Sirion Server Layout

Preface

This document is a guideline for setting up a TCP/IP server for the reception of Sirion module signals.

General description

The Sirion module is a TCP/IP enabled module that sends alarm panel events to a receiving station. In order to achieve this, a connection to the alarm panel simulating a PSTN line is used. The connected alarm panel uses this line just like it would do with a normal PSTN line, expecting to find a dedicated events receiver on the other end. Sirion is simulating the receiving station by simulating this receiver capable of decoding two communication formats; Contact ID and Ademco Express. The events received on this simulated receiver are sent in real time over TCP/IP to a Monitoring Station. This requires a receiver system capable of decoding the TCP/IP encoded data and inserting them into the existing setup as a normal receiver would do.
Figure 1 below shows a typical setup for the Sirion module and monitoring station connection.
Before we proceed any further it is important to understand the minimum requirements and some basic TCP/IP principles.

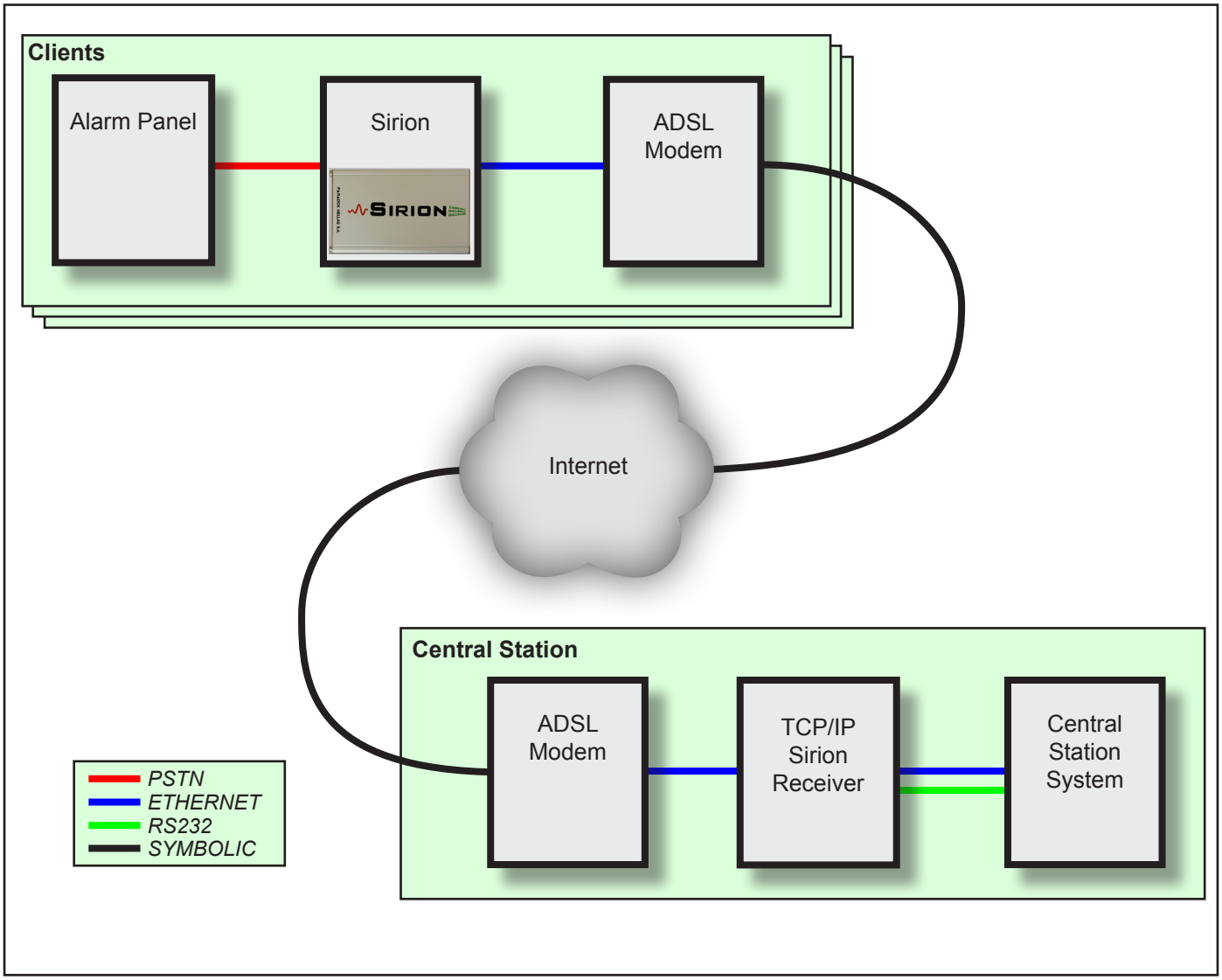

Figure 1. General connection diagram.

