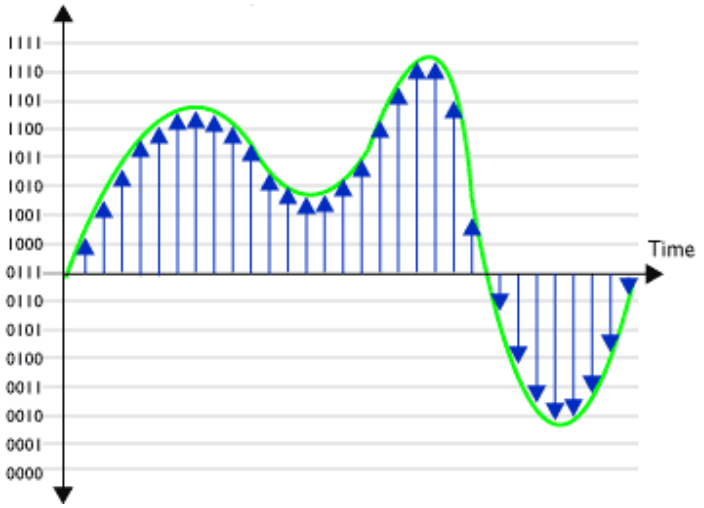
Analogs Input/Output

Analog **D**igital **C**onverter

Digital **A**nalog **C**onverter

ADC: Analog Digital Converter

- What is the ADC – Analog to Digital Conversion?
  - Signals in the nature to be controlled by any embedded system are normally ANALOG.
  - In order to work with those signals the embedded systems need a code representing the instantaneous value of each signal.



- Important parameters of the conversion:
  - Conversion code – number of bits
  - SPAN.
  - Number and type of analog inputs (differencial…)
  - Conversion time – Sampling time
  - Conversion errors: quantification, gain, lineality, offset …

ADC: Characteristics

- LPC1768 has a 12bits A/D converter:
  - SAR based conversor.
  - 8 multiplexed analog inputs.
  - Max. conversion frequency 200KHz.
    - → Conversion time 5us
    - → 65 x T_{ADC}, $f_{ADC} \leq 13MHz$
  - SPAN defined between V_{REFN} to V_{REFP} ($V_{REFP} - V_{REFN} \leq V_{DDA}$)
  - **Single** (manual) or **Burst** conversion (continual conversion)
  - Conversion can be started:
    - With a direct order (software)
    - With a transition of an external pin.
    - With a timer (output compare).

ADC: Pins and Registers

Pins:

AD0.0 to AD0.7	Analog inputs, The ADC can read the voltage in any of these 8 pins, the digital function is disabled in a pin when ADC is enabled in it.
V_{REFP}, V_{REFN}	Voltage references, Provide the work range for the convertion. It is usually 3.3v and GND.
V_{DDA}, V_{SSA}	Analog power and ground, they are also 3.3v (isolation is recommended in the user guide but not done here) and GND.

Registers:

ADCR	A/D Control Register. The ADCR register must be written to select the operating mode before A/D conversion can occur.
ADGDR	A/D Global Data Register. This register contains the ADC's DONE bit and **the result of the most recent A/D conversion**.
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.
ADDR0 to ADDR7	A/D Channel Data Register. This register contains the result of the most recent conversion completed on each from 0 to 7.
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.

ADC Registers

Name	Access	Description	Address Offset
AD0CR	R/W	A/D Control Register	0x0000
AD0GDR	R/W	A/D Global Data Register	0x0004
AD0INTEN	R/W	A/D Interrupt Enable Register	0x000C
AD0DR0	R/W	A/D Channel 0 Data Register	0x0010
AD0DR1	R/W	A/D Channel 1 Data Register	0x0014
AD0DR2	R/W	A/D Channel 2 Data Register	0x0018
AD0DR3	R/W	A/D Channel 3 Data Register	0x001C
AD0DR4	R/W	A/D Channel 4 Data Register	0x0020
AD0DR5	R/W	A/D Channel 5 Data Register	0x0024
AD0DR6	R/W	A/D Channel 6 Data Register	0x0028
AD0DR7	R/W	A/D Channel 7 Data Register	0x002C
AD0STAT	RO	A/D Status Register	0x0030

ADC: Configuration

- **Basic configuration**
  - Power on ADC with PCADC (PCONP)
  - Configure PCLK_ADC in PCLKSEL0 (Default F_cclk/4).
  - Configure pins for ADC inputs in PINSEL and PINMODE.
    - No pull-up nor pull-down
  - Enable IRQ in NVIC.
  - Select Interrupt Priority in NVIC.

- **Advanced configuration**
  - **Fine tunning the ADC (ADTRIM)**
    - For tunning ADC and DAC internal offset.
  - **With DMA**
    - To send conversions directly to memory.
    - When configured, DMA starts automatically with IRQ (disable NVIC)
    - To be used in BURST mode.
    - DMA configured to transfer 1, 4 or 8 words according to the burst length.

