SENG440 Embedded Systems

– Lesson 6: Standard Peripherals–

Mihai SIMA

msima@ece.uvic.ca

Academic Course

© 2019 Mihai SIMA

Copyright © 2019 Mihai SIMA

All rights reserved.

No part of the materials including graphics or logos, available in these notes may be copied, photocopied, reproduced, translated or reduced to any electronic medium or machine-readable form, in whole or in part, without specific permission. Distribution for commercial purposes is prohibited.

Disclaimer

The purpose of this course is to present general techniques and concepts for the analysis, design, and utilization of embedded systems. The requirements of any real embedded system can be intimately connected with the environment in which the embedded system is deployed. The presented design examples should not be used as the full design for any real embedded system.

© 2019 Mihai SIMA

Lesson 6: Standard peripherals for embedded systems

 © 2019 Mihai SIMA

Lesson 6: Course Progress

- **Software optimization techniques**
  - What is wrong with plain software
  - Profile driven compilation
  - Efficient C programming
- **Standard peripherals for embedded systems**
  - **Timers, counters**, watchdog timers, real-time clocks
  - **Digital-to-Analog (D/A) and Analog-to-Digital (A/D) converters**
  - Pulse-Width Modulation (PWM) peripherals
  - Universal Asynchronous Receiver/Transmitters (UART)
- **Hardware – software – firmware.**
  - A taxonomy of digital systems

© 2019 Mihai SIMA

Standard peripherals for embedded systems – Introduction

- Standard peripheral = single-purpose engine that implements a specific Input/Output (I/O) task
- Design a system following a modular approach
  - Choose a (general-purpose) processor (for example, ARM)
  - Augment it with standard peripheral(s) (which is well documented)
  - Program the resulting system
- The standard peripheral can be purchased as a stand-alone device or as an Intellectual Property (IP) block (this distinction is not needed when the architecture is the only concern – why?)
- Peripherals provide the interface to the outer world, thus the **accuracy** is a concern for the designer
- The programmer has the responsibility to deal with this accuracy

© 2019 Mihai SIMA

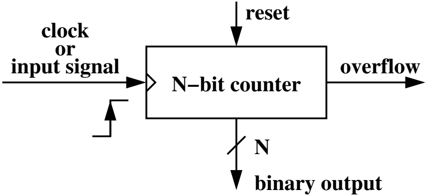
Benefits and drawbacks when using standard peripherals

- Some benefits of the modular approach
  - Non-Recurring Engineering (NRE) cost is typically low, since the standard module is predesigned
  - The unit cost may be low, since the standard module is mass-produced
  - High reliability, since the standard module was used in the past
  - Previous experience with the standard module under consideration is likely to be available
  - The new module can free the host processor for other tasks
- Some drawbacks of the modular approach
  - The resulting design is usually not optimal, so the system size and power consumption may be large
  - Due to rapidly evolving standards, the system is prone to become obsolete and has to be redesigned

© 2019 Mihai SIMA

Timers and counters

- Timer: measures time intervals by counting clock pulses
  - To generate timed output events (e.g., hold trafic light green for 10s)
  - To measure input events (e.g., measure a car's speed)
- Counter: it counts pulses on a general input signal rather than clock
  - For example, it counts cars passing over a sensor



- Examples of standard timers/counters in embedded systems: 8051 has two 16-bit (on-chip) counters, 9513 (stand-alone) chip contains five general-purpose 16-bit counter, to name a few.

© 2019 Mihai SIMA

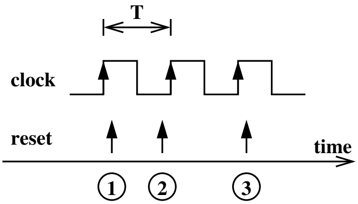
Resolution and accuracy of timers and counters

- **Resolution**: the smallest detectable incremental change of input parameter that can be detected by the instrument.
- **Accuracy**: the maximum difference that will exist between the actual value (which must be measured by a primary or good secondary standard) and the indicated value at the output of the instrument.
- Both resolution and accuracy can be expressed either as a percentage of full scale or in absolute terms.
- Be aware! Accuracy is a measure of the capability of an instrument to faithfully indicate the value of the measured signal. This term is NOT related to resolution; however, the accuracy level can never be better than the resolution of the instrument.
- Timers and counters have a intrinsic error of 2 bits!