TCP/IP principles

A very good analogy of the TCP/IP system over the Internet is the Public Telephone System. Since most people are familiar with it, will be used to explain some of the following:

IP Address

The IP address of a network node is its unique ID on the Network that is connected. It is the equivalent of a telephone number. Every network node connected directly to Internet must have a unique IP address in order to be able to properly communicate.

Static IP Address

A static IP address is one that is assigned to a network node (an ADSL modem for example) that is always the same. When a modem is disconnected it does not have an assigned IP address. Once the modem connects, during a successful user authentication process, gets a unique IP address from the Internet Service Provider (ISP). If the type of account is such, the same user gets always the same IP address.
This is in opposition to the Dynamic IP address where every time the customer connects the first free unique IP address from the ISP is assigned (every ISP has a number of allocated IP addresses that can use for connecting customers). This random IP address at every connection prevents other users/computers from reaching him. It's like if someone keeps changing the telephone number of a subscriber; she would be able to dial out but she could not be reached because no one would know her telephone number at a specific time.

Port Forwarding

Imagine a company with 100 employes all working at the same location. This imaginary company has one public telephone number, an internal telephone switch and one internal telephone for each of it's employes with his own extension. One of his employes is in charge of orders and another for technical support. A customer calling the main telephone number would need to know the correct extension in order to speak with the person he wants.
This is the perfect analogy for the IP address and the TCP/IP port numbers as used for port forwarding. An external system connecting to the Static IP of this company, would need to end up connected with the appropriate system behind this IP. The system administrator must set up two different ports, one for orders and one for technical. Once a computer system is connected to the main IP it requests the specific port. The request for connection is automatically transferred to the appropriate internal server IP according to the requested port (IP requests are almost always accompanied by the wanted port). This automated process is called port forwarding and uses a table that describes the associations between ports and Internal IP addresses. In most cases this table is kept in the ADSL modem. The mechanism for defining it differs from manufacturer to manufacturer, but the principle is the same.
With Sirion for example, the default port for communication is 7700. This port must be forwarded to the internal IP of the computer where the Sirion Server is installed.

