TI-RSLK MAX

Texas Instruments Robotics System Learning Kit

Texas Instruments

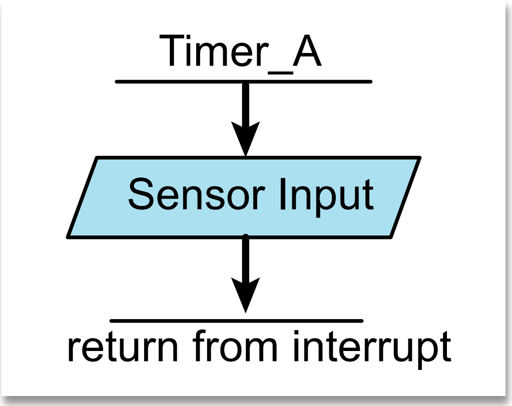
Module 15

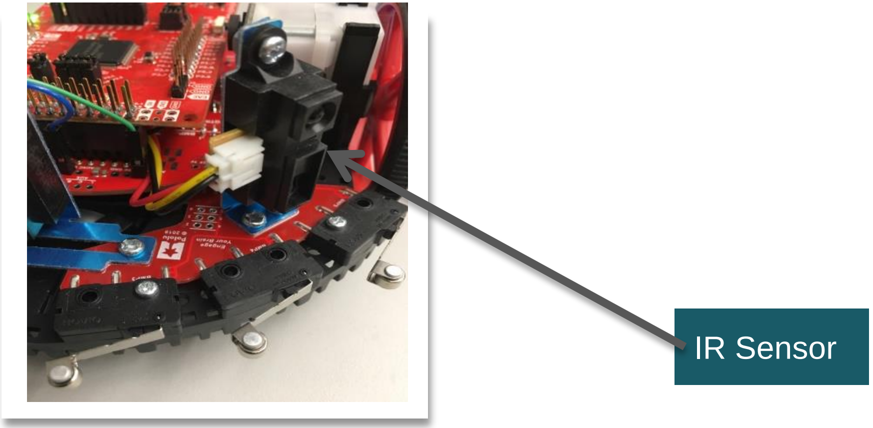
Lecture: Data Acquisition Systems - Theory

Data Acquisition Systems

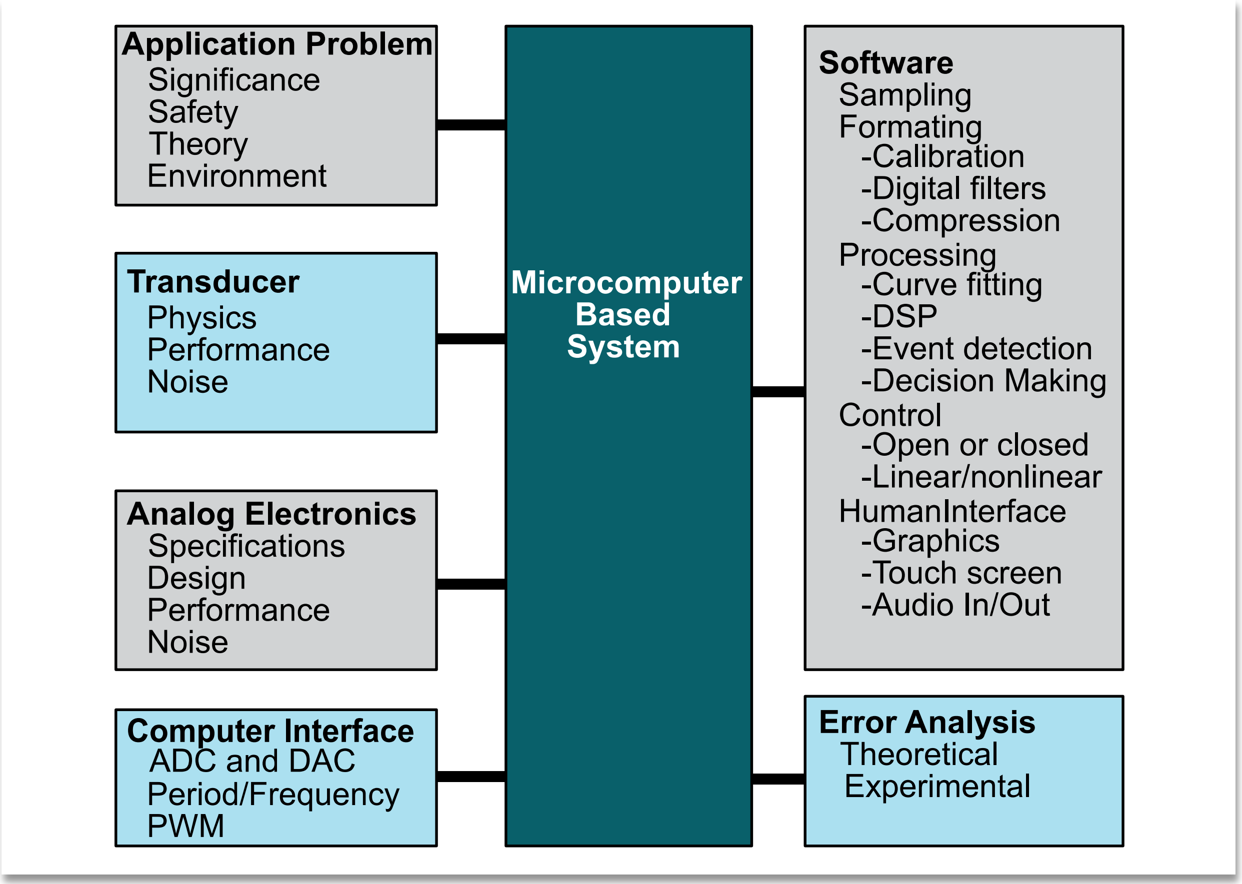
You will learn in this module

- Signals & Sampling
  - ADC, DAC
  - Range, resolution, precision
  - Successive approximation
- MSP432
  - Software driver
  - Spectrum Analyzer
  - Central Limit Theorem

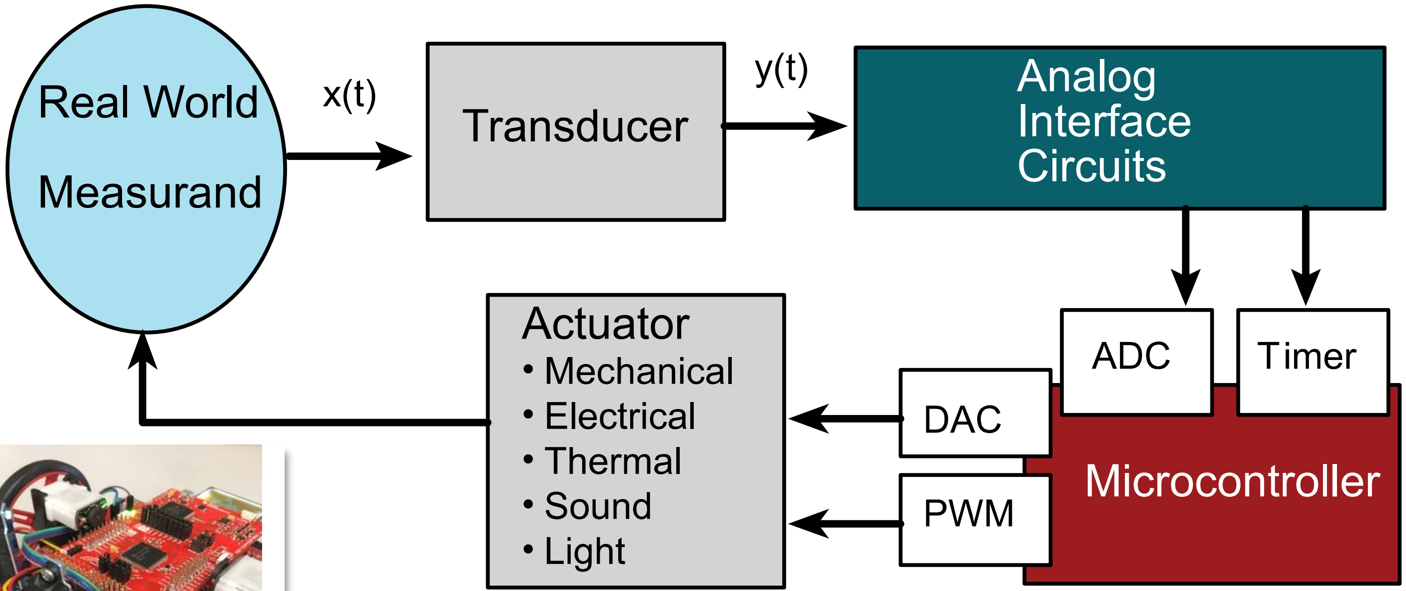


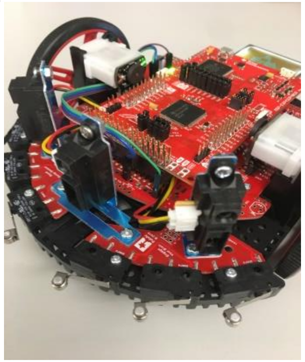


Data Acquisition Systems



A Control System includes a Data Acquisition System





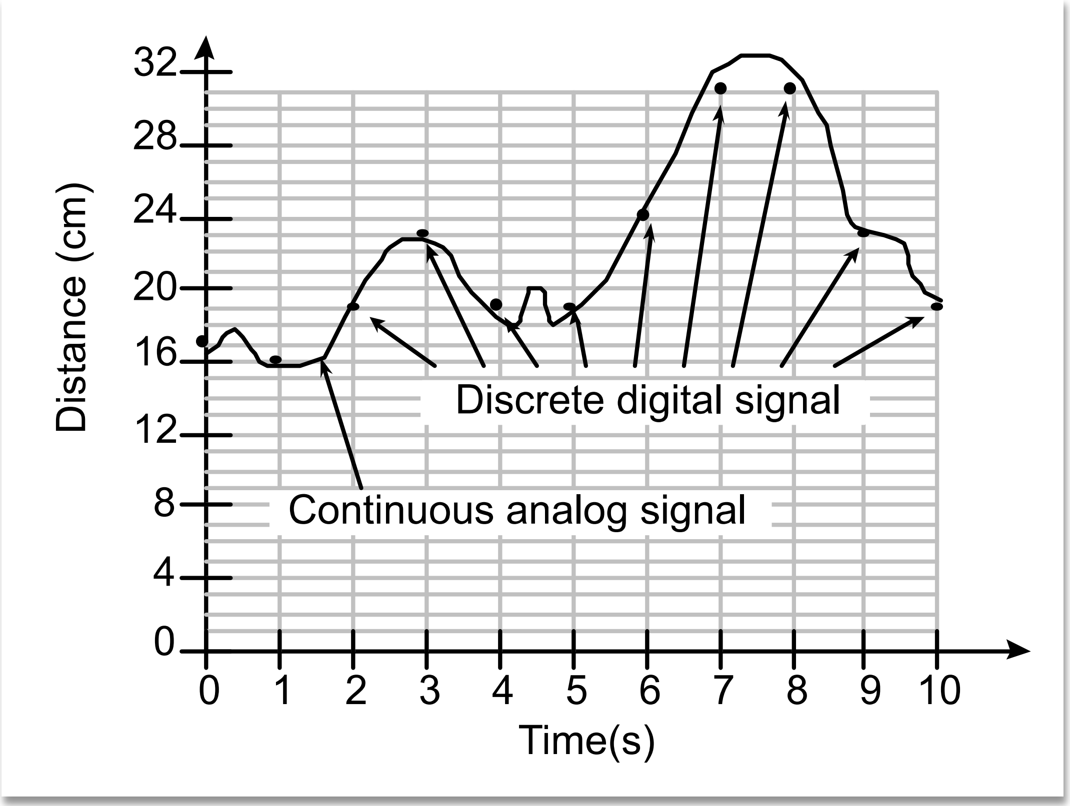
Sampling: conversion from analog to digital

Amplitude

- Range
- Resolution
- Precision

Time domain

- Sampling rate, f_s
  - 0 to ½ f_s
- Number of samples
  - Buffer size N
- Frequency resolution
  - f_s/N



