Name: ..... Student ID: .....
Name: ..... Student ID: .....

Laboratory 09
Introduction to Data Conversion

1. Introduction

Physical quantities around us such as sound, temperature, brightness, color, etc. can be an important input of embedded systems. To process these quantities by any electronic processors, they first have to be converted into electrical signals by a device known as a ***sensor***. Most of sensors however give electrical output signals in analog forms. Digital processors such as a microcontroller used in embedded system, however does process data and gives the result in digital domain. Therefore it is a need for converting the analog signal into digital data before being processed by the processor. The circuit designed to perform such a task is generally called an ***Analog-to-Digital Converter (ADC)***. On the opposite way, human being can not understand the processing results from the digital processor. The results have to be converted back into an analog form, then devices such as ***transducers*** and ***actuators*** transform them into physical quantities that we can perceive. Consequently, there is also a need for converting the digital data into the analog signal and the circuit that performs this task is called **Digital-to-Analog Converter (DAC)**.

Today both ADC and DAC are available in the market as a chip. It can be interfaced with microcontroller through GPIO ports. Most microcontrollers however, have both ADC and DAC as its standard built-in peripheral, including the LPC1769. In this lab, we will learn some background knowledge of data conversion and explore its ability using built-in ADC/DAC of LPC1769.

2. Background knowledge

2.1 Digital-to-Analog Converter (DAC)

DAC is a circuit that converts a binary input number into an analog output voltage which is proportional to that input number. DAC is widely used to create continuously varying signals, for example to generate analog waveforms

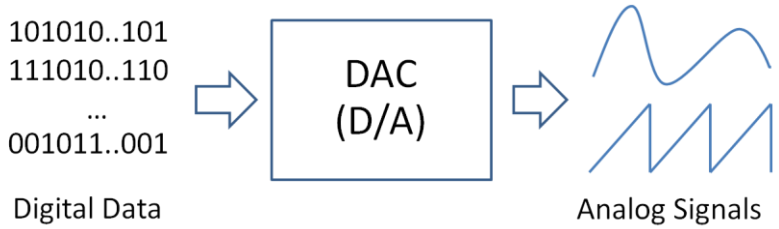


Fig. 1. Conceptual diagram of DAC

2.1.1 Model and Parameter of DAC

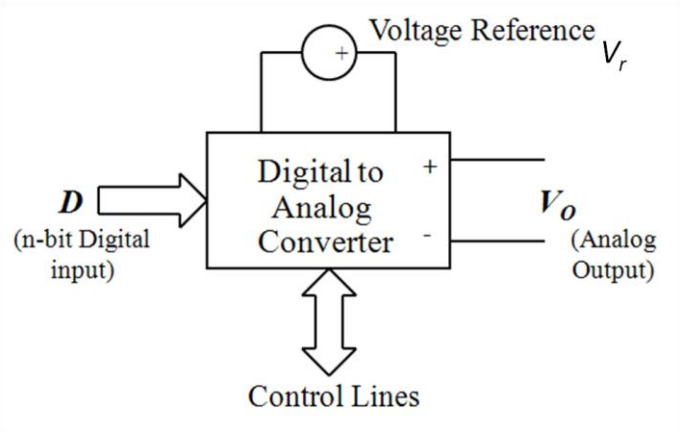


Fig. 2. Model of DAC

The relationship between input and output of DAC can be expressed as

$$V_o = D\frac{V_r}{2^n} = D \cdot (resolution)$$

where

V_o = Analog output that DAC can create
V_r = Reference voltage for using in conversion
D = Digital input word
n = Number of bit of DAC

For each input digital value, there is a corresponding analog output. The number of possible output values is given by **2^n**. The step size is **Vr /2^n** and called the ***resolution***. The difference between its maximum and minimum output values is called ***range***. The time duration DAC needed to complete a conversion is called **conversion time** and its invert called **update rate**

