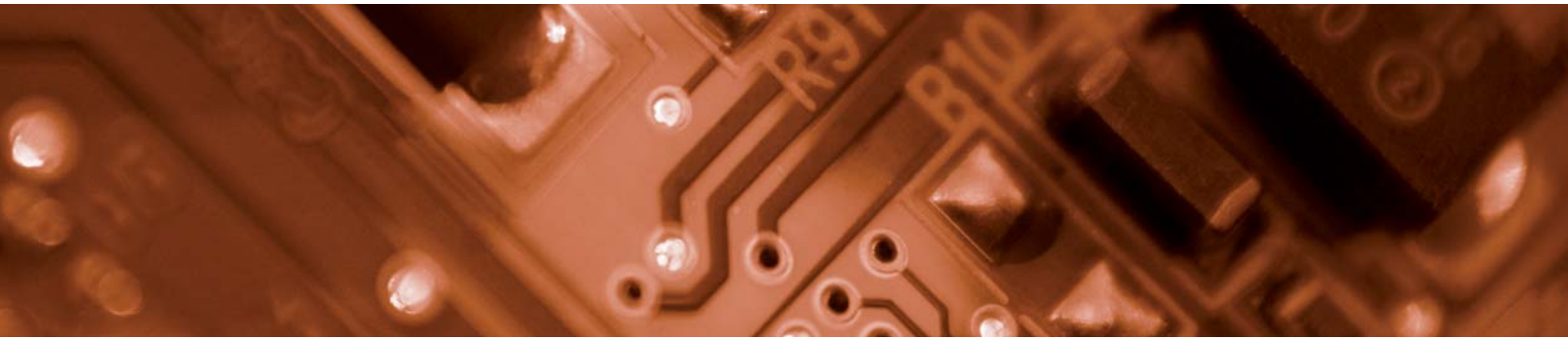
bsia
british security industry association

Universal SPT Interface

– an industry guide



February 2009

For other information please contact:

British Security Industry Association
t: 0845 389 3889
f: 0845 389 0761
e: info@bsia.co.uk
www.bsia.co.uk

Form No. 251. Issue 1

Contents

1. Scope

The purpose of this document is to define a universal interface for use between supervised premises transceivers (SPT) and supervised premises applications (SPA) such as I&HAS (Intruder and Hold-up Alarm System, video transceivers, audio monitoring systems and others.

The need for this interface to be defined has come about due to:

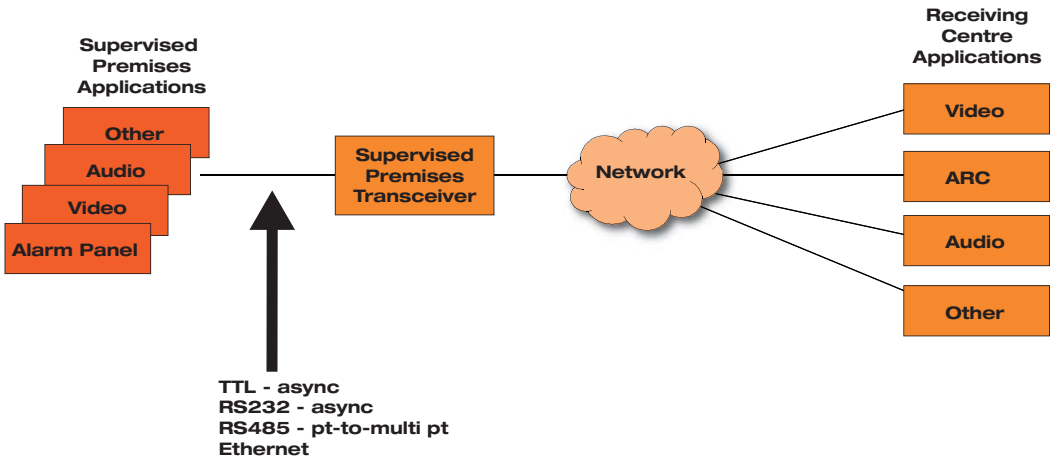
a) The increased complexity in the communication process between supervised premises and alarm receiving centre applications leading to the introduction of independent suppliers of supervised premises transceiver equipment.
b) Multiple supervised premises applications wanting to use the same transmission networks, with each application potentially having its own transmission requirements.

