Lab 10: Analog to Digital Converter (ADC)
Instructor: Prof. Yifeng Zhu
Spring 2016

Goals

1. Understand basic ADC concepts, such as successive-approximation, sampling error, resolution, and data alignment
2. Map a GPIO pin to an ADC input
3. Use software trigger to make ADC conversions, and deploy polling method to read ADC results
4. Understand the tradeoff between conversion accuracy and conversion speed by configuring ADC sample time
5. Understand the concept of ADC resolution
6. Understand the difference between regular conversions and injected conversions for a sequence of ADC channels

Pre-lab Assignment

1. Read Chapter 20 Analog to Digital Conversion (ADC)
2. Complete the pre-lab assignment

Lab Demo

1. ***Part 1***. When the voltage input is higher than 2.0V, the LED is light up. When the voltage is lower than 1.0V, the LED is off. The input voltage can be controlled manually by using a potentiometer. Use a voltage meter to verify it.
2. ***Part 2***. Implement infrared (IR) proximity sensor. When an object gets close to the IR sensor, light up the LED.
3. ***Something cool***. The following gives a few examples.
    a. Use a timer to periodically trigger the ADC
    b. Show the voltage measurement on the LCD (in terms of volts)
    c. Using the potentiometer to control the rotation speed of a stepper motor
    d. Count how many times your hand waves over the IR sensor
    e. Measure the distance of an object from the IR sensor
    f. Use the analog watchdog in the processor to trigger ADC when an object gets too close to the sensor

Post-lab Assignment

1. Write your answer in Readme.md and submit it to the gitlab server.

Reference Voltage

The reference voltage to the analog-to-digital (ADC) and digital- to-analog (DAC) converters can be provided externally or internally.

- If an ***external*** voltage is selected as the reference voltage, the external reference voltage should be applied to the VREF+ pin.

- If an ***internal*** voltage is selected as the reference voltage, the VREF+ pin can provide the reference voltage for external components.
    - The internal voltage reference is enabled by setting the ENVR bit in the VREFBUF_CSR register.
    - The internal voltage reference can be selected by the VRS bits in the VREFBUF_CSR register.
        - If VRS is set, the voltage reference is ~2.5V.
        - If VRS is cleared, the voltage reference is ~2.048V.

ADC: Migrating from STM32L1 to STM32L4

	STM32L1	**STM32L4**
Instance	ADC1	ADC1, ADC2, ADC3
Max Resolution	12 bits	12 bits, or 16 bits via digital oversampling
Mode	single/continuous / scan / discontinuous	single/continuous/scan/ discontinuous dual mode
Reference Voltage	External	External or internal

STM32L4 has a new ADC architecture. Register names and controls are different from STM32L1. Therefore, the flowchart given Figure 20-11 of Textbook (page 457) should be modified, as summarized below.

1. STM32L4 adds a new register named an *Analog Switch Control Register* (GPIO_ASCR) for each GPIO port. When a bit in GIPO_ASCR is set, the corresponding GPIO pin is connected to the ADC input. After setting the mode of PA.1 as analog, the following instruction is required to connect PA.1 to ADC.

```
GPIOA->ASCR |= GPIO_ASCR_EN_1;
```

   where `GPIO_ASCR_EN_1` is defined in *`stm32l476xx.h`* as follows

```
#define GPIO_ASCR_EN_1 ((uint32_t)0x00000002)
```

2. **Steps to set up PA1 (ADC12_IN6) for ADC1**:
    1. Enable ADC clock bit RCC_AHB2ENR_ADCEN in the RCC->AHB2ENR register.
    2. Disable ADC1 by clearing the ADC_CR_ADEN in the ADC1->CR register.
    3. Enable I/O analog switches voltage booster (SYSCFG_CFGR1_BOOSTEN) in the ADC123_COMMON->CCR register.
    4. Set ADC_CCR_VREFEN bit in the ADC123_COMMON->CCR register to enable the conversion of internal channels. This is required to make conversion of internal channels.
    5. Configure the ADC prescaler to select the frequency of the clock to the ADC (set clock not divided) in ADC123_COMMON->CCR.
    6. Configure ADC_CCR_CKMODE bits in ADC123_COMMON->CCR to set the ADC clock mode as synchronous clock mode (HCLK/1).
    7. Configure all ADCs as independent (clear ADC_CCR_DUAL bits) in ADC123_COMMON->CCR

