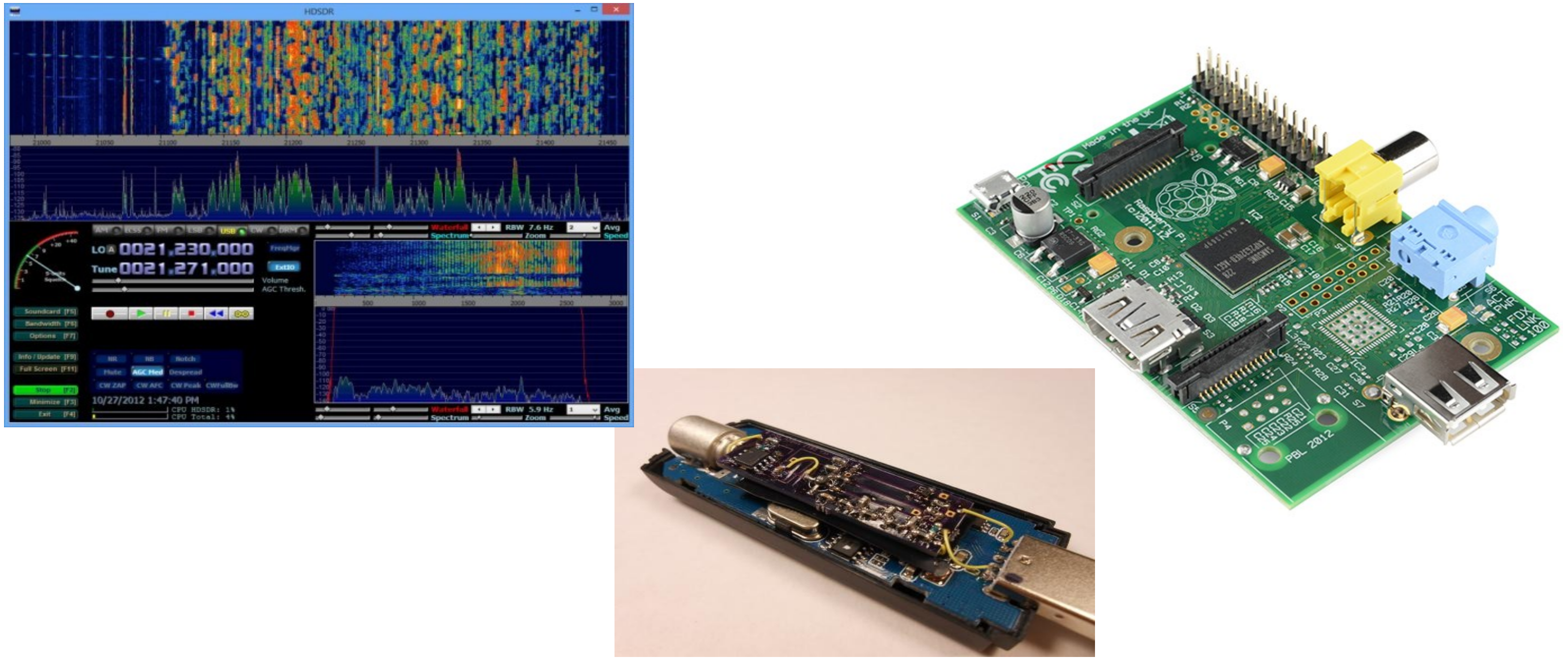
SDR – the Guts

(SDR = Software Defined Radio)