© 2019 Mihai SIMA

Resolution and accuracy of timers and counters (cont'd)

- Let **clock** period be 10ns
- If we count 20,000 clock pulses, then 200μs have passed
- N = 16 means the counter will count up to 65,536, that is, after 655.35μs
- **Resolution** is 1 clock period (10ns)
- **Accuracy** is 2 clock periods – in fact, it is $\pm$ 1 clock period ($\pm 10\mathrm{ns}$)



Position 1: **reset** occurs slightly after the rising edge of the clock arrives

Position 2: **reset** occurs slightly before the rising edge of the clock arrives

Position 3: **reset** occurs slightly after the rising edge of the clock arrives

© 2019 Mihai SIMA

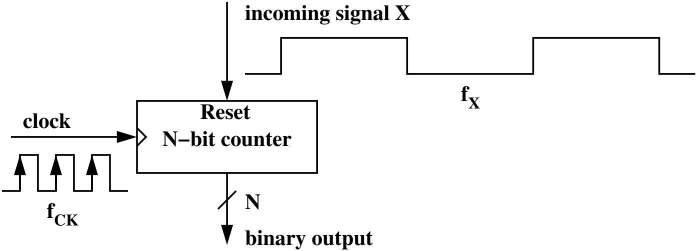
Timer structures – overview

- **Cascaded counters**
  - Example: two 16-bit counters can be cascaded to form a 32-bit counter
- **Interval timer**
  - Signals when a specific time interval has elapsed
- **Reaction timer**
  - Measures time between cause and effect
- **Frequency divider** (prescaler)
  - Divides the clock frequency
- **Watchdog timer**
  - Indicates that it has not been reset for a predefined amount of time
  - Used in detecting failures and/or self-reset

© 2019 Mihai SIMA

Timers and counters: worked example

The frequency, f_X, of the incoming signal X, which drives the reset input of the N-bit counter depicted below, is to be measured.



1. Assuming f_{CK} = 1MHz and N=10 bits, what is the minimum frequency, f_X, that can be measured?
2. Assuming f_{CK} = 10MHz, what is the maximum frequency, f_X, that can be measured?

© 2019 Mihai SIMA

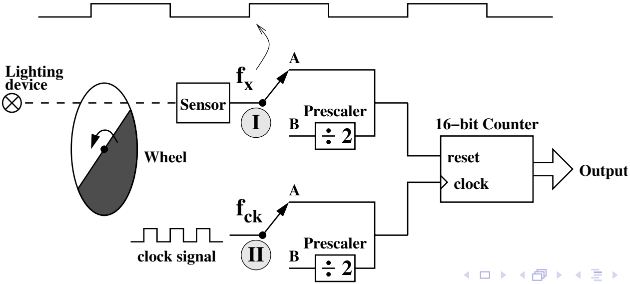
Timers and counters: worked example

- $f_X(\mathrm{min})$ when $f_{CK} = 1\mathrm{MHz}$ and N=10 bits
  - Minimum frequency means maximum period
  - $f_{CK} = 1\mathrm{MHz}$ means $T_{CK} = 1\mu\mathrm{sec}$
  - $T_X(\mathrm{max}) = 2^N \cdot T_{CK} = 2^{10} \cdot 1\mu\mathrm{sec} = 1024\mu\mathrm{sec} \approx 1msec$
  - $f_X(\mathrm{min}) = 1/T_X(\mathrm{max}) \approx 1k\mathrm{Hz}$
- $f_X(\mathrm{max})$ when $f_{CK} = 10\mathrm{MHz}$
  - Maximum frequency means minimum period
  - The counter counts only when $T_X \geq T_{CK}$, otherwise the reset occurs too soon
  - $T_X \geq T_{CK} \Rightarrow f_X \leq f_{CK} \Rightarrow f_X(\mathrm{max}) = 10\mathrm{MHz}$

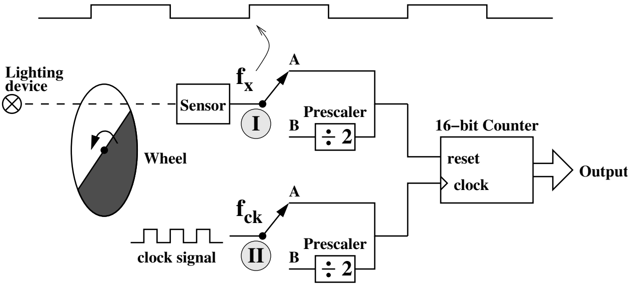
© 2019 Mihai SIMA

Timers and counters: second worked example

- Half of the disc is opaque, the other half is transparent
- f_x is the disc revolution frequency that is being measured
- $f_{ck} = 10$ MHz is the frequency of the clock signal
- The prescaler divides the signal frequency by 2
- Each of the swithes I and II can be either in position A or B
- The reset is active on the high level ('1')



Timers and counters: second worked example



1. What is the minimum revolution frequency than can be measured?
2. With what resolution is the minimum revolution frequency measured?
3. How does this resolution change if the 16-bit counter is replaced with a 32-bit counter?

