Scientific Electronic Instrumentation

Peter Stallinga, September 2009

Universidade do Algarve

Outline

0. Introduction

1. Signal conditioning

- Op-amp circuits
- Wheatstone bridge
- Noise
- Lock-In amplifier
- Cables (coaxial, twisted pair, optical fiber, etc.)

2. Signal generation (Sensors)

- Actuators: Relais, solid-state switch
- Temperature: Thermo-couple, PT100, diode, LM35DZ, bimetal
- Optical: LDR and optodiode.
- Movement: RPM and Doppler
- Length
- Bending (extensiometers)
- Acceleration
- Angle (Gray code)
- Magnetic field: Hall and NMR
- Humidity
- Pressure: Membrane, Piranni, Penning
- (Sound) pressure sensors (microphone) and actuators (speakers)
- Stepper motors
- *MEMs*
- *Mass: QCM*
- *Gravity*

3. Signal acquisition

- Analog-digital converters (ADC/DAC)
- Interfacing
- Parallel port
- Serial port (null modem, DTE/DXE, handshaking)
- USB
- GPIB
- Programming (LabVIEW, MatLab, Python, PASCAL)

4. Signal processing

- *Simple digital filter (MA, etc.)*
- *Feedback (PID control)*

5. A scientific laboratory

- *Vacuum pumps*
- *Cryogenic equipment*
- *Monochromator*
- *Measurements:*
    - *o Hall*
    - *o 4-Point*
    - *o Admittance*

Bibliography:

- "Instrumentation for engineers and scientists", J. Turner and M. Hill, Oxford Science Publications, 1999, ISBN 0-19-856517-8
- "The art of electronics", 2nd ed., P. Horowitz and W. Hill, Cambridge University Press, 1989, ISBN 0-521-37095-7
- Sebenta Prof. J. Bastos, Universidade do Algarve.
- "MEMS Mechanical Sensors". S. Beeby, G. Ensell, M. Kraft, N. White, Artech House, 2004, ISBN 1-58053-536-4.
- "Sistema Internacional de Unidades (SI)", 3ª ed., G. de Almeida, Plátano Edições Técnicas, 2002, ISBN 972-707-162-7
- Wikipedia: http://www.wikipedia.org
- Lecture notes Electronics II, P. Stallinga, Universidade do Algarve
- "Electronic Instruments", 4th ed., D.R. Patrick, Prentice Hall, 1992, ISBN 0-13-251208-4

Introduction

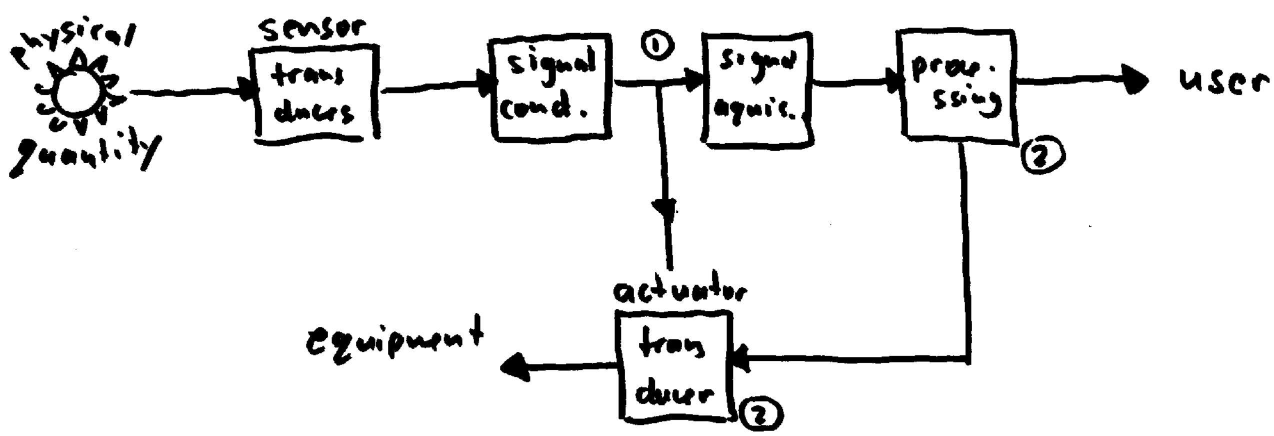
Instrumentation

Measuring physical quantities, processing them, and causing actions.

Imagine we want to control the temperature of a room. We need to measure the temperature, translate this to a processable signal, aquire the signal, process the signal (for instance computer), take decissions and switch the air conditioning (or any other temperature changing element). There are many ways of doing this. Not always are all steps necessary. The computer can be bypassed, especially when the processing is simple: "air-co on when too warm, off when too cold".

Some parts are explained in detail in other lectures (electronics, control, programming).

We will decribe here the steps of going from a physical quantity to a measurement system



(1) Either directly to the actuator or to a signal aquisition / processing element

(2) In case of monitoring / measurement system no actuator is needed
In case of control (of temperature, for instance) an actuator is needed and the "monitor" (user) can be absent.

<u>Transducers</u> are elements that convert one type of signal (or energy in one domain) to another type of signal (or energy in a domain). As a good example is a temperature sensor which translates temperature (unit K) to to electrical voltage (V) in a thermo couple. Also the other direction exists. A heating element translates electrical energy into heat. These kind of elements are normally called actuators and can be seen as "output" elements

where sensors can be seen as "input" elements.

Signal conditioning

Normally, at this stage the system signal is in the electrical domain (voltage or current) and the signal has to be prepared for aquisition. As an example, a thermocouple (type K) has a sensitivity of $41\ \mu V/^\circ C$. If we were to connect this directly to an 5V ADC of 8 bit, the resolution is $5V/2^8 = 19.5\ mV = \frac{19.5\ mV}{41\ \mu V/^\circ C} =$

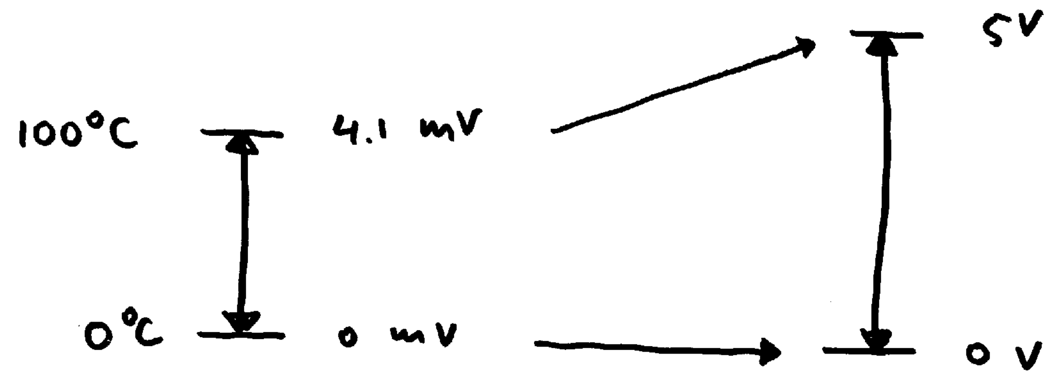
$476\ ^\circ C$

Even for a much more expensive 16 bit ADC the resolution is only $\frac{5}{2^{16}} = 76.2\ \mu V =$

$$\frac{76.2\ \mu V}{41\ \mu V/^\circ C} = 1.9\ ^\circ C$$

(For the moment ignoring the noise aspects).

Obviously, it is best to amplify the signal until it falls in the range of input voltages for the range of sensor signals expected. If we want, for instance, to have a system measuring between 0°C and 100°C, based on



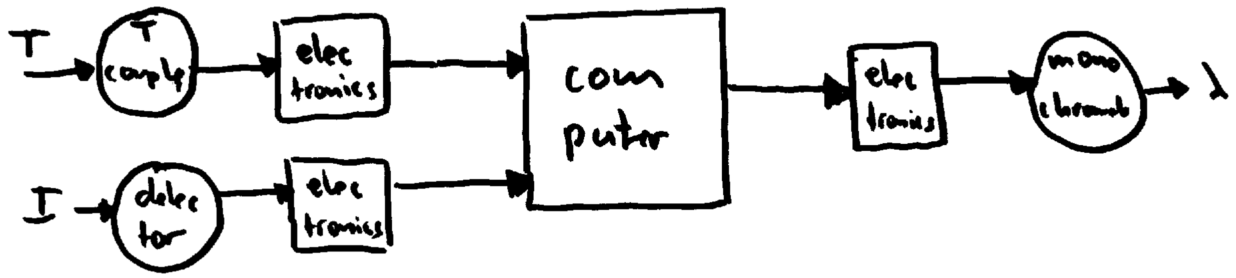
a 5 V ADC, we need to "map" the output of the sensor whose range is 0°- 100°C or in volt: 0 mV - 4.1 mV to the input of the ADC 0-5 V. This is an easy case, we just need a linear amplifier of a factor $\frac{5000}{4.1} = 1220$. In some cases do we also have to remove offset. The Wheatstone bridge can be used for that. This will all be discussed in the first chapter.

In some cases the signal has to be translated from one domain (for instance current) to another (for instance voltage).

Signal Acquisition

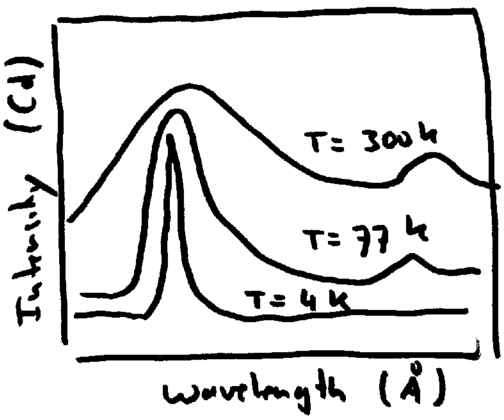
This is the process of converting the signal to digital format and entering it to a computer, where further data treatment can take place

<u>Processing</u>. This can consist of things like digital filtering, curve fitting, statistics, decission making and graphical representation, etc.



As an example an experimental set up for measuring a luminescence spectrum based on a monochromator that selects the wave length of the light to be measured (λ), a photo deatector that measures (transduces) light intencity (I), and a thermocouple.

The computer processes all the information and shows, on the screen, or printer, or on file. the spectra as a function of temperature



Note that in this case the computer also makes "decissions". It sets the wavelength of the monochromator. In some cases the decisions are simpler, for instance "switch on the light" (when the sensor detecting light intensity gave a small signal). In many cases, these decisions can be made outside the computer, thus reducing the cost, and implemented with electronics.

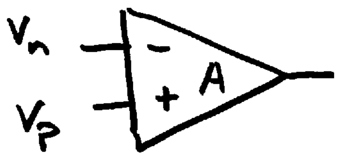
Finally, one basic problem of (scientific) measurement is that the measurement itself is changing the measured quantity. As an example, when we want to measure the temperature, we can place a thermocouple close to the sample. However, this is a source of heat, or a heat sink, thus changing the temperature itself. This is a recurring problem and something we have to take into account when designing a system.

1

Signal Conditioning

Signal conditioning, as discussed in the introduction means altering the signal in magnitude, spectral distribution, or domain (current ⇄ voltage, e.g.) to make it suitable for the rest of the circuit.

Operational Amplifiers



$A(V_p - V_n)$

The most famous, simple, amplifier is the operational amplifier (opamp). For this lecture we will use the ideal opamp in the analysis and design of the circuits. This means the following:

Ideal Op Amp:

1. Infinite gain ($A = \infty$)
2. Infinite input resistance ($r_i = \infty$)
3. Zero output resistance ($r_o = 0$)

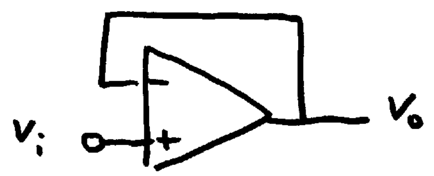
The direct result of (1) is that the amplifier will have zero difference at the entrance:

(1a) $V_o = A(V_p - V_n) \Rightarrow V_p - V_n = \frac{V_o}{A} = \frac{V_o}{\infty} = 0$

$V_p = V_n$

This can be used in the analysis of Op Amp circuits and is valid for circuits that are not in "saturation" (V_o equal to, limited by the power supply voltage)

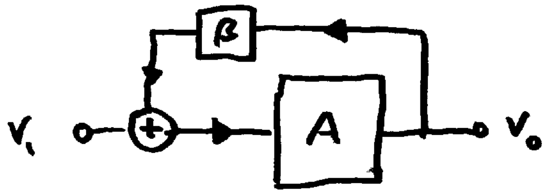
The easiest circuit we can make with an op-amp is a tension-follower:



since $V_p = V_n$ and $V_o = V_n$, $V_i = V_p$

$$\boxed{V_o = V_i}$$

Sometimes op-amp circuits are best (easiest) described in terms of feedback.



$$V_o = (V_i + \beta V_o) \cdot A$$

The gain of the circuit is then

$$\frac{V_o}{V_i} = \frac{A}{1 - A\beta}$$

For our ideal opamp $(A = \infty)$ this becomes

$$\frac{V_o}{V_i} = -\frac{1}{\beta}$$

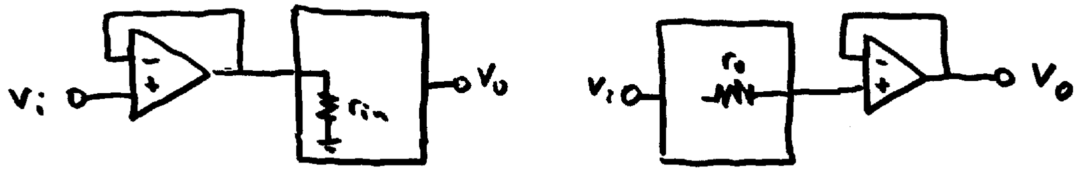
with β the feedback factor

$$\beta \equiv \frac{V_n}{V_o}\Big|_{V_i = 0}$$

As an example, for the tension follower, $\beta = -1$, and

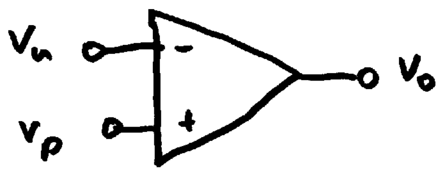
$$\frac{V_o}{V_i} = 1 \Rightarrow V_o = V_i$$

The advantage of a tension follower is that it change the output and input resistances of parts of circuits. This avoids unnecessary signal losses.



ampl. with low r_{in} ampl. with high r_o

A <u>comparator</u> is a circuit that has either $+V_{cc}$ or $-V_{cc}$ (the power supply voltage) as output, depending on the difference of the voltage at the input terminals



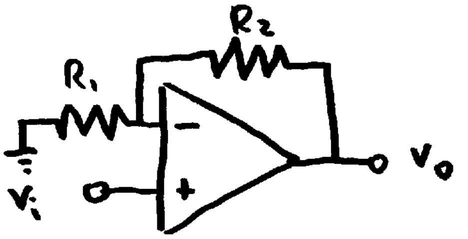
$V_n < V_p$: $V_o = +V_{cc}$

$V_n > V_p$: $V_o = -V_{cc}$

A popular comparator is the "311" which has a high switching speed $-V_{cc} \leftrightarrow +V_{cc}$.

A <u>non-inverting amplifier</u> can be made by negative feedback to the amplifier.

To calculate the gain is easy:

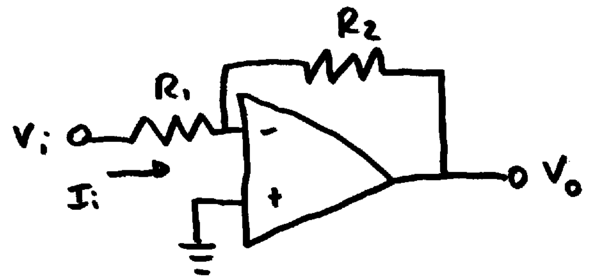


$$V_n = V_p = V_i$$

$$V_n = \frac{R_1}{R_1 + R_2} \cdot V_o$$

$$\Rightarrow \quad \frac{V_o}{V_i} = \frac{R_1 + R_2}{R_1}$$

An inverting amplifier is equal to the one above but with V_i connected to R_1



Because $V_n = V_p$ (rule (1a)), $V_n = 0\,V$ and that is why it is called virtual ground. The input current can then be calculated

$$I_i = \frac{V_i}{R_1}$$

This current cannot enter the op amp (rule (2)) and can thus only "escape" via R_2. The voltage induced by this current at the output is then

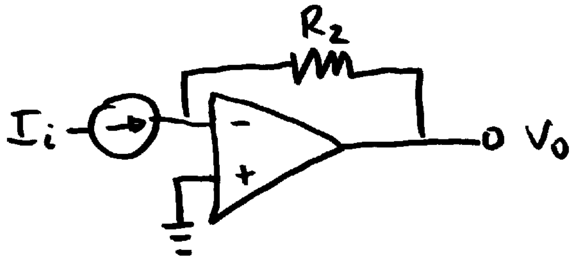
$$V_o = V_n - I_i \cdot R_2$$

$$= 0 - \frac{V_i}{R_1} \cdot R_2$$

$$= -\frac{R_2}{R_1} V_i$$

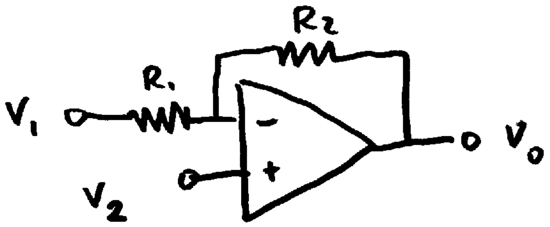
$$\Rightarrow \quad \frac{V_o}{V_i} = -\frac{R_2}{R_1}$$

Current-to voltage converter



$$\frac{V_o}{I_i} = -R_2$$

A differential amplifier :

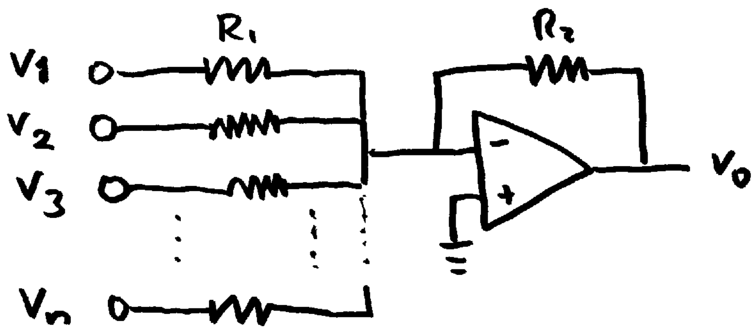


Because the op amps are linear components, we can use the super position principle

$$V_o = \frac{R_1 + R_2}{R_1} \cdot V_2 - \frac{R_2}{R_1} \cdot V_1$$

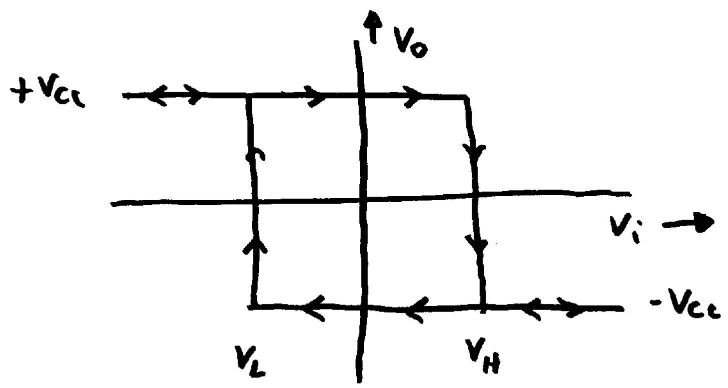
$$\approx \frac{R_2}{R_1} (V_2 - V_1)$$

A Summing amplifier

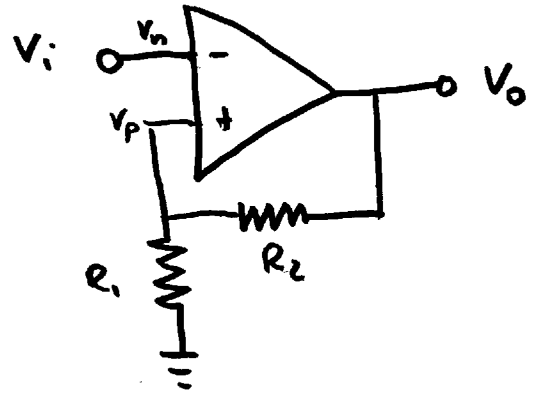


$$V_o = -\frac{R_2}{R_1} \sum_{i=1}^{n} V_i$$

A <u>Schmitt trigger</u> is like a comparator, but it has a memory effect. The output of the Schmitt trigger does not only depend on the input voltage but also on the history.



To achieve this, positive feedback is used:



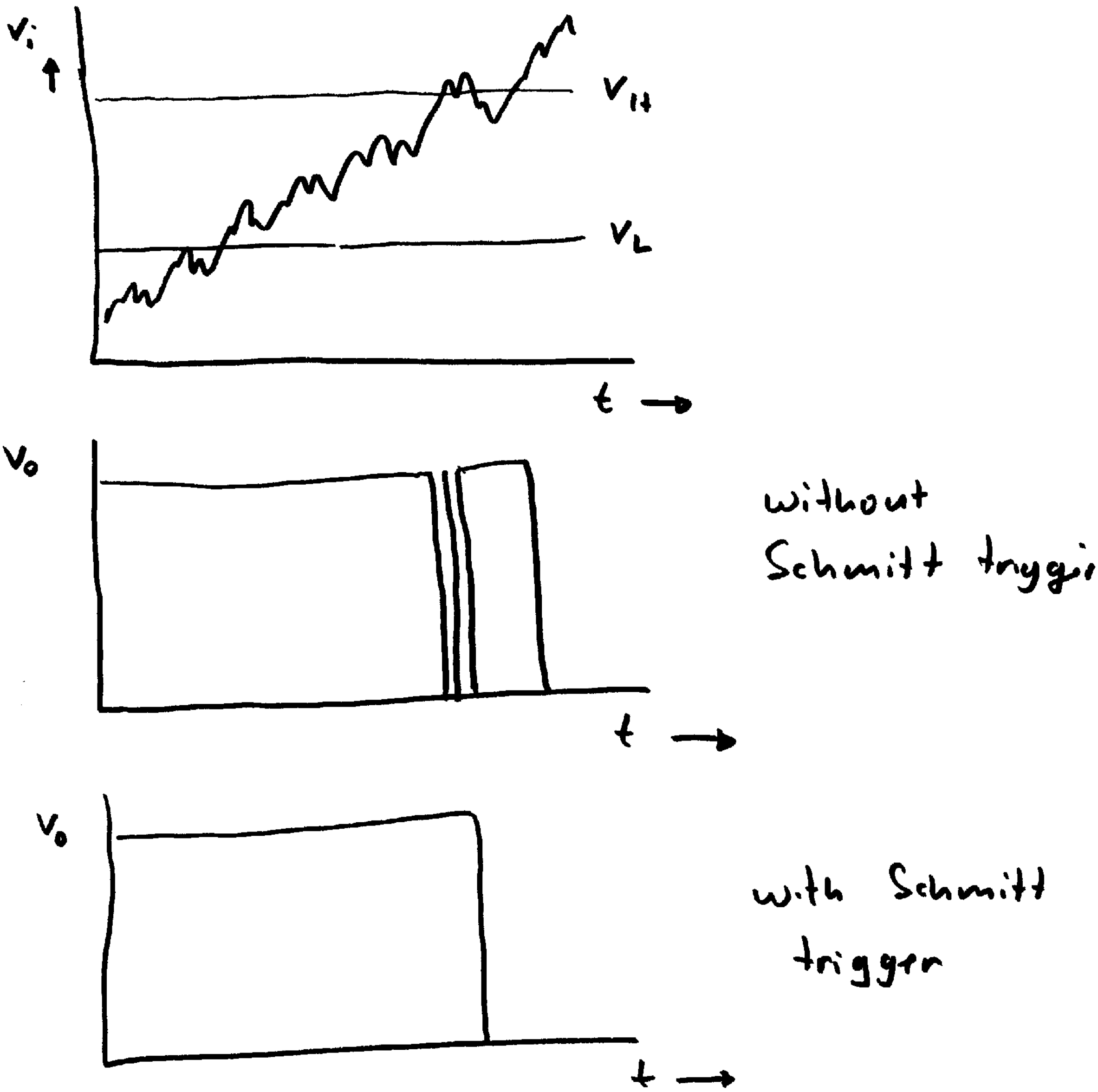
$$V_p = \frac{R_1}{R_1+R_2} V_0$$

If $V_0 = +V_{cc}$ then $V_p = +\frac{R}{R_1+R_2} V_{cc}$. Now if we increase V_i from $-\infty$ upwards, there comes a point where $V_i > V_p$, and thus, since the opamp works as a comparator, V_0 becomes $-V_{cc}$. This makes $V_p = -\frac{R_1}{R_1+R_2} V_{cc}$. To make switch again, we have to lower V_i to below this value. Thus, to summarize:

$$V_L = -\frac{R_1}{R_1+R_2} V_{cc}$$

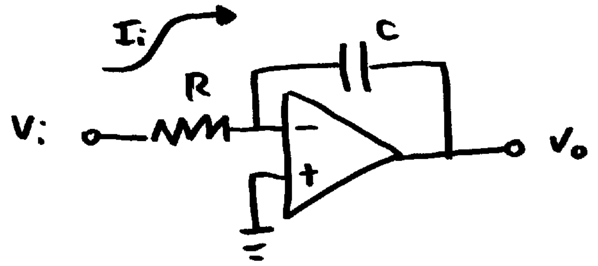
$$V_H = +\frac{R_1}{R_1+R_2} V_{cc}$$

The advantage of a Schmitt-trigger lies in the fact of eliminating noise



Imagine switching the light in a room based on a detector ....