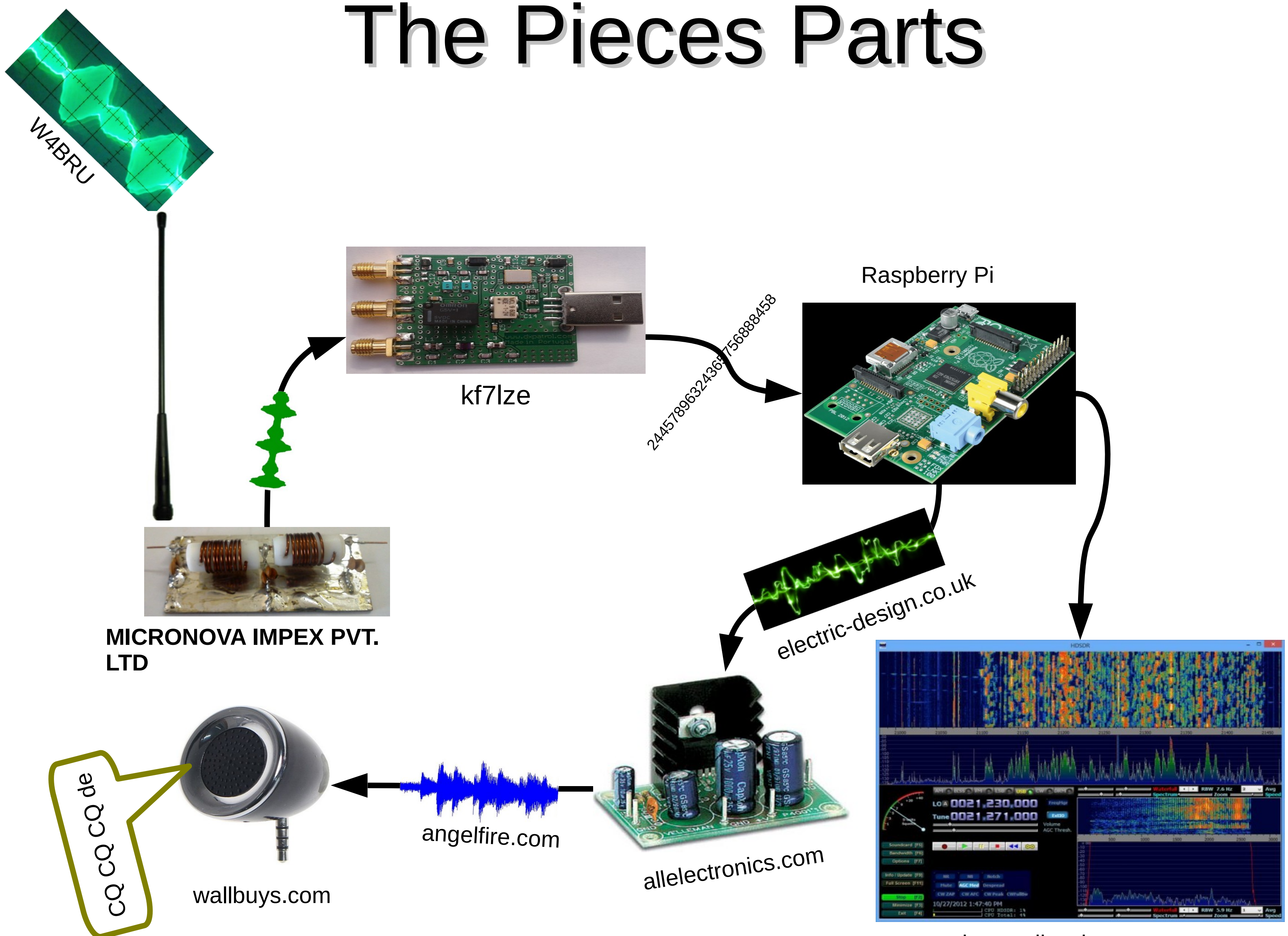
Bruce MacAlister, W4BRU



SDR Topics

- The pieces parts
- Words and abbreviations
- What you know
- The innards

The Pieces Parts



W4BRU

MICRONOVA IMPEX PVT. LTD

kf7lze

24457896324365756888458

Raspberry Pi

electric-design.co.uk

hamradioscience.com

allelectronics.com

angelfire.com

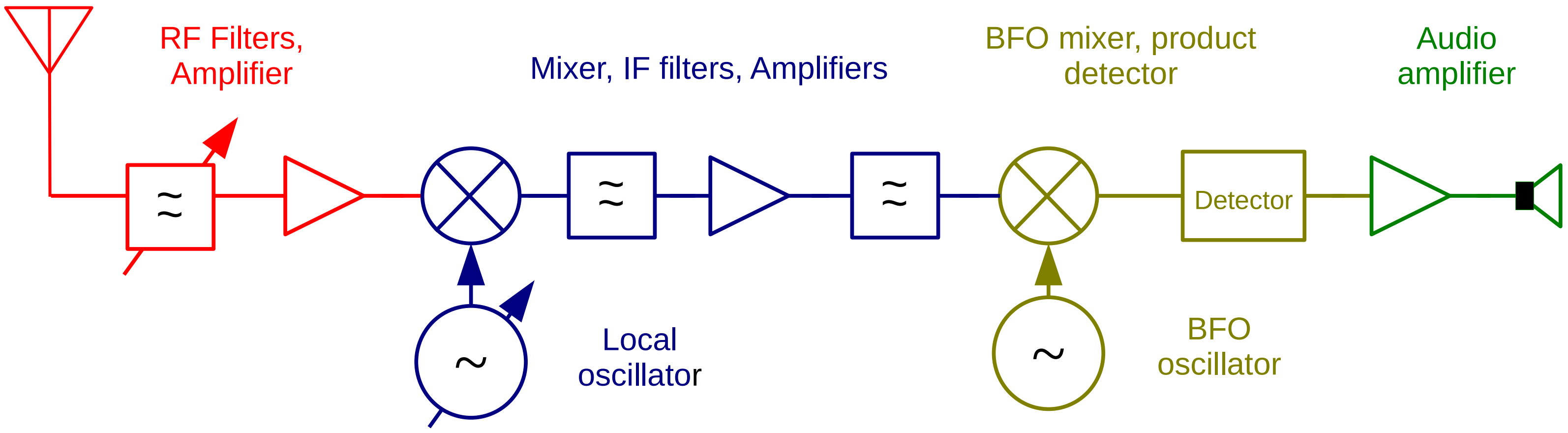
wallbuys.com

CQ CQ CQ de

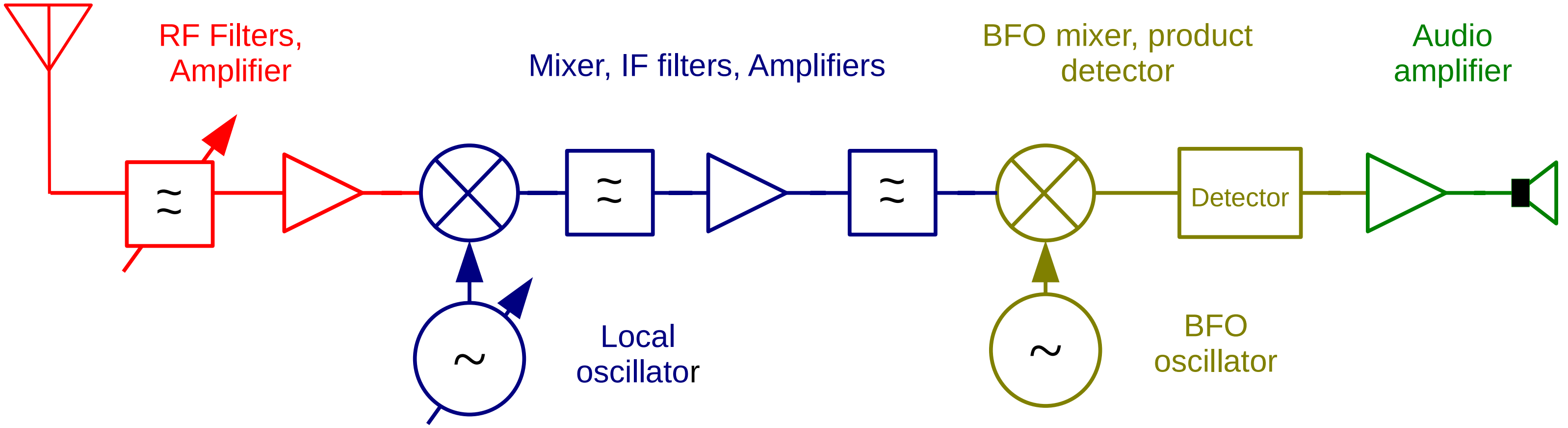
Words & Abbreviations

- SDR = Software Defined Radio
- DSP = Digital Signal Processing used for SDR, manufacturing lines, the NSA, etc
- Downconverter = takes selected segment of RF to baseband
- Baseband = RF signal at some bandwidth an ADC can handle

What We Know: Superheterodyne



What's New with SDR



RF Filters, Amplifier

Mixer, IF filters, Amplifiers

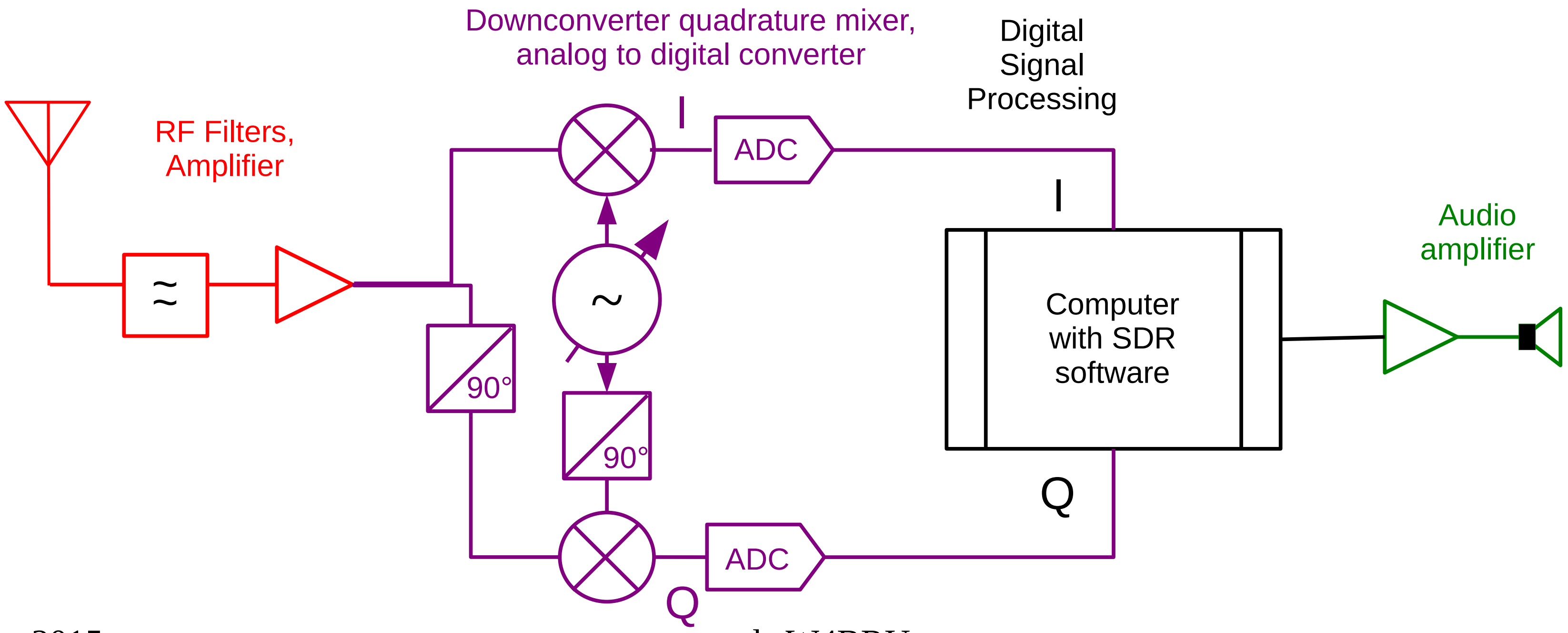
BFO mixer, product detector

Audio amplifier

Detector

Local oscillator

BFO oscillator



Downconverter quadrature mixer, analog to digital converter

Digital Signal Processing

RF Filters, Amplifier

I

ADC

I

Computer with SDR software

Audio amplifier

90°

90°

Q

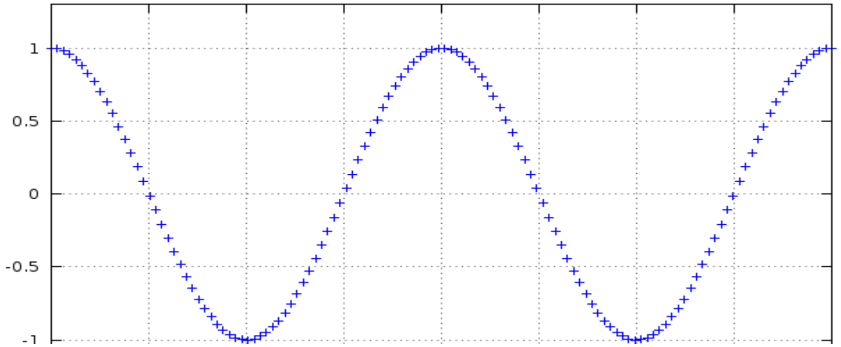
ADC

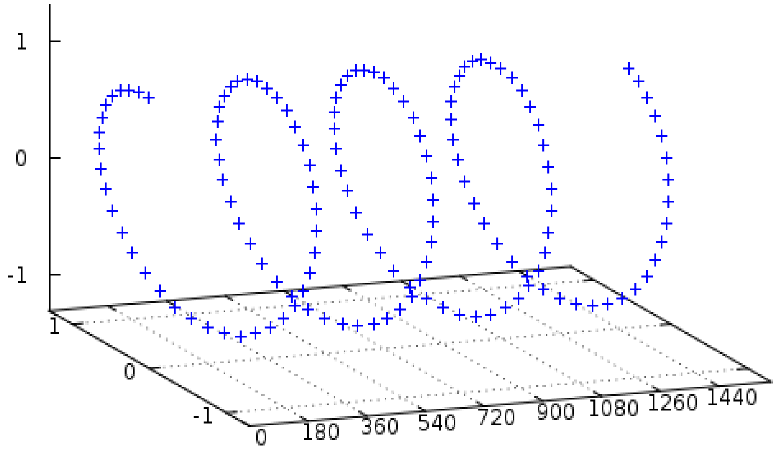
Q

More Words & Abbreviations

- IQ = In-phase and Quadrature-phase versions of the signal
- I = original signal downconverted to baseband
- Q = original signal shifted 90-degrees then downcoverted to baseband
- ADC = Analogue to Digital Converter converts voltage levels to digital numbers for computers
- Digital Filter = computer program that executes algorithms that model analog filters

Why I and Q?

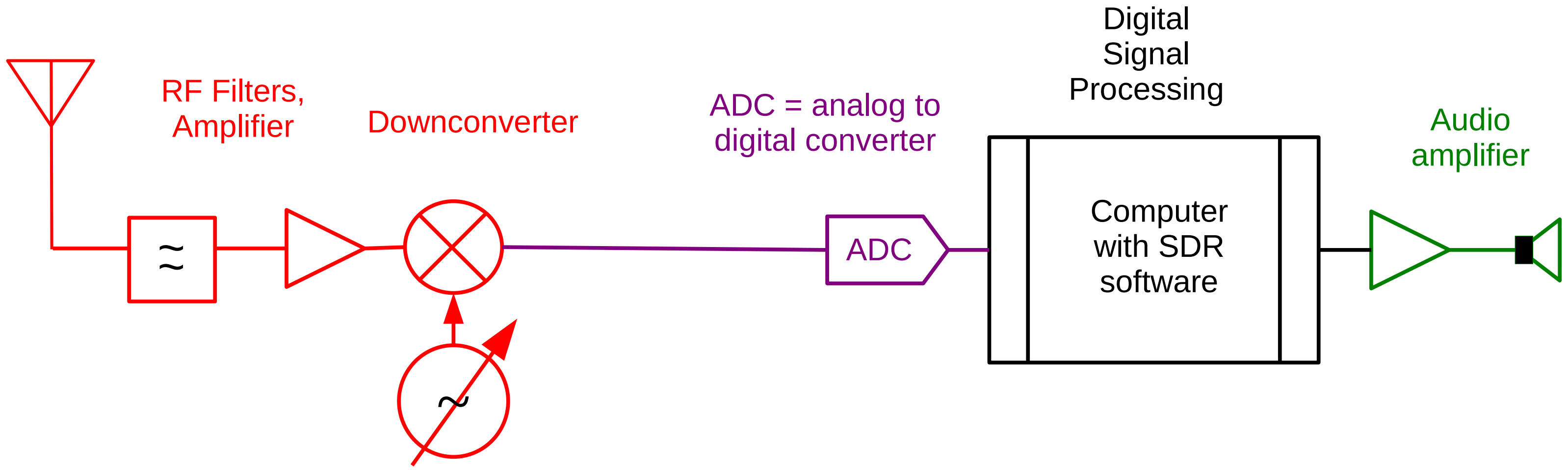




- I = typical waveform
- Q = “side” view
- Together = a visualization of the entire cycle
- Critical for phase modulations (eg, PSK), valuable for others