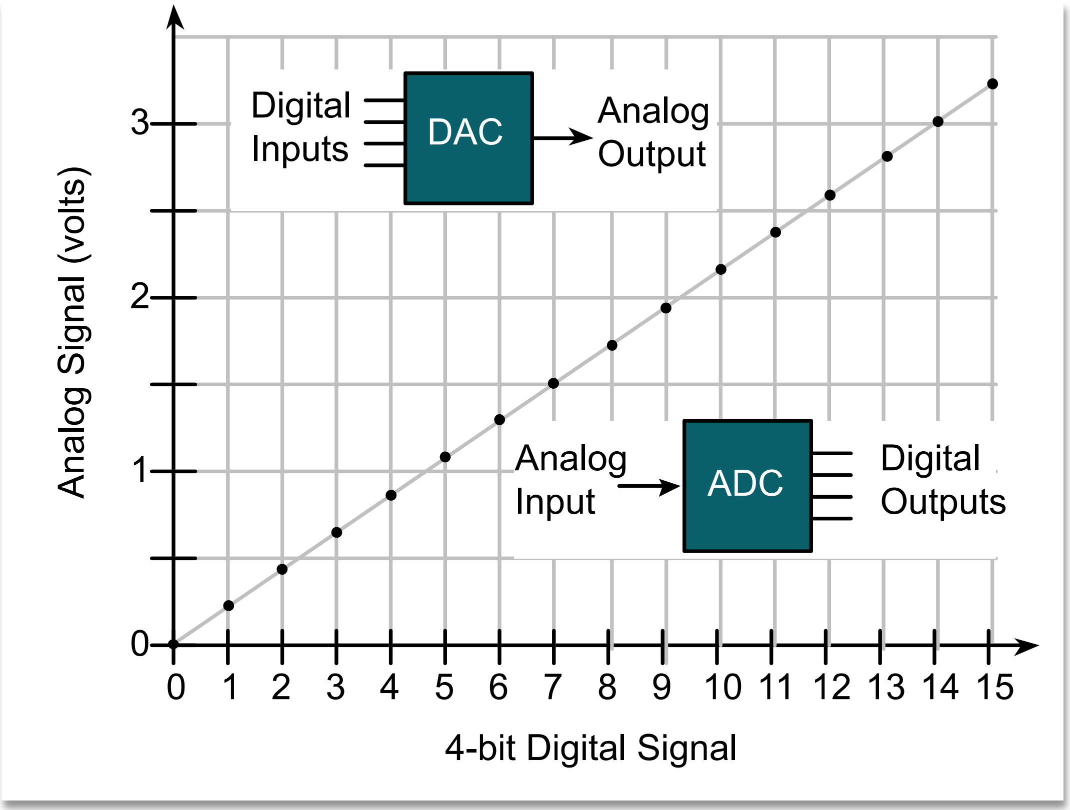
DAC versus ADC

DAC

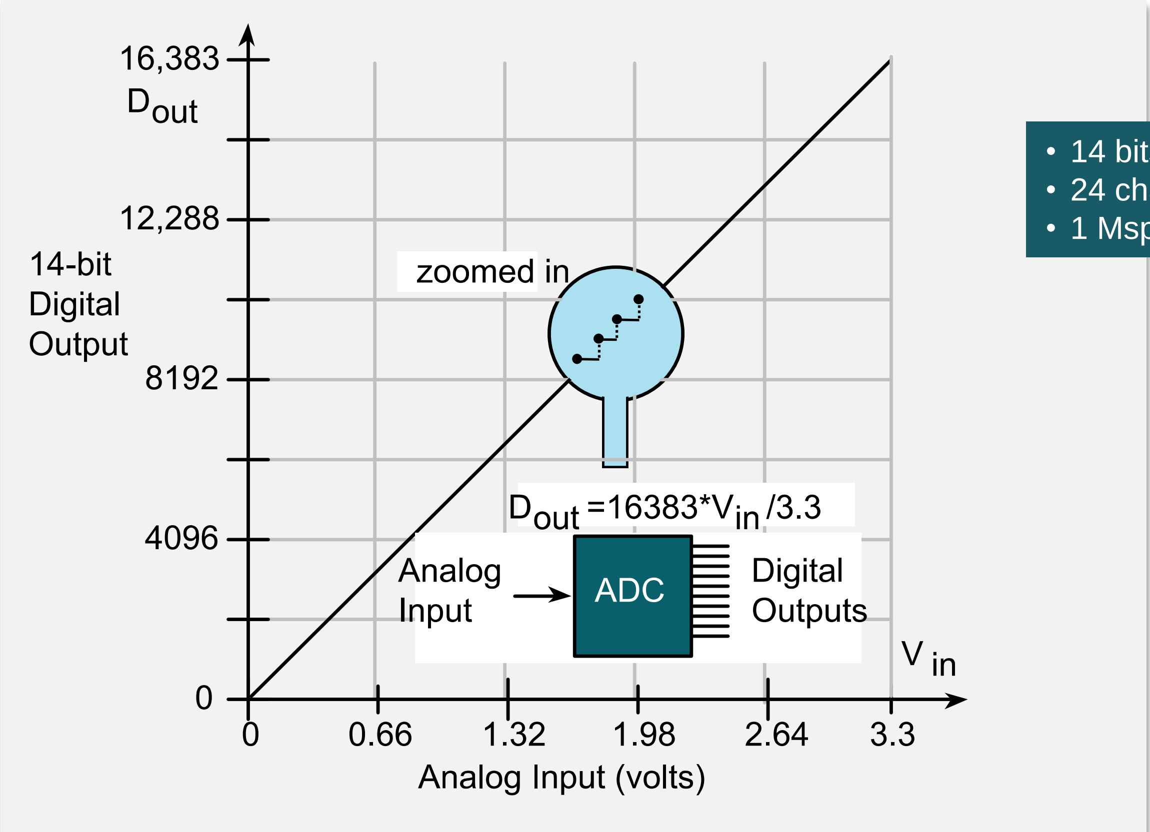
- Digital to Analog
- uC output
- Signal generation

ADC

- Analog to Digital
- uC input
- Measurements



MSP432 ADC14



- 14 bits
- 24 channels
- 1 Msps

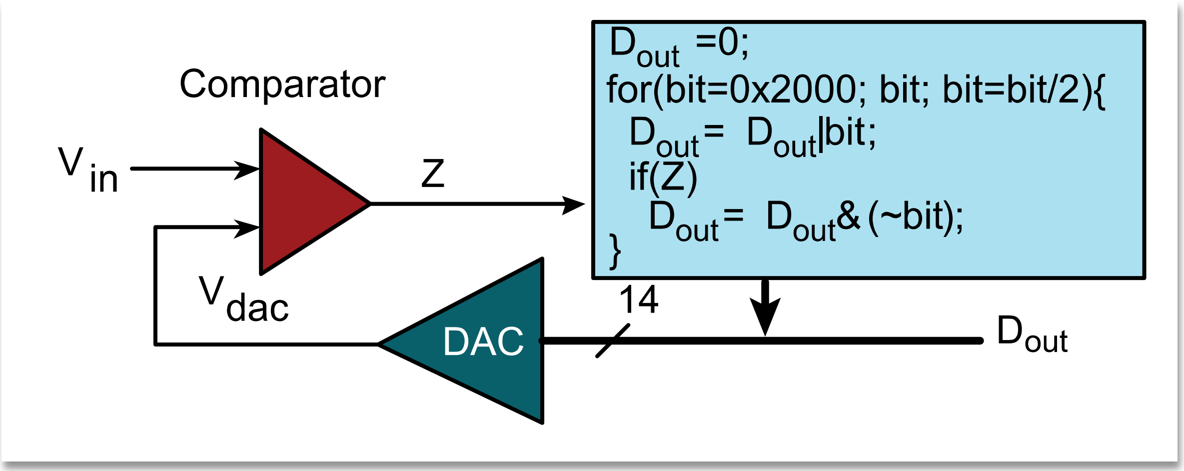
Successive Approximation

8-bit Successive Approximation Game

- I pick a number from 0 to 255
- You can guess
- I will respond high or low (same)
- How many guesses will it take you?

What is your first guess?

Successive Approximation – How it works



8-bit Successive Approximation Game

- You asked, “what is bit 7?”
- You asked, “what is bit 6?”

…

- You asked, “what is bit 0?”

- Uses a DAC
- Uses a comparator
- 1 bit/clock

Good information https://e2e.ti.com/blogs_/b/msp430blog/archive/2016/05/10/how-to-leverage-the-flexibility-of-an-integrated-adc-in-an-mcu-for-your-design-to-outshine-your-competitor-part-1

ADC14 Software Initialization

ADC14->CTL0

31-30	29-27	26	25	24-22	21-19	18-17	16
PDIV	SHSx	SHP	ISSH	DIVx	SSELx	CONSx	BUSY
15-12	11-8	7	6-5	4	3-2	1	0
SHT1x	SHT0x	MSC		ON		ENC	SC

- Clock (speed/power)
- Sample and hold (noise)
- Sequence or single channel
- Reference (range)
- Enable
- Start sample

ADC14 Software Initialization

ADC14->CTL1

31-28	27 – 24	22	21	20-16
	CH3MAP – CH0MAP	BATmap		CStartAdr

15-6	5 – 4	3	2	1-0
	RES	DF	REFBURST	PWRMD

- Address
- Resolution
- Format
- Power

ADC14->IFGR0

31		5	4	3	2	1	0
IFG31	…	IFG5	IFG4	IFG3	IFG2	IFG1	IFG0

- Conversion complete

ADC14 Software Initialization

ADC->MCTL[n]

31-16	15		14	13	12	11-8
	WINCTH		WINC	DIF		VRSEL
7	6	5	4 – 0			
EOS			ADC14INCHx			

- Comparator
- Differential/single
- Reference
- Channel

ADC14 Software Conversion

1. Wait for BUSY to be zero
2. Start conversion
3. Wait for completion
4. Read result

```
uint32_t ADC_In6(void){
  while(ADC14->CTL0&0x00010000){};
  ADC14->CTL0 |= 0x00000001;
  while((ADC14->IFGR0&0x01) == 0){};
  return ADC14->MEM[0];
}
```

Periodic Interrupt and Mailbox

1. Sample ADC
2. Run digital filter
3. Save in global
4. Set semaphore

```
void SysTick_Handler(void){
  uint32_t RawADC;
  P1OUT ^= 0x01;
  P1OUT ^= 0x01;
  RawADC = ADC_In6();
  ADCvalue = LPF_Calc(RawADC);
  ADCflag = 1;    // semaphore
  P1OUT ^= 0x01;
}
```

9us

Summary

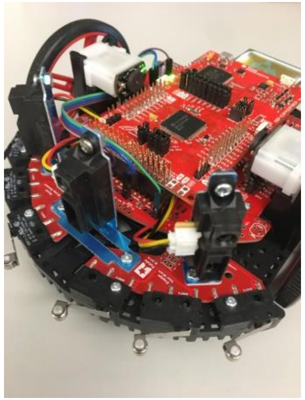
Analog to Digital Converter

- Successive Approximation
- Range
- Resolution
- Precision

Software

- Initialization
- Mailbox

$$\frac{100}{n}\sum_{i=0}^{n}\frac{\left|x_{ti}-x_{mi}\right|}{x_{t\max}}$$



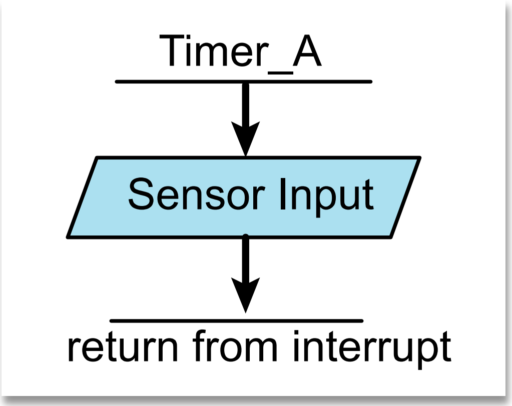
Module 15

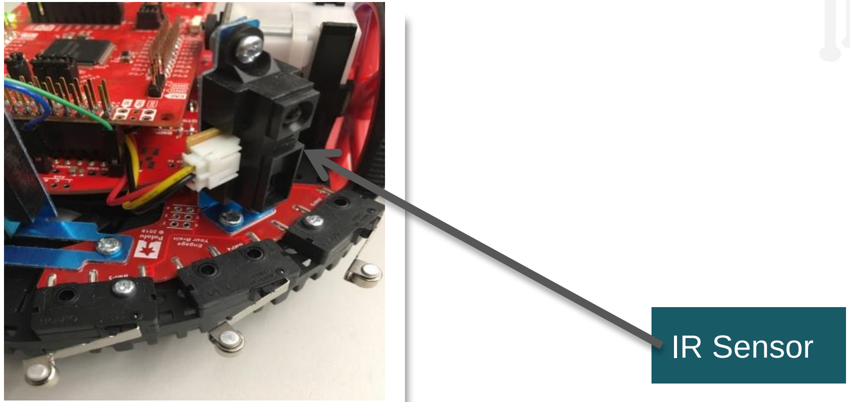
Lecture: Data Acquisition Systems – Performance Measurements

