etherSense

SENSORS / ETHERNET INTERFACE

User's Manual

ircam
Centre
Pompidou

ircam
Centre
Pompidou

For further information, please contact :

Paola Palumbo
Marketing department
IRCAM
1, place Igor-Stravinsky
75004 Paris
France

Tel. (33) 01 44 78 14 87
Fax (33) 01 44 78 43 55
E-mail : admin-forum@ircam.fr

IRCAM
1, place Igor-Stravinsky
75004 Paris
France
Tel. (33) 01 44 78 48 43
Fax (33) 01 44 78 15 40

Safety instructions

⚠ Warning
Before using the device, make sure you have read the following instructions carefully, as well as the instructions for use.
Do not open or modify the device or its mains adapter, except when this manual tells you exactly how to do so.
Do not try to repair the interface or the components inside it, except when this manual recommends it. Please contact IRCAM in case of problems.
Do not use the device or store it in the following conditions : - Extreme temperatures, or exposed to direct sunlight. - Damp areas. - Dusty areas. - Areas prone to strong vibrations.
If you replace the main adapter, make sure its output polarity, voltage and current are correct.
Do not insert any objects or pour any liquid into the device.
Protect the device against violent shocks.
Before using the device in a foreign country, make sure the main adapter provided is compatible with the main supply.
If the device will not be used for a long period, disconnect the adapter from the mains.
Never place heavy objects on the device.
Never touch the device or the adapter with wet hands when it is plugged in.
Before moving the device, make sure the mains adapter and/or any external elements are disconnected from the unit.
Before cleaning the device, make sure the mains adapter is electrical unplugged.
If the device is used somewhere where it is likely to be struck by lightning, unplug it.

Notes

ircam
Centre
Pompidou

Table of Contents

About EtherSense

You have just acquired an ***EtherSense*** interface. In order to benefit fully from its possibilities, you are invited to read this user manual carefully.

Before using this device, please start by reading the safety instructions and the comments on the use of the device. To make sure you fully master this device, we recommend you read this manual in its entirety.

It is also suggested you keep this manual at hand. when you use the device.

Main characteristics

Analog inputs

The EtherSense is a Sensor to Ethernet conversion device with 32 analog inputs dispatched on 2 daugther boards (16 channels on each board). These inputs are digitized using a 16 bit analog to digital converter (ADC) at a rate of 1 kHz for 16 inputs (500 Hz if 32 inputs are used).

Sensors

Any kind of analog sensors can easily be used with the EtherSense either by directly connecting them on the Sub-D plug or by using the pigtail Jack to Sub-D cable included in the package. Active sensors can be powered by +5 V power supply available on the plug.

Protocols

The digitized values are exported through Open Sound Control (OSC) protocol. OSC messages flow over UDP / IP, one of the Ethernet protocol layer.

Liquid Crystal Display (LCD) and joystick

The liquid crystal display and the joystick allow the user to change the configuration of the device using menus. Changing a parameter has an immediate effect on the operation of the device. The LCD also allows the user to monitor sensors signals with bargraphs.

Comments on the use of the device

In addition to the *Safety instructions* located at the beginning of this document, the following pages explain maintenance procedures for the device when it is to be moved or used.

Power supply

- Do not plug the device into a domestic mains circuit in which are also plugged one or several high power devices capable of generating electromagnetic interference (such as electric engines or dimmer switches).
- The mains adapter may start to heat after long periods of use. This is perfectly normal and is not a sign of malfunction.
- Before connecting ***EtherSense*** to other devices, switch off the power to *all* devices. This precaution significantly reduces the risk of failure of the device or of those connected to it.

Arrangement and storage

- Using the device next to power amplifiers (or devices with large power transformers) may disrupt its operation. In order to avoid any problems, change the orientation of the device or move it as far away as possible from the source of interference.
- Do not leave the device in direct sunlight, next to sources of heat, in a closed vehicle or in areas prone to extreme temperatures. Excessive heat may warp or discolour the box of the device.

Maintenance

- For regular cleaning of the device, use a soft, fluff-free cloth on the box. Never use water or detergents. Never press on the liquid crystal display.
- Never use hydrocarbons, thinners, alcohol, acids or solvents on the device, or the box will warp or get discoloured.

Repairing the device

- It is important to note that data saved in the device may be lost when the device is sent away to be repaired.
- When the device is being repaired, precautions are taken not to lose any data, but it is impossible to guarantee this.

Additional precautions

- It is recommended you be careful when manipulating buttons or connectors on the device. Brusque manipulation of the device may lead to failure or malfunction.
- Do not touch, hit or apply strong pressures to the liquid crystal display of the device.
- Connecting / disconnecting cables must be done holding the connector itself, and not the cable it is joined to. Never pull or push on the cable: that way, you will avoid breaking cables and creating short-circuits.
- When carrying the device, it is recommended, if at all possible, to place it in its box, along with its instruction manual.

Layout of the user manual

This manual details how the sensors to Ethernet interface works, what the different parameters are for, and how to set them. It is made up of four sections and two appendices.

Section 1: Getting started

This section details the basic operations available on the interface.

Section 2: Interface configuration guide

This section details what each parameter in the different configuration menus does.

Section 3: OSC commands and data

This section details how to communicate with the EtherSense using OSC protocol.

Section 4: Software

This section details the softwares and patches available on the CD included in the package

Appendice A : How to change basic network settings

This appendice details which parameters have to be set up properly and how to change them in Windows and Mac OS.

Appendice B : Network basics

This appendice gives advanced informations concerning the Network protocols used : Ethernet, IP, UDP.

It also provides advices to set up a functionnal installation.

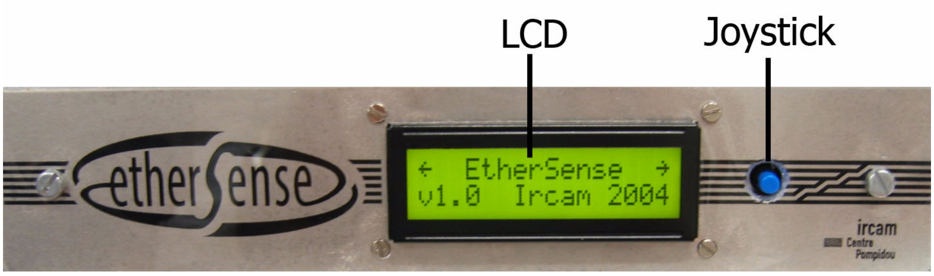
Appendice C : Connect sensors

This appendice gives advices to design a custom sensor installation

Index of terms used

General Overview

Front Panel



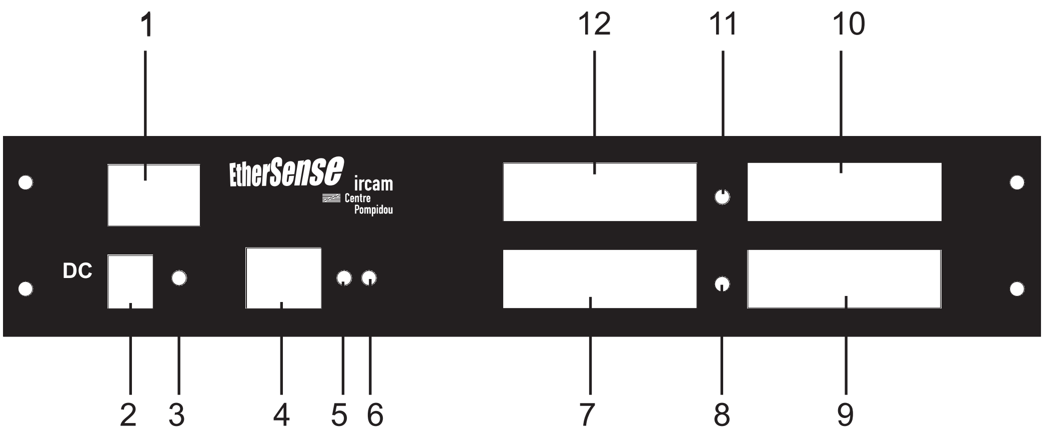
1. LCD

LCD screen that displays menus and parameters to configure the interface: IPs, Port, ID, channel and card status ...

2. Joystick

Used to navigate in the menus : *Left* - *Right* - *Up* - *Down* and *Validation* by pushing the joystick

Rear Panel



1. Power switch

Switch ON and OFF the device

2. Power supply socket

Connector for DC adapter (+9 V, 1100 mA)

3. Power LED

Power and diagnostic red LED, blink continuously in case of daughter board internal connection failure.

4. Ethernet / LAN connector

RJ45 plug to connect to LAN (10/100baseT).

5. Link LED

Yellow LED indicating that link is established.

6. Data LED

Green LED indicating data transmition / reception.

7. Input connector : channels 1- 8 for card 2

Sub-D 15 connector to connect analog channels 1 to 8 on Card number 2.

8. +5 LED for card 2

Sub-D 15 power LED, if ON : the +5V sensors supply is ok, if OFF, the +5V is short circuited.

9. Input connector : channels 9 - 16 for card 2

Sub-D 15 connector to connect analog channels 9 to 16 on Card number 2.

10. Input connector : channels 9-16 for card 1

Sub-D 15 connector to connect analog channels 9 to 16 on Card number 1.

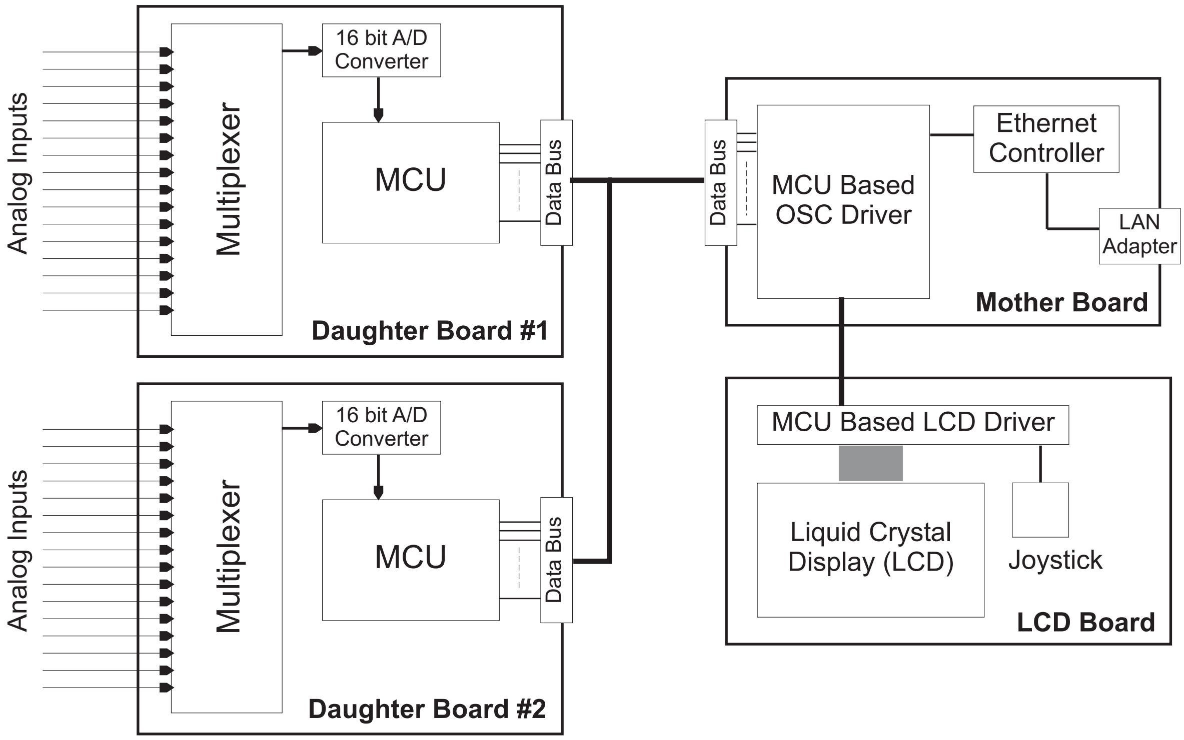
11.+5 LED for card 1

Sub-D 15 power LED, if ON : the +5V sensors supply is ok, if OFF, the +5V is short circuited.

12. Input connector : channels 1 - 8 for card 1

Sub-D 15 connector to connect analog channels 1 to 8 on Card number 1.

Internal architecture of the *EtherSense*



The EtherSense is composed by :

- 1 mother board (Ethernet handling)
- 1 LCD board (configuration)
- 2 daughter boards (digitizing sensors)

The daughter boards multiplexers sequentially connects the different analog inputs to the 16 bit Analog to Digital Converter. It outputs a digital value stored in the Daughter Board microcontroller. This process runs continuously.

The mother board scans the requested daughter board, downloads the digital values, formats them as an OSC (OpenSound Control) compliant message then sends this message to the host computer on the network through the Ethernet controller.

The parameters of the system are accessible through the LCD. A set of menus can be used to configure several dynamic parameters such as :

- Mother board IP address and ID

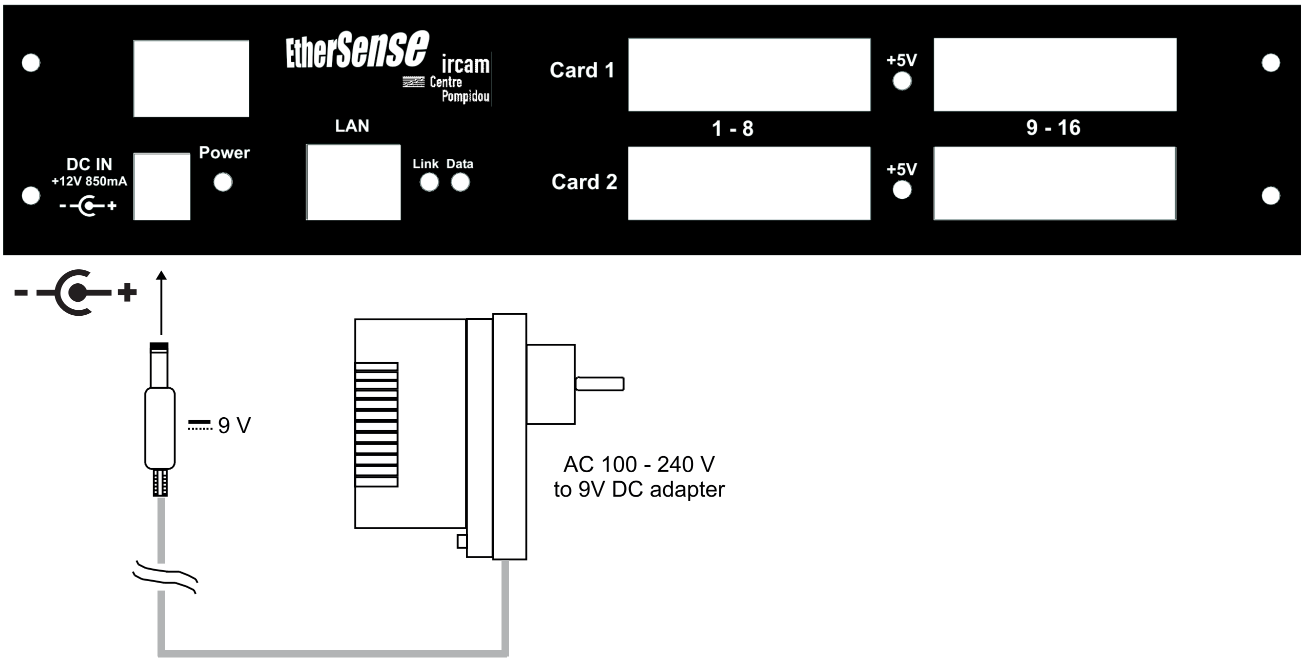
- Host computer IP address and UDP port

It also monitors the analog inputs of the 32 channels with bargraphs on the LCD.

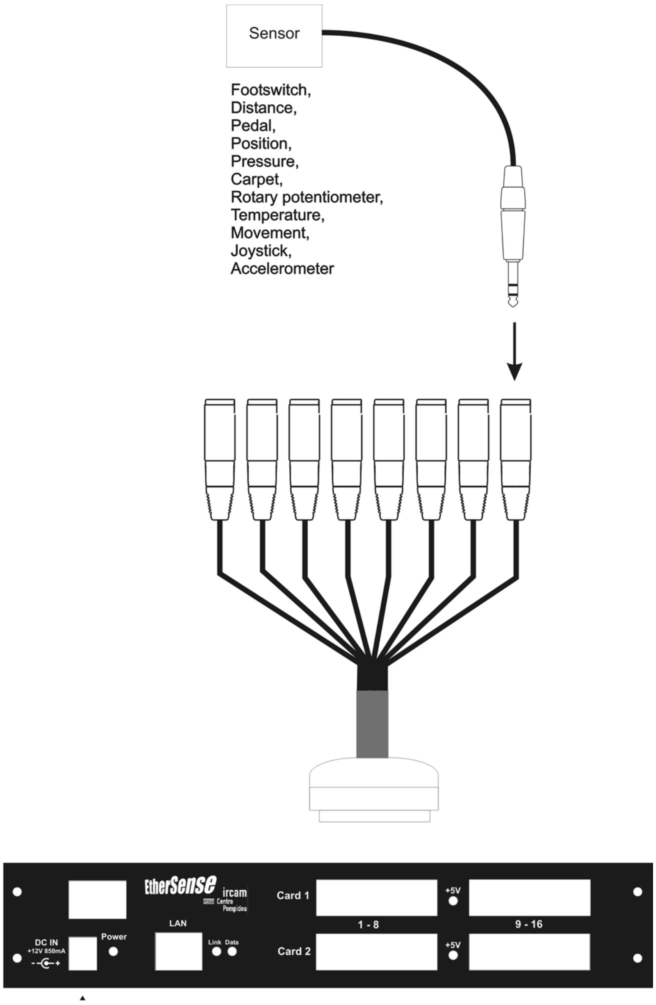
Section 1 - Getting Started

Powering up

Take the ***EtherSense*** and its AC adapter. Connect the output of the AC adapter to the device, then plug the adapter into a domestic AC socket.



Connecting sensors to the EtherSense



One pigtail Jacks to Sub-D cable is delivered with the EtherSense. The Sub-D connectors may be plugged on the female Sub-D connectors of the interface. Each of the 8 female 1/4’’ jacks from a pigtail cable may be connected to a male 1/4’’ jack cable from a sensor.