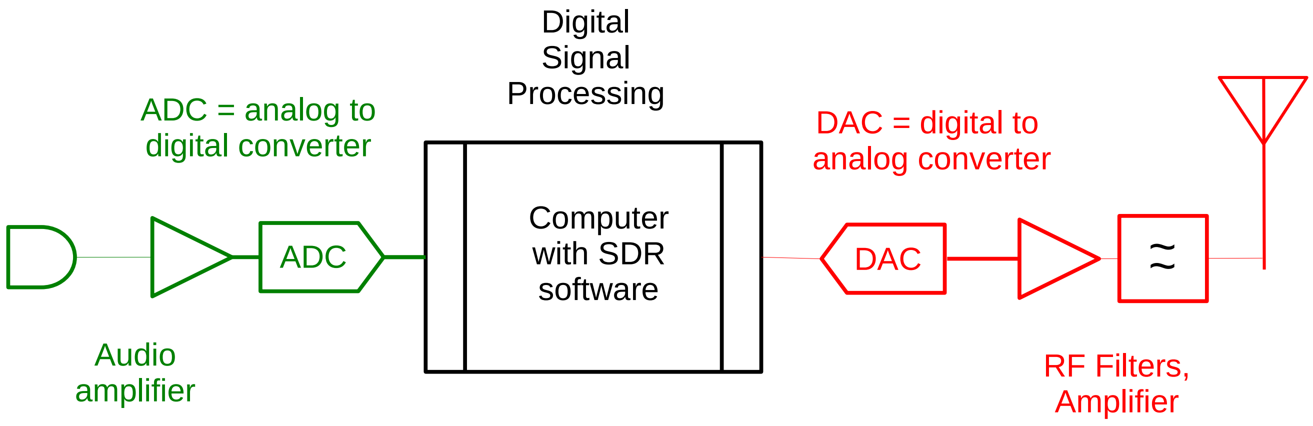
Source: http://whiteboard.ping.se/SDR/IQ