Data Acquisition Systems

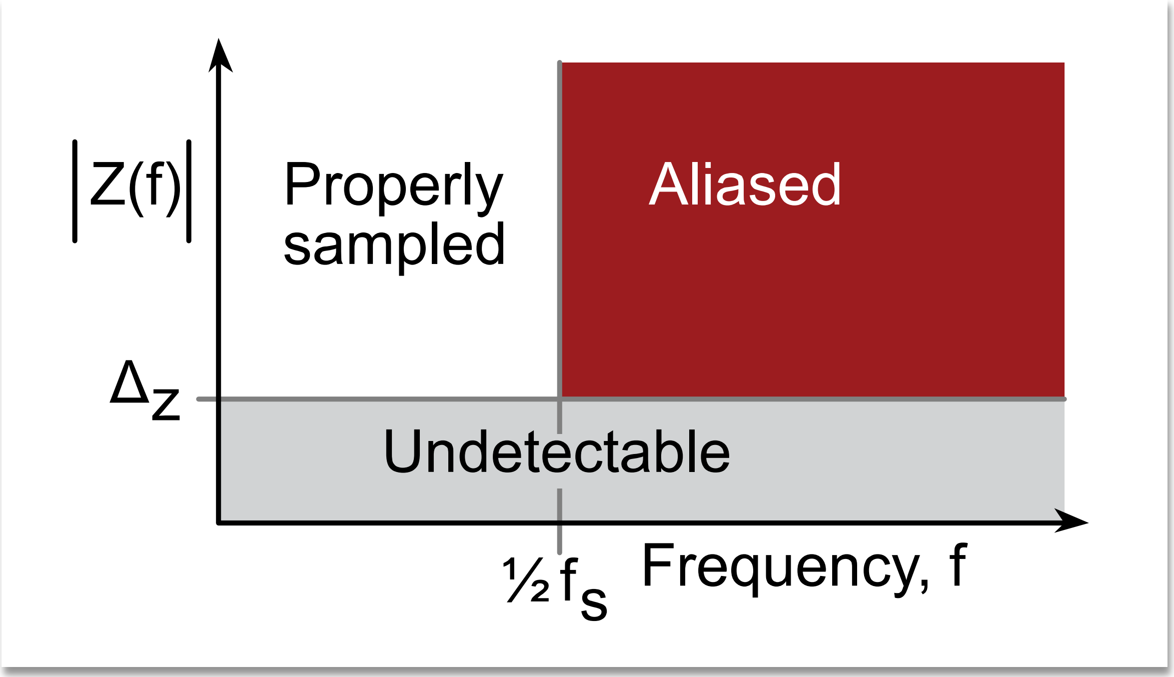
You will learn in this module

- Analog to Digital Converter
  - Sampling, Nyquist Theorem
  - Digital filtering
- Noise and statistics
  - Probability Mass Function
  - Spectrum Analyzer
  - Central Limit Theorem
- Data Acquisition Systems
  - Range, resolution, precision
  - Calibration
  - Accuracy





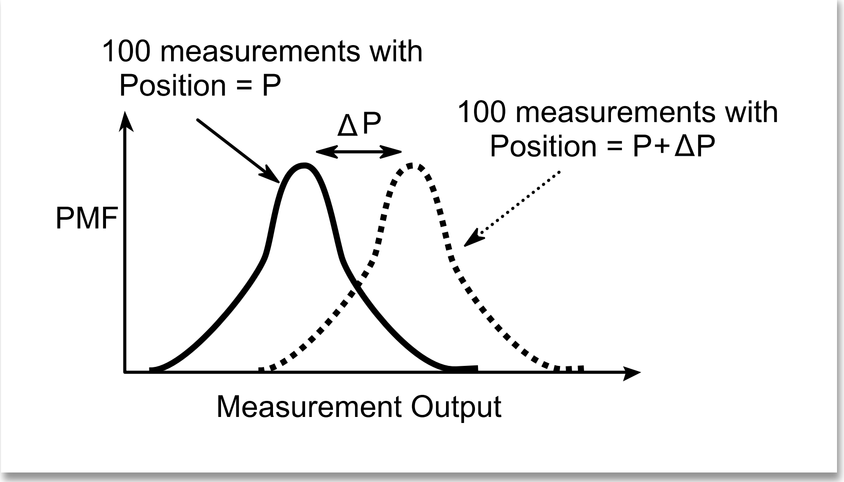
Nyquist Theorem



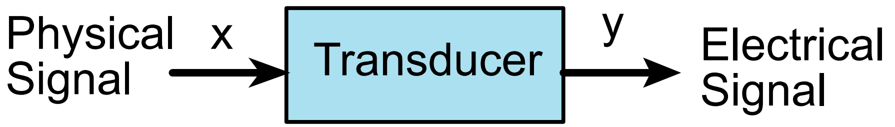
The **Nyquist Theorem** states that if the signal is sampled with a frequency of f_s, then the digital samples only contain frequency components from 0 to ½ f_s.

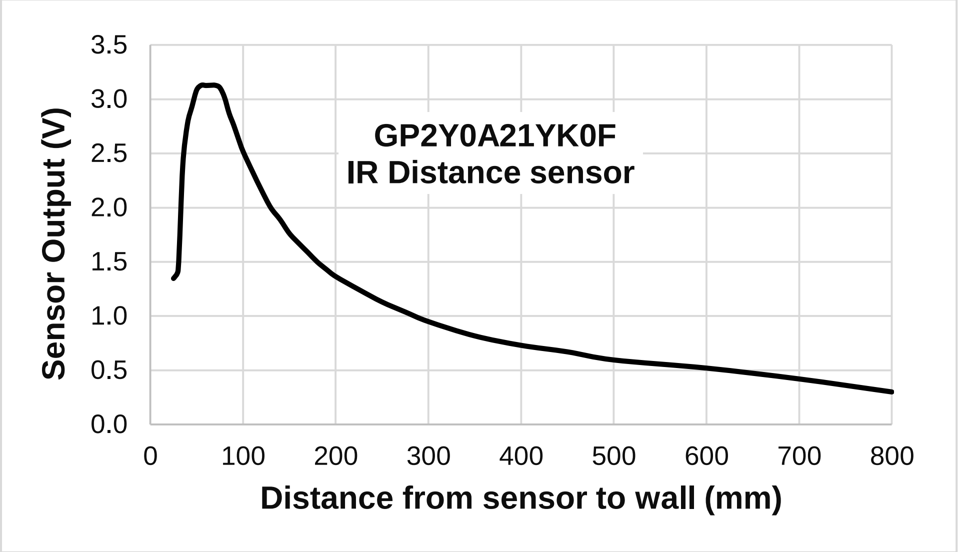
Statistics

- Probability Mass Function (PMF)
- Average (μ = mean)
- Standard deviation (σ = sigma)
- Range (max-min)
- Coefficient of variation, CV = σ/μ
- Precision $\log_2(\mu/\sigma)$
- Resolution, Δ



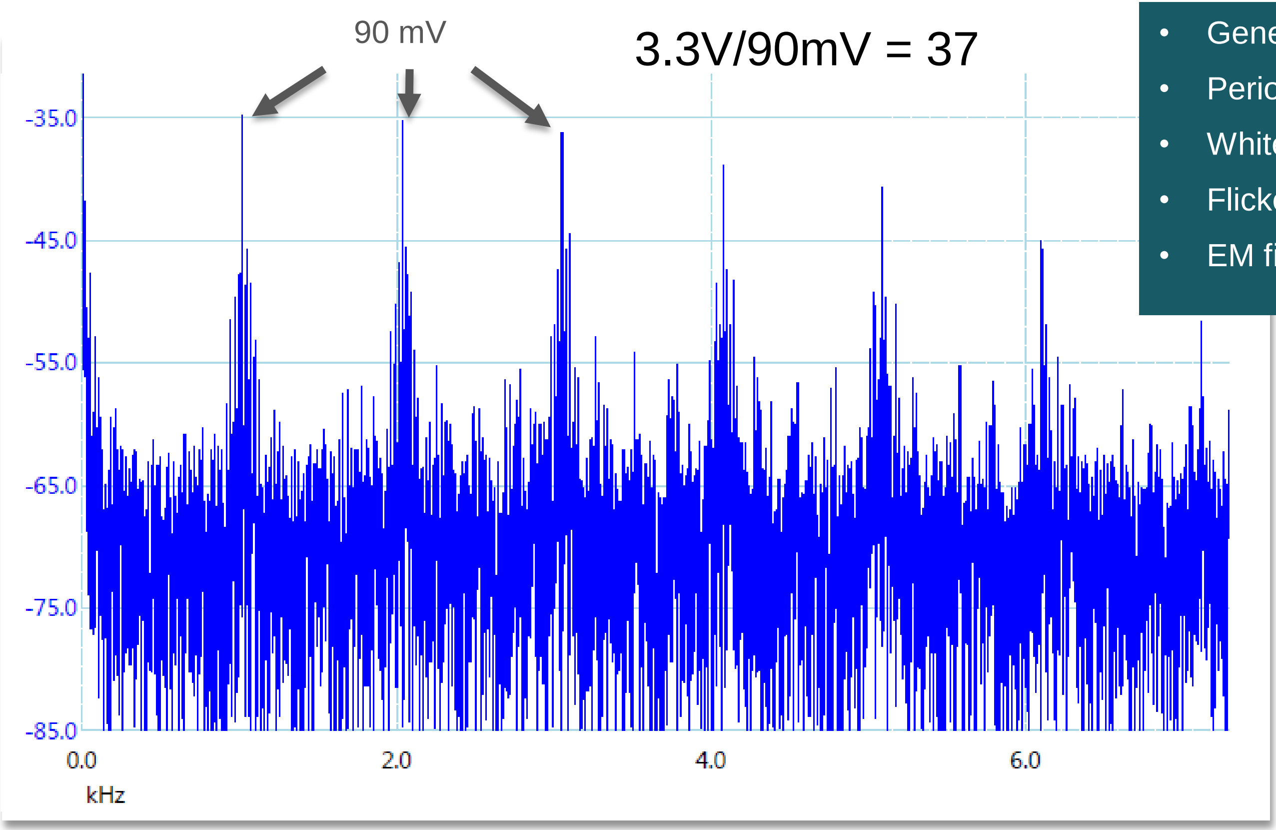
Transducer





- Nonmontonic
- Hyperbolic
- Noisy

GP2Y0A21YK0F IR distance sensors are noisy



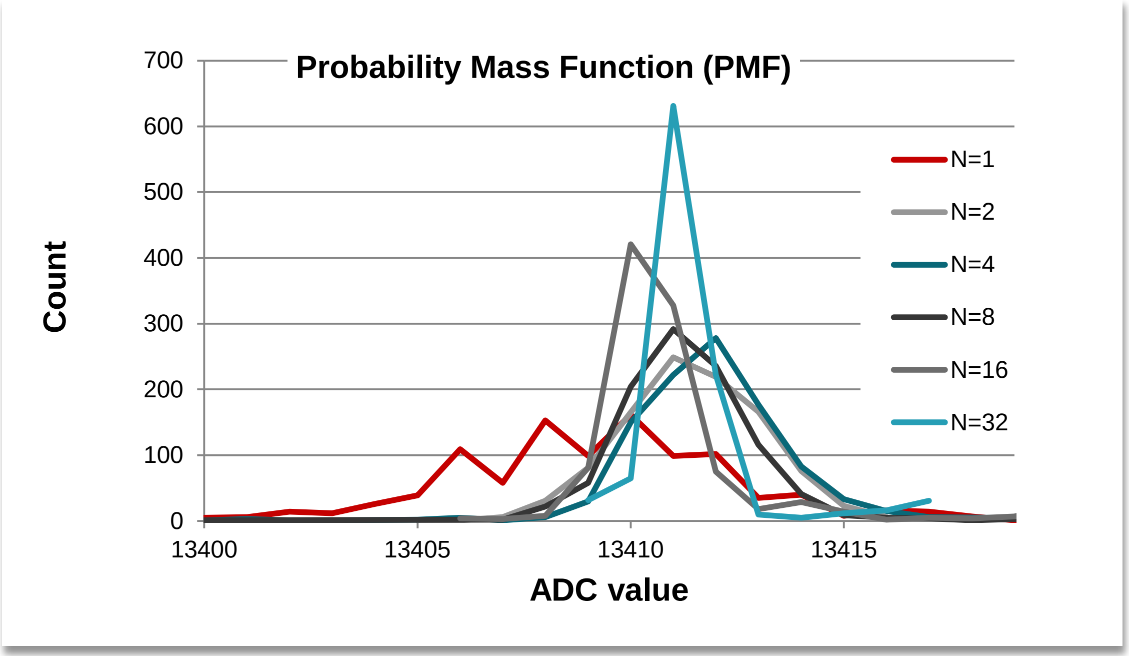
- Generation/recombination
- Periodic
- White
- Flicker, 1/f
- EM field pickup