8. By default, the ADC is in deep-power-down mode where its supply is internally switched off to reduce the leakage currents. Therefore, software needs to wait up ADC. The ADC_Wakeup() function is provided in the project template.
9. Configure RES bits in ADC1->CFGR to set the resolution as 12 bits.
10. Select right alignment in the ADC1->CFGR register.
11. Clear ADC_SQR1_L bits in ADC1->SQR1 to select 1 conversion in the regular channel conversion sequence.
12. Specify the channel number 6 as the 1st conversion in regular sequence (ADC1->SQR1)
13. Configure the channel 6 as single-ended (ADC1->DIFSEL).
14. Select ADC sample time in ADC1->SMPR1. The sampling time must be long enough for the input voltage source to charge the embedded capacitor to the input voltage level.
15. Select ADC as discontinuous mode by clearing the ADC_CFGR_CONT bits in ADC1->CFGR.
16. Clear ADC_CFGR_EXTEN bits in ADC1->CFGR to select software trigger
17. Enable ADC1 by setting the ADC_CR_ADEN bit in the ADC1->CR register
18. Wait until ADC1 is ready (i.e., wait until ADC_ISR_ADRDY bit in ADC1->ISR is set by hardware)

3. **Using the software to trigger one ADC conversion**:
    1. Software can start one ADC conversion by setting the ADC_CR_ADSTART bit in the ADC1->CR register
    2. Software has to wait the completion of ADC conversion by checking whether ADC_CSR_EOC_MST in the ADC123_COMMON->CSR register has been set by the hardware.
    3. The conversion result is saved in register ADC1->DR.

4. **ADC Wakeup.** By default, the ADC is in deep-power-down mode where its supply is internally switched off to reduce the leakage currents.

```
void ADC_Wakeup (void) {
  int wait_time;
  // To start ADC operations, the following sequence should be applied
  // DEEPPWD = 0: ADC not in deep-power down
  // DEEPPWD = 1: ADC in deep-power-down (default reset state)
  if ((ADC1->CR & ADC_CR_DEEPPWD) == ADC_CR_DEEPPWD)
     ADC1->CR &= ~ADC_CR_DEEPPWD; // Exit deep power down mode if still in that state

  // Enable the ADC internal voltage regulator
  // Before performing any operation such as launching a calibration or enabling the ADC,
  // the ADC voltage regulator must first be enabled and the software must wait for the
  // regulator start-up time.
  ADC1->CR |= ADC_CR_ADVREGEN;

  // Wait for ADC voltage regulator start-up time
  // The software must wait for the startup time of the ADC voltage regulator
  // (T_ADCVREG_STUP, i.e. 20 us) before launching a calibration or enabling the ADC.
  wait_time = 20 * (80000000 / 1000000);
  while(wait_time != 0) {
    wait_time--;
  }
}
```

Part 1. Measuring the Input Voltage Adjusted by a Potentiometer

A potentiometer (pot) is a three-terminal variable resistor. It uses a sliding contact and works as an adjustable voltage divider. When two outer terminals connected to Vcc and the ground respectively, the center terminal generates a voltage that varies from 0 to Vcc, depending on the position of the sliding contact.

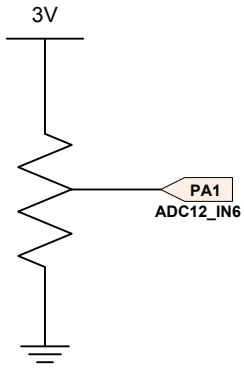

Figure 1. Measure the voltage from a potentiometer-based divider (Vcc = 3V)

The following refers to question 1 of the post-lab assignment. The objective is to identify the reference voltage.

- Use a multi-meter to measure the voltage on PA1:

  Voltage of PA1 = _____

- In the debug environment, find out the value of the data register (DR) of the ADC:

  ADC DR register = _____

- Calculate the reference voltage:

  Reference voltage = _____

Part 2. Infrared (IR) Proximity Sensor

In this part, you will implement a very simple proximity sensor by using an infrared transmitter and an infrared receiver. When an obstacle moves closer, the infrared captures more infrared that is bounced back by the obstacle.

It is recommended to put a dark tape around the receiver or place a partition between the transmitter and receiver to reduce infrared leakage.

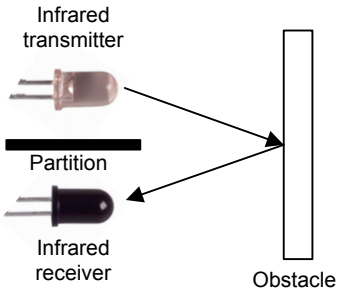

Figure 2. Infrared Proximity Sensor

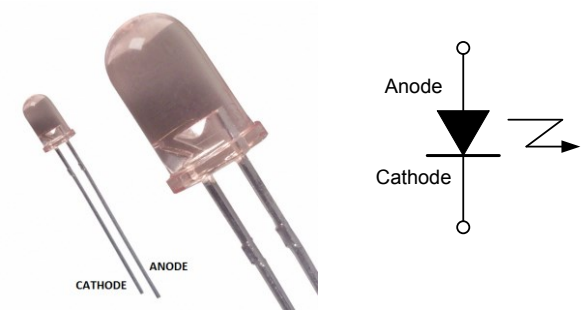

Figure 3. QED123 Infrared Light Emitting Diode (Infrared Transmitter)