Let the computer do the Q



ADC on I (in phase) do I-Q digitally with computer code

SDR Transmission



Digital Signal Processing

ADC = analog to digital converter

DAC = digital to analog converter

ADC

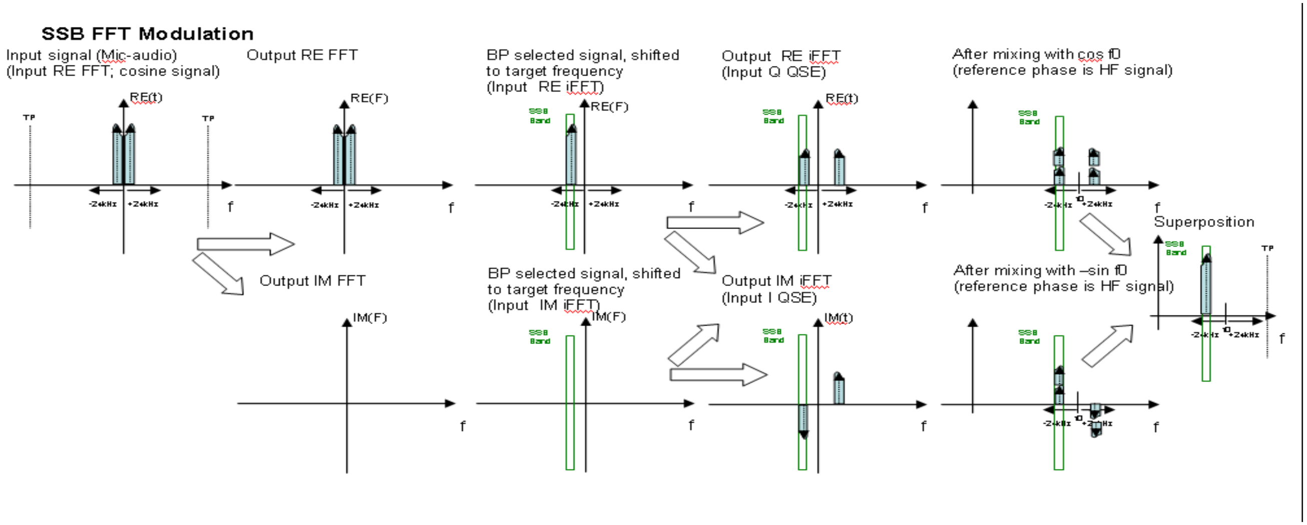
Computer with SDR software

DAC

≈

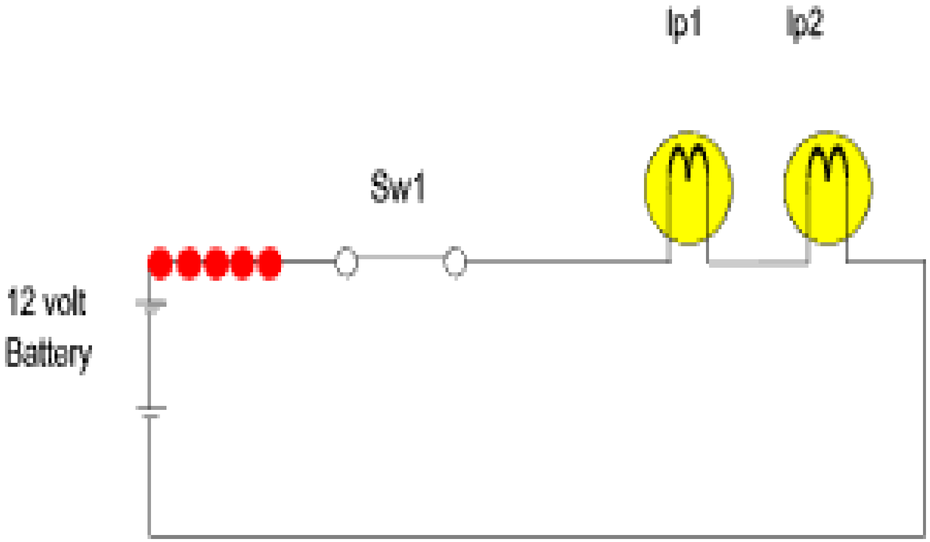
Audio amplifier

RF Filters, Amplifier

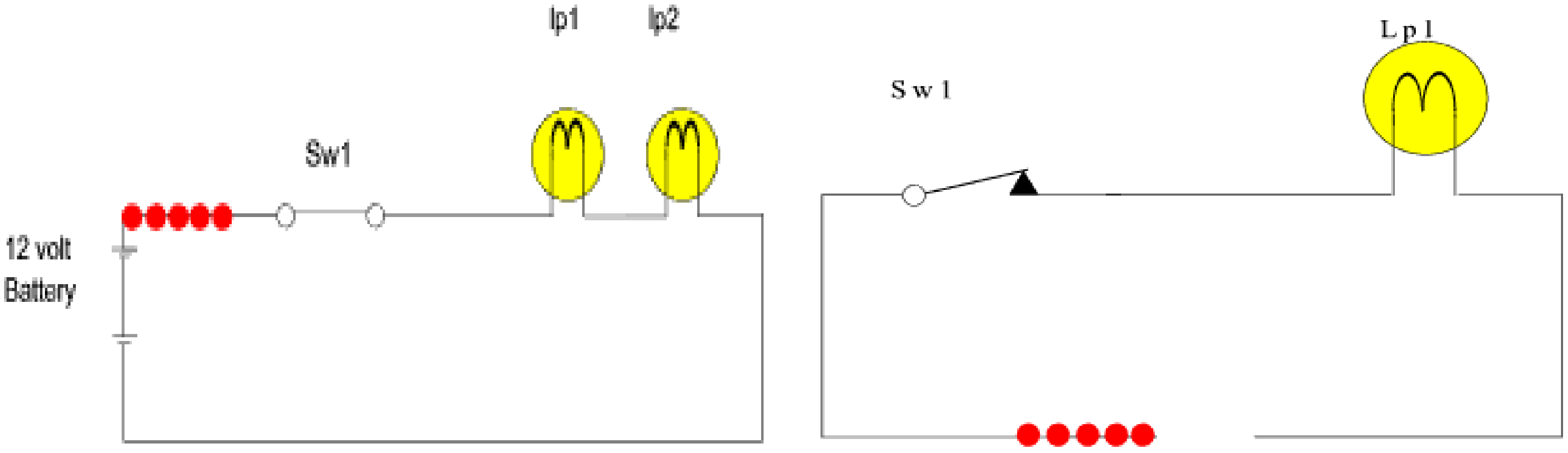


From DG5MK website

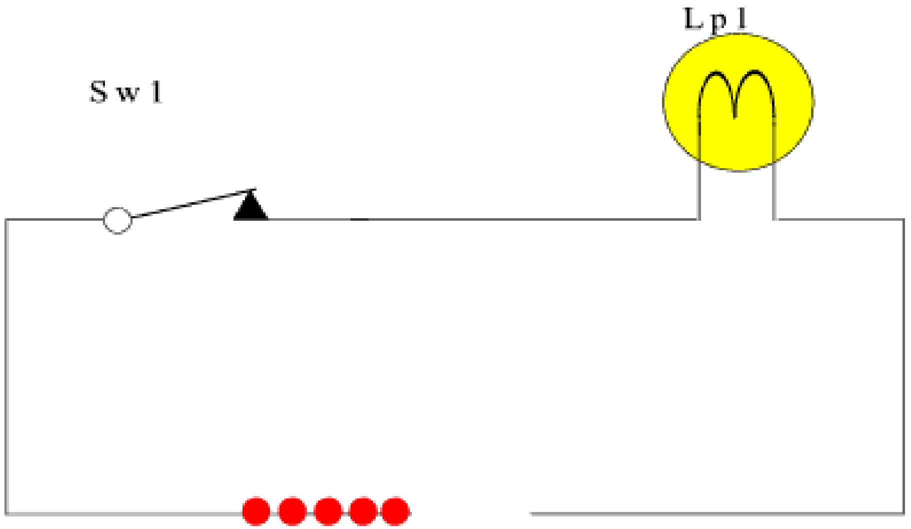
Signal visualizations - Flow



Signal visualizations - Flow



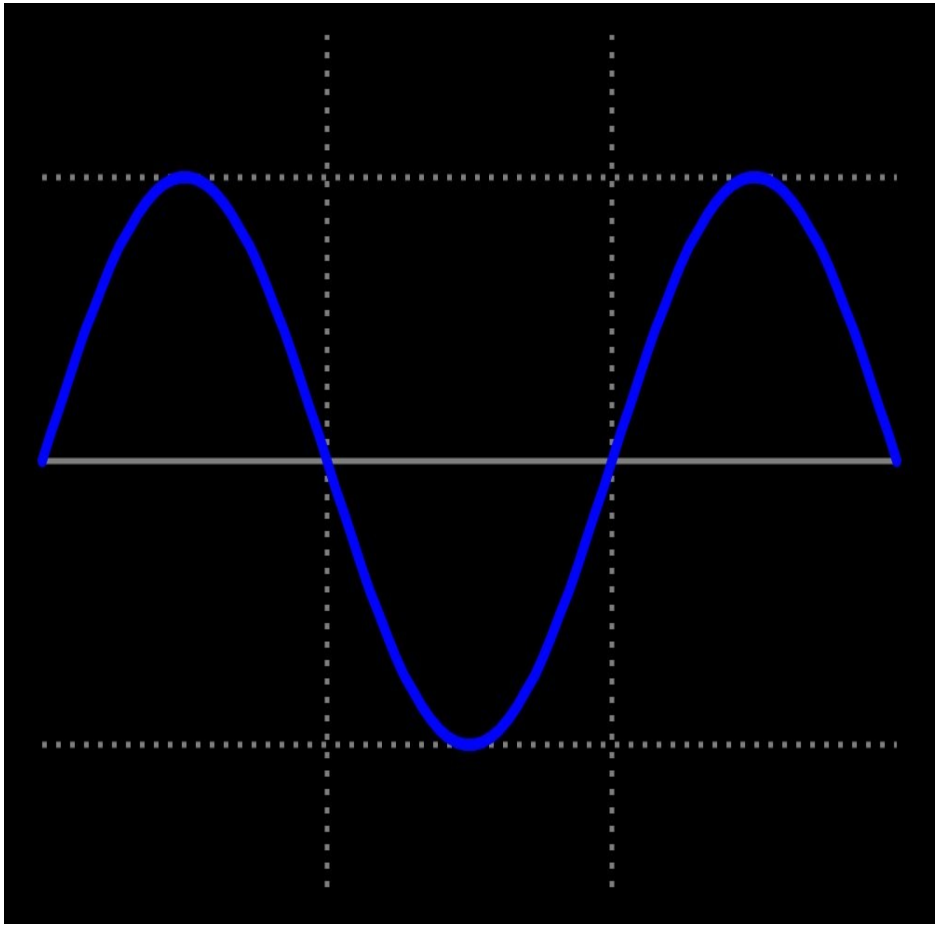
Signal visualizations - Flow



Alternating Power Source

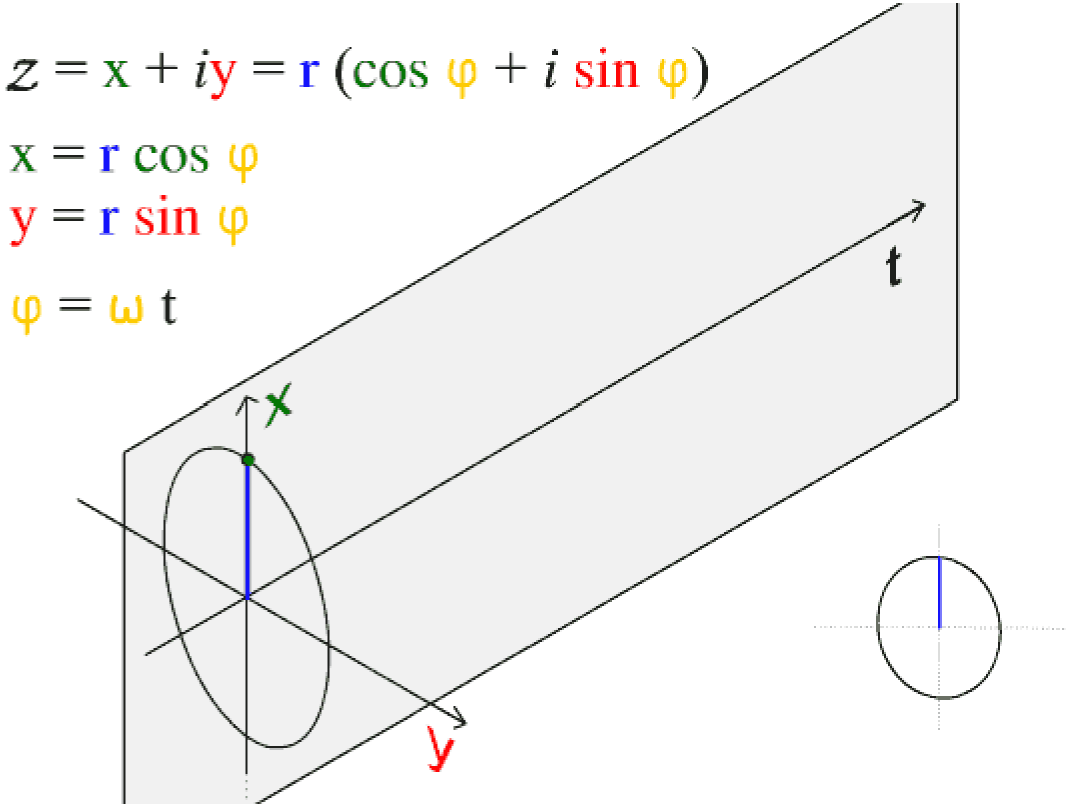
What can you say about this signal?

Characterization of a Signal



What does a sine wave say about the signal?

Characterization of a Signal



$$z = x + iy = r(\cos\varphi + i\sin\varphi)$$

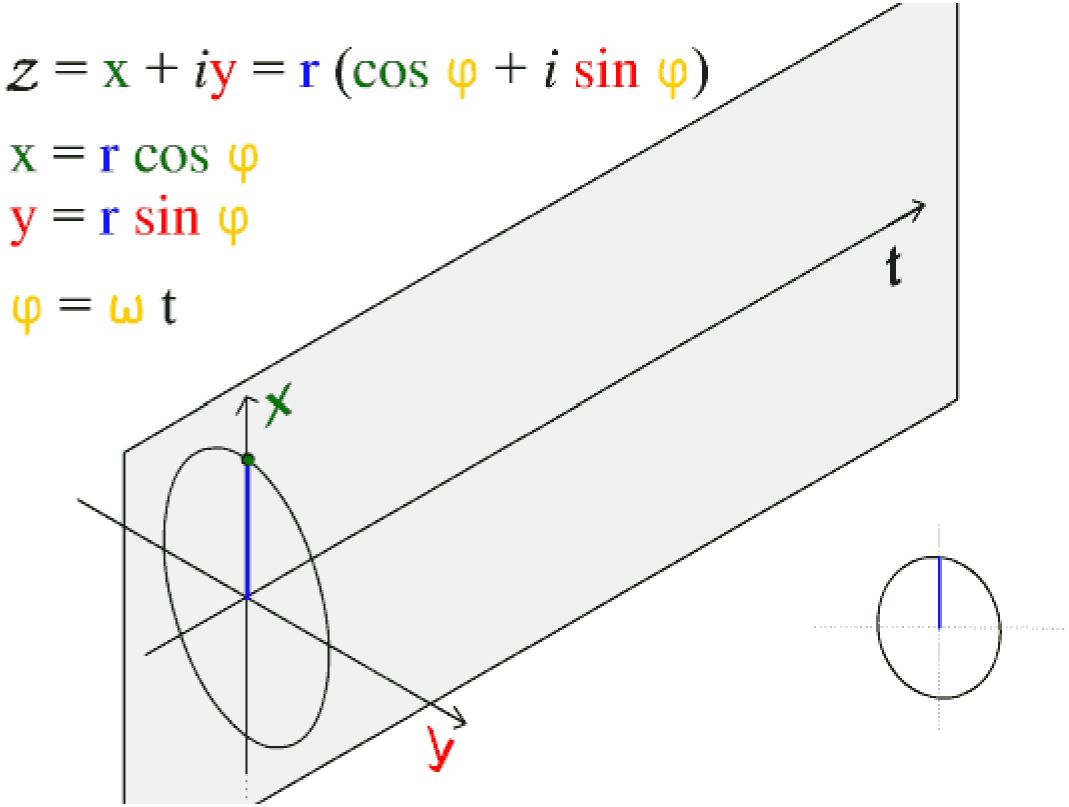
$$x = r\cos\varphi$$

$$y = r\sin\varphi$$

$$\varphi = \omega t$$

What does a phasor chart say about the signal?

