Recap

Output compare interrupts
Metrowerks Codewarrior

Overview

Analog to Digital Convertor
Transducer: mechanical, electrical
Using output compare interrupts to establish sampling

Range(volts) = Precision(alternatives) • Resolution(volts)

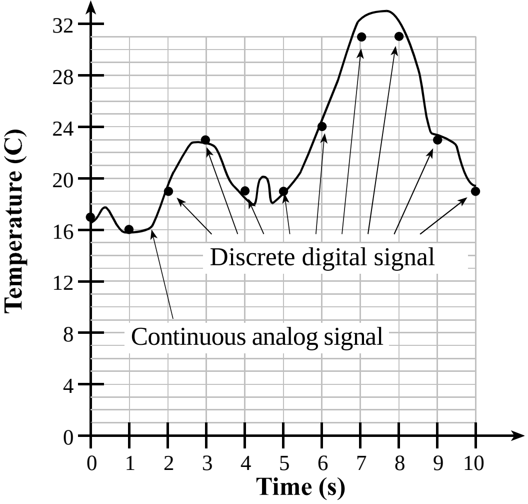


Figure 11.1. An analog signal is represented in the digital domain as discrete samples.

10-bit Analog to digital Converter

Analog input $0 \le V_{in} \le +5$
Digital output $0 \le$ `ATD0DR0` ≤ 1023
Digital Output is about $1024*V_{in}/5$
or Digital Output is about $1023*V_{in}/5$

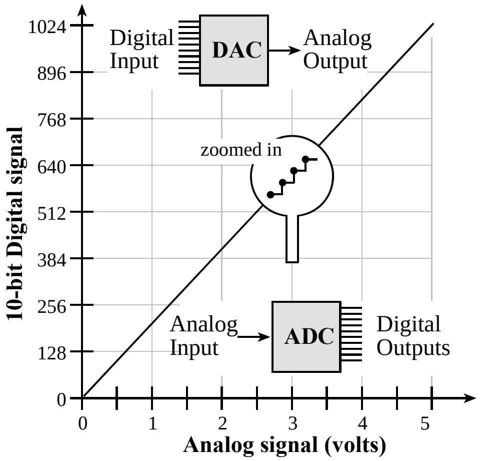
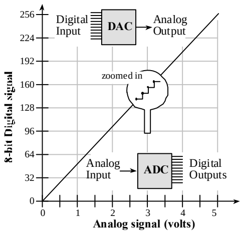


Figure 11.2. Input/output functions of a DAC and an ADC.



8-bit Analog to digital Converter

Analog input $0 \le V_{in} \le +5$

Digital output $0 \le$ `ATD0DR0` ≤ 255

Digital Output is about $256*V_{in}/5$

or Digital Output is about $255*V_{in}/5$

Find the 16 analog inputs on the 9S12DG128

`ATD0CTL2=$80;` set bit 7 to enable ADC
`ATD0CTL3=$08;` sequence length=1
`ATD0CTL4` bit 7=0 for 10-bit, =1 for 8-bit ADC
Bits 4-0 internal clock (how fast the ADC runs)
Let `m` be the bottom 5 bits of `ATD0CTL4`
Internal clock is ½E/(`m`+1)

Internal clock can be between 500kHz and 2 MHz

At 24 MHz
2 MHzmake `ATD0CTL4=$05 ; m=5 (7us)`
1 MHzmake `ATD0CTL4=$0B ; m=11(14us)`

`ATD0CTL5` write channel number to start ADC

- channel number $80 to $87

`ATD0STAT` bit 7 SCF

- cleared by write to `ATD0CTL5`
- set when ADC finished

`ATD0DR0` first 10-bit ADC result

- precision 10-bit, 1024 alternatives
- range 0 to +5V
- resolution (5-0)/1024 = 5 mV

Analog Input (V)	Digital Output		
0.000	%0000000000	$000	0
0.005	%0000000001	$001	1
2.500	%1000000000	$200	512
3.750	%1100000000	$300	768
5.000	%1111111111	$3FF	1023

Table 11.6. **Straight binary** format used.

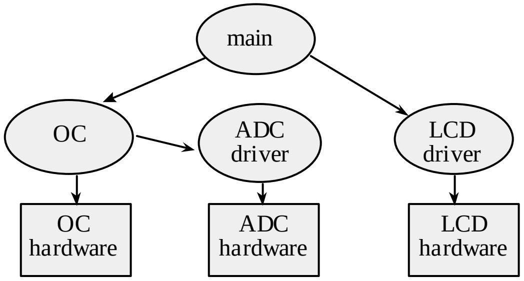
`ATD0DR0` first 8-bit ADC result

- precision 8-bit, 256 alternatives
- range 0 to +5V
- resolution (5-0)/256 = 19.5 mV

Analog Input (V)	Digital Output		
0.00	%00000000	$00	0
0.02	%00000001	$01	1
2.50	%10000000	$80	128
3.75	%11000000	$C0	192
5.00	%11111111	$FF	255

Table 11.6. **Straight binary** format used.