ADC: Conversion modes

- **Manual** mode (software controlled)
  - Power on ADC: PDN(AD0CR[21]) = 1
  - Configure manual model: BURST(AD0CR[16]) = 0
  - Conversion for a single channel, selected in SEL(AD0CR[7:0])
  - Start the conversion in START(AD0CR[26:24]) + EDGE(AD0CR[27])
  - Can be started with an event in P0.10 (INT0), P1.27(CAP1.0), MAT0.1, MAT0.3, MAT1.0, MAT1.1

- **BURST** mode
  - Power on ADC: PDN(AD0CR[21]) = 1
  - Configure Burst mode: BURST(AD0CR[16]) = 1
  - Select channels to be converted in SEL(AD0CR[7:0]).
  - No manual conversion allowed!!! START(AD0CR[26:24]) = **0b000**

ADC: Results of conversions

- Results can be globally read at **AD0GDR** (Global Data Register)
  - Last conversion in RESULT bits (AD0GDR[**15:4**])
  - Number of last channel converted in CHN bits (AD0GDR[**26:24**])
  - Overrun flag in OVERRUN bit (AD0GDR[30])
    - Clear when read AD0GDR .
  - End of Conversion flag in DONE bit (AD0GDR[31])
    - Clear when read AD0GDR.
    - Can interrupt if AD0GINTEN(AD0INTEN[8]) = 1
- Results can be individually read for each channel in **AD0DRn** (Data Register of Channel **n**)
  - Last conversion for channel **n** in RESULT bits (AD0DRn[**15:4**]).
  - Overrun flag for channel **n** in OVERRUN bit (AD0DRn[**30**])
    - Clear when read AD0DRn
  - End of conversion flag for channel **n** in DONE bit (AD0DRn[**31**])
    - Clear when read AD0DRn
    - Can interrupt if AD0INTEN(ADINTEN[n]) = 1 and AD0INTEN(ADINTEN[8]) = 0

ADC: Control Register

AD0CR

Bit	Symbol	Value	Description	Reset
7:0	SEL		Selects which of the AD0.7:0 pins is (are) to be sampled and converted. For AD0, bit 0 selects Pin AD0.0, and bit 7 selects pin AD0.7. In software-controlled mode, only one of these bits should be 1. In hardware scan mode, any value containing 1 to 8 ones is allowed. All zeroes is equivalent to 0x01.	0x01
15:8	CLKDIV		The APB clock (PCLK_ADC0) is divided by (this value plus one) to produce the clock for the A/D converter, which should be less than or equal to 13 MHz.	0
16	BURST	1	The AD converter does repeated conversions at up to 200 kHz, scanning (if necessary) through the pins selected by bits set to ones in the SEL field. **Remark:** START bits must be 000 when BURST = 1 or conversions will not start.	0
		0	Conversions are software controlled and require 65 clocks.	
20:17	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
21	PDN	1	The A/D converter is operational.	0
		0	The A/D converter is in power-down mode.	
23:22	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
26:24	START		When the BURST bit is 0, these bits control whether and when an A/D conversion is started:	0
		000	No start (this value should be used when clearing PDN to 0).	
		001	Start conversion now.	
		010	Start conversion when the edge selected by bit 27 occurs on the P2.10 / EINT0 / NMI pin.	
		011	Start conversion when the edge selected by bit 27 occurs on the P1.27 / CLKOUT / USB_OVRCRn / CAP0.1 pin.	
		100	Start conversion when the edge selected by bit 27 occurs on MAT0.1.	
		101	Start conversion when the edge selected by bit 27 occurs on MAT0.3.	
		110	Start conversion when the edge selected by bit 27 occurs on MAT1.0.	
		111	Start conversion when the edge selected by bit 27 occurs on MAT1.1.	
27	EDGE		This bit is significant only when the START field contains 010-111. In these cases:	0
		1	Start conversion on a falling edge on the selected CAP/MAT signal.	
		0	Start conversion on a rising edge on the selected CAP/MAT signal.	
31:28	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

ADC: Global Data Register

AD0GDR

Table 532: A/D Global Data Register (AD0GDR - address 0x4003 4004) bit description

Bit	Symbol	Description	Reset value
3:0	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
15:4	RESULT	When DONE is 1, this field contains a binary fraction representing the voltage on the AD0[n] pin selected by the SEL field, as it falls within the range of V_{REFP} to V_{REFN}. Zero in the field indicates that the voltage on the input pin was less than, equal to, or close to that on V_{REFN}, while 0xFFF indicates that the voltage on the input was close to, equal to, or greater than that on V_{REFP}.	NA
23:16	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
26:24	CHN	These bits contain the channel from which the RESULT bits were converted (e.g. 000 identifies channel 0, 001 channel 1...).	NA
29:27	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
30	OVERRUN	This bit is 1 in burst mode if the results of one or more conversions was (were) lost and overwritten before the conversion that produced the result in the RESULT bits. This bit is cleared by reading this register.	0
31	DONE	This bit is set to 1 when an A/D conversion completes. It is cleared when this register is read and when the ADCR is written. If the ADCR is written while a conversion is still in progress, this bit is set and a new conversion is started.	0