50 Hz analog LPF

16 Hz digital LPF

$dB_{FS} = 20 \log_{10}(V/3.3)$

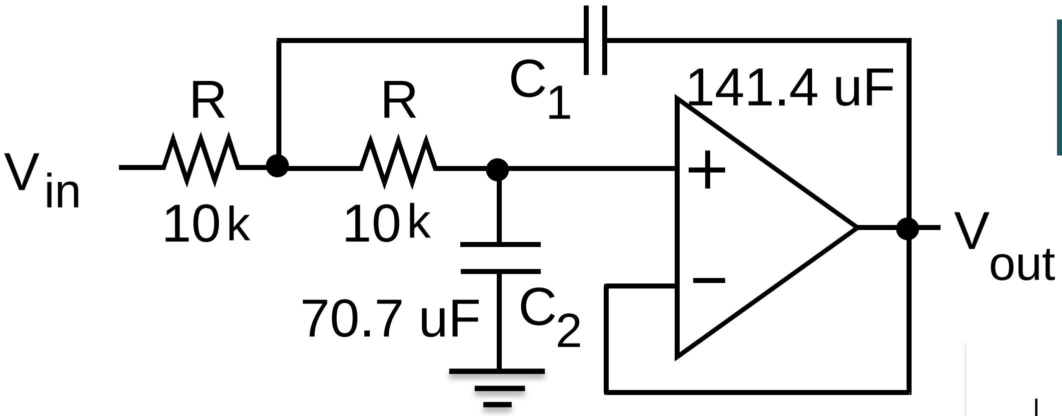
Probability Mass Function (PMF)



CLT states that as independent random variables are added, their sum tends toward a Normal distribution.

- Constant input
- Average of last N samples
- f_s = 1000 Hz

Analog Low Pass Filter to remove Aliasing



Two-pole Butterworth LPF

$$\left|\frac{V_{out}}{V_{in}}\right| = \sqrt{\frac{1}{1+\left(f/f_c\right)^4}}$$

1) Select the cutoff frequency, f_c (50 Hz)

2) Divide the two capacitors by $2\pi f_c$ (let C_{1A}, C_{2A} be the new values)

$C_{1A} = 141.4\mu F/2\pi f_c$ (0.45μF)

$C_{2A} = 70.7\mu F/2\pi f_c$ (0.225μF)

3) Locate two standard value capacitors (with the 2/1 ratio)

$C_{1B} = C_{1A}/x$ (0.44μF, two 0.22 μF in parallel)

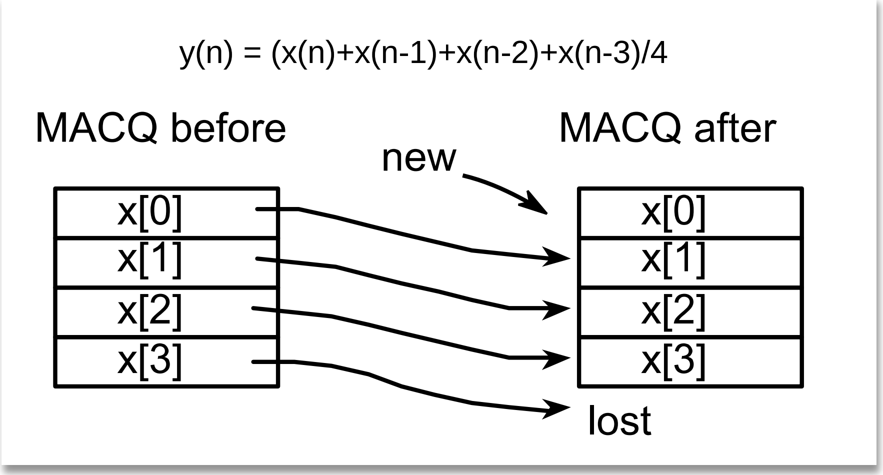
$C_{2B} = C_{2A}/x$ (0.22μF)

4) Adjust the resistors to maintain the cutoff frequency

$R = 10k\Omega \bullet x$ (10k, $f_c = 51$ Hz)

See https://www.ti.com/design-tools/signal-chain-design/webench-filters.html

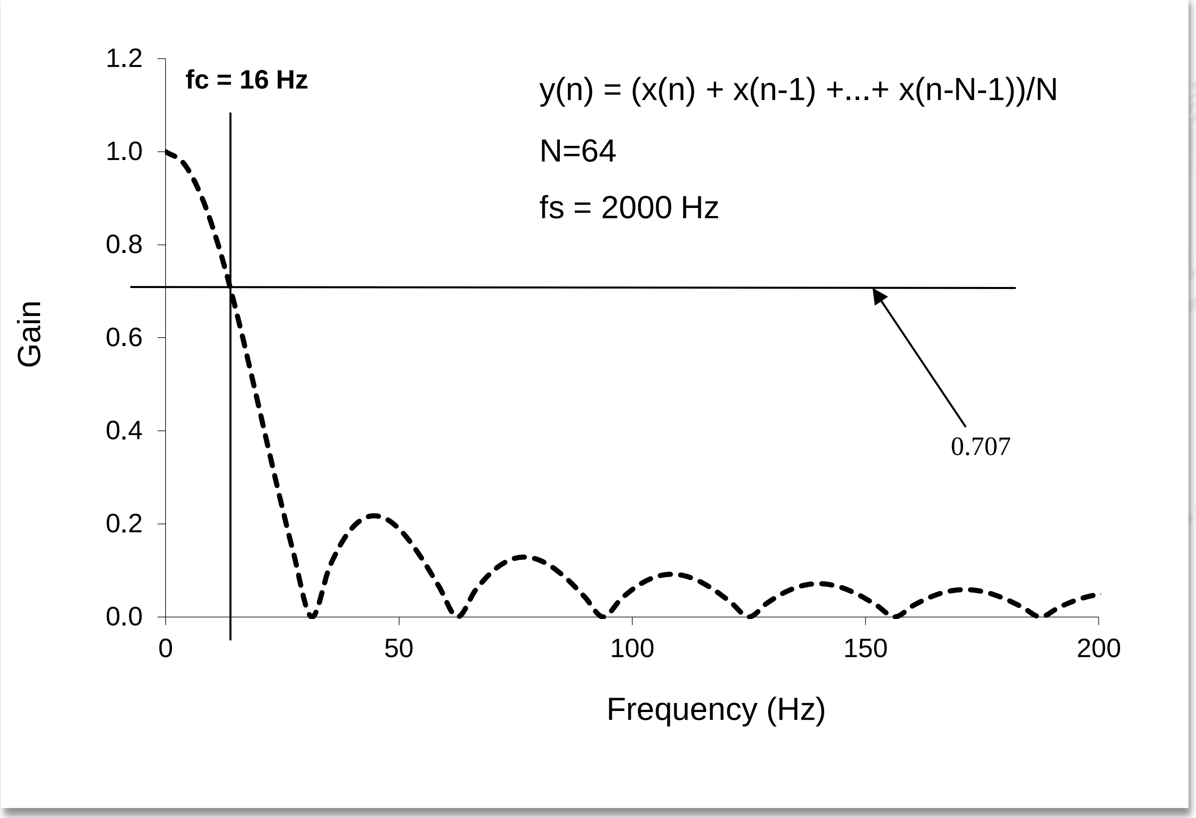
Digital Filtering



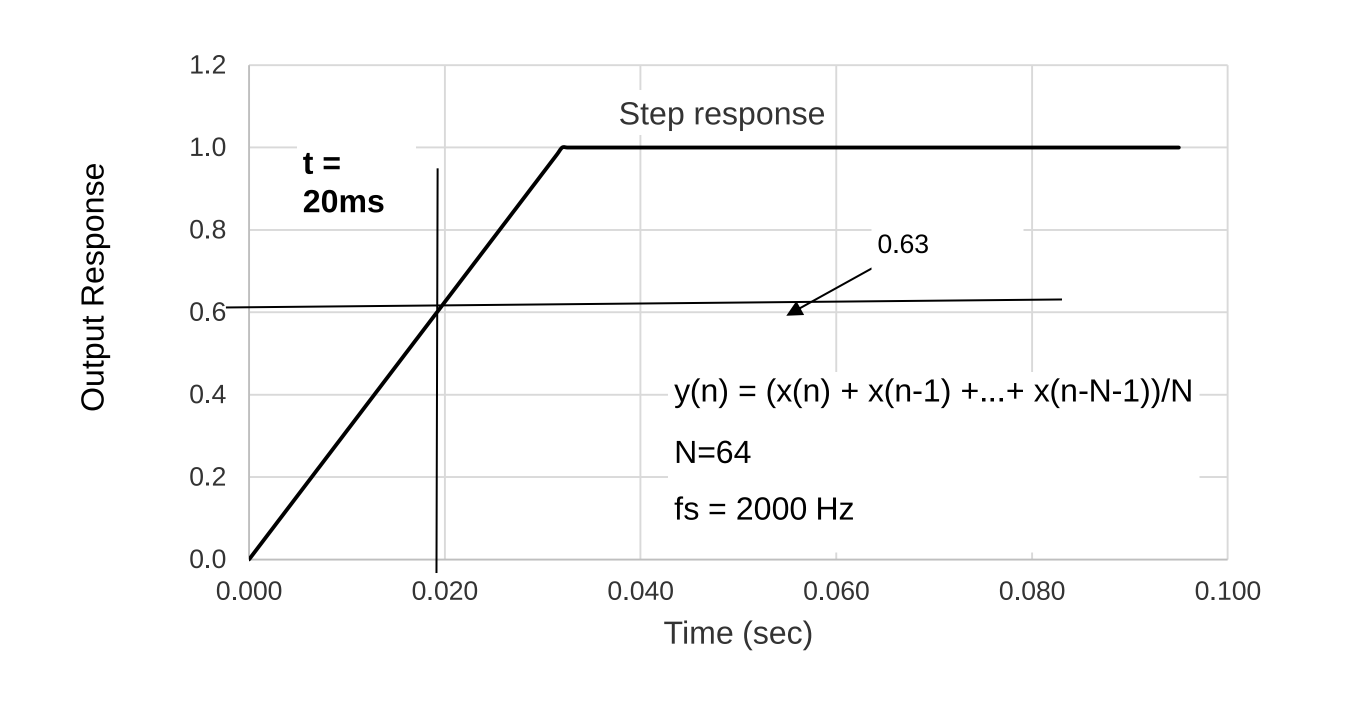

```
x[3] = x[2];
x[2] = x[1];
x[1] = x[0];
x[0] = ADC_In6();
y = (x[0]+x[1]+x[2]+x[3])/4;
```

