GEC ALSTHOM

T&D

Courier Protocol

for K-Bus, IEC870 and general implementations

R6511 B

GEC ALSTHOM T&D PROTECTION & CONTROL LIMITED, St Leonards Works, Stafford ST17 4LX, England
Tel: 01785 223251 Telex: 36240 Fax: 01785 212232

Preface

This document details the IEC870 communication standard used on GEC ALSTHOM T&D Protection & Control relays in sufficient detail to enable third parties to interface these relays to other systems and to develop other slave devices (relays) which utilise this communication system. It describes how messages are formed and transmitted from one device to another. It does not describe what the messages mean as this is dependent on the application to which IEC870 is put and the particular format of the data that is transmitted.

This guide should be used with the appropriate user guide for the particular application that the IEC870 system will be used with.

The information and illustrations found in this book are not binding. GEC ALSTHOM T&D Protection & Control Ltd. reserve the right to modify products in line with a policy of continuous product improvement. Information in this document is subject to change without notice.

Table of Contents **Page**

Chapter 1: Introduction

The Courier Protocol binds the Courier language to a particular communication network. It defines a set of low-level rules and implementation details to enable many of the automatic features of a Courier based system to be performed on different systems. Some of the lower layers of the protocol may be implemented by the communication network itself and therefore the protocol layers may be network specific.

Courier transactions, Courier commands, message structures and so on are described fully in the Courier User Guide, to which reference should be made.

The Courier Protocol described in this document is oriented on the ISO-OSI model 2 of open systems interconnection, the enhanced performance architecture (EPA).

Layer	Name
7	Application
6	
5	
4	
3	
2	Link
1	Physical

Model 2: Enhanced Performance Architecture (EPA)

Layers 1 and 2 of the model relate to the physical network which can presently be either K-Bus or IEC870-5 ft1.2. Reference should be made to the user guide for the particular communication network being used.

The Courier Protocol equates to layer 7 of the model, the application layer. Layers 3 to 6 are non-existent in the OSI EPA model. To simplify the description of the Courier Protocol, this application layer is further divided into sub-layers, as the following sections will detail.

Chapter 2: Protocol Rules

Courier uses a set of protocol rules to define the low level operation of Courier and how single request and reply messages may be sent over the communication network.

2.1. COURIER PROTOCOL RULES SUMMARY

- There may be multiple slave devices on a communication bus, but only one master can be active at any time.
- Courier uses a 'Master-Slave' or 'Polled' protocol.
- Slave devices cannot initiate a communication transaction.
- No slave devices will reply to a global message broadcast from a master control unit.
- A slave device must receive a global message twice in succession with no other intervening valid message, before it will accept and execute the command.
- A slave device will only send a communication message in response to a request from a master control unit to a unique address.
- A slave device must reply to a request within a response time of 5ms.
- If a slave device cannot generate the correct reply within the response time, it will send an empty BUSY message as a reply, thus allowing the master control unit to use the bus again.
- Slave responses (except blocked transactions) are always of the same type and length for a given request.
- Any communication messages received in error will be completely ignored, as if there was no message at all.
- If a master control unit does not receive a reply to its request within the time-out period, it will retry several times before it determines that the slave device is not communicating.

2.2. COURIER MESSAGE STRUCTURE

The full Courier message structure is described in detail in the Courier User Guide. An overview is given here to give some understanding of the format of the examples in the following sections and in particular to introduce the concept of the Reply Control Field which is handled by the Courier Protocol, although its components are described in the Courier User Guide.

A Courier request message consists of 3 fields: an address field, a length field and a user data field. As far as the application is concerned a response message also consists of the same three fields, but in reality an additional reply control field is inserted by the Courier Protocol in the slave device and handled by the Courier Protocol in the master control unit.

Courier Request Message Format

Address Field	Length Field	User Data Field

Courier Reply Message Format

Address Field	Length Field	Reply Control Field	User Data Field

2.2.1. REPLY CONTROL FIELD

The reply control field is only present in messages transmitted to the master control unit as it indicates various items of status information about the slave device. It is not present in the reply to a Reset Remote Link Command. It consists of one or more Courier data packets terminated with a data packet of type 5Dh, indicating the Courier status byte.

DTL time Tag (38h) + (04h)	4 byte millisecond timer count value	DTL IEC870 time & date	Real-time time-tag	DTL Status 5Dh	Status

↑ This data packet is only required if slave device has no real time clock

↑ This data packet optionally inserted by a protocol converter.

↑ Additional data packets may be inserted here in future.

The first packet is an optional time-tag which is used for the time-alignment of events within different relays. Slave devices which include their own real time clock will omit this data packet since time alignment will not be necessary. Other slave devices will include a data packet with a four byte millisecond timer count value and a DTL of 38h.