Integrators can be made by placing a capacitor in the feedback loop



The resistor "translates" the voltage into a current, $I_i = V_i/R$. This current is forced through the capacitor C and is thus charged. The amount of charge in C is equal to the integrated current $(Q(t=0)=0)$

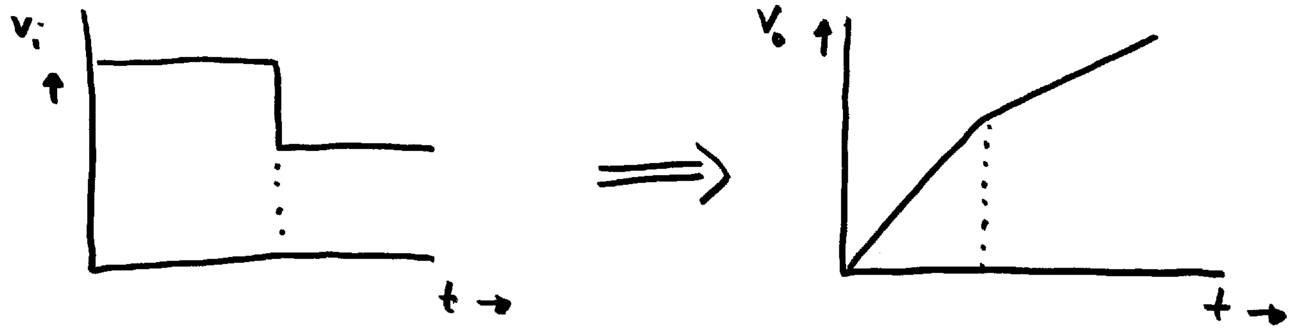
$$Q(t) = \int_0^t I_i(t')\,dt' = \int_0^t \frac{V_i(t)}{R}\,dt$$

The voltage drop induced by this charge is

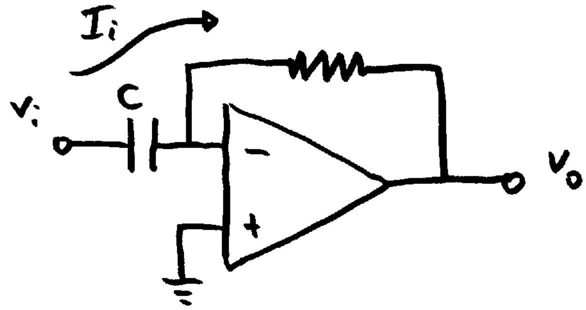
$$\Delta V_c = \frac{Q(t)}{C}$$

Since one side of the capacitor is connected to (virtual) ground, the voltage at the output is

$$V_o(t) = -\frac{Q(t)}{C} = -\frac{1}{CR}\int_0^t V_i(t)\,dt$$



A <u>differentiator</u> is made by exchanging the resistance and capacitor



The input current is equal to

$$I_i = C \cdot \frac{dv}{dt}$$

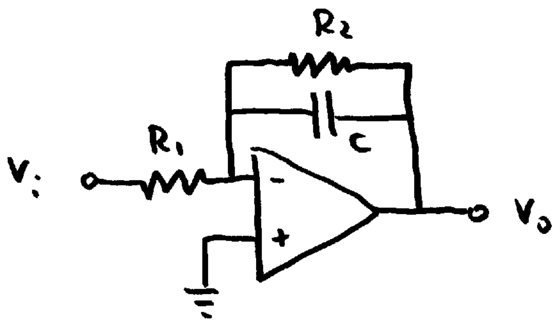
because of the differentiating effect of a capacitor. The current is translated into a voltage by the resistance R

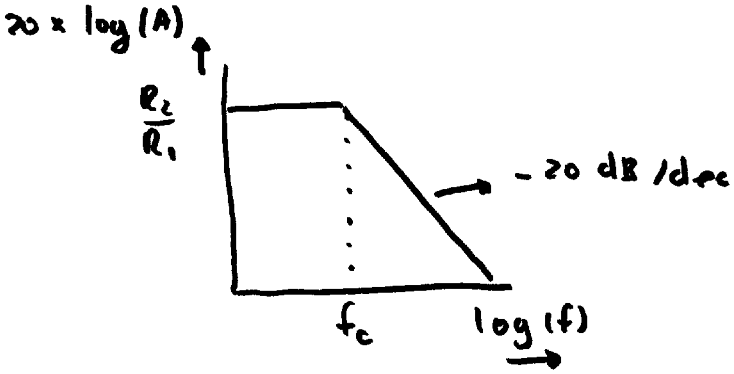
$$V_o(t) = -C \cdot \frac{dv}{dt} \cdot R$$

In both cases, the differentiator and integrator the signals at the output might be limited by the power supply. In the case of the integrator, it means that, for instance, DC signals at v_i can only be integrated up to a certain time. For the differentiator, it means that signals cannot change too fast

—//—

Low-pass filter

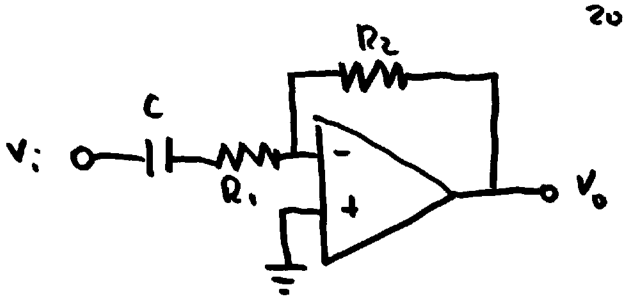


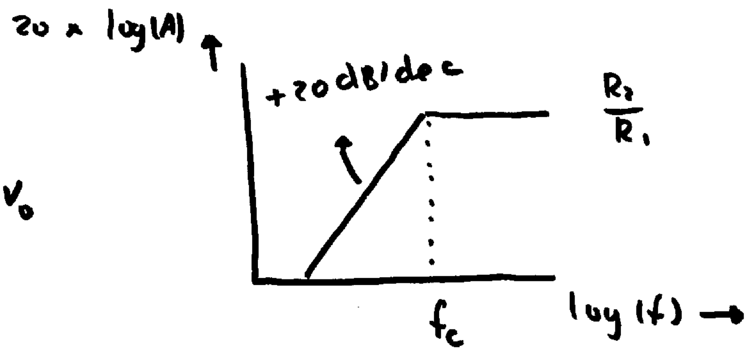


The cut-off frequency given by this active filter is

$$f_c = \frac{1}{2\pi R_2 C} \quad , \quad \frac{v_o}{v_i} = \frac{-R_2/R_1}{1 + f/f_c}$$

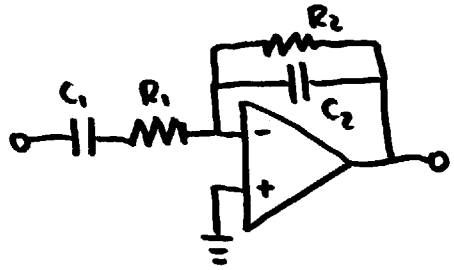
High pass filter

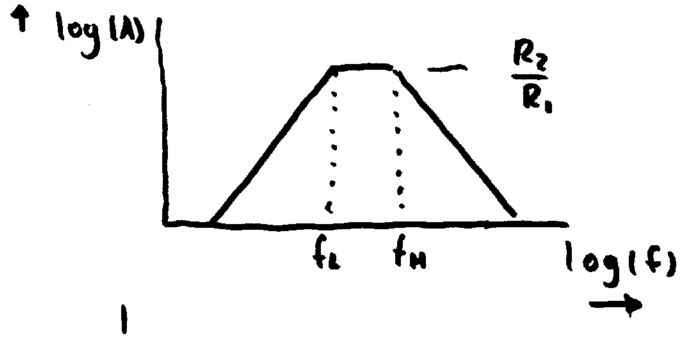




$$f_c = \frac{1}{2\pi R_1 C} \quad , \quad \frac{v_o}{v_i} = \frac{-R_2/R_1}{1 + f_c/f}$$

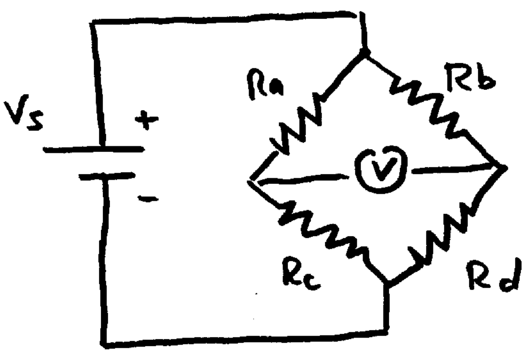
Band - pass filter





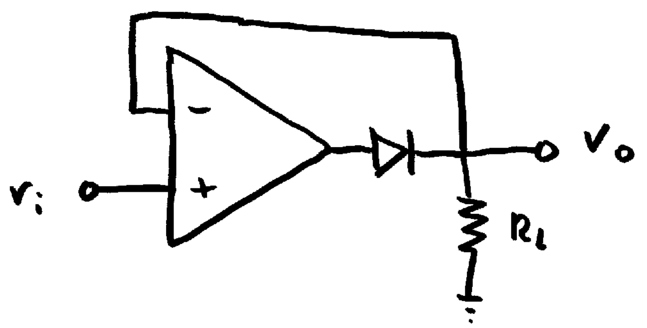
$$f_L = \frac{1}{2\pi C_1 R_1} \quad , \quad f_H = \frac{1}{2\pi C_2 R_2}$$

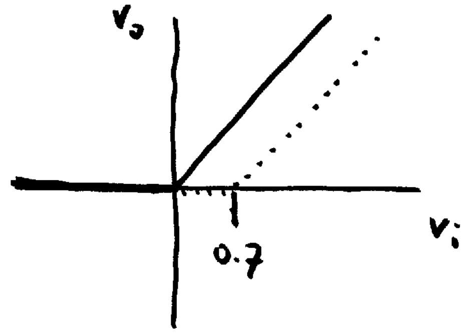
A wheatstone bridge is a small circuit for reducing the offset of the signal of a sensor (see introduction chapter). The standard consists of a voltage supply (V_s), 4 resistances (of which one is the sensor) and a voltmeter (v).



Imagine Ra is the sensor. If the other resistances (Rb, Rc and Rd) are all equal to Ra (at the calibration point), the output voltage is zero. When the sensor resistance changes, a voltage is measured that is, for small changes ($\delta R << R$) linearly proportional to the resistance change. Thus, when the relation change-in-resistance to change-in-measured-quantity (for instance temperature) is linear, the overal relation $V/\delta X$ is linear.

Super diode





This circuit is an ideal rectifyer.

$$V_o = V_i \quad \text{for} \quad V_i > 0$$
$$V_o = 0 \quad \text{for} \quad V_i < 0$$

and compared to a normal diode, doesn't have the switch-on effect of 0.7 V.

Instrumental operational amplifiers

Some manufacturers make special opamps for instrumentation. For instrumentation, the requirements are normally higher in terms of noise, offset, CMRR (common-mode rejection ratio), and power supply stability.

Noise

There are several ways of eliminating noise. The simplest one is by filtering it off, for instance with the filters described before, based on opamps. More advanced filters are the Butterworth, Bessel and Chebyshev filters, among others. (see H&H for an overview). Some types of noise can be eliminated at the source, others not.

Types of noise

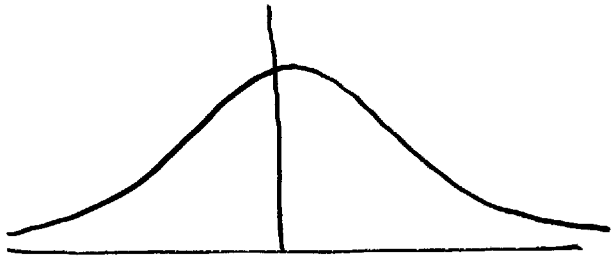
① Johnson noise: This is an open-circuit ($I=0$) voltage source that is unavoidable.

$$V_{rms} = \sqrt{4kTR\,\Delta f}$$

with: k = Boltzmann constant, T = temperature (K), R = resistance, Δf = bandwidth under consideration

To reduce it, the temperature can be reduced, or the bandwidth narrowed (filtering).

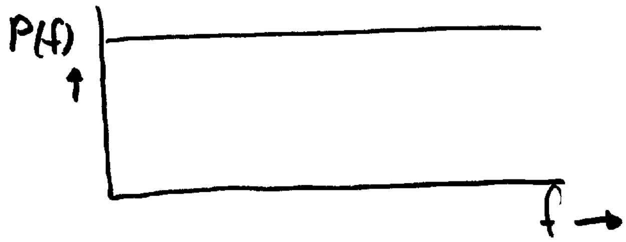
The amplitude of the noise is unpredictable (that is why it is noise), but has a Gaussion probability distribution



$$p(v) = \frac{1}{v_n\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{v}{v_n}\right)^2\right)$$

$$v_n = v_{rms}$$

This is a type of "white noise" meaning that the spectral density is constant



power spectrum

(2) <u>Shot noise</u> occurs when a current is present

$$I_{rms} = \sqrt{2qI\Delta f}$$

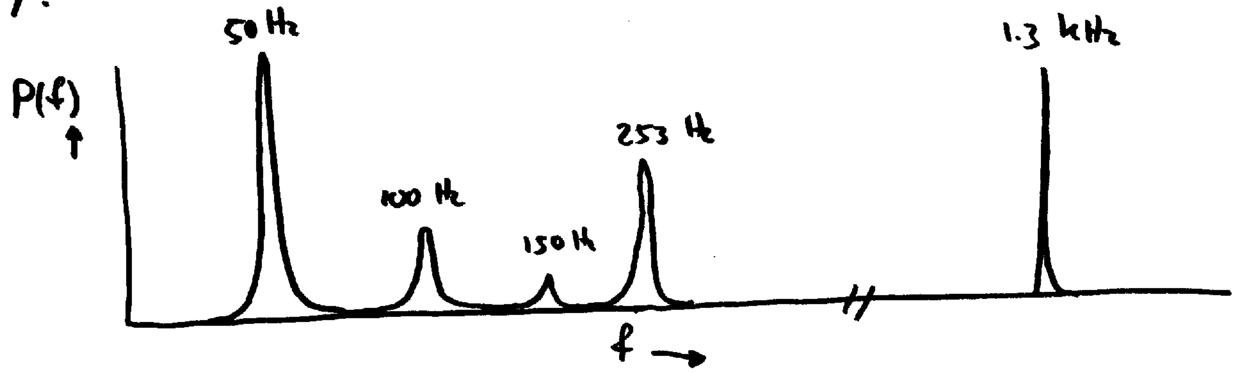
with q the elementary charge, I the DC current and Δf the bandwidth. This is also a form of white noise.

(3) <u>1/f noise</u> Many sources of noise have a 1/f spectral distribution. This means that

measuring at low frequencies is especially difficult since the signal

is normally obscured by noise.
whereas Johnson noise and shot noise are unavoidable, 1/f noise can normally be circumvented by increasing our measurement frequency (see Lock-in detector to be discussed later).

(4) <u>Interference</u>. These are all sources of "noise" that have their own origin. All unwanted parts of the spectrum caused by other equipment etc. Notorious is the 50 Hz "hum" of the electricity net. Avoid measurements at 50 Hz (and its harmonics!!). Other sources may include your neighbor scientist modulating his magnetic field at 1.3 kHz, a vacuum pump rotating at 253 Hz, a car or truck with idling engine (and vibrating our experimental setup).



(Example of interference noise spectrum)

Do not perform electrical measurements (of any frequency) when in the room there are blinking TL luminescent light bulbs switched on.

Signal - to - noise

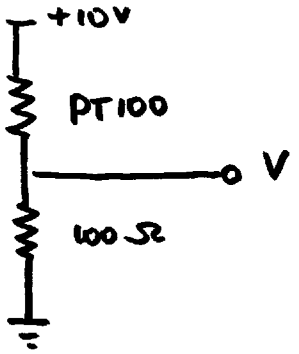
An engineer (and scientist alike) when designing a measurement set up is always talking about "the signal-to-noise ratio". This, obviously is the ratio of wanted information, the "signal" to the unwanted voltage, "noise".

$$SNR \text{ or } S/N = \sqrt{\frac{V_s^2}{V_n^2}}$$

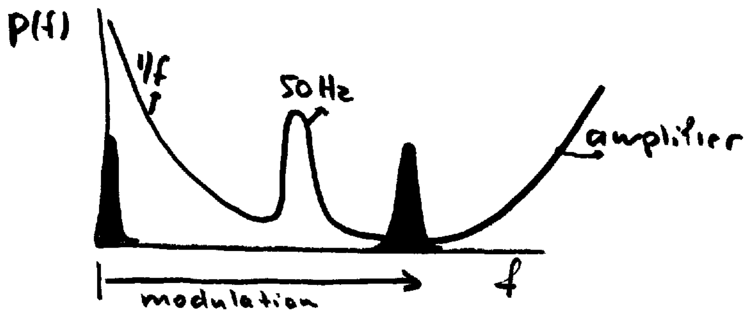
Generally speaking, a signal is measurable if the S/N is greater than 2. Optimization of a measurement system consists of increasing the signal and reducing the noise. The latter can consist of narrowing the bandwidth of a filter. This then limits the speed at which can be measured (the information of <u>changes</u> of the signal appear in the side bands of the spectrum and cannot be filtered off !!)

Lock-in detector

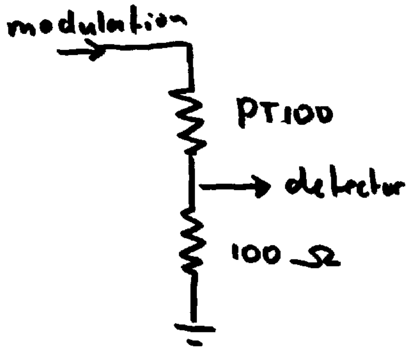
Imagine you want to measure the resistance of a PT 100 temperature sensor. The easiest way to do this would be to measure the voltage in a voltage devider, see Figure on the left.

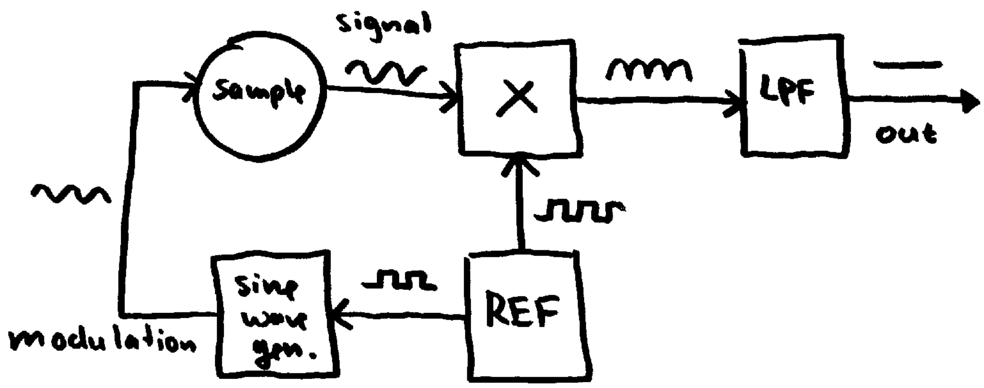


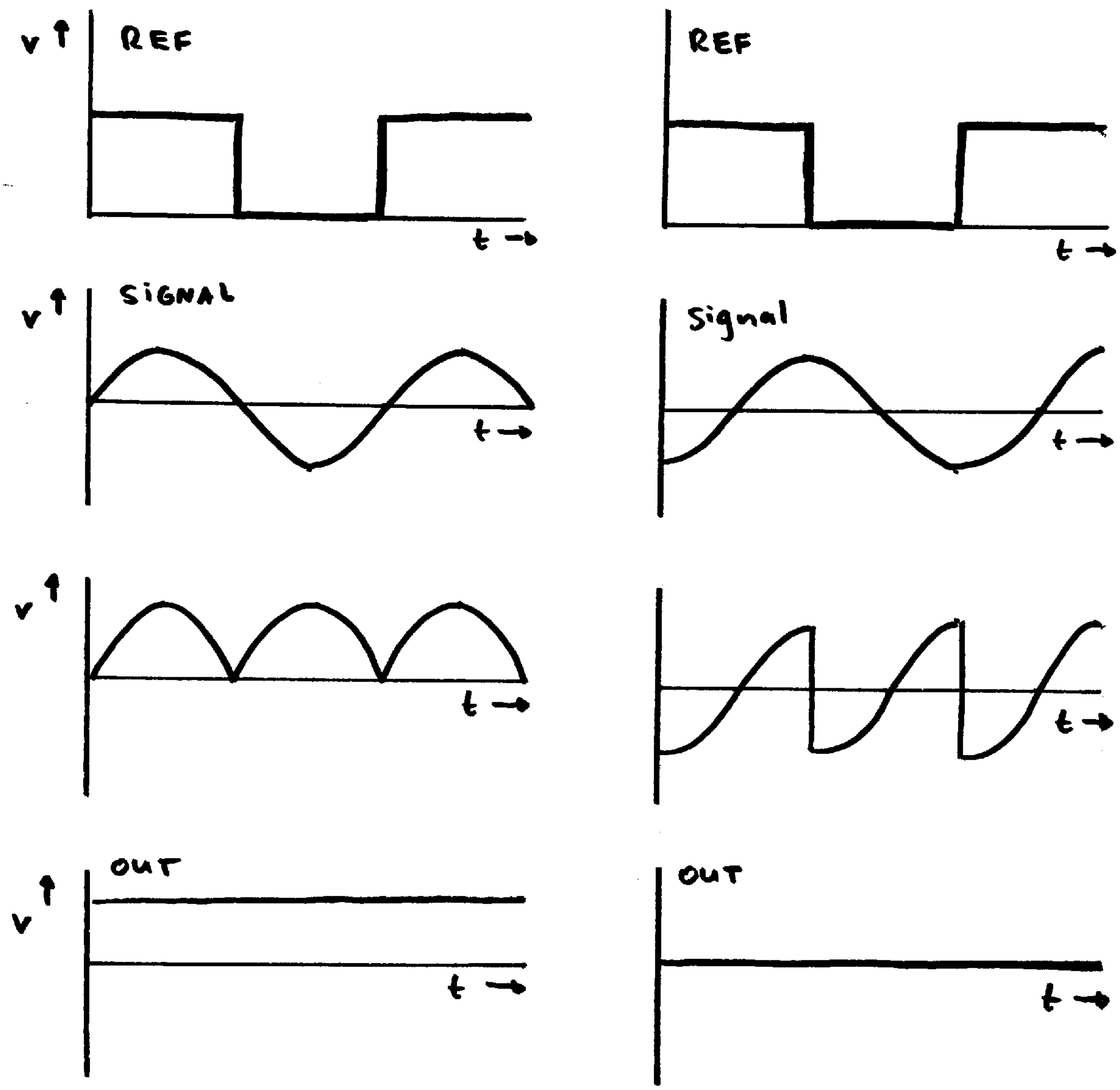
However, as discussed before, because of 1/f noise, this type of measurement is very noisy. It would be better to measure the resistance in a part of the noise spectrum with low density



The way to do this is with modulation techniques. Instead of using a DC probing voltage (+10 V in figure), we use an AC signal, for instance 1 KHz. Then, and this is the crucial part, we only analyze that part of the signal that has exactly that frequency and is in phase with the modulation signal. This can be done with a lock-in detector / amplifier. The figures below show how this works.