Lab 8 is real-time data acquisition system

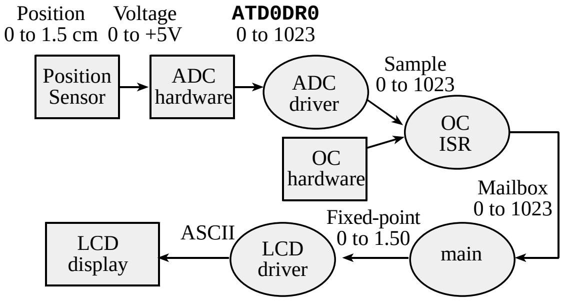


Lab 8: Design a position meter

- **Hardware**
- **Software**

- **Hardware**
  - **Transducer**
  - **Electronics**
  - **ADC**
- **Software**
  - **ADC device driver**
  - **Timer routines**
    - **Output compare interrupts**
  - **LCD driver**
  - **Measurement system**
    - **How fast to update**
    - **Fixed-point number system**
    - **Algorithm to convert ADC into position**

A data flow graph is one of the first design steps



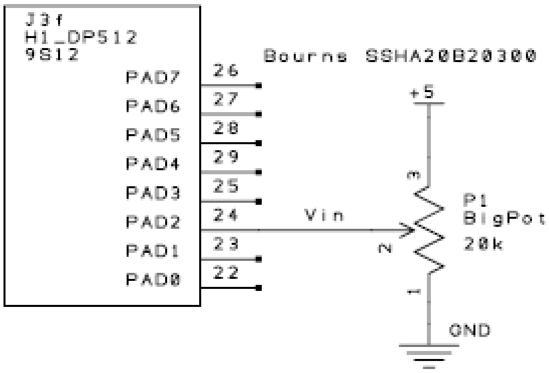
A transducer converts position into resistance

Solder wires to pins 1,2,3
Glue transducer to a solid
Position metric ruler (for calibration and testing)
Create a hair-line cursor

Question: what is $\mathbf{R_{12}} + \mathbf{R_{23}}$ at all times?

Question: what are $\mathbf{R_{12}}$ and $\mathbf{R_{23}}$ when cursor is at 1cm?

An electrical circuit converts resistance in to a voltage



Question: what is $\mathbf{V_{in}}$ at 1cm?

Question: what is **`ATD0DR0`** at 1cm?

Question: what do you want to display on the **LCD** at 1cm?

1) run **Lab 8 starter file, first main program, write ADC**

2) see ADC device driver

3) show **`ADC_Init`**

Turns it on
Sets it to 10-bit mode

4) show **`ADC_In`**

write channel number to **`ATD0CTL5`**
wait for SCF flag in **`ATD0STAT`**
read 10-bit result from **`ATD0DR0`**

5) show **`Data`**

complexity abstraction (what we want to do)

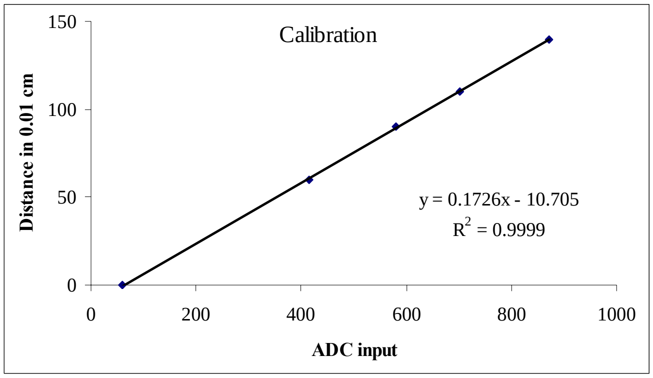
divide a complex problem into simple subcomponents

functional abstraction (how we do it)

divide a problem into modules
grouped by function
draw a data flow graph

Run solution to Lab 8, show how to prove real time DAS

start ADC at a regular rate (every 100ms)
measure the jitter
$\mathbf{100ms-\delta t < t_n - t_{n-1} < 100ms+\delta t}$



Lab 8
Sample ADC every 0.1s
Map (Data is 0 to 255) into (Position is 0000 to +1600)
Option A) use a linear function
 Position = (Constant1*Sample)/Constant2 + Constant3
 Position = (65536*Sample)/Constant4 + Constant3
Option B) Use a paired calibration table (S[i],P[i])
 S[i] are ADC samples measured at corresponding positions P[i]
 Given sample, find i such that S[i]<=sample<S[i+1]
 Use linear interpolation (look up **etbl** in TExaS help)
 position = P[i]+((sample-S[i])*(P[i+1]-P[i]))/(S[i+1]-S[i])
Option C) Create a 256-entry calibration table (P[ATD0DR0])
Fixed-Point output (using Lab 6 code)
 123 is displayed a “1.23 cm”

The bottom line

ADC converts analog to digital
Controlling time is important