Time alignment is only possible if the communication propagation delay time between a slave device and a master control unit is short and consistent. In instances where a protocol converter is used to transfer data over modems or other transmission mediums, time alignment is partially performed at the protocol converter itself. This is done by the protocol converter inserting a real-time time-tag in IEC870 time & date data packet format immediately after the millisecond timer count data packet. Time synchronisation is discussed chapter 4.

The last packet is always present and contains the status information of the slave device. The DTL has a value of 5Dh and is followed by a single status byte. This status packet will ALWAYS be the last packet in the reply control field. Should the reply control field be extended in the future by adding additional packets, this status packet can be used to determine the start of the user data field. This status byte is used to perform automatic data retrieval from the slave device which is discussed in a later chapter, and to inform the master control unit of various changes of state that have occurred.

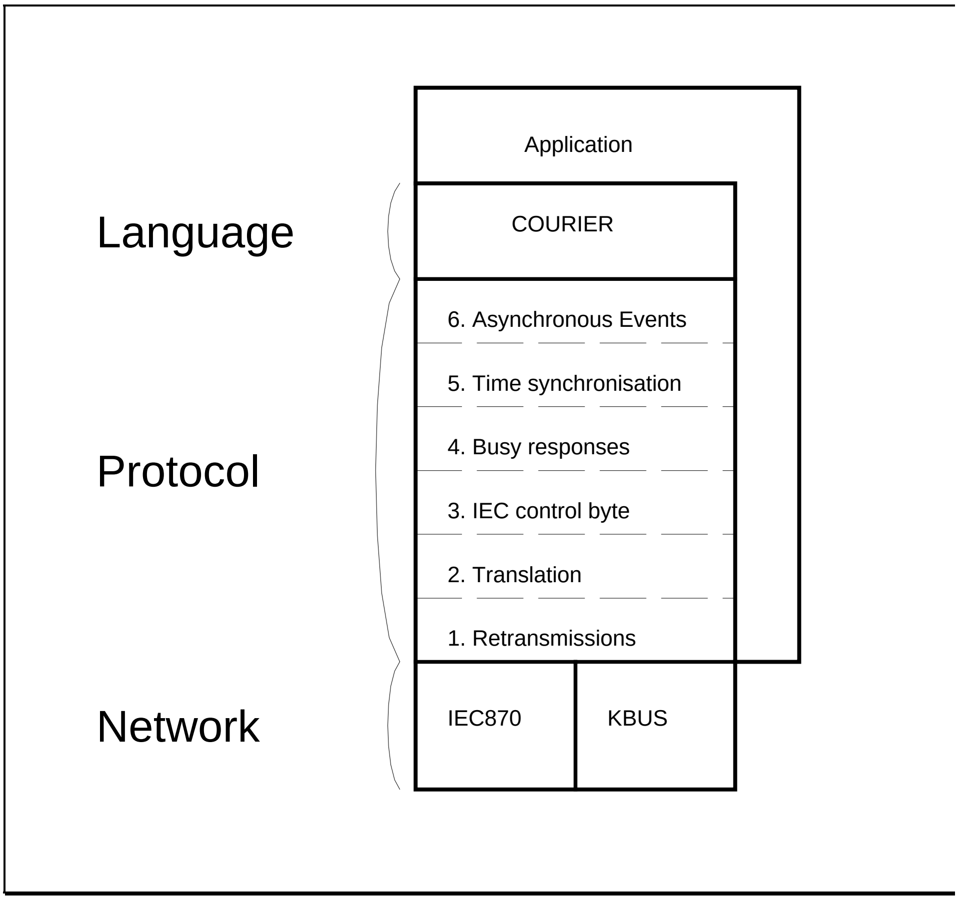
The reply control field is transparent to the application layer. It allows additional information about the slave device to be provided automatically and made directly available to the application, rather than including it in the response to a request message.

The Courier protocol is described in the next two sections, both for the master control unit and the slave device. Each protocol layer is described as a single process, yet having two data paths: one for requests and one for responses, each having separate inputs and outputs. Each layer will process the input request to produce an output request and the input response to produce an output response. In some layers this process may be a straight copy, in others it will be a trivial conversion. In the master control unit an input response from a slave device might generate further output requests and corresponding input responses before the output response is returned to the next higher layer.

Examples of the expected fields to be found in these inputs and outputs are given for each layer.

2.3. COURIER PROTOCOL OVERVIEW

(MASTER CONTROL UNIT)


Application
Language
COURIER
6. Asynchronous Events
5. Time synchronisation
4. Busy responses
Protocol
3. IEC control byte
2. Translation
1. Retransmissions
Network
IEC870
KBUS

The Courier Protocol in the master control unit is divided into 6 distinct layers. An application request formatted into a Courier message is passed into layer 6 of the protocol. This will use the lower layers of the protocol to transmit the message and receive the response, which will be returned from this layer back to the application. In the process of obtaining the response from the slave device, other information may be extracted from the slave device such as: any asynchronous events that may have occurred; changes in value to the plant status, plant control words and changes to other flags in the Courier status byte; time synchronisation information and information relating to the communication status of the slave device. These are passed back to the application via other means rather than included with the response to the original request; exactly how is determined by the application.