Timers and counters: second worked example

- Minimum revolution frequency (maximum revolution period)
  - Switch I on A, switch II on A:
    - Period: $T_X(\max) = (2^{16} - 1) \cdot T_{CK} = 65535 \cdot 0.1\mu\text{sec} \approx 6.6m\text{sec}$
    - Frequency: $f_X(\min) = 1/T_X(\max) \approx 152\text{Hz}$
  - Switch I on A, switch II on B:
    - Period: $T_X(\max) = (2^{16} - 1) \cdot 2T_{CK} = 65535 \cdot 0.2\mu\text{sec} \approx 13.2m\text{sec}$
    - Frequency: $f_X(\min) = 1/T_X(\max) \approx 76\text{Hz}$
  - Switch I on B, switch II on A:
    - Period: $2T_X(\max) = (2^{16} - 1) \cdot T_{CK} = 65535 \cdot 0.1\mu\text{sec} \approx 6.6m\text{sec}$
    - Frequency: $f_X(\min) = 2/T_X(\max) \approx 304\text{Hz}$
  - Switch I on B, switch II on B:
    - Period: $2T_X(\max) = (2^{16} - 1) \cdot 2T_{CK} = 65535 \cdot 0.2\mu\text{sec} \approx 13.2m\text{sec}$
    - Frequency: $f_X(\min) = 2/T_X(\max) \approx 152\text{Hz}$

© 2019 Mihai SIMA

Timers and counters: second worked example

- Minimum revolution frequency – measurement resolution
- **Resolution**: the smallest detectable incremental change of input parameter that can be detected by the instrument.
- Minimum frequency: switch I on A, switch II on B:
  - Period: $T_X(\max) = (2^{16} - 1) \cdot 2T_{CK} = 65535 \cdot 0.2\mu\text{sec} = 13.1070m\text{sec}$
  - Frequency: $f_X(\min) = 1/T_X(\max) = 10^3/13.1070\text{Hz}$
  - Period: $T_X(\max - 1) = (2^{16} - 2) \cdot 2T_{CK} = 65534 \cdot 0.2\mu\text{sec} = 13.1068m\text{sec}$
  - Frequency: $f_X(\min) = 1/T_X(\max) = 10^3/13.1068\text{Hz}$
  - Resolution = $10^3/13.1068 - 10^3/13.1070 = 1.16 \cdot 2^{-3}\text{Hz}$
- HOMEWORK: How does this resolution change if the 16-bit counter is replaced with a 32-bit counter?

© 2019 Mihai SIMA

Timers and counters: second worked example

- **Accuracy**: the maximum difference that will exist between the actual value (which must be measured by a primary or good secondary standard) and the indicated value at the output of the instrument.
- Accuracy is given by the clock signal
  - The accuracy of the clock frequency gives us the accuracy of the measured frequency
  - Clock frequency has a drift with temperature
  - Clock frequency also experiences an aging fenomenon
  - Quartz-based oscillators are used to generate the clock signal
- HOMEWORK: given a clock frequency of $10M\mathrm{Hz} \pm 0.01\%$, determine the accuracy of the measured signal

 © 2019 Mihai SIMA

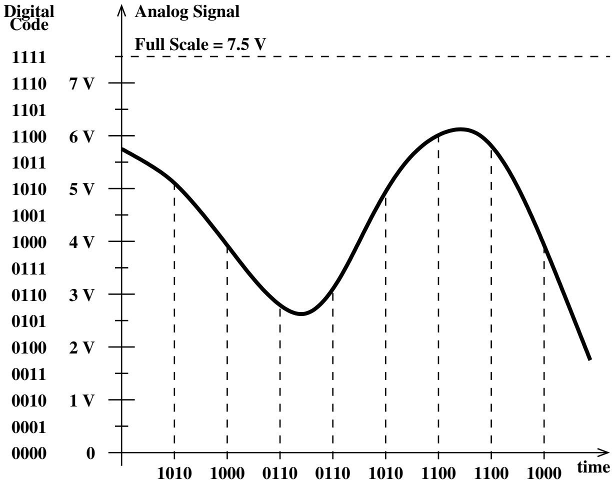
Digital-to-Analog and Analog-to-Digital Converters

- Digital-to-Analog Converter (DAC): binary values $\mapsto$ analog values
- Analog-to-Digital Converter (ADC): analog values $\mapsto$ binary values
- The analog signal will generally be a voltage or a current
- Either the input or output is digital $\longrightarrow$ the signal is quantized
- An N-bit DAC/ADC can have only 2^N possible analog outputs/inputs
- Without losing generality, consider the following **Full Scale** (FS) values:
  - the binary value is represented on a 10-bit word $\longrightarrow$ FS digital = $2^{10} = 1024$
  - the analog voltage ranges from 0 V to 10 V $\longrightarrow$ FS analog = 10 V
- **Resolution** is

$$\text{resolution} = \frac{\text{FS analog}}{\text{FS digital}} = \frac{10}{1024}\,\mathrm{V} = 9.54\,\mathrm{mV}$$

© 2019 Mihai SIMA

Example: 4-bit DAC / ADC



- Full-scale
  - Analog = 7.5 V
  - Digital = 1111_h
- Resolution
  - Analog = 0.5 V
  - Digital = 1_h

© 2019 Mihai SIMA

Resolution of DAC/ADC

- Consider the following:
  - The binary value is represented on a 10-bit word $\mapsto$ FS digital = $2^{10} = 1024$
  - The analog voltage ranges from 0 V to 10 V $\mapsto$ FS analog = 10 V
- **Resolution** may be expressed in several different ways:
  - **Millivolts** (mV):

$$\text{resolution} = \frac{\text{FS analog}}{\text{FS digital}} = \frac{10}{1024}\,\mathrm{V} = 9.54\,\mathrm{mV}$$