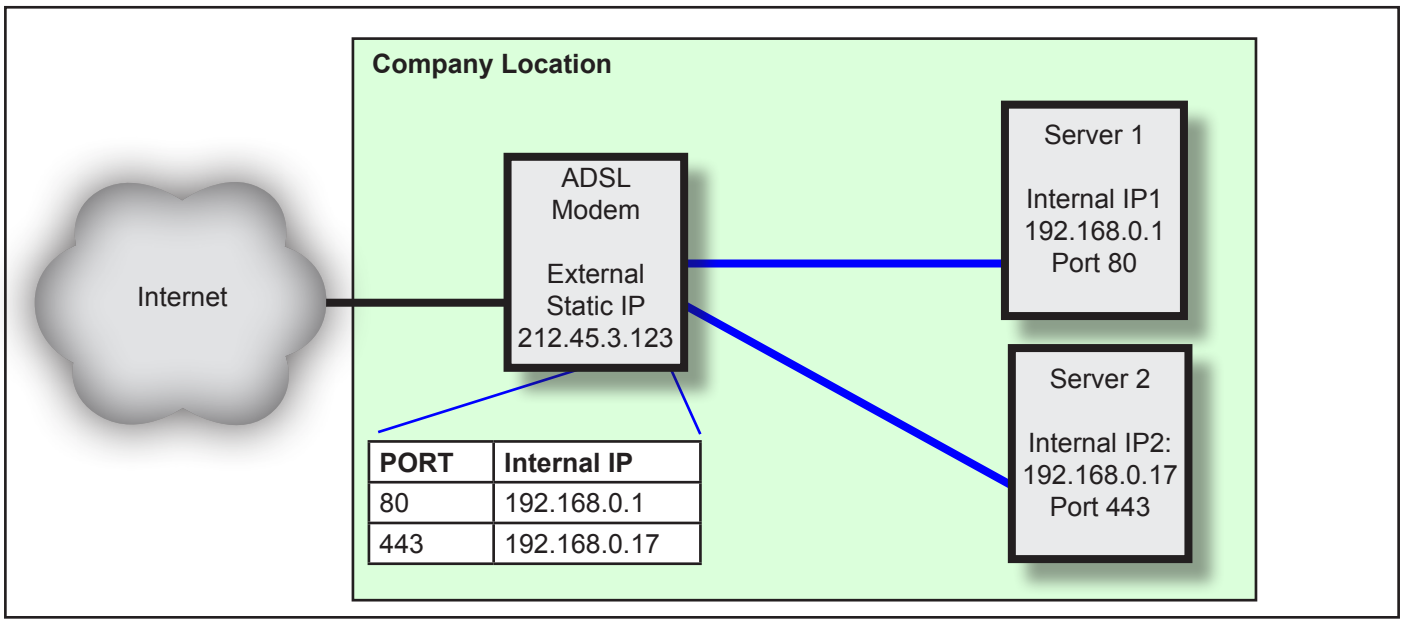

Figure 2. Port forwarding.

Minimum requirements

The minimum requirements depend on the setup that will be chosen. Thus the requirements will be summarized in every section that follows. The absolute minimums which are common for any of the following setups are:

- Internet connection
- Static IP for the Internet connection
- ADSL modem that supports port forwarding and network address translation

Monitoring Station Setups

Introduction

Monitoring Station's events handling system may have any of many different approaches and implementations. The principle though is the same : Incoming events in the form of characters are inserted at some point with various methods. Thus the problem we are faced with is this: **An alarm panel event has been received through TCP/IP; inject it at a point with an accepted protocol into the existing system**.

Paradox Hellas Sirion Server (PH Sirion Server)
In the following paragraphs the PH Sirion Server is mentioned. This is a software package provided by Paradox Hellas S.A. at no cost to anyone interested.
It is a WIN32 application that has the following capabilities :

- Listens on a TCP/IP port for incoming events (from Sirion units)
- Simulates the MCDI expresium receiver (more receivers will be added)
- May send the received events to a serial port or
- May send the received events over TCP/IP to internal software
- Keeps log files with all received events

Paradox Hellas Communication Protocol Library
In some setups the monitoring station may include support for the Sirion messages directly into their software. Sirion has a communication protocol, the PH Communication Format (PHCF). PHCF includes support for simple or secure communications and is an evolving protocol with backward compatibility. In order to assist software developers Paradox Hellas provides the full specifications as well as a runtime library for the WIN32 platform. This library automates the encoding, decoding, error checking as well as AES encoding/decoding. It is available to interested parties after a Non Disclosure Agreement is signed.

Setup 1: Sirion Server on standalone PC, serial connection to main Software

In this setup, the PH Sirion Server is run on a standalone PC. The port forwarding sends to this PC all the incoming events. The Sirion Server formats this messages in MCDI expresium and sends them to the Main Receiving Software via a serial port. The Main Receiving Software is running on a different computer and listens for such messages on one of it's serial ports. It is configured in a way that expects to receive MCDI expresium format on the used serial port. In figure 3 such a system is depicted.

Requirements

- PC with Win32 platform (WinXP, Win2000, Windows 2000/2003 server)
- Free RS232 port on the 'Sirion Server" PC
- Free RS232 port on the computer running the Main Receiving Software.
- RS232 cross over cable.