ADC: Global Data Register

AD0INTEN

Table 533: A/D Status register (AD0INTEN - address 0x4003 400C) bit description

Bit	Symbol	Value	Description	Reset value
0	ADINTEN0	0	Completion of a conversion on ADC channel 0 will not generate an interrupt.	0
		1	Completion of a conversion on ADC channel 0 will generate an interrupt.	
1	ADINTEN1	0	Completion of a conversion on ADC channel 1 will not generate an interrupt.	0
		1	Completion of a conversion on ADC channel 1 will generate an interrupt.	
2	ADINTEN2	0	Completion of a conversion on ADC channel 2 will not generate an interrupt.	0
		1	Completion of a conversion on ADC channel 2 will generate an interrupt.	
3	ADINTEN3	0	Completion of a conversion on ADC channel 3 will not generate an interrupt.	0
		1	Completion of a conversion on ADC channel 3 will generate an interrupt.	
4	ADINTEN4	0	Completion of a conversion on ADC channel 4 will not generate an interrupt.	0
		1	Completion of a conversion on ADC channel 4 will generate an interrupt.	
5	ADINTEN5	0	Completion of a conversion on ADC channel 5 will not generate an interrupt.	0
		1	Completion of a conversion on ADC channel 5 will generate an interrupt.	
6	ADINTEN6	0	Completion of a conversion on ADC channel 6 will not generate an interrupt.	0
		1	Completion of a conversion on ADC channel 6 will generate an interrupt.	
7	ADINTEN7	0	Completion of a conversion on ADC channel 7 will not generate an interrupt.	0
		1	Completion of a conversion on ADC channel 7 will generate an interrupt.	
8	ADGINTEN	0	Only the individual ADC channels enabled by ADINTEN7:0 will generate interrupts.	1
		1	Only the global DONE flag in ADDR is enabled to generate an interrupt.	
31:17	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

ADC: Data Registers

AD0DR0…AD0DR7

Table 534: A/D Data Registers (AD0DR0 to AD0DR7 - 0x4003 4010 to 0x4003 402C) bit description

Bit	Symbol	Description	Reset value
3:0	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
15:4	RESULT	When DONE is 1, this field contains a binary fraction representing the voltage on the AD0[n] pin, as it falls within the range of V_{REFP} to V_{REFN}. Zero in the field indicates that the voltage on the input pin was less than, equal to, or close to that on V_{REFN}, while 0xFFF indicates that the voltage on the input was close to, equal to, or greater than that on V_{REFP}.	NA
29:16	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
30	OVERRUN	This bit is 1 in burst mode if the results of one or more conversions was (were) lost and overwritten before the conversion that produced the result in the RESULT bits.This bit is cleared by reading this register.	
31	DONE	This bit is set to 1 when an A/D conversion completes. It is cleared when this register is read.	NA