  - **Percentages of full-scale**

$$\text{resolution} = \frac{1}{\text{FS digital}} \times 100\% = \frac{1}{1024} \times 100\% = 0.098\%$$

  - **Parts-per-million (ppm) of full-scale**

$$\text{resolution} = \frac{1}{\text{FS digital}} \times 10^6\text{ppm} = \frac{1}{1024} \times 10^6\text{ppm} = 977\text{ppm}$$

© 2019 Mihai SIMA

Errors of DAC/ADC

- Intrinsic to the conversion itself:
  - **Quantization** error
- DC errors:
  - **Offset** error
  - **Full-scale** error
  - **Gain** error
  - **Integral nonlinearity** error
  - **Differential nonlinearity** error
- Drift errors = variations in a given error due a change in temperature
  - Offset-error drift
  - Gain-error drift

© 2019 Mihai SIMA

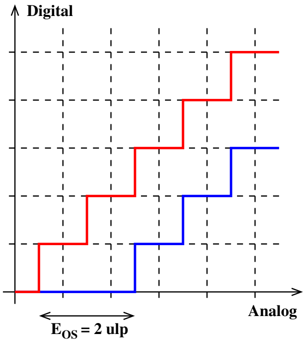
Offset error

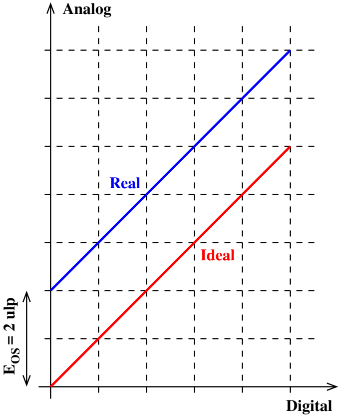
- Also called **zero-scale** error
- For unipolar converters
  - it indicates how well the actual transfer function matches the ideal transfer function at a single point (the zero point)
  - For an ideal data converter, the first transition occurs at 0.5LSB above zero
  - ADC: the zero-scale voltage is applied to the analog input and is increased until the first transition occurs
  - DAC: offset error is the analog output response to an input code of all zeros
- For bipolar converters
  - The error at zero-scale is at the midpoint of the bipolar transfer functions
- **Offset-error drift** is the variation in offset error due a change in ambient temperature, typically expressed in ppm/$^\circ$C.

© 2019 Mihai SIMA

Offset error – ADC and DAC

The difference between the real and ideal values at zero scale.

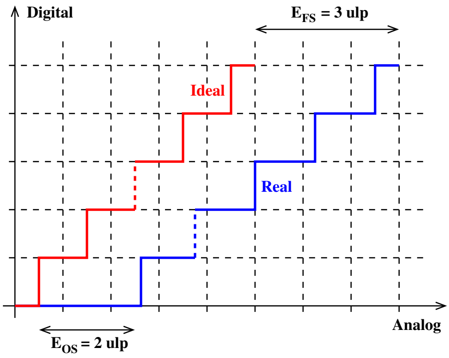


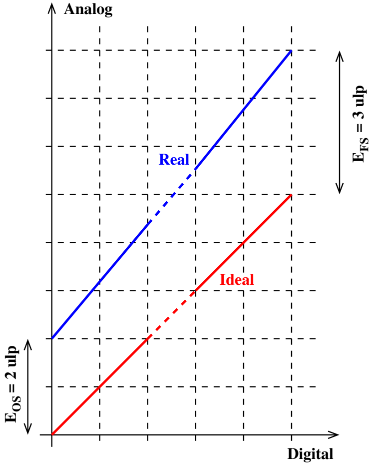


© 2019 Mihai SIMA

Full-scale error – ADC and DAC

The difference between the real and ideal values at full scale.





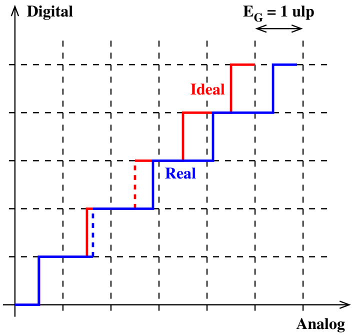
Gain error

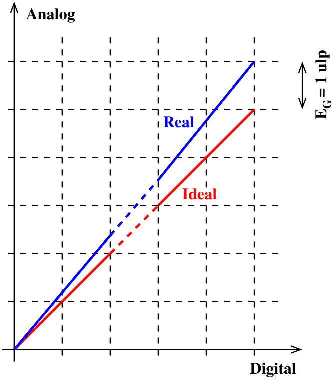
- Indicates how well the slope of a real transfer function matches the slope of the ideal transfer function
- Expressed in ulp (LSB) or as a percent of full-scale range (%FSR)
- Can be calibrated out in hardware or software
- Gain error is the full-scale error minus the offset error.
- Gain-error drift is the variation in gain error due to a change in ambient temperature, typically expressed in ppm/°C

© 2019 Mihai SIMA

Gain error

The difference between full-scale and offset errors.