2.1.2 DAC input-output characteristic

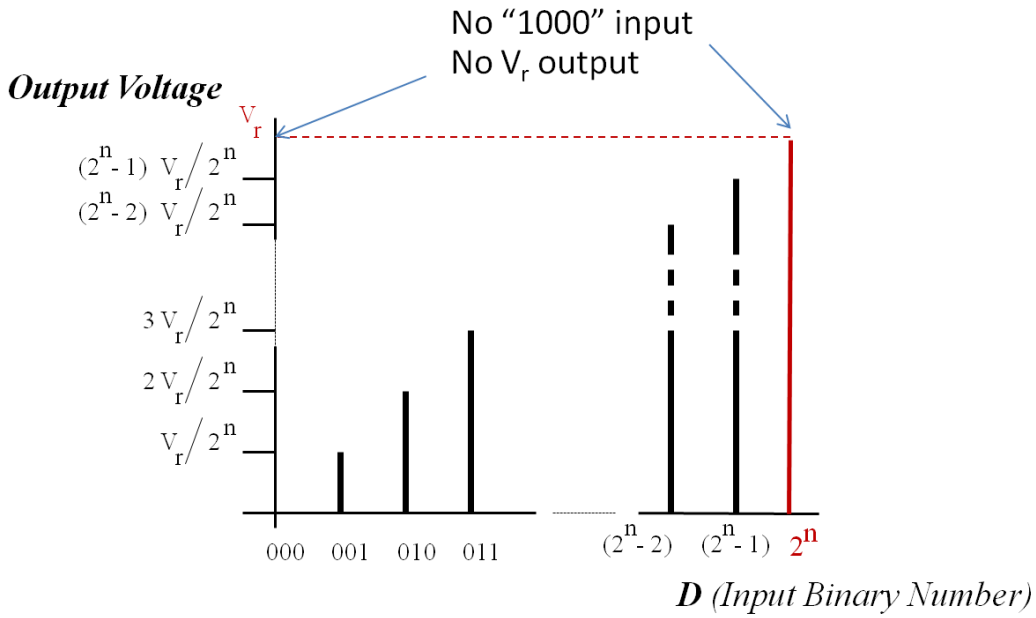


Fig. 3. Transfer characteristic of DAC: 3-bit DAC

2.2 Analog-to-Digital Converter (ADC)

An ADC is an electronic circuit whose digital output is proportional to its analog input. It measures the input voltage and gives a binary output number proportional to its size. There are many types of analog inputs depending on its sources and its applications such as medical, industrial, entertainment, etc., yielding differences in their frequency and amplitude requirement. As a result, many types of ADC have been developed, with characteristics optimized for these differing applications.

2.2.1 Model and Parameter of ADC

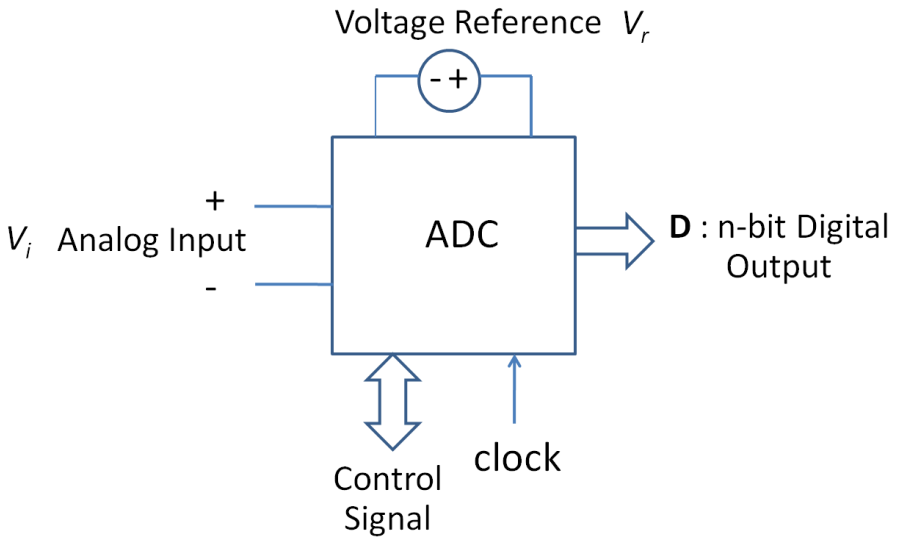


Fig. 3. Model of ADC

The relationship between input and output of ADC can be expressed as

$$D = \frac{V_i}{V_r} 2^n$$

where

D = Digital output word: integer 0 to (2^n-1)