ADC: Status Register

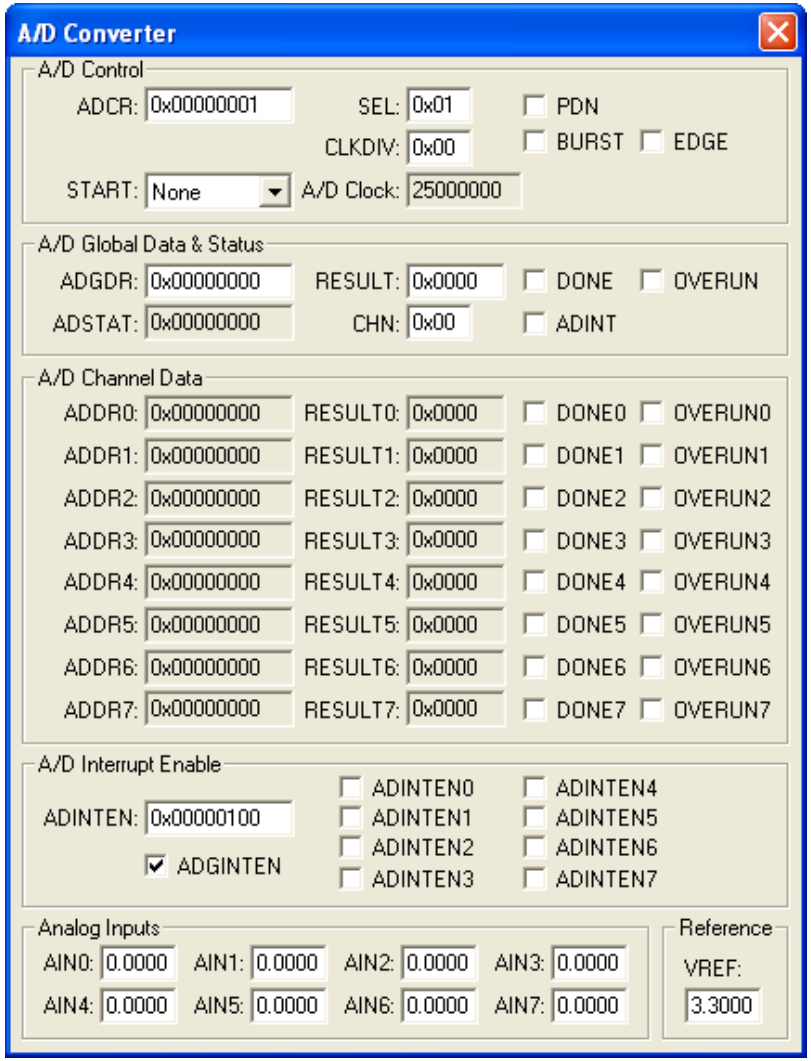
AD0STAT

Table 535: A/D Status register (AD0STAT - address 0x4003 4030) bit description

Bit	Symbol	Description	Reset value
0	DONE0	This bit mirrors the DONE status flag from the result register for A/D channel 0.	0
1	DONE1	This bit mirrors the DONE status flag from the result register for A/D channel 1.	0
2	DONE2	This bit mirrors the DONE status flag from the result register for A/D channel 2.	0
3	DONE3	This bit mirrors the DONE status flag from the result register for A/D channel 3.	0
4	DONE4	This bit mirrors the DONE status flag from the result register for A/D channel 4.	0
5	DONE5	This bit mirrors the DONE status flag from the result register for A/D channel 5.	0
6	DONE6	This bit mirrors the DONE status flag from the result register for A/D channel 6.	0
7	DONE7	This bit mirrors the DONE status flag from the result register for A/D channel 7.	0
8	OVERRUN0	This bit mirrors the OVERRRUN status flag from the result register for A/D channel 0.	0
9	OVERRUN1	This bit mirrors the OVERRRUN status flag from the result register for A/D channel 1.	0
10	OVERRUN2	This bit mirrors the OVERRRUN status flag from the result register for A/D channel 2.	0
11	OVERRUN3	This bit mirrors the OVERRRUN status flag from the result register for A/D channel 3.	0
12	OVERRUN4	This bit mirrors the OVERRRUN status flag from the result register for A/D channel 4.	0
13	OVERRUN5	This bit mirrors the OVERRRUN status flag from the result register for A/D channel 5.	0
14	OVERRUN6	This bit mirrors the OVERRRUN status flag from the result register for A/D channel 6.	0
15	OVERRUN7	This bit mirrors the OVERRRUN status flag from the result register for A/D channel 7.	0
16	ADINT	This bit is the A/D interrupt flag. It is one when any of the individual A/D channel Done flags is asserted and enabled to contribute to the A/D interrupt via the ADINTEN register.	0
31:17	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

ADC: Keil Debugger

- Peripherals→A/D converter



ADC: Electrical Characteristics (I)

ADC Electrical Characteristics

Table 18. ADC characteristics (full resolution)

V_{DDA} = 2.7 V to 3.6 V; T_{amb} = −40 °C to +85 °C unless otherwise specified; ADC frequency 13 MHz; 12-bit resolution.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{IA}	analog input voltage		0	-	V_{DDA}	V
C_{ia}	analog input capacitance		-	-	15	pF
E_D	differential linearity error	[1][2]	-	-	±1	LSB
$E_{L(adj)}$	integral non-linearity	[3]	-	-	±3	LSB
E_O	offset error	[4][5]	-	-	±2	LSB
E_G	gain error	[6]	-	-	0.5	%
E_T	absolute error	[7]	-	-	4	LSB
R_{vsi}	voltage source interface resistance	[8]	-	-	7.5	kΩ
$f_{clk(ADC)}$	ADC clock frequency		-	-	13	MHz
$f_{c(ADC)}$	ADC conversion frequency	[9]	-	-	200	kHz

[1] The ADC is monotonic, there are no missing codes.

[2] The differential linearity error (E_D) is the difference between the actual step width and the ideal step width. See Figure 26.

[3] The integral non-linearity ($E_{L(adj)}$) is the peak difference between the center of the steps of the actual and the ideal transfer curve after appropriate adjustment of gain and offset errors. See Figure 26.

[4] The offset error (E_O) is the absolute difference between the straight line which fits the actual curve and the straight line which fits the ideal curve. See Figure 26.

[5] ADCOFFS value (bits 7:4) = 2 in the ADTRM register. See *LPC17xx user manual UM10360*.

[6] The gain error (E_G) is the relative difference in percent between the straight line fitting the actual transfer curve after removing offset error, and the straight line which fits the ideal transfer curve. See Figure 26.

[7] The absolute error (E_T) is the maximum difference between the center of the steps of the actual transfer curve of the non-calibrated ADC and the ideal transfer curve. See Figure 26.

[8] See Figure 27.

[9] The conversion frequency corresponds to the number of samples per second.

ADC: Electrical Characteristics (II)

ADC Electrical Characteristics

Table 19. ADC characteristics (lower resolution)

T_{amb} = –40 °C to +85 °C unless otherwise specified; 12-bit ADC used as 10-bit resolution ADC.

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
E_D	differential linearity error		[1][2]	-	±1	-	LSB
$E_{L(adj)}$	integral non-linearity		[3]	-	±1.5	-	LSB
E_O	offset error		[4]	-	±2	-	LSB
E_G	gain error		[5]	-	±2	-	LSB
$f_{clk(ADC)}$	ADC clock frequency	$3.0\ V \le V_{DDA} \le 3.6\ V$		-	-	33	MHz
		$2.7\ V \le V_{DDA} < 3.0\ V$		-	-	25	MHz
$f_{c(ADC)}$	ADC conversion frequency	$3\ V \le V_{DDA} \le 3.6\ V$	[6]	-	-	500	kHz
		$2.7\ V \le V_{DDA} < 3.0\ V$	[6]	-	-	400	kHz

[1] The ADC is monotonic, there are no missing codes.

[2] The differential linearity error (E_D) is the difference between the actual step width and the ideal step width. See Figure 26.

[3] The integral non-linearity ($E_{L(adj)}$) is the peak difference between the center of the steps of the actual and the ideal transfer curve after appropriate adjustment of gain and offset errors. See Figure 26.