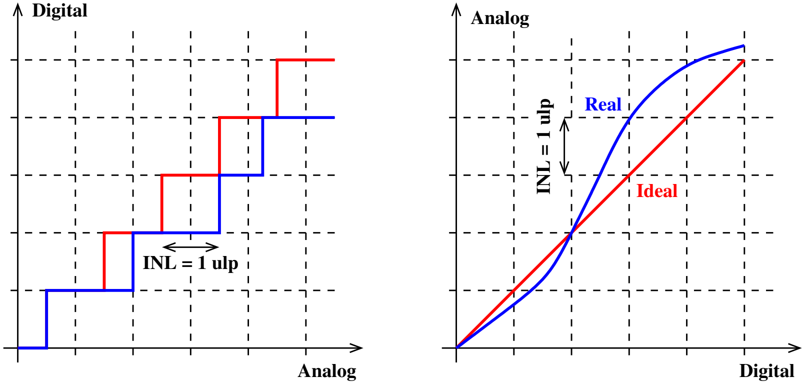
© 2019 Mihai SIMA

Integral Non-Linearity (INL) Error

- INL is the deviation of an actual transfer function from a straight line
- After nullifying offset and gain errors, the straight line is either:
  - a best-fit straight line, or
  - a line drawn between the end points of the transfer function
- INL is often called relative accuracy

© 2019 Mihai SIMA

Integral Non-Linearity (INL) Error



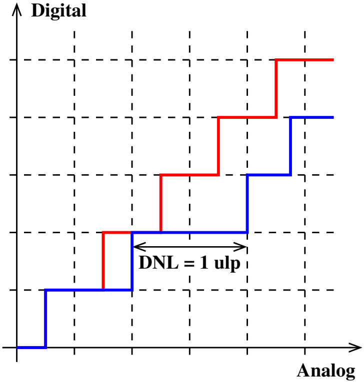
© 2019 Mihai SIMA

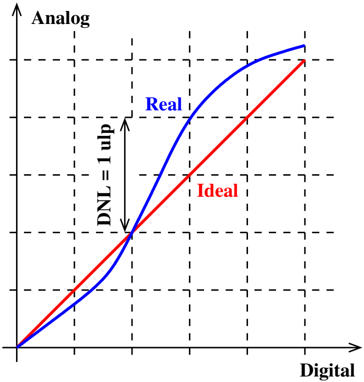
Differential Non-Linearity (DNL) Error

- For Analog-to-Digital Converters (ADC)
  - The analog-input levels that trigger any two successive output codes should differ by one ulp (DNL = 0)
  - Any deviation from one LSB is defined as DNL
- For Digital-to-Analog Converters (DAC)
  - DNL error is the difference between the ideal and the measured output responses for successive DAC codes
  - An ideal DAC response would have analog output values exactly one code (ulp) apart (DNL = 0)
- A DNL of less than or equal to 1 ulp guarantees monotonicity

© 2019 Mihai SIMA

Differential Non-Linearity (DNL) Error





© 2019 Mihai SIMA

Example: Analog Devices AD7942

- AD7942 is an Analog-to-Digital Converter
- 14 bit resolution with no missing codes
- Integral Non-Linearity (INL) error: ± 1 LSB
  - The deviation of each code from a line drawn from zero to full scale
- Differential Non-Linearity (DNL) error: ± 0.7 LSB
  - Ideally, code transitions are 1 LSB apart. DNL is the maximum deviation from this ideal value

© 2019 Mihai SIMA

How to build an ADC given an DAC

- DAC + feedback and a sequential algorithm = ADC
- A way to compute the inverse of a function is to build a system with feedback, make the gain of the feedforward path as large as possible and let the feedback path have its gain equal to the function to be inverted.

$$a = \frac{A}{1 + Af}$$

  where A is the feedforward gain, f is the feedback gain, and a is the gain of the global system. By making A very large, $1 + Af \approx Af$, and

$$a = \frac{A}{1 + Af} \approx \frac{A}{Af} = \frac{1}{f}$$

  The gain of the system with feedback is the inverse of the initial function.
- ADC = Comparator as feedforward path and DAC as feedback path

© 2019 Mihai SIMA

How to build an ADC given an DAC (cont'd)

- Use a Voltage-Controlled-Oscillator (VCO) and a counter
  - What accuracy would such system have?
- Parallel ADC
  - Conversion in one step by means of a resistor or capacitor divider
  - Beyond the scope of the course
- Both DAC and ADC converters need a reference signal, which has influence on the conversion accuracy

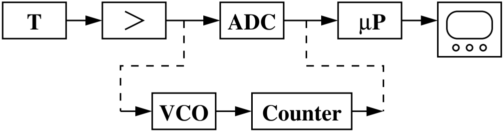
© 2019 Mihai SIMA

DAC and ADC on-line documentation!