Averaging Low Pass Filters

- Linear Filter
- Finite Impulse Response
- Low pass

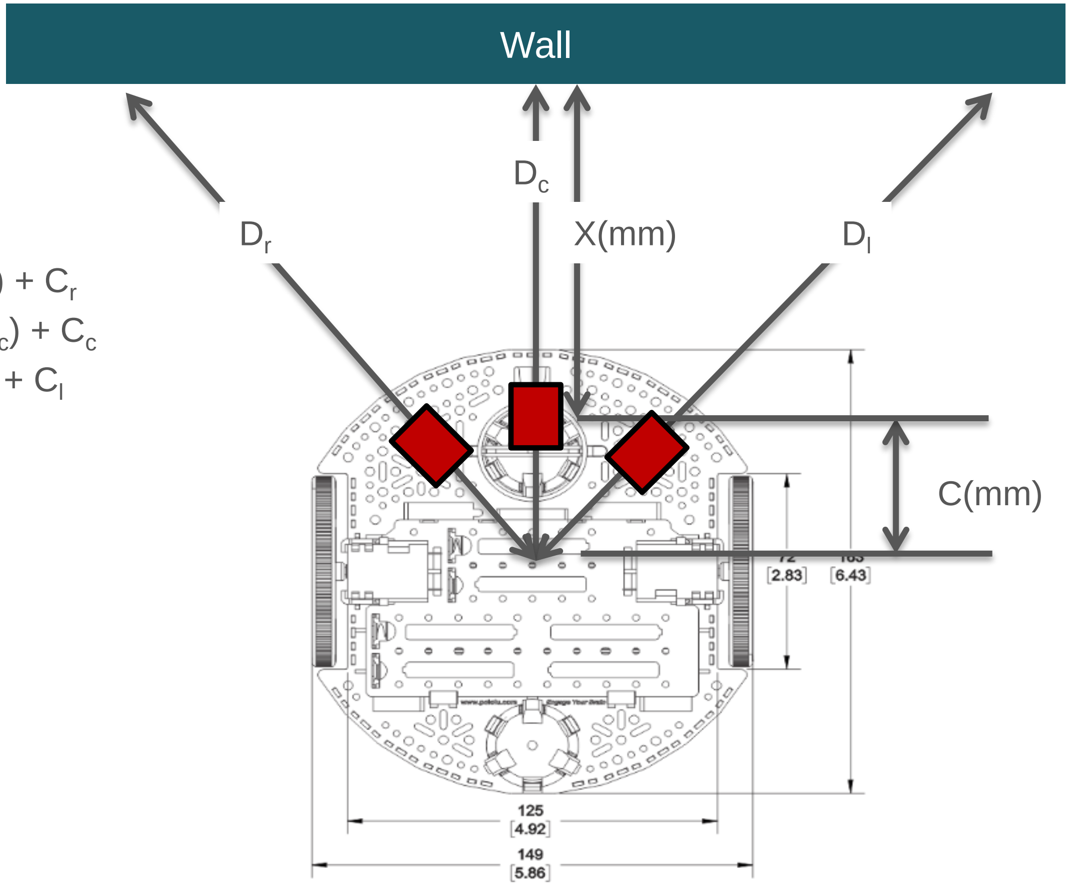


Averaging Low Pass Filters



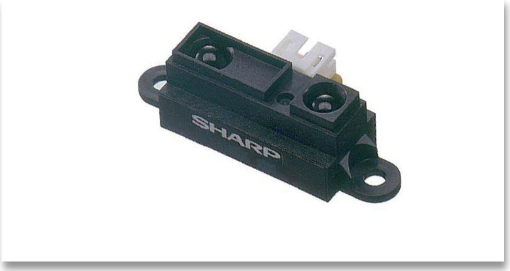
Distance to wall

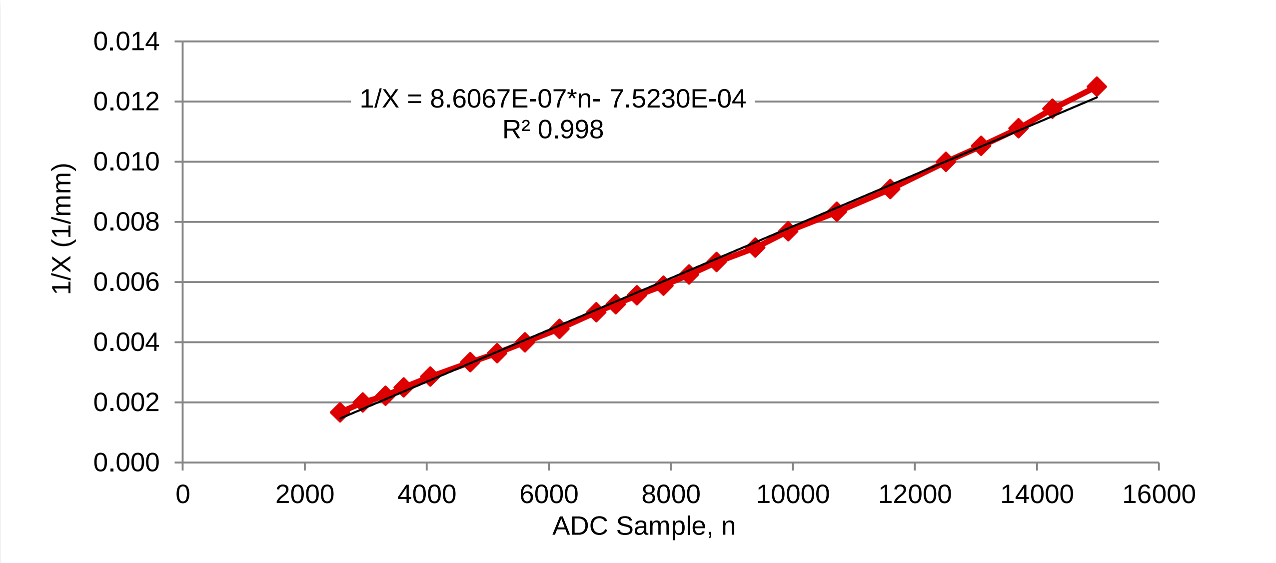
$$D_r = A_r/(n_r + B_r) + C_r$$
$$D_c = A_c/(n_c + B_c) + C_c$$
$$D_l = A_l/(n_l + B_l) + C_l$$



Calibration

- Distance, X, from the sensor to wall, 80 to 400mm
- ADC value, n
- Linear fit 1/X versus n
- Solve for X = A/(n+B)
- Add distance to central point, D = A/(n+B)+C





Summary

Analog to Digital Converter

- Noise

Sampling

- Nyquist Theorem, Aliasing
- Central Limit Theorem

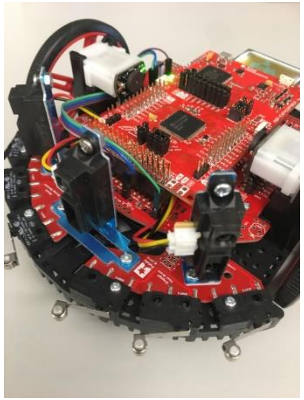
Filters

- Analog LPF
- Digital LPF

Data Acquisition Systems

- Calibration
- Accuracy

$$\frac{100}{n}\sum_{i=0}^{n}\frac{\left|x_{ti}-x_{mi}\right|}{x_{t\max}}$$



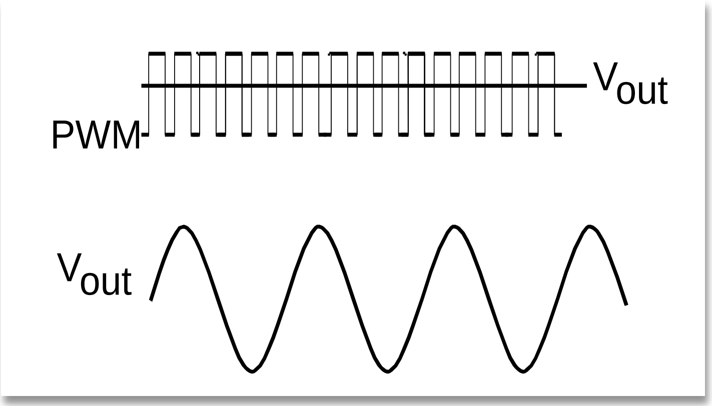
Module 15

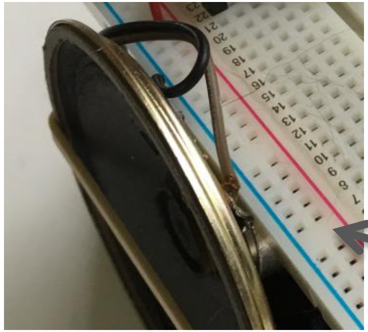
Lecture: Data Acquisition Systems – Sound generation

Data Acquisition Systems

You will learn in this module

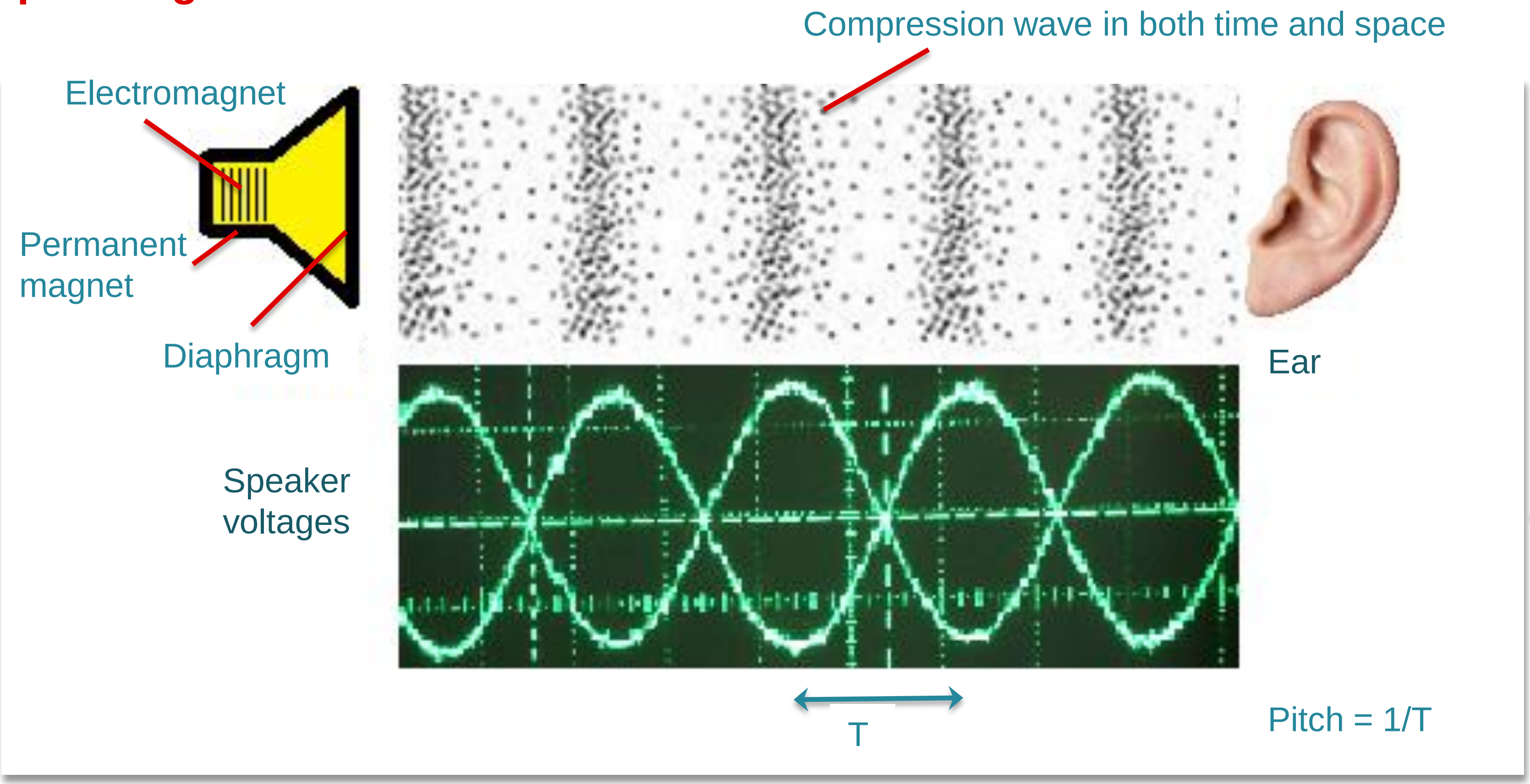
- Signals & Sampling
  - PWM, DAC
  - Range, resolution, precision
- Sound
  - Transducer
  - Analog Circuit
  - Sampling
  - Filtering