This interface standard will not provide a universal one-off implementation, but instead will provide a solid framework which can be used by the developers of supervised premises applications and transceivers to more easily develop compatible interface handlers. The aim being to make the time taken to complete the integration process as short as possible.

What this document does not cover is:

a) The functionality of the supervised premises applications.
b) The delivery mechanisms for the transfer of data to and from the supervised premises.
c) The content and processing of the data being transported except in a limited number of cases.

2. Functions



The above diagram shows the location of the interface covered by this document namely the connection between the supervised premises applications (SPA) and the supervised premises transceiver (SPT).

The diagram shows a number of applications and these could be connected over multiple physical interfaces to the SPT.

The requirement to be able to use an Ethernet interface between the SPAs and the SPT is for consideration. In this case the SPT may be a device on a shared LAN where the SPT is being used to provide the secure transmission of data to the receiving centre applications. It is believed that whatever underlying transport protocol is used on the LAN (TCP or UDP) the same messaging interface described here could be used.

The functions that the interface should support are given below. While this may not be an exhaustive list, it should be considered as the basis for the capability of the standard:

a) Transfer of short format information, such as event reporting, from the SPA for delivery to a receiving centre application.
b) Transfer of short format information, such as control commands, from a receiving centre application to the SPA.
c) Transfer of system data to and from the SPA and a receiving centre application.
d) Transfer of 'local' control and configuration information between a SPA and the SPT.
e) Handling for streaming data such as audio and video.

The design of the interface standard allows for the concurrent operation of different functions.

The design of the interface standard provides a flexible solution which will allow for the addition of new functions in the future.

From these functions it is clear that there are three types of data format that need to be supported:

a) Short format data – single packets of data that may simply require a return acknowledgement. Examples are event reports and configuration data (SPT configuration and not necessarily that covered by traditional 'upload/download' functions).
b) File transfer data – blocks of data sent as part of a set of data. An example would be the transfer of the system configuration of a SPA to or from a receiving centre application (upload/download).
c) Streaming data – for applications using video or audio the interface must be able to stream the packets to and from the receiving centre application with as little interference as possible

In order to support multiple concurrent functions, the interface must provide some form of virtual channelling between the SPA and the SPT.

In the case of I&HAS applications (intruder alarm panels) the SPT may be considered as a component of the I&HAS. If this is the case, then the SPT has certain requirements imposed on it with regard to the communication between it and the rest of the system. It may be good practice to employ these principles for other applications. The requirements are as follows:

a) Substitution detection.
b) Tamper detection (detection of any interference with the operation of the interface).
c) Reporting of the SPT communication status.

I&HAS performance is defined by a set of standards which allow for different grades of system. The level of sophistication required to meet the above requirements will depend on the overall security grade for the system as a whole. This means that the resulting interface standard must allow for all levels of grading.

Whether or not the interface needs to be encrypted is for further consideration. However, for higher levels of substitution detection this may be implied.

Note: It is believed that the current standards covering I&HAS do not clearly indicate whether or not the SPT is a 'component'. If it is not then there are no specified requirements. This is further confused by the definition of the location of the interface.

The interface standard does not presume a level of processing capability for either the SPA or SPT. For example the SPT may be a simple modem device which is used purely for the delivery of event information in which case it may not be capable of running a high speed data interface nor a sophisticated encryption and substitution detection algorithm.

3. Interface

3.1 Electrical

Four types of physical interface are considered appropriate for this interface:

a) TTL – conforming to published TTL thresholds
b) RS-232 (V.24/V.28) – conforming to published standards
c) RS-485 – conforming to published standards
d) Ethernet – for further consideration

Options a) and b) provide for flexibility of installation. In some cases the SPT may be an integral component of a SPA, in other words it is mounted inside the SPA enclosure, or it might be a completely separate unit mounted in its own enclosure. The option of a TTL interface therefore allows for cost savings to be made where an extended interconnection is not required. In either case, the operation of the interface will be the same.