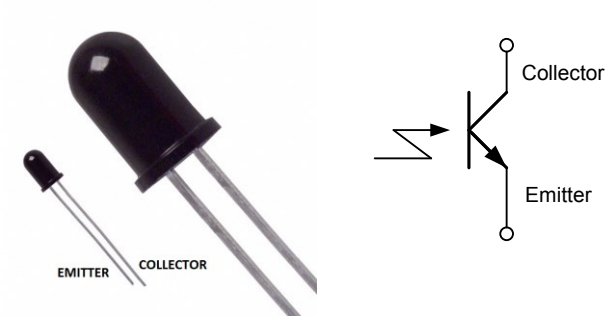

Figure 4. QSD124 Infrared Phototransistor (Infrared Receiver)

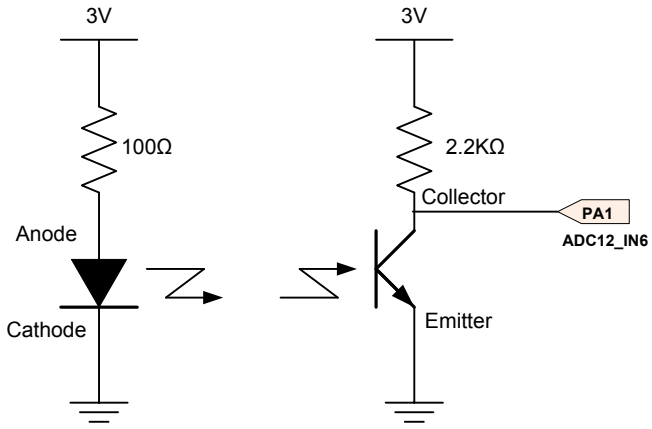

Figure 5. Basic Connection Diagram

ECE 271 pre-Lab Assignment
Lab 10: Analog to Digital Converter (ADC)

Student Name: _____

1. Setting of ADC Control Registers

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**ADCx_ISR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JQOVF	AWD3	AWD2	AWD1	JEOS	JEOC	OVR	EOS	EOC	EOSMP	ADRDY
	Value																																
0x04	**ADCx_IER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JQOVFIE	AWD3IE	AWD2IE	AWD1IE	JEOSIE	JEOCIE	OVRIE	EOSIE	EOCIE	EOSMPIE	ADRDYIE
	Value																																
0x08	**ADCx_CR**	ADCAL	ADCALDIF	DEEPPWD	ADVREGEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JADSTP	ADSTP	JADSTART	ADSTART	ADDIS	ADEN
	Value																																
0x0C	**ADCx_CFGR**	JQDIS	AWD1CH[4:0]					JAUTO	JAWD1EN	AWD1EN	AWD1SGL	JQM	JDISCEN	DISCNUM [2:0]			DISCEN	Res.	AUTDLY	CONT	OVRMOD	EXTEN[1:0]		EXTSEL [3:0]				ALIGN	RES [1:0]		Res.	DMACFG	DMAEN
	Value																																
0x0C	**ADCx_CFGR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ROVSM	TROVS	OVSS[3:0]				OVSR [2:0]			JOVSE	ROVSE
	Value																																
0x14	**ADCx_SMPR1**	Res.	Res.	SMP9 [2:0]			SMP8 [2:0]			SMP7 [2:0]			SMP6 [2:0]			SMP5 [2:0]			SMP4 [2:0]			SMP3 [2:0]			SMP2 [2:0]			SMP1 [2:0]			SMP0 [2:0]		
	Value																																
0x18	**ADCx_SMPR2**	Res.	Res.	Res.	Res.	Res.	SMP18 [2:0]			SMP17 [2:0]			SMP16 [2:0]			SMP15 [2:0]			SMP14 [2:0]			SMP13 [2:0]			SMP12 [2:0]			SMP11 [2:0]			SMP10 [2:0]		
	Value																																
0x1C	Reserved	Res.																															
0x20	**ADCx_TR1**	Res.	Res.	Res.	Res.	HT1[11:0]												Res.	Res.	Res.	Res.	LT1[11:0]												
	Value																																
0x24	**ADCx_TR2**	Res.	Res.	Res.	Res.	HT2[[7:0]												Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LT2[7:0]							
	Value																																
0x28	**ADCx_TR3**	Res.	Res.	Res.	Res.	HT3[[7:0]												Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LT3[7:0]							
	Value																															0	0
0x2C	Reserved	Res.																															
0x30	**ADCx_SQR1**	Res.	Res.	Res.	SQ4[4:0]					Res.	SQ3[4:0]					Res.	SQ2[4:0]					Res.	SQ1[4:0]					Res.	Res.	L[3:0]			
	Value																																
0x34	**ADCx_SQR2**	Res.	Res.	Res.	SQ9[4:0]					Res.	SQ8[4:0]					Res.	SQ7[4:0]					Res.	SQ6[4:0]					Res.	SQ5[4:0]				
	Value																																
0x38	**ADCx_SQR3**	Res.	Res.	Res.	SQ14[4:0]					Res.	SQ13[4:0]					Res.	SQ12[4:0]					Res.	SQ11[4:0]					Res.	SQ10[4:0]				
	Value																																
0x3C	**ADCx_SQR4**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SQ16[4:0]					Res.	SQ15[4:0]				
	Value																																
0x40	**ADCx_DR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	regular RDATA[15:0]															
	Value																																
0x44-0x48	Reserved	Res.																															
0x4C	**ADCx_JSQR**	Res.	JSQ4[4:0]					Res.	JSQ3[4:0]					Res.	JSQ2[4:0]					Res.	JSQ1[4:0]					JEXTEN[1:0]	JEXTSEL [3:0]				JL[1:0]		
	Value																																