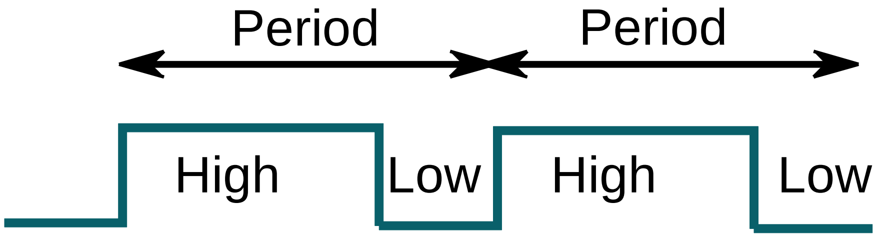


Speaker

Speaker generates sound

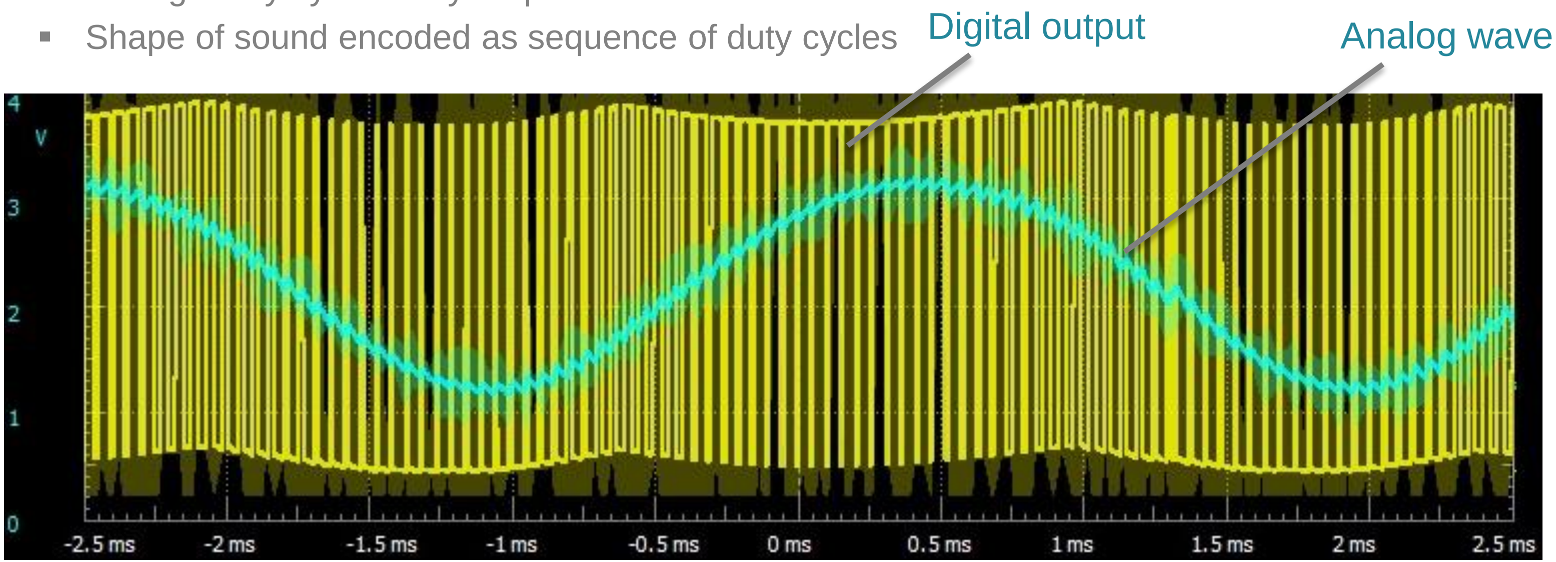


Sampling: PWM DAC



Pulse width modulation

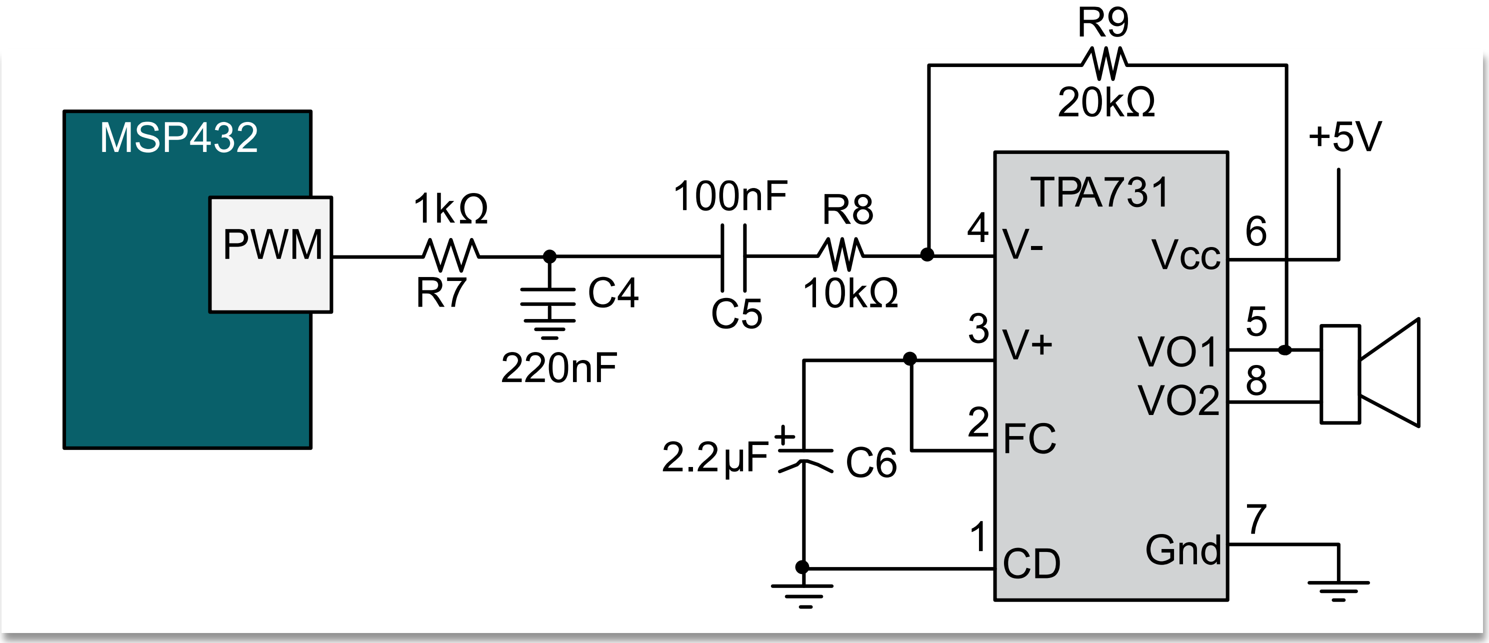
- 20 kHz fixed-period digital output
- Variable duty cycle
- Analog low pass filter removes 20 kHz
- Change duty cycle every 50 µs
- Shape of sound encoded as sequence of duty cycles



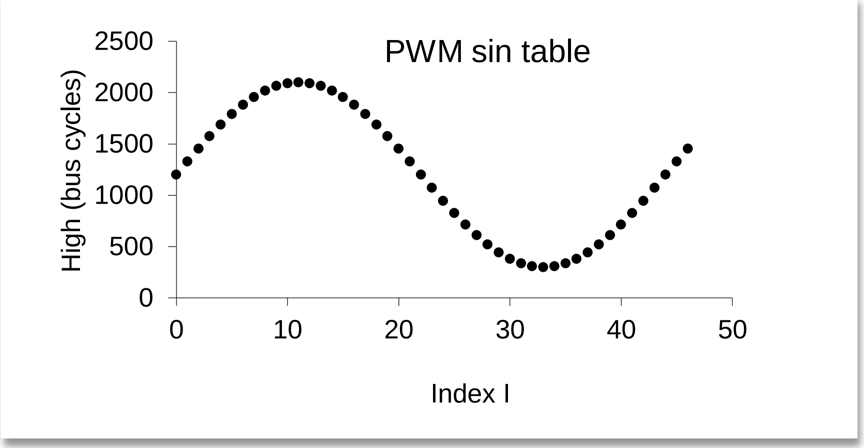
Interface between microcontroller and speaker

LPF to reject PWM frequency
HPF to reject DC from PWM DAC
Provide power to speaker
TPA731 adds DC offset of 2.5V
Gain of 4

LPF cutoff = 1/(2πC4R7) =723Hz
HPF cutoff = 1/(2πC5R8)=159Hz
C6 creates DC offset of 2.5V
Gain=2*R9/R8 = 4



Software to generate PWM outputs



f_{PWM} = 48 MHz/2424 = 19.8 kHz

f_{sound} = 48 MHz/(2424*45) = 440 Hz

$$\text{Dutycycle} = \frac{\texttt{High}}{\texttt{High} + \texttt{Low}} = \frac{\texttt{High}}{\texttt{Period}}$$

```
#define Period 2424
const uint16_t wave440[45] = {
  1212,1339,1463,1583,1695,1798,1890,1968,2032,2079,2110,
  2123,2119,2097,2058,2002,1931,1846,1748,1640,1524,1402,
  1276,1148,1022,900,784,676,578,493,422,366,
  327,305,301,314,345,392,456,534,626,729,
  841,961,1085};
```

Software to generate PWM outputs

```
uint32_t startTime;
void SysTick_Wait2(uint32_t delay){
  volatile uint32_t elapsedTime;
  do{
    elapsedTime = (startTime-SysTick->VAL)&0x00FFFFFF;
  }
  while(elapsedTime <= delay);
  startTime = SysTick->VAL;
}
```

```
while(1){
  High = wave440[i];
  Low = Period-High;
  SysTick_Wait2(Low);
  P3->OUT |= 0x40;    // P3.6 high
  SysTick_Wait2(High);
  P4->OUT &= ~0x40;   // P3.6 low
  i = (i+1)%45;
}
```

$$\text{Dutycycle} = \frac{\mathbf{High}}{\mathbf{High} + \mathbf{Low}} = \frac{\mathbf{High}}{\mathbf{Period}}$$

Summary

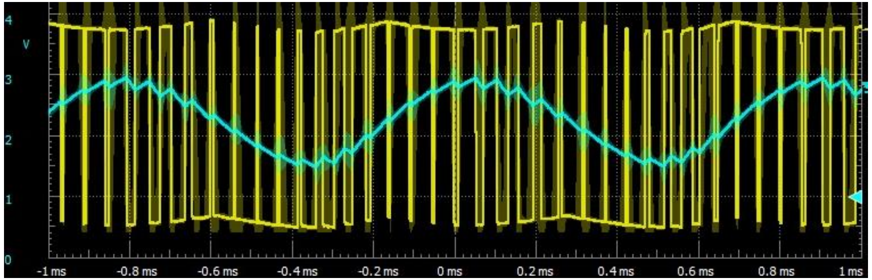
DAC Precision
- Number of different duty cycles
- 48MHz/20kHz = 2400 alternative ≈ 11 bits

DAC Range
- 0 to 3.3V

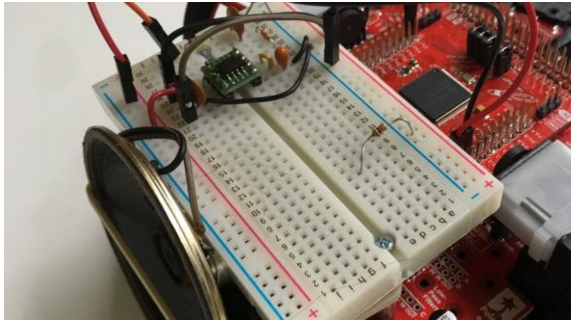
DAC Resolution
- 3.3V/2400
- Limited by noise and LPF cutoff
- Use spectrum analyzer to measure SNR