n = Number of bit of conversion output

$\mathbf{V_i}$ = Analog input of ADC

$\mathbf{V_r}$ = Reference voltage for using in conversion

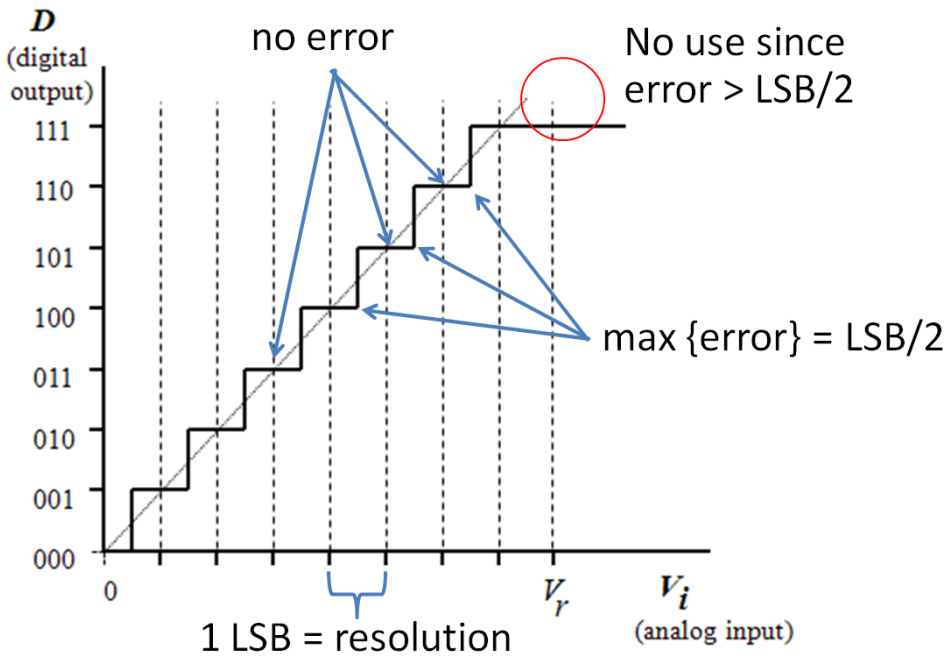


Fig. 4. Transfer characteristic of 3-bit ADC

Range: The difference between maximum and minimum permissible input values. The input range of the ADC is directly linked to the value of the voltage reference; in many ADC circuits the range is actually equal to the reference voltage. Analog inputs that exceed the maximum or minimum permissible input values are likely to be digitized as maximum and minimum values respectively, i.e. a limiting (or 'clipping') action takes place.

Resolution: The resolution is $V_r / 2^n$ which is the size of a single step on the staircase characteristic.

Quantization error: The error due to mapping an interval of analog value to an integer digital word. It is equal to value of digital output – value analog input