Characterization of a Signal



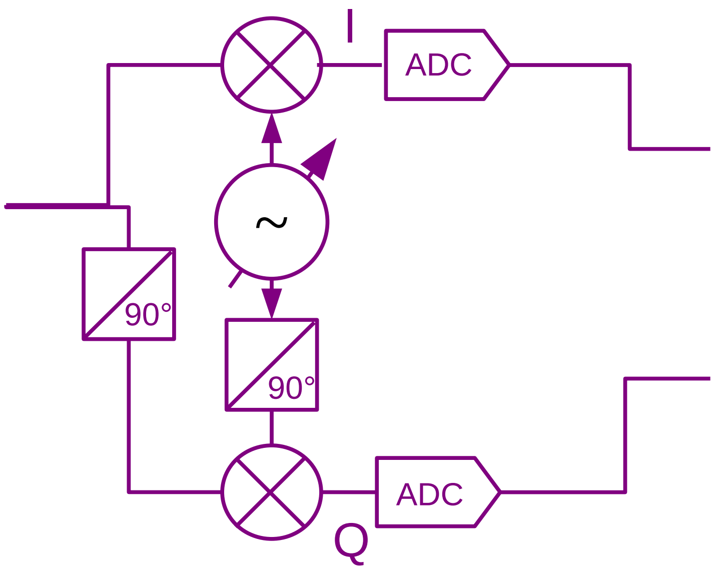
A phase vector (phasor) represents sinusoidal function whose

- amplitude (A)
- frequency (ω)
- phase (θ)

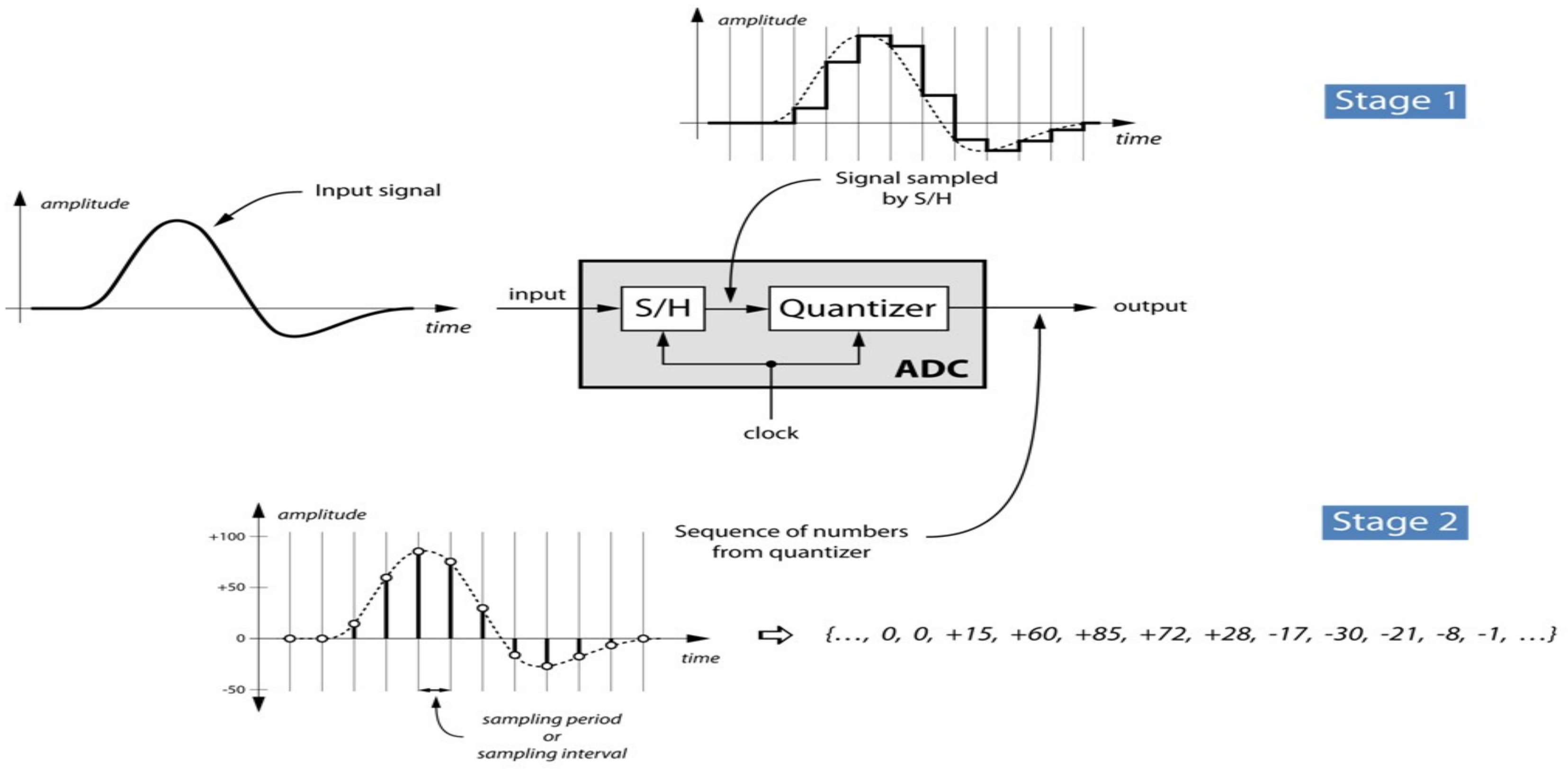
are time-invariant.