Use of an RS-485 type interface will allow for a multi-drop interface providing connectivity for multiple SPAs to one SPT. In this case the SPT will be the bus master, which means that it will not be able to connect to existing peripheral buses provided by some I&HAS equipment. In order for the messaging protocol defined here to be able to operate over an RS-485 type interface it will be necessary to include a message wrapper which handles the addressing of devices on the bus and bus management.

However, by providing an RS485 interface it does make it possible for the SPT to connect to existing bussed systems as a slave device. In this case, the underlying transport mechanism will have to comply with that of the host system, but the messaging protocol would be as defined in this document.

The use of other interface technologies is not restricted and the principles of operation would be the same, but the technology must be appropriate to the application.

For the serial type interfaces, in order to keep the physical interface as simple as possible, it will be restricted to a simple three wire interconnect consisting of send data, receive data and signal ground. Any need for flow control will be handled within the messaging protocol.

Send data (SD) will refer to data from a SPA to the SPT.

Receive data (RD) will refer to data from the SPT to a SPA.

For serial type interfaces data transfer will be asynchronous using traditional methods. Character definition will be 8 data bits, no parity, one stop bit.

The data rate for a serial interface will be configurable. Auto-bauding solutions such as that used by AT command modems are not considered to be appropriate in this application.

Suitable (sensible) precautions should be taken to protect the interface against connection to other interfaces which present voltages outside of the expected operating range.

3.2 Mechanical

On the grounds that a serial interface is a simple three wire interconnect, the recommendation is that the physical interface be a simple screw terminal presentation using 5mm/0.2" pitch components. The interface should also allow for a separate earth connection which may be useful if an interference screen is required. This would give the option to connect the screen to signal ground or chassis earth as appropriate and/or where available.

The order of the connections is not critical, but should be clearly marked using the symbols 'SD', 'RD', '0V' and the IEC earth symbol.

Where other interconnect technologies are employed then the appropriate physical connection arrangements for that technology will be implemented.

4. Protocol description

As stated in the scope this guide does not include a definitive and detailed protocol description. Instead the guide provides a framework that can be used and re-used for different SPA/SPT integrations without restricting innovation and/or imposing unnecessary development overhead. However, that said it will include sufficient detail to provide basic support for I&HAS applications.

The framework specification should provide for the following:

1) Packet format
2) Header definitions
3) Fixed communication channels
4) Substitution detection procedures
5) Limited messaging procedures

4.1 Packet format & header definitions

The packet format is a simple two part 'header – type – length' structure. The two parts are sent sequentially with no logical gap between them.

The first part is fixed length and consists of the following:

a) The header field identifies the virtual channel.
b) The length of the second part.
c) 1 byte checksum of the bytes in the first part.

Note. The purpose of using a fixed length checksumed first part to the packet is that it makes it easier to detect corruption in the length data and therefore reduces the risk of attempting to handle corrupted messages that are excessively long.

The second part of the packet contains the message data and consists of the following:

a) The type field identifies the type of message applicable to the channel
b) The data block
c) 1 byte checksum of the bytes in the second part.

There will be no end of packet marker thereby providing data transparency.

The use of a checksum is mandatory. Details of the specified checksum process are given at the end of this document.

The header field will be 1 byte thereby providing for somewhere in the region of 255 channels although some may be reserved for special functions i.e. data rate negotiation.

The length field will be two bytes long. The length includes the type field, the data block and the checksum included in the second part of the packet.

The type field will be two bytes long which should provide for more than enough message types. Note that the field type is unique to the channel. Therefore the same field type can have different meanings for each channel.

The minimum packet length is therefore 7 bytes and the maximum packet length is 65540 bytes.

The maximum data block size is 65533 bytes.

4.2 Fixed communication channels

The idea of fixed communication channels refers to having pre-defined channels which are always used for the same function. At present three can be identified:

1) Channel used for communication between the SPA and the SPT. This is primarily used for configuration information from the SPA to the SPT and for status information in the other direction.
2) Substitution detection or security channel. This channel is used for managing the substitution detection monitoring process if employed and negotiating any other security functions that may be required i.e. encryption.
3) Initial link establishment and services negotiation – this is a dedicated channel used for negotiating initial communication parameters, such as security and data rate negotiation, and for channel management.