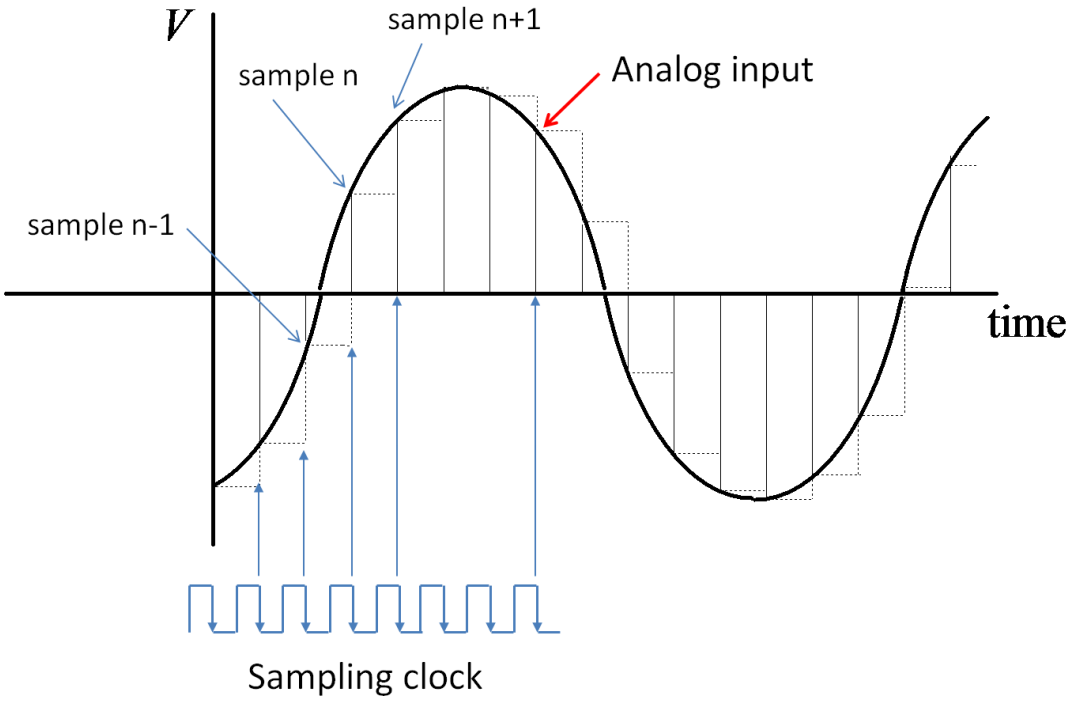
Conversion time: The time ADC needed to complete the conversion

According to the DAC characteristic in Fig.3, it can be concluded as follows

- The output value is precisely correct for the input voltage at the middle of the step.
- The greatest quantization error occurs at either end of the step and is equal to one half of the step width, or half of one least significant bit (LSB) equivalent of the voltage scale.
- The more steps there are representing the range, the narrower they will be, and hence the quantization error is reduced.
- More steps are obtained by increasing the number of bits in the ADC process but this will increase the complexity and cost of the ADC and the time it takes to complete a conversion

2.2.3 Sampling frequency of ADC

In the operation of ADC, its sample and hold (S/H) circuit uses the sampling clock to sample analog input signal and hold this voltage sample. The sample is then quantized, measured and translated into digital output data. In order to retain a complete analog information, the sampling frequency has to be at least two times greater than frequency of the analog signal. This is known as Nyquist criterion. If the Nyquist criterion is not satisfied, then a phenomenon called **aliasing** occurs i.e. a new lower frequency is generated.



3. DAC and ADC of LPC1769

3.1 DAC of LPC1769

Features:

10-bit digital to analog converter

Resistor string architecture

Buffered output

Selectable speed vs. power

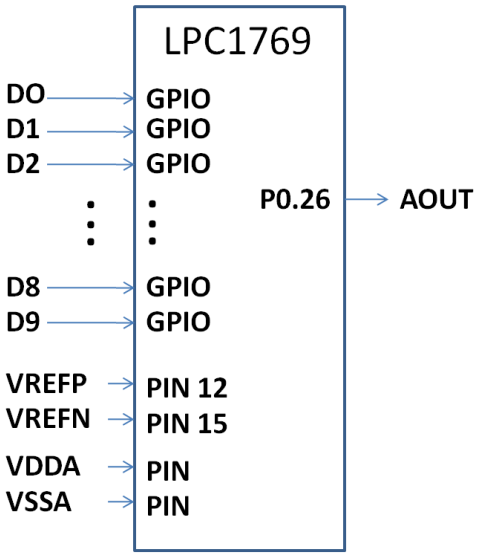
Maximum update rate of 1 MHz.

Basic configurarion

The DAC is configured using the following registers:

1. Power: The DAC is always connected to VDDA. Register access is determined by PINSEL and PINMODE settings.
2. Clock: In the PCLKSEL0 register [23:22] select PCLK_DAC
(00=> CCLK/4, 01=> CLCK/1, 10=> CLCK/2, 11=> CLCK/8)
3. Pins: Enable the DAC pin through the PINSEL registers. Select pin mode for port pin with DAC through the PINMODE registers. This must be done before accessing any DAC registers.
4. DMA: The DAC can be connected to the GPDMA controller.