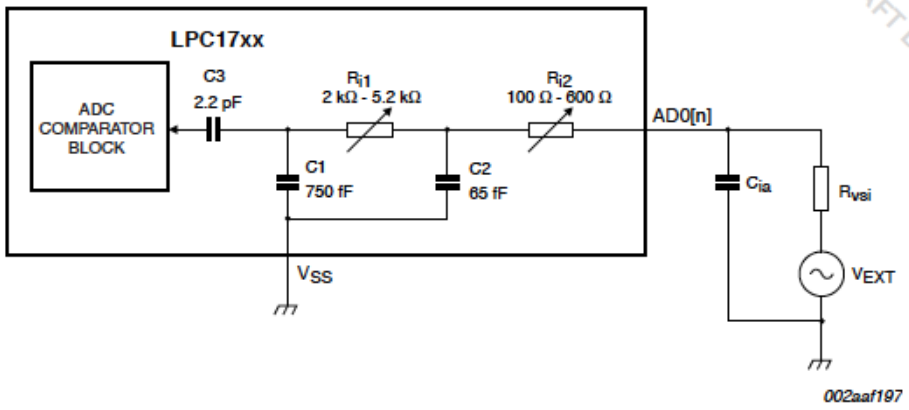
[4] The offset error (E_O) is the absolute difference between the straight line which fits the actual curve and the straight line which fits the ideal curve. See Figure 26.

[5] The gain error (E_G) is the relative difference in percent between the straight line fitting the actual transfer curve after removing offset error, and the straight line which fits the ideal transfer curve. See Figure 26.

[6] The conversion frequency corresponds to the number of samples per second.

ADC: Electrical Characteristics (III)

- ADC Electrical Characteristics



The values of resistor components R_{i1} and R_{i2} vary with temperature and input voltage and are process-dependent (see Table 20).

Parasitic resistance and capacitance from the pad are not included in this figure.

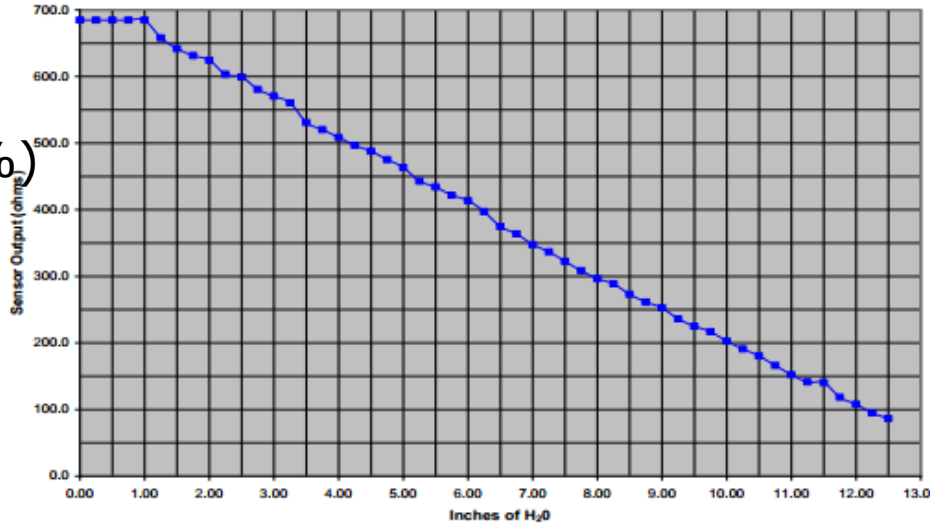
Fig 27. ADC interface to pins AD0[n]

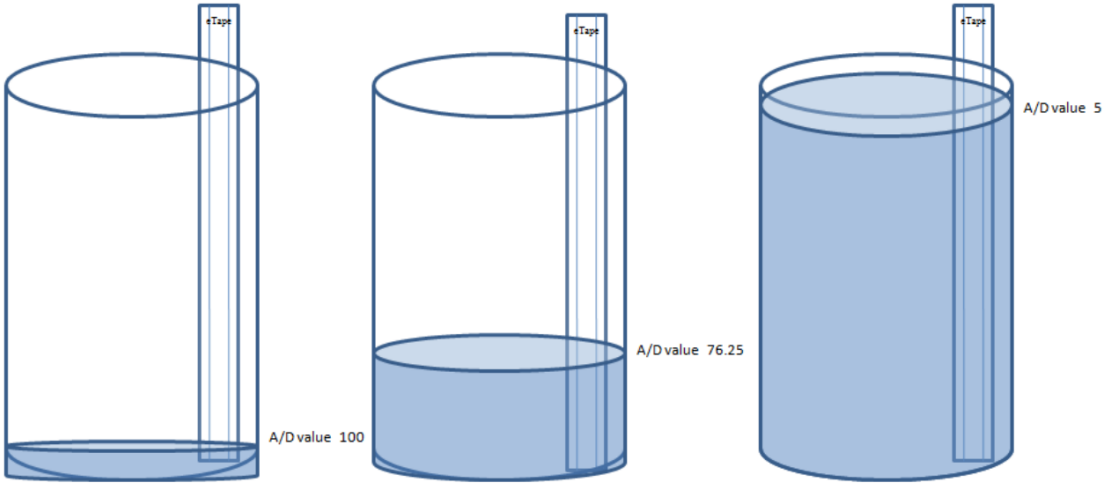
Table 20. ADC interface components

Component	Range	Description
R_{i1}	2 kΩ to 5.2 kΩ	Switch-on resistance for channel selection switch. Varies with temperature, input voltage, and process.
R_{i2}	100 Ω to 600 Ω	Switch-on resistance for the comparator input switch. Varies with temperature, input voltage, and process.
C1	750 fF	Parasitic capacitance from the ADC block level.
C2	65 fF	Parasitic capacitance from the ADC block level.
C3	2.2 pF	Sampling capacitor.