4.3 Message procedures

Message procedures will be kept to a minimum in order to keep them simple, easy to implement and thereby make them reliable.

The message procedures defined here should provide sufficient capabilities for the two end-points to negotiate and maintain operational channels.

Hardware flow control is not provided for this interface and it will therefore be up to the message procedures to manage this.

In its simplest form the message process can be a simple half duplex process where each message in one direction has to be acknowledged in some way in the other. However, this may depend on the nature of the SPA functionality.

In applications other than those where data is to be streamed from the SPA to the receiving centre, it is likely that an end to end, half-duplex message exchange process will be employed.

In data streaming applications however, the SPA is likely to want to stream data as fast as it can with the receiving centre process occasionally sending back responses. In this case it is quite likely that the SPT will need to flow control the SPA itself in order to fit the streaming data to the network being used. In this case it will be a simple case of agreeing a local flow control procedure for the channel being used.

The message procedures described here assume a serial asynchronous link. If other physical links are used such as Ethernet then some procedures will not be required i.e. data rate negotiation.

One fixed channel is currently defined. This is as follows:

Channel number 1 – Initial link negotiation channel and SPA/SPT control/maintenance channel

4.3.1 Initial link negotiation procedure

Channel number 1 – Header byte 0x01		
Type field	**Description**	**Data field**
0x0001	keep-alive	SPT status information SPT to SPA Byte 1
		0x01 ATS1 interface status
		0x02 ATS1 path status
		0x03 ATS2 interface status
		0x04 ATS2 path status
		0x05
		0x06
		0x07
		0x08
		0x09
0x0002	acknowledgement	SPA status information SPA to SPT TBD
0x0003	negative ack.	Rejected requests (where appropriate) SPA to SPT
0x0004	service request	Byte 1 SPT to SPA
		0x01 300bps
		0x02 1200bps
		0x03 2400bps
		0x04 9600bps
		0x05 19200bps
		0x06 38400bps
		0x07 57600bps
		0x08 115200bps
		0x09 230400bps
		Byte 2 SPT/SPA services request
		B1.b0 substitution protection enable
		B1.b1
		B1.b2
		B1.b3
		B1.b4
		B1.b5
		B1.b6
		B1.b7
0x0005	channel close	None SPA to SPT
0x0006	channel close	None SPT to SPA

The purpose of this channel is to provide a means for establishing basic communication between the SPA and the SPT and to provide a means whereby status and control information can be passed between the two devices.

The communication process will be initiated by the supervised premises transceiver using this channel. At start-up the data rate will be 2400bps. The SPT will start communication by transmitting a keep-alive signal on this channel and it will expect to receive an acknowledgement within a configurable timeout. The rate of keep-alive transmissions will be configurable with a default of 10 seconds. The timeout for receipt of an acknowledgement to any message on this channel will be configurable with a default of 5 seconds.

Once successful communication has been established then the SPT and the SPA can negotiate other settings for the operation of this channel.

To negotiate the channel settings the SPT will request of the SPA the modes of operation that it wishes to use. The SPA accepts these settings by returning an acknowledgement. If the SPT requests any settings that the SPA can not accept, then it will return a negative acknowledgement whose data field contains the settings that it will accept. Depending on the sophistication of the settings required or supported, it may take a few attempts for the two ends to agree a working configuration.

Once a configuration has been agreed, the two ends will activate any changes required bearing in mind that this may involve a change of data rate. The new configuration will be confirmed by a re-establishment of the keep-alive process.

Note that if an RS-485 type interface is being used then the data rate is likely to be fixed by the installation (i.e. all devices on the bus have to operate at the same rate). In this case the same link establishment procedure will be used but there will be no change in actual data rate even if a different rate is negotiated.

The keep-alive signal continues to be transmitted on this channel irrespective of what else is happening on the interface. Keep-alives are transmitted when the channel has been idle in both directions for more than five seconds.