The protocol layers are described separately below.

2.3.1. LAYER 6: ASYNCHRONOUS EVENT LAYER.

Description

The top layer of the Courier protocol is the interface between the Courier transactions of the application and the communication network. It is responsible for returning a valid response to a request message. If one cannot be obtained, an error condition is returned. This layer also examines the status byte of the response message to determine if any asynchronous information exists in the slave device and issues the relevant Courier commands to extract this information. The flags examined are the Plant status flag, the Control status flag and the Event Flag. Any changes of state to the Trip, Alarm, Out of Service and Record flags are recorded for use by the application.

Disturbance records cannot be directly extracted easily at this layer since they require blocked transactions which may already be in progress (blocked transactions cannot be nested). Therefore this facility should be made part of the application. However, these can be extracted at this layer if it is possible to keep track of whether the application is in the middle of a blocked transaction and then only extract the records when this is not the case.

Input Request

AA..AA	0	Len_{c1}	Courier Request

Output Request

AA..AA	0	Len_{c1}	Courier Request

Input Response

AA..AA	0	Len_{c2}	Status	Courier Reply

Output Response

AA..AA	0	Len_{c3}	Courier Reply

Note that Len_{c2} differs from Len_{c3} by the length of the status field.

Direct Output

Event Records,
Status byte changes

2.3.2. LAYER 5: TIME SYNCHRONISATION LAYER.

Description

This layer has the responsibility for synchronising all time tagged events from different slave devices. It has no effect on requests. The time tag information is extracted from the reply message, converted to a time synchronised real time value by using the time stamp from when the message was received in layer 1, and stored for use by the application.

Input Request

AA..AA	0	Len_{c1}	Courier Request

Output Request

AA..AA	0	Len_{c1}	Courier Request

Input Response

AA..AA	0	Len_{c2}	Time	Status	Courier Reply

+ Time stamp information from layer 1.

Output Response

AA..AA	0	Len_{c3}	Status	Courier Reply

Note that Len_{c2} does not equal Len_{c3}

Direct Output

Time synchronisation information.

2.3.3. Layer 4: Busy Response Layer.

Description

This layer is responsible for returning a valid response to a request. A slave device must return a response within a set response time. If it cannot form its reply within this time, it will return a message with an empty courier reply field and the Busy flag set in the status byte. On detection of a set Busy flag, this layer will send the Poll Buffer command to the slave device. When a valid response is returned with the Busy flag clear, the response will be passed to the next higher level.

Input Request

AA..AA	0	Len_C	Courier Request

Output Request

AA..AA	0	Len_C	Courier Request

OR

AA..AA	0	Len_C	Poll Buffer

Input Response

AA..AA	0	Len_C	Time	Status (busy set)

OR

AA..AA	0	Len_C	Time	Status	Courier Reply

Output Response

AA..AA	0	Len_C	Time	Status	Courier Reply

Direct Output

None

Busy Timeout

A facility to break out of this layer and return an error code should be provided for the condition of a slave device continuously returning a Busy response. This may take the form of a maximum number of busy responses or a time limit for returning a non-busy response. This is to prevent a lock-up situation from occurring in the master control unit.

Optimization

If a device returns a busy response, continual issuing of the Poll Buffer command will slow down the overall response time since the device will be spending most of its time responding to the Poll Buffer command instead of processing the original request. To fully utilize the communication bandwidth, it is advantageous to interleave requests to other devices during a device's busy period. This complicates the communication routines as it means queuing messages for each device separately and working in a parallel manner, but results in a dramatic increase in overall communication throughput.

2.3.4. LAYER 3: IEC CONTROL LAYER.

Description

This layer is responsible for the logical connection between a master control unit and a slave device and for inserting and removing the IEC control byte.

Before communication with a slave device can take place, the master control unit must establish that the slave device is able to communicate. It does this by issuing the Reset Remote Link command using the IEC control byte and receiving a confirmation. When the confirmation is received, a logical connection exists until the slave device stops responding to request messages. If this condition arises, the logical connection is broken and must be re-established before further communication takes place. Information about the communication status of the slave device is made available to the application.

Request Messages

For request messages, the format of the IEC control byte is as follows:

7	6	5	4	3	2	1	0
RES	PRM	FCB	FCV	Function			

The RES bit is reserved and is set to 0.

The PRM bit indicates the direction of the message and should be set to 1.