- DAC input set up can be 2 ways
  1) No dedicated input pins. Set any GPIO pins to be the input pins
  2) Software writes data to DAC data register (DACR)
- DAC output pin setup
  - Use P0.26 be an analog out or "AOUT" of DAC
  - It shares with ADC (AD0.3) and UART (RXD3)
  - Set PINSELC1 register [21:20]=[10] to select AOUT
  - Set PINMODE1 register [21:20]=[10] to select no pull-up, no pull-down resister



*Xpresso board **internally connects** VREFP to VDDA and VREFN to VSSA*

Register Description

Name	Description	Access	Reset value[1]	Address
DACR	D/A Converter Register. This register contains the digital value to be converted to analog and a power control bit.	R/W	0	0x4008 C000
DACCTRL	DAC Control register. This register controls DMA and timer operation.	R/W	0	0x4008 C004
DACCNTVAL	DAC Counter Value register. This register contains the reload value for the DAC DMA/Interrupt timer.	R/W	0	0x4008 C008

DACR

Bit	Symbol	Value	Description	Reset Value
5:0	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
15:6	VALUE		After the selected settling time after this field is written with a new VALUE, the voltage on the AOUT pin (with respect to V_{SSA}) is VALUE × ((V_{REFP} - V_{REFN})/1024) + V_{REFN}.	0
16	BIAS[1]	0	The settling time of the DAC is 1 μs max, and the maximum current is 700 μA. This allows a maximum update rate of 1 MHz.	0
		1	The settling time of the DAC is 2.5 μs and the maximum current is 350 μA. This allows a maximum update rate of 400 kHz.	
31:17	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

DACCTRL

Bit	Symbol	Value	Description	Reset Value
0	INT_DMA_REQ	0	This bit is cleared on any write to the DACR register.	0
		1	This bit is set by hardware when the timer times out.	
1	DBLBUF_ENA	0	DACR double-buffering is disabled.	0
		1	When this bit and the CNT_ENA bit are both set, the double-buffering feature in the DACR register will be enabled. Writes to the DACR register are written to a pre-buffer and then transferred to the DACR on the next time-out of the counter.	
2	CNT_ENA	0	Time-out counter operation is disabled.	0
		1	Time-out counter operation is enabled.	
3	DMA_ENA	0	DMA access is disabled.	0
		1	DMA Burst Request Input 7 is enabled for the DAC (see Table 544).	
31:4	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

DACCNTVAL

Bit	Symbol	Description	Reset Value
15:0	VALUE	16-bit reload value for the DAC interrupt/DMA timer.	0

Operation

Default mode using without DMA:

1. Select update rate via BIAS bit (DACR[16])
2. Write 10-bit data to DACR [15:6]
3. Analog output will be available on AOUT pin after a conversion time

4. To use timer and interrupt flag,
    a. write 16-bit data to DACCNTVAL
    b. enable counter (CNT_ENT = 1) when times out interrupt flag will set (INT_DMA_REQ=1)

DMA mode:

Similar to default mode but need to enable counter (CNT_ENA=1) and enable DMA (DMA_ENT=1)

DMA mode with double buffer:

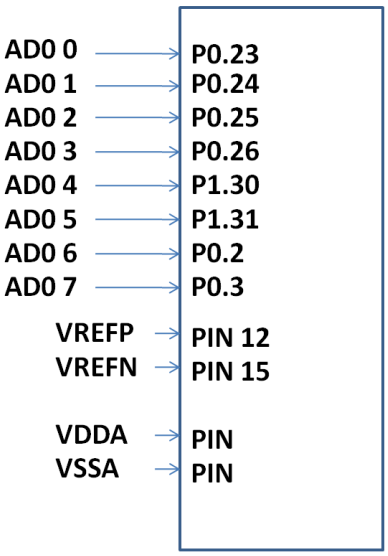
In this mode, any data write to DACR will not go directly to the DACR. It only load the pre-buffer. Then the DACR itself will be loaded from the pre-buffer whenever the counter reaches zero. This can prevent an overwritten of data in DACR during conversion.