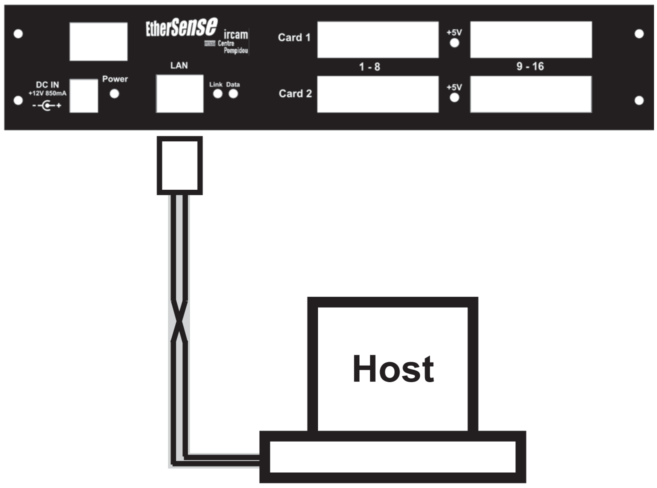
See Section x to have more informations on sensors connection (sensors supply voltage, Sub-D wiring, advices for better accuracy ...)

Connecting to a computer

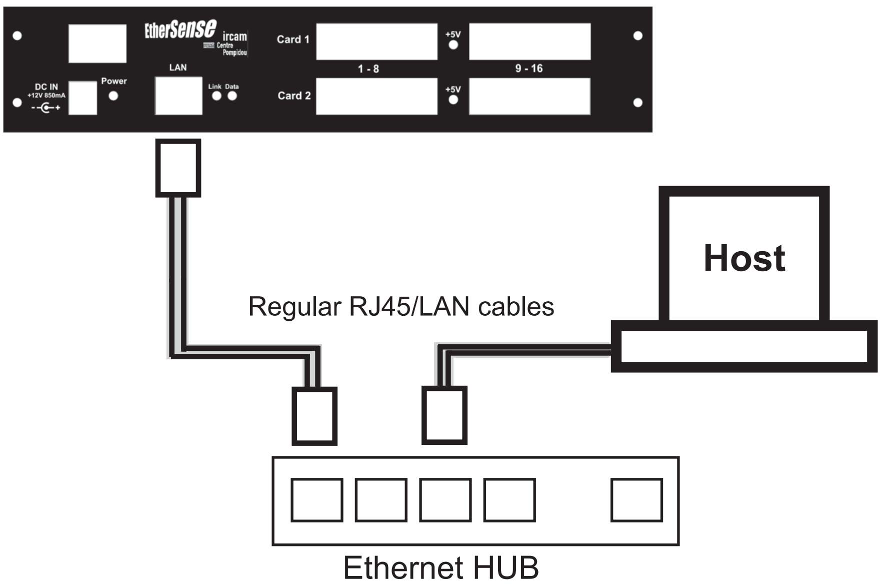
The EtherSense is an Open Sound Control / Ethernet device and need an host computer with an OSC compatible application running, such as Max/MSP, Pure Data ... Therefore, the EtherSense needs to be connected through an Ethernet cable to the host computer.

There are three ways to connect the EtherSense to an host computer :

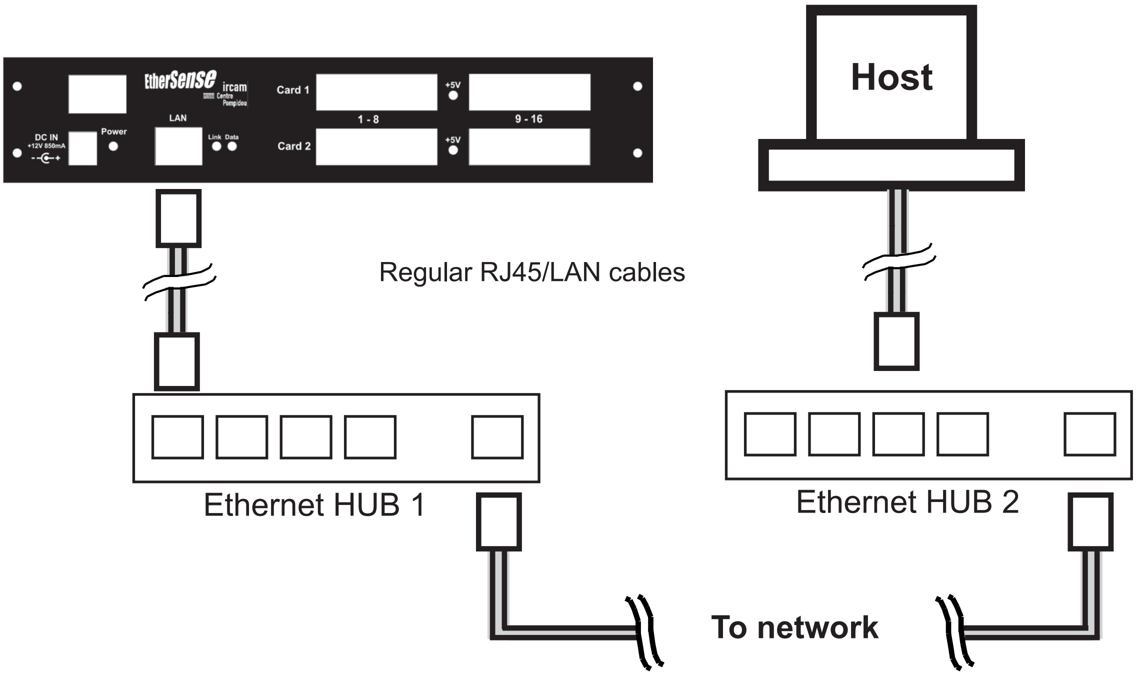
- on a single computer : the EtherSense is connected directly to the host with an Ethernet/RJ45 **crossed** cable (included in the package).



- on a hub : the host computer and the EtherSense are connected to the same Ethernet hub (not included) with regular (non-crossed) Ethernet cables (not included) :



• on a multiple computer or network installation : both the host computer and the EtherSense are connected to the same network with regular cables and Ethernet hubs (or RJ45 Ethernet wall plug ....)



In case of a complex or multiple LAN structure / infrastructure, contact your system administrator to ensure EtherSense can «be in touch» with the desired Host computer.

Once the RJ45 cable is connected, the yellow led a link is established, the green led indicates data transmission / reception

IP configuration

To communicate, directly, through a hub or a network, the EtherSense and the host computer must have IP addresses and UDP ports properly set according to the network installation.

Go to Section 2 to have further informations concerning the configuration of the two devices.

Go to Appendice A to know how to change basic network parameters on your Operating System (Windows 95/98/ME/NT/XP, Mac OS 9 / X)

Go to Appendice B for informations on basic networking.

Welcome screen / Initialization

EtherSense	< EtherSense >	< EtherSense >	< EtherSense >	< EtherSense >
Scan card 1	2 cards found	Connecting ...	Testing UDP port	Connection OK

When the device is powered up, it displays a welcome screen and starts the initialization process :

- it scans the connected daughter board (EtherSense is an open platform and up to 16 cards can be connected to the same mother board)
- it displays the number of connected cards :
- then it tries to connect to the network : a connection is established if the host computer is present on the network and answers an ARP request (see Appendice A for details on networking)
- it then waits for the host UDP port to be opened (for example, a Pure Data patch using the object «DumpOSC 4482» opens the port 4482.)
- once the port is open, the connection is considered as OK, and the EtherSense starts sending values according to its saved configuration :

Scrolling through the menus

The menus displayed on the liquid crystal display (LCD) can be browsed using the left and right direction of the joystick

The display of the different menus along the horizontal axis is cyclic as shown on the above figure :

< EtherSense >	<EtherSense IP>	< Host IP >	< UDP Port >	<EtherSense ID>	< Card 1 >	< Contrast >
Connection OK	10. 0. 0. 2	10. 0. 0. 1	UDP Port = 4482	ID = 1	16 bit - 16 ch	5

Section 2 - Interface configuration guide

This section explains how to modify the configuration of the ***EtherSense***, as well as the role of each configuration parameter. It also details all the LCD menus.

Before modifying the configuration of the device

The configuration is accessible using the joystick :

- «Press *Left*» indicates that you have to press the joystick in the left direction, idem for *Right*, *Up* and *Down*.
- «Press *OK*» indicates that you have to push the joystick on its center without giving any direction.

5 parameters can be modified and saved through 5 menus using the Joystick and the LCD :

- EtherSense IP address
- Host IP address
- UDP port
- EtherSense ID
- Contrast

The EtherSense uses previously saved values on startup.

Another menu displays the status of connected card and sensors.

IP addresses configuration

IP addresses are the most important paramaters to configure. If either the EtherSense or the Host IP address is wrong, communication is impossible.

The host IP address on the EtherSense must be the same as the Host computer IP address; The Host computer IP address must be set on the computer itself and depends on the network installation : single computer with Ethernet crossed cable, Ethernet hub, network with gateways...

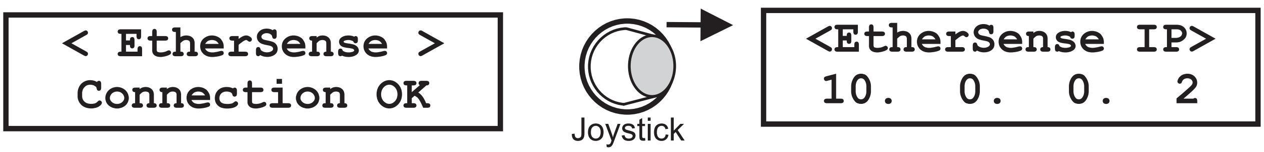
The host IP setup also depends on the Operating System (OS) of the host computer : MAC (OS 9.x, OS X) or PC (Windows NT, XP, 98, 2000 or Linux)

Moreover, other machines IP addresses cannot be re-used on the same network otherwise there will result a conflict. You must know which IP you can use for your computer and for the EtherSense.

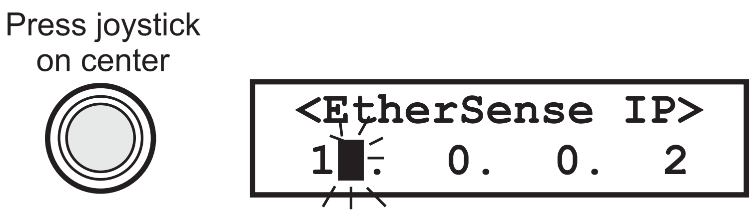
A special section is dedicated to the configuration of the host IP address : see section trucmuche.

Setting the EtherSense IP address

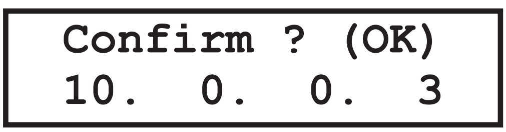
- Go to «EtherSense IP» Menu by pressing the *Right* direction on the joystick



- The menu displays the current IP address of the EtherSense
- If you want to change it, press *OK*
- The first IP field goes to edit mode : the cursor is blinking on the first byte of the IP address :



- Modify the value of the field using the joystick : press *Up* to increment the value, *Down* to decrement.
- Once you have reach the desired value, press *Right* to go to the next field ...
- When the 4 bytes are set, press *Ok* to validate the address, you will get the following menu :

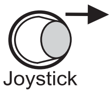


• Press *OK* again to validate the IP or any other key (direction) to cancel and keep the last address.

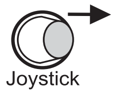
Setting the Host IP address on the EtherSense

• Press *Right* 2 times from the Home Menu or one time from the «EtherSense IP» menu to go to the «Host IP» menu :

```
< EtherSense >
Connection OK
```




```
<EtherSense IP>
10.  0.  0.  2
```




```
<    Host IP   >
10.  0.  0.  1
```

• Repeat the previous procedure («Setting the EtherSense IP address) : press *Ok* to edit, modify each field of the address using the joystick, press *Ok* validate, press *Ok* again to confirm or any other direction to cancel.

UDP port configuration

The EtherSense sends data to an host computer through an Ethernet cable using IP / UDP / OSC protocol. Those protocols are superimposed, see Appendice A. Once the IP addresses are set, the UDP port needs to be configured. The host application filters UDP messages according to the UDP port.

• Go to the «UDP Port» menu by pressing *Right* 3 times on the joystick :

```
<   UDP Port   >
UDP Port = 4482
```

• Then press *OK* to edit the port value, the cursor blinks :

```
<   UDP Port   >
UDP Port = 448█
```

• use *Up* and *Down* to modify the value, the press *OK* to validate :

```
Confirm ? (OK)
UDP Port = 4482
```

• Press *OK* to confirm or any other direction to cancel.

EtherSense ID configuration

Several EtherSense (up to 99) can send and receive data on the same network. Therefore, each EtherSense must have its own ID (a number from 1 to 99). The OSC messages containing the digitized values are sent with the name of the device (including its ID : /Ethersense02/Card01 x x x ...) as the header of the message, see section z for OSC data and commands.

• Go to «EtherSense ID» menu by pressing *Right* 4 times on the joystick :

```
< EtherSense ID >
     ID = 1
```

• Press *OK* to edit the value, the cursor blinks on the field :

```
< EtherSense ID >
     ID = █
```

• Modify the value using *Up* and *Down*, and press *Ok* to validate :

```
 Confirm ? (OK)
     ID = 4
```

• Press *OK* again to confirm new value or any other direction to cancel.

Card and channels vizualisation / monitoring

It is possible to check wether a sensor is connected or not by vizualising the value of the corresponding channel.

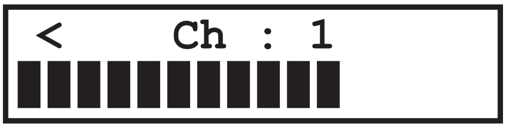
Go to the vizualisation menu by pressing 5 times *Right* from the home menu :

```
<    Card 1    >
 16 bit - 16 ch
```

The card to be displayed can be choosen using *Up* or *Down* key. The LCD displays the configuration of the card : number of bits available and number of channels.

Press *Ok* to select the card to be displayed : the first 16 channels are displayed as bar graphs, each bar represents a channel. If the card has more than 16 channel (custom card), the other channels can be viewed using Up (and then Down) key. This will display channels 17 to 32, and then 33 to 48 ...

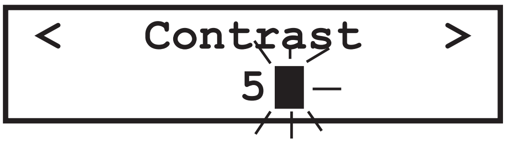
By pressing *Right*, you can acces the single channel display menu. The displayed channel can be changed using *Up* and *Down* key.



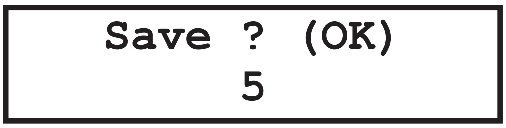
Contrast setting

The contrast of the LCD cand be modified:

- Go to «Contrast menu» by pressing *Left* one time from Home menu :



- The cursor blinks directly and the contrast can be modified dynamicaly from 0 to 10 using *Up* and *Down*.
- You can go to other menus using *Left* or *Right*, or save the current value by pressing *Ok*, (the EtherSense uses the saved value on startup) :



Section 3 - OSC commands and data

This section shows how to communicate with the EtherSense through the OSC protocol. It supposes that you are running an OSC compliant application such as Max/Msp, Pure Data, Reaktor 4, Super Collider ...

(See http://www.cnmat.berkeley.edu/OpenSoundControl/ and www.opensoundcontrol.org for further informations.)

Quick Reference

Commands :

Commands UDP port : 4483 (always open) or user defined port (the one set up in LCD menu 3)

Mother Board

/Who	Connected EtherSense scan (broadcast on port 4483)
/MB/Conf/Request	Ask Mother Board configuration
/MB/Conf/Set/Id 1	Set Mother Board Id to 1
/MB/Conf/Set/Port 4482	Set Mother Board UDP port to 4482
/MB/Conf/Set/HostIP 10 0 0 1	Set Mother Board Host IP to 10.0.0.1
/MB/Reset	Reset Mother Board
/MB/FactoryReset	Reset Mother Board to factory values

Daughter Board

/DB/Period 1 10	Set Daughter Board 1 sampling period to 10 ms
/DB/Run 1	Set Daughter Board 1 to Run mode

/DB/Stop 1	Stops Daughter Board 1
/DB/Boost 1	Set Daughter Board 1 to Boost mode
/DB/Req 1	Request values from Daughter Board 1
/DB/All	Request values from all connected Daughter Boards
/DB/Gains/Request 1	Request Daughter Board 1 gain values
/DB/Gains/Set 1 x x x …	Set Gains for Daughter Board 1
/DB/Gains/Save 1	Save gain values for Daughter Board 1
/DB/Calibrate 1	Calibrate Daughter Board 1 (set all channels to 0)
/DB/Format 1 10	Set Daughter Board 1 to 10 bit resolution
/DB/Average 1 1	Activate average filter on Daughter Board 1 (values : 0, 1, 2)
/DB/Reset 1	Reset Daughter Board 1
/DB/FactoryReset 1	Reset Daughter Board 1 to factory values
/Reset/All	Reset the whole device (= Reboot)

Messages :

Messages UDP port : user defined port (use 4482 for example)

Answer to the command /Who : (broadcast on port 4483)

/Identification/Ethersense01 10 0 0 2 4482 Identification data from device 1:
IP address = 10.0.0.2 - UDP port = 4482

Answers to /MB/Conf/Request :

/MB/Conf/Id 1	Mother Board Id = 1
/MB/Conf/Port 4482	UDP port = 4482
/MB/Conf/HostIP 10 0 0 1	Host IP = 10.0.0.1
/MB/Conf/NBDB 2	Number of connected cards
/MB/Conf/DBList 1 2 7	List of connected cards

/Ethersense01/Card01 12 152 23 62 ...Data from the card 1 of the EtherSense 1

Answer to the command /DB/Gains/Request 1 :

/Ethersense01/Card01/Gains x x x x …Gains from the card 1 of the EtherSense 1

/Msg No card 3 General OSC message (errors ...)

Introduction to OSC syntax

In an host application, OSC messages look like «/MB/Conf/Set/Id 12» (target identifier and parameters values). The structure of the message is transparent for the user but it migth be interesting to know how it is coded.

An OSC message has the following structure : header(identifier), data type, data. Each field is padded on 4 bytes (32 bits) : if a field has only 9 bytes, 3 0x00 will be added

Example : command to change mother board ID :

/MB/Conf/Set/Id 12 is coded like this :

```
'/'  'M'  'B'  '/'
'C'  'o'  'n'  'f'
'/'  'S'  'e'  't'
'/'  'I'  'd'  0x00
','  'i'  0x00 0x00
0x00 0x00 0x00 0x0C
```

The header has 15 bytes, so one 0x00 is added to fit 4 bytes padding.