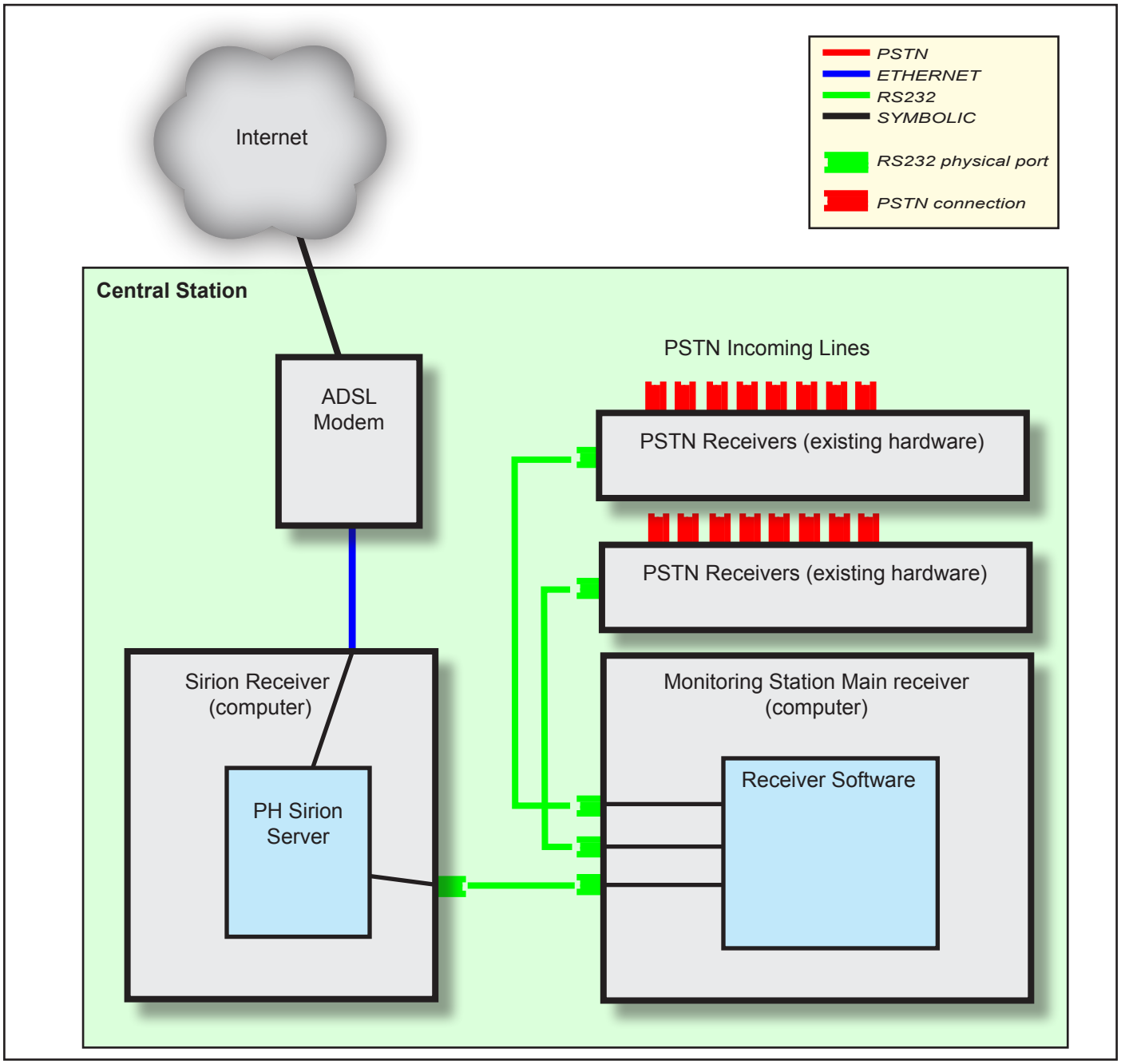

Figure 3. PH Sirion Server with serial link.

Setup 2: Sirion Server with Main Software on one Computer, RS232 loopback

In this setup, the PH Sirion Server is run on the same PC with the Main Receiving Software. The port forwarding sends to this PC all the incoming events. The Sirion Server formats this messages in MCDI expresium and sends them to the Main Receiving Software via a serial port. The Main Receiving Software listens for such messages on another serial port and it is configured to receive MCDI expresium format. These two physical serial ports are linked together with a crossover serial cable.
In figure 4 such a system is depicted.

Requirements

- PC with Win32 platform (WinXP, Win2000, Windows 2000/2003 server) for Sirion Server and Main Software.
- Two free RS232 ports on this computer.
- RS232 cross over cable.