For the Reset Remote Link command, the remaining bits are zero resulting in an IEC control byte of 40h. If any of the address bytes are FFh, the request is a global request. For these requests, the value 44h should be used. All other requests use a function code of 11 (0Bh) with the Frame Count Valid (FCV) bit set to 1 indicating that the Frame Count Bit (FCB) is valid. The FCB bit alternates for each new request sent to a slave device to prevent duplications of messages being accepted. The IEC control byte will therefore alternate between the values 5Bh and 7Bh. The first message sent after the Reset Remote Link command will be expected to have the FCB bit set (7Bh). This layer keeps track of the correct FCB bit state for each slave device address.

Response Messages

For response messages, the format of the IEC control byte is as follows:

7	6	5	4	3	2	1	0
RES	PRM	ACD	DFC	Function			

The RES bit is reserved and is set to 0.

The PRM bit indicates the direction of the message and should be set to 0. The ACD and DFC bits are not used for the Courier Protocol and should be set to zero. The confirmation to the Reset Remote Link command uses a function code of 0. All other responses will have a function code of 08h, the response to a Request-Respond function of 0Bh.

Input Request

AA..AA	0	Len_{c1}	Courier Request

Output Request

AA..AA	0	Len_{c2}	61h	IEC	Courier Request

Note that Lenc1 does not equal Lenc2

Input Response

AA..AA	0	Len_{c3}	61h	IEC	Time	Status	Courier Reply

Output Response

AA..AA	0	Len_{c4}	Time	Status	Courier Reply

Note that Lenc3 does not equal Lenc4

Direct Output

Change in communication status of slave devices.

2.3.5. LAYER 2: TRANSLATION LAYER.

Description

The translation layer rearranges the order and adjusts some of the various message fields for correct transmission over the network. The translation layer will typically be different for each type of network used.

Input Request

AA..AA	0	Len_C	61h	IEC	Courier Request

Output Request

IEC870 format:

Len_i

IEC	AA..AA	0	Courier Request	

KBUS format:

AA..AA	0	Len_C	61h	IEC	Courier Request

Input Response

IEC870 format:

Len_i

IEC	AA..AA	0	Time	Status	Courier Reply

KBUS format:

AA..AA	0	Len_C	61h	IEC	Time	Status	Courier Reply

Output Response

AA..AA	0	Len_C	61h	IEC	Time	Status	Courier Reply

Direct Output

None

Notes

The translation layer is non-existent for a KBUS network since the message format is the same. However, for an IEC870 network, the IEC control byte and the length byte move position. Note also that the length byte is stored in the IEC870 header and must be adjusted to include the length of the address field and the removed byte of 61h.

2.3.6. LAYER 1: RETRANSMISSION LAYER.

Description

The retransmission layer is responsible for obtaining a response from the slave device of the correct format. After issuing a request message, it will wait for a timeout period for a response. If no response is returned within this time, the request will be repeated for a number of retransmissions. If still no response is returned, an error condition is passed back to the next layer.

The time at which each valid response is received should be recorded and made available to higher layers of the protocol (layer 5) and the application for time synchronisation purposes.