’,’ indicates data type - one ’i’ for one integer, and two 0x00 added for 4 bytes padding

0x0C for 12 on the Less Significant Bytes (Big Endian)

Note : EtherSense does not implement [Bundles]. If an OSC packet containing bundles is received by an EtherSense, only the first OSC address pattern will be proceeded. Refer to OSC specifications for further informations.

OSC objects suite

To send OSC messages over an Ethernet link, OSC compliant applications need objects to generate the OSC chains as detailed before.

The website http://www.cnmat.berkeley.edu/OpenSoundControl/ lists all OSC compliant applications and proposes OSC objects kits to download for those applications.

Here is a non exhaustive list of OSC compliant applications and the name of the objects to use :

- Max/MSP : (download objects at : http://cnmat.cnmat.berkeley.edu/OpenSoundControl/Max/)
    - PC version : udp-read, udp-write, OpenSoundControl, osc-route
    - Mac version : otudp read, otudp write, OpenSoundControl, osc-route
- Pure Data : sendOSC, dumpOSC, OSCroute
- EyesWeb : StringToOSC, ScalarToOSC, StringFromOSC, ScalarFromOSC
- ...

Please refer to the Help files of the chosen application or to the template patches included in the software package on the CD-ROM.

OSC config from Host computer

Identification

The scan command «/Who» sends a request using a broadcast IP address (255.255.255.255) and the UDP port 4483. This message is received by all the connected devices on the network.

Each connected EtherSense sends back an OSC message using the same address (255.255.255.255) and the same port (4483). This message contains

- the OSC Address Pattern : /Identification indicating that the message is the answer to the scan command
- the EtherSense name : /Ethersense02 for example
- its IP address sent as 4 bytes (from 0 to 255 and not separated by points)
- its UDP port sent as one byte

for example :

/Identification/Ethersense02 196 102 74 39 4482

This function allows a user to know very quickly the parameters needed for the communication with one or several devices.

Caution : this function is network dependant : some network configurations do not allow broadcast messages, if an EtherSense is used on a network with routers / switches, ask the network administrator.

EtherSense IP

You must know the IP address of the EtherSense you want to communicate with. Refer to Section 2 to configure the IP address using the LCD. This cannot be done through Ethernet because you need an IP address to send a command to an EtherSense !

UDP port

In a UDP message, two port numbers are used : source port of the message and destination port. Usually, if there is an answer to the message, the reply uses the previous source port as the destination port. In the case of the EtherSense, the UDP port is defined and all the messages coming from the device will be sent to this port.

The EtherSense filters incoming UDP messages according to their port too. Two ports are opened for incoming messages :

- the UDP port set through LCD configuration
- the port 4483 is always opened

Messages can be sent to the EtherSense on port 4483 or to the user defined port while messages from the EtherSense will be sent only to the user defined port.

Notes :

- the port 4483 (always open for commands) has been chosen because it is not assigned for any existing application (see http://www.iana.org/assignments/port-numbers for further informations)
- choose the port 4482 as user defined port for EtherSense messages. This port is not assigned.

ID

To send messages to an EtherSense, there is no need to specify the ID since each EtherSense has its own IP address. But the ID is part of all the messages coming from an EtherSense : /Ethersense12 will prepend all the messages coming from the EtherSense with the ID 12. This is usefull especially when there are several devices on the same network. This allows the host application to listen to only one UDP port : each EtherSense sends its data to the same host on the same port. Data parsing is then done via the ID number.

Mother Board config

All the commands destined to the Mother Board begin with /MB.

Config request

To retrieve the configuration of the mother board, use the following command syntax without any arguments :

/MB/Conf/Request

You will get 4 OSC answers :

/MB/Conf/Id 1	Give mother board ID
/MB/Conf/Port 4482	Give UDP port
/MB/Conf/HostIP 10 0 0 1	Give saved Host IP
/MB/Conf/DBList 1 2	Give the list of connected cards

Mother Board Parameters change

Mother board ID :

/MB/Conf/Set/Id 2 — Set mother board ID to 2

- the ID is an integer from 1 to 99

Change the UDP port :

/MB/Conf/Set/Port 4482 — Set mother board UDP answer port to 4482

- the port is an integer from 1 to 65535
- do not choose a standard port, use for example 4482

Change the host IP ; data will be sent to this address automatically on start up or on run command:

/MB/Conf/HostIP 10 0 0 1 — Set host IP to 10.0.0.1

- each byte of the IP address is an integer from 0 to 255
- they are passed separately, not linked with points '.'

Daughter board configuration

The internal architecture of the EtherSense allows up to 16 daughter boards to be connected to the mother board. Each daughter is independent and can have its own function. Each card has an hardware Id that can be set using the jumper on the board.

In the case of the box version, 2 digitizing boards are connected and they use IDs 1 and 2. Their main function is to digitize their 16 channels continuously.

Editable daugther board parameters :

- analog gain of each channel
- bit depth
- calibration
- average filter

All the commands destined to the Daughter Board begin with /DB.

Download modes

There are 2 ways to download the data from the boards to the OSC compliant application :

- automatic : continuously with or without sampling period (Run or Boost mode)
- using a request (polling)

Automatic mode :

Each connected card is independent concerning its status. It can be :

- STOP : not sending
- in RUN mode : sending data at the configured sampling period
- in BOOST mode : sending data at the max speed (more than 1000 Hz), without UDP checksum computation

On startup, the last configured modes are recalled and applied.

Commands :

/DB/Run 1	Start card 1 in run mode
/DB/Stop 1	Stop card 1
/DB/Boost 1	Put card 1 in boost mode

- card ID goes from 1 to 16

Change sampling period :

/DB/Period 1 10	Set card 1 period to 10 ms

- Sampling period goes from 1 to 65535 ms
- Each connected card has its own sampling period, for example : Card 1 can send data every 10 ms and Card 2 every 100 ms
- Warning : The global bandwidth of the device is 1000 data packets per second (every one

ms). This data rate is to be shared between the differents cards of the device, meaning that two cards cannot send data at 1 ms each. The best available sampling period for 2 cards in Run mode is 2 ms. Conversly, if a single card is in Run Mode, a 1 ms sampling period can be achieved.

Request mode :

The data can be request by the following messages :

/DB/Req 1 — Request values from card 1

/DB/All — Request values from all cards

- only one answer is sent back

Data type

In the two modes, data are sent using the following syntax, example with card 1 of the EtherSense which Id is 2 :

/Ethersense02/Card01 120 1599 2 65000 12 ..... (with the 16 values corresponding to the 16 channels)

Gains configuration

The gain of each channel can be set to the following values : 1, 2, 4, 5, 8, 10, 16, 32, using the command /DB/Gains/Set followed by the card number and the 16 gain codes.

gain	1	2	4	5	8	10	16	32
code	0	1	2	3	4	5	6	7

example for card no 1 : /DB/Gains/Set 1 0 0 0 0 1 1 1 1 0 0 0 0 4 4 2 1

Current gains can be saved using : /DB/Gains/Save 1, where the daughter board id is an integer parameter from 1 to 16.

Current gains can be retrieved using : /DB/Gains/Request 1, where the daughter board id is an integer parameter from 1 to 16.

They are sent by the EtherSense using the following syntax :

/Ethersense01/Card01/Gains 0 0 1 1 0 0 ... with the 16 values

Advanced user functions :

3 other functions have been implemented for advanced users.

Calibration

/DB/Calibrate 1

The device can be calibrated for a special setup. When calibrating, all the channels values are set to 0. For example, if you have an installation with sensors starting from non zero values, you can set the device in order to have zero for the lowest value of your sensors : leave the sensors in a certain state and the use the command /DB/Calibrate 1 (for card 1 ..), then this state of the sensors will the state 0.

Notes :

- the device is factory calibrated
- if the range of a sensor goes from 10000 to 50000 and if you calibrate the device when the sensor is on its lowest position (10000) then the new range of the sensor will be 0 to 40000.
- this function is usefull if you connect a breakout cable or a breakout box that modify the zero value. Just calibrate the device once the installation is set without sensors or with sensors on the lowest position.
- the command calibrate all the channels of one card.

Data format : bit depth

The bit depth of the card can be set to a value from 1 to 16 :

- 1 bit : values are 0 or 1
- 2 bits : 0, 1, 2, 3
- ...
- 10 bits : 0 to 1023
- 16 bits : 0 to 65535

The number of bits can be modified by the command /DB/Format :

/DB/Format 1 10 sets the card 1 to 10 bits

- this can be usefull if you are using ON-OFF sensors
- if the installation has 3 or 4 bits of noise, you can set the card format to 12 bits for example.

Averaging filter

An averaging filter has been implemented to reduce sensors noise and integrate digitized values.

/DB/Average 1 1 sets the card 1 to Average mode 2.

There are 3 filtering modes :

- 0 : normal mode without averaging

- 1 : simple averaging, the value sent is the mean between the last digitized value and the current digitized value
- 2 : strong averaging, the value sent is the mean between the last sent value (already a mean) and the current digitized value, providing a stronger integration.than in mode 1.

Messages

General messages and error messages are sent using the identifier /Msg

/Msg No card 2	OSC Message (errors ...)

Section 4 - Software

This section details the software package included with the EtherSense.

Introduction

The EtherSense uses the Open Sound Control Protocol to communicate with an host computer. The communication between the device and the computer is made using the OSC commands and syntaxes. Those commands can be directly implemented in higher level applications or custom patches as shown in section 3. A software package is included to get started with the EtherSense. This package contains a graphic configuration and a midi routing application and the patches corresponding to this application for Max/MSP (4.3 or later). A simple PureData is also provided.

First of all, you must know and/or set up the IP of your computer and of the EtherSense : please refer to appendice A to get it or change it.

Installation

OSC Objects

Make sure that OSC objects are installed on your system if you want to use the patches. The applications should work as standalone applications and do not require the installation of the OSC objects.

The object for Max/MSP are located at the following url :

http://cnmat.cnmat.berkeley.edu/OpenSoundControl/Max/

Applications

PC Installation

The applications has no installer program, the files are located on the CD-ROM in the directory :

Drive:\WindowsXP\Appli\

Copy all the files from this folder to a folder on your hard drive such as c:\EtherSense\ and run EtherSense.exe

Mac Installation

Files are located in :

Drive:\MacOS\OS9\Appli\ or Drive:\MacOS\OSX\Appli depending on your OS.

Copy this folder on your hardrive.

Patches

To use the patches simply copy them in a working directory on your hard drive and open them with Max/MSP.

The files are OS dependent and are located in :

Drive:\WindowsXP\Patches\MaxMSP\ or Drive:\WindowsXP\Patches\PureData\

Drive:\MacOS\OS9\Patches\

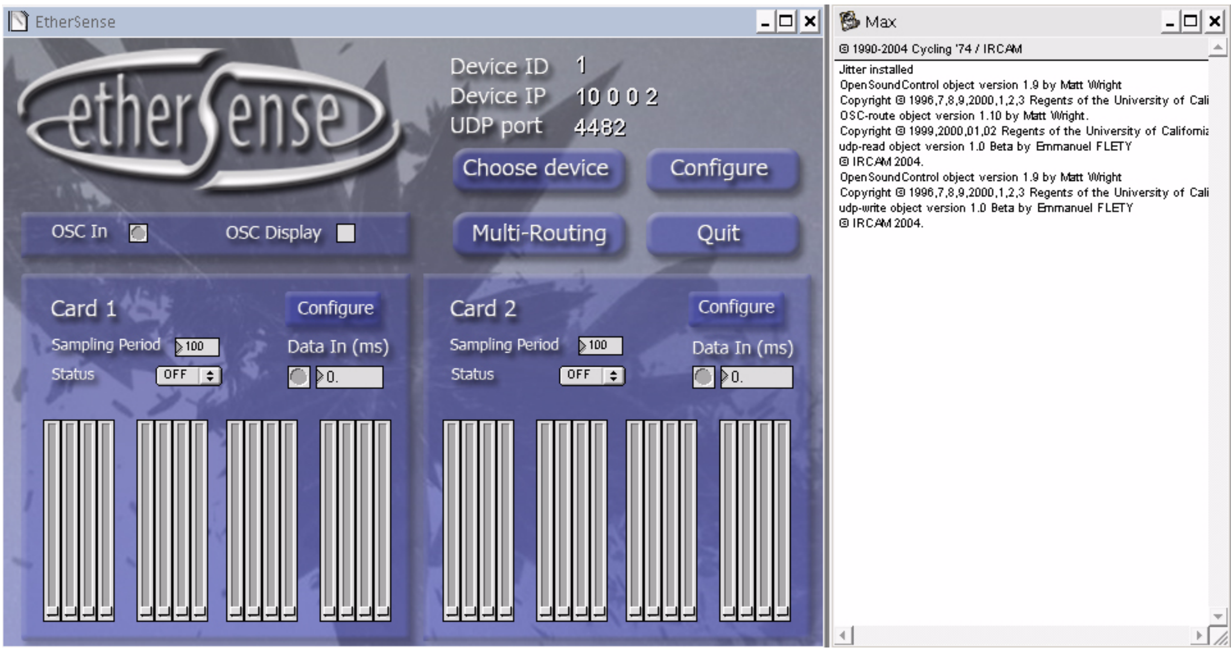
Drive:\MacOS\OSX\Patches\

They are grouped in a set of .txt and .jpg files all necessary.

Configuration and display patch

WindowsXP : Launch EtherSense.exe and the main window appears.

Main Window



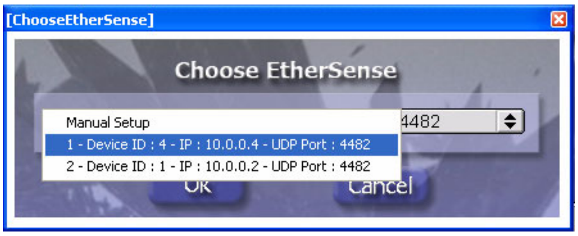
The main window is divided into 2 zones :

- Patch window : includes Device parameters, OSC status, Routing, Card 1 and Card 2 configurations and data,
- Status window : displays messages and OSC data if «OSC Display» is checked.

Choose device

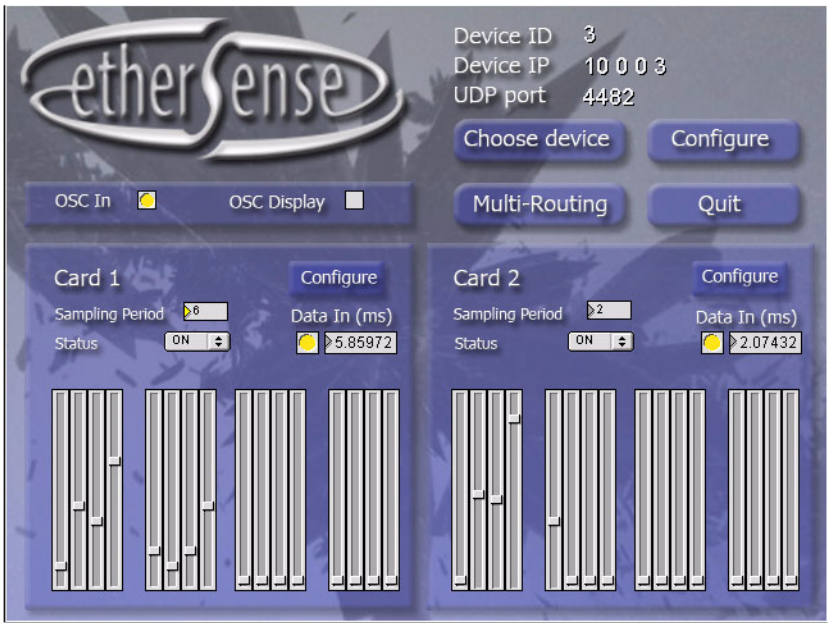
Click on the *Choose Device* button to scan connected devices. You will be asked to choose from devices who answered scan message (be carefull to the network configuration : this is an Ethernet broadcast message using IP address 255.255.255.255 which is disabled by some firewalls , routers and switches) or to set manually the parameters of the target device.

In the next example, 2 devices are connected :

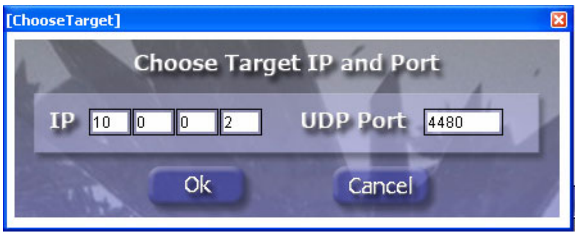


Choose one device or choose «*Manual Setup*» and then click *Ok* (or *Cancel* ...)

• If you have selected a device, the configuration of the selected EtherSense should be received and the panel updated. (if not, check that your IP corresponds to the Host IP displayed on the LCD)



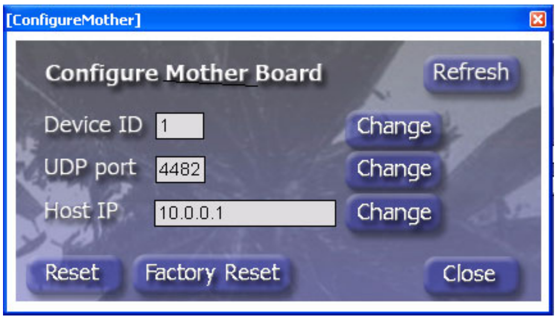
• If you selected «*Manual Setup*», you will be asked for the IP and the Port of the target EtherSense that you want to communicate with :



This manual option is very usefull in the case of complex network installation (Lan with sub nets or routers, Internet, ...) or if a firewall is activated (some firewalls do not allow broadcast messages so the scanning procedure can be done properly).

Configure

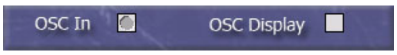
Click on the *Configure* button to configure mother board network parameters of a connected device.



The basic parameters (ID, UDP port and host IP) can be changed. Those parameters can also be changed using the menus on the LCD. The changes are saved on the device.

The mother board can be reset or set back to factory preset.

OSC options



The «OSC In» bang indicates incoming OSC messages. OSC messages (Incoming and Outgoing) can be displayed in the Status window by checking the «OSC Display» check box.

Warning : if one or both cards periods are set below 10 ms, you can expect video and display problems if «OSC Display» is checked. This relies on the computer capabilities.