- **National Semiconductor**:
  - General issues: **www.national.com**
  - Online course:
    `https://training.ti.com/choosing-best-adc-architect`
- **Analog Devices**: **www.analog.com**
- **Texas Instruments**: **www.ti.com**
- **Maxim/Dallas**: **www.maxim-ic.com**

© 2019 Mihai SIMA

Measurement systems – common organization



- **Transducer**: a device that converts one type of energy to another.
  - Thermoelectric:
    - thermocouple – converts temperature into an electrical signal
    - thermistor – converts temperature into an electrical signal
  - Electroacoustic:
    - microphone – converts changes in air pressure into an electrical signal
    - piezoelectric crystal – converts pressure changes into electrical signal
    - hydrophone – converts changes in water pressure into an electrical signal
  - Photoelectric:
    - photodiode – converts changing light levels into an electrical signal
    - phototransistor – converts changing light levels into an electrical signal

Measurement systems – common organization

- **Goal**: Measure the physical phenomenon and display the quantity.
- The measurement process is not perfect, it is affected by errors.
- **Error sources**:
  - Transducer is typically non-linear
  - Instrumentation Amplifier (IA) has gain and offset errors
  - Analog-to-Digital Converter (ADC) has gain, offset, and non-linearity errors
  - MicroProcessor performs calculations – there are round-off errors
- These errors can be compensated out simultaneously because
  - Gain and offset errors of IA and ADC add up
  - Linearity errors of Transducer and ADC also add up
- Some of these errors can be corrected in software!
  - Gain and offset errors can be corrected using simple arithmetic
  - Non-linearity can be corrected by means of a look-up table

© 2019 Mihai SIMA

Measurement systems – common organization

- Two major questions can be posed:
  - What is the **resolution** of the displayed result?
  - What is the **accuracy** of the displayed result?
- **Resolution** is given by the number of bits of the ADC
  - 8-bit ADC: 1 step out of 256 means 0.4% quantization error
  - 10-bit ADC: 1 step out of 1024 means 0.1% quantization error
  - 10-bit ADC with 0-5V range means resolution of 5mV
- **Accuracy** is much more difficult to determine
  - **Gain and offset errors** are typically compensated out at the analog circuitry level (but software solutions possible)
  - **Non-linearities** are much more difficult to null out
- **Non-linearities**: integral and differential
  - **Integral non-linearity** can be nulled out in software by a look-up table
  - **Differential non-linearity**: we typically live with it

© 2019 Mihai SIMA

Measurement systems – worked example

- Assume we want to measure temperature from 0 to 100 degrees
- Transducer: **thermoresistor** having the following characteristics:

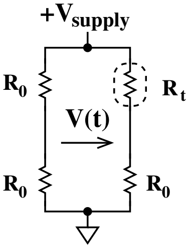
$$R_t = R_0[1 + k(t - t_0)] \quad \Longleftrightarrow \quad \frac{\Delta R}{R_0} = k\Delta t$$

  where R_t is the resistence at temperature t, R_0 is the resistence at temperature t_0, and k is the temperature coefficient of resistance
- Numerical example: $R_0 = 100\,\Omega$, $t_0 = 0^\circ C$, $k = 0.01\,^\circ C^{-1}$
  - At $t = 1^\circ C$: $R_t = 100 + 100 \cdot 0.01 \cdot 1 = 101\,\Omega$
  - At $t = 10^\circ C$: $R_t = 100 + 100 \cdot 0.01 \cdot 10 = 110\,\Omega$
  - At $t = 100^\circ C$: $R_t = 100 + 100 \cdot 0.01 \cdot 100 = 200\,\Omega$

© 2019 Mihai SIMA

Measurement systems – worked example

- Typically the thermoresistor is part of a resistance bridge (three resistors of 100 Ω and the other is the thermoresistor itself)



- The voltage across the bridge depends non-linearly on temperature:

$$V(t) = \left[\frac{R_0}{R_0 + R_0} - \frac{R_0}{R_0 + R_t}\right] V_{\text{supply}} = \frac{kt}{2(2 + kt)} V_{\text{supply}}$$

© 2019 Mihai SIMA

Measurement systems – worked example

- Including the other errors:

$$V_{\text{ADC}} = \left[\frac{kt}{2(2+kt)}V_{\text{supply}} + V_{\text{offset}}\right] \cdot g_{\text{IA}} \cdot g_{\text{error}} \cdot f_{\text{non-linear ADC}}$$

  where:

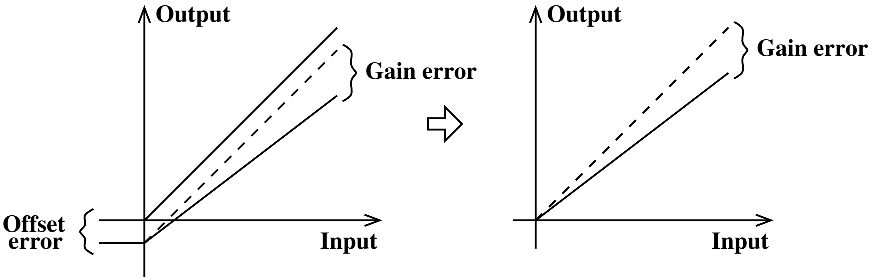
  - V_{ADC} is the signal to be sampled and quantized
  - V_{offset} is the offset (voltage)
  - g_{IA} is the instrumentation amplifier gain
  - g_{error} is the (multiplicative) gain error
  - $f_{\text{non-linear ADC}}$ models the ADC non-linearity

- Ideally, $V_{\text{offset}} = 0$, $g_{\text{error}} = 1$, $f_{\text{non-linear ADC}} = 1$.