ADC: Aplicattions (sensors)

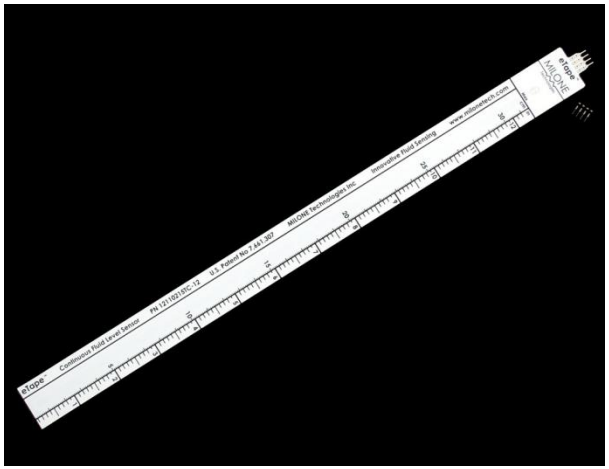
- ***Milone eTape***
- Liquid level sensor:
- Variable resistor (60-550 Ω ± 20%)





(A/D max – A/D meas) / (A/D max – A/D min) = % Full

(100 – 76.25) / (100 - 5) = .25 or 25% full



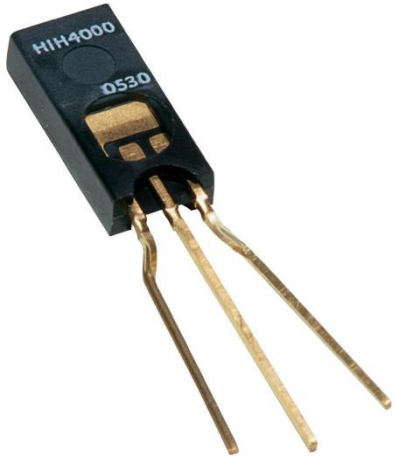
ADC: Aplicattions (sensors)

HIH-4000-001

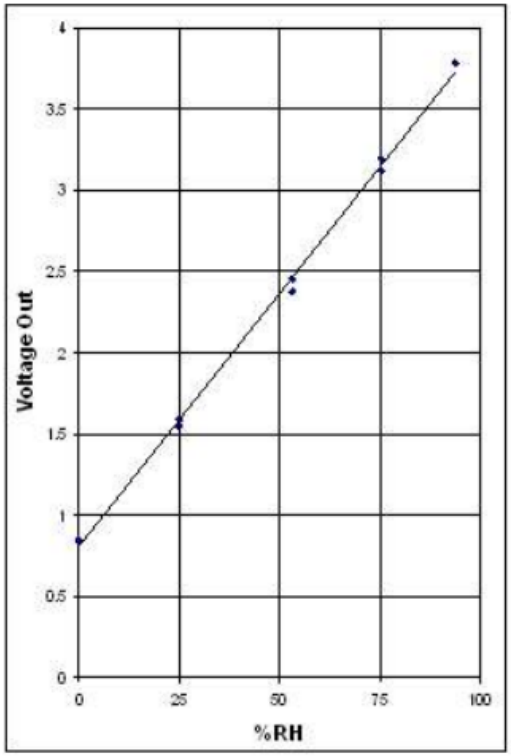
HIH-4000 Series Integrated Circuity Humidity Sensor, 2,54 mm (0.100 in) Lead Pitch SIP

Features

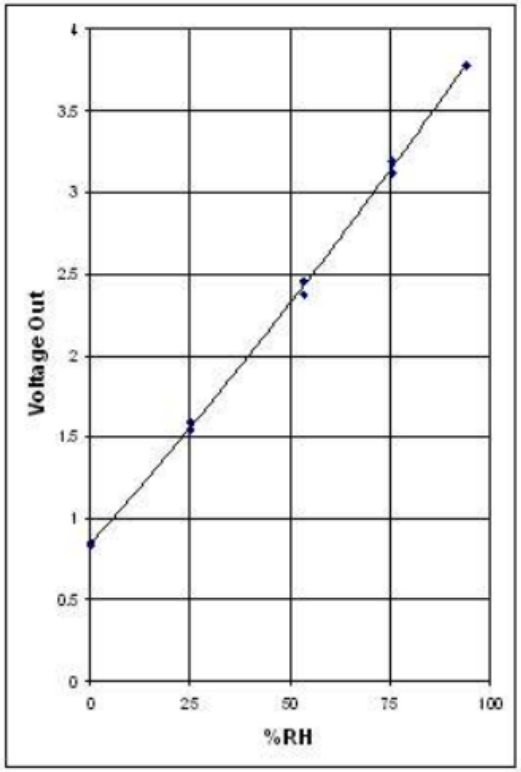
- Molded thermoset plastic housing with cover
- Linear voltage output vs %RH
- Laser trimmed interchangeability
- Low power design
- High accuracy
- Fast response time
- Stable, low drift performance
- Chemically resistant



TYPICAL BEST FIT STRAIGHT LINE



TYPICAL 2nd ORDER CURVE FIT



Digital **A**nalog **C**onverter

DAC: Digital Analog Converter

DAC Characteristics

- 10bits DAC conversor.
- Decoupled low impedance output (buffered).
- F_{DAC} max 1MHz.