Cards display and configuration

Card 1 and card 2 are separated. They can have different periods, status (On, Off, Boost), resolution (1 to 16 bits), gains setup, calibration ...

Period and status can be directly set on the panel. Other parameters can be set on the configuration panel (Click on the *Configure* button)

Example :

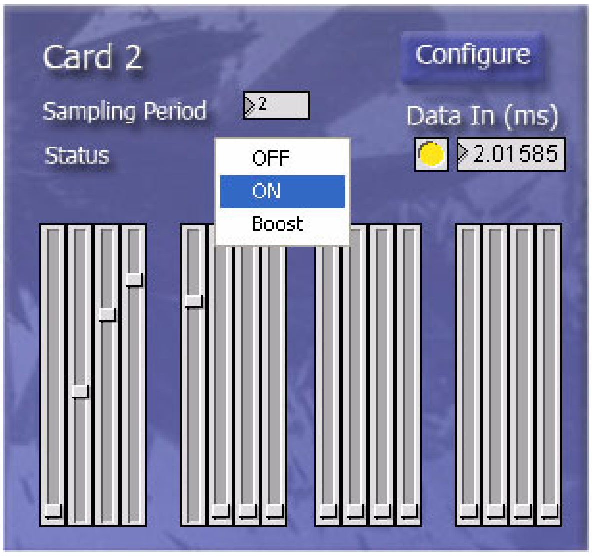
Panel for Card 2

• Period is 2 ms and can be changed by entering the desired period in the «Sampling period» number box. Period minimum is 1 ms and maximum is 65535 ms

• «Data In» indicates the approximative rate of incoming data concerning the card and should be equal to the period.

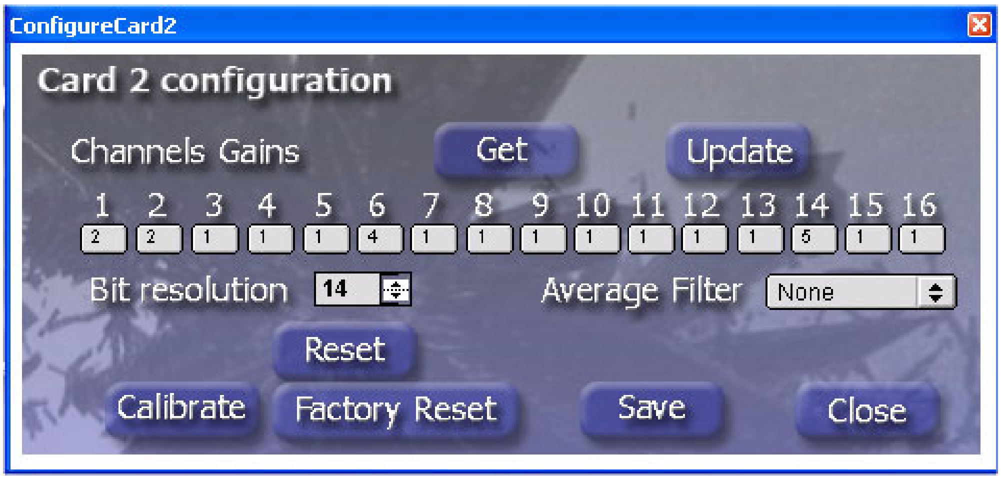
• Status is ON.

OFF : no data come from the card
ON : data come at the chosen period
Boost : data come as fast as possible (around 0.8 ms)



Extended Card parameters

If you click on the *Configure* button, the following window will appear :



Gains :

The gain of each channel can be set to 1, 2, 4, 5, 8, 10, 16 or 32 depending on the range of the corresponding sensor. Click Update to send new values to the EtherSense and save to write them in Eeprom so that the values will always be used.

Bit depth :

The number of significant bits can be modified from 1 to 16 depending on the use of the sensor and on the ambient noise : having 16 stable bits is very hard since the step is 76µV in 16 bit mode and this is under most environment ambient noise (HF, cell phones, electrical pertubations ...). Therefore the bit depth can be modified to have less significant bits and a stable value. The device can be set to 1 bit depth if you have only ON/OFF sensors.

Averaging filter :

An averaging filter has been implemented to reduce sensors noise and integrate digitized values.

There are 3 filtering modes :

- None : normal mode without averaging
- Light : simple averaging, the value sent is the mean between the last digitized value and the current digitized value
- Strong : strong averaging, the value sent is the mean between the last sent value (already a mean) and the current digitized value, providing a stronger integration than in Light mode.

Calibrate

The device can be calibrated for a special setup. When calibrating, all the channels values are set to 0. For example, if you have an installation with sensors starting from non zero values, you can set the device in order to have zero for the lowest value of your sensors : leave the sensors in a certain state and click Calibrate button, then this state of the sensors will be the state 0.

Notes :

- the device is factory calibrated
- if the range of a sensor goes from 10000 to 50000 and if you calibrate the device when the sensor is on its lowest position (10000) then the new range of the sensor will be 0 to 40000.
- this function is usefull if you connect a breakout cable or a breakout box that modify the zero value. Just calibrate the device once the installation is set without sensors or with sensors on the lowest position.

the command calibrate all the channels of one card.

Reset

Reset the card.

Factory Reset

Set the card back to factory presets : gains values are set to 1, bit mode to 16 bits, average filter to None, and calibration to factory calibration.

Save

Writes the current configuration to eeprom. The saved configuration will be used by default on startup.

Multi-Routing patch

This part of the application is accessible through the *Multi-Routing* button. It has been designed to redirect data coming from the EtherSense to :

- Max messages : send

- OSC messages

- MIDI data.

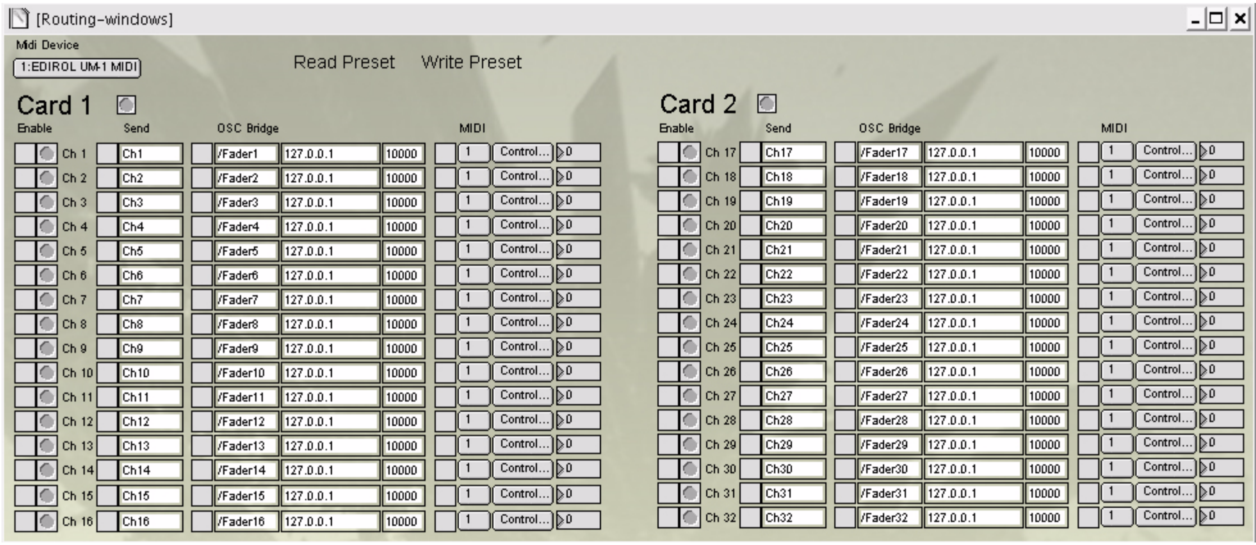
Each channel can be enabled and redirect to one or all type of data. Check the check box corresponding to the channel to enable redirection.

Max messages

Check the «Send» checkbox and enter the name of the send message in the text «Send» field. Use a corresponding receive object in another patch to retrieve data.

Note : in the Standalone mode, this field is not available because two instances of Max cannot run at the same time. As a matter of fact, a standalone application is actually a Max patch run by Max RunTime.

.



OSC messages

Check the «OSC Bridge» Checkbox to enable OSC redirection. Specify the OSC pattern and the destination IP address and UDP port.

Note : use 127.0.0.1 to redirect to localhost (to the same computer).

MIDI

Check the «MIDI» checkbox to enable MIDI redirection.

Choose :

- the MIDI channel : from 1 to 16
- the type of MIDI message : Control Change or Pitch Bend
- the controller number in case of Control Change message : from 1 to 127

Note : a scaling is done to format data coming from the EtherSense to MIDI format :

- for Control Change messages : scaling converts 0-65535 range to 0-127 range (16 bits to 7 bits)
- for Pitch Bend messages : scaling converts 0-65535 range to 0-16343 range (16 bits to 10 bits)

Therefore, be careful to the bit depth chosen for the cards : bit depth must be 16 bits.

Basic Patch

A basic patch for Max is provided. It implements all the OSC commands in a very simple way. Use it as a reminder to design your own patch.

Pure Data Patch

A very simple patch is included in the package. It uses the main OSC commands to communicate with the EtherSense.

Appendice A - How to change basic network settings on the Host computer

This section details the procedure to change basic network settings for Microsoft Windows 95, 98, ME, 2000, XP, NT and Apple Mac OS 9 and X.

If you are not familiar with networking and if you want to learn about important networking topics, please read «Appendice B - Network Basics»

Which parameters ?

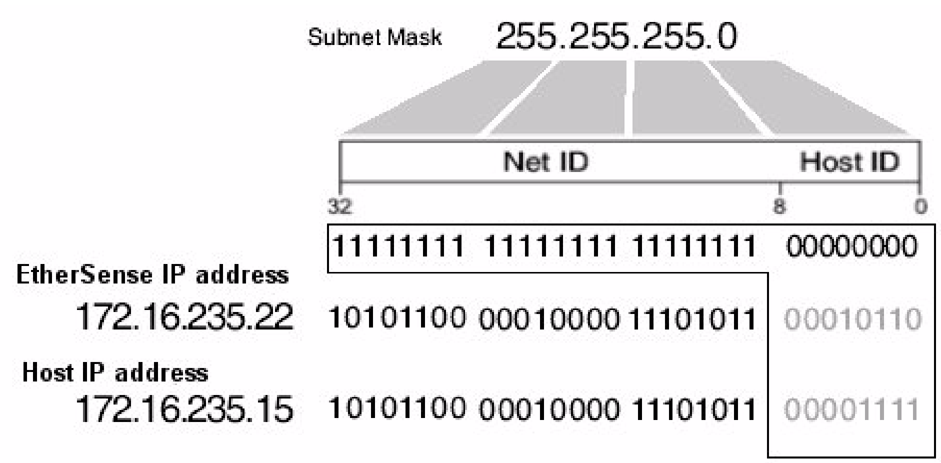
To establish a communication with an EtherSense, it is important to properly setup several network parameters such as the IP addresses and the Subnet mask. In addition, the software firewall - if enabled - has to be configured not to block EtherSense incoming packets.

IP address and Subnet mask

The EtherSense and the Host computer should be in the same sub-network.

For other cases (Internet, different sub-networks, complex network with gateways/routers/switches), please contact your Network administrator.

As explained in the « Appendice B - Network basics» - section «Gateway and Subnet mask», an IP address is formed by 4 bytes and divided in two parts : the NetID and the HostID



Note: IP 176.16.235.15 and 176.16.235.22 and mask 255.255.255.0 are used here as examples.

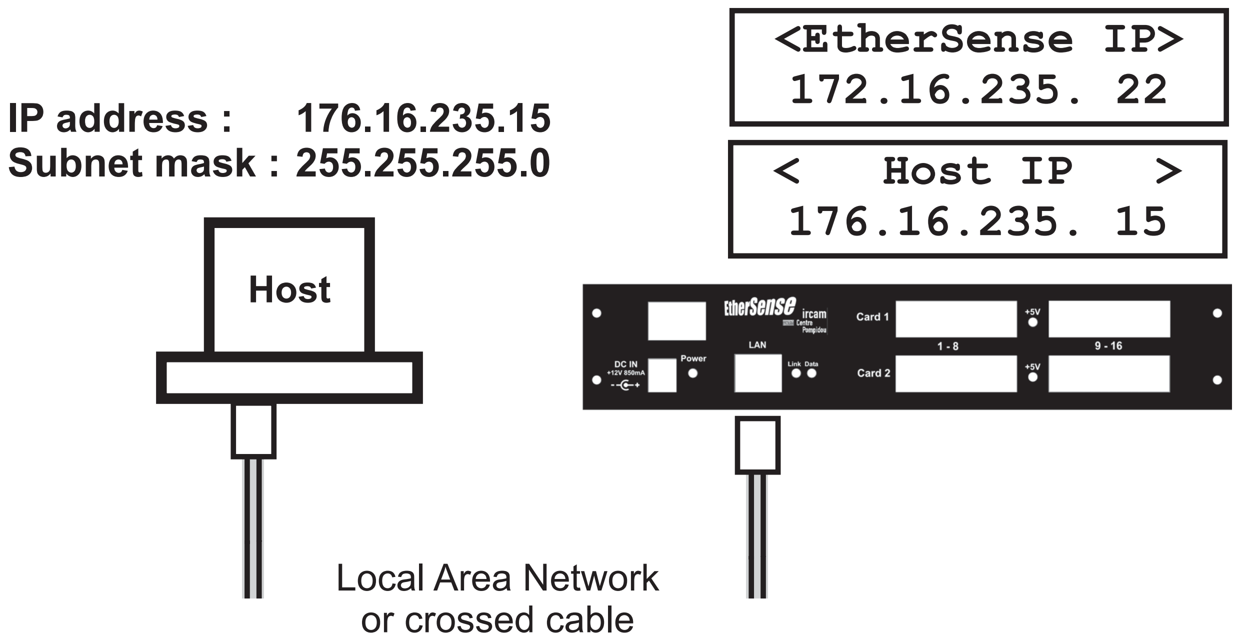
The NetID is defined by the bytes of the subnet mask which value is 255 and the HostID by the bytes which value is 0.

The NetID of the EtherSense IP address and the Host IP address **must** be the same.

The HostID of the EtherSense IP address and the Host IP address **must** be different.

Section 2 - Interface Configuration Guide explains how to set the EtherSense own IP and also the EtherSense Host IP, the address of the Host computer where data will be sent.

The EtherSense Host IP and the Host IP must be the same.



Note: IP 176.16.235.15 and 176.16.235.22 and mask 255.255.255.0 are used here as examples.

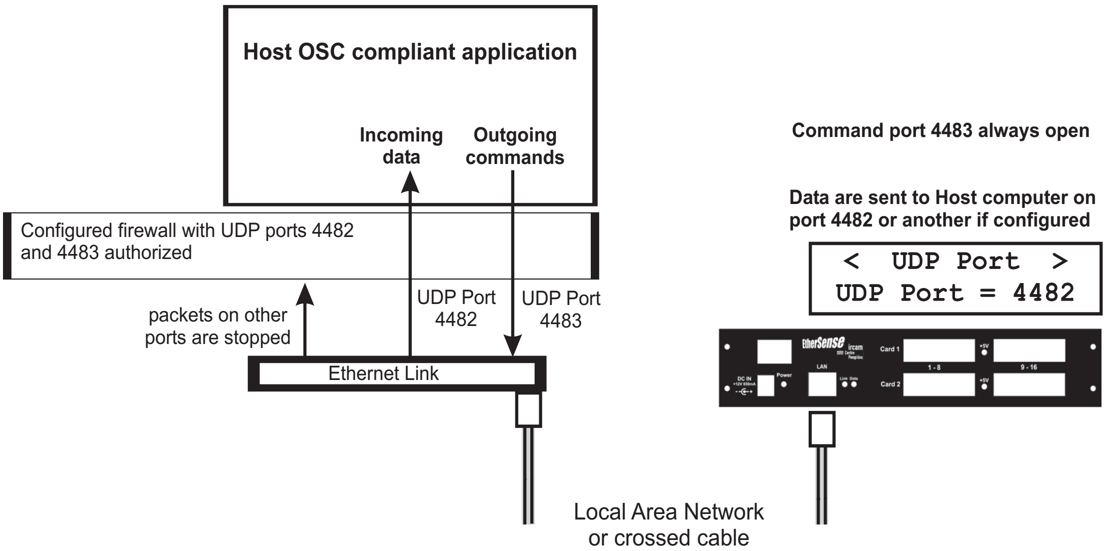
Firewall

A software firewall is a software designed to prevent unauthorized access to a computer from a network or the Internet.

To prevent blocking EtherSense incoming packets, the software firewall (such as Windows XP's or Norton one) should be disabled. If the computer is connected to a network or to the Internet and if the firewall has to be enabled, it should configured not to block packets incoming from the EtherSense.

Therefore, two UDP ports should be opened :

- the port 4483 to send commands to the device
- the port 4482 or another port that has to be configured on the EtherSense to get data from it.