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x50-0x5C	Reserved	Res.																															
0x60	**ADCx_OFR1**	OFFSET1_EN	OFFSET1_CH[4:0]					Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OFFSET1[11:0]											
	Value																																
0x64	**ADCx_OFR2**	OFFSET2_EN	OFFSET2_CH[4:0]					Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OFFSET2[11:0]											
	Value																																
0x68	**ADCx_OFR3**	OFFSET3_EN	OFFSET3_CH[4:0]					Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OFFSET3[11:0]											
	Value																																
0x6C	**ADCx_OFR4**	OFFSET4_EN	OFFSET4_CH[4:0]					Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OFFSET4[11:0]											
	Value																																
0x70-0x7C	Reserved	Res.																															
0x80	**ADCx_JDR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JDATA1[15:0]															
	Value																																
0x84	**ADCx_JDR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JDATA2[15:0]															
	Value																																
0x88	**ADCx_JDR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JDATA3[15:0]															
	Value																																
0x8C	**ADCx_JDR4**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JDATA4[15:0]															
	Value																																
0x8C-0x9C	Reserved	Res.																															
0xA0	**ADCx_AWD2CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWD2CH[18:0]																		
	Value																																
0xA4	**ADCx_AWD3CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWD3CH[18:0]																		
	Value																																
0xA8-0xAC	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0xB0	**ADCx_DIFSEL**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIFSEL[18:0]																		
	Value																																
0xB4	**ADCx_CALFACT**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CALFACT_D[6:0]							Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CALFACT_S[6:0]						
	Value																																

2. Master and slave ADC common registers (ADC123_COMMON)

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**ADCx_CSR**	Res.	Res.	Res.	Res.	Res.	JQOVF_SLV	AWD3_SLV	AWD2_SLV	AWD1_SLV	JEOS_SLV	JEOC_SLV	OVR_SLV	EOS_SLV	EOC_SLV	EOSMP_SLV	ADRDY_SLV	Res.	Res.	Res.	Res.	Res.	JQOVF_MST	AWD3_MST	AWD2_MST	AWD1_MST	JEOS_MST	JEOC_MST	OVR_MST	EOS_MST	EOC_MST	EOSMP_MST	ADRDY_MST
		slave ADC2																					master ADC1										
	Value																																
0x04	**Reserved**	Res.																															
0x08	**ADCx_CCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CH18SEL	CH17SEL	VREFEN	PRESC[3:0]				CKMODE[1:0]		MDMA[1:0]		DMACFG	Res.	DELAY[3:0]				Res.	Res.	Res.	DUAL[4:0]				
	Value																																
0x0C	**ADCx_CDR**	RDATA_SLV[15:0]																RDATA_MST[15:0]															
	Value																																

ECE 271 Lab Demo
Lab 10: Analog to Digital Converter (ADC)

Demo Part 1, Part 2, and something cool to TA.

ECE 271 post-Lab Assignment
Lab 10: Analog to Digital Converter (ADC)

Write your answer to the following questions in Readme.md and submit it to the gitlab server.

1. For 12-bit ADC, we know that

$$ADC\ Result = \ floor\left(2^{12} \times \frac{V}{V_{REF}} + \frac{1}{2}\right)$$

   Design an experiment to find out V_{REF}

2. When the voltage output from the potentiometer-based voltage divider is lower than 1.0V, the LED is turned off. What constant value should the ADC DR register be compared with?

3. When the voltage output from the potentiometer-based voltage divider is higher than 2.0V, the LED is light up. What constant value should the ADC DR register be compared with?

4. What is the maximum distance at which your sensor can reliably detect your hand?