DAC Speed
- Set by PWM period
- New duty cycle every 50 µs

Sound
- Loudness set by power to speaker
- Pitch set by size of duty cycle array
- Voice set by shape duty cycles in array
- Duration
- Envelope (time varying amplitude)



$$f_{PWM} = 48\text{ MHz}/2424 = 19.8\text{ kHz}$$
$$f_{sound} = 48\text{ MHz}/(2424*45) = 440\text{ Hz}$$



Module 15

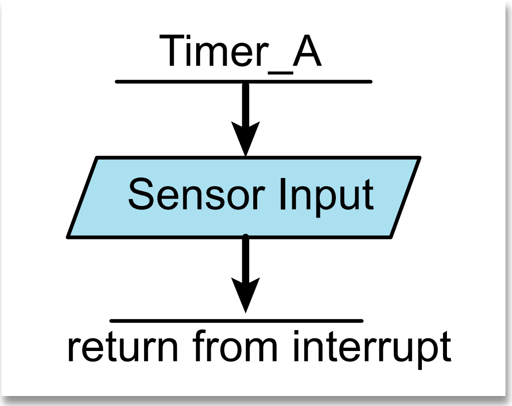
Lecture: Data Acquisition Systems – Sound recording

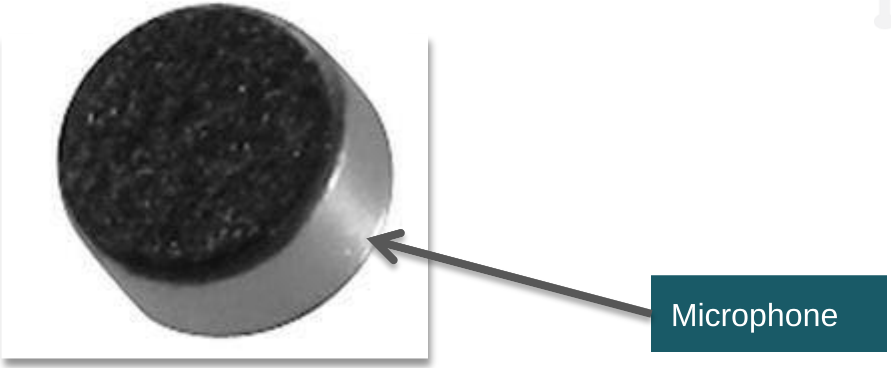
Data Acquisition Systems

You will learn in this module

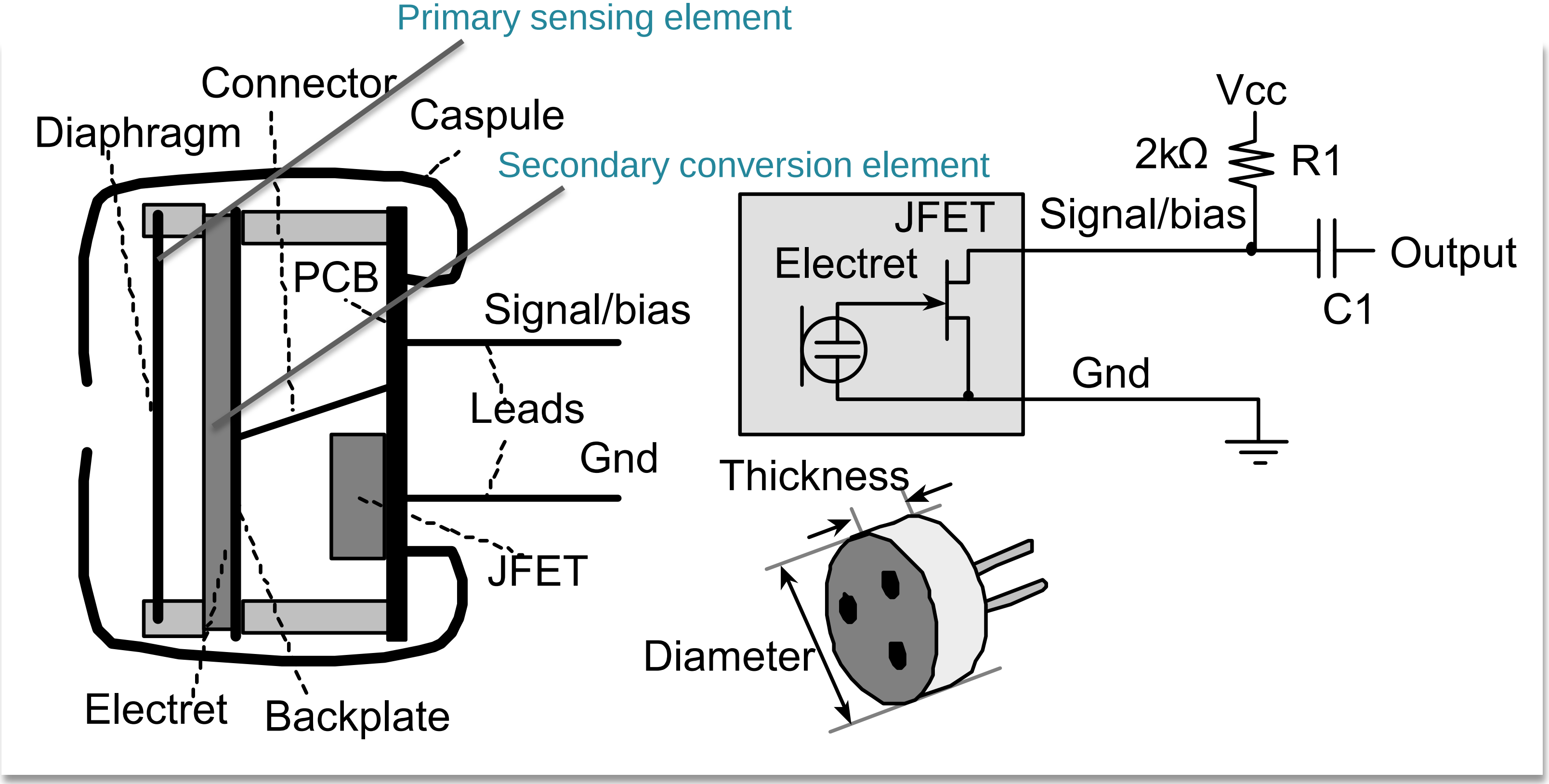
- Signals & Sampling
  - ADC, DAC
  - Range, resolution, precision
  - Successive approximation

- Sound
  - Transducer
  - Analog Circuit
  - Sampling
  - Filtering
  - Pitch Recognition





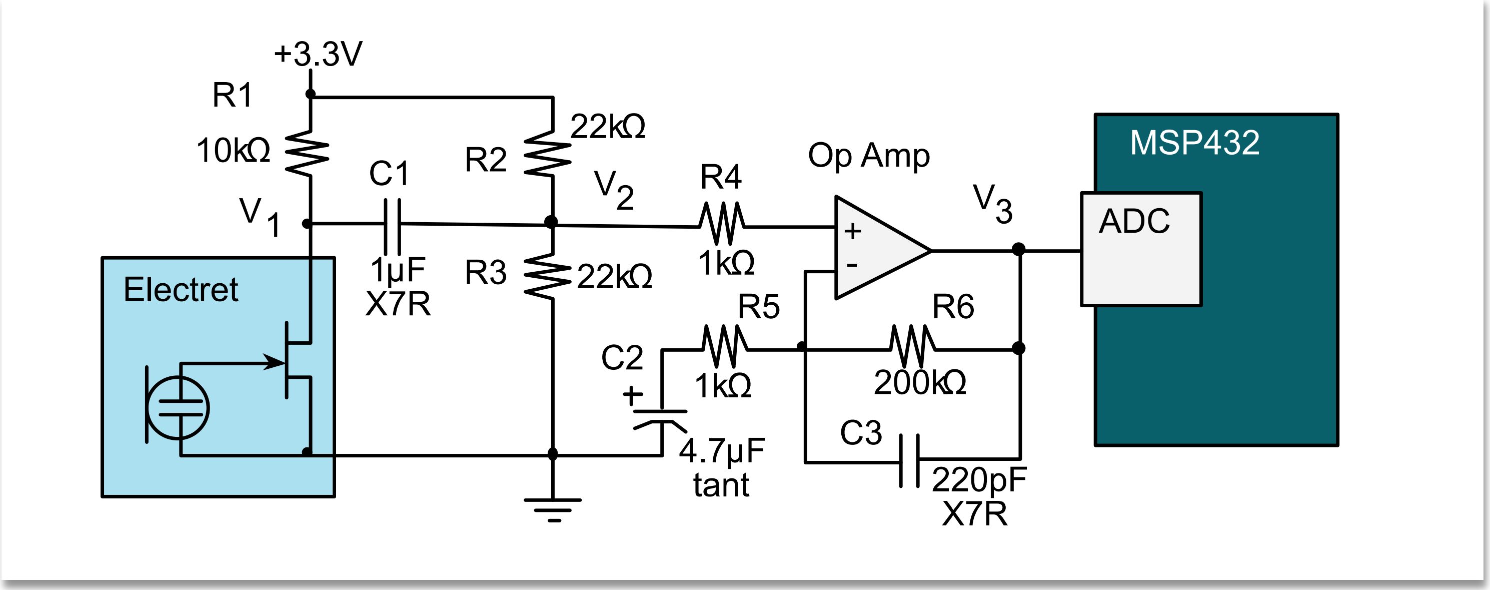
Transducer


Primary sensing element
Connector
Diaphragm
Caspule
Secondary conversion element
PCB
Signal/bias
Leads
Gnd
JFET
Electret
Backplate
Vcc
2kΩ
R1
JFET
Electret
Signal/bias
Output
C1
Gnd
Thickness
Diameter

Interface between microphone and microcontroller

- Provide power to microphone
- HPF to reject DC
- Add DC offset of 1.65V
- Gain of 100
- LPF to prevent aliasing

Gain=1+R6/R5 = 201
HPF cutoff = 1/(2πC1(R2||R3)=14Hz
LPF cutoff = 1/(2πC3R6) =3600Hz



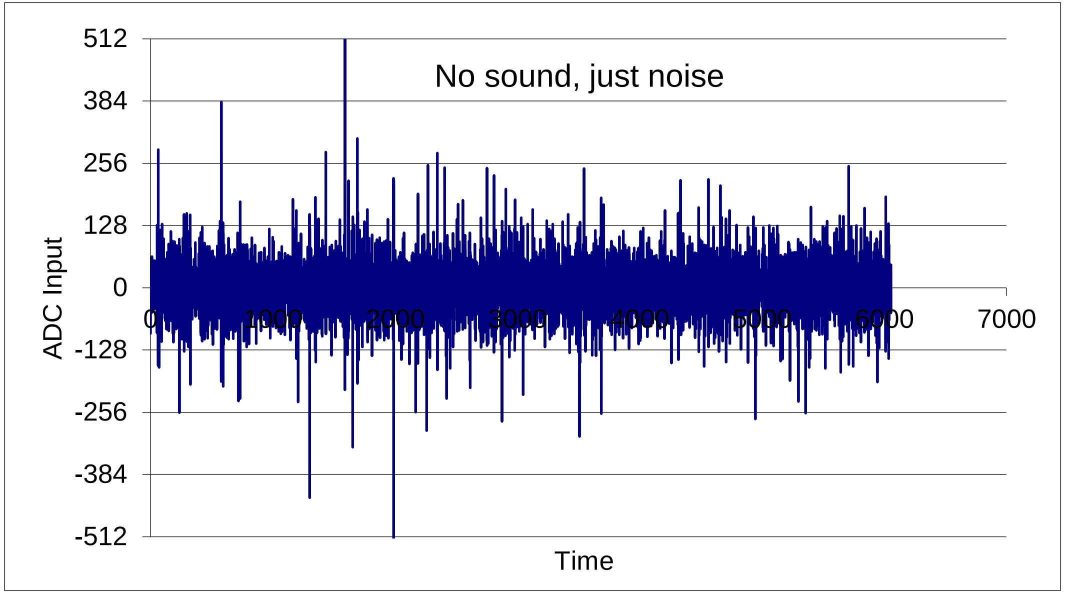
Sampling: adaptive noise rejection

Noise

- Comes from microphone
- Very large
- Random
- Not correlated to itself

Signal

- Comes from sound pressure
- Highly correlated to itself



http://www.ti.com/lit/an/spra657/spra657.pdf

Cross correlation to see if signals are the same shape

Assumptions
- Sampled at f_s
- Data has zero average (just shape)