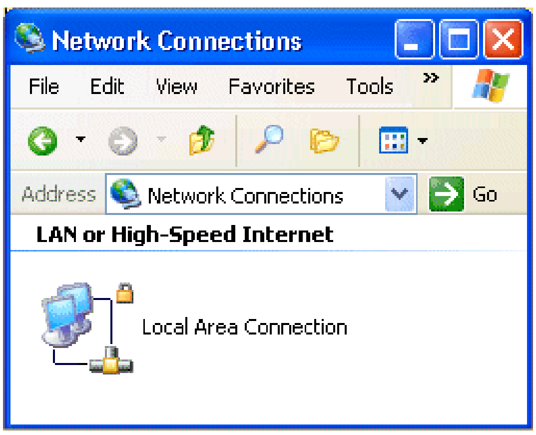
Windows

Windows 2000 / XP

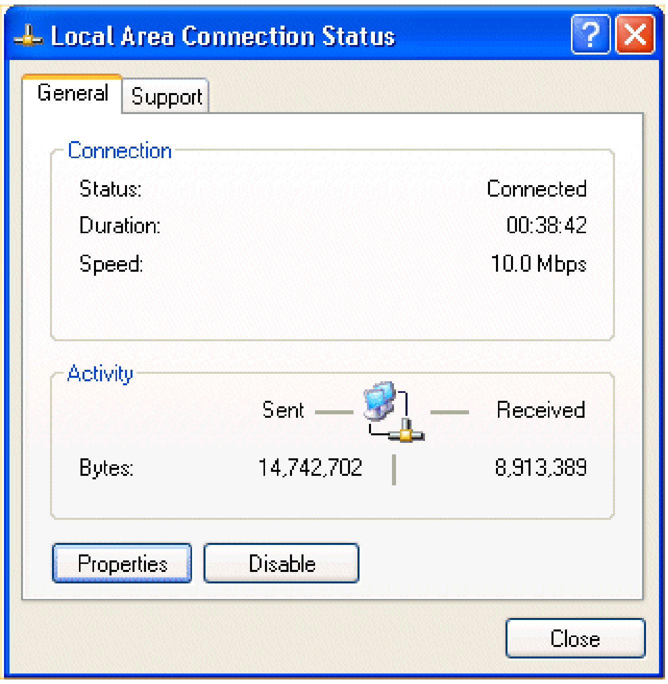
IP address and subnet mask

Open the Windows XP Networking control panel by clicking the Start button

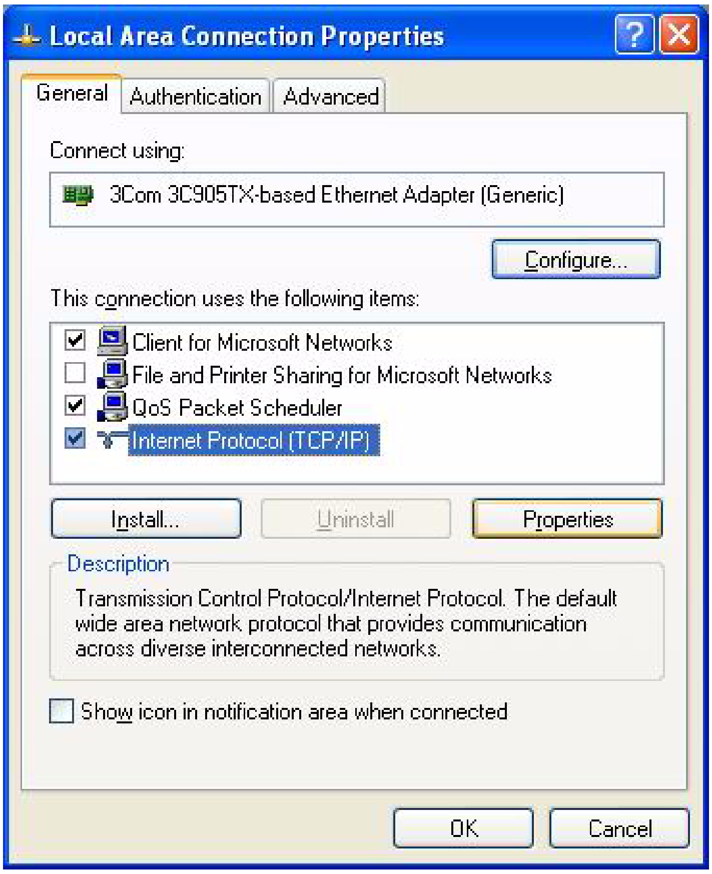
Select ***Settings***, then select ***Network and Dial-up Connections***. Double-click the *Local Area Connection* icon in the resulting Network and Dial-up Connections window as shown :



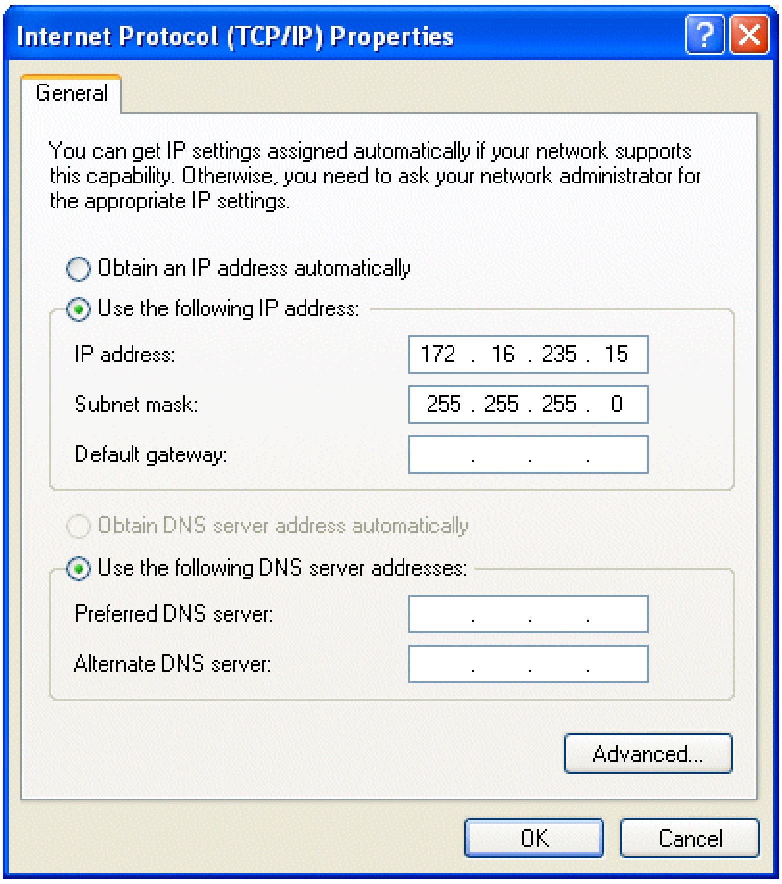
This will bring up the *Local Area Connection Status* window :



Select the ***Properties*** button to get the window below



Select ***Internet Protocol (TCP/IP)*** and then left-click on the ***Properties*** button. The *Internet Protocol (TCP/IP) Properties* window appears :

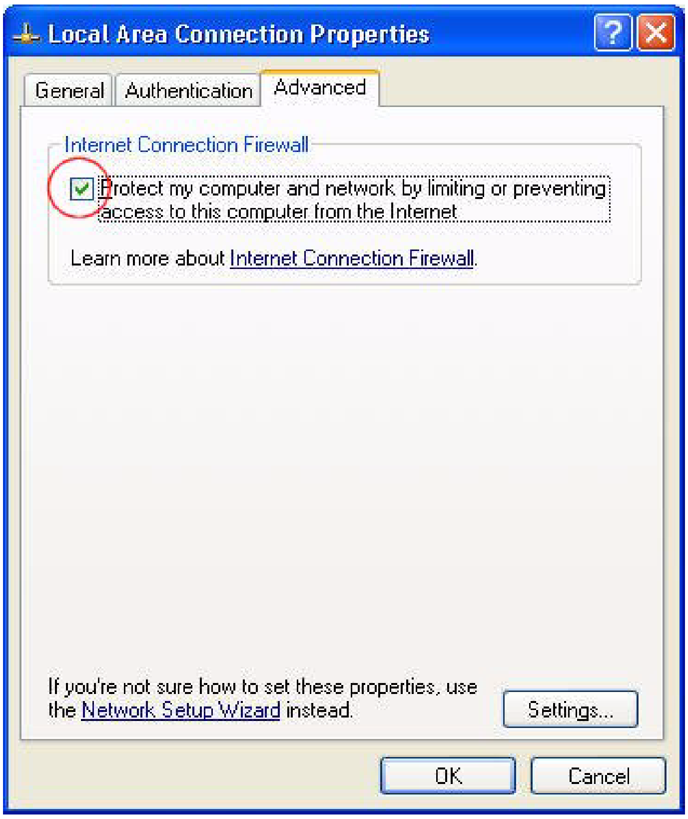


Enter the IP address and the Subnet mask and click Ok

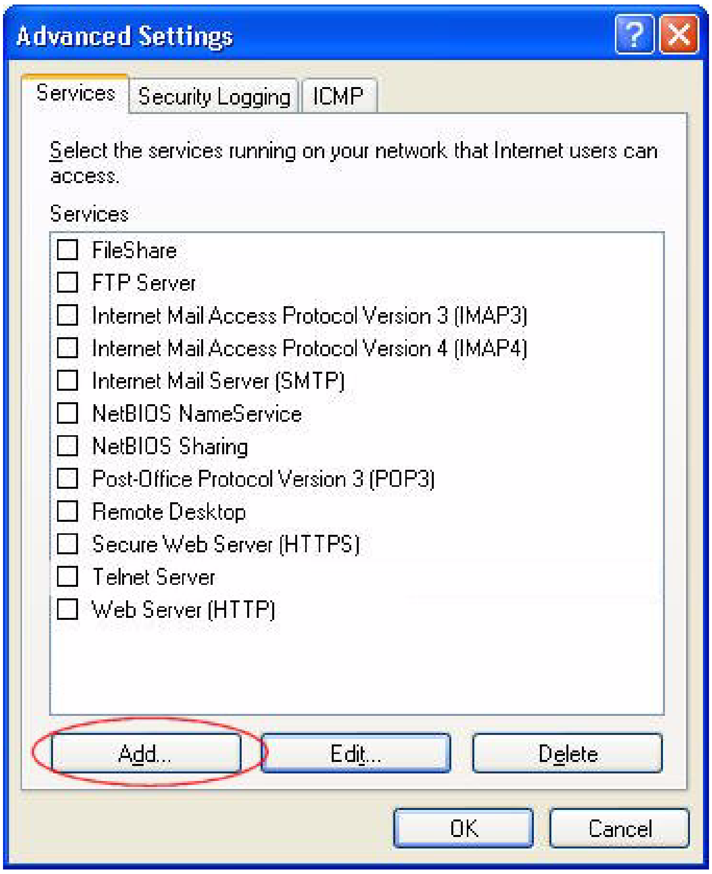
Note: IP 176.16.235.15 and mask 255.255.255.0 are used here as examples.

Windows XP Firewall

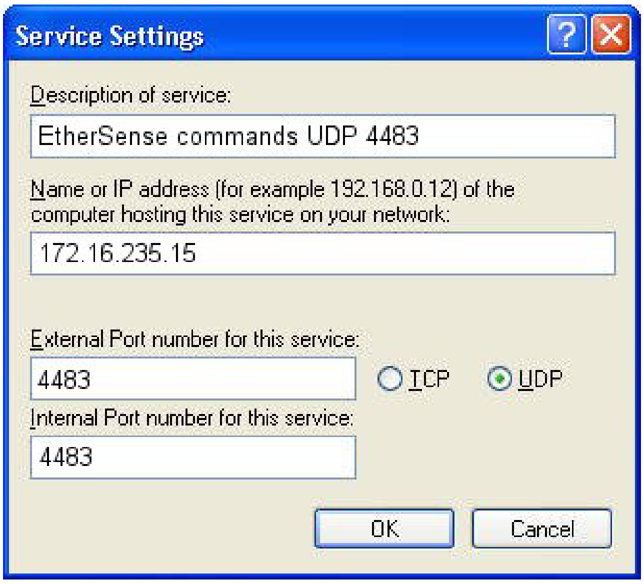
Follow the previous steps to go to the *Local Area Connection Status* window and select the *Advanced* tab.



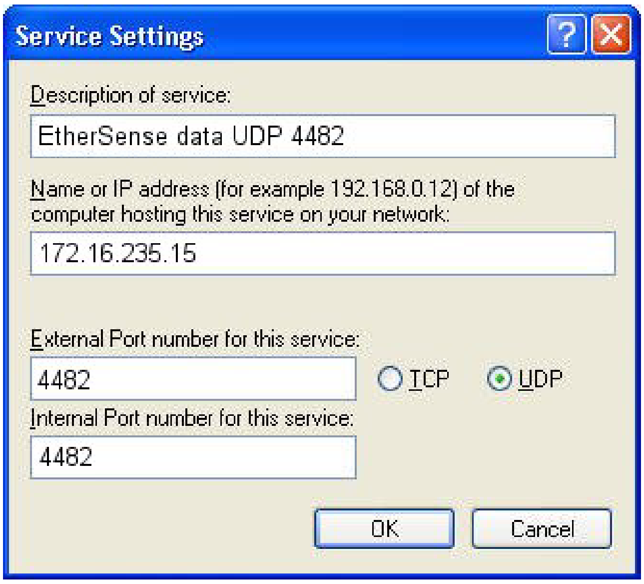
Disable the Firewall by unchecking the highlighted check box. If the firewall has to be enabled, check this box and then click on the ***Settings*** button to configure the firewall.



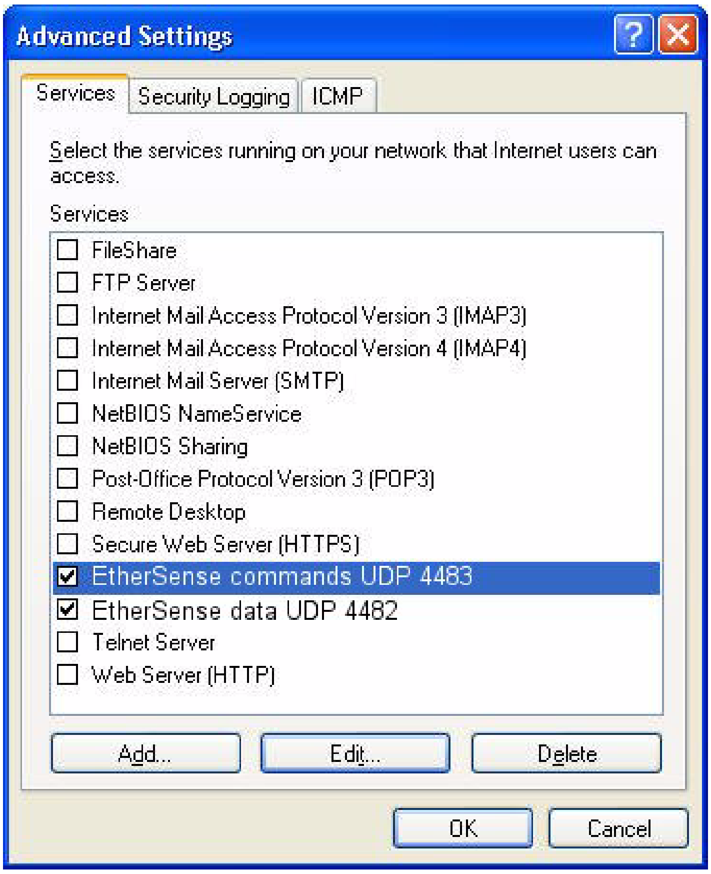
Click the ***Add*** button to add a new service.



Configure the «command» service as shown on the figure, using the IP of the computer and the UDP port 4483. Two ports have to be opened so repeat the procedure for the «data» service using again the computer IP and this time another UDP port, the one that is set on the EtherSense (default is 4482).



Valid with the ***Ok*** button and you will get the next window :

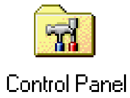


Windows 95/98/ME

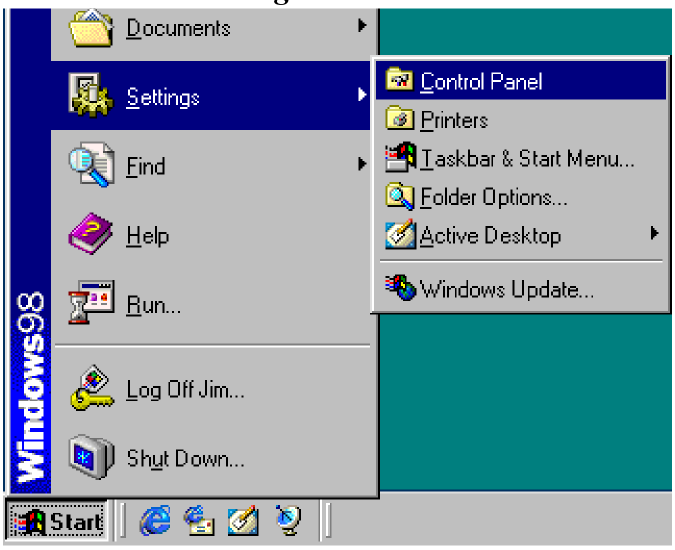
Open Windows 98 Control panel :

Double-click the *My Computer* icon on the desktop

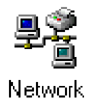
Double-click the *Control Panel* icon



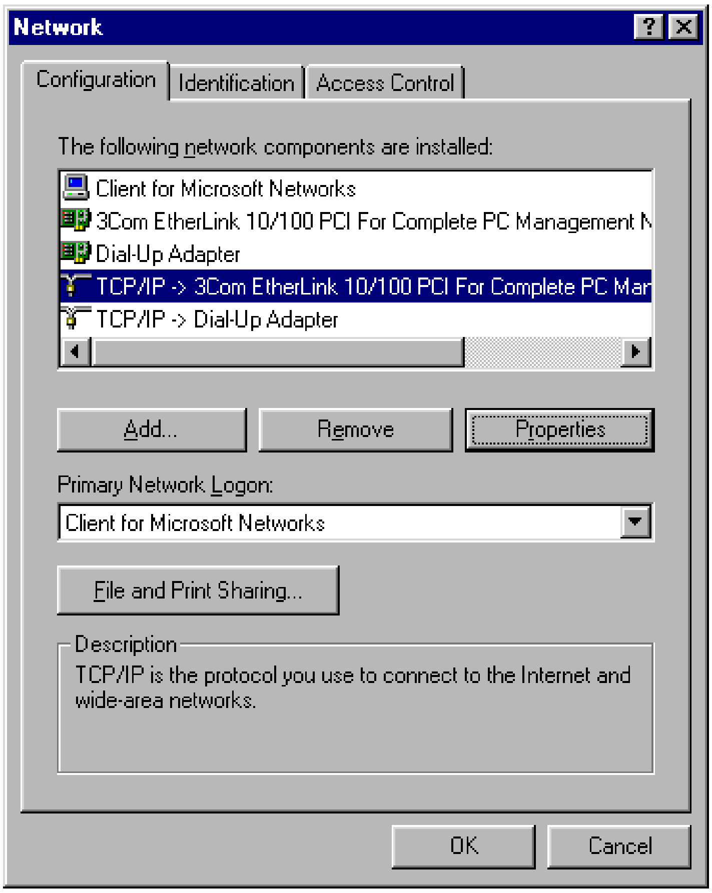
Or click on the ***Start*** button then select ***Settings*** and ***Control Panel*** as shown



Double-click the *Network* icon.