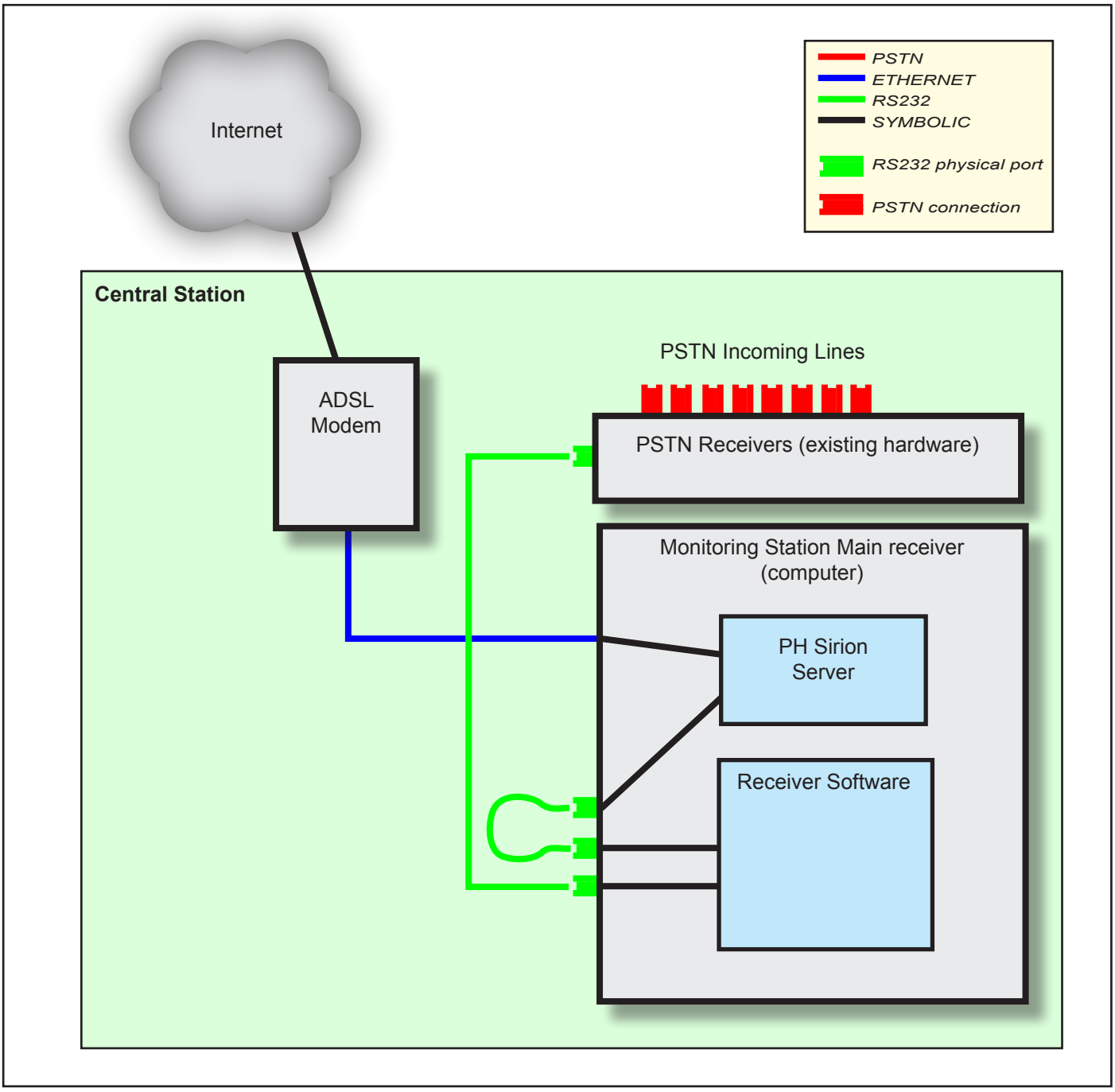

Figure 4. PH Sirion Server With loopback serial link.

Setup 3: Sirion Server on standalone PC, virtual serial connection to main Software via ethernet.

In this setup, the PH Sirion Server is run on a dedicated PC. On the PC with the Main Receiving Software there is installed a virtual serial com port driver. This virtual com port is linked over ethernet and TCP with the Sirion Server software. With this setup anything that Sirion Server receives from it's public interface is forwarded to the virtual com port. The Main Receiving Software listens on this virtual com port.
As in previous examples, port forwarding sends to the Sirion Server PC all the incoming events. The Sirion Server formats this messages in MCDI expresium and sends them to the Main Receiving Software via ethernet and virtual serial port. The Main Receiving Software listens for such messages and is configured to receive MCDI expresium format.
In figure 5 such a system is depicted.

Requirements

- PC with Win32 platform (WinXP, Win2000, Windows 2000/2003 server) for Sirion Server
- Virtual serial port driver installed on computer with Main Receiving Software. Must support raw data transfer.

Note that regarding the virtual com port the Sirion Server is acting as a TCP server and the virtual com driver as the client.