Back to Signal Conversion

Downconverter quadrature mixer,
analog to digital converter



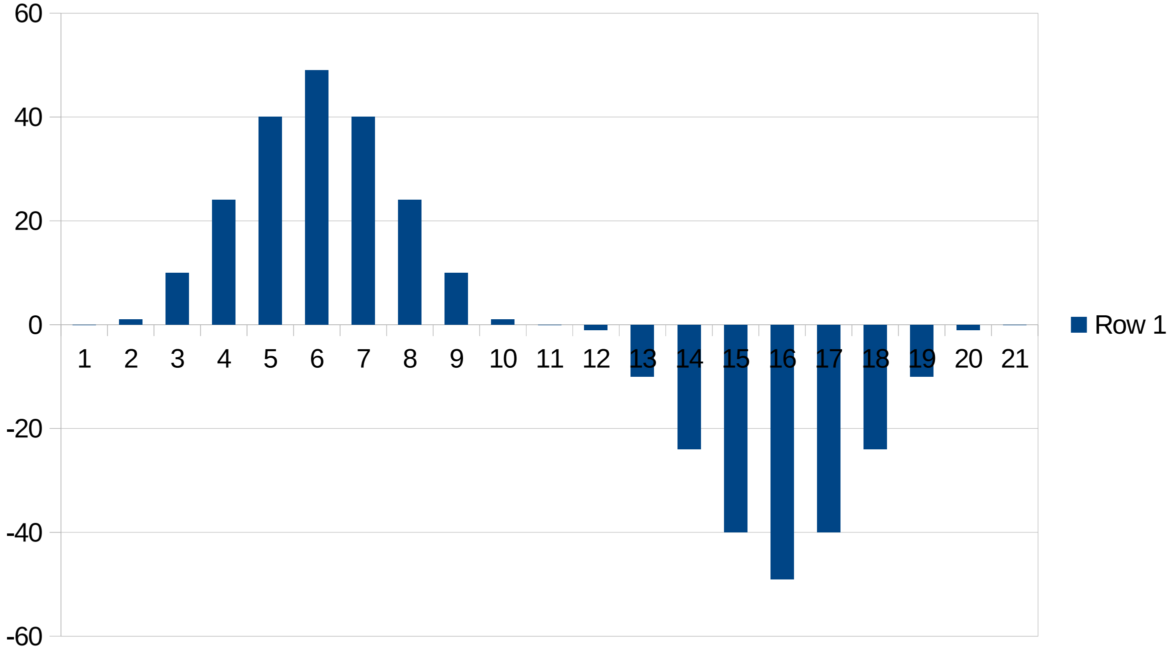
Analog to Digital Converter



Source: Nutaq, Quebec, Canada

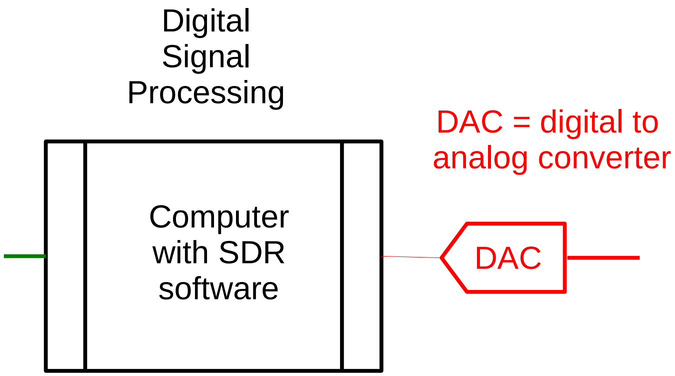
ADC output

0	1	10	24	40	49	40	24	10	1	0	-1	-10	-24	-40	-49	-40	-24	-10	-1	0



Nyquist = sample at least twice the highest frequency expected

Digital to Analog Converter



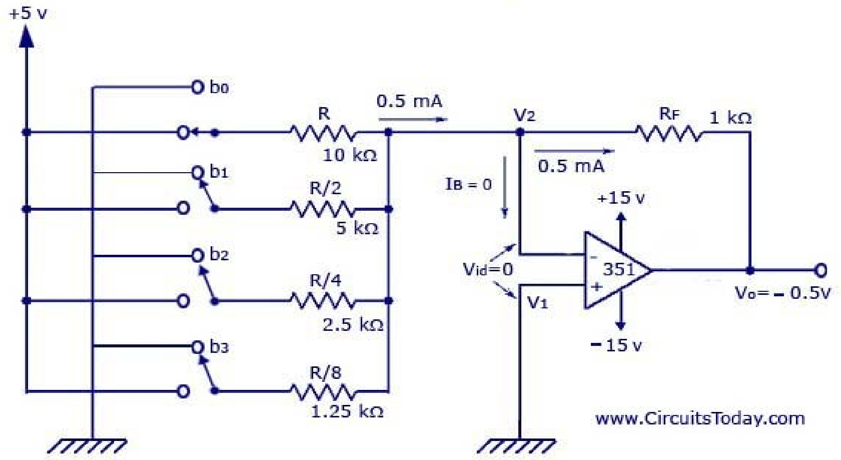
Digital Signal Processing

Computer with SDR software

DAC = digital to analog converter

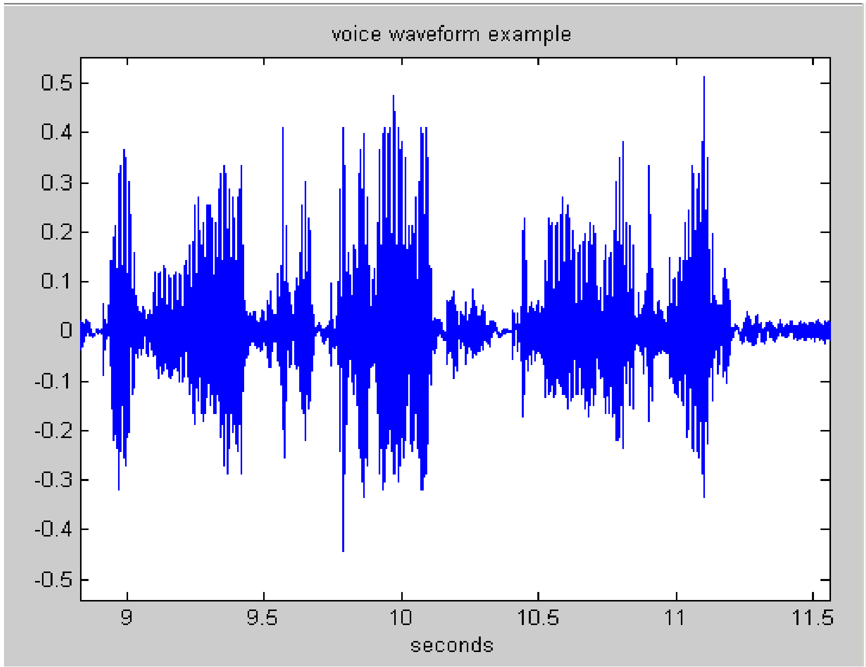
DAC

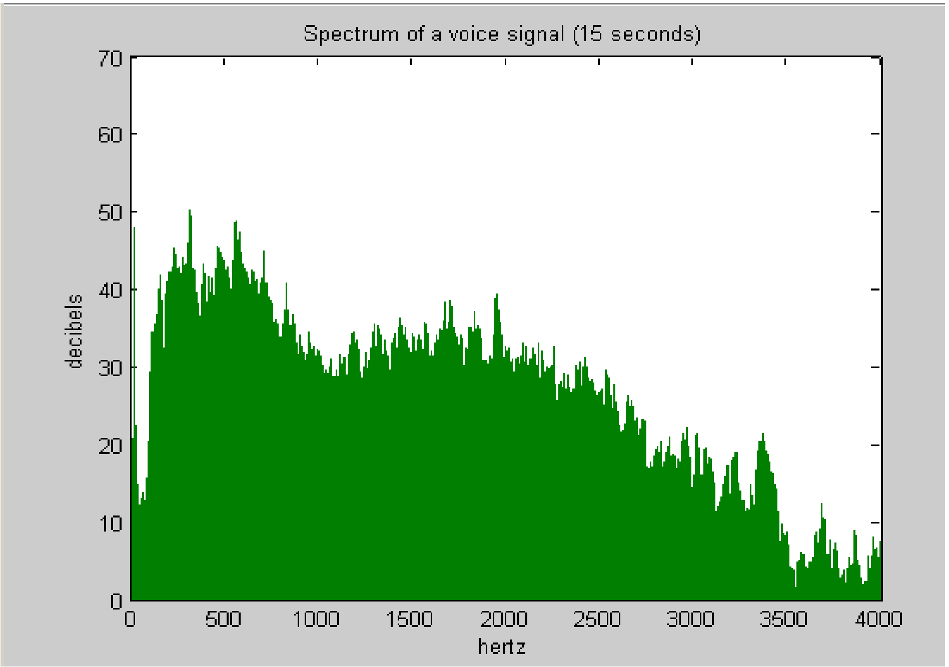
D/A Converter With Binary Weighted Resistors



www.CircuitsToday.com

Fast Fourier

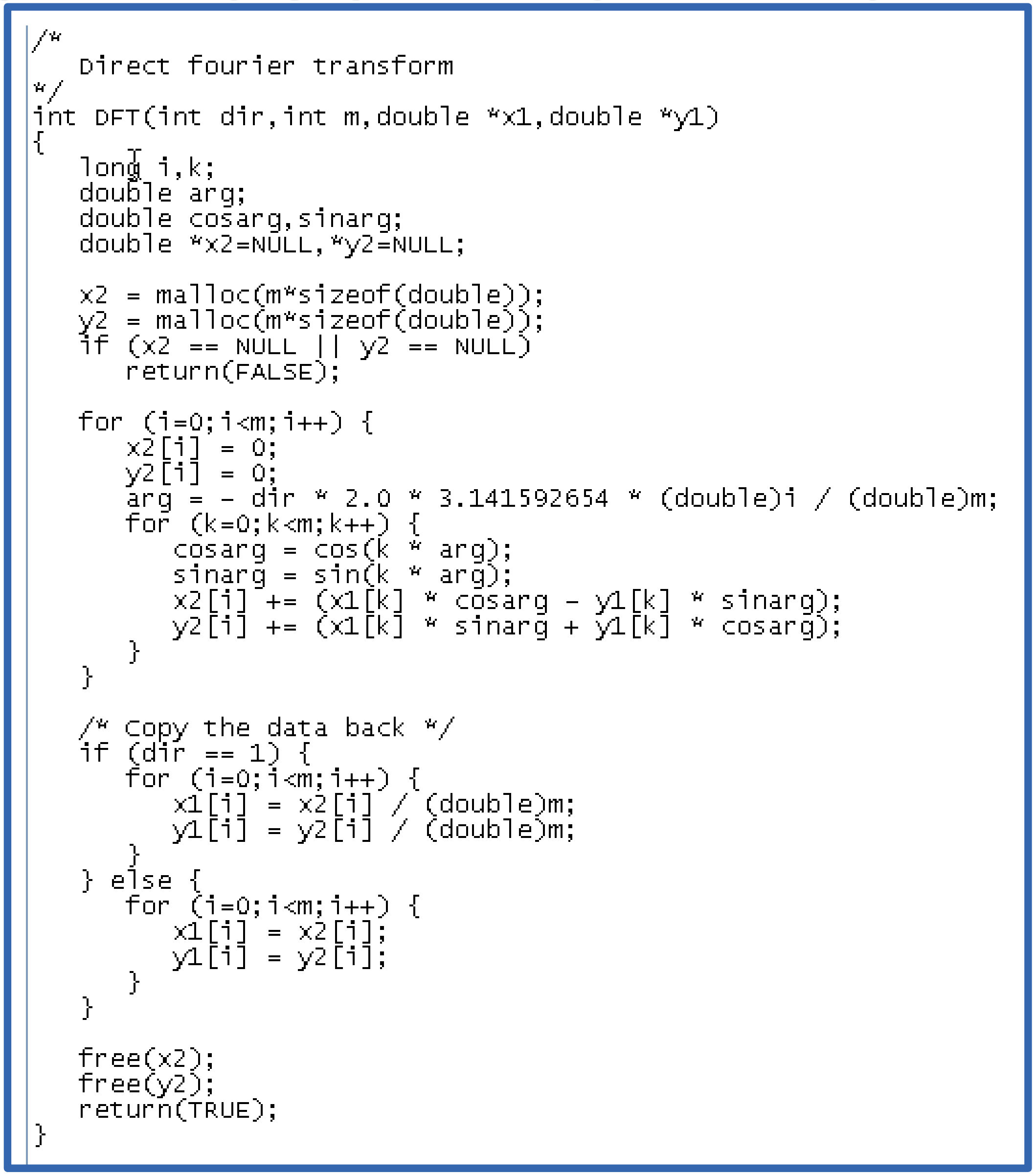




Wikipedia commons

$$X_k = \begin{cases} E_k + e^{-\frac{2\pi i}{N}k} O_k & \text{for } 0 \le k < N/2 \\ E_{k-N/2} + e^{-\frac{2\pi i}{N}k} O_{k-N/2} & \text{for } N/2 \le k < N. \end{cases}$$

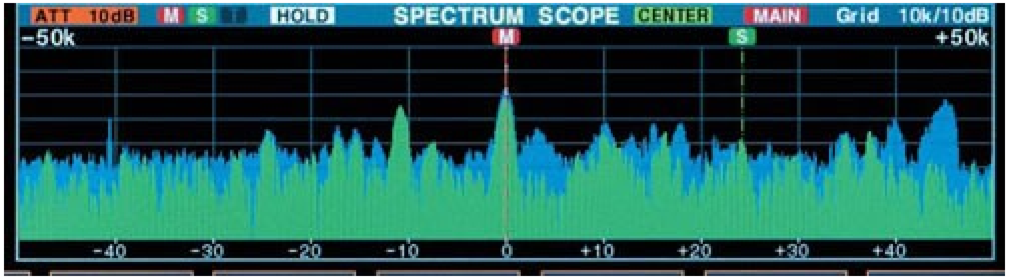
For those who write in C



```
/*
   Direct fourier transform
*/
int DFT(int dir,int m,double *x1,double *y1)
{
   long i,k;
   double arg;
   double cosarg,sinarg;
   double *x2=NULL,*y2=NULL;

   x2 = malloc(m*sizeof(double));
   y2 = malloc(m*sizeof(double));
   if (x2 == NULL || y2 == NULL)
      return(FALSE);

   for (i=0;i<m;i++) {
      x2[i] = 0;
      y2[i] = 0;
      arg = - dir * 2.0 * 3.141592654 * (double)i / (double)m;
      for (k=0;k<m;k++) {
         cosarg = cos(k * arg);
         sinarg = sin(k * arg);
         x2[i] += (x1[k] * cosarg - y1[k] * sinarg);
         y2[i] += (x1[k] * sinarg + y1[k] * cosarg);
      }
   }

   /* Copy the data back */
   if (dir == 1) {
      for (i=0;i<m;i++) {
         x1[i] = x2[i] / (double)m;
         y1[i] = y2[i] / (double)m;
      }
   } else {
      for (i=0;i<m;i++) {
         x1[i] = x2[i];
         y1[i] = y2[i];
      }
   }

   free(x2);
   free(y2);
   return(TRUE);
}
```