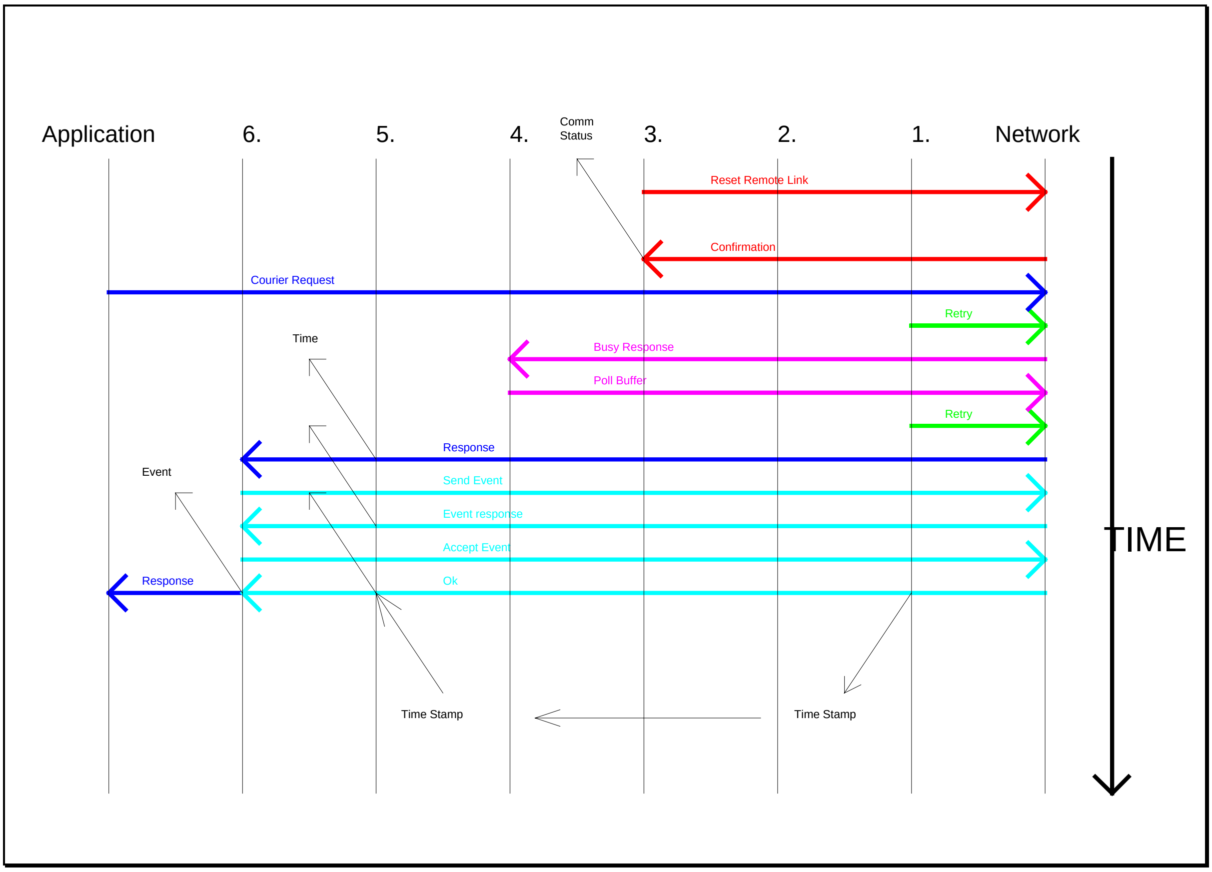
Time-out Period

The timeout period needs to be determined for each slave device address to account for the different propagation and transmission delays due to the network topology. The slave device will have a fixed maximum response time of 5ms.

Number of Retries

<u>Request-Respond messages</u>
The number of retries for normal Request-Respond messages should be set to 9.
<u>Send-Confirm messages</u>
These are only used for the Reset Remote Link command and should have no retries.
<u>Send-No Reply messages</u>
Global messages will have no response. However, they must be received twice in succession before being actioned by a slave device. The number of retries should be set to 2 to force 3 global messages to be sent, thus allowing one of them to be received in error. Care should be used when sending the same global command twice in succession to this layer as this would issue 6 consecutive global messages the same, which may be actioned 3 times. Action should be taken to avoid this if necessary (possibly by sending a Poll Status command to each address).

The flow of messages and information between the master control unit Courier protocol layers can be illustrated in the following diagram:



Application
6.
5.
4.
Comm Status
3.
2.
1.
Network
Reset Remote Link
Confirmation
Courier Request
Retry
Time
Busy Response
Poll Buffer
Retry
Response
Event
Send Event
Event response
Accept Event
TIME
Response
Ok
Time Stamp
Time Stamp

A logical connection is made to the slave device by layer 3 issuing a Reset Remote Link command. Layer 3 identifies that the slave device is present at that address and makes this communication status information available to the application.

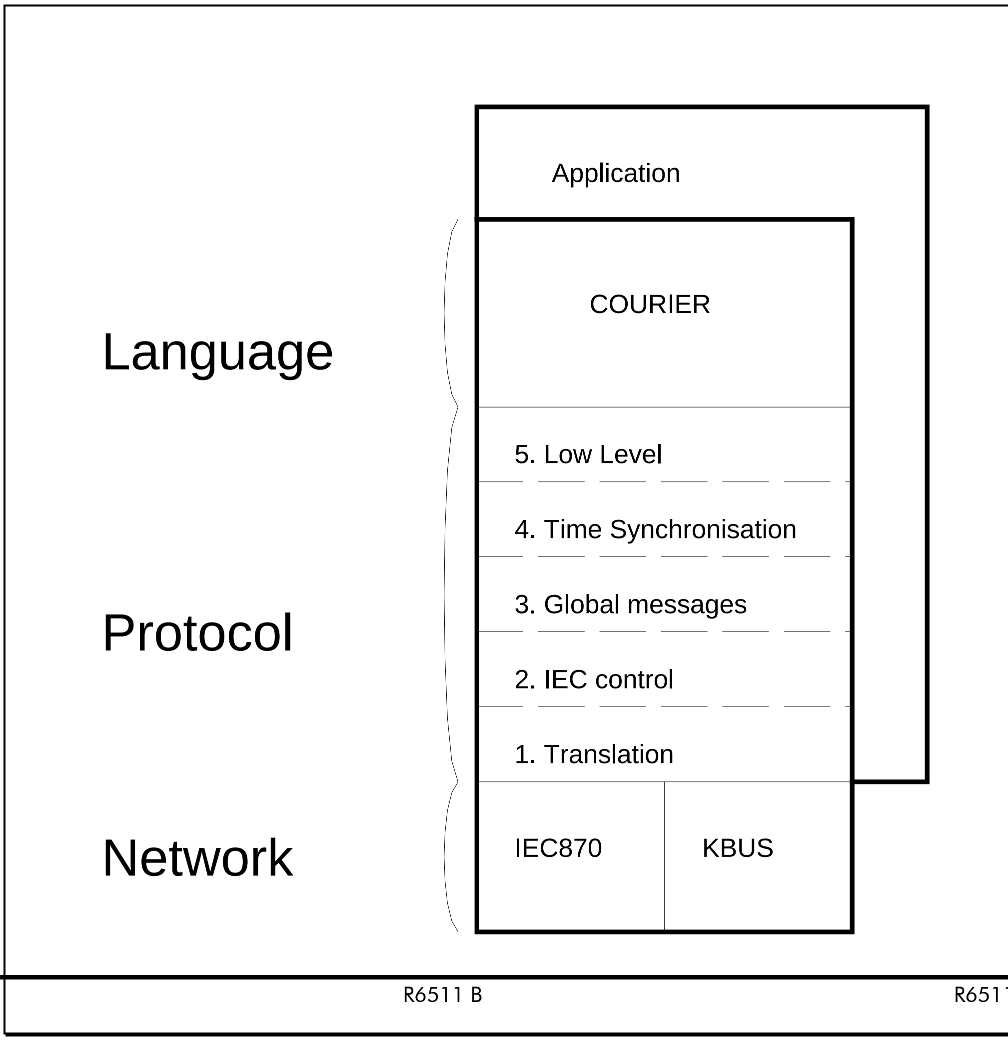
A Courier request is then made by the application. Layer 1 retries the request and obtains a response. Layer 4 detects that this is a busy response and issues a Poll Buffer request, which again Layer 1 has to retry before it receives a response.