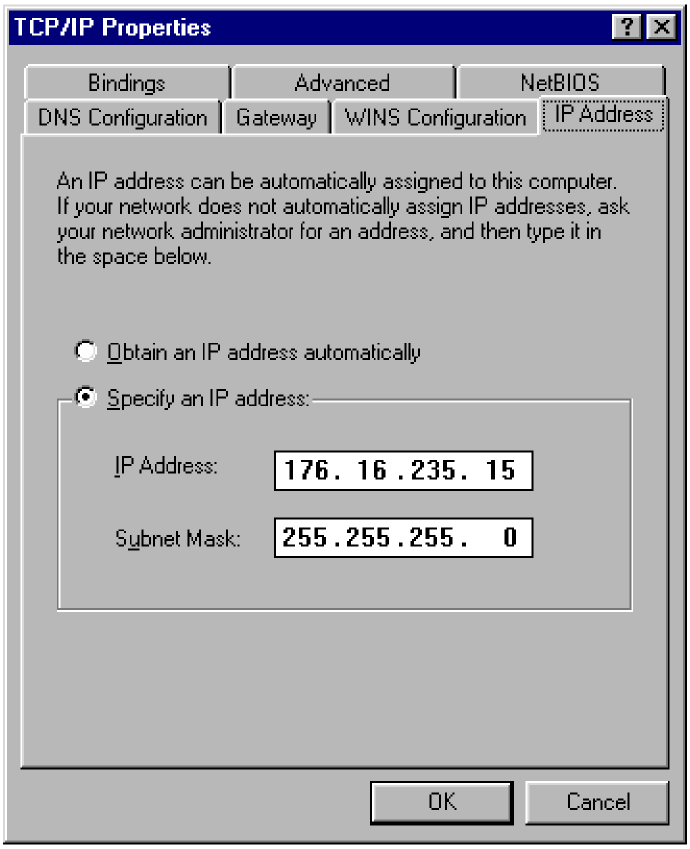
The Network window appears :



Select *TCP/IP -> Your Network Interface Card (NIC)* in the «*The following network components are installed*» list.

Click the ***Properties*** button.

The *TCP/IP Properties* window appears :



Click on the ***IP Address*** tab.

Select ***Specify an IP*** address.

In the *IP Address* field, type the IP address and in the *Subnet Mask* field type the subnet mask .

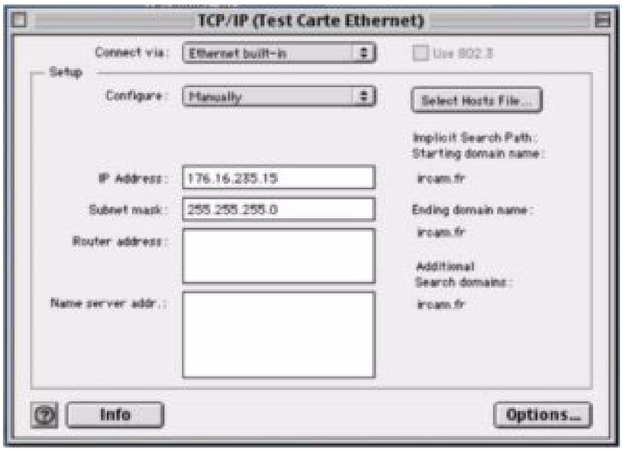
Note: IP 176.16.235.15 and mask 255.255.255.0 are used here as examples.

Macintosh

Mac OS 9

Open the **TCP/IP** control panel :

Then enter the IP address and the Subnet mask in the right fields.



Quit and save.

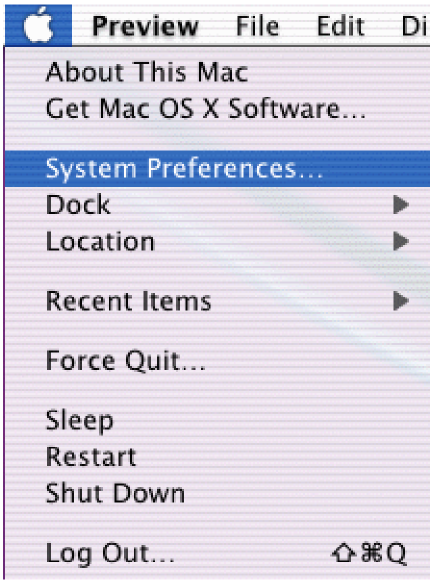
Firewall

Mac OS 9 has no native firewall.

MAC OS X

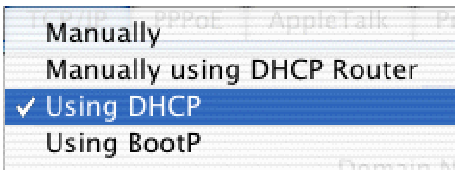
IP address and subnet mask

Go into your Apple, and *System Preferences* :

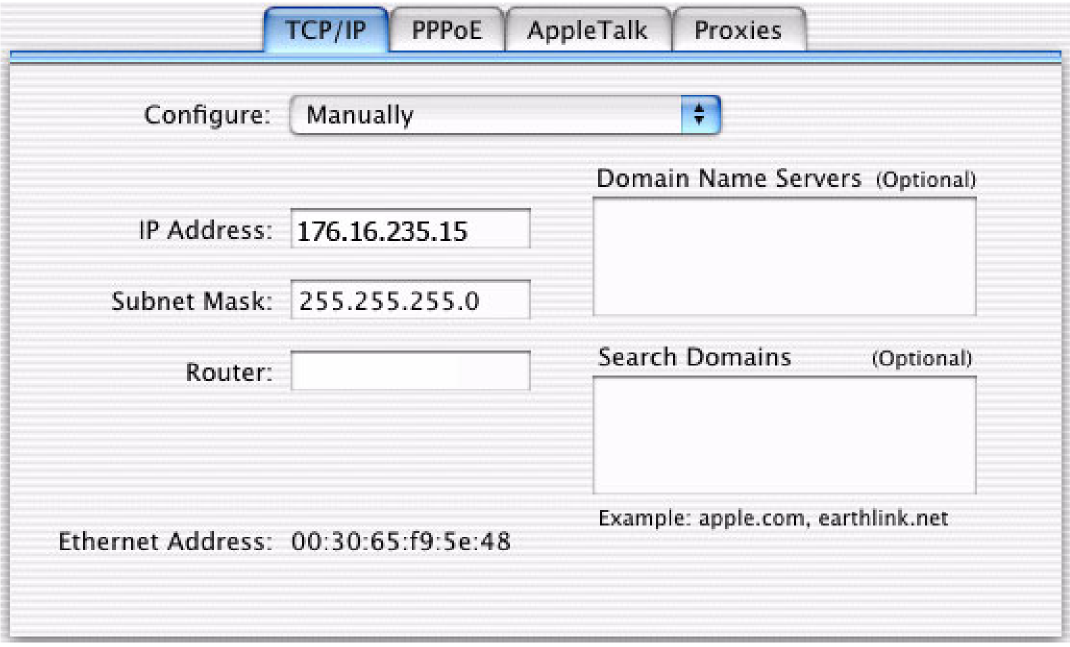


Click on *Network*

In the TCP/IP tab, if you are set for DHCP, you will see your IP Address
Click Configure to change between *Manually* or *Using DHCP*

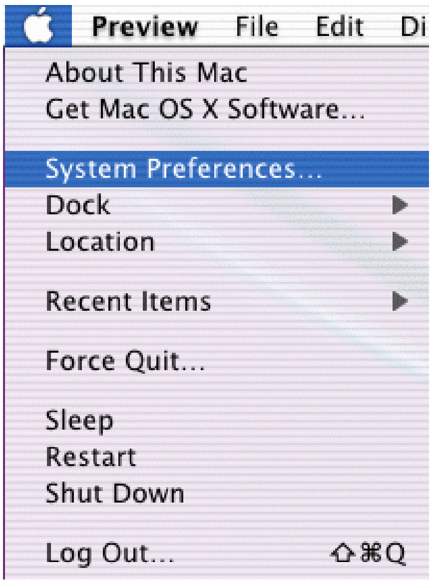


Enter the IP and Subnet Mask manually :



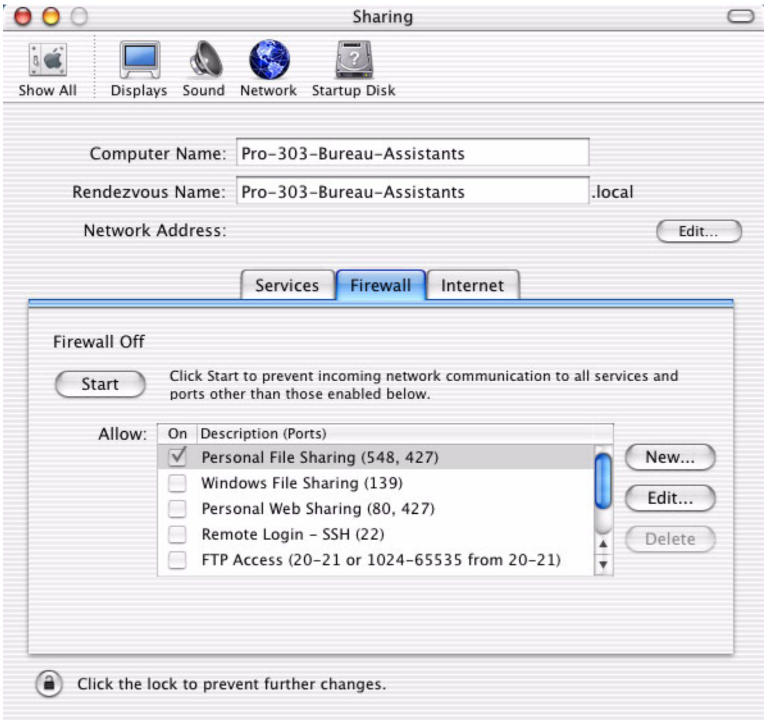
Firewall

Go to *System Preferences* :

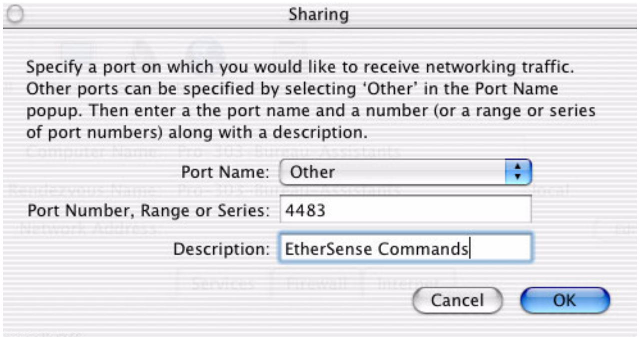


And choose *Sharing*

The following window will appear :



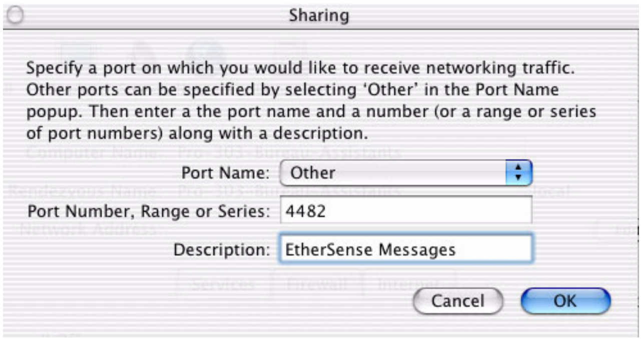
Click on the *Firewall* tab. The status of the firewall will be displayed : On or Off. To use the EtherSense, it should be disabled but if you plan to use the device on a network connected to the Internet, the firewall should be enabled. In this case, the two ports used by the EtherSense must be opened. Click on the *New* button to specify a new allowed port :



Then choose *Other* for the port name (the port used by the EtherSense must not be standard ports).

Enter 4483 in the port number field to allow OSC commands traffic.

Repeat the same procedure to open the port 4482 or any other else (must be set also on the device ...)



Appendice B - Network basics

This section details the protocols used to transfert Open Sound Control messages over an Ethernet connection. It is intended for those who are not familiar with networking.

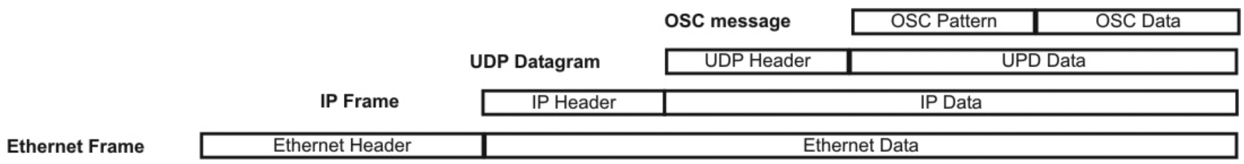
This section is widely inspired from a very well explained document found on the Internet :

Source : «TCP/IP - Ethernet for beginners», W&T, 1999, *http://www.WuT.de*

Note : Reprints, in whole or in part, is permitted if reference to the source, including Internet address (W&T, *http://www.WuT.de*) is indicated.

Introduction

The EtherSense uses the Open Sound Control (OSC) protocol to communicate with an Host computer and send digitized data from its analog channels. For speed reasons, OSC has been implemented on the User Datagram Protocol (UDP). UDP packets are embedded into IP packets. IP packets are transmitted on Ethernet frames :



This section details the things which are really important to understanding of the technologies in use here and explains how to change basic network parameters in Windows and Mac OS 9 & 10.

Basic functions of networks

Every user of a computer has certainly had experience connecting two terminal devices together, such as PC and printer, PC and modem, or PC and PC. The connection is made using a cable specially designed for the application, through which data are sent back and forth between the two devices.

All network topologies have one basic thing in common : every network participant has its own address. The actual data are «packed» into a frame of additional information (e.g. recipient address, sender address and checksum).

The address information in the resulting data packets can be used to get the actual data to the correct recipient over commonly used paths. The example of a letter is not really different : you put the letter in an envelope with the sender and receiver address. The letter carrier then knows where to deliver the letter and the recipient can tell where it came from and reply if needed.

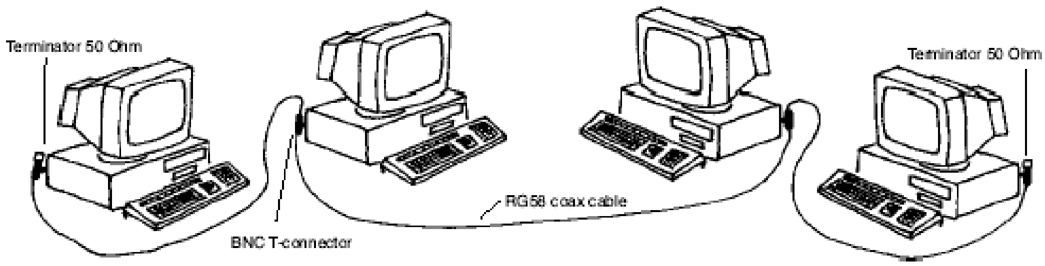
In data transfer within a network, the receiver has the additional option of verifying the contents of the data for completeness using a checksum.

Ethernet

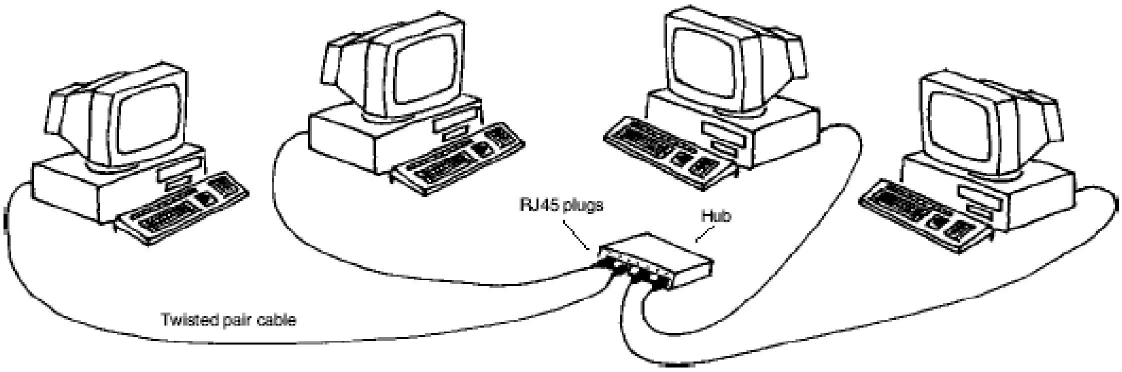
Ethernet is today the most widely used network standard. Ethernet originally ran at a transmission speed of 10Mbit/s.

There are three basic physical models :

10Base2 Also known as Thin Ethernet, Cheapernet, or simply BNC network. All the stations are interconnected through a coax cable (RG58, 50 Ohm wave impedance). The cable must be terminated on both ends with a 50-Ohm terminator.



10BaseT Each participant is connected to a so-called hub (star distributor) which passes all data packets along equally to all stations. 10BaseT is thus star-shaped physically, but works logically like 10Base2 on a bus principle.



10Base 5 (often referred to as «Yellow Cable») was the original Ethernet standard and is hardly used today.

In response to increasingly large data quantities, the 90's saw the development of Fast Ethernet with a transmission speed of 100Mbit/s; here there are two basic physical models:

100Base T4 Just as in 10BaseT each station is connected to a hub through its own twosted-pair cable, with the hub passing all data packets to all stations. 100BaseT4 is hardly ever used any more in new installations.

100BaseTX represents today's usual standard for 100Mbit networks. 100BaseT4 and 100BaseTX differ only on the physical level in the method of data transmission. In addition, 100BaseTX requires higher quality cable.

The Ethernet address – also called MAC-ID or node number – is «burned» into the physical Ethernet adapter (each EtherSense, network card, printer server, router ...) by the manufacturer, so it is fixed for each terminal device and may not be changed. The Ethernet address is a 6-byte value which is generally expressed as a hex number.

example : 00-C0-3D-00-27-8B

The first three hex values represent the manufacturer's code, and the last three are numbered serially by the manufacturer..

Every Ethernet address is supposed to be unique in the world !

There are four different types of Ethernet data packets, which are used depending on the application:

Data packet type	Application
Ethernet 802.2	Novell IPX/SPX
Ethernet 802.3	Novell IPX/SPX
Ethernet SNAP	APPLE TALK Phase II
Ethernet II	APPLE TALK Phase I, TCP/IP

In general, Ethernet data packets of the type Ethernet II are used in connection with UDP-TCP/IP. Here is how an Ethernet II data packet is constructed :

...	**00C03D00278B**	**03A055236544**	**0800**	**user data**	check-sum
Preamble	Destination	Source	Type	Data	FCS

Preamble	The bit sequence with constant alternating between 0 and 1 is used for identifying the start of the packet and for synchronization. The end of the Preamble is indicated by the bit sequence 11.
Destination	Ethernet address of the recipient.
Source	Ethernet address of the sender..
Type	Indicates the higher-order application (example : IP = Internet Protocol = 0800h).
Data	User data.
FCS	Checksum.