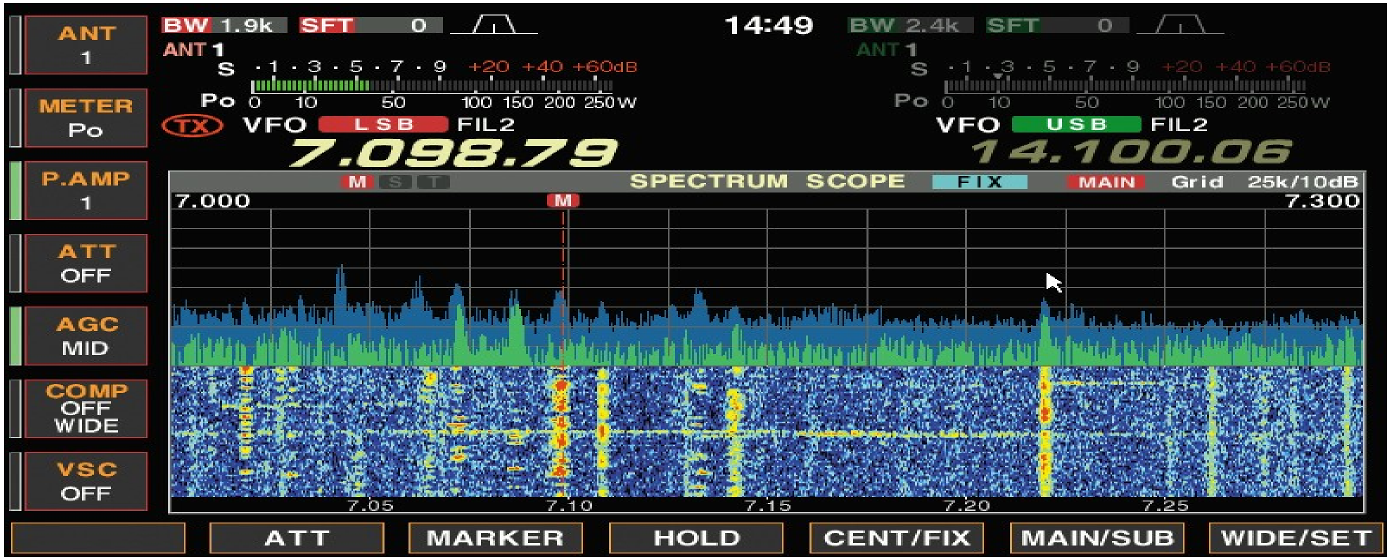
Wikipedia:
Fast Fourier
Transform

Ham radio spectrum display



ATT 10dB M S HOLD SPECTRUM SCOPE CENTER MAIN Grid 10k/10dB

Icom America



ANT 1

METER Po

P.AMP 1

ATT OFF

AGC MID

COMP OFF WIDE

VSC OFF

BW 1.9k SFT 0

14:49

BW 2.4k SFT 0

ANT 1

TX VFO LSB FIL2

7.098.79

ANT 1

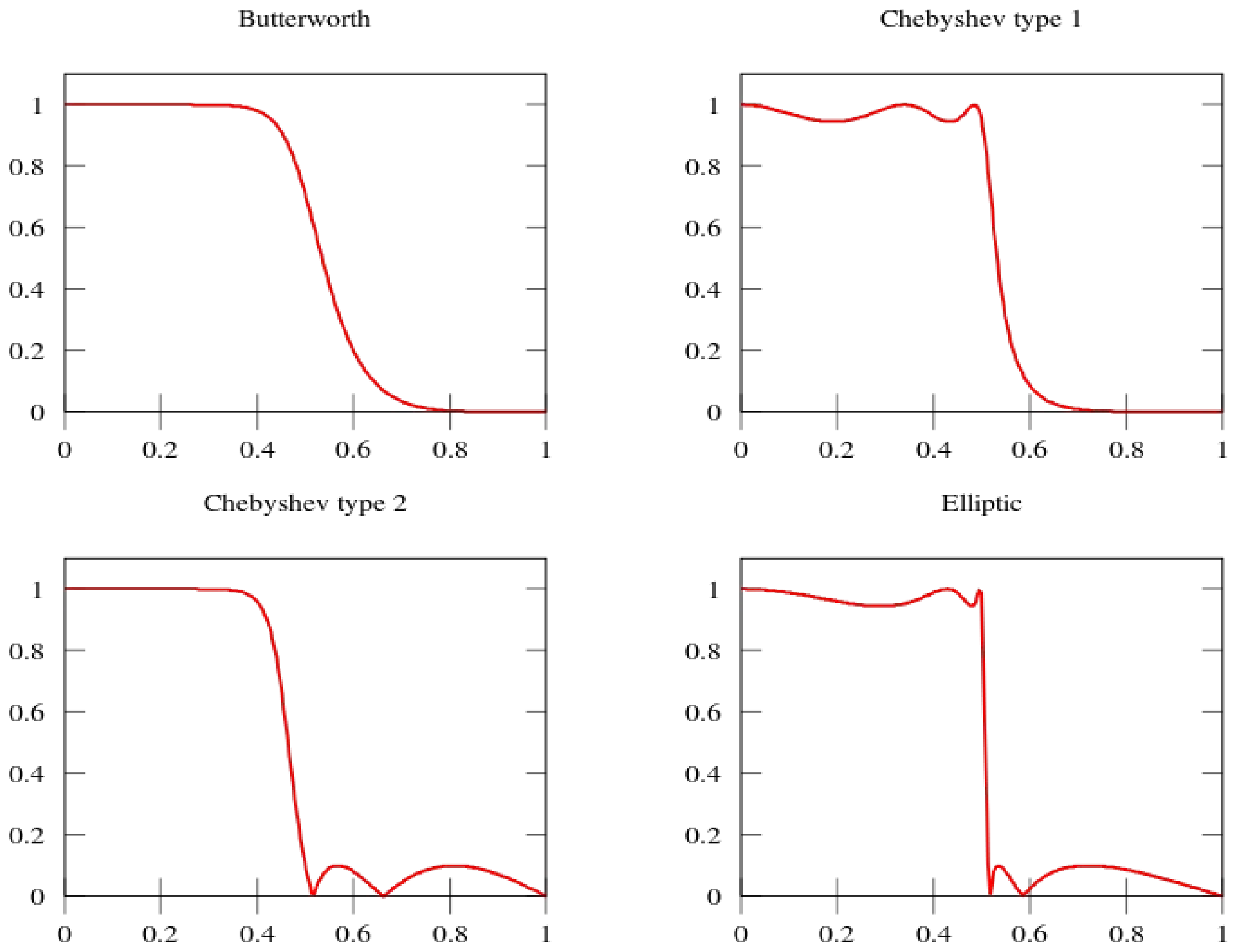
VFO USB FIL2

14.100.06

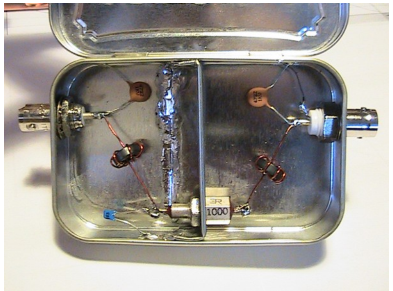
M S T SPECTRUM SCOPE FIX MAIN Grid 25k/10dB

S · 1 · 3 · 5 · 7 · 9 +20 +40 +60dB
Po 0 10 50 100 150 200 250 W
S · 1 · 3 · 5 · 7 · 9 +20 +40 +60dB
Po 0 10 50 100 150 200 250 W
7.000
M
7.300
7.05
7.10
7.15
7.20
7.25
ATT
MARKER
HOLD
CENT/FIX
MAIN/SUB
WIDE/SET

Radio filters



Alessio Damato via Wikimedia Commons



Carol F. Milazzo, KP4MD
via QSL.com

Filter nodes – more is usually better

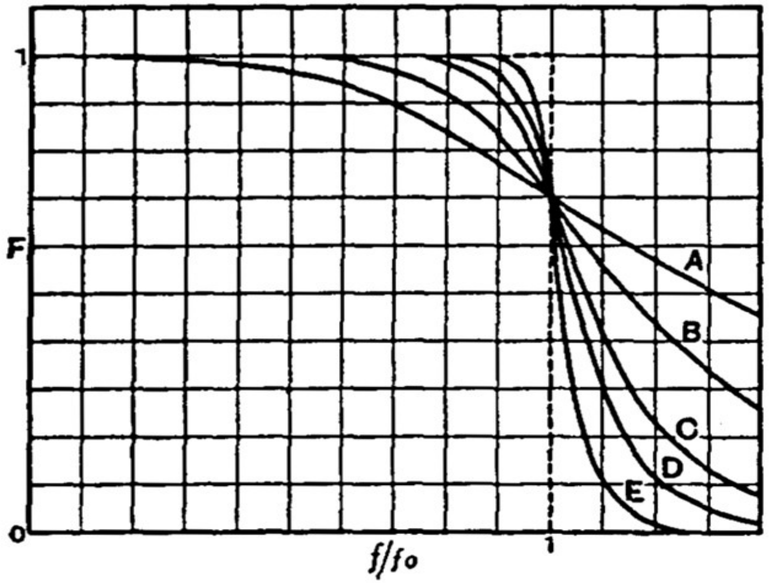
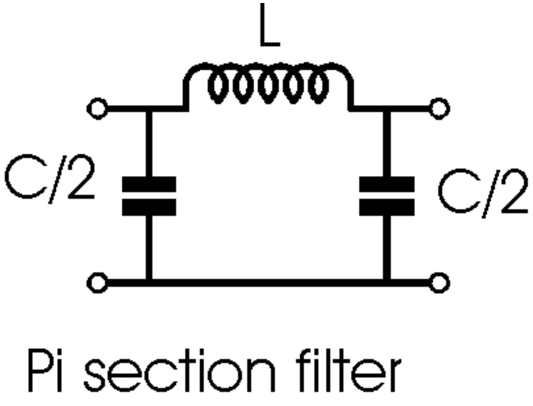


Fig. 3.

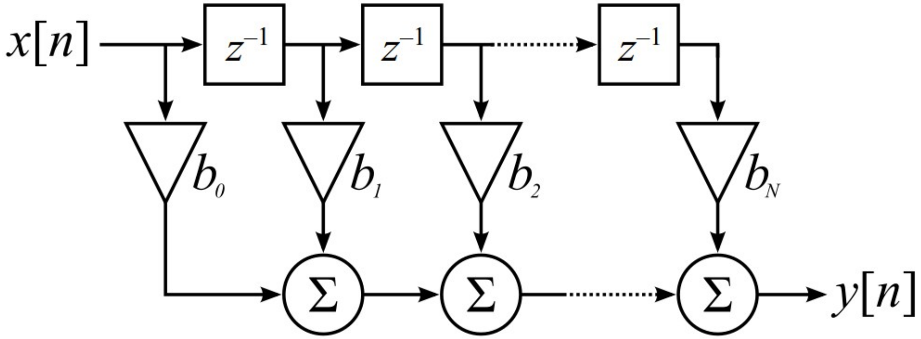
Stephen Butterworth 1885-1958 - Experimental Wireless page 536-541. Via Wikipedia



Radio-Electronics.com

Leymel Industries South Africa

Digital filters



BlanchardJ via Wikimedia Commons

- Lots of nodes feasible, sharper shapes, pass and stop bands
- Filter curves not feasible with analog filters
- More robust, no capacitors and inductors
- Smaller
- No impedance matching
- No signal attenuation

If you must know...