The response is a valid Courier response to the initial request and layer 5 performs any time synchronisation required. Layer 6 detects that an event has taken place in the slave device. It therefore issues a Send Event Command. The slave device returns the event straight away and again Layer 5 performs any time synchronisation required. Layer 6 then issues the Accept Event command and the slave device returns OK. The event is stored for use by the application

Layer 6 finally returns the slave device's response to the initial application request.

2.4. COURIER PROTOCOL OVERVIEW

(SLAVE DEVICE)


Application
Language
COURIER
5. Low Level
4. Time Synchronisation
3. Global messages
Protocol
2. IEC control
1. Translation
Network
IEC870
KBUS

The Courier Protocol in the slave device is divided into 5 distinct layers and works in the reverse manner to the protocol layers in the master control unit. A request message is received by layer 1 of the protocol and successively passed up through the layers to the application. This may then respond by returning a Courier response message to layer 5 which is then passed back through the layers to the communication network.

Some of the lower layers (1-3) may be performed by the network itself but are included here for those networks which do not inherently perform these functions.

The protocol layers are described separately below in the reverse order to the layers in the master control unit so that the flow of messages can be more easily followed.

2.4.1. Layer 1: Translation Layer.

Description

The translation layer rearranges the order and adjusts some of the various message fields for correct transmission over the network. The translation layer will typically be different for each type of network used.

Input Request

IEC870 format:

Len_i	IEC	AA..AA	0	Courier Request

KBUS format:

AA..AA	0	Len_c	61h	IEC	Courier Request

Output Request

AA..AA	0	Len_c	61h	IEC	Courier Request

Input Response

AA..AA	0	Len_c	61h	IEC	Time	Status	Courier Reply

Output Response

IEC870 format:

Len_i	IEC	AA..AA	0	Time	Status	Courier Reply

KBUS format:

AA..AA	0	Len_c	61h	IEC	Time	Status	Courier Reply

Direct Output

None

Notes

The translation layer is non-existent for a KBUS network since the message format is the same. However, for an IEC870 network, the IEC control byte and the length byte move position. Note also that the length byte is stored in the IEC870 header and must be adjusted to include the length of the address field and the removed byte of 61h.

2.4.2. LAYER 2: IEC CONTROL LAYER.

Description

This layer is responsible for the logical connection between a master control unit and a slave device and for inserting and removing the IEC control byte.

Before communication with a slave device can take place, the master control unit must establish that the slave device is able to communicate. It does this by issuing the Reset Remote Link command using the IEC control byte and receiving a confirmation. This layer is responsible for returning the confirmation to this command. This layer also inserts the correct IEC control byte in response messages and removes it from request messages.

Request Messages

For request messages, the format of the IEC control byte is as follows:

7	6	5	4	3	2	1	0
RES	PRM	FCB	FCV	Function			

The RES bit is reserved and is set to 0. It should be accepted in either state.

The PRM bit indicates the direction of the message and will be set to 1.

For the Reset Remote Link command, the remaining bits are zero resulting in an IEC control byte of 40h. If any of the address bytes are FFh, the request is a global request. For these requests, the value 44h will be used and no response will be returned. All other requests use a function code of 11 (0Bh) with the Frame Count Valid (FCV) bit set to 1 indicating that the Frame Count Bit (FCB) is valid. The FCB bit alternates for each new request sent to a slave device to prevent duplications of messages being accepted. The IEC control byte will therefore alternate between the values 5Bh and 7Bh. The first message received after the Reset Remote Link command with the FCV bit set will be expected to have the FCB bit set (7Bh).