Measured data
- x(n) is current sample
- x(n-1) is sample Δt ago
- x(n-2) is sample $2\Delta t$ ago

Prerecorded data template
- y(n) is current sample
- y(n-1) is sample Δt ago
- y(n-2) is sample $2\Delta t$ ago

Use cross correlation to see if same shape
- R_{xy} = +large means same shape
- R_{xy} = 0 means not same shape
- R_{xy} = -large means same, but inverted shape

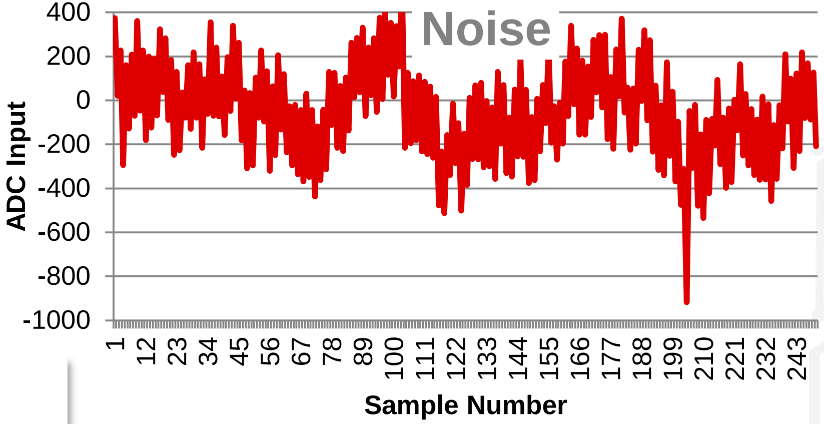
$f_s = 1/\Delta t$

$$R_{xy}(m) = \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} x(n) * y(n-m)$$

Autocorrelation

$f_s = 1/\Delta t$

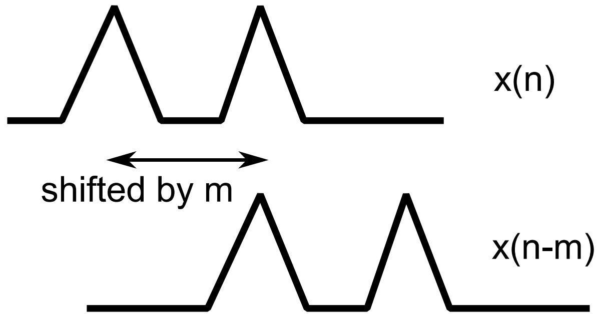
x(n) is current sample
x(n-1) is sample Δt ago
x(n-2) is sample $2\Delta t$ ago
x(n-m) is sample $m\Delta t$ ago



$$R_{xx}(m) = \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} x(n) * x(n-m)$$

$$Rxx2 \approx \frac{127}{128} Rxx2 + \frac{1}{128}\, x(n) * x(n-2)$$

Signal

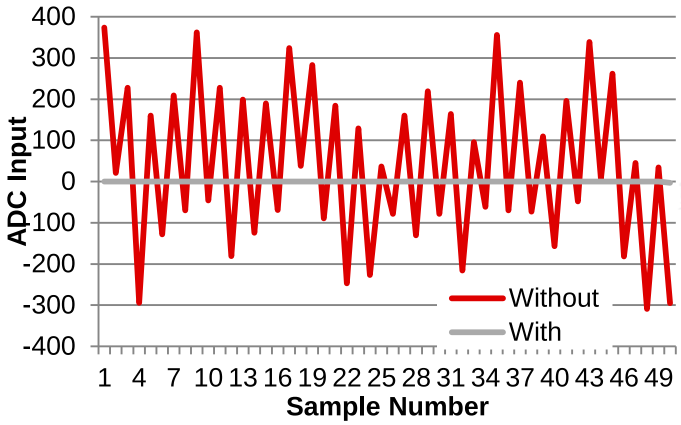


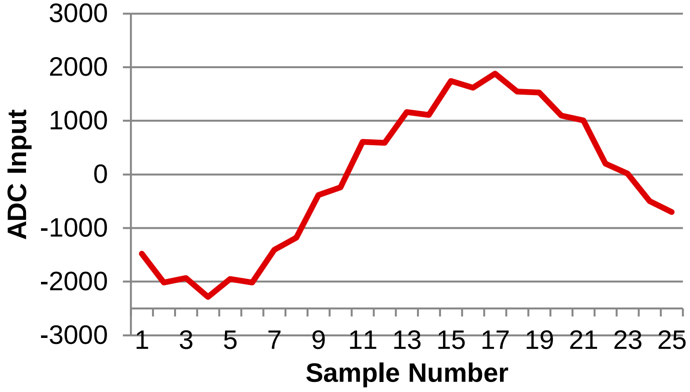
Noise reject filter

Autocorrelation

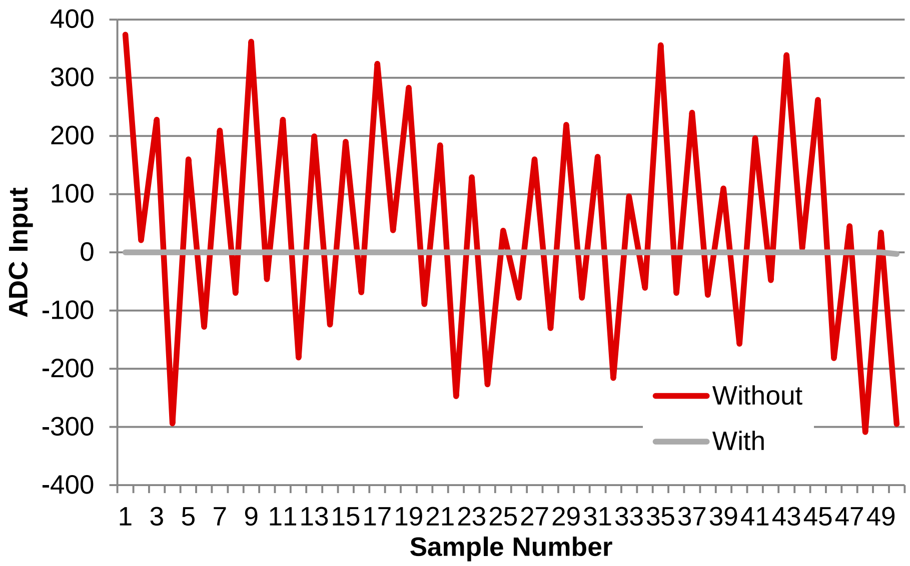
- Rxx2 = M-1 if signal correlated to itself
- Rxx2 = 0 if signal uncorrelated to itself

```
int32_t t0,t1,t2; // last 3 inputs/32
int32_t Rxx2;     // autocorrelation factor
#define K 128     // how fast it responds
#define M 128
int32_t NoiseReject(int32_t x){
  t2 = t1;
  t1 = t0;
  t0 = x/32;
  Rxx2 = ((K-1)*Rxx2 + t0*t2)/K;
  if(Rxx2 < -M) Rxx2 = -M;
  if(Rxx2 >  M) Rxx2 =  M;
  return (Rxx2*x)/M;
}
```

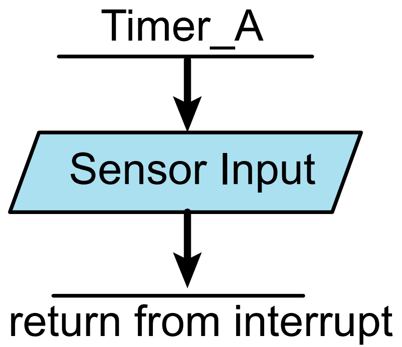




Results of noise suppression



Data acquisition with digital filtering




```
volatile int32_t Raw,Sound;
#define DC 8192
void Program15_2_ISR(void){   // runs at fs=10kHz
  Raw = ADC_In23()-DC;        // sample P8.2
  Sound = NoiseReject(Raw);   // improves SNR
  ADCflag = 1;                // semaphore
}
```

Use sound as command input to robot

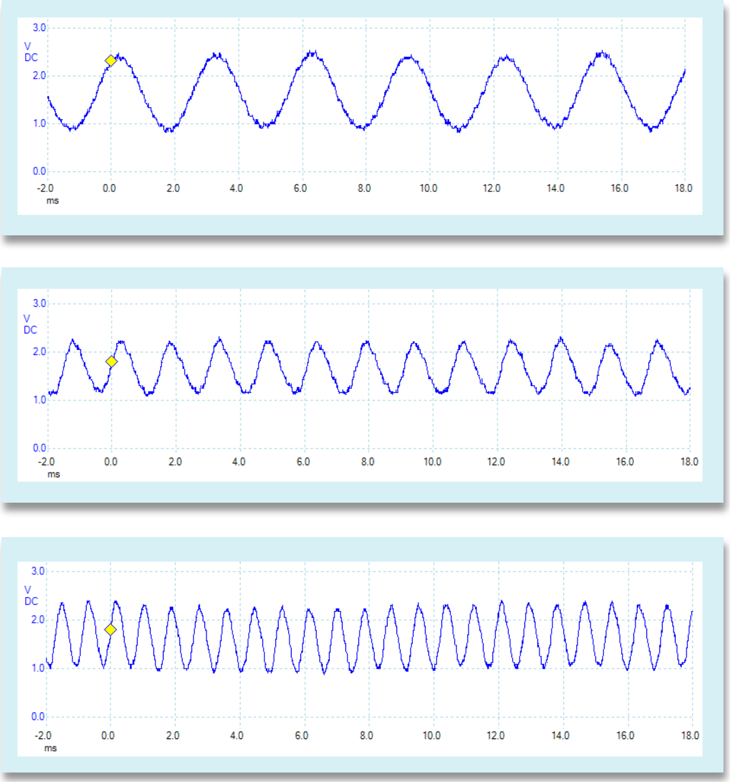
Initial training

- A small number of sine-wave sounds
- Train by recording example of each
- Examples are the templates y(n)
  - Fixed sampling rate
  - Variable size to capture complete waves

Use cross correlation to distinguish

- Calculate R_{xy} for each template
  - Use finite size buffers
  - Calculate max R_{xy} for various m
- Pitch recognition
  - Best match is largest R_{xy}
  - Above a threshold

Frequency Key Shifting (FSK)



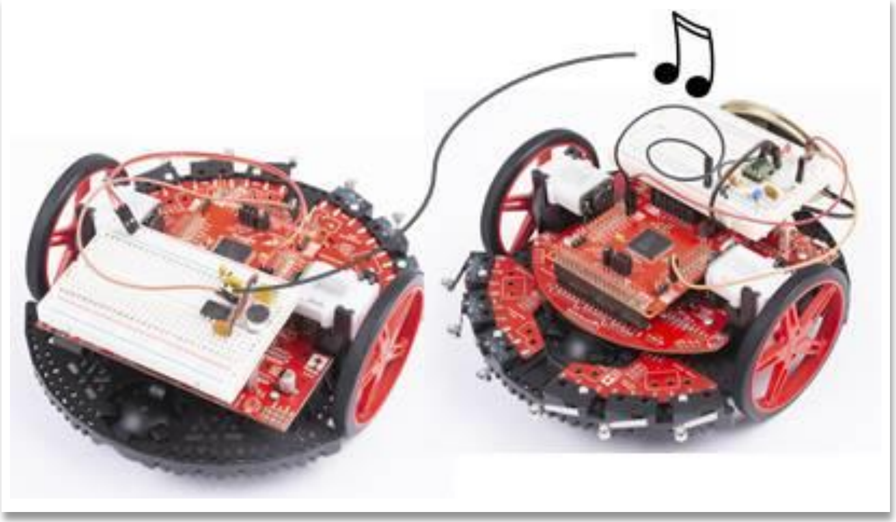
Summary

Sound recording

- Transducer
- Gain
- Analog filter

Digital processing

- Noise rejection using autocorrelation
  - Noise does not correlate with itself
  - Signal does correlate with itself
- Pitch recognition (FSK)
  - Training session
  - Cross correlation to find best match



ti.com/rslk

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265