3.2. ADC of LPC1769

Features

12-bit successive approximation analog to digital converter.
Input multiplexing among 8 pins.
Power-down mode.
Measurement range VREFN to VREFP (typically 3 V; not to exceed VDDA voltage level).
12-bit conversion rate of 200 kHz.
Burst conversion mode for single or multiple inputs.
Optional conversion on transition on input pin or timer match signal.

- Pin map for ADC
- Registers set up
    - PINSEL 0,1,3: select pin function to be ADC
    - PINMODE 0,1,3: select no pull-up, no pull-down resister
    - VREFP and VREFN are internally connected by Xpresso board



Register Description

Generic Name	Description	Access	Reset value[1]	AD0 Name & Address
ADCR	A/D Control Register. The ADCR register must be written to select the operating mode before A/D conversion can occur.	R/W	1	AD0CR - 0x4003 4000
ADGDR	A/D Global Data Register. This register contains the ADC's DONE bit and the result of the most recent A/D conversion.	R/W	NA	AD0GDR - 0x4003 4004
ADINTEN	A/D Interrupt Enable Register. This register contains enable bits that allow the DONE flag of each A/D channel to be included or excluded from contributing to the generation of an A/D interrupt.	R/W	0x100	AD0INTEN - 0x4003 400C
ADDR0	A/D Channel 0 Data Register. This register contains the result of the most recent conversion completed on channel 0.	RO	NA	AD0DR0 - 0x4003 4010
ADDR1	A/D Channel 1 Data Register. This register contains the result of the most recent conversion completed on channel 1.	RO	NA	AD0DR1 - 0x4003 4014
ADDR2	A/D Channel 2 Data Register. This register contains the result of the most recent conversion completed on channel 2.	RO	NA	AD0DR2 - 0x4003 4018
ADDR3	A/D Channel 3 Data Register. This register contains the result of the most recent conversion completed on channel 3.	RO	NA	AD0DR3 - 0x4003 401C
ADDR4	A/D Channel 4 Data Register. This register contains the result of the most recent conversion completed on channel 4.	RO	NA	AD0DR4 - 0x4003 4020
ADDR5	A/D Channel 5 Data Register. This register contains the result of the most recent conversion completed on channel 5.	RO	NA	AD0DR5 - 0x4003 4024
ADDR6	A/D Channel 6 Data Register. This register contains the result of the most recent conversion completed on channel 6.	RO	NA	AD0DR6 - 0x4003 4028
ADDR7	A/D Channel 7 Data Register. This register contains the result of the most recent conversion completed on channel 7.	RO	NA	AD0DR7 - 0x4003 402C
ADSTAT	A/D Status Register. This register contains DONE and OVERRUN flags for all of the A/D channels, as well as the A/D interrupt/DMA flag.	RO	0	AD0STAT - 0x4003 4030
ADTRM	ADC trim register.	R/W	0x0000 0F00	AD0TRM - 0x4003 4034

[1] Reset value reflects the data stored in used bits only. It does not include reserved bits content.

Operation

- There are 8 channels. We can select one or multiple channels to operate from SEL bit of AD0CR[7:0]
- ADC needs a clock frequency of 13 MHz. It can be slower by using divider factor specified by CLKDIV value in AD0CR[15:8]. Usually faster is better.
- ADC can operate as a single or repeated conversion. Highest speed for repeated conversion is 200 kHz. Therefore, according Nyquist criterion, the fastest input signal frequency is 200/2 kHz. However, usually input frequency is several times slower than that. When multiple channel are used in repeated mode, it starts scanning from lower enable channel number to higher enable channel number (such as from channel 0 to 7)
- There are 8 conditions available to start the single conversion such as edge-triggered signals from MAT, CAP pin
- When conversion is complete, the most recent result is available to access in two registers.
    - The A/D Global Data register (**AD0GDR**): contains the results of selected input pins (R/W)