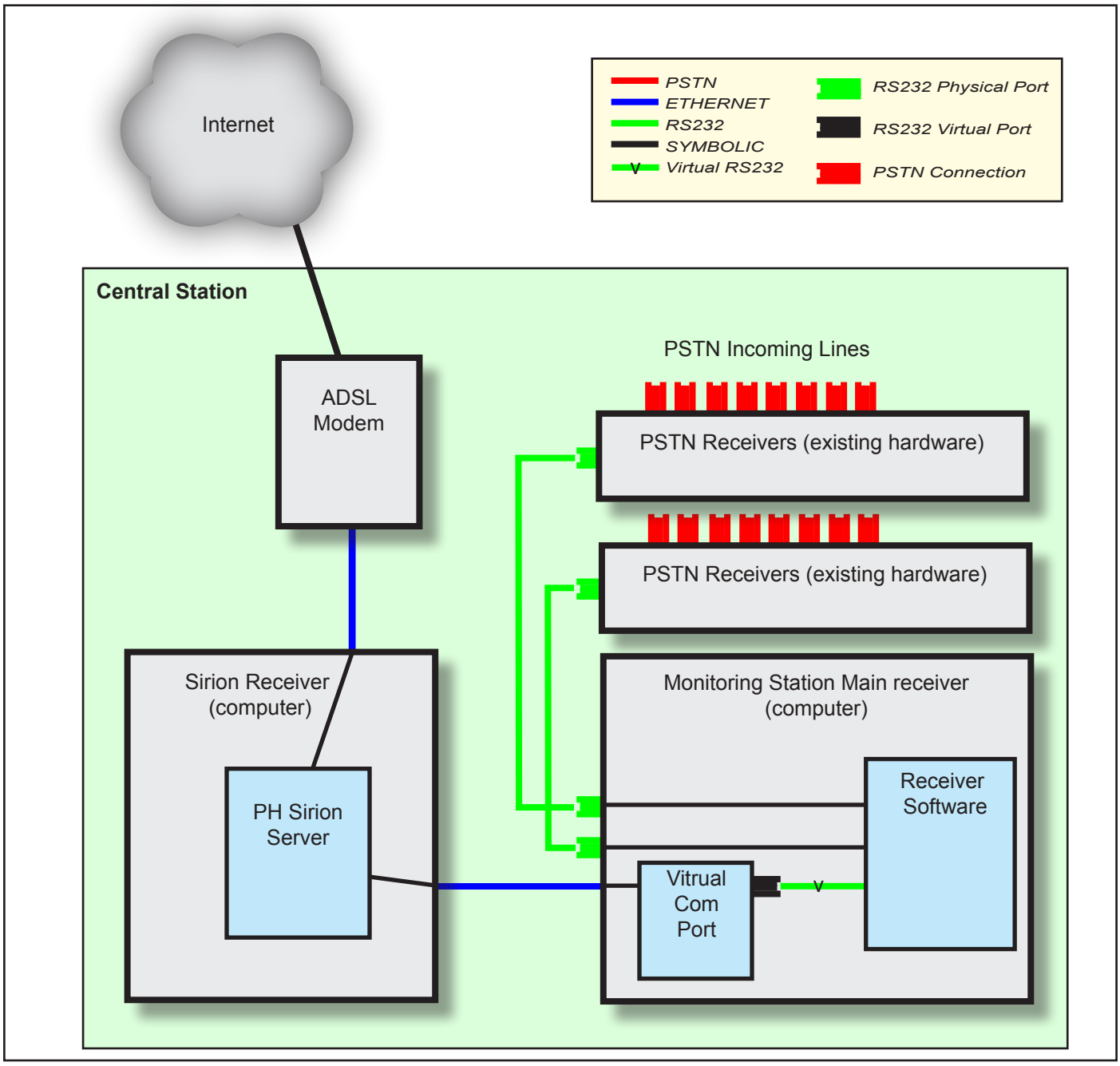

Figure 5. PH Sirion Server with virtual serial link over TCP.

Setup 4: Sirion Server on standalone PC, TCP connection from main Software (via ethernet).

In this setup, the PH Sirion Server is run on a dedicated PC. The Main Receiving Software is capable of opening a TCP connection and receiving event data formatted in MCDI expresium format over it. With this setup anything that Sirion Server receives from it's public interface is forwarded over the TCP connection to the Main Receiving Software. As in previous examples, port forwarding sends to the Sirion Server PC all the incoming events. The Sirion Server formats this messages in MCDI expresium and sends them to the Main Receiving Software via ethernet and TCP sockets. The Main Receiving Software listens for such messages and is configured to receive MCDI expresium format. In figure 6 such a system is depicted.

Requirements

- PC with Win32 platform (WinXP, Win2000, Windows 2000/2003 server) for Sirion Server
- Main Receiving Software capable of connecting with TCP/IP with communications receiver. Must support MCDI expresium format over this link.