This byte is handled by the IEC870 network which will discard messages with the FCB bit in the wrong state, requiring a Reset Remote Link command to be sent before further communication can proceed.
This value is ignored when using KBUS networks since Courier does not depend on its value as commands are designed such that duplications have no detrimental effect. Global messages can be detected by an address of FFh. The Reset Remote Link command must be supported, however.

Response Messages
For response messages, the format of the IEC control byte is as follows:

7	6	5	4	3 2 1 0
RES	PRM	ACD	DFC	Function

The RES bit is reserved and is set to 0.
The PRM bit indicates the direction of the message and should be set to 0. The ACD and DFC bits are not used for the Courier Protocol and should be set to zero. The confirmation to the Reset Remote Link command uses a function code of 0. All other responses will have a function code of 08h, the response to a Request-Respond function of 0Bh.

Input Request

AA..AA	0	Len_{c1}	61h	IEC	Courier Request

Output Request

AA..AA	0	Len_{c2}	Courier Request

Note that Len_{c1} does not equal Len_{c2}

Input Response

AA..AA	0	Len_{c3}	Time	Status	Courier Reply

Output Response

AA..AA	0	Len_{c4}	61h	IEC	Time	Status	Courier Reply

Note that Len_{c3} does not equal Len_{c4}

Direct Output

None.

2.4.3. Layer 3: Global Message Layer.

Description

The global message layer is responsible for receiving two successive identical valid global requests before it passes one of them up to layer 4 to be further processed. Non-global requests are passed straight through. All responses are passed straight through this layer.

2.4.4. Layer 4: Time Synchronisation

Description

This layer is responsible for inserting the current value of the slave device's free running millisecond clock for slave devices which require time synchronisation. The time inserted should be the value of the clock when the first byte of the message is transmitted.

Input Request

AA..AA	0	Len_{c1}	Courier Request

Output Request

AA..AA	0	Len_{c1}	Courier Request

Input Response

AA..AA	0	Len_{c2}	Status	Courier Reply

Output Response

AA..AA	0	Len_{c3}	Time	Status	Courier Reply

Note that Len_{c2} does not equal Len_{c3}

Direct Output

None

2.4.5. LAYER 5: LOW LEVEL

Description

This layer is responsible for performing low level Courier commands such as the Poll Status and Poll Buffer commands. Depending on application it may also perform the event extraction routines.

In response to the Poll Status command it should return an empty message with the current value of the communication status byte. This request should not ordinarily result in a busy response unless the device is already busy as a result of a previous operation. Doing so compromises the effectiveness of the communication system as a whole.

The Poll Buffer command should retransmit the last response frame if the BUSY flag is clear. If the BUSY flag is set, it will simply return the communication status byte; only the BUSY flag need be valid, the other flags are indeterminate.

Input Request

AA..AA	0	Len_C	Courier Request

OR

AA..AA	0	Len_C	Poll Buffer

OR

AA..AA	0	Len_C	Poll Status

Output Request

AA..AA	0	Len_C	Courier Request

Input Response

AA..AA	0	Len_C	Courier Reply

Output Response

AA..AA	0	Len_C	Status (busy set)