© 2019 Mihai SIMA

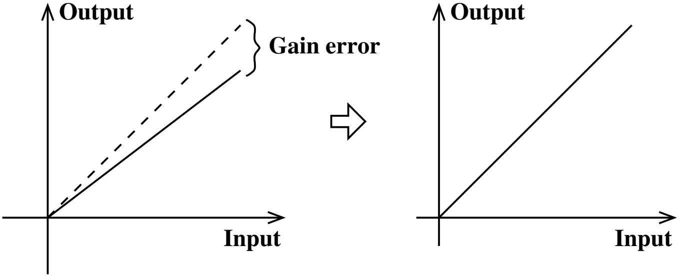
How to compensate out the errors

- **Offset**: make sure $V(t) = 0$ and adjust the output to zero



How to compensate out the errors

- **Gain**: make sure $V(t) = \max$ and adjust the output to full scale



How to compensate out the errors

- With offset and gain errors compensated:

$$V_{\text{ADC}} = \frac{kt}{2(2+kt)} V_{\text{supply}} \cdot g_{\text{IA}} \cdot f_{\text{non-linear ADC}}$$

- The transducer non-linearity can be compensated out in software:
  - Look-up table that implements the inverse function
  - Polynomial approximation of the inverse function
- The inverse function:

$$t = \frac{4y}{k(1-2y)}$$

- If we can measure the ADC integral non-linearity, we can compensate it and the transducer non-linearity simultaneously, as a whole

© 2019 Mihai SIMA

Sensitivity, resolution, and accuracy – worked example

- Assume a perfectly linear 10-bit ADC and the mentioned thermoresistor
- Determine the sensitivity and resolution for measuring the temperature
  - around $t = 0^\circ C$
  - around $t = 50^\circ C$
  - around $t = 100^\circ C$
- Linearize the transfer function in software
- Assume a real ADC and determine the sensitivity and resolution for measuring the temperature
- Same questions for accuracy

© 2019 Mihai SIMA

Measurement systems – worked example

- Full-scale is at $100^\circ C$, which means $V = 1/6 = 0.167$ Volts
- The sensitivity of the (normalized) voltage across the bridge, V, with respect to temperature, t

$$\frac{\mathrm{d}V}{\mathrm{d}t} = \frac{k}{2(2+kt)^2}$$

- Sensitivity depends on temperature:
  - At $t = 0^\circ C$, the sensitivity is $k/8\,C^{-1}$. Therefore, there is a change of $0.01/8 = 0.00125$ Volts per degree.
  - At $t = 50^\circ C$: sensitivity = $k/12.5\,C^{-1}$. Therefore, there is a change of $0.01/12.5 = 0.000833$ Volts per degree.
  - At $t = 100^\circ C$: sensitivity = $k/18\,C^{-1}$. Therefore, there is a change of $0.01/18 = 0.000555$ Volts per degree.
- With a 10-bit ADC, the quantization step is $0.167/2^{10} = 0.000163$ Volts/bit

© 2019 Mihai SIMA

Measurement systems – worked example

- The voltage across the bridge relates to the ambient temperature
- At temperature $t = 0^\circ C$
  - $0.00125/0.000163 = 7.6$ bit/C
  - $0.000163/0.00125 = 0.13$ C/bit
- At temperature $t = 50^\circ C$
  - $0.000833/0.000163 = 5.1$ bit/C
  - $0.000163/0.000833 = 0.19$ C/bit
- At temperature $t = 100^\circ C$
  - $0.000555/0.000163 = 3.4$ bit/C
  - $0.000163/0.000555 = 0.29$ C/bit

© 2019 Mihai SIMA

Measurement systems – worked example

- **Linearize the transfer function in software** means we implement the inverse function in software
- The inverse function:

$$y = \frac{4x}{k(1-2x)}$$

- The composite function is the identity function
- How to implement the inverse function
  - Look-up table (1024 entries for a 10-bit converter)
  - Piece-wise linear approximation
  - Higher-order polynomials (not the common case)
- Some precision is always lost when doing software linearization
  - This is due to the inherent compression
  - Expansion does not affect precision

© 2019 Mihai SIMA

Measurement systems – worked example

- What is the loss in precision due to software linearization?
- The point where the slope of the inverse function curve equals the ideal straight line is given by:

$$\left(\frac{4x}{k(1-2x)}\right)' = \frac{100}{\frac{1}{6}} = 600 \Longrightarrow x = \frac{0.55}{6}$$

- The value of the inverse function at $x = 0.55/6$ is $y = 45$
- The straight line value at $x = 0.55/6$ is $y = 55$
- The difference between the inverse function and the straight line at $x = 0.55/6$ is $\Delta = 10$, which means 10% of the full-scale value