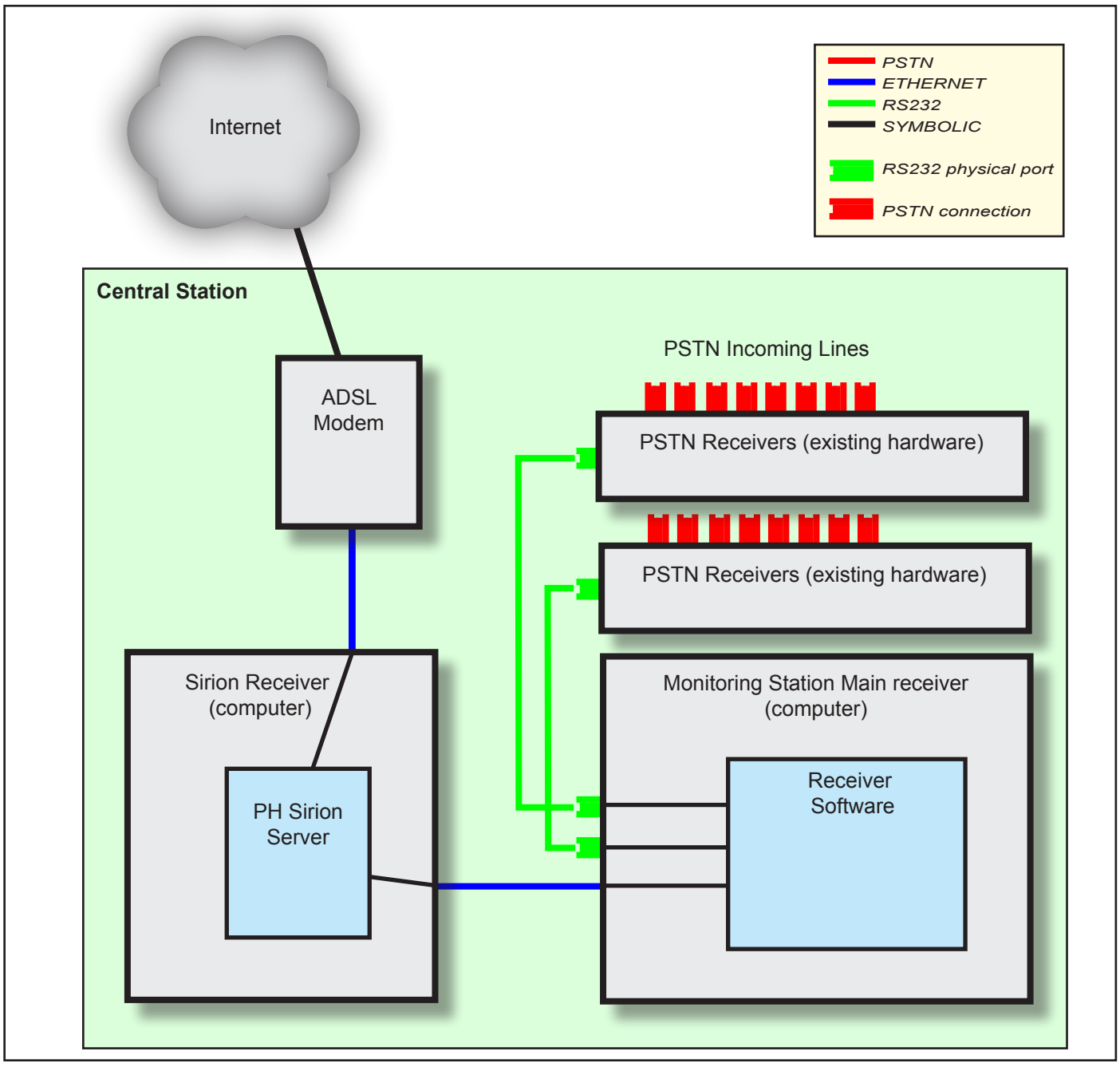

Figure 6. PH Sirion Server with direct TCP link.

Setup 5: TCP connections directly to Main Software.

In this setup, the Main Receiving Software is capable of receiving events from Sirion units over TCP connections. The received format is described in the “PHComunicationFormat” document. Since this protocol has to be implemented in software Paradox Hellas provides the main encapsulation and encryption/decryption routines in a win32 dll file. The use of this file is optional.
Port forwarding from the ADSL modem sends all the incoming events to the Main Software. The Main Receiving Software listens for such messages and inserts them directly (after decode and authentication) in the Monitoring Station’s System.
Figure 7 shows such a system.

Requirements

- Software development of the PH Communication format in the Main Receiving Software.