Operation:

- Configure pins for ADC inputs in PINSEL (no PCONP)
- Conversion sequence:
  1. Data to be converted is written to VALUE bits (DACR[15:6])
  2. DAC output settles to:

$$\mathbf{Vo = VALUE \times ((V_{REFP} - V_{REFN})/1023) + V_{REFN}}$$

  1. Settling time can be configured in BIAS bit (DACR[16])
     - 0 → 1us max. 700 uA max. f_{DAC} max 1MHz
     - 1 → 2,5 us max, 350 uA max. f_{DAC} max 400KHz

DAC: Registers (I)

Table 538. DAC registers

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**. D/A Converter Register

Table 539: D/A Converter Register (DACR - address 0x4008 C000) bit description

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

DAC: Registers (II)

- **DACCTRL**. D/A Control register

Table 540. D/A Control register (DACCTRL - address 0x4008 C004) bit description

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 543).	
31:4	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

- **DACCTNVAL.** D/A Converter Counter Value register

Table 541: D/A Converter register (DACR - address 0x4008 C008) bit description

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

DAC: Electrical Characteristics

- DAC Electrical Characteristics

Table 21. DAC electrical characteristics

V_{DDA} = 2.7 V to 3.6 V; T_{amb} = –40 °C to +85 °C unless otherwise specified

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
E_D	differential linearity error		-	±1	-	LSB
$E_{L(adj)}$	integral non-linearity		-	±1.5	-	LSB
E_O	offset error		-	0.6	-	%
E_G	gain error		-	0.6	-	%
C_L	load capacitance		-	200	-	pF
R_L	load resistance		1	-	-	kΩ

DAC_senoidal: Ejemplo 1

Generación señal senoidal de frecuencia F_out

- Generar N muestras/ciclo de 10 bits en array

```
void genera_muestras(uint16_t muestras_ciclo)
{
  uint16_t i;
  //señal senoidal
  for(i=0;i<muestras_ciclo;i++)
  muestras[i]=(uint32_t)(511+511*sin(2*pi*i/N_muestras)); // Ojo! el DAC es de 10bits
}
```

- Configurar Timer para interrupciones periódicas a Fout*N_muestras

```
void init_TIMER1(void)
{
    LPC_SC->PCONP|=(1<<2);              //  Power ON
    LPC_TIM1->PR = 0x00;                //  Prescaler =1
    LPC_TIM1->MCR = 0x03;               //  Reset TC on Match, e interrumpe!
    LPC_TIM1->MR0 = (F_pclk/F_out/N_muestras)-1;  // Cuentas hasta el Match
    LPC_TIM1->EMR = 0x00;               //  No actúa sobre el HW
    LPC_TIM1->TCR = 0x01;               //  Start Timer
    NVIC_EnableIRQ(TIMER1_IRQn);        //  Habilita NVIC
    NVIC_SetPriority(TIMER1_IRQn,1);    //  Nivel 1 prioridad
}
```

- Interrupción del Timer

```
void TIMER1_IRQHandler(void)
{
  static uint16_t indice_muestra;
  LPC_TIM1->IR|= (1<<0);                          // borrar flag
  LPC_DAC->DACR= muestras[indice_muestra++] << 6; // bit6..bit15
  indice_muestra&= N_muestras-1;                  // contador circular
}
```

DAC_audio: Ejemplo 2

Gen. señal de audio (muestras en Flash mediante script en Matlab)

- Script en Matlab para generar fichero ***audio_muestras.h*** que contiene *array* con muestras de 8 bits.

```
close all;
clear all;
escala=100; %factor para ajustar la amplitud de las muestras entre 0-255 (DAC de 8 bits)

r = audiorecorder(8000, 16, 1); % audiorecorder(Fs, N_bits, canales);

record(r); % grabando....
pause(2.0); % 2 segundos
stop(r); % stop

play(r); % reproduce lo grabado
datos = getaudiodata(r, 'int16'); % get data as int16 array
```

DAC_audio: Ejemplo 2 (cont.)

Gen. señal de audio (muestras en Flash mediante script en Matlab)

```
y=datos/escala; % valores entre -127 y 128
figure(1)
plot(y)

y=abs(min(y))+y; % Sin valores negativos (escribir en el DAC valores entre 0 y 255)
figure(2)
x=redondeo(y); % DAC 8 bits (ojo valores enteros!! menores que 255)
plot(x);
pause(0); % Observa la amplitud y modifica el factor de escala si se precisa
L=length(x)
```

```
fichero = 'audio_muestras.h';
fid=fopen(fichero, 'w'); % generamos el array,  muestras[]={}
fprintf(fid,'const uint8_t muestras[]={');
for j=1:L
    if j == L
    fprintf(fid,'%d',[x(j)]);

    else fprintf(fid,'%d,',[x(j)]); % Añado las comas
    end
end
fprintf(fid,'}; \n\r');
fclose(fid);
```

DAC_audio: Ejemplo 2 (cont.)