The reference (REF) is used, after converting to a sine-wave, to modulate the signal. The modulated signal is normally filtered with a bandpass filter (not shown) and then multiplied by the square-wave reference. For an in-phase signal this results in perfect rectification of the signal. This signal is then filtered with a low pass filter which leaves only the DC component. The left side of the figure shows the various stages for an in-phase signal. The right side shows the same for an 90° out-of-phase signal which results in zero volt at the output (OUT)

$$\text{signal} = A \sin(\omega t)$$

$$\text{REF} = \frac{4}{\pi} \sum_{n=1,3,5\ldots}^{\infty} \frac{1}{n} \sin(n\omega t)$$

$$\text{SXR} = \frac{4A}{\pi} \sum_{n=1,3,5}^{\infty} \frac{1}{n} \sin(n\omega t) \sin(\omega t)$$

$$= \frac{4A}{\pi} \sum_{n=1,3,5}^{\infty} \frac{1}{2n} \Big[\underbrace{\cos((n-1)\omega t)}_{n=1} + \cos((n+1)\omega t) \Big]$$

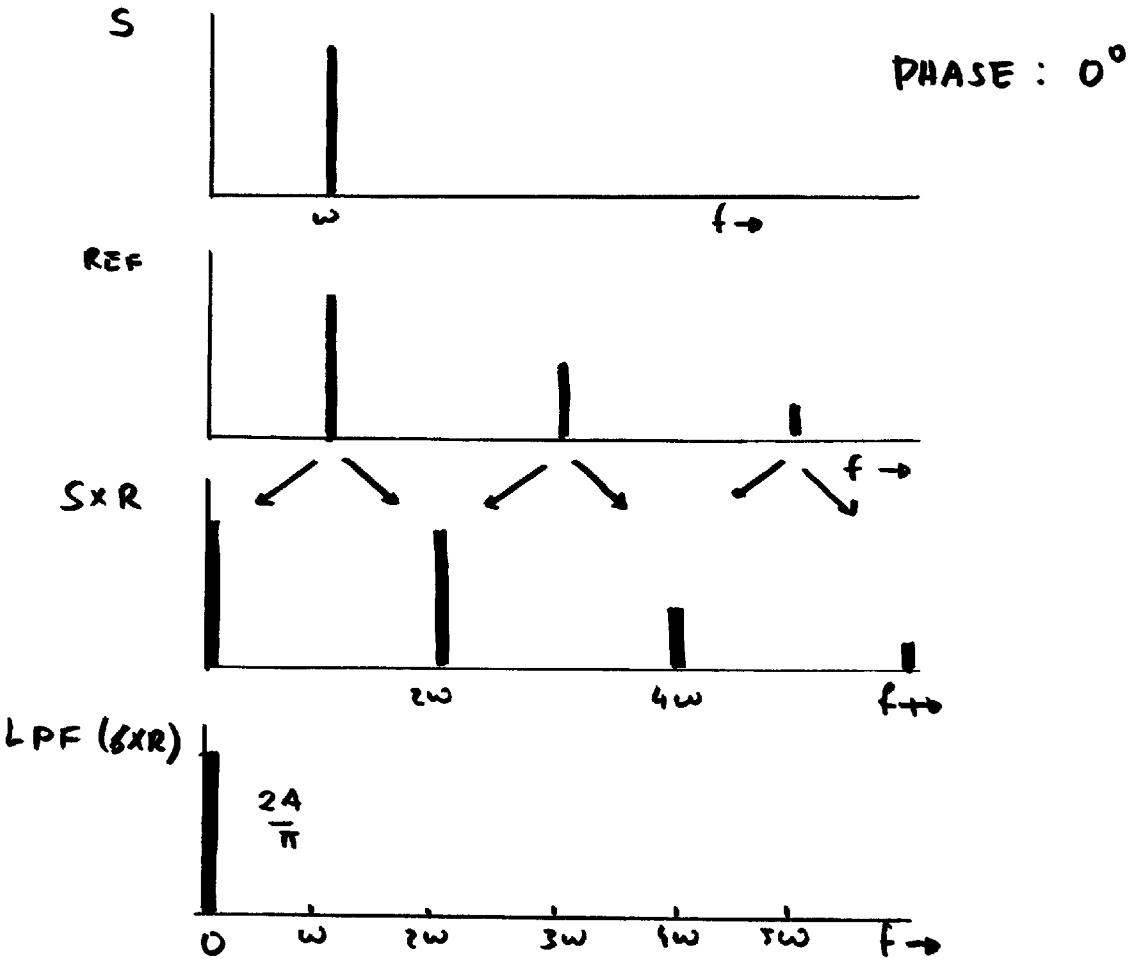
$$\text{LPF}(\text{SXR}) = \frac{4A}{\pi} \cdot \frac{1}{2} \Big(+ \cos(2\omega t) + \ldots \Big)$$

(all other components filtered off!)

only $n=1$

$$\sin a \sin b = \frac{1}{2} \big(\cos(a-b) - \cos(a+b) \big)$$

The final signal is directly proportional to amplitude (A) of signal



For 90° out of phase signal :

$$\text{Signal} = A\cos(\omega t)$$

$$\text{REF} = \frac{4}{\pi}\sum_{n=1,3,5\ldots}^{\infty}\frac{1}{n}\sin(n\omega t)$$

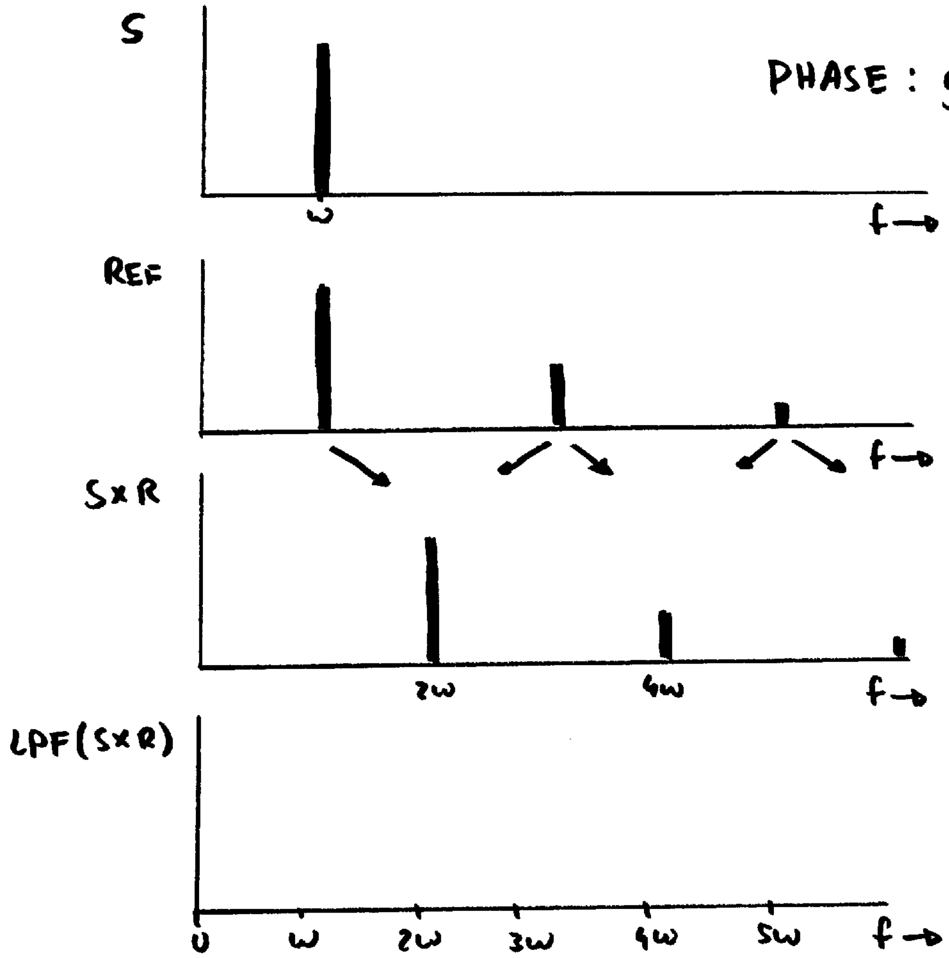
$$S \times R = \frac{4A}{\pi}\sum_{n=1,3,5}^{\infty}\frac{1}{n}\cos(\omega t)\sin(n\omega t)$$

$$= \frac{4A}{2\pi}\sum_{n=1,3\ldots}^{\infty}\frac{1}{n}\left[\sin((n-1)\omega t) + \sin((n+1)\omega t)\right]$$

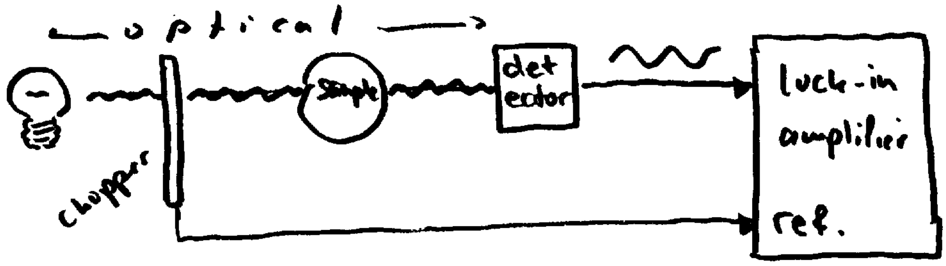
$$= 0 + x\sin(2\omega t) + \ldots$$

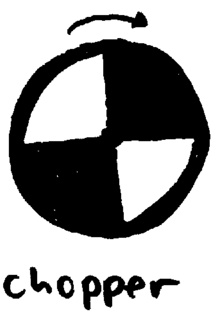
$$LPF(S \times R) = 0$$

$$\sin a \cos b = \frac{1}{2}\left[\sin(a+b) + \sin(a-b)\right]$$



Modulation doesn't necessarily have to be done electrically. Very common is the use of a "chopper" in optical measurements. A chopper is a





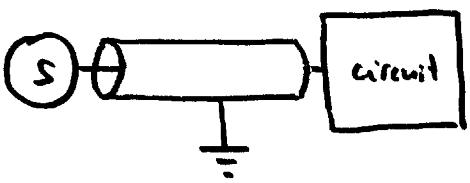
rotating disk with holes. This will modulate the electrical signal of the detector.

Noise and cables

Noise often enters the signal via the cables. They can work as antennas for all the electromagnetic pollution in the air. This is especially important for radio-frequency noise (MHz - GHz)

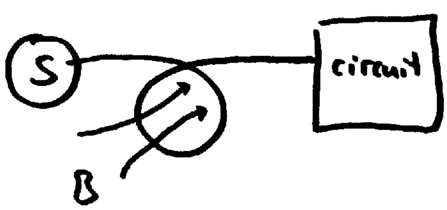
← signal cable catching EM noise

The easiest way to prevent this is to build a Faraday cage around the cable. This shields the electric part of the electromagnetic waves.



← signal cable shielded (COAX)

The standard shielded cable is coax (BNC). It however does not shield the magnetic part of the noise. This kind of radiation is especially important when the cables form loops (but not only).



The flux of a loop

$\phi = A \cdot B$, with

A the area and B the magnetic

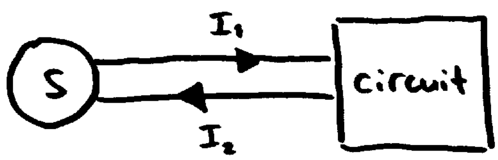
field. When the flux is changing, this induces a current in the cable

$$I \sim \frac{d\phi}{dt}$$

This can be either a change of magnetic field, for instance $B = B_0 \sin \omega t$, or a change of area (moving the cables)

$$I \sim A \cdot \frac{dB}{dt} + B \frac{dA}{dt}$$

A simple sollution is to use two wires (twisted pair). The circuit analyses the <u>difference</u>



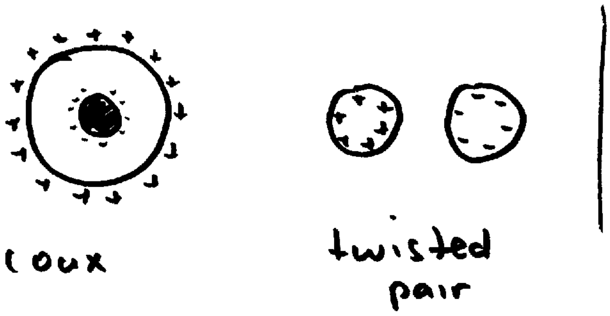
← twisted pair cables

in current. One wire has positive current. The other wire has exactly opposite current. Any noise caught by one wire is also picked up by the other cable and the difference stays the same:

$$\left.\begin{array}{l} I_1 = I_s + \delta I \\ I_2 = -I_s + \delta I \end{array}\right\} \quad I_1 - I_2 = 2 I_s$$

no noise!

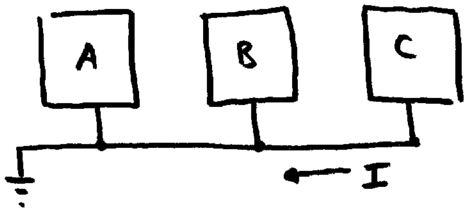
Both types of cable (coax and twisted pair) have inherent capacitance. A 50 Ω coax cable has about 100 pF/m capacitance, while a twisted pair has about 60 pF/m. This is caused by the fact that any two metallic objects can store charge and work as capacitors and is unavoidable. It can only be reduced by increasing the distance between the objects. That is also why twisted pair cables work better (have less capacitance), because the distance between the two poles is larger. When this capacitance is a problem, single-pole wires should be used, at the cost of increased noise.



Earth

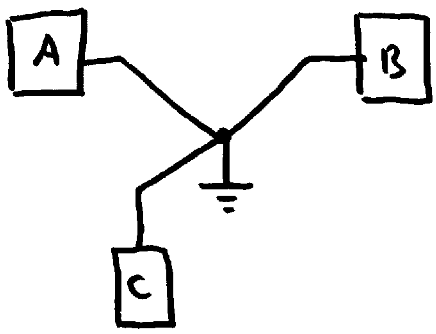
Some types of noise and offset are caused by the wrong grounding of the equipment.
The cable/wire used for grounding ideally has zero resistance, but in reality has some

resistance. when there is current flowing to ground, this can cause that different parts of the circuit or different units see a different ground, when the equipment ground is connected "in serie".



← The ground that 'A' feels can be different from the one that 'B' or 'C' sees.

To avoid these problems, the equipment should be ground in a central place. All equipment feels the same ground. No noise of one unit will lift the ground of the other and introduce noise there

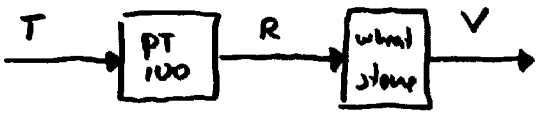


← "Star" grounding

(obviously, this ground then has to be of very low resistance, otherwise we just delayed the problem)

2 Signal generation "Sensors & Actuators"

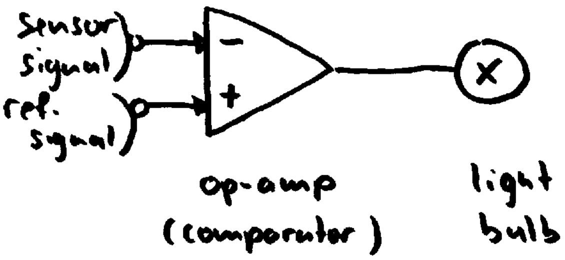
Transducers are elements that translate a signal ("information") from one domain to another. An example is a PT100 temperature sensor that "translates" (or "maps") the physical quantity "temperature" to another physical quantity, namely "resistance". This then in turn can easily be translated into a more processable quantity "voltage" by placement in a wheatstone bridge, as discussed in chapter 1.



examples of transducers.

Normally, when talking about transducers, one thinks about transducers from the physical domains (temperature, distance, strain, etc.) to the electrical domain (voltage, current). These are called sensors. However, the opposite also exist, namely transducers that translate electrical signal into a physical quantity. Examples are the closing of a door, a traffic light, etc. It is a little funny to see a light bulb as a transducer, but that is exactly what it is. These "output" transducers are normally called actuators.

A common problem in the design of systems with actuators is that the electronics of signals is normally low power while the final element (light bulb, engine, air coditioning, etc.) is high power. An example is a warning light that must switch on when the temperature is too high.

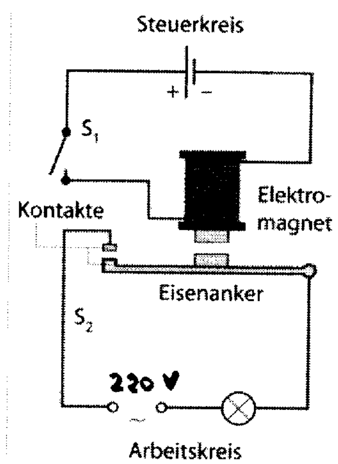


← Bad design

We could use a comparator op amp (see ch. 1) and connect a lightbulb at the output. The lightbulb should switch on when the sensor signal becomes larger than the reference signal. The problem is that the maximum driving power of an opamp is some mW at best, while a lightbulb is easily some tens of Watts.

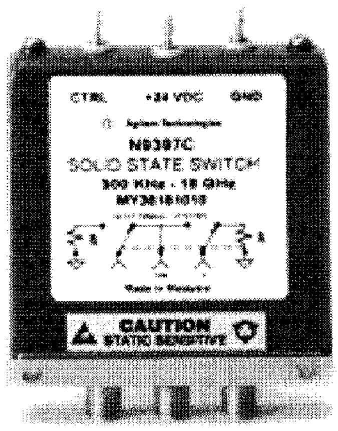
A solution is to use a <u>relais</u>. This is an element that can switch a large current with the use of a small current. The small current goes through a coil and thus creates a strong magnetic field. This magnetic field attracts a small magnet connected to a mechanical switch. This closes

http://www.laurentianum.de/physikmuseum/relais.htm



a secondary circuit that can supply large currents. See the picture on the left. The "arbeitskreis" (power circuit) can be powered by 220 V, enough to switch on the lightbulb. The control circuit (Steuerkreis) has small current which can easily be supplied by an op amp. This kind of elements can be used for very large currents. and can be recognized in many systems by it's characteristic "clicking" sound when switching (for example the indicators in a car 

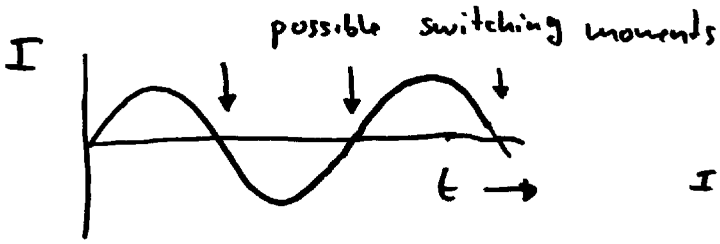
More modern is the solid-state switch,



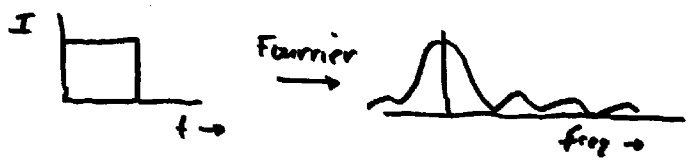
http://www.agilent.com

which has the advantage of having no mechanical elements: when currents are interrupted (not when they are started!), large sparks can occur because currents cannot change instantaneously. The currents continue, and combined with the open circuit (high resistance!), a large power is dissipated $P = I^2 R$. This causes a spark. This continues until all energy of the current is dissipated (remember a static current causes a magnetic field with associated energy). The sparks cause oxidation of the contacts and eventually it's failure.

A solid state switch suffers less from this problem. Moreover, the good ones are intelligent and can switch when the current is zero (in AC power supply)



It also avoids emitting in all spectrum when switching abruptly current



Otherwise, the solid-state switch works effectively like a relais.

Temperature sensors

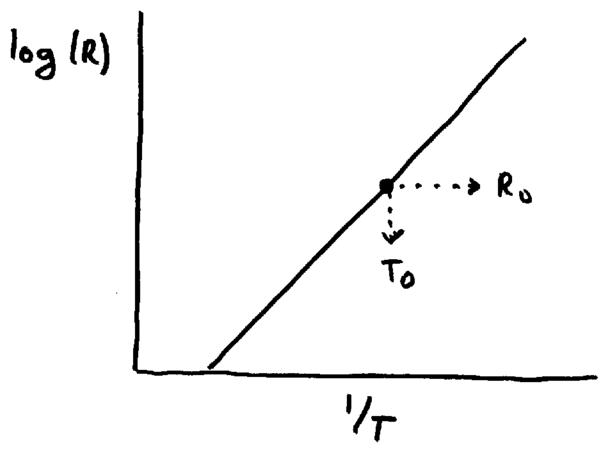
Temperature sensors are based on the fact that an electronic parameter (R, V, I) depends on the temperature.

Thermistor. A thermistor is a resistance whose value depends on temperature. They are very cheap, but the relation T-R is complicated, for instance

$$R = R_0 \exp\left(\beta \left(\frac{1}{T} - \frac{1}{T_0}\right)\right)$$

Note the negative temperature coefficient (NTC): increments in temperature reduce the resistance. This is because they are made of semiconductors

$$\alpha \equiv \frac{dR}{dT} = -R\, \beta/T^2$$



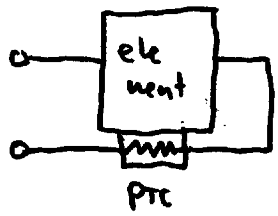
Typical values:

$R_0 = 1\,k\Omega$

$\alpha = -45\ m\Omega/K$

$T = -60°C - +150°C$

NTC thermistors are good for measuring the temperature, but not for controlling it. If we want to protect an element from overheating we should place a PTC in series with it. When the temperature increases, the PTC increases in resistance and reduces the current. With an NTC, on the other hand, we would have a runaway system ("positive feedback") where the current and the temperature keep increasing.



PT 100 A very good and accurate temperature sensor is a PT 100 resistor, named after the fact that it is made of platinum and has a calibrated 100 Ω resistance at 0 °C. At other temperatures:

$$R(t) = 100 \cdot (1 + \alpha T + \beta T^2 + \gamma T^3)$$

(T in °C)

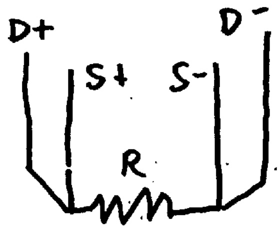
example: $\alpha = 3.9 \; 10^{-3} \, / °C$

$\beta = -5.8 \; 10^{-7} \, / (°C)^2$

$\gamma = -4.2 \; 10^{-12} \, / (°C)^3$

Thus, in first order, $\frac{dR}{dT} = 0.39 \; \Omega / °C$

Because of it's low resistance, in measing the resistance of the lead cables becomes important. To avoid this, normally 4-point measurement techniques are used

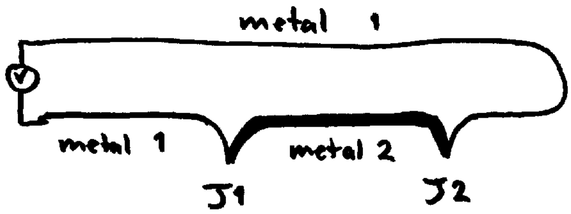


The "drive" (D+ and D-) supplies a current. The voltage accross the sample is measured with "sense" (S+ and S-). These wires do not carry current and not voltage drop occurs in those wires. Then

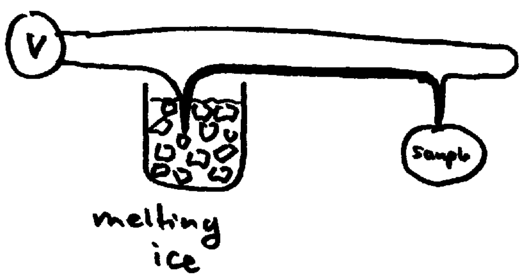
$$R = \frac{V_{S+} - V_{S-}}{I_{D+ \to D-}}$$

Thermo couples

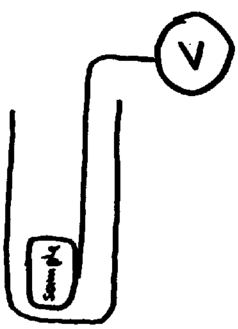
A thermo couple is a transducer that translates temperature directly into voltage. It consists of



a wire of two types of metal. Because these metals have different electron affinity, at the junctions of the metals a small voltage drop is present. This drop depends on the temperature. When the entire wire is at the same temperature, the drop ΔV at one junction J1 is compensated by the drop $-\Delta V$ at the symmetric junction J2 and the total external measured voltage is zero. However, when the two junctions are at <u>different</u> temperature, the offset at one junction ΔV_1 is not cancelled by the second junction and an external voltage results. The idea is to put one of the junctions at a well defined temperature (for instance the triple point of water). the second junction is then connected to the object to measure.



There exist many types of thermocouples, pairs of metal wires. Each has its own application (temperature range, etc.). A serious problem of thermocouples is that they themselves introduce a heat sink or source. Imagine measuring a sample at 4 k (-269°C). One end of the thermocouple cable is fixed mechanically at the sample at 4 k with obviously good thermal contact. The other side is at room temperature (the voltmeter). The



wire is made of metal and metals do not only conduct electrically well, but also thermally. The large temperature gradient then causes a large heat input and often easily increase the temperature some degrees. To avoid this we have to use extremely thin lead wires. This works well for a PT100 resistance (with 4-point measurement technique) and diodes.

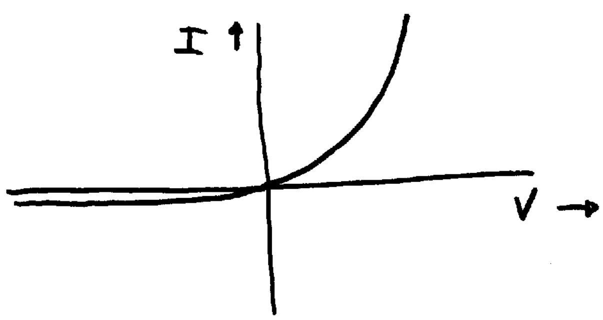
Modern temperature systems based on thermocouples often have built-in temperature compensation, meaning that it can correct for the fact that the reference junction is at a different, not calibrated temperature (for instance room temperature). This then eliminates the need for a temperature reference, making the system simpler and cheaper.

Use type K, unless you have a good reason not to

Diode

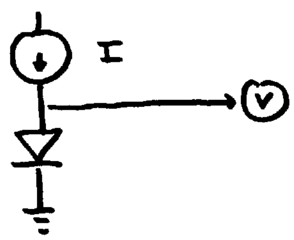
As we know from the first electronics lesson, a diode current is equal to

$$I = I_0 \cdot \left[\exp\left(\frac{qV}{kT}\right) - 1\right]$$



with I_0 a constant depending on the type of diode. (size, material, etc.)

Thus, the combination of V and I tells us the temperature.



Normally, the circuit shown on the left is used. A constant current I is forced through the diode. The voltage measured then gives us the temperature (ignoring the term "-1"):

$$T = \frac{qV}{k\ \ln(I/I_0)}$$

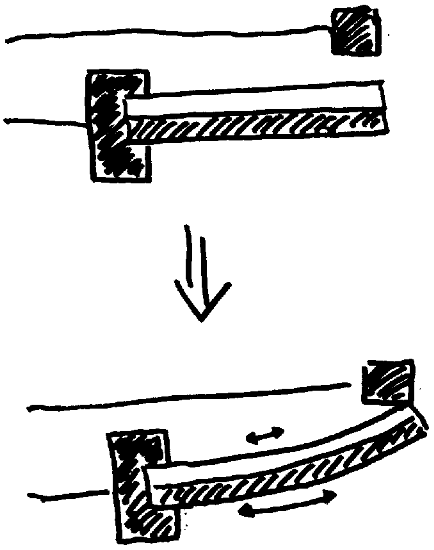
A popular diode sensor is the LM35 which is an integrated circuit that includes the current source and everything. A "plug and play" sensor with 10.0 mV/°C, and 0 V at 0°C.

This is a good and cheap and easy to use sensor for non-scientific purposes.

Non-electronic temperature sensors

Not always is exact knowledge of the temperature needed. Not always do we want to design a complicated and expensive circuit to determine the temperature. Not always is contact with the sample allowed or possible.

A very simple mechanical switch is the bimetal. This is a small stick that consists of two layers of different metals. When the stick heats, the metals expand, but because they are diffrent metals, they expand at a different rate. Because of this, the stick bends, until, for a certain temperature, it bends so far as to make contact with a metallic pad and closes the circuit.