The structure of the other Ethernet packets differs only in the *Type* and *Data* fields, to which a different function is assigned according to the packet type. This means an Ethernet data packet possesses all the necessary properties for sending data in local networks from one station to another. Ethernet alone is not however capable of addressing different networks. In addition, Ethernet works connectionless : the sender does not receive any confirmation from the recipient that the packet actually arrived. Higher-order protocols such as UDP-TCP/IP need to be used in any case if an Ethernet network has to be connected with multiple networks.

UDP/TCP/IP – the most important protocols

As far back as the 1960's the American military gave out the assignment of creating a protocol which would enable a standardized exchange of information between any number of various networks regardless of the hard and software used. The result of specification was TCP/IP protocol, which was introduced in 1974.

Although TCP and IP are always named together, they are really two complementary protocols. The Internet protocol IP takes over the actual addressing and delivery of the data packets, while the overlying Transport Control Protocol TCP is responsible for transporting the data and making it secure. UDP is another transport protocol, which like TCP lives above IP. But in contrast to TCP, UDP is connectionless. Each data packet is treated like a separate mailing, and there is no confirmation as to whether a packet was received.

IP – Internet Protocol

Internet Protocol makes it possible to assemble an indefinite number of individual networks into an overall network. This means it enables data exchange between any two network stations located respectively in any given individual network. The physical implementation of the networks and transmission paths (Ethernet, token ring, ISDN ...) is immaterial here. The data are sent to the recipient regardless of these differences.

IP addresses

Under IP every network station has a unique Internet address, often referred to as the «IP Address». This Internet address is a 32-bit value that for better readability is always expressed in the form of four decimal numbers (8-bit values) separated by decimal points (dot notation).

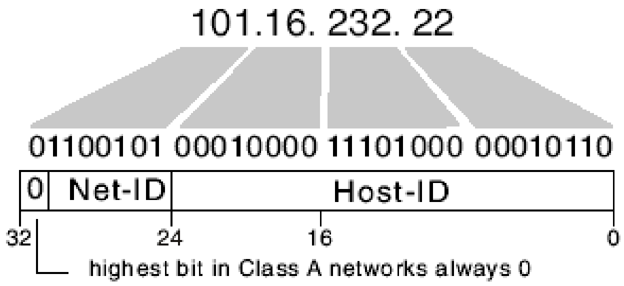
The Internet address is divided into Net ID and Host ID, whereby the Net ID is used for addressing the network and the Host ID for addressing the network station within a network.

Telephone numbers are constructed similarly. There also a distinction is made between the area code and the subscriber's number.

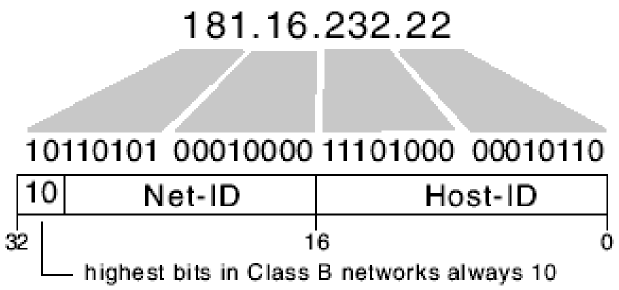
Which part of the IP address belongs to the Net ID and which to the Host ID depends on the size of the network.

Addressing normal networks involves one of three network classes:

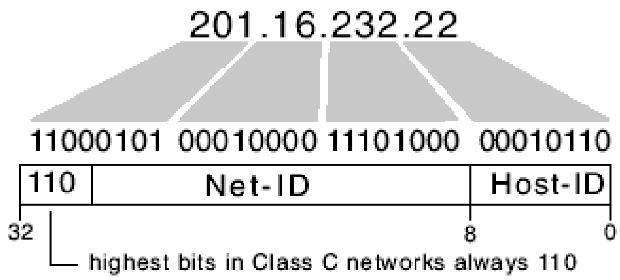
Class A: The first byte of the IP address is used for addressing the network, and the last three bytes address the network station.



Class B: The first two bytes of the IP address are used for addressing the network, and the last two bytes address the network station.



Class C: The first three bytes of the IP address are used for addressing the network, and the last byte for addressing the network station.



The following table lists the basic information for the different network classes:

	possible values of network addresses	possible number of networks		possible number of hosts/network	
Class A	1.xxx.xxx.xxx–126.xxx.xxx.xxx	125	(2^7)	approx. 16 000000	(2^{21})
Class B	128.0.xxx.xxx–191.255.xxx.xxx	approx. 16000	(2^{14})	approx. 65000	(2^{16})
Class C	192.0.0.xxx–223.255.255.xxx	approx. 2 000000	(2^{21})	254	(2^8)

In addition to those listed above, there are also Class D and Class E networks whose address ranges lie above the Class C networks. Class D and Class E networks have little significance in practice, since they are used only for research purposes and special tasks. The normal Internet user will never come into contact with these classes.

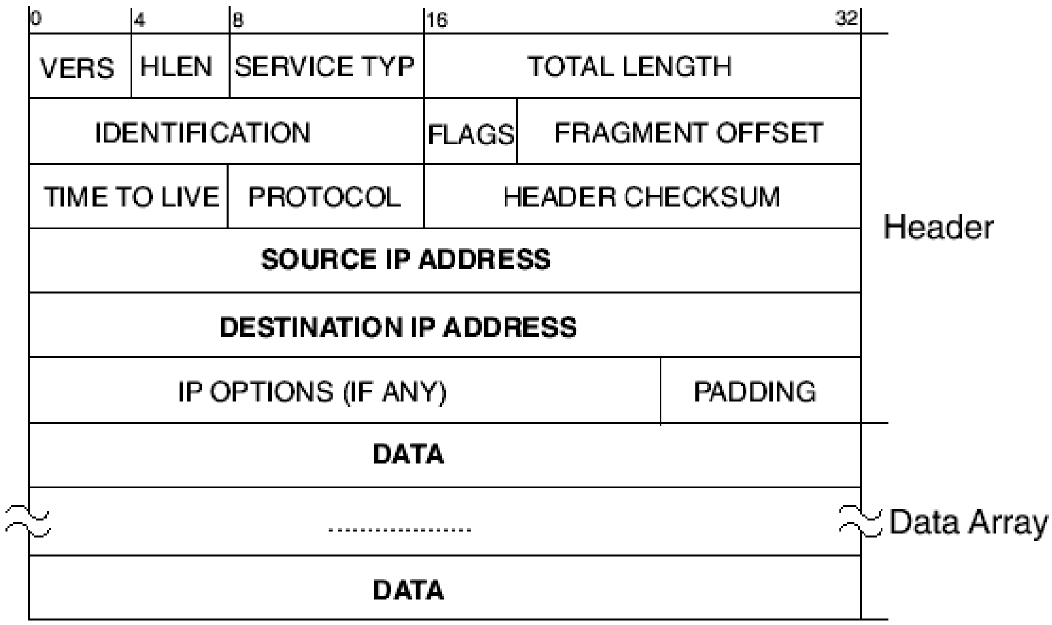
For networks which are to be directly linked with the Internet, a commission called InterNIC assigns an available Net ID and decides based on the intended network size which network class applies. The network operator (administrator) is free to select the assignment of the Host ID to the network station and the resulting IP address. He must however keep in mind that an IP address can be assigned only once at a time.

Caution: ***An IP address must be unique within the entire interconnected network!***

IP Data Packets

The user data are also packed into a frame of addressing information when data are sent over the Internet. IP data packets contain in addition to the user data a variety of address and additional information located in the so-called «header».

Structure of an IP packet



We will restrict ourselves here to explaining the most important address information:

source IP address : IP address of the sender

destination IP address : IP address of the recipient

TCP – Transport Control Protocol

However this protocol is not used with the EtherSense, it might be very interseting to know it as far as it is widely spread.

Because IP is an unsecured, connectionless protocol, it generally works together with the overlayed TCP, which takes over security and handling of the user data.

TCP establishes a connection between two network stations for the duration of the data transmission. When establishing the connection, conditions such as the size of the data packets are specified, which then apply to the entire connection session.

TCP can be compared with a telephone connection. Participant A dials Participant B; Participant B accepts the connection by picking up the handset, and this connection remains until ended by one of the participants.

TCP works on the so-called Client-Server principle:

Whichever network participant establishes the connection (takes the initiative) is called the client. The client makes use of a service offered by the sever, whereby depending on the service one server can accomodate several clients at one time.

The participant to whom the connection is made is called the server. A server does nothing on his own, but just waits for a client to make contact with him.

In reference to TCP, the terms TCP Client and TCP Server are used.

TCP verifies the sent user data with a checksum and assigns a sequential number to each sent packet. The receiver of a TCP packet uses the checksum to verify correct receipt of the data. Once a TCP server has correctly received a packet, a predetermined algorithm is sued to calculate an acknowledgement number from the sequential number. The acknowledgement number is returned to the client with the next packet it sends as an acknowledgement. The server likewise assigns a sequential number to the packets it sends, which is then in turn acknowledged by the clilent with an acknowledgement number.

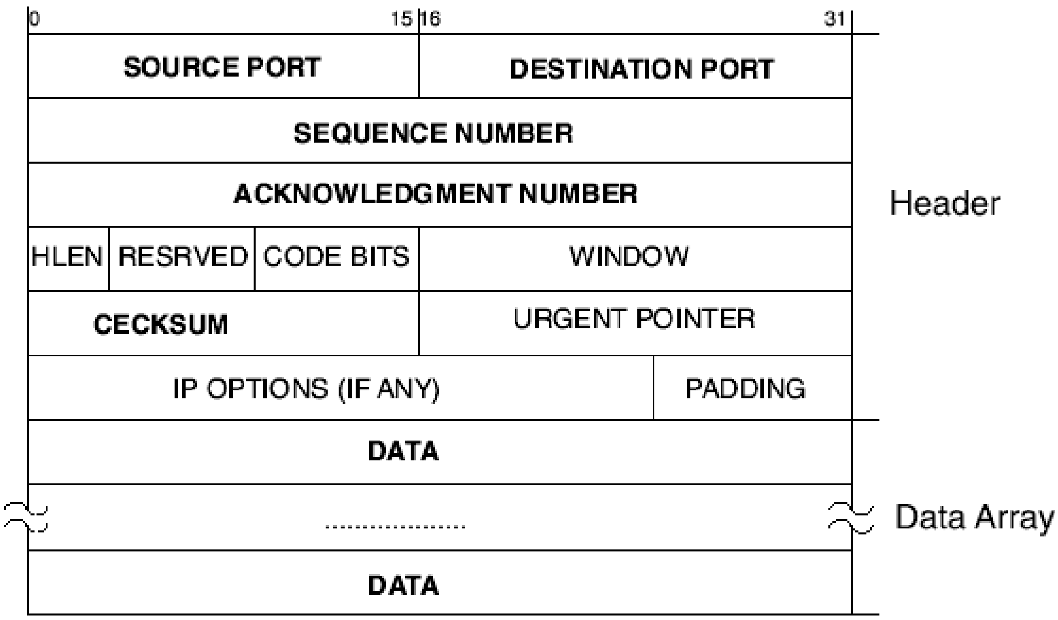
All of this ensures that any loss of TCP packets will be noticed, and that if needed they can be resent in the correct sequence.

In addition, TCP directs the user data on the destination computer to the correct application program by accessing various applications – also called services – through various port number. Thus telnet for example can be reached through Port 23, and FTP through Port 21.

If one compares a TCP packet with a letter to an official agency, the port number would correspond to the room number at the office building. If for example the Sanitation Department is located in Room 312 and you address a letter to this room, you are already indicating that you wish to use the services of the Sanitation Department.

Like IP, TCP also packs the user data into a frame containing additional information. Such TCP packets are constructed as follows :

Structure of a TCP packet



Source Port : Port number of the sender's application

Destination Port : Port number of the receiver's application

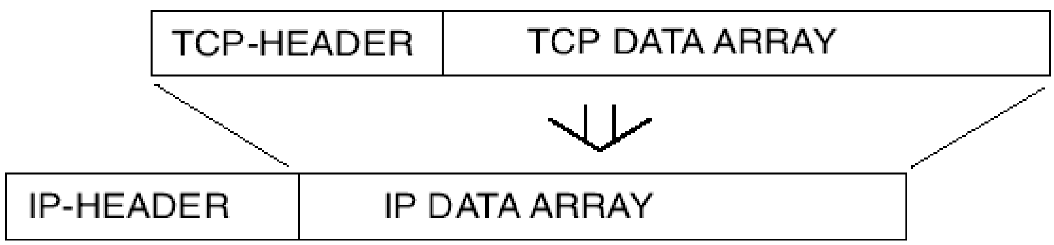
Sequence No : Offset of the first data byte relative to the start of the TCP flow (guarantees that the sequence is maintained)

Acknowl. No : Sequence No. expected in the next TCP packet

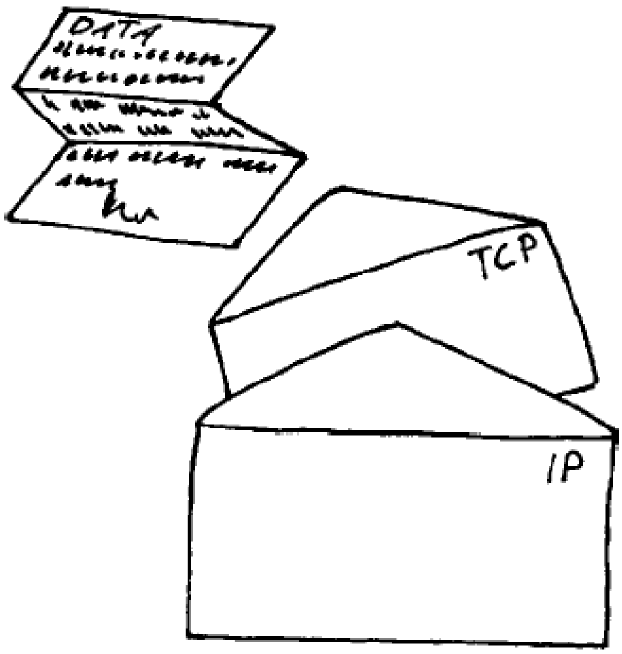
Data : User data

This TCP packet is inserted into the data array of the IP packet..

Construction of a TCP/IP data packet



The user data are placed in something like an envelope (TCP packet), which in turn is placed in another envelope (IP packet).



UDP - User Datagram Protocol

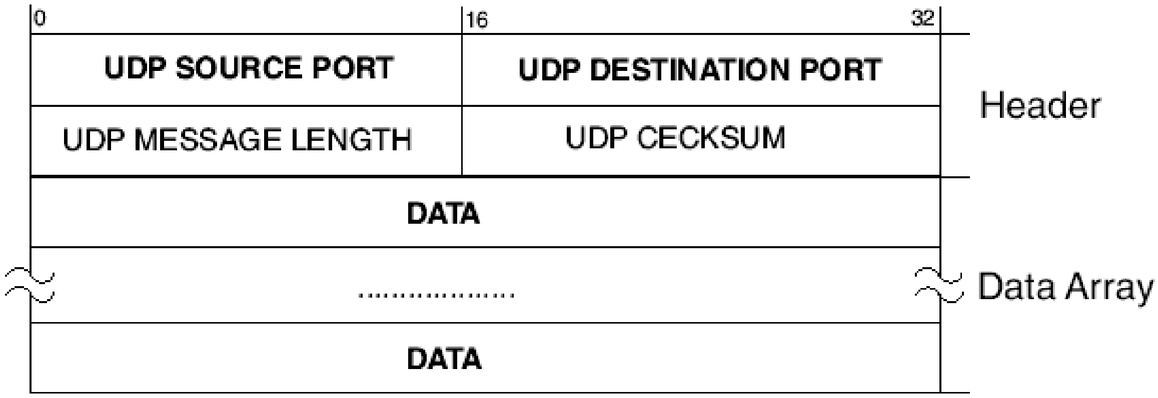
UDP is another transport protocol, which like TCP lives above IP.

But in contrast to TCP, UDP is connectionless. Each data packet is treated like a separate mailing, and there is no confirmation as to whether a packet was received.

Since UDP does not require connections to be established and broken off and therefore no timeout situations can arise, UDP can be faster than TCP : if a packet is lost, data transmission will continue unhindered as long as there is a higher protocol responsible for repetitions.

Data integrity under UDP should in any case be handled by the application program.

Construction of a UDP data packet



Source Port : Port No. of the sending application (reply port for receiver).

Destination Port : Target port at the receiver where the data should arrive.

The rule of thumb is:

• TCP is generally used for continuous data streams or large quantities of data, as well as in situations where a high degree of data integrity is required.

• UDP makes sense when transmission parameters are changing frequently and when data integrity can be assured by a high-order protocol.

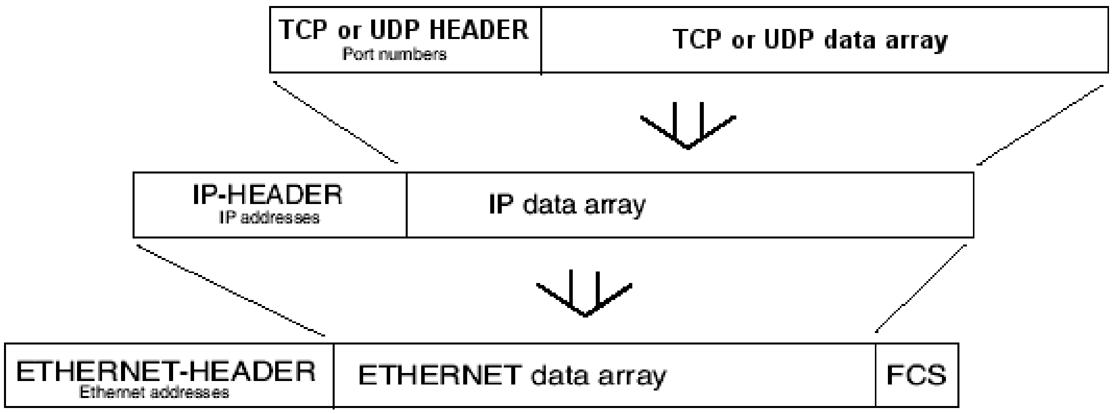
TCP or UDP / IP / Ethernet

TCP/IP and UDP/IP are purely logical protocols and always need a physical foundation.

As already mentioned earlier, Ethernet is the most widely used of the physical network topologies. This is also why you find Ethernet as the physical basis in most UDP-TCP / IP networks.

