
Rank	3
Channel	Channel Temperature Sensor
Sampling Time	56 Cycles

- ADC2 için IN1 seçimi yapılır.
- ADC2 için tek kanal olduğundan Scan Mode seçimi yapılmaz.

ADC_Settings

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

ADC_Regular_ConversionMode

Number Of Conversion	1
External Trigger Conversion Source	Regular Conversion launched by software
External Trigger Conversion Edge	None
Rank	1
Channel	Channel 1
Sampling Time	56 Cycles

- Interrupt kullanacağımızdan NVIC Settings kısmından Enabled yapılır.
  Sadece ADC1 değil ADC2 ve ADC3 için interrupts Enabled yapılmış olur.

NVIC Interrupt Table	Enabled	Preemption Priority	Sub Priority
ADC1, ADC2 and ADC3 interrupts	☑	0	0

Kod Kısmı

- ADC için Interrupt Mode kullanımı yapacağız.

```
 96      *** Interrupt mode IO operation ***
 97      ===================================
 98      [..]
 99        (+) Start the ADC peripheral using HAL_ADC_Start_IT()
100        (+) Use HAL_ADC_IRQHandler() called under ADC_IRQHandler() Interrupt subroutine
101        (+) At ADC end of conversion HAL_ADC_ConvCpltCallback() function is executed and user can
102            add his own code by customization of function pointer HAL_ADC_ConvCpltCallback
103        (+) In case of ADC Error, HAL_ADC_ErrorCallback() function is executed and user can
104            add his own code by customization of function pointer HAL_ADC_ErrorCallback
105        (+) Stop the ADC peripheral using HAL_ADC_Stop_IT()
```

```
1038 HAL_StatusTypeDef HAL_ADC_Start_IT(ADC_HandleTypeDef* hadc)
```

- Keseme olduğunda it.c dosyasındaki ADC_IRQHandler fonksiyonuna gelir ve fonksiyon içerisindeki HAL_ADC_IRQHandler fonskiyonunu çalıştırır.
  Bizim main.c dosyasında hazır olarak yazılmıştır.

```
206 void ADC_IRQHandler(void)
207 {
208   /* USER CODE BEGIN ADC_IRQn 0 */
209
210   /* USER CODE END ADC_IRQn 0 */
211   HAL_ADC_IRQHandler(&hadc1);
212   HAL_ADC_IRQHandler(&hadc2);
213   /* USER CODE BEGIN ADC_IRQn 1 */
214
215   /* USER CODE END ADC_IRQn 1 */
216 }
```

- 1578.satırdaki fonksiyonu main.c dosyasında kullanarak Interrupt'a girdiğinde çalışacak.

```
1578 __weak void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef* hadc)
```

- İçerisine önce hangi ADC'yi çalıştırdığını öğrenmemiz gerekiyor. Bunun için _HAL_ADC_GET_FLAG kullanmamız gerekiyor.

```
129       (+) __HAL_ADC_GET_FLAG: Get the selected ADC's flag status
```

```
548 #define __HAL_ADC_GET_FLAG(__HANDLE__, __FLAG__) ((((__HANDLE__)->Instance->SR) & (__FLAG__)) == (__FLAG__))
```

- Temperature için Datasheet kısmından aşağıdaki tabloları kullanıyoruz.

Temperature sensor characteristics

Symbol	Parameter	Min	Typ	Max	Unit
T_L(1)	V_{SENSE} linearity with temperature	-	±1	±2	°C
Avg_Slope(1)	Average slope	-	2.5	-	mV/°C
V_{25}(1)	Voltage at 25 °C	-	0.76	-	V
t_{START}(2)	Startup time	-	6	10	µs
T_{S_temp}(2)	ADC sampling time when reading the temperature (1 °C accuracy)	10	-	-	µs

Temperature sensor calibration values

Symbol	Parameter	Memory address
TS_CAL1	TS ADC raw data acquired at temperature of 30 °C, V_{DDA}= 3.3 V	0x1FFF 7A2C - 0x1FFF 7A2D
TS_CAL2	TS ADC raw data acquired at temperature of 110 °C, V_{DDA}= 3.3 V	0x1FFF 7A2E - 0x1FFF 7A2F

- Aşağıdaki formül üzerinden hesaplıyoruz.

  Temperature (in °C) = {(V_{SENSE} – V_{25}) / Avg_Slope} + 25

  – V_{25} = V_{SENSE} value for 25° C
  – Avg_Slope = average slope of the temperature vs. V_{SENSE} curve (given in mV/°C or µV/°C)

```
23 /* Private includes ----------------------------------
24 /* USER CODE BEGIN Includes */
25 #define VREFIN_CAL ((uint16_t*)((uint32_t)0x1FFF7A2A))
26 #define V25 (float) 0.76
27 #define Avg_Slope (float) 0.0025
28
29 uint16_t adc1_value[3], adc2_value;
30 float Vadc1, Vadc2, Vsense, Vdda, temperature;
31 int count=0;
32 /* USER CODE END Includes */
```

```
121   /* USER CODE BEGIN 2 */
122   HAL_ADC_Start_IT(&hadc1);
123   HAL_ADC_Start_IT(&hadc2);
124   /* USER CODE END 2 */
```

```c
/* Private user code ---------------------------------------------------------*/
/* USER CODE BEGIN 0 */
void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef* hadc)
{
    if(__HAL_ADC_GET_FLAG(&hadc1, ADC_FLAG_EOC) != RESET)
    {
        adc1_value[count]= HAL_ADC_GetValue(&hadc1);
        count++;      // count=0 => IN0, count=1 => Vrefint, count=2 => Temperature
        if(count==3)
        {
            count=0;
        }
        Vdda= (float) 3.3 * (*VREFIN_CAL) / adc1_value[1];
        Vadc1= Vdda * adc1_value[0] / 4095;

        Vsense = Vdda * adc1_value[2] / 4095;
        temperature= ((Vsense - V25) / Avg_Slope) + 25;
    }
    if(__HAL_ADC_GET_FLAG(&hadc2,ADC_FLAG_EOC) != RESET)
    {
        adc2_value=HAL_ADC_GetValue(&hadc2);
        Vadc2= Vdda * adc2_value / 4095;
    }
}
/* USER CODE END 0 */
```

Variable Name	Address/Expression	Read Value
Vadc1	0x20000028	2.9433491
Vadc2	0x2000002c	1.9205443
Vdda	0x20000030	2.9363744
Vsense	0x20000034	0.782018
count	0x20000038	0
temperature	0x2000003c	33.065987