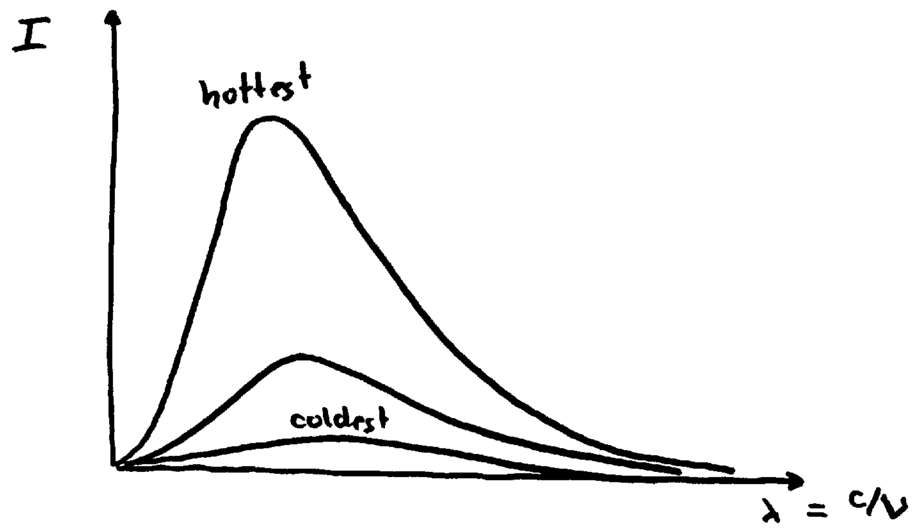
Thus, the result is a binary sensor, with two states, $T < T_x$ or $T > T_x$, with T_x the set-point temperature. The sensor is not very occurate, but often accuracy is not needed. Imagine the bimetal placed in a flame. We just want to know if the flame is on or not. If the flame is off, the gas flow has to be cut to avoid filling the room with unburnt gas.

Another non-electronic sensor is based on the principle that hot bodies emit light and the color of the radiation is determind by the temperature.

infrared
red
blue
ultra violet

hotter ↓ ↑ colder

Just by looking at an object, we can estimate it's temperature. Ex: Lava is aproximately 1000°-1200°C and therefore glows red.



Planck law of black body radiation:

$$I(\nu) = \frac{2h}{c^2} \cdot \nu^3 \cdot \frac{1}{\exp\left(\frac{h\nu}{kT}\right) - 1}$$

h = Planck's constant $h = 6.6 \times 10^{-34}$ J/s
c = speed of light $c = 3.0 \; 10^{8}$ m/s
k = Boltzmann constant $k = 1.38 \; 10^{-23}$ J/K
ν = frequency ($\nu = c/\lambda$, with λ wavelength)

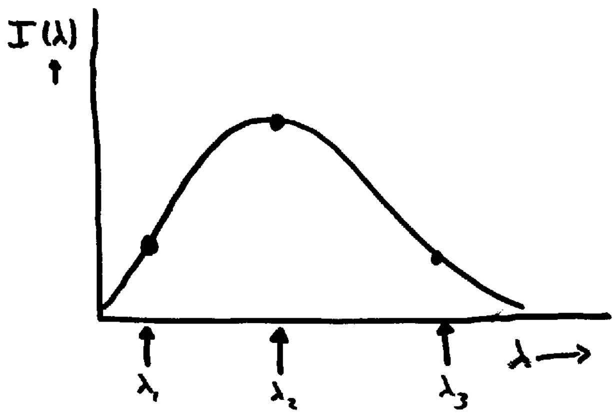
The maximum of radiation:

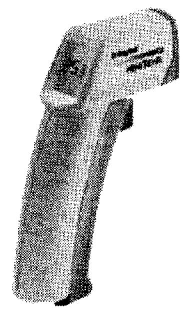
$$\frac{dI(\nu)}{d\nu} = 0 \Rightarrow \boxed{\lambda_{max} = \frac{b}{T}}$$ Wien law

$b = 2.898 \times 10^{-3}$ m·K (Wien constant)

T = temperature (kelvin!)

To measure the temperature, the intensity is measured at three fixed wavelengths, from





"temperatur gun"

which the black-body radiation curve can be reconstructed and the temperature estimated.

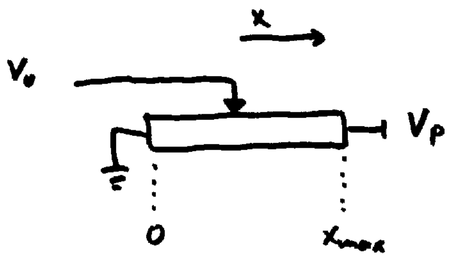
The advantage is that it is contactless (see the "temperature gun" figure above) and works at a distance. It can even be done for remote sensing the temperature of the seas (with correction because a sea is not a black body).

remote sensing the earth

Displacement Sensors

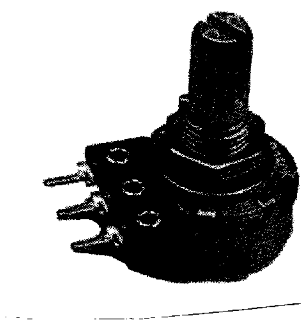
These are sensors to measure position is space.

The simplest one is a potentiometer, shown here schematically on the left. The output voltage depends linearly on the distance x.



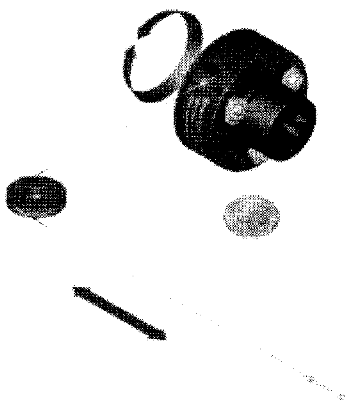
$$V_0 = \frac{x - x_{max}}{x_{max}} \cdot V_p$$

Similar to this are the rotational potentiometers



They translate "angle" into resistance. The principle of operation is the same.

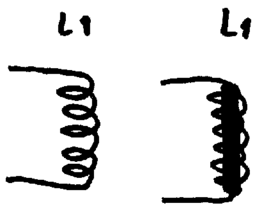
Some models exist that use rotational potentiometers to measure translational movement (see image on the left).



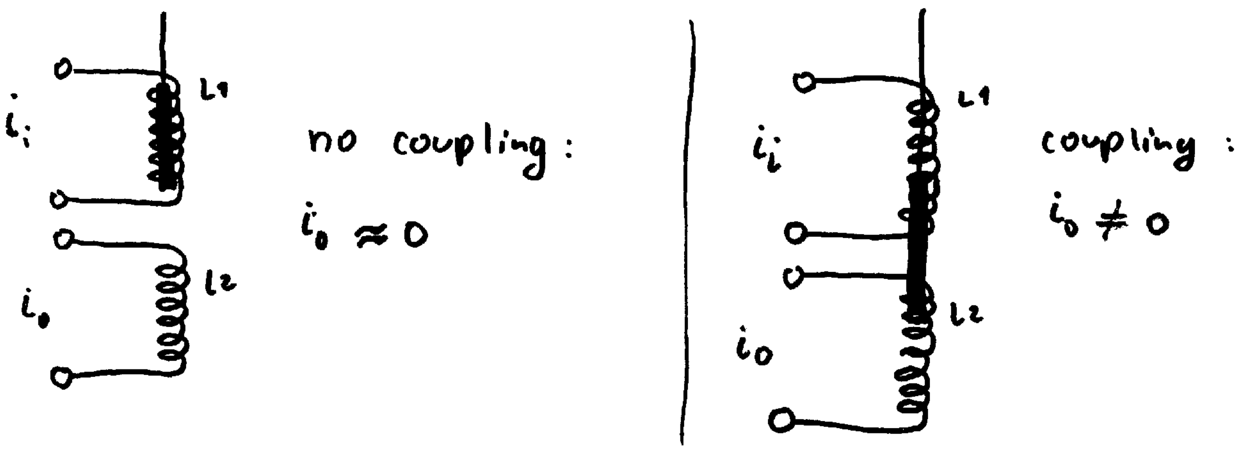
Since they are all based on mechanical contact, they suffer from contact problems (noise). Contactless sensors are preferred.

A more advanced, contactless, distance sensor, is the LDT (linear displacement transducer) working on the principle of inductance (compared to the resistance shown above).

It is based on the effect that the inductance of a coil is increased by the introduction of a ferromagnetic cilynder in it's center, or

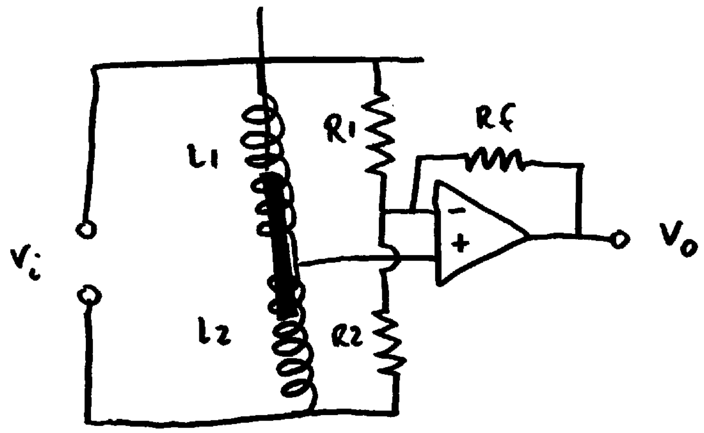


the coupling between two coils is increased by such ferromagnetic bars present.



The first coil is driven with a sine wave. When the bar is completely inside one of the two coils (for example top one, as shown on the left), the coupling is minimal. No signal is observed at the exit, $i_o \approx 0$. When the bar is inserted, the coupling is strong. Currents in the primary coil, L1, create an oscillating magnetic field which is efficiently guided by the ferromagnetic bar into the secondary coil, L2, where they create an oscillating current, io, observable outside.

A similar setup is shown here below



The two coils are used in a voltage divider

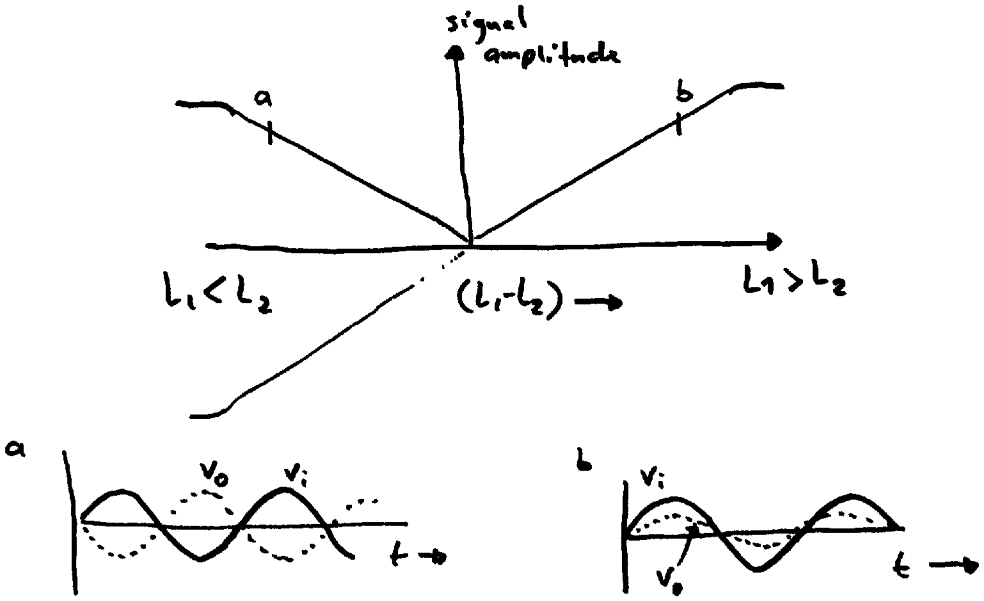
$$V_+ = \frac{L_2}{L_2+L_1} \cdot V_i$$

Another voltage divider is based on resistors

$$V_- = \frac{R_2}{R_2+R_1} \cdot V_i \qquad \left(R_1 = R_2 : V_- = \tfrac{1}{2} V_i\right)$$

These two signals are fed into an opamp in a differential amplifier configuration.

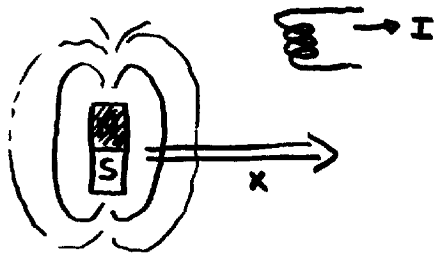
when the ferromagnetic bar is in the center the two coils have equal inductance; $L_1 = L_2$ and $V_+ = \frac{1}{2}V_i$. The output of the op-amp is then 0. When the bar is move one way or the other, L_1 and L_2 are mismatched and the output is no longer 0.

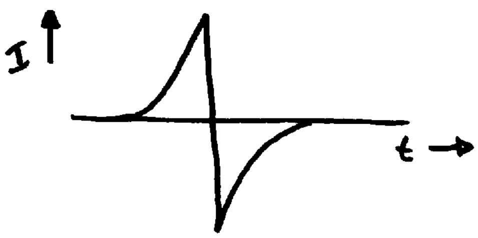


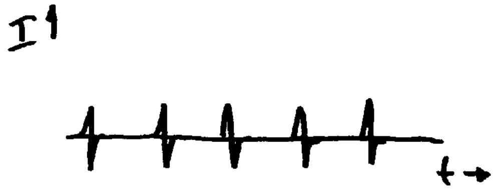
This sensor is very linear and has the advantage of being contactless (the ferromagnetic bar does not touch mechanically the coils). Its only effect is adding mass to the object measured. Contactless measurements have preference because of this reduced interaction <u>and</u> because of them being more stable over time.

RPM (rotations per minute)

Maxwell's equations tell us that current causes a magnetic field ($\nabla \times H = J$) and changing magnetic field causes electric field ($\nabla \times E = -\frac{dB}{dt}$) and thus current. We can make use of this second effect for a movement sensor. More precisely, for a sensor that detects the passage of an object. We connect a magnet to the object and place a coil where we want the passage of the object to be detected. With a multimeter we measure the current of the coil. When the magnet is <u>moving</u> past the coil, the time derivative of the magnetic field is large and the current large as well. Exactly when the magnet is aligned with the coil, the current inverts.

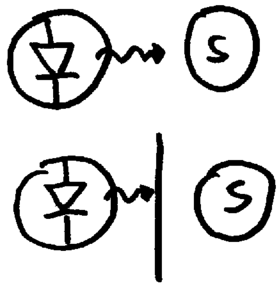






An RPM sensor can be made by counting the number of spikes per minute. A hard disk has this sensor where it is also used for synchronisation to detect when a sector of the disk starts.

Similar to this is an optical binary detector (on/off), that consists of an LED (or laser) combined with an optical sensor. The presence or interruption



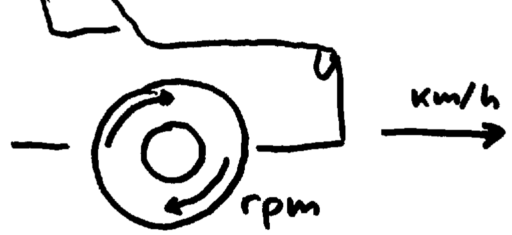
of light on the sensor detects the presence or absence of an object in the light path.

Famous is the "write protect" slide on floppy disks.

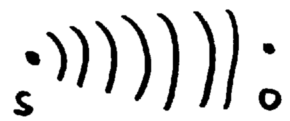
(The sensor (S) for light detection is discussed seperately later)

Speed

A simple system for measuring speed is to measure the RPM of a wheel connected to it and translate this to km/h based on the radius of the wheel. Note that the instrument has to be recalibrated when different wheels are connected and tires wearing out can make the car seem to go faster



A contactless way of measuring speed is by using the Doppler effect.



The frequency of the waves emitted by the source, f_s is received by an object (o) with a different frequency f_o that depends on the velocity of the source and the object

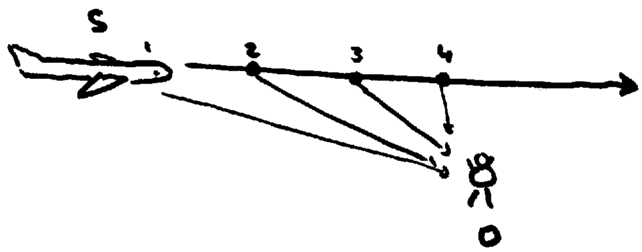
$$f_o = \left(\frac{v + v_o}{v - v_s} \right) f_s \qquad \text{(Doppler)}$$

where v is the velocity of the waves (300 m/s for soundwaves, $3 \cdot 10^8$ m/s for electromagnetic waves) and v_o and v_s are the velocity of the receiving object and the emitting source respectively.

Spectacular examples are when the source is moving with the speed of propagation of the waves, $v_s = v$ in which case the above

Doppler equation is a singularity, $f_o = \infty$

This effect occurs when, for instance, an airplane moves with the speed of sound and a sonic boom occurs. All soundwaves from



all points arrive at the same time at the observer. 1, 2, 3, 4 ...

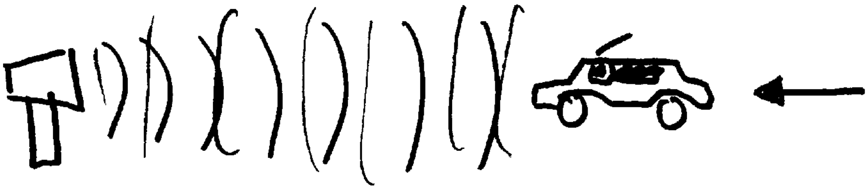
For a <u>static</u> source, $v_s = 0$ and

$$f_o = \left(\frac{v + v_o}{v} \right) \cdot f_s$$

In case this is reflected back to the source we can again use Doppler's equation (now with the source in movement and the observer static). What comes back is:

$$f_s' = \left(\frac{v}{v - v_o} \right) \cdot \left(\frac{v + v_o}{v} \right) \cdot f_s$$

$$= \left(\frac{v + v_o}{v - v_o} \right) f_s$$



Taylor expansion:

$$f_s' = \left(1 + 2\frac{v_0}{v} + \mathcal{O}\left(\frac{v_0^2}{v^2}\right)\right) f_s$$

For $v_0 << v$ (the object moving much slower than the wave velocity).

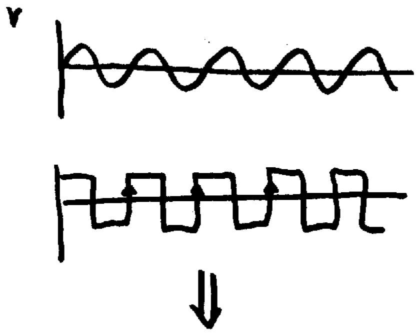
$$\Delta f_s = f_s' - f_s \approx 2\frac{v_0}{v}$$

Thus, the difference frequency is directly proportional to the object velocity v_0. For police radar measurement, the used wave is radar (GHz electromagnetic waves) and $v = c = 3 \cdot 10^8$ m/s. Even for a Formula 1 car at top speed $v_0 << v$.
The delayed question is: how to measure frequency?!
The first step is normally to multiply the returning signal with the source signal

$$f_s \times f_s' \longrightarrow \Delta f_s + (f_s + f_s') \ldots$$

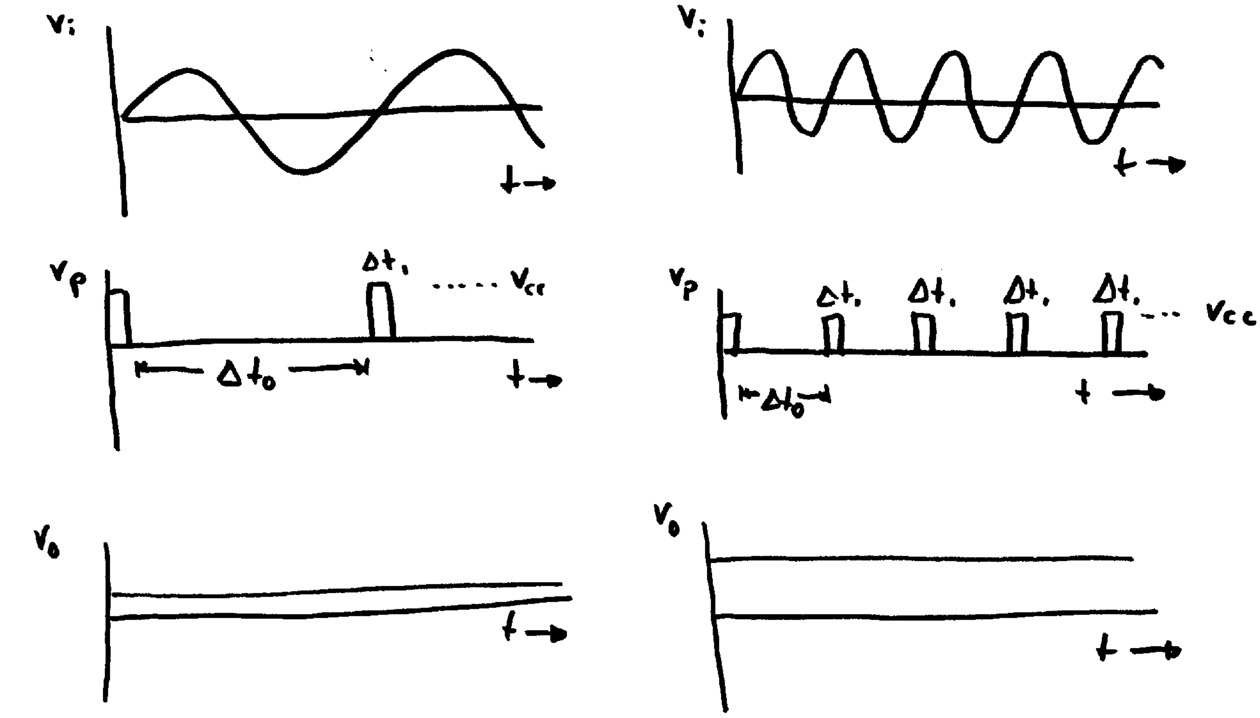
Frequency (counter).

A frequency meter can be made by converting the signal to digital format and <u>count</u> the number of pulses with digital electronics.

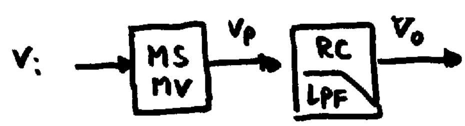


The conversion to square waves can be done by a high-gain amplifier (see, for example the comparator). while the counting can be done by counters based on flip-flops (digital electronics). Periodically resetting the counters gives a value for the frequency at regular intervals. The advantage is that we directly have the speed in digital format.

An analog meter can be made by converting the signal to pulses with predefined width. Every time the signal crosses a certain level, a pulse of width Δt_1 is generated. Between these pulses there is the 'dead time', Δt_0 that depends on the frequency



$$\Delta t_1 + \Delta t_0 = \frac{1}{f}$$



These pulses can be made, for instance, with mono-stable multi vibrators (MSMV). After low-pass filtering the output voltage is proportional to the fractional time the pulse is present

$$V_0 = \frac{\Delta t_1}{\Delta t_1 + \Delta t_0} \cdot V_{cc} = (\Delta t_1 \cdot V_{cc}) \cdot f$$

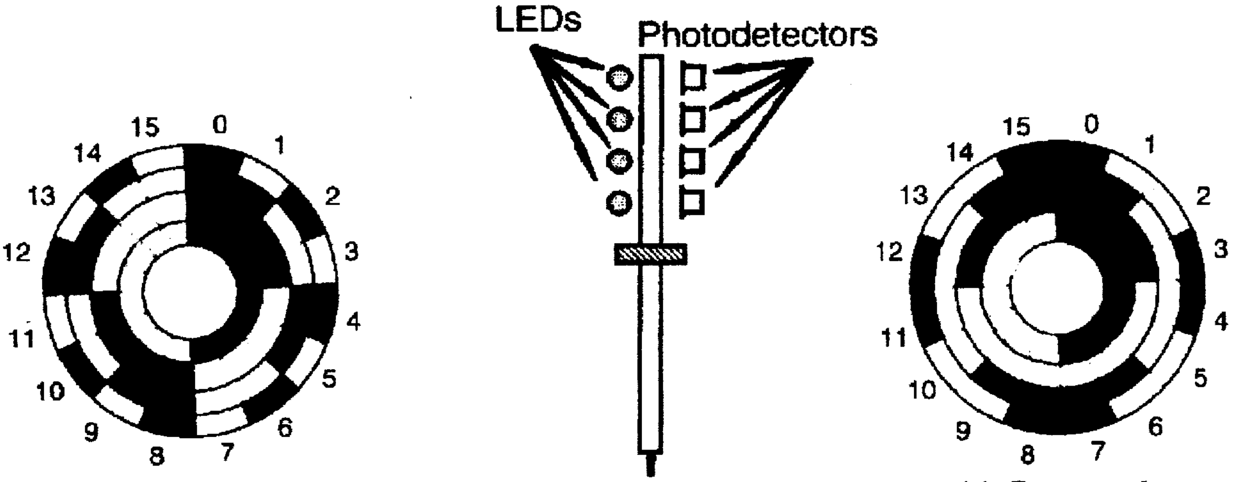
(with V_{cc} the pulse height). Then we have directly the output voltage as a linear function of the input frequency.

Voltage-to-frequency converter

Sometimes we want instead of $f \to V$, we want the opposite, $V \to f$. This is simple. There are special integrated circuits for that, so-called VCO's (voltage-controlled oscillator), which we will not further analyze here.