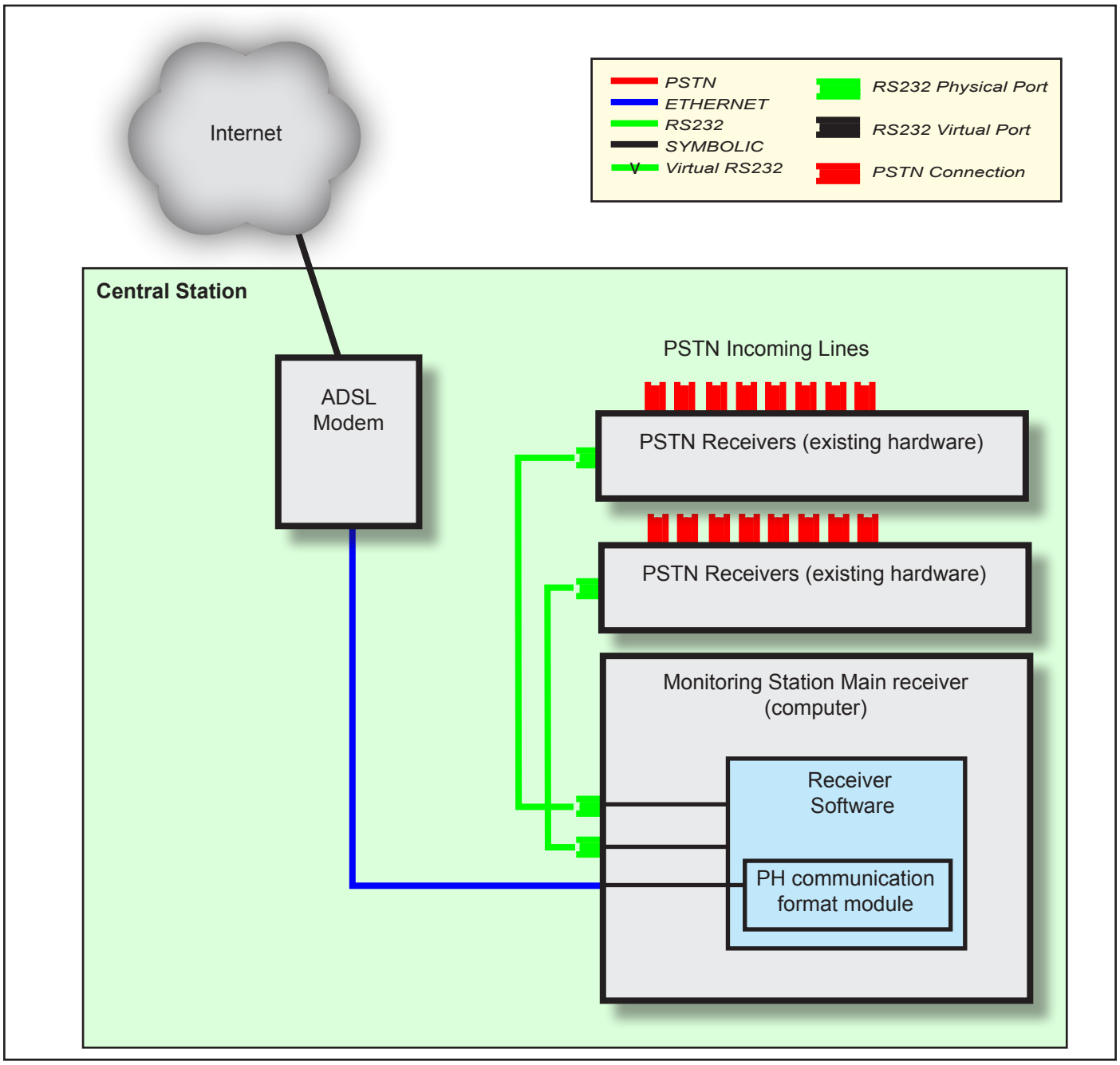

Figure 7. Main Receiving Software with embeded Sirion Server .

Epilogue

The setups described in the previous sections, are indicative. There are many more combinations that will produce a successful receiving system.
Also, the format that the Sirion Server software sends over the local link to the main receiving software is not limited to MCDI Expresium.
Newer versions will support more formats. As of version 1.03 there is support in beta stage for simulation of Osborne Hofman, Surgard and Raw format.
The latest, will send the raw data that the alarm panel sends in the DTMF format (Ademco express or Contact ID digits).

The latest versions of this document, Sirion Server and anything else related to Sirion is located in the following address :

http://www.paradox.gr/sirion/sirion_links.html

END-OF-DOCUMENT