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_N z^{-N}}{1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_M z^{-M}}$$

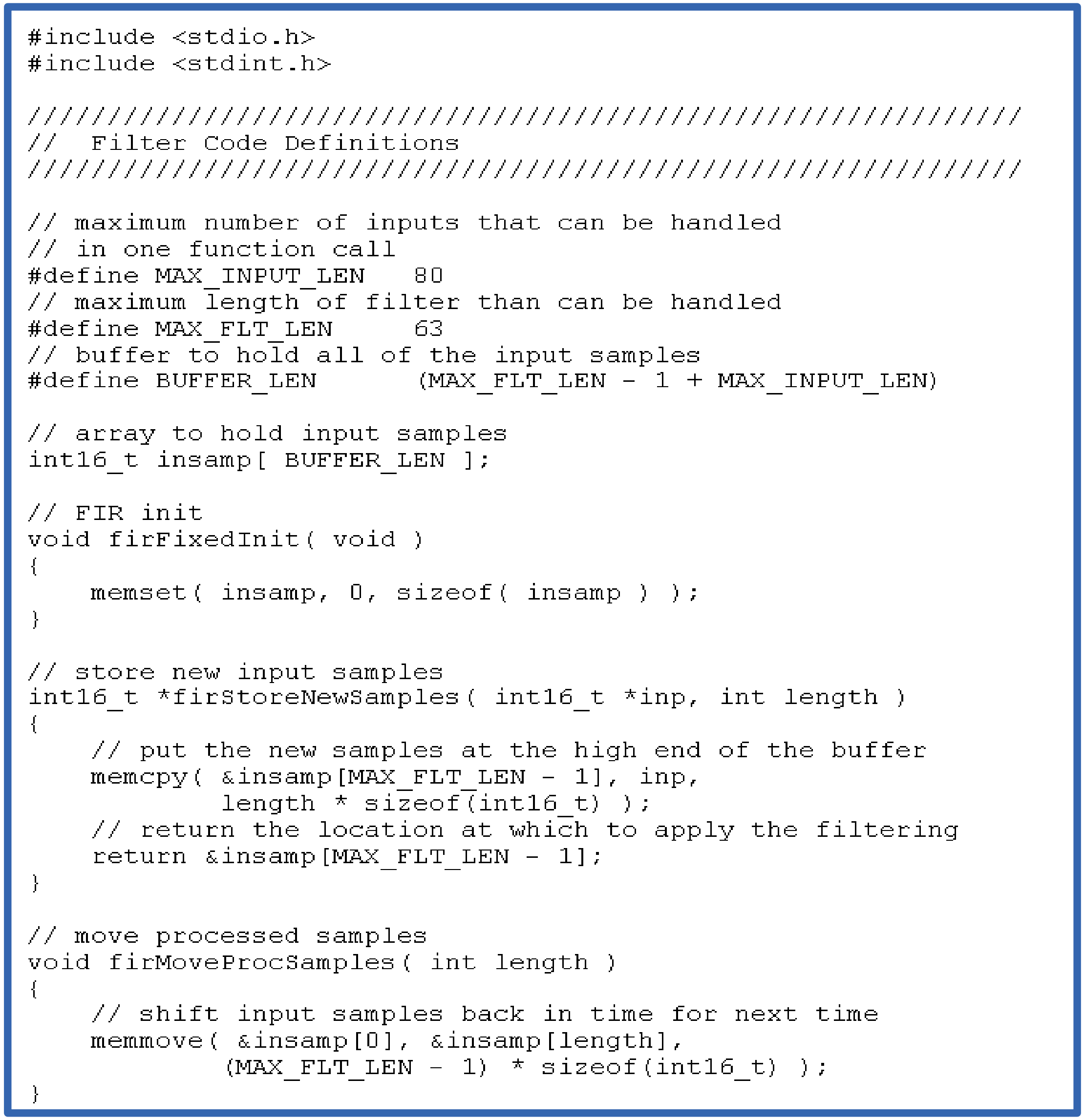
$$y_n = \sum_{k=0}^{n-1} h_k x_{n-k}$$

$$\sum_{m=0}^{M-1} a_m y_{n-m} = \sum_{k=0}^{n-1} b_k x_{n-k}$$

$$H\left(e^{j\omega}\right) = \frac{1}{3} + \frac{1}{3} e^{-j\omega} + \frac{1}{3} e^{-j2\omega}.$$

- Transfer function
- Impulse response
- Infinite Impulse response
- Moving average FIR filter

For those who write in C



```
#include <stdio.h>
#include <stdint.h>

//////////////////////////////////////////////////////////////
//  Filter Code Definitions
//////////////////////////////////////////////////////////////

// maximum number of inputs that can be handled
// in one function call
#define MAX_INPUT_LEN   80
// maximum length of filter than can be handled
#define MAX_FLT_LEN     63
// buffer to hold all of the input samples
#define BUFFER_LEN      (MAX_FLT_LEN - 1 + MAX_INPUT_LEN)

// array to hold input samples
int16_t insamp[ BUFFER_LEN ];

// FIR init
void firFixedInit( void )
{
    memset( insamp, 0, sizeof( insamp ) );
}

// store new input samples
int16_t *firStoreNewSamples( int16_t *inp, int length )
{
    // put the new samples at the high end of the buffer
    memcpy( &insamp[MAX_FLT_LEN - 1], inp,
            length * sizeof(int16_t) );
    // return the location at which to apply the filtering
    return &insamp[MAX_FLT_LEN - 1];
}

// move processed samples
void firMoveProcSamples( int length )
{
    // shift input samples back in time for next time
    memmove( &insamp[0], &insamp[length],
            (MAX_FLT_LEN - 1) * sizeof(int16_t) );
}
```

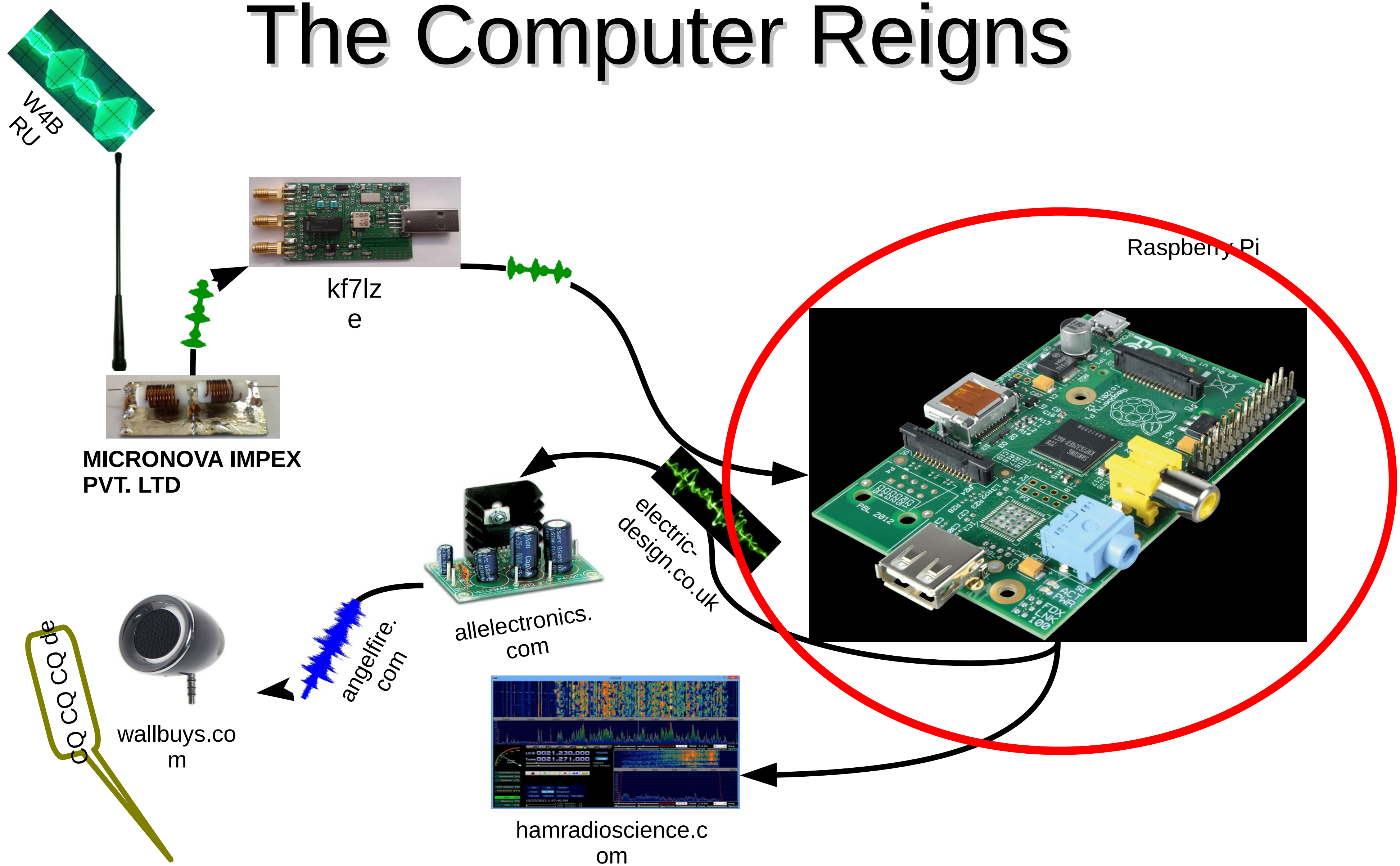
Shawn's DSP Tutorials

https://sestevenson.wordpress.com/implementation-of-fir-filtering-in-c-part-3/

Typical SDR objectives

- Suppress noise
- Emphasize signal
- Detect and display digital signals like PSK, CW, RTTY
- Compress radio size
- Use less expensive components, e g tiny computers in place of capacitors and coils
- Flexible, modulation

The Computer Reigns



W4BRU

kf7lze

MICRONOVA IMPEX PVT. LTD

Raspberry Pi

electric-design.co.uk

allelectronics.com

angelfire.com

wallbuys.com

hamradioscience.com

CQ CQ CQ de