Angle

To measure angle, we can use a rotational potentiometer (see earlier in this chapter). However, these have contact (and friction) with the object measured. Moreover, they have a limit in angle (for instance 0°-300°). Imagine a wind direction meter. This has to be able to keep on rotating in the same direction forever. A solution is a set of LED-optical sensor pairs as described before. Between them is sandwiched a disk with open and closed places, see Figure below (left)



a) Binary b) Gray code

Source: Turner & Hill

The state of the four photodetectors then tell us the angle. With 4 LED-OD's we have 2^4 states:

MSB	.....		LSB		
0	0	0	0	0	0° - 22.5°
0	0	0	1	1	22.5 - 45°
0	0	1	0	2	45° - 67.5°
0	0	1	1	3	67.5° - 90°
0	1	0	0	4	90° - 112.5°
0	1	0	1	5	112.5° - 135°
0	1	1	0	6	135° - 157.5°
0	1	1	1	7	157.5° - 180°
	⋮				
1	1	1	1	15	337.5° - 360° (0°)

The problem with this system is that at some positions, more than 1 "bit" changes. Take for example the state 15 changing into state 0. All 4 bits change at this transition. The problem is that they do not exactly change at the same time. Imagine first goes the outside bit (LSB), then the inside bit (MSB), then bit 2, then bit 3. What we get that in a short interval of time

1 1 1 1 = 15
↓
1 1 1 0 = 14
↓
0 1 1 0 = 6
↓
0 0 1 0 = 2
↓
0 0 0 0 = 0

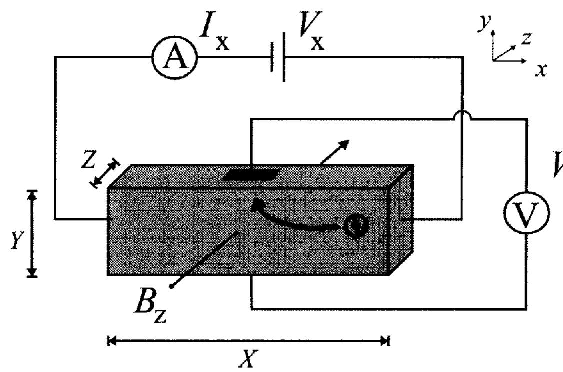
It emulates the wind changing direction very rapidly. To avoid this, Gray code has been invented. In this code only one bit can change at a transition

0 = 0000
1 = 0001
2 = 0011
3 = 0010
4 = 0110
⋮
15 = 1000

(see Figure on the previous page (right side)

Magnetic field

Hall probe



Charges moving in a magnetic field feel an effective force proportional to

$$\bar{F}_B = q \cdot \bar{B}_z \times \bar{v}_x$$

(this force is perpendicular to both the field B and the velocity v). In the Figure above, a positive charge moving along the x direction in a field along z, will be bend towards y. Charges build up there and create a compensating electric field in this direction (y). At steady state there is no net current in this direction, and the two forces F_B and F_E must cancel.

$$F_B = q \cdot B_z \langle V_x \rangle$$

$$F_E = q \cdot E_y = q \cdot \frac{V_y}{Y}$$

$\langle V_x \rangle$: velocity in x-direction

X, Y, Z : sizes of sample in x,y,z direction

further

$$I_x = q \cdot n \cdot \langle V_x \rangle \cdot Z \cdot Y$$

n = density of charge

q = 1.6×10^{-19} C

μ = charge mobility

$\Rightarrow$

$$B_z = \frac{n \cdot q \cdot V_y \cdot Z}{I_x}$$

Alternatively, if we don't know the density of charge (n), we can use the equation

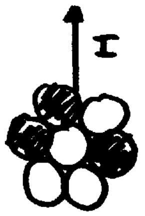
$$\langle V_x \rangle = \mu E_x = \mu \frac{V_x}{X}$$

to get

$$B_z = \frac{1}{\mu} \cdot \frac{V_y \cdot X}{V_x \cdot Y}$$

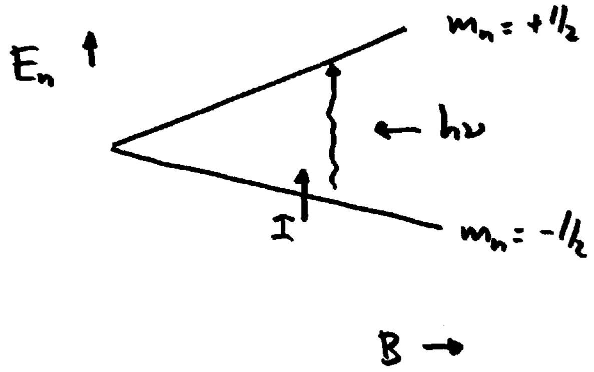
<u>NMR probe</u>

The NMR (nuclear magnetic resonance) probe is based on the effect that a nuclear particle (proton or neutron) has an associated nuclear spin. The energy of this spin depends on the magnetic field and has two distinct possibilities



$$E_n = g_n \gamma_n (\bar{B} \cdot \bar{I}) \quad (\cdot = \text{in produkt})$$

$$= m_n g_n \gamma_n B \quad , \quad \text{with } m_n = \pm \frac{1}{2}$$



note
$g_n = 2.0023$
$\gamma_n = 5.05 \; 10^{-27}$ J/T
$h = 6.63 \; 10^{-34}$ J/Hz
ν = photon frequency

The spin can absorb energy (electromagnetic; photon) when the energy of the photon is equal to the difference in the two energy levels

$$h\nu = E_{+1/2} - E_{-1/2}$$

$$= \frac{1}{2} g_n \gamma_n B - \left(-\frac{1}{2} g_n \gamma_n B\right)$$

$$= g_n \gamma_n B$$

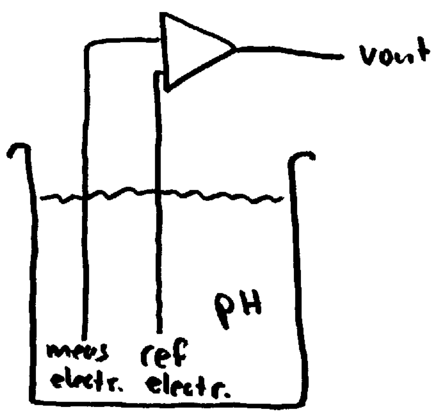
Thus, measuring the frequency of resonance directly gives the magnetic field:

$$B = \frac{h\nu}{g_n \gamma_n}$$

For instance, at 1 T, the frequency is 15.32 MHz.
An NMR probe consists of a feedback loop to keep the system in resonance and a frequency meter to measure ν. The above equation then give the magnetic field (h, g_n and γ_n are constants)

pH sensor

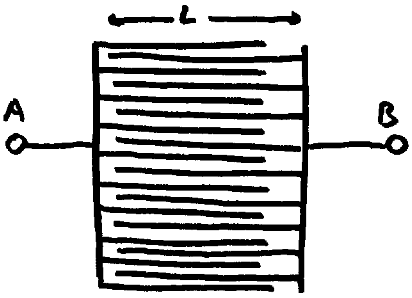
A pH sensor is basically a battery whose output voltage varies with the pH level of the solution. It contains of two different electrodes placed in the solution. Each has



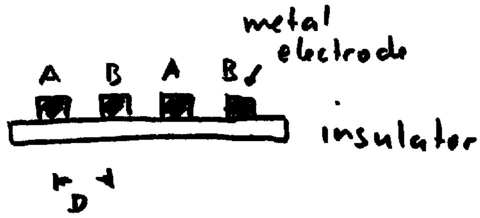
an associated electrochemical potential. The two potentials are fed into a high impedance amplifier (why must it be high impedance ?). The output voltage is thus a measure for in pH of the solution. Normally - 700 mV for an acid solution with pH equal to 0 and +700 for base solutions with pH of +14

Humidity

Humidity can be measured by determining the conductivity of air. Because this conductivity is rather low (even wet air), the signals are rather weak.



top view



cross-section

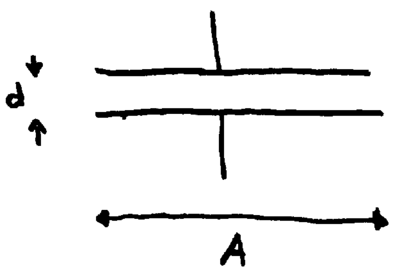
The resistance of the sensor is $\sim \rho \cdot \frac{D}{L}$, with ρ the resistivity, D the distance between two electrodes and L the length. To get reasonable ('measurable') values, D has to be small and L large. See for instance the inter-digitated electrodes in the figure. This brings the resistance in the kΩ range.

Pressure sensor

A very common pressure sensor is a microphone. More precisely, it is a differential

pressure meter, in the sense that it measures changes in pressure. It converts this into voltage. How does it work?

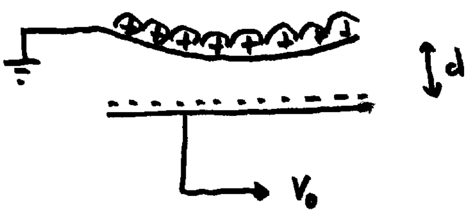
A microphone consists of a simple capacitor made of two metal sheets. The capacitance depends



on the area and the distance between the sheets

$$C = \frac{\varepsilon_0 \cdot A}{d}$$

One of the sheets is a thin, flexible membrane that can move (bend) easily upon changes of pressure. Moreover, one of the membranes is permanently charged (electrostatically). The other membrane has compensating mobile charge.



The effect is that any changes of the distance between the sheets (d) cause changes in output voltage.

Remember that charge, voltage and capacitance have a direct relation

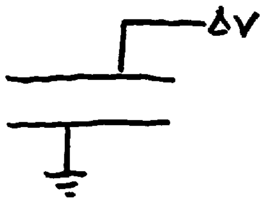
$$C = \frac{Q}{\Delta V}$$

with the amount of charge Q being fixed, any changes in capacitance, have direct effect on the voltage:

$$\Delta V(d) = \frac{Q}{C(d)} = \frac{Q}{\varepsilon_0 A} \cdot d$$

Thus, pressure changes cause the membrane to move, changing d, and this causes a change of output voltage ΔV.

An alternative scheme is to use a constant bias (ΔV). Any change in membrane position



then causes a change in stored charge

$$Q(d) = \Delta V \cdot C(d) = \frac{\varepsilon_0 A V}{d}$$

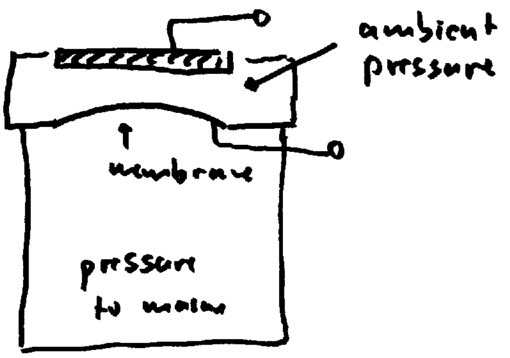
This causes an external current

$$I(t) = \frac{dQ}{dt} = \frac{dQ}{dd} \cdot \frac{dd}{dt}$$

$$= -\frac{\varepsilon_0 A V}{d^2} \cdot \frac{dd}{dt}$$

Obviously, this second scheme can only measure changes in pressure. With a static pressure, the signal is zero (because then $I(t) \sim \frac{dd}{dt} = 0$).

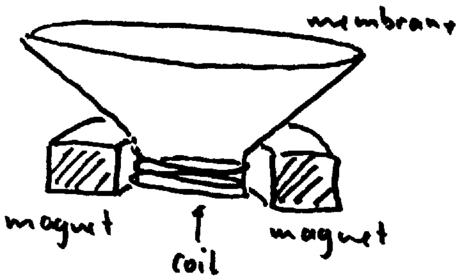
The first scheme can be used to measure static pressure as shown in the picture.



The bending of the membrane is caused by a differential pressure (relative to the environment)

Pressure actuator

The oposite of a pressure sensor is a pressure actuator, $V \rightarrow p$. This is called a speaker and normally works on the principle of magnetic forces.



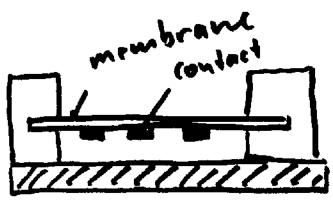
Currents pass through a coil connected physically to the membrane. These currents induce a magnetic field.

interacts with the static magnetic field. Thus, the currents cause a movement of the membrane and alternating currents cause pressure waves (a.k.a. "sound").

Note that the functions of sensor and actuator are reversible. A microphone can be used as a speaker and a speaker as a microphone. Obviously this will result in transducers of low quality, since they have been optimized for either sensing or acting.

Pressure switch

A pressure switch is also made from a membrane. This time metal contacts are placed on the membrane and they make contact when a pressure is too high or too low.

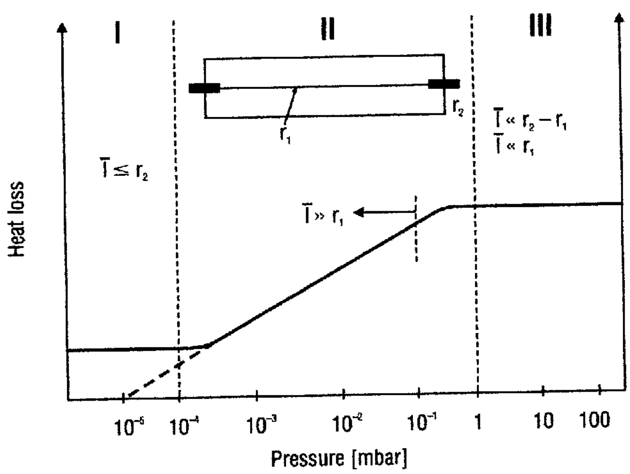


The capacitor for a pressure sensor is good when the pressure is c to atmospheric values, but it is not good for vacuum sensing; it cannot

detect / distinguish between pressures of, for instance, 1 mbar and 0.1 mbar because it measures pressure relative to atmosphere (1 bar): 1% error is 10 mbar.
There exist special vacuum sensors that depend on physical properties of air.

"Pirani" pressure sensor

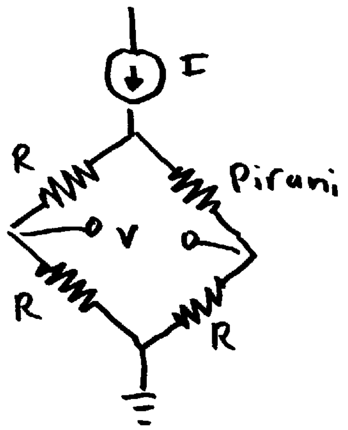
This pressure sensor is based on the principle that thermal conductivity of a gas depends on it's pressure. A thin wire is heated up to 100-150°C and its heat loss is measured, see Figure below. The sensor can for instance



I Thermal dissipation due to radiation and conduction in the metallic ends
II Thermal dissipation due to the gas, pressure-dependent
III Thermal dissipation due to radiation and convection

Fig. 3.10 Dependence of the amount heat dissipated by a heated filament (radius r_1) in a tube (radius r_2) at a constant temperature difference on the gas pressure (schematic diagram)

LEYBOLD VACUUM PRODUCTS AND REFERENCE BOOK 2001/2002 be placed in a constant-current set up.



When the Pirani sensor is in vacuum it has difficulty loosing heat and the wire heats up. This increases its resistance and brings the wheatstone bridge off-balance.

In modern equipment, the voltage supplied is regulated to keep the resistance (and temperature) constant.

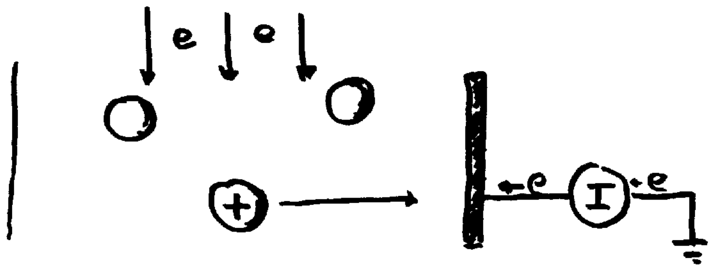
The pressure range is approximately 10^{-4} mbar to atmospheric pressure.

The output voltage depends on the gas measured, because the thermal conductivity is different for different gasses.

"Penning" pressure sensor

This pressure sensor is based on the ion-izability of a gas, which depends on the pressure.

The gas is partially ionized by electrons that are accelerated in a strong electric field. The positively charged ions of the gas are collected and neutralized by a measuring electrode.

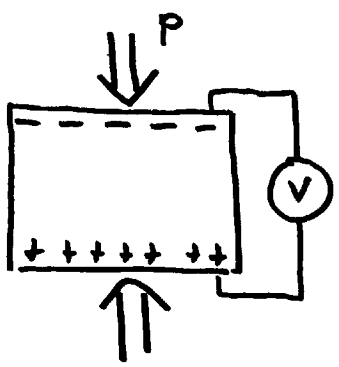


The current needed for this neutralization is a measure for the number of ionized gas particles and thus for its density (proportional to the pressure)

- The sensor depends on the gas, since some gasses are easier to ionize than others.
- A "cold cathode" sensor (Penning) works with high voltages (~ 2 kV)
- To increase length of interaction of electrons with gas, the system is placed in a strong magnetic field. The electrons have a spiral path
- good for very high vacuum (~ 10^{-9} mbar)

Piëzo electric effect

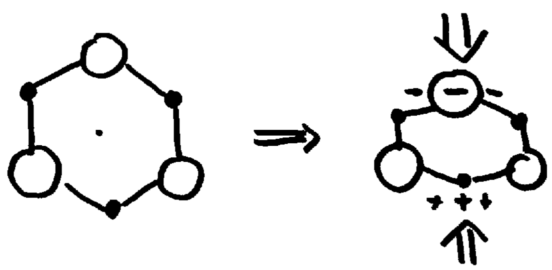
This effect was discovered by the brothers (Pierre and Jacques) Curie. A sample that is squeezed (Greek *piezein* = squeeze) by pressure, creates a voltage at it's edges.



Necessary for this to happen:

- The sample should be a single crystal
- The crystal (of units/molecules) should not have inversion symmetry

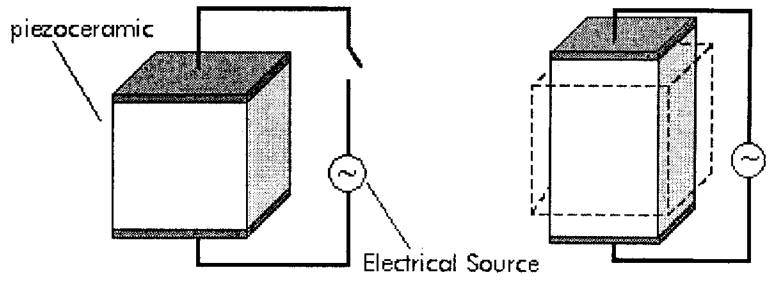
This is because the basic ingredient of the piëzo electric effect is a molecule, that, when deformed, becomes polarized, see Figure below.



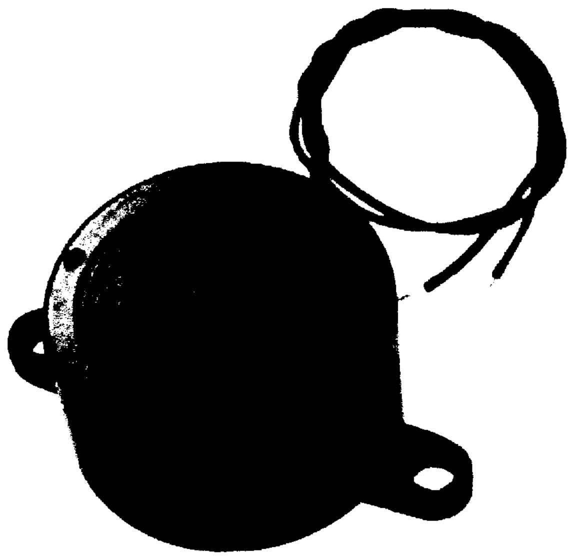
non inversion symmetric molecule before and after deformation

The combined effect of all the molecules in the crystal causes an external measurable voltage.

The oposite effect also exists, namely the effect that a non-inversion symmetric crystal deforms upon the application of an external magnetic field.

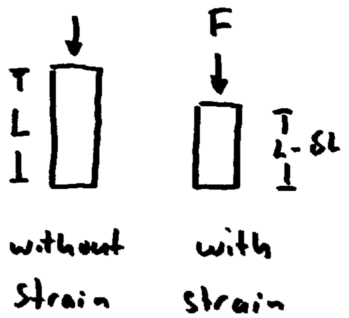


The piëzo sensor can thus be used for various things, including a pressure sensor, that works similar to a pressure sensor based on a capacitor.
Moreover, the piëzo electric effect can also be used in a pressure actuator (speakers). for sound output. Examples are watches and other low quality sound systems.



Strain

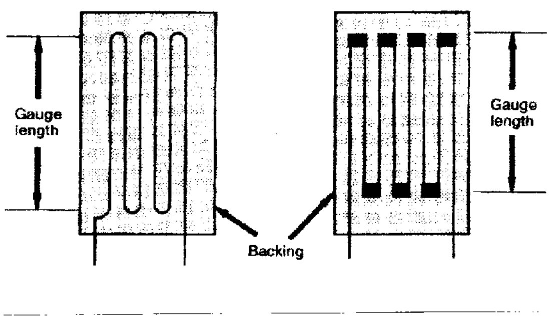
"Strain" is the deformation of an object as caused by an external force ("stress"). Without going into the details of how exactly stress causes strain, here we will only describe a way to quantify strain.



Strain is the relative change in size in a dimension of an object.

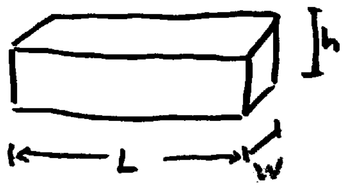
For instance: $\varepsilon = \frac{\delta L}{L}$

The simplest electrical way to measure strain is to connect a wire to it and measure the resistance, see Figure below.



The gauge factor κ is then defined as the ratio of relative change in R to relative change in L:

$$\kappa \equiv \frac{\delta R/R}{\delta L/L} = \frac{\delta R/R}{\varepsilon}$$



$$R = \rho \frac{L}{w \cdot h}$$

In first order, when stress is applied (along L), only that dimension changes length: The gauge factor is then

$$K = \frac{dR/R}{dL/L} = \frac{dR}{dL} \cdot \frac{L}{R} = \frac{\rho}{w \cdot h} \cdot \frac{L}{\rho \frac{L}{wh}} = 1$$

In second order, the stress along L also changes the dimensions perpendicular to L. For instance, when the length of an object is increased, its thickness w and h decrease. This can make the gauge factor much larger than 1. (see example)

Moreover, some materials (semiconductors) do not depend linearly upon lenght. This can also make $K >> 1$

example :

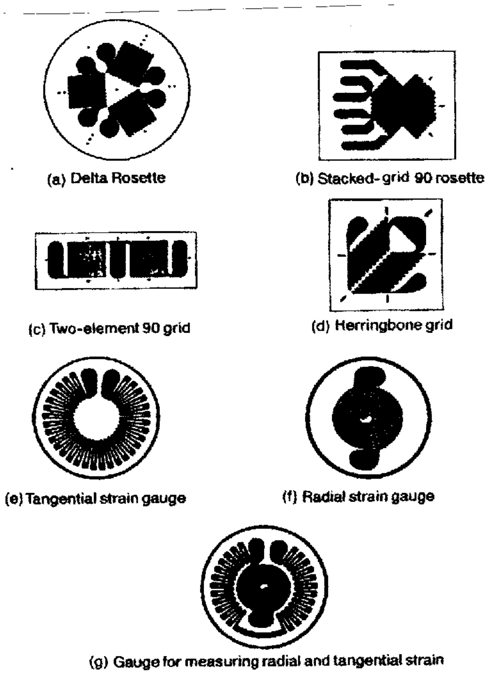
what is the gauge factor when all parameters depend on the stress and length :

$h \rightarrow h(L)$

$w \rightarrow w(L)$ } Poissons ratio $\nu \equiv -\dfrac{dw/w}{dL/L}, \dfrac{dh/h}{dL/L}$

$\rho \rightarrow \rho(L)$ (Piëzo electric effect)

$K =$?



(a) Delta Rosette

(b) Stacked-grid 90 rosette

(c) Two-element 90 grid

(d) Herringbone grid

(e) Tangential strain gauge

(f) Radial strain gauge

(g) Gauge for measuring radial and tangential strain

Flow meters

hot-wire meter

In the same way vacuum was measured in a Piranni sensor, the flow of air (or liquid) can be measured.

The amount of heat transfered from a hot object to a cold medium is given by

$$P = \alpha A \Delta T$$

with ΔT the diffeerence in temperature between the object and the medium. A the area of the object and α the heat-transfer coëfficient that depends on the speed of the medium:

$$\alpha = \alpha_0 + \alpha_1 \sqrt{v}$$

with α_0 and α_1 constants and v the velocity of the medium.