© 2019 Mihai SIMA

Measurement systems – worked example

- We want to represent a range of $100 - 45 = 55$ degrees using only $100 - 55 = 45$ degrees
- With a 10-bit converter we want to compress $1024 \cdot 55/100 = 563$ values into $1024 \cdot 45/100 = 461$ values
- If we compress, say, 512 values into 256, we lose one bit of significance ($\log_2(512/256) = 1$)
- In our case we lose $\log_2(563/461) = 0.29$ bits of significance
  - We write this as 0.29 LSB
- The non-linearity errors are added to this imperfection
  - Example: integral non-linearity (INL) is ± 1 LSB
  - Worst case scenario: total error is INL + SW-linearization error

© 2019 Mihai SIMA

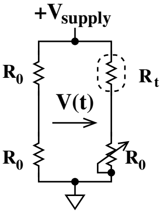
Resolution and accuracy – worked example

- Consider AD9200: an ADC from Analog Devices
  Over the entire ambient temperature range:
  - DNL: ± 0.5 LSB typ., ± 1 LSB max.
  - INL: ± 0.75 LSB typ., ± 2 LSB max.
  - Offset error: 0.4% typ., 1.2% max. of Full-Scale
  - Gain error: 1.4% typ., 3.5% max. of Full-Scale
- First observation: offset and gain error are very large!
- Our first concern is to cancel out these two errors.

© 2019 Mihai SIMA

Resolution and accuracy – worked example

- Although the offset and gain errors can be compensated out by using potentiometers, these errors can be eliminated only at a single ambient temperature.



- The offset and gain errors are so large, than dynamic compensation in software is needed if a good accuracy is to be achieved.
- Software compensation of offset error requires a simple addition/subtraction
- Software compensation of gain error requires a simple multiplication
- Note that a reference signal (e.g., a voltage) is needed on board

© 2019 Mihai SIMA

Resolution and accuracy – worked example

- Resolution (which is given by quantization and linearization) = 0.5 LSB + 0.29 LSB = 0.79 LSB
- Linearization also has an impact on accuracy.
- Let's assume that with dynamic compensation in software of offset and gain error of both the resistance bridge and ADC, the error is reduced to 2% of full-scale (2 LSB)
- Worst case scenario:
  Total error = 2 LSB (INL) + 1 LSB (DNL) + 0.5 LSB (quantization) + 0.29 LSB (linearization) + 2 LSB (offset and gain) $\approx$ 6 LSB

© 2019 Mihai SIMA

Resolution and accuracy – worked example

- The probability of having really bad luck, so that all the errors add up constructively is small.
- The probability of having really good luck, so that all the errors add up destructively is also small.
- As a rule of thumb, an error_1 = 3 units and an error_2 = 4 units give a total error of $\sqrt{3^2 + 4^2} = 5$ units.
- The final system needs to be tested, and the units that fail the test are simply eliminated.

© 2019 Mihai SIMA

VCO+counter based analog-to-digital conversion

- **Voltage-Controlled Oscillator** (VCO)
  - Input: the voltage to be measured
  - Output: a rectangular signal having its frequency related to the input voltage (linear dependence is common)
- Current-Controlled Oscillator includes a high-precision resistor (to convert the current into a voltage according to the Ohm law) plus a VCO
- We can define non-linearity errors for the VCO, too.
- Analog-to-digital conversion is achieved by VCO + counter
- The conversion time is large, but the conversion technique provides inherent low-pass filtering (thus the noise is automatically removed)

© 2019 Mihai SIMA

Analog-to-digital and digital-to-analog conversion

- **Precision** is the most important issue
- **Conversion time** can also be an issue
  - Low conversion time for ADC and DAC devices
  - Large conversion time for VCO+counter
- VCO + counter based conversion provides inherent low-pass filtering
- Quantization is a lossy process
- Software linearization is also a lossy process

© 2019 Mihai SIMA

UARTs – serial transmission

- UART: Universal Asynchronous Receiver Transmitter
  - Data is written into the UART port and transmitted serially
  - Data is received serially and read into host processor
- Five to eight bits of data (usually)
- Parity bit for error checking
- Two additional bits: start and stop
- Transmission rate:
  - **Bit rate** = number of bits per second
  - **Baud rate** = number of signal changes per second

© 2019 Mihai SIMA

Serial transmission

- Classic example: for '0' transmit 0V, for '1' transmit 5V
  - The bit rate = the baud rate
- Second example: for '00' transmit -5V, '01' transmit -2.5V, '10' transmit +2.5V, '11' transmit +5V
  - The bit rate = twice the baud rate
- Third example: for '00' transmit $A\sin(\omega t)$, '01' transmit $A\sin(\omega t+\pi/2)$, '10' transmit $A\sin(\omega t+\pi)$, '11' transmit $A\sin(\omega t+3\pi/2)$
  - The bit rate = twice the baud rate
- It is important to know the difference between bit rate and baud rate, since a given bit rate requires a minimum processing speed, which in turn translates to computing power and real-time response requirements

 © 2019 Mihai SIMA

Questions, feedback

?

Notes I

 © 2019 Mihai SIMA