- The A/D Data registers (**AD0DR 0-7**): contains the result of each individual input pin (Read Only)
- It is important to consistently access just only AD0GDR or AD0DR since its flag bits can otherwise get out of synch
- Result = x000 means voltage on the input pin was less than, equal to, or close to that on VREFN (i.e. GNDA)
- Result = xFFF means voltage on the input pin is close to, equal to, or greater than that on VREFP (i.e. VDDA)

- We can set ADC to generate interrupt signal for each channel when conversion on that channel is complete. This is done by setting bit in the A/D Interrupt Enable Register (**AD0INTEN**)
- There are two types of status to check
  - DONE: meaning that the conversion is complete
  - OVERRUN: meaning that one or more conversions was (were) lost and overwritten before the conversion that produced the result in the RESULT bits.
  - Both status are available in both data registers (**AD0GDR** and **AD0DR**). They are also mirrored to A/D Status register (**ADSTAT**).

The detail how to configure ADC registers can be found in the LPC1769 user's guide chapter 29 page 585-589

6. Programming DAC and ADC using mbed library

API summary for mbed analog output

Function	Usage
AnalogOut	Create an AnalogOut object connected to the specified pin
write	Set the output voltage, specified as a percentage (float)
write_u16	Set the output voltage, represented as an unsigned short in the range [0x0, 0xFFFF]
read	Return the current output voltage setting, measured as a percentage (float)
operator=	An operator shorthand for write()

Example1

```
/* Sawtooth waveform on DAC output. View on oscilloscope
*/
#include "mbed.h"
```

```
AnalogOut Aout(p0_26);
float i;
int main() {
        while (1){
                for (i=0;i<1;i=i+0.1){ // i is incremented in steps of 0.1
                Aout=i;
                wait(0.001); // wait 1 millisecond
                }
        }
}
```

Note that by default, **Aout** takes a floating point number between 0.0 and 1.0 and outputs this to port 0.26. The actual output voltage on port 0.26 is between 0 V and 3.3 V, so the floating point number that you output is scaled to this.

API summary for mbed analog input

Function	Usage
AnalogIn	Create an AnalogIn object, connected to the specified pin
read	Read the input voltage, represented as a float in the range (0.0-1.0)
read_u16	Read the input voltage, represented as an unsigned short in the range (0x0e0xFFFF)

Example2.

```
/*Program Example2: Inputs signal through ADC, and outputs to DAC. View DAC output on
oscilloscope. To demonstrate Nyquist, connect variable frequency signal generator to
ADC input. Allows measurement of conversion times, and explores Nyquist limit.
*/
#include "mbed.h"
AnalogOut Aout(p0_26); //defines analog output on port 0.26
AnalogIn Ain(p0_23); //defines analog input on port 0.23
DigitalOut test(p0_9);
float ADCdata;
int main() {
      while(1) {
            ADCdata=Ain; //starts A-D conversion, and assigns analog value to ADCdata
            Aout=ADCdata; // transfers stored value to DAC, and forces a D-A conversion
      }
}
```

7. Experiment

1. Testing DAC using Example 1. Measure output signal with oscilloscope. Edit the program to generate

   1.1 A smoother sawtooth signal at frequency 1kHz and peak-to-peak amplitude 1 V

   1.2 A triangle waveform at frequency 1kHz and peak-to-peak amplitude 1 V

   1.3 A sinusoidal waveform at frequency 1kHz and peak-to-peak amplitude 1 V

2. Testing ADC using Example 2. Using a signal generator to generate 1 kHz sin wave with 1V peak-to-peak amplitude and 500 mV offset. Measure the input and the output signals with oscilloscope channel 1 and channel 2 respectively. Observe the difference between the input and the output waveform.

   2.1 Slowly increase frequency of input signal to find the highest input frequency that still produces an acceptable output waveform.

3. Program DAC registers to produce the same results as in question 1.

4. Program ADC registers to produce the same results as in question 2.

8. Reference

[1] Rob Toulson and Tim Wilmshurst, “Fast and Effective Embedded Systems Design Applying the ARM mbed”, Elsevier 2012.
[2] UM10360 LPC176x/5x User manual, NXP, April 2014