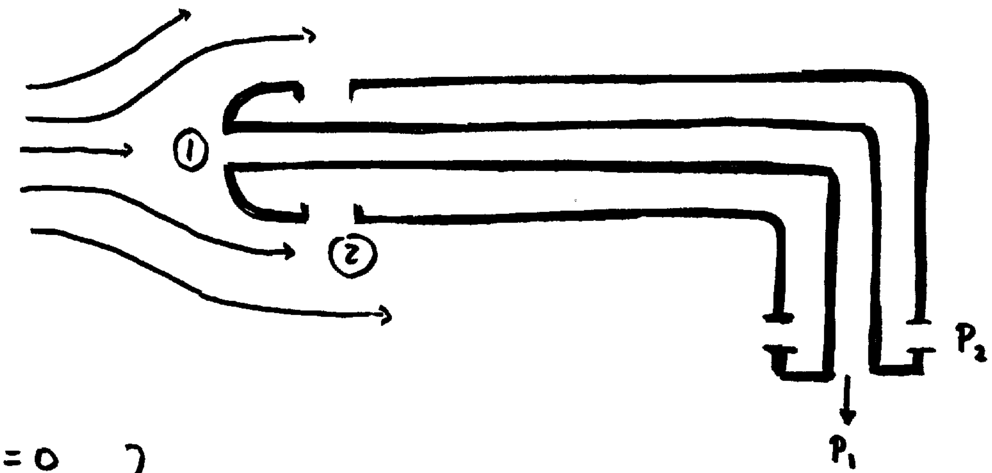
The power consumed by the object is $\frac{V^2}{R}$ or I^2R. The resistance depends on the temperature. The trick is thus to main R constant and measure the current or voltage needed to maintain this. This will then give the speed of the medium via the above two equations (if a_0, a_1, A and ΔT are known)

Pitot tube

This depends on the Bernoulli Equation :

$$\frac{v^2}{2} + gh + \frac{P}{\rho} = \text{constant}$$

- v = speed of air
- g = gravitational acceleration ($9.8\ m/s^2$)
- h = height
- P = pressure
- ρ = density of air



① : $v = 0$
② : $v = v$

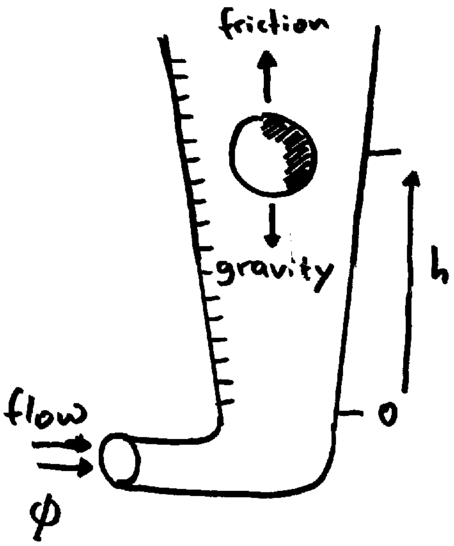
$$\frac{v^2}{2} + gh_2 + \frac{P_2}{\rho} = gh_1 + \frac{P_1}{\rho}$$

Thus (since $h_1 \approx h_2$)

$$v = \sqrt{\frac{2(P_1 - P_2)}{\rho}}$$

The problem is reduced to measuring (difference in) pressure, see earlier in this chapter.

Flow meter (mechanical)



A ball is floating in the air flux. The gravitational force

$$F_{\downarrow} = m \cdot g$$

is thus equal to the friction

$$F_{\uparrow} = 6 \pi r \eta v$$

(m = massa, g = gravitational acceleration, r is radius of object, η = fluid viscosity, v is air speed). [This is Stoke's Equation that is good for small spheres.]

At equilibrium

$$v = \frac{m \cdot g}{6 \pi r \eta}$$

which is constant.

The speed at the place of the ball is given by the airflow ϕ (m^3/s) divided by the area A (m^2):

$$v = \frac{\phi}{A} \quad (m/s)$$

The area is given by the area of the tube divided by the crossection of the ball

$$A = \pi (R^2 - r^2)$$

Finally, the radius of the tube depends on the height in a linear way.

$$R = r + \alpha h$$

(at $h=0$, the radius of the tube is exactly equal to R). Total (ignoring term $\alpha^2 h^2$):

$$\phi = \frac{12 \pi^2 r^2 \eta \alpha}{mg} \cdot h$$

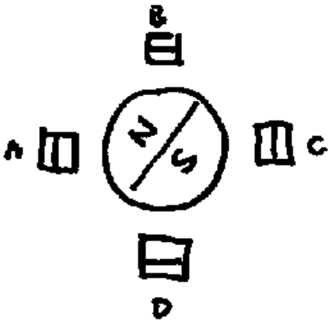
Stepper Motor

A stepper motor is a motor that can make small exact steps in a rotation. For each pulse supplied to the motor, the axis advances one step, typically a few degrees. By supplying pulses with a certain determined frequency, the motor will rotate with an exact velocity. Thus, the stepper motor has two important applications:

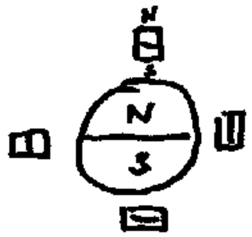
- position control
- velocity control

It is not difficult to understand how this works.

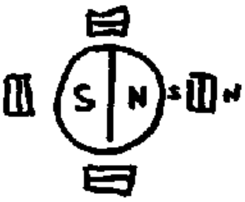
The central core of the motor, the 'axis" consists of a permanent magnet. Around it, the 'fixed' part of the engine, are electro-magnets, basically coils. When current is passing through the coils, a magnetic field is created and the permanent magnet connected to the axis will align itself to this field.



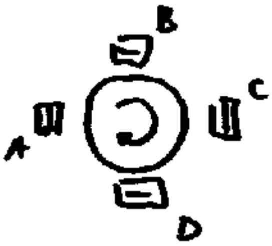
← stepper motor with 4 poles. The axis has a permanent magnet $\frac{N}{S}$. in this case randomly oriented. ⊟ is a coil.

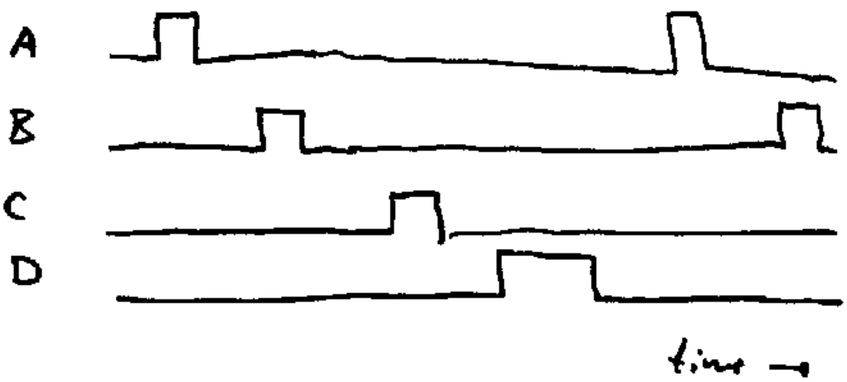


← passing a current through a coil will align the axis-magnet to that coil.



← a current in the next coil will rotate the axis further.





A sequence of pulse will make the axis rotate, with an exact velocity given by the frequency of repetition of pulses.

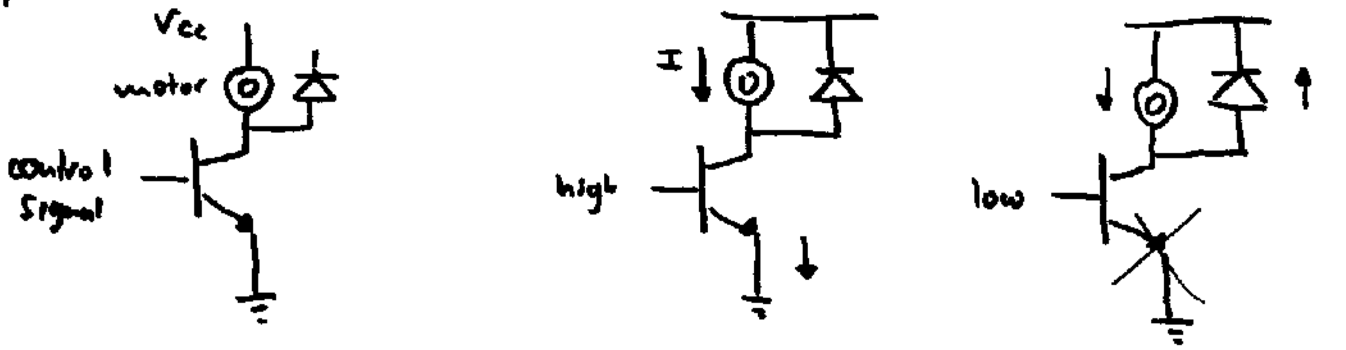
The direction is given by the pulse sequence. For example, the sequence A-B-C-D-A-B.. will rotate the engine clockwise, while the sequence A-D-C-B-A will make it rotate counterclockwise.

The stepper engine can have more poles at the permanent magnet and more coils in the static part. This can increase the resolution significantly. In the above example a single step was 90°.(*) Commercial stepper engines are of the order 3.6° per step, or, in other words 100 steps for a full rotation instead of the 4 given above.

- Note that the length of the step pulse is not relevant (as long as it is long enough for the engine to make one step).

To drive a stepper motor we cannot directly connect a (power) transistor. The problem is that a stepper motor is not only a device that converts electric energy into kinetic energy (rotation), but also reverse. When

(*) by putting two coils (or two) [illegible] the motor rotates, for instance by inertia after a pulse, a large current is induced in the coils. This creates voltages up to hundreds of volts! This can destroy a transistor easily. Therefore we have to "protect" the transistors by shunt diodes. For instance:



circuit — normal operation — motor continues to turn with control off

To make the motor make a step, the control is set to "high", opening the transistor. The current Ic goes through the motor, through the coil. After making the control line "low", the motor can continue to turn, inducing a large current in the coil. This current cannot go through the transistor because it is closed. A large voltage would build up. To prevent this the diode can be placed, as shown in the figure.

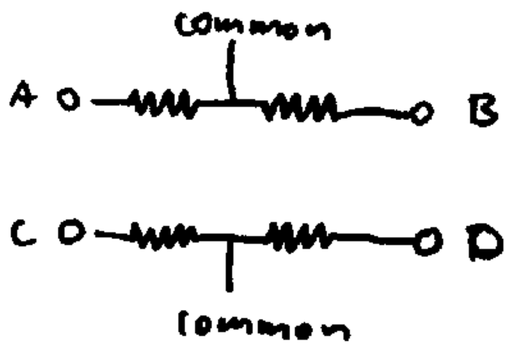
Voltage peaks are also caused by 'inertia' of current : current creates magnetic field, magnetic field is (magnetic) energy ($E \propto \int B^2 dx^3$). Energy cannot disappear instantaneously and thus current continues, even if the circuit is opened.

Stepper motors exist in many types.
Most important are

unipolar ⟷ bipolar
permanent magnet ⟷ reluctance

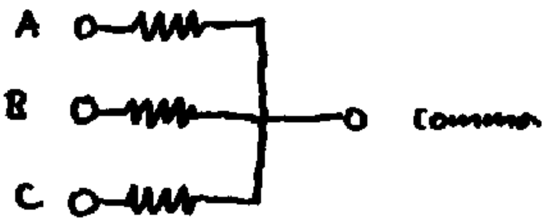
The ones with permanent magnets can be recognized by the friction that is felt when rotating it manually, while the reluctance motors run more smoothly.

Unipolar vs bipolar signifies if the driving pulses are (can be) both polarities or not.
Examples:

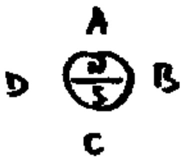


unipolar 4-phase (6 wires)

bipolar 3-phase (4 wires)



Half-steps can be made by applying a pulse to two consequtive coils. Superposition of the magnetic fields would cause a magnetic orientation in-between.



For instance the sequence

A, AB, B, BC, C, CD, D, DA,

cause a full rotation in 8 steps instead of 4, increasing the resolution to 45° instead of 90°

- A stepper motor with 5 wires is almost certainly 4-phase unipolar
- A stepper motor with 6 wires is probably also 4-phase unipolar, but with two common power wires. They may both be the same color
- A stepper motor with only 4 wires is most likely bipolar.

http://www.doc.ic.ac.uk

	PM	VR
holding torque	small	large
own torque	yes	none
oscillation at step	smaller	larger
resonance	smaller	larger
resolution	limited	large
control	phases ≥ 2	phases ≥ 3

"stepper motors"

PM = permanent magnet

VR = variable reluctance

3 Signal Acquisition

At this point we have the signal in a clean electronic format, free of noise (ideally) and with a measurable magnitude. We have calibrated our sensor so that we know exactly the relation between the final voltage (or current) and the measured physical parameter. For simple systems this can be enough to generate an output signal (power to a heater, switching on the light, etc.). For more advanced systems, some decision making and processing of signal (information) is preferably done on a computer. Thus, we need to design a way to convert the signal to digital format and communicate it to a computer. This chapter deals with these aspects which are generally called "data acquisition"

- translation from the electronic domain to the digital domain and vice-verse
- communication of the digital information with a computer.

The first part, conversion of the signal is done with analog-to-digital converters (ADCs) and digital-to-analog-converters (DACs). The second part involves communication standards such as RS232, parallel port, GPIB (HPIB / IEEE 488), etc.

ADCs

An ADC is an analog-to-digital converter and translates an input voltage to an output bit pattern. For linear ADC's, the output bit pattern ("number") is a linear function of the input voltage (ignoring offset). These ADCs have three important parameters

- input voltage range → $V_{max} - \Delta V_{min}$
- resolution in V (number of bits) → n
- resolution in time (sampling rate) → Δt

Analyzing the first two, the resolution in voltage thus is

$$\Delta V = \frac{V_{max} - V_{min}}{2^n - 1}$$

For example, an ADC with 0 to 5 V range with 8 bits thus has a resolution of $\Delta V = 5V/256 = 19.6\ mV$.

The simplest ADC is thus a 1-bit ADC that only determines if the signal is positive or negative. The most accurate ADCs have about 24 bits and have thus resolutions in the order

$$\Delta V = \frac{10\ V}{2^{24}} = 0.6\ \mu V$$

Increasing the number of bits has no use because the resolution will drop below the noise level.

The other important parameter is the sampling rate. It is no use sampling faster than (two times) the maximum frequency of information, as it will only increase the cost of the equipment.

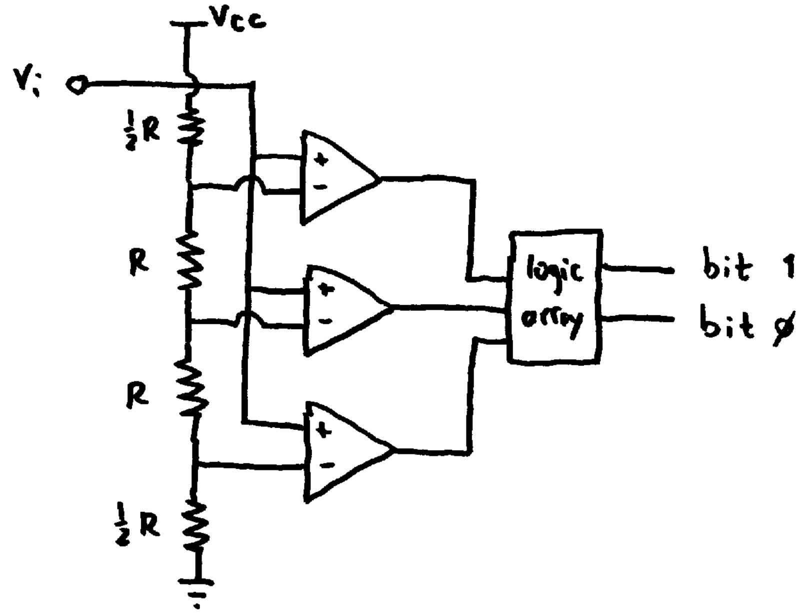
$$f_{sample} = 2 \cdot f_{max} \qquad \text{(Shannon-Nyquist)}$$

The most popular (most sold) ADC is probably a 16 bit, 44.1 kHz sampler that is used for audio samplers (upto 22.05 kHz sound).

Types of ADC

(1) Flash type ADC. This has 2^n-1 comparators that compare the signal to 2^n-1 reference signals. Each comparator has a binary output signal telling if the signal is larger (1) or smaller (0) than the reference signal of that

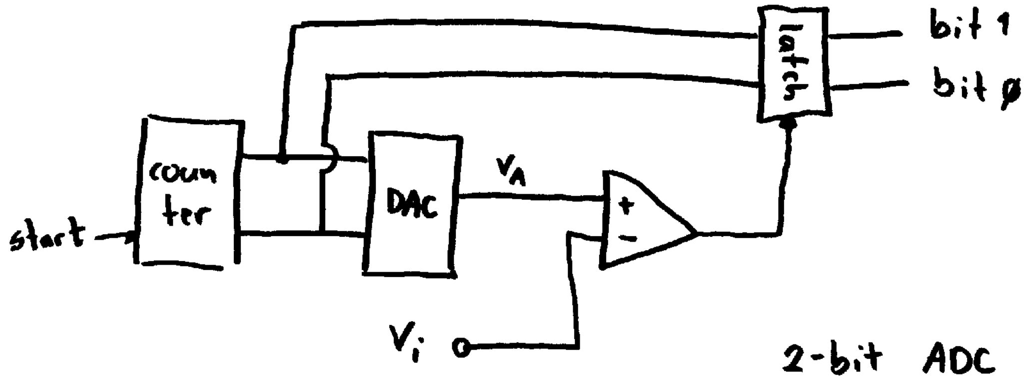
comparator. For example a 2-bit flash ADC might look something like shown in the figure below



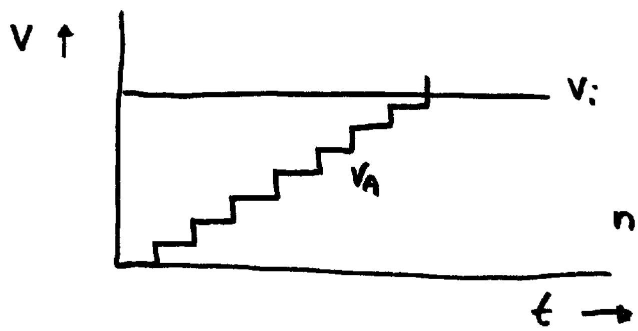
The conversion analog → digital is (nearly) instantaneous. The price to pay is a very complicated circuit and grows rapidly ($\sim 2^n$) with the number of bits. Flash conversters are normally limited to 8 bits, but go easily up to GHz sampling rates.

② δ delta encoded ADC. A digital counter is fed into a DAC and compared with a single comparator to the signal. The counter

Stops when the DAC signal is larger than the input signal. The value of the counter is then copied to the ADC output, see for example the 2-bit ADC in the figure below



2-bit ADC

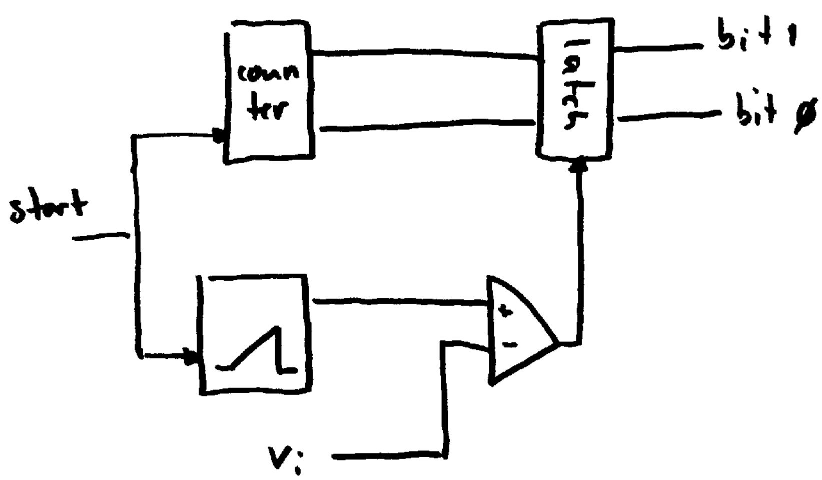


n-bit ADC signals

The electronics of this type of ADC are simpler, but it is also much slower. In the worst case, the sample is ready only after 2^n clock cycles. There exists improved version, in which for instance the last value is kept and used as a starting point for the next counting (the counter then can count up and down).

Another improvement (pipeline ADC) first determines the MSB (most significant bits) and then subtracts this coarse voltage with a DAC and a diferential amplifier from the input signal. In the second step the remaining bits are found (LSB).

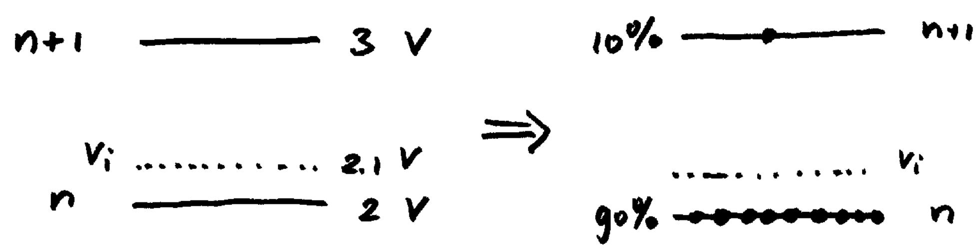
③ Analog-ramp ADC. This is similar to the delta ADC, but instead of a DAC, an analog signal is used to compare



2 bit analog-ramp ADC

The resolution of an ADC can be increased by adding noise at the input and using over-sampling (sampling more than once for every desired sample).

Imagine an ADC with 1 volt resolution that

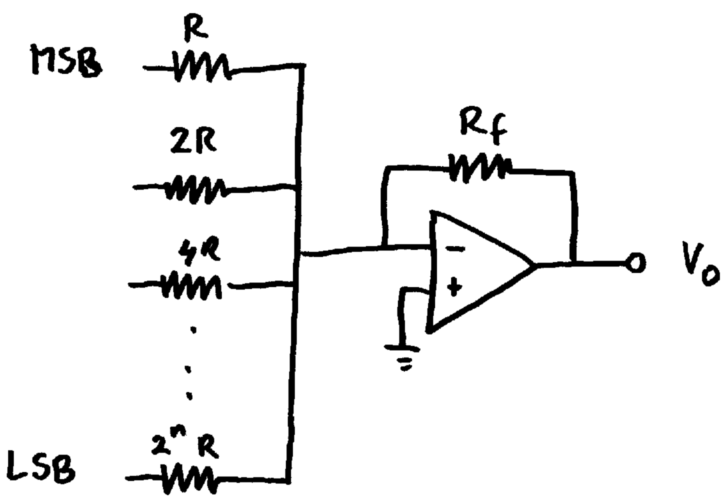


receives a signal of 2.1 V without noise. This would result in an ADC value of n, representing 2V. For 100 samples this results 100 times in the signal n = 2 V. Adding noise at the input with an amplitude larger than the digitalization size (1 V) or resolution will cause the ADC to sometimes return n+1 (3V). For a large factor of oversampling, the <u>average</u> of this will give 2.1 V. In fact, the resolution is increased by a factor N equal to the oversampling rate. Avreraging can be done digitally.

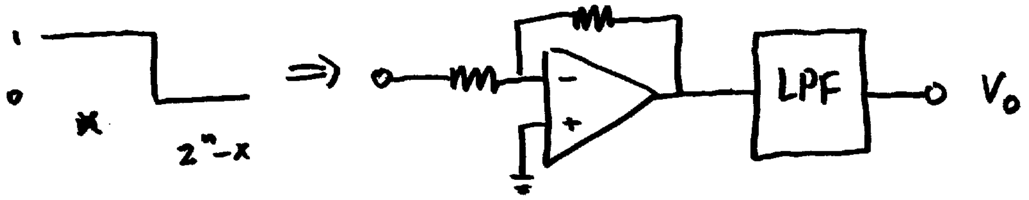
$$\Delta V = \frac{V_{max} - V_{min}}{(2^n-1)\cdot N}$$

DACs

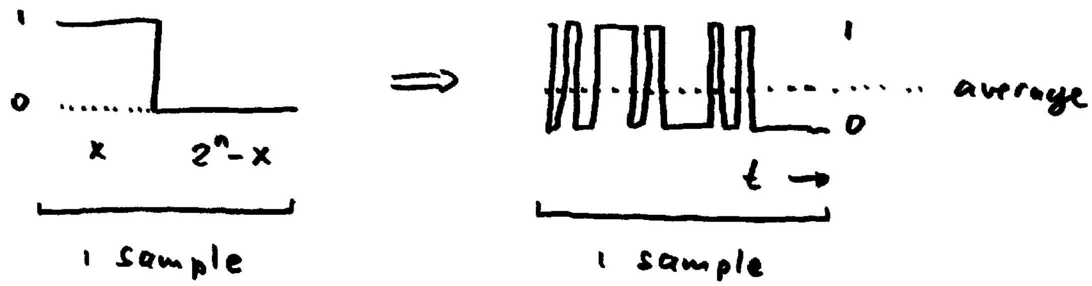
A DAC (digital-analog converter) is doing the opposite, conversion to analog. and works something like given in the figure below



Also for DACs oversampling can be used. Imagine we have an n bit DAC from 0 to 10 V. The resolution is thus $\frac{10}{2^n}$ and we have a value x between 0 and 2^n. The output voltage, $x \cdot \frac{10}{2^n}$, can be generated by placing x times a logical 1 (≡ 10 V) at the output and $2^n - x$ times a logical 0 (≡ 0V) and then taking the average. In other words we can use a 1-bit DAC:



In more advanded DACs the bit pattern at the entrance is randomized to optimize the signal quality



In many cases, ADCs and DAC are part of the equipment (for example a multimeter or oscilloscope) and we just have to communicate with it.