If either the SPA or the SPT see a failure of two consecutive keep-alives then the link will be reset and each end will return to the start-up process to re-establish and re-negotiate the link with the SPT transmitting keep-alive packets at the default rate. All sessions on any other virtual channel will be cleared.

The keep-alive process and the responding acknowledgements can also be used to pass status information between the SPA and SPT. The exact definition of this facility is for further discussion but from the SPT's point of view will include information on the status of its transmission path(s).

Either entity can close the link cleanly by using a channel close message. On receipt of this message on channel 1 the receiving entity will first issue a close on any other open channel and then cease operation on channel 1.

Both the SPA and SPT should record the establishment and any failure of the communication link. The reporting of a failure of the link is outside the scope of this guide.

4.3.1.1 Substitution Detection

Once the link has been established and the keep-alive process is operating, the SPT has the option to challenge the authenticity of the SPA that it is communicating with. This is done by periodically, sending a challenge message containing a seed code. The SPA applies a hashing function to the seed using a pre-shared key (known only to the SPT and SPA) and returns the result. The SPT is then able to confirm that the connected SPA is the one that it should be communicating with.

The hashing algorithm used is given below:

```
Unsigned long hash32(unsigned long seed, unsigned long key)
{
seed = (seed ^ 61) ^ (seed >> 16);
seed = seed + (seed << 3);
seed = seed ^ (seed >> 4);
seed = seed * key;
seed = seed ^ (seed >> 15);
return seed;
}
```

Type field	Description	Data field	
0x0007	ID challenge	4 byte seed code	SPT to SPA
0x0008	ID response	4 byte result code	SPA to SPT

4.3.2 Additional channel establishment

Channel 1 is used purely for providing a low level link establishment process and for providing a means of conveying status and control information between the two devices. For the transfer of application data between the two devices the SPA or SPT should bring into service additional channels as required. The transfer of data within the channel is outside the scope of this guide. However, the establishment of the link should follow the same procedure as for channel 1. The direction of some of the messages may be reversed depending on which side is responsible for establishing the link.

It should be possible for either end to request a channel to be opened. The question is how does either end know which channel is to be opened and what is required of the channel?

An SPT will be supplied with support for 1 or more SPA transmission services. The services available will be known and it would therefore be possible to supply the installer with details of them. If a 'standard' SPT is to be provided then the SPA will know what connections to establish. Otherwise it will be necessary for the SPA to be configurable with the details of the channel that it needs to use. This could be as simple as configuring the SPA with a channel number.

Once the initial channel negotiation has been completed, it will then be possible to open additional channels as required.

To open a channel the requesting device will transmit a service request packet (packet type 0x00004) for that channel. The content of the service request data field will be dependent on the function of the channel and is outside the scope of this guide. However, the same procedure as used for the initial channel negotiation should be used for negotiating the setup of the channel.

In the event that there is a clash of service request packets i.e. either end receives a service request when it is expecting an acknowledgement or negative acknowledgement to the service request that it sent, the SPT will have precedence and the SPA should attempt to respond to the service request it has received. If no response is received within the normal timeout periods then either end should try again. Where possible some random function should be applied to the retry time in order to prevent a further clash.

Note: It is not a requirement of this guide for keep-alives to be used in all channels other than the SPA-SPT control channel. Hence, other channels can be completely idle once they have been established but are not transferring data. For example, a channel used for the transfer of event information will only be used when an event needs to be reported.

4.3.3 Definition of further 'standard' channel types

To make this document more useful details of how to operate two basic SPA/SPT connections have been included. The first covers the transmission of event information from some form of alarm panel (I&HAS, Fire, Building maintenance etc) and the second covers alarm panel remote maintenance (upload/download etc).

Note: The term 'alarm panel' in this context is not restricted to I&HAS equipment, but covers any equipment type that has event information that has to be transferred to a monitoring centre and/or has a remote support capability.

4.3.3.1 Event reporting channel

The channel number used is for mutual agreement between the SPT and SPA installation and is likely to be configurable.

The channel establishment phase will be initiated by the SPA. In the service request packet, the SPA will identify what services, if any, it wants of the SPT (currently none are identified but this may be amended later).