Gen. señal de audio (muestras en Flash mediante script en Matlab)

- Funciones en Keil (reproducción señal de audio)
  - Configuración interrupción externa 0 (P2.10) pulsador ISP en Mini-DK2

```
void init_Externas()
{
// Configuración interrupciones externas
LPC_PINCON->PINSEL4|=(0x01<<20);     // P2.10 es entrada interrup. EXT 0 (KEY 1 en MIni-DK2)
LPC_SC->EXTMODE|=(1<<0);             // Por Flanco,
LPC_SC->EXTPOLAR=0;                  // de bajada
NVIC_SetPriority(EINT0_IRQn, 1);     // Menor prioritaria!!! ;  sin CMSIS: NVIC->IP[18]=(4<<3);
NVIC_EnableIRQ(EINT0_IRQn);          // sin CMSIS: NVIC->ISER[0]=(1<<18);
}
```

  - Interrupción pulsador ISP (P2.10) inicio reproducción

```
void EINT0_IRQHandler()
{
LPC_SC->EXTINT=(1<<0); // Borrar flag Externa 0
LPC_TIM1->TCR=0x01;   //Start Timer .
}
```

  - Interrupción del Timer 1 a Fs → Saca las muestras almacenadas en memoria

```
void TIMER1_IRQHandler(void)
{
  static uint16_t indice_muestra;
  LPC_TIM1->IR|= (1<<0);                          // Borrar flag
  LPC_DAC->DACR= muestras[indice_muestra++]<<8; // 8 bits!!!!
  if(indice_muestra==N_muestras-1){
                                  indice_muestra=0;
                                  LPC_TIM1->TCR=0x02; //Stop Timer and reset, DAC= 0V.
                                  LPC_DAC->DACR=0;    // 0 V
                              }
}
```

ADC_DAC: Ejemplo 3

Generación señal senoidal y muestreo de entrada analógica (AD0.0)

- Función que genera N muestras de un ciclo (señal senoidal) y almacena en array

```
void genera_muestras(uint8_t muestras_ciclo)
{
  uint8_t i;
  for(i=0;i<muestras_ciclo;i++)
    muestras[i]=1023*(0.5 + 0.5*sin(2*pi*i/muestras_ciclo)); // Ojo! el DAC es de 10bits
}
```

- Configuración del DAC y del Timer 1 que saca las muestras al DAC

```
// Timer 1 interrumpe periódicamente a F = F_out x N_muestras !!!!
// La muestra correpondiente del array,  se saca al DAC en cada interrupción
void TIMER1_IRQHandler(void)
{
  static uint8_t indice_muestra;
  LPC_TIM1->IR|= (1<<0);              //
  LPC_DAC->DACR= muestras[indice_muestra++] << 6; // ?
  indice_muestra&= N_muestras-1;              // ?
}

void init_DAC(void)
{
  LPC_PINCON->PINSEL1|= (2<<20);     // DAC output = P0.26 (AOUT)
  LPC_PINCON->PINMODE1|= (2<<20);    // Deshabilita pullup/pulldown
  LPC_SC->PCLKSEL0|= (0x00<<22);     // CCLK/4 (Fpclk después del reset) (100 Mhz/4 = 25Mhz)
  LPC_DAC->DACCTRL=0;                //
}
```

ADC_DAC: Ejemplo 3 (cont.)

Generación señal senoidal y muestreo de entrada analógica (AD0.0)

- Función de IRQ del ADC (Interrumpe a Fs)

```
void ADC_IRQHandler(void)
{
  voltios=  ((LPC_ADC->ADGDR >>4)&0xFFF)*3.3/4095; // se borra automat. el flag DONE al leer ADCGDR
}
```

- Configuración del ADC y del Timer 0 que fija la Fs

```
void init_ADC(void)
{
  LPC_SC->PCONP|= (1<<12);           // POwer ON
  LPC_PINCON->PINSEL1|= (1<<14);     // ADC input= P0.23 (AD0.0)
  LPC_PINCON->PINMODE1|= (2<<14);    // Deshabilita pullup/pulldown
  LPC_SC->PCLKSEL0|= (0x00<<8);      // CCLK/4 (Fpclk después del reset) (100 Mhz/4 = 25Mhz)
  LPC_ADC->ADCR= (0x01<<0)|          // Canal 0
                 (0x01<<8)|          // CLKDIV=1    (Fclk_ADC=25Mhz /(1+1)= 12.5Mhz)
                 (0x01<<21)|         // PDN=1
                 (4<<24);            // Inicio de conversión con el Match 1 del Timer 0 (MAT0.1)

  LPC_ADC->ADINTEN=(1<<8);           // Hab. interrupción fin de conversión global (DONE global)
  NVIC_EnableIRQ(ADC_IRQn);          //
  NVIC_SetPriority(ADC_IRQn,2);      //
}

/*  Timer 0 en modo Output Compare (reset T0TC on Match 1)
  Counter clk: 25 MHz   MAT0.1 : On match, Toggle pin/output (P1.29)
  Cada 2 Match se provoca el INICIO DE CONVERSIÓN DEL ADC
  Habilitamos la salida (MAT0.1) para observar la frecuencia de muestreo del ADC */

void init_TIMER0(void)
{
    LPC_SC->PCONP|=(1<<1);             //
    LPC_PINCON->PINSEL3|= 0x0C000000; //
    LPC_TIM0->PR = 0x00;               //
    LPC_TIM0->MCR = 0x10;              //
    LPC_TIM0->MR1 = (F_pclk/F_muestreo/2)-1; // Se han de producir DOS Match para iniciar la conversión!!!!
    LPC_TIM0->EMR = 0x00C2;            //
    LPC_TIM0->TCR = 0x01;              //
}
```