RS 232

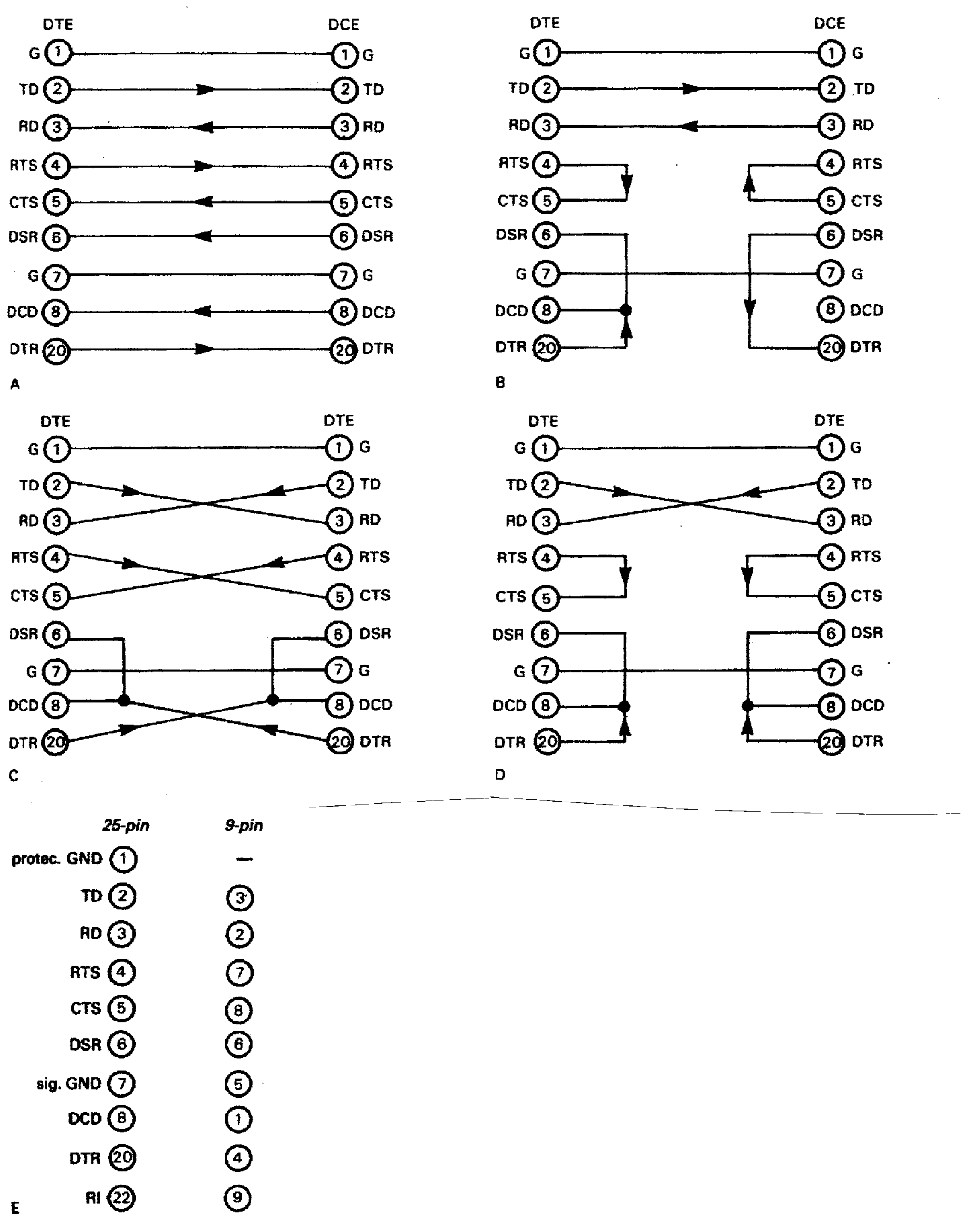
RS232 is the old standard of communication between digital equipment and is for instance the bace of modern USB communication.
The first version was designed for linking a typewriter to a modem (modulator - demodulator), or in general a DTE (data terminal equipment) to a DCE (data communication equipment). The original connector pin assignment is given in the figure on the next page.
It consists of a 25 pin (DB25) female connector for the DCE and a 25 pin male connector for the DTE. Later versions used only the most important connections and the connector was reduced to 9 pin (DB9). The older versions use "handshaking" protocols which means that the data is accompanied by signals on other pins telling the status of the DCE and DTE

for instance whether data is ready to be transmitted or if it is received well. These handshake lines come in pairs, for instance RTS / CTS (request to send / clear to send). The DTE sends a request to send when it is ready, but waits until the DCE is ready (CTS signal) before actually sending the data.

TABLE 10.4. RS-232 SIGNALS

Name	Pin number 25-pin	Pin number 9-pin	Direction (DTE↔DCE)	Function (as seen by DTE)	
TD	2	3	→	transmitted data	} data pair
RD	3	2	←	received data	
RTS	4	7	→	request to send (= DTE ready)	} handshake pair
CTS	5	8	←	clear to send (= DCE ready)	
DTR	20	4	→	data terminal ready	} handshake pair
DSR	6	6	←	data set ready	
DCD	8	1	←	data carrier detect	} enable DTE input
RI	22	9	←	ring indicator	
FG	1	–		frame ground (= chassis)	
SG	7	5		signal ground	

"Note that the signal names make sense only as viewed by the DTE: For instance, pin 2 is called TD ("transmitted data") by <u>both</u> sides, even though the DTE asserts it and the DCE receives it. Thus, the name of a pin isn't enough to tell you if it's an input or output – you also need to know whether the device thinks it's a DTE or DCE" (Horowitz & Hill, The Art of Electronics, including images shown here).



when two DTEs (or DCEs) are connected to eachother (for instance two computers), the TD (transmitted data) of one has to be connected to the RD (received data) of the other, and vice verse. This crossing of signals is called a null-modem cable, see the figure on the previous page. Note that all signals, including handshakes are crossed.

Often, especially in modern equipment, the handshake signals are ignored. In this case, handshaking is done by software and is encoded in the data. In this case only three wires are needed, RD, TD and ground. To make sure that hardware-handshaking equipment can communicate with software-handshaking equipment, it is best to auto-terminate the handshaking pins, see the right sides of the figure on the previous page.

Finally, the figures shows the pin number assignment for a DB25 connector. The 9-pin DB9 translation table is given in the last part of the figure.

Conclusion: To connect two pieces of equipment it is needed to know what type of equipment it is (DCE or DTE) and what kind of handshaking is used. Most modern equipment uses DTE-DTE with software handshaking where only three pins are connected and the rest ignored (not even nicely terminated)

RS232 is a serial connection, which means that information is sent 1 bit at a time. Contrasting this are parallel connections (printer port, parallel port and GPIB) where data is sent one byte at a time. the difference is based on the difference in hardware. RS232 has one data line and a parallel port has 8.

Signals at the datalines have the following properties:

* consist of "0"s and "1"s where a logical 0 is a voltage more than +3 V and a logical 1 is a voltage below -3 V. (max ± 25 V)
* Starts with a startbit, a logical "0"
* followed by a "character" that contains the information (the "data" being transmitted), and consists of 7 or 8 bits
* In case the character is 7 bits, the sequence is followed by a parity bit that verifies the integrity of the data bits.
* This is followed by a stop bit of arbitrary length (minimum 1 bit), namely a logical "1".

* The information can be sent at various speeds this is called the bit rate, or "baud", describing the number of bits per second transmitted (including parity bit). Standards are 300, 1200, 2400, 4800, 9600 and 19.200 baud with the

first one barely adequate for typewriters and the last one barely adequate for transmitting files from one computer to the other.

The communication type and speed is then summarized in a single word, for instance 9600 N 8 1

9600 = (1), N = (2), 8 = (3), 1 = (4)

(1) speed (baud)

(2) parity, "N" = no, "E" = even, "O" = odd (counting the number of "1"s in the character just sent, even or odd, with the parity bit that value needed to make it odd or even)
example even parity, 0110100 → parity bit will be 1.
"M" = mark, "S" = space (always 1 or always 0)

(3) number of databits (7 or 8)

(4) number of stop bits (1 or 2)

A typical RS232 bit pattern is given in the next figure

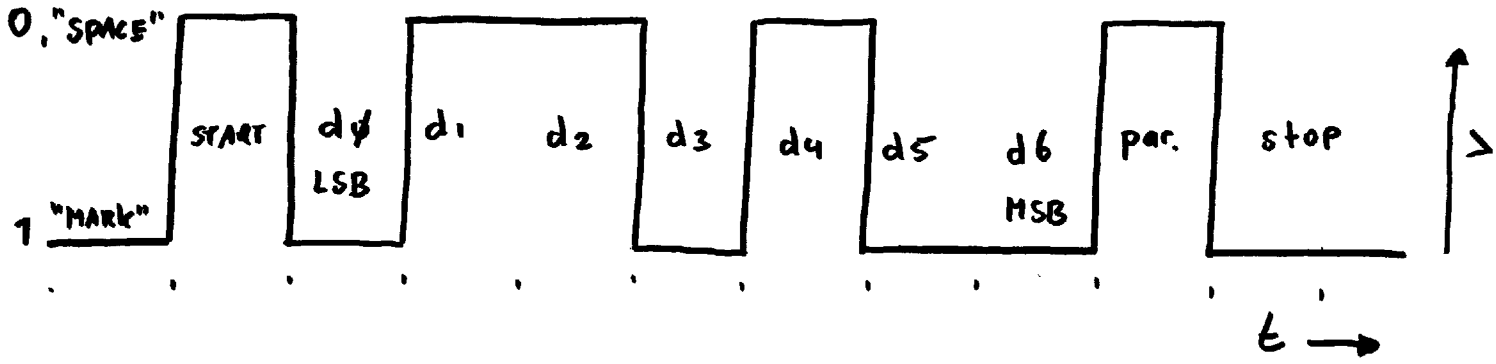
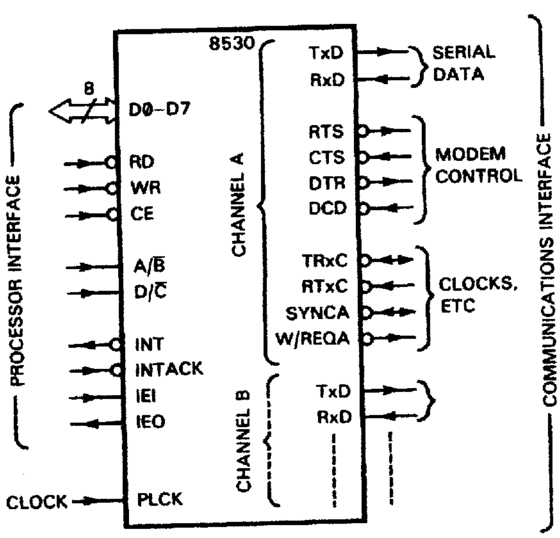


figure : Sending of character 1101001 (ASCII "i", dec. 105, hex 69) via protocol E71. Note that the bits are sent in reverse order, first the LSB and at the end the MSB.

Decimal	Octal	Hex	Binary	Value	
000	000	000	00000000	NUL	(Null char.)
001	001	001	00000001	SOH	(Start of Header)
002	002	002	00000010	STX	(Start of Text)
003	003	003	00000011	ETX	(End of Text)
004	004	004	00000100	EOT	(End of Transmission)
005	005	005	00000101	ENQ	(Enquiry)
006	006	006	00000110	ACK	(Acknowledgment)
007	007	007	00000111	BEL	(Bell)
008	010	008	00001000	BS	(Backspace)
009	011	009	00001001	HT	(Horizontal Tab)
010	012	00A	00001010	LF	(Line Feed)
011	013	00B	00001011	VT	(Vertical Tab)
012	014	00C	00001100	FF	(Form Feed)
013	015	00D	00001101	CR	(Carriage Return)
014	016	00E	00001110	SO	(Shift Out)
015	017	00F	00001111	SI	(Shift In)
016	020	010	00010000	DLE	(Data Link Escape)
017	021	011	00010001	DC1	(XON) (Device Control 1)
018	022	012	00010010	DC2	(Device Control 2)
019	023	013	00010011	DC3	(XOFF) (Device Control 3)
020	024	014	00010100	DC4	(Device Control 4)
021	025	015	00010101	NAK	(Negative Acknowledgement)
022	026	016	00010110	SYN	(Synchronous Idle)
023	027	017	00010111	ETB	(End of Trans. Block)
024	030	018	00011000	CAN	(Cancel)
025	031	019	00011001	EM	(End of Medium)
026	032	01A	00011010	SUB	(Substitute)
027	033	01B	00011011	ESC	(Escape)
028	034	01C	00011100	FS	(File Separator)
029	035	01D	00011101	GS	(Group Separator)
030	036	01E	00011110	RS	(Request to Send)(Record Separator)
031	037	01F	00011111	US	(Unit Separator)
032	040	020	00100000	SP	(Space)
033	041	021	00100001	!	(exclamation mark)
034	042	022	00100010	"	(double quote)
035	043	023	00100011	#	(number sign)
036	044	024	00100100	$	(dollar sign)
037	045	025	00100101	%	(percent)
038	046	026	00100110	&	(ampersand)
039	047	027	00100111	'	(single quote)
040	050	028	00101000	(	(left/opening parenthesis)
041	051	029	00101001	)	(right/closing parenthesis)
042	052	02A	00101010	*	(asterisk)
043	053	02B	00101011	+	(plus)
044	054	02C	00101100	,	(comma)
045	055	02D	00101101	-	(minus or dash)
046	056	02E	00101110	.	(dot)
047	057	02F	00101111	/	(forward slash)
048	060	030	00110000	0	
049	061	031	00110001	1	
050	062	032	00110010	2	
051	063	033	00110011	3	
052	064	034	00110100	4	
053	065	035	00110101	5	
054	066	036	00110110	6	
055	067	037	00110111	7	
056	070	038	00111000	8	
057	071	039	00111001	9	
058	072	03A	00111010	:	(colon)
059	073	03B	00111011	;	(semi-colon)
060	074	03C	00111100	<	(less than)
061	075	03D	00111101	=	(equal sign)
062	076	03E	00111110	>	(greater than)
063	077	03F	00111111	?	(question mark)
064	100	040	01000000	@	(AT symbol)
065	101	041	01000001	A	
066	102	042	01000010	B	
067	103	043	01000011	C	
068	104	044	01000100	D	
069	105	045	01000101	E	
070	106	046	01000110	F	
071	107	047	01000111	G	
072	110	048	01001000	H	
073	111	049	01001001	I	
074	112	04A	01001010	J	
075	113	04B	01001011	K	
076	114	04C	01001100	L	
077	115	04D	01001101	M	
078	116	04E	01001110	N	
079	117	04F	01001111	O	
080	120	050	01010000	P	
081	121	051	01010001	Q	
082	122	052	01010010	R	
083	123	053	01010011	S	
084	124	054	01010100	T	
085	125	055	01010101	U	
086	126	056	01010110	V	
087	127	057	01010111	W	
088	130	058	01011000	X	
089	131	059	01011001	Y	
090	132	05A	01011010	Z	
091	133	05B	01011011	[	(left/opening bracket)
092	134	05C	01011100	\	(back slash)
093	135	05D	01011101	]	(right/closing bracket)
094	136	05E	01011110	^	(caret/circumflex)
095	137	05F	01011111	_	(underscore)
096	140	060	01100000	`	
097	141	061	01100001	a	
098	142	062	01100010	b	
099	143	063	01100011	c	
100	144	064	01100100	d	
101	145	065	01100101	e	
102	146	066	01100110	f	
103	147	067	01100111	g	
104	150	068	01101000	h	
105	151	069	01101001	i	
106	152	06A	01101010	j	
107	153	06B	01101011	k	
108	154	06C	01101100	l	
109	155	06D	01101101	m	
110	156	06E	01101110	n	
111	157	06F	01101111	o	
112	160	070	01110000	p	
113	161	071	01110001	q	
114	162	072	01110010	r	
115	163	073	01110011	s	
116	164	074	01110100	t	
117	165	075	01110101	u	
118	166	076	01110110	v	
119	167	077	01110111	w	
120	170	078	01111000	x	
121	171	079	01111001	y	
122	172	07A	01111010	z	
123	173	07B	01111011	{	(left/opening brace)
124	174	07C	01111100	|	(vertical bar)
125	175	07D	01111101	}	(right/closing brace)
126	176	07E	01111110	~	(tilde)
127	177	07F	01111111	DEL	(delete)

The table on the previous page shows the conventional translation of bit patterns to text, ASCII. Note also the charaters 017 (XON) and 019 (XOFF) used for software handshaking, as described before.

Integrated circuits (ICs) have been developed to deal with RS232 communication. These are called UART or USART which stands for Universal synchronous asynchronous Receiver / Transmitter. All modern equipment with RS232 ports incorporate one of these, for instance the Zilog 8530 in PC's.



Zilog 8530 "serial communications controller" (UART) signals.

[Notice the signals in the figure (TxD, etc.)]

On the left side enters a full character of 8 bits coming from the processor of the computer (D0..D7) and on the right side the character is transmitted serially to another UART (TxD and RxD). The Zilog 8530 has two serial ports, "Channel A" and "Channel B" representing the two serial connections COM1 and COM2 of a PC.

The way the IC is embedded into the computer is described and treated in other disciplines ("Embedded systems"). Important for us to know is only how to set it up (bitrade, etc.) and how to place characters at the UART and how to read them from the UART.

The information as to where the information about the COM port control resides is written in the BIOS of the computer (whereas the

UART can be in any address of the memory, the BIOS <u>always</u> stores the information as to where the UART data is placed at the same address, namely (hexadecimal $)

BIOS table:

Start Address	Function
0000:0400 h	COM1's Base Address
0000:0402 h	COM2's Base Address
0000:0404 h	COM3's Base Address
0000:0406 h	COM4's Base Address

COM Port Addresses in the BIOS Data Area;

A

Often, but not always, they are pointing to

"standard" COM addresses

Name	Address	IRQ
COM 1	3F8 h	4
COM 2	2F8 h	3
COM 3	3E8 h	4
COM 4	2E8 h	3

Standard Port Addresses

B

These values are the first addresses of a block of 8 "registers", (so called base address)

At these addresses are the information needed to control the UART, see next table

Base Address	DLAB	Read/Write	Abr.	Register Name
+ 0	=0	Write	-	Transmitter Holding Buffer
	=0	Read	-	Receiver Buffer
	=1	Read/Write	-	Divisor Latch Low Byte
+ 1	=0	Read/Write	IER	Interrupt Enable Register
	=1	Read/Write	-	Divisor Latch High Byte
+ 2	-	Read	IIR	Interrupt Identification Register
	-	Write	FCR	FIFO Control Register
+ 3	-	Read/Write	LCR	Line Control Register
+ 4	-	Read/Write	MCR	Modem Control Register
+ 5	-	Read	LSR	Line Status Register
+ 6	-	Read	MSR	Modem Status Register
+ 7	-	Read/Write	-	Scratch Register

C

DLAB means the MSB (bit 7) of the line-control register (LCR) at base address + 3. If this bit is set to 1, the first two bytes have a different meaning, as shown in the table above. They then program the bitrate via the Divisor latch.

The UART has an internal clock oscillating at around 1.8432 MHz. Per bit, at least 16 clock cycles are needed. Therefore, the highest speed of communication via RS232 is 1.8432 MHz / 16 = 115.200 kbps. For lower speeds we have to program the division factor. Imagine we want to set up the UART for 9600 bps. We then have to divide 115200 bps by 12. This factor is stored in the "Divisor Latch" in two bytes (base address +0 and base address +1, with bit 7 of LCR set to 1).

Speed (BPS)	Divisor (Dec)	Divisor Latch High Byte	Divisor Latch Low Byte
50	2304	09h	00h
300	384	01h	80h
600	192	00h	C0h
2400	48	00h	30h
4800	24	00h	18h
9600	12	00h	0Ch
19200	6	00h	06h
38400	3	00h	03h
57600	2	00h	02h
115200	1	00h	01h

D

Table of Commonly Used Baudrate Divisors

The other parameters (parity checking, etc) are programmed via the Line Control Register (LCR) at base address +3, see table C and E. Note again the MSB (bit 7) of this register, which is used to change the function of the first two registers. The other important

Bit 7	1	Divisor Latch Access Bit		
	0	Access to Receiver buffer, Transmitter buffer & Interrupt Enable Register		
Bit 6	Set Break Enable			
Bits 3, 4 And 5	Bit 5	Bit 4	Bit 3	Parity Select
	X	X	0	No Parity
	0	0	1	Odd Parity
	0	1	1	Even Parity
	1	0	1	High Parity (Sticky)
	1	1	1	Low Parity (Sticky)
Bit 2	Length of Stop Bit			
	0	One Stop Bit		
	1	2 Stop bits for words of length 6,7 or 8 bits or 1.5 Stop Bits for Word lengths of 5 bits.		
Bits 0 And 1	Bit 1	Bit 0	Word Length	
	0	0	5 Bits	
	0	1	6 Bits	
	1	0	7 Bits	
	1	1	8 Bits	

E

: Line Control Register

bits are 3, 4 and 5 (parity), bit 2 (stop bits) and bits 0 and 1 (word length). For instance, to set up the device for "9600 N 8 1" we do the following

* set bit 7 of LCR to 1

  port [BA + 3] := port [BA + 3] OR 128

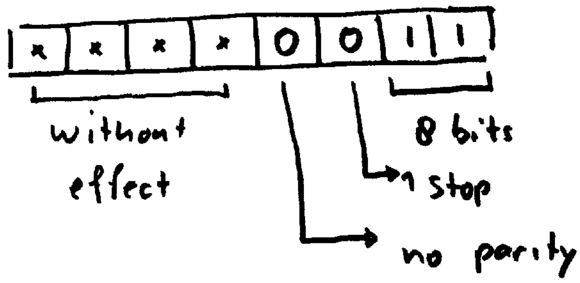
* write divisor latch bytes

  port [BA] := 12;

  port [BA + 1] := 0;

* write LCR

  port [BA + 3] := 3;



⊛ set bit 7 of LCR back to 0 (in order to have access to data transmitted)

```
port [BA+3] := port [BA+3] AND 128;
```

Note that here we used PASCAL code. In other languages the commands are similar

port [x] means byte at address x of memory

Now we are ready to send or receive data. When sending a character, we first have to check if the last character was sent. In other

Bit	Notes
Bit 7	Error in Received FIFO
Bit 6	Empty Data Holding Registers
Bit 5	Empty Transmitter Holding Register
Bit 4	Break Interrupt
Bit 3	Framing Error
Bit 2	Parity Error
Bit 1	Overrun Error
Bit 0	Data Ready

Line Status Register

F

words, if the output buffer is empty. This is bit 6 of the Line Status Register (LSR), see table F. When ready, we can place a character in the Transmitter Holding Buffer (base address + 0, see table C).

```
(*)   repeat
      until (port [BA + 5] AND 64) = 64;
      port [BA] := Ord (c);
```

(c is character to send)

Reading is a similar process. We have to wait until a byte is ready (bit ø of LSR):

```
(*)   repeat
      until (port [BA + 5] AND 1) = 1;
      c := Chr ( port [BA] );
```

For completeness sake, the next page shows some more registers of the UART.

Bit	Notes
Bit 7	Carrier Detect
Bit 6	Ring Indicator
Bit 5	Data Set Ready
Bit 4	Clear To Send
Bit 3	Delta Data Carrier Detect
Bit 2	Trailing Edge Ring Indicator
Bit 1	Delta Data Set Ready
Bit 0	Delta Clear to Send

Modem Status Register

Bit	Notes
Bit 7	Reserved
Bit 6	Reserved
Bit 5	Autoflow Control Enabled (16750 only)
Bit 4	LoopBack Mode
Bit 3	Aux Output 2
Bit 2	Aux Output 1
Bit 1	Force Request to Send
Bit 0	Force Data Terminal Ready

Modem Control Register

Bit	Notes
Bit 7	Reserved
Bit 6	Reserved
Bit 5	Enables Low Power Mode (16750)
Bit 4	Enables Sleep Mode (16750)
Bit 3	Enable Modem Status Interrupt
Bit 2	Enable Receiver Line Status Interrupt
Bit 1	Enable Transmitter Holding Register Empty Interrupt
Bit 0	Enable Received Data Available Interrupt

Interrupt Enable Register

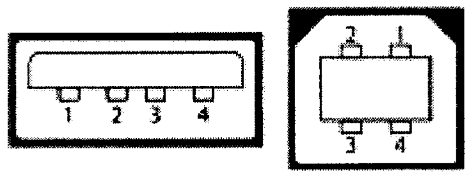
All tables taken from internet pages at

http://www.beyondlogic.org/serial/serial.html

Other serial interfacing standards:

USB Universal Serial Bus

USB logo



USB A and USB B connectors

USB is similar to the RS232 standard with 3 wires (TD, RD e GND)

The connector is a 4-pin as shown above

The fourth pin is used to supply power to the connected devices

pin	function
1	power (V ~ 5 V)
2	D -
3	D +
4	GND

The D-/D+ pair (the "data pair") is a set of twisted pair cables in which the data on the same line is going in both directions, thus differing from the RS232 standard where a pin is always or sending or receiving data.

Other features of USB :

* up to 127 devices can be connected to a host
* USB 2.0 up to 480 Mbits/s

connection	speed
Lo	1.5 Mbits/s
Full	12 Mbits/s
Hi	480 Mbits/s

Since more than 2 devices are interconnected and the same data line (D-/D+) is used for sending and receiving, a much more complicated protocol for communication is needed.

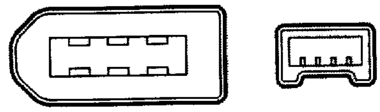
Firewire

This is another serial interfacing standard, also known as i.Link (Sony) or IEEE 1394 and originates from Apple.

FireWire Logo

It is similar to USB, but has some differences :

* up to 63 devices connected to host



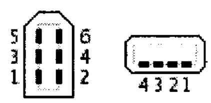
6 pin and 4-pin Firewire connectors.

* up to 800 Mbits/s

* power up to 45 watts (6 pins)

* peer-to-peer (no CPU needed for communication between two devices, say, a camera and a printer)

Both USB and FireWire can make connections while the equipment is switched on.



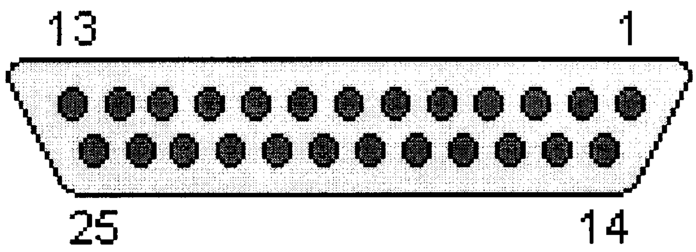
4-pin connector	6-pin connector	Signal	Color	Description
	1	VCC	white	+30V unregulated DC
	2	GND	black	Ground
1	3	TPB-	orange	Twisted pair B
2	4	TPB+	blue	Twisted pair B
3	5	TPA-	red	Twisted pair A
4	6	TPA+	green	Twisted pair A

* <u>two</u> data lines (two twisted pairs) compared to one in USB.