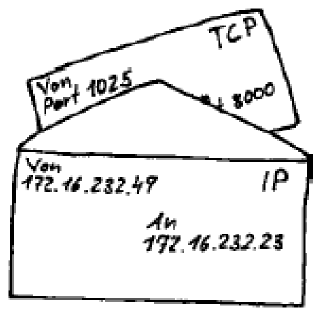
UDP-TCP/IP and Ethernet are merged by embedding each UDP-TCP/IP packet into the data array of an Ethernet packet.

Construction of a TCP/IP-Ethernet data packet



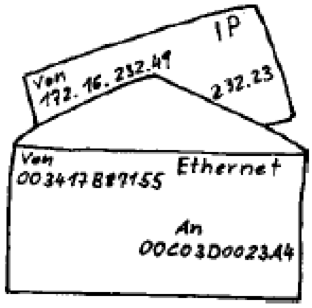
The user data pass through several driver levels on their way from the application on the PC into the network:

- The application program decides which other network stations should receive the data, and hands the IP address and TCP port number to the TCP/IP driver (also called the TCP/IP stack).
- The TCP/IP driver coordinates the organization of the TCP connection.
- The user data handed over by the application program are divided depending on size into smaller, transmittable blocks.
- Each data block is first packed by the TCP driver into a TCP packet.
- The TCP driver hands over the TCP packet and the IP address of the target to the IP driver.
- The IP driver packs the TCP packet into an IP packet.



- The IP driver looks up the Ethernet address of the target specified by the IP address (more on this later) in the so-called ARP table (Address Resolution Protocol) and hands the packet together with the determined Ethernet address to the Ethernet card driver.

• The Ethernet card driver packs the IP packet into an Ethernet packet and sends this packet to the network through the network card.



At the receiver end the procedure is carried out in reverse order:

• The Ethernet card recognizes from the destination Ethernet address that the packet is intended for the network station and passes it to the Ethernet driver.

• The Ethernet driver isolates the IP packet and passes it to the IP driver.

• The IP driver isolates the TCP packet and passes it to the TCP driver.

• The TCP driver checks the contents of the TCP packet for correctness and passes the data using the port number to the correct application.

This multi-layered transmission procedure may seem incredibly complicated at first glance. But only such strict separation of logical protocol (TCP/IP) and physical protocol (Ethernet) makes it possible to exchange data among networks and independent of hardware considerations.

ARP – Address Resolution Protocol

As we have seen, the IP driver hands both the IP data packet and the physical Ethernet address to the Ethernet card driver. To determine the Ethernet address of the target, the IP driver uses Address Resolution Protocol (ARP).

Every TCP/IP-capable computer contains an ARP table. The ARP table is updated as needed by the TCP/IP driver and contains the relationship of IP addresses to Ethernet addresses.

When an IP packet needs to be sent, the IP driver first looks to see whether the desired IP address is already contained in the ARP table. If yes, the IP driver passes the determined Ethernet address together with its IP packet to the Ethernet card driver.

If the desired IP address can't be found, the IP driver initiates an ARP request. An ARP request is an all-call (also referred to as a broadcast) to all the stations in the local network.

To make sure the broadcast is noticed by all the network stations, the IP driver uses FF FF FF FF FF FF as the Ethernet address. An Ethernet packet addressed with FF FF FF FF FF FF is always read by all network stations.

The desired IP address is specified in the IP packet as the destination, and the identifier for ARP is indicated in the Protocol field of the IP header.

Whichever network station recognizes its own IP address in this ARP request confirms this with an ARP reply. The ARP reply is a data packet addressed to the ARP request sender on both the Ethernet level and the IP level, with the ARP identifier in the Protocol field.

The IP driver can now associate the Ethernet address obtained from the ARP reply with the desired IP address, and enters it in the ARP table.

In normal situations the entries do not remain permanently in the ARP table.

If an entered network station is not contacted within a certain time (around 2 min. under Windows), the corresponding entry is deleted. This keeps the ARP table streamlined and allows exchange of hardware components while maintaining the IP address. These time-restricted entries are also referred to as dynamic entries.

In addition to dynamic entries there are also static entries, which the user himself creates in the ARP table. The static entries can be used for passing the desired IP address to new network components which do not yet have an IP address.

Now we know what information is needed for a TCP/IP Ethernet connection in the local network. What we don't have yet is the information for allowing an extra-network connection.

Gateway and Subnet Mask

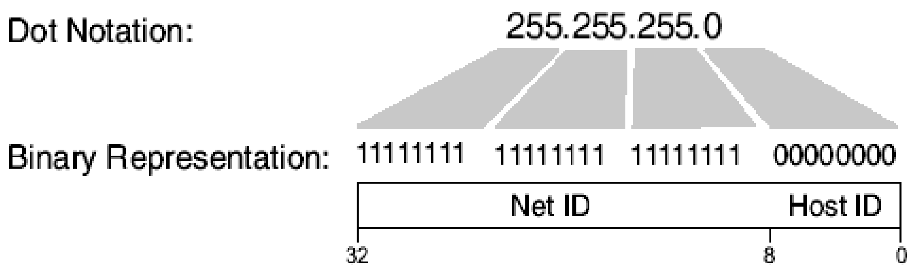
Whether a receiver to whom the connection is to be made is located in the same network as the sender is recognized from the Net ID - the part of the IP address which addresses the network. If this part of the IP address is the same for the sender and receiver, then both reside in the same network, and if there is no agreement, then the receiver can be found in a different network.

The various individual networks are connected to each other through gateways/routers, together forming the Internet.

For network classes A, B and C it is clearly defined which part of the IP address is the Net ID and which the Host ID.

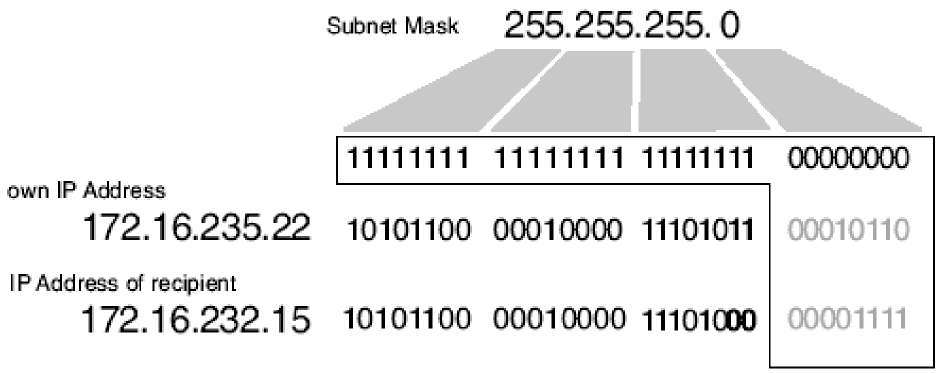
It is however possible to divide a network - regardless of which network class - into sub-networks. But the Net ID provided by the individual network classes is not sufficient for addressing such subnets; you must allocate a part of the Host ID for addressing the subnets. In plain English this means that the Net ID gets bigger and the Host ID correspondingly smaller.

Which part of the IP address is interpreted as the Net ID and which part as the Host ID is specified by the subnet mask. Just like the IP address, the subnet mask is a 32-bit value represented in dot notation. If you look at the subnet mask in binary format, the Net ID section is filled with 1's and the Host ID section with 0's.



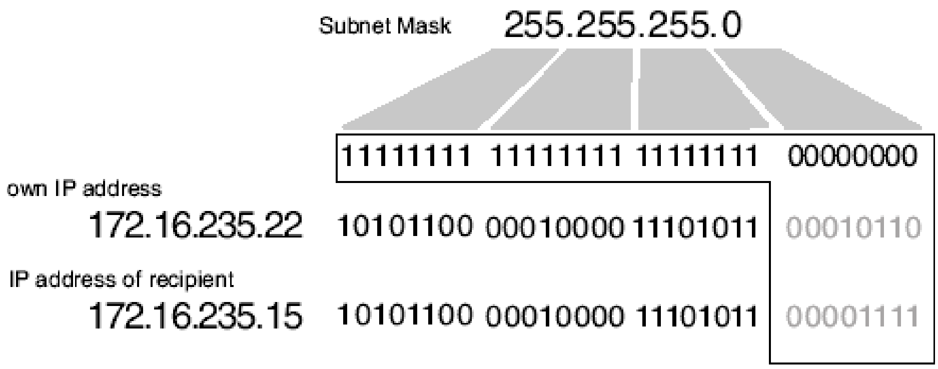
Before each data packet is sent, the IP driver compares its own IP address with that of the receiver. In this process the bits in the Host ID are masked over the part of the subnet mask which is filled with zeros.

If the interpreted bits are identical in both IP addresses, the selected network station is located in the same subnet.



In the example above the IP driver can determine the Ethernet address through the ARP and pass it to the network card driver for direct addressing.

But if even one of the processed bits is different, the selected network station does not live in the same subnet. In this case the IP packet must be passed for further transmission to the target network through the gateway or router.



The IP address of the desired network station is entered in the IP packet. The IP driver uses the ARP to determine not the Ethernet address of the desired network station, but rather the Ethernet address of the router.

Gateways and routers are basically nothing more than computers having two network cards. Ethernet data packets which are received at card A are unpacked by the Ethernet driver, and the received IP packet is passed to the IP driver. This verifies whether the target IP address belongs to the subnet connected to card B and the packet can be delivered directly, or whether the IP packet needs to be passed to a different gateway.

In this way a data packet can pass through several gateways or routers on its way from one network station to another. Whereas on the IP level the IP address of the receiver is entered along the entire path, on the Ethernet level only the next gateway is addressed. The Ethernet address of the receiver is only inserted into the Ethernet packet on the link from the last gateway/router to the receiver.

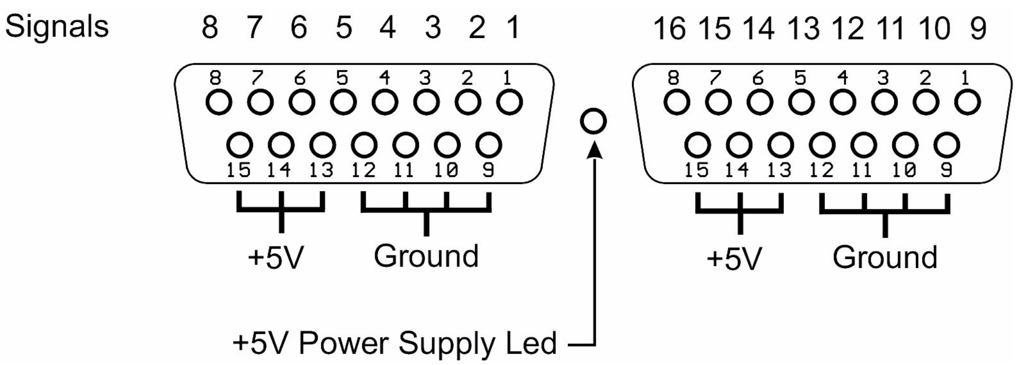
In addition to routers which connect one Ethernet subnet with another Ethernet subnet, there are routers which change the physical medium - for example from Ethernet to token ring or ISDN. While the IP addressing remains the same over the entire route, the physical addressing vom one router to another is adjusted to the physical conditions required on the links.

Appendice C - Connect Sensors

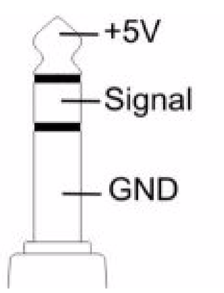
This section gives helpful advices on connecting sensors to an EtherSense.

Wiring

Sub-D 15 connectors for one card



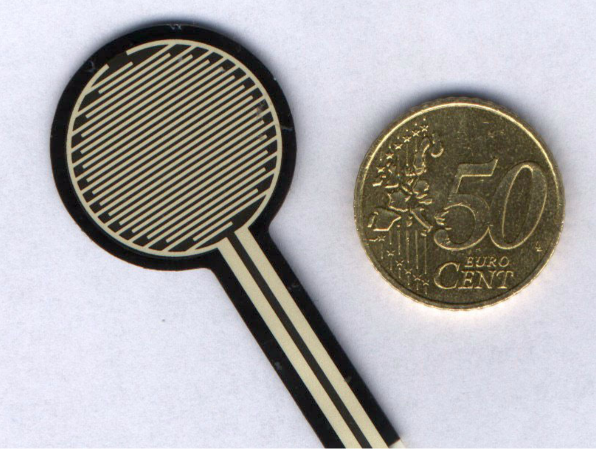
Male jack 1/4’’ - 6.35 mm to use with the breakout cable :

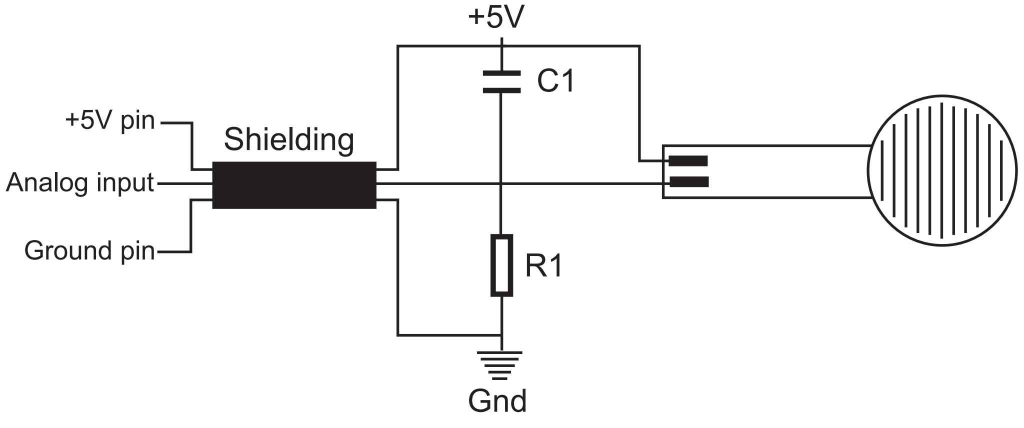


Sensors

Using an FSR pressure sensor

An FSR (Force Sensitive Resistance) sensor measures the mechanical pressure applied to its surface. It comes in the form of a sensitive disk, of variable size depending in the model. The sensor is made of a material whose electrical resistance (ohmic resistance) goes down when the pressure on the sensor goes up. By wiring this sensor into a potential divider form, a continuous analog signal which is proportional to the pressure applied to the sensor can be obtained.



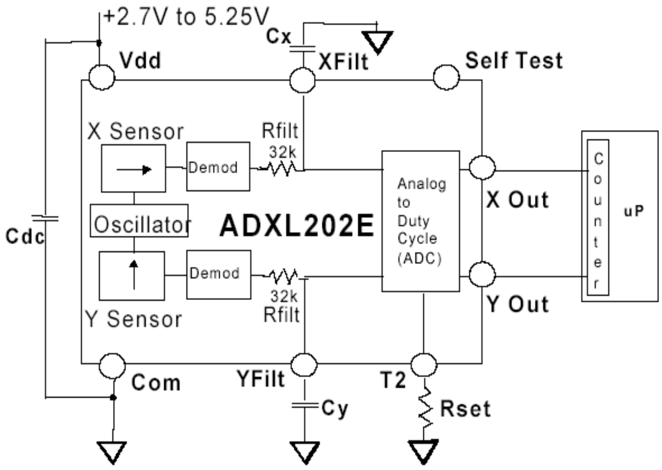


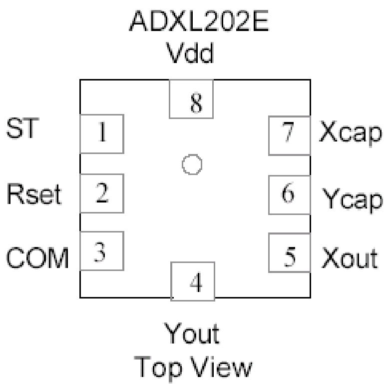
R1 : resistor. Possible values 47kOhms to 100kOhms depending on the required sensitivity (the higher the resistance, the higher the sensitivity to pressure)

C1 : plastic capacitor 220 nF

Using an accelerometer : ADXL202E

This sensors measures accelerations on two axis with a full-scale range of +/- 2g. It can measure both dynamic acceleration (e.g., vibration) and static acceleration (e.g., gravity). Its outputs are digital signals (Xout, Yout, not usable by the EtherSense) and analog voltage (XFilt=Xcap, YFilt=Ycap).

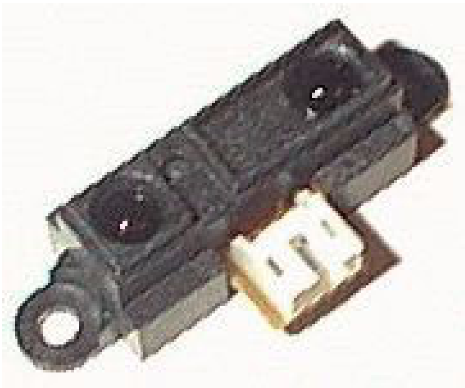




Use 50nF capacitors for Cx and Cy (between XFilt (and YFilt) and Ground) to filter values at 100 Hz. Smoother values might be obtained by increasing capacitor values.
Connect Xcap (XFilt) and Ycap (YFilt) to analog input pins of the EtherSense, Vdd. to +5V pin and Com to Ground Pin

A decoupling capacitor (Cdc) should be used between Vdd (= +5V) and Ground : use 100nF capacitor.

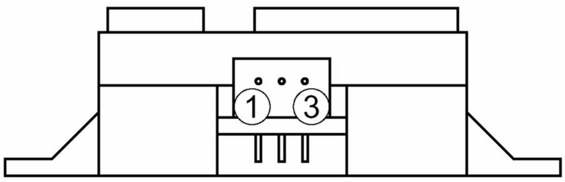
Sharp Infrared Rangefinder GP2D12



This infrared device senses the distance to a reflective object within a range of 10 to 80 cm. It outputs a continuous voltage proportional to the measured distance.

Powered with +5V, the device will only output 2.5V full scale. However the 16 bit resolution of the EtherSense is sufficient get usable values. Moreover an analog gain (internal to the EtherSense) can be applied to the choosen channel.

The Sharp IR is known to have a very noisy output and current spikes. Therefore it is recommended to use a separate regulated and filtered (using a self inductance) power supply for this sensor.



Pin 1 : Signal

Pin 2 : Ground

Pin 3 : Vcc = +5V (or up to +7V)

Index

Numerics

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

R

S

T

U

V

W

X

Notes

Notes

Notes