The SPA will format event messages using the same message structure as is currently used by the common alarm receivers. This approach allows for different message types to be transferred, with a header provided to indicate the format of the event data. Using the header information the SPT can then determine what, if any, processing of the event data is needed before transmitting the event across the alarm transmission system.

The format of the messaging interface is commonly detailed in guides produced by alarm receiving equipment manufacturers.

An example of a SIA type message and a Contact ID message are given below. Other formats follow similar principles.

SRRL[#AAAAAA|EMMZZZZ/MMZZZZ/MMZZZZ][DC4]

Where,

S : Beginning transmission of the new SIA protocol
RR : Receiver number 00-FE
L : Line number 0-E
[: Beginning data delimiter
: Account ID block code
AAAAAA : Account ID, maximum sixteen digits.
| : Field separator
E : Function block code
MM : Event code or modifier
ZZZZ : Zone code, or user code, or time/date information
/ : Data code packet separator
] : Ending data delimiter
[DC4] : Terminator, 14 Hex

In some implementations additional function blocks are permitted in the same message. In this case each function block is delimited by a '|' (pipe) field separator.

5RRLs18AAAAQXYZGGCCC[DC4]
Where,

5 : Protocol number.
RR : Receiver number.
L : Line number.
s : Space.
18 : Contact-ID format identifier.
AAAA : Four digit account codes.
Q : Qualifier, E=New event or
opening, R=New restore or closing.
P=Previous event
XYZ : Class code and event code.
GG : Group number.
CCC : Zone codes or user ID.
[DC4] : Terminator,14 Hex

For both of the above examples, the receiver number and line number are not applicable but they could be used to convey some other piece of information.

Note: This channel will not be used for passing any configuration information from the SPA to the SPT. For example in some situations it might be desirable for the SPA to configure the SPT with details of the receiving centre it should be reporting to but this will be handled in a separate channel.

Type field	Description	Data field	
0x0007	Event	Event message code SPT to SPA	SPA to SPT4 byte seed
0x0008	Acknowledge	None	SPT to SPA
0x0009	Negative ack	None	SPT to SPA

Events are submitted by the SPA n-times (configurable) with a timeout of x-seconds (configurable).

If the SPA fails to get a response from the SPT after the n-attempts then it will record a failure to communicate. When the SPA has a new event to report it will start the process again, beginning with the first event in the queue.

If the SPA receives a negative ack for each of its n-attempts then it will record a failure to communicate and log that event as not submitted. When the SPA has a new event to report it will start the process again, but starting with the new event on the basis that there may be some corruption of the previous event that is stopping it from being transmitted.

Either entity can close the channel using the channel close command.

4.3.3.2 Alarm panel upload/download channel

The channel number used is for mutual agreement between the SPT and SPA installation and is likely to be configurable.

The channel establishment phase will be initiated by the SPT in response to a remote request for an upload/download (UDL) session to be opened. In the service request packet the SPT will identify what services, if any, it wants of the SPA (currently none are identified but this may be amended later).

Once the channel is established the two entities will simply pass data transparently between the panel and the application.

Type field	Description	Data field	
0x0007	UDL data	UDL data	SPA to SPT
0x0008	UDL data	UDL data	SPT to SPA

It is not thought necessary to implement an ack/nack process at this level as it will normally be the upper layers of the application that will take care of lost packets. In this case if either entity receives a badly formed packet it will simply discard it on the grounds that the SPA or the remote UDL application will re-send or recover the communications process.

Either entity can close the channel using the channel close command.

4.3.4 Channel multiplexing

The procedures defined here imply a statistical multiplexing approach to getting data from different channels across the link. Where a physical link is being used for a number of different channels which have differing transfer characteristics the implementation at both ends must ensure that high use channels do not block the transmission of data of lower throughput, but potentially more important, channels i.e. event reporting while a streaming application is running.

4.3.5 Checksum procedure

The checksum is calculated as the modulo 256 sum of all of the bytes in the block to be protected, exclusive OR'ed with 0xFF.

Acknowledgements

Thanks go to the TC/1 technical committee working group of the Security Equipment Manufacturers Section of the BSIA.