Parallel Interfacing :

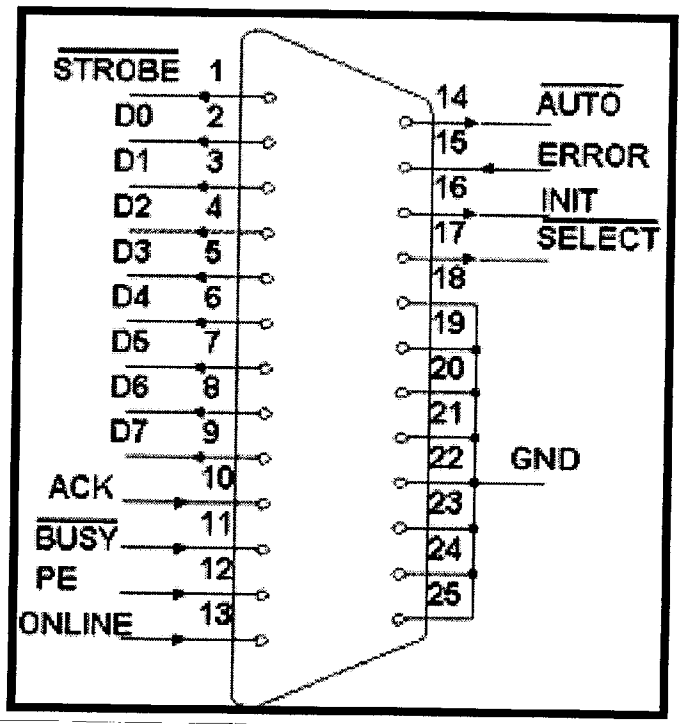
In this type of communications, more than 1 bit is sent at a time. Normally complete 8-bit characters are sent.

Parallel port



As seen at back of computer (female)

This port has 8 datalines (1 byte) that can output data (see D0... D7).



The other pins are used for ground (GND) and handshaking

input handshake :

Ack	Acknowledge
Busy	Printer busy
PE	paper end
SEL	Select
Error	Error

output handshake

SELIN	select in
INIT	Initialize
AUTO	Auto feed

They are TTL signals ("0" = 0V, "1" = 5V)

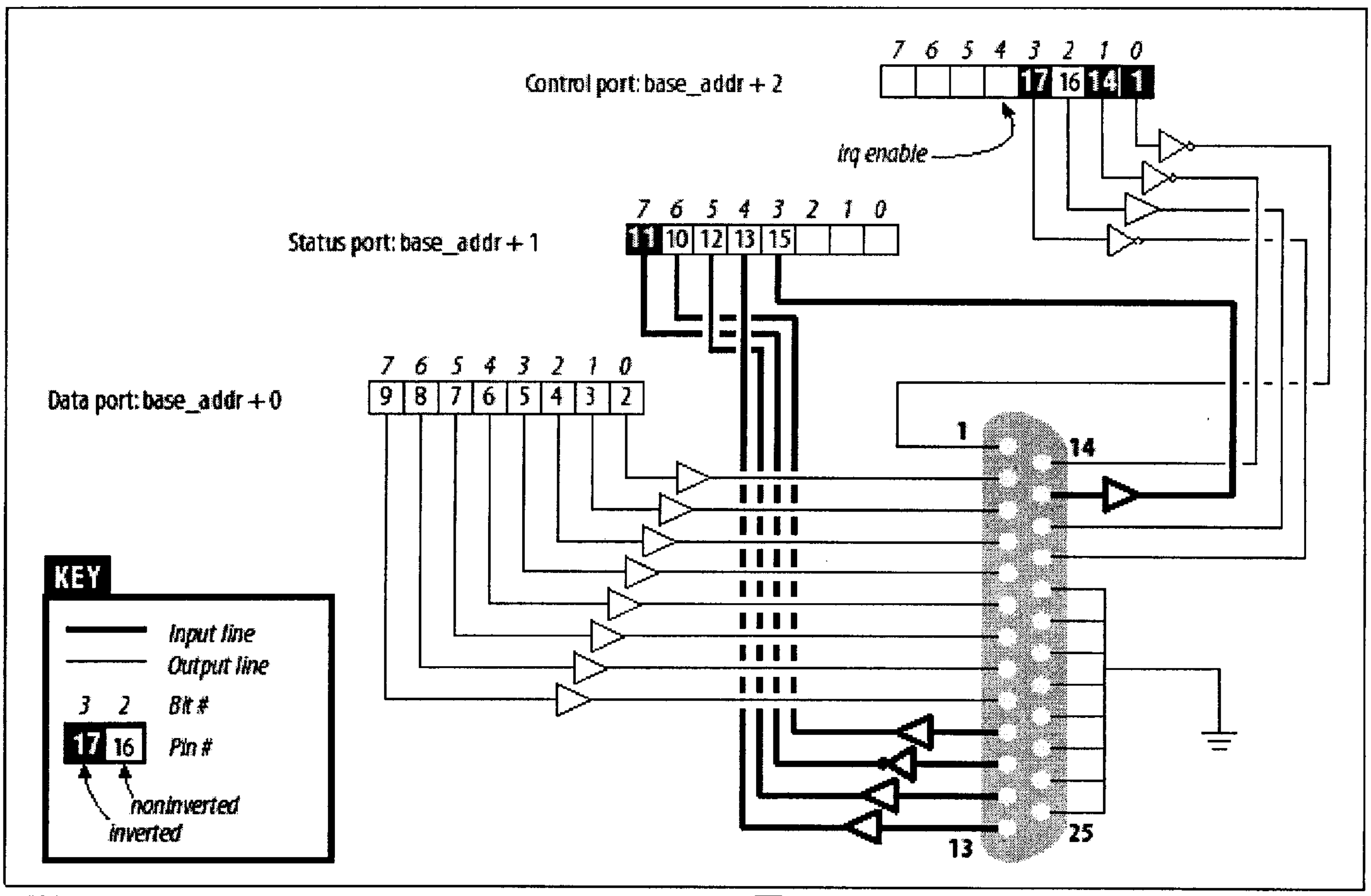
A $\overline{\text{bar}}$ over a signal means it is inverted ("0"/FALSE = 5V, "1"/TRUE = 0 V).

In modern versions, the parallel port can also be used for input. In any case, the parallel port, like the serial port has to be configured, see tables on next page

---

Hardware Layout of parallel port : →
↓

Address		MSB							LSB
	Bit:	7	6	5	4	3	2	1	0
Base	Pin:	9	8	7	6	5	4	3	2
Base+1	Pin:	~11	10	12	13	15			
Base+2	Pin:					~17	16	~14	~1



For example, to see if the printer at LPT1 is busy, we have to read pin 11 of this port. This is connected to bit 7 of the byte at address base_addr+1 of LPT1, which is 379 (hexadecimal)

Relevant tables: (beyondlogic.com)

Pin No (D-Type 25)	Pin No (Centronics)	SPP Signal	Direction In/out	Register	Hardware Inverted
1	1	nStrobe	In/Out	Control	Yes
2	2	Data 0	Out *	Data	
3	3	Data 1	Out *	Data	
4	4	Data 2	Out *	Data	
5	5	Data 3	Out *	Data	
6	6	Data 4	Out *	Data	
7	7	Data 5	Out *	Data	
8	8	Data 6	Out *	Data	
9	9	Data 7	Out *	Data	
10	10	nAck	In	Status	
11	11	Busy	In	Status	Yes
12	12	Paper-Out / Paper-End	In	Status	
13	13	Select	In	Status	
14	14	nAuto-Linefeed	In/Out	Control	Yes
15	32	nError / nFault	In	Status	
16	31	nInitialize	In/Out	Control	
17	36	nSelect-Printer / nSelect-In	In/Out	Control	Yes
18 - 25	19-30	Ground	Gnd		

. Pin Assignments of the D-Type 25 pin Parallel Port Connector.

*: if port is bidirectional, both read and write are possible, like in the COM ports.

Start Address	Function
0000:0408	LPT1's Base Address
0000:040A	LPT2's Base Address
0000:040C	LPT3's Base Address
0000:040E	LPT4's Base Address (Note 1)

- LPT Addresses in the BIOS Data Area;

Address	Notes:
3BCh - 3BFh	Used for Parallel Ports which were incorporated on to Video Cards - Doesn't support ECP addresses
378h - 37Fh	Usual Address For LPT 1
278h - 27Fh	Usual Address For LPT 2

Port Addresses

DATA:

Offset	Name	Read/Write	Bit No.	Properties
Base + 0	Data Port	Write (Note-1)	Bit 7	Data 7
			Bit 6	Data 6
			Bit 5	Data 5
			Bit 4	Data 4
			Bit 3	Data 3
			Bit 2	Data 2
			Bit 1	Data 1
			Bit 0	Data 0

Data Port

Note: if port is bidirectional, read and write are possible

STATUS:

Offset	Name	Read/Write	Bit No.	Properties
BASE +1	STATUS PORT	READ	7	Busy
			Bit 6	Ack
			Bit 5	Paper Out
			Bit 4	Select In
			Bit 3	Error
			Bit 2	IRQ (Not)
			Bit 1	Reserved
			Bit 0	Reserved

Status Port

CONTROL:

Offset	Name	Read/Write	Bit No.	Properties
Base + 2	Control Port	Read/Write	Bit 7	Unused
			Bit 6	Unused
			Bit 5	Enable Bi-Directional Port
			Bit 4	Enable IRQ Via Ack Line
			Bit 3	Select Printer
			Bit 2	Initialize Printer (Reset)
			Bit 1	Auto Linefeed
			Bit 0	Strobe

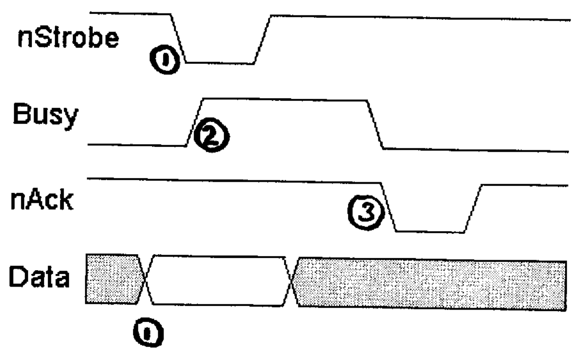
Control Port

* The parallel port, like the serial port is a 1-to-1 communication, with one piece of equipment directly connected to one other piece.

The parallel port has a few handshake lines that control the flow of data

pin	name
1	nStrobe
10	nAck
11	Busy

Centronics Handshake



① When the terminal (computer) is ready to send data to the printer, the terminal puts the data on the data lines (for example write in address $378) and then generates a STROBE pulse to the printer (bit ∅ of base_address+2)

② when the printer sees the strobe it responds with a BUSY signal (bit 7 of base_address+1)

③ When printer has read the data it puts the ACKNLG signal (bit 6 of base_address+1). We can continue sending data

④ If the printer buffer is full the printer keeps the Busy signal asserted.

("n" means inverted logic, 0 V = "TRUE")

GPIB

GPIB stands for General Purpose Interface Bus and is another name for the HPIB, named by the inventors, Hewlett-Packard. It has also been made a standard, IEEE 488 by the Institute of Electrical and Electronics Engineers.

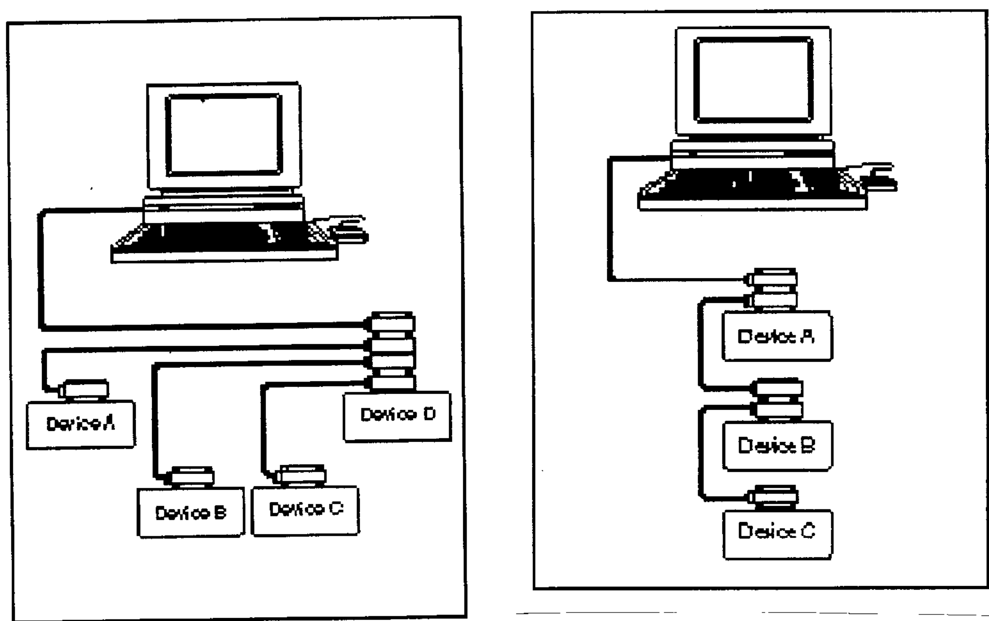
IEEE standard	Description
488	GPIB
802	WiFi
1284	Parallel Port
1294	USB
1394	Fire Wire

The GPIB is a standard of parallel interfacing which differs from a parallel port in that

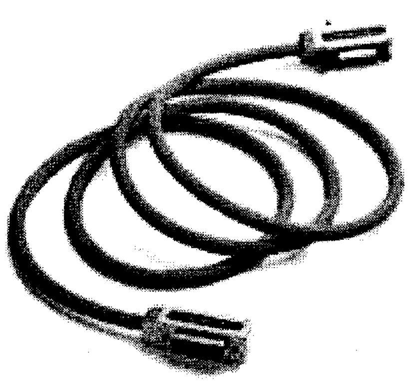
* GPIB can connect up to 15 devices to the host
* 1 MByte/second
* twisted pair signals
* specific instrument hand shaking, like

a SRQ (service request) which means that an instrument needs attention (to be serviced) somehow, for instance because it has finished a measurement.

GPIB is designed with scientific instrumentation in mind. Most scientific equipment (multimeters) etc) come equiped with a GPIB interface, although modern equipment is more likely to have USB interfaces.

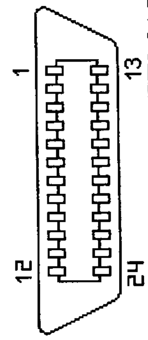


ways to connect GPIB devices
a) star, b) line

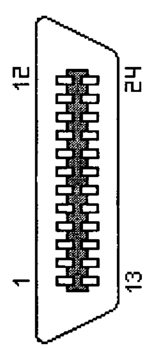


GPIB cable

PIN	Signal	Description
1	Data I/O 1	
2	Data I/O 2	
3	Data I/O 3	
4	Data I/O 4	
5	EOI	End Or Indentify has two uses. EOI is asserted on the last byte of a data transfer. This signals all devices that no more data should be expected on the transfer.
6	DAV	Data Valid is a handshake line indicating that the active talker has placed data on the data lines.
7	NRFD	Not Ready For Data is a handshake line indicating that one or more active listeners is not ready for more data. Note the active talker should then wait before sending any more data on the bus.
8	NDAC	Not Data Accepted is a handshake line indocating that one or more active listeners has not accepted the current data byte. Note the active talker should leave the current byte asserted on the data lines until it has been accepted ot timed out.
9	IFC	Interface Clear is under the exclusive control of the system controller. When it is active high all devices on the bus are returned to an idle state and the bus is cleared
10	SRQ	Service Request can be set by a devive on the interface to indicate it is in need of service. SRQ could be set at the completion of a task. E.g. finished doing a measurement, or when an error as occurred.
11	ATN	Attention is used primarily to differentiate between command mode and data mode. When ATN is TRUE (I.E. Active high) information on the bus is a command and when ATN is FALSE (Active LOW) the information on the bus is data.
12	SHIELD GROUND	(Earth)
13	Data I/O 5	
14	Data I/O 6	
15	Data I/O 7	
16	Data I/O 8	
17	REN	Remote Enable may be set by the system controller to allow other devices to operate in remote mode.
18	P/O Twisted pair with 6	
19	P/O Twisted pair with 7	
20	P/O Twisted pair with 8	
21	P/O Twisted pair with 9	
22	P/O Twisted pair with 10	
23	P/O Twisted pair with 11	
24	SIGNAL GROUND	



IEEE 488 24 PIN FEMALE CONNECTOR



IEEE 488 24 PIN MALE CONNECTOR

Standard computers (PC's) do not come equiped with a GPIB interface and such interfaces have to be bought seperately. For this reason no standard programming of the GPIB interface exists; every interface comes with its own method of controlling it.
To make life easier, the manufacturer normally supplies a "driver" with it (meaning the software in a low-level language to control it), often together with a high-level support routines in a modern language, such as C, PASCAL, or python, etc. An example of such a "library" is given on the next page, the description of PASCAL routines for a specific GPIB interface. Via these routines, a general PASCAL program has access to the code of the driver once these have been loaded into memory

```
UNIT ieeepas;
{
        Turbo Pascal UNIT to interface IEEE-488 subroutines

        Copyright (C) 1989 Capital Equipment Corp.
}

{------------------------------------------------------------------}
INTERFACE
{------------------------------------------------------------------}

PROCEDURE IEEEInitialize (addr,level : integer);
PROCEDURE IEEEtransmit (cmd : string; var status : integer);
PROCEDURE IEEEreceive (var s : string; maxlen : word; var len : word;
                   var status : integer);
PROCEDURE IEEEsend (addr : integer; s : string; var status : integer);
PROCEDURE IEEEenter (var s : string; maxlen : word; var len : word;
                 addr : integer; var status : integer);
PROCEDURE IEEEspoll (addr : integer; var poll : byte; var status : integer);
PROCEDURE IEEEppoll (var poll : byte);
PROCEDURE IEEEtarray (var d; count : word; eoi : boolean; var status :
integer);
PROCEDURE IEEErarray (var d; count : word; var len : word; var status :
integer);

FUNCTION IEEEsrq : boolean;

PROCEDURE IEEEsetport (boardnum : integer; ioaddr : word);
PROCEDURE IEEEboardselect (boardnum : integer);
PROCEDURE IEEEdmachannel (c : integer);
PROCEDURE IEEEsettimeout (t : word);
PROCEDURE IEEEsetoutputEOS (eos1,eos2 : byte);
PROCEDURE IEEEsetinputEOS (eos : byte);

PROCEDURE IEEEcommands (var cmds; count : word;var status : integer);
PROCEDURE IEEEatn (atn : boolean; var status : integer);
PROCEDURE IEEEren (ren : boolean);
PROCEDURE IEEEoutp (reg : word;value : byte);
FUNCTION IEEEinp (reg : word) : byte;
PROCEDURE Enable488ex (enable : boolean);
PROCEDURE Enable488sd (enable : boolean; timing : integer);
FUNCTION IEEElistenerPresent (addr : integer) : boolean;
FUNCTION GPIBBoardPresent : byte;
```

The most important ones are IEEEsend and IEEE enter for sending and receiving ascii strings to and from a specific device. Every IEEE device has its own unique "address" (addr), so that only one device will execute the command send (s) or only one device sends a reply (s). "status" contains information of the success of the instruction, "max len" is the maximum length of the string to be sent by the device and "len" is the actual length of the string just sent.

IEEE srq : is there a device that needs attention?

IEEE ren : put device in "remote" (often this disables the front panel buttons of the devices)

GPIB Board Present : is there actually a GPIB board present in the computer? useful at the start of a program.

Other GPIB interface cards + driver will have other instructions. Read the instructions!

Example program based on the IEEE driver of the previous pages

```
PROGRAM TestIEEE;

  { Include the library with routines: }
Uses ieeepas;

Var s: string;
    status: integer;
    len: word;
Const addr=13;

begin
    { check if the GPIB board is present: }
  if (GPIBBoardPresent<>0) then
    begin
      writeln('No GPIB interface found. Program halted');
      halt;
    end;
     { the IFcard also has an IEEE address. }
     { make the IFcard a controler at addr. 21: }
  IEEEInitialize(21, 0);
     { send a command to device at addr 13: }
     {  (ask for its identity)             }
  IEEEsend(13, '*IDN?', status);
     { receive a reply to that question: }
  IEEEenter(s, 255, len, 13, status);
     { show reply on screen: }
  writeln('Equipment at IEEE address 13:');
  writeln(s);
end.
```

The output of the program might be something like

```
Equipment at IEEE address 13:
Oxford ITC601, rev. 5.01
```

LabVIEW

(Laboratory Virtual Instrumentation Engineering Workbench). This is a programming language that is specifically designed for interfacing with (scientific) instruments.

Early programming languages had fully text for design (writing) and the "interface" screen when the program was running. Examples are PASCAL, BASIC, C, C++, etc. More modern programming languages made use of the Windows (and equivalent) graphical environments and had easy tools for designing a graphical interface. The final program was WIMP style (windows - icon - mouse-pointer), while the program itself was still fully written in text. Visual BASIC, Delphi, etc.

LabVIEW is a programming language where also the programming itself is done in a graphical way.

A program thus consists of two parts

1. The design of the GUI (graphical user interface) which has the input and output (to the user).
2. The controlling program, called "Diagram"

The GUI has simple output boxes (text) or indicators (ON/OFF) and even complicated graphs which can directly visualize measurements. It can also have input elements (text boxes &c. numerical values, binary "switches", etc.).

The controlling part is invisible when the programming is running, but is the engine of the program. It reads the inputs of the GUI and processes the data and generates output for the GUI elements.

As an example,

in PASCAL :

```
writeln ('Hello world');
```

in C :

```
printf (" Hello world \n");
```

in Visual BASIC :

```
textbox1.value = "Hello World";
```

(with textbox1 a GUI element)

in LabVIEW

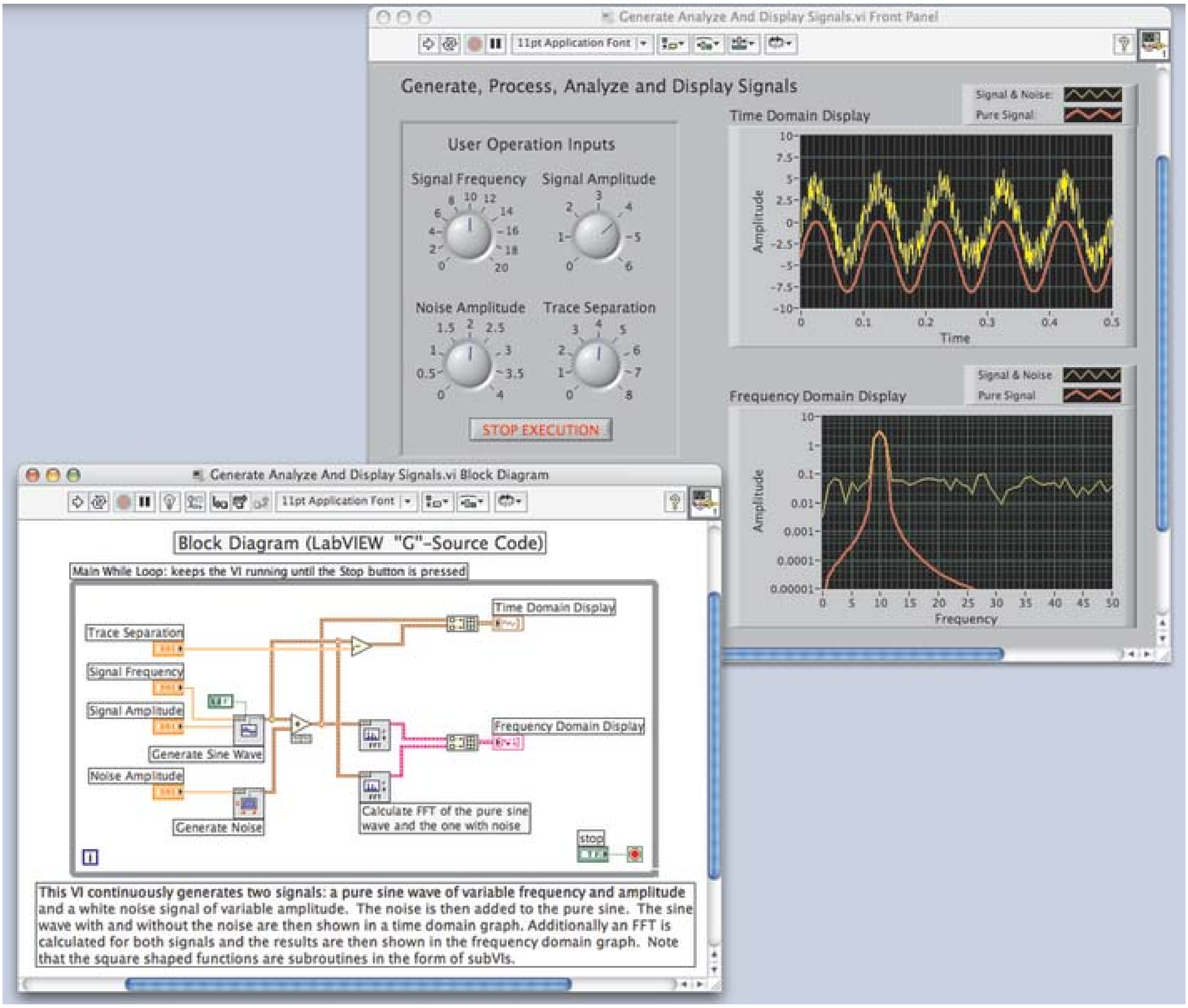
Knowledge of an imperative programming language (for instance PASCAL or C) helps in understanding LabVIEW programming.

There exist many good manuals and tutorials and user guides on the internet. At the practical lectures we will make a small program that communicates with some equipment.

Many companies of equipment supply LabVIEW drivers with their instrument which facilitates programming.

NATIONAL INSTRUMENTS

LabVIEW



LabVIEW example program interface and diagram