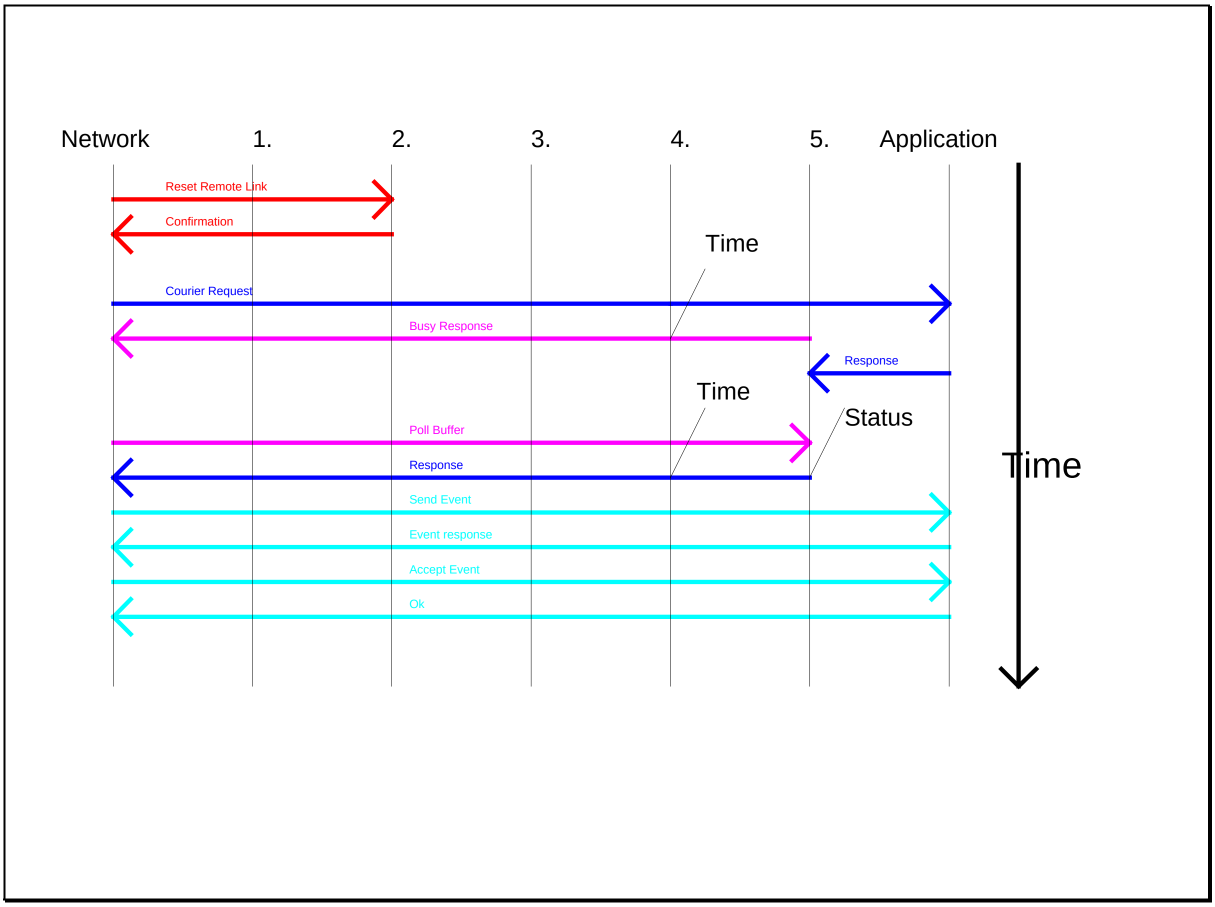
OR

AA..AA	0	Len_C	Status	Courier Reply

OR

AA..AA	0	Len_C	Status

The flow of messages and information between the slave device Courier protocol layers can be illustrated in the following diagram:


Network
1.
2.
3.
4.
5.
Application
Reset Remote Link
Confirmation
Time
Courier Request
Busy Response
Response
Time
Status
Poll Buffer
Response
Time
Send Event
Event response
Accept Event
Ok

A reset remote link command is received which Layer 2 intercepts and returns a confirmation.

A Courier request is then received which is passed to the application layer. Layer 5 does not receive a response from the application within the 5ms time out period and so it returns a busy response to the network.

The busy response prompts a Poll Buffer command which is handled by layer 5. By this time the application has returned a response, so this is returned as the reply to the Poll Buffer command with the current status byte (otherwise another Busy response would have been returned).

Because the event bit was set in the previous response, the master control unit then issues a send event command. The application returns the event response. This is acknowledged with an accept command which clears the event flag (as there are no more events) and the application returns an OK reply code.

In all responses, the status byte is added at layer 6, the time is added at layer 5 and the IEC control byte is added at layer 2.

2.5. IEC CONTROL FIELD

The IEC870 Control field consists of 2 bytes. The first is 61h to identify the field whilst the second contains information that characterises the direction of the message, the type of service provided and supports control functions for suppressing the losses or duplication of messages. This is described as the IEC870 control byte in the IEC870-5-2 5.1.2 specification.

There are three service classes which are described in the following table:

Link Service Class	Function	Explanation
S1	SEND/NO REPLY	Transmit message; neither acknowledgement nor answer is requested within the link layer.
S2	SEND/CONFIRM	Transmit message; acknowledgement is requested within the link layer.
S3	REQUEST/RESPOND	Transmit message; a response is requested within the link layer; the response will contain data.

Link service classes.

The format of the IEC870 control byte is as follows:

7	6	5	4	3	2	1	0	
RES	PRM=1	FCB	FCV	Function				Master---->Slave
	=0	ACD	DFC					Slave---->Master

IEC870 control byte format.

<u>RES: Reserved.</u>
This bit is reserved. It should be transmitted as a 0, but should be accepted in either a 0 or a 1 state.

<u>PRM: Primary message.</u>
Indicates the direction of the message, 1 for master to slave messages, 0 for slave to master messages.

<u>FCB: Frame Count Bit.</u>
An alternating bit 0/1 for successive SEND/CONFIRM or REQUEST/RESPOND services per slave device. The Frame Count bit is used for suppressing the duplications of information transfers: The master control unit alternates the FCB bit for each new SEND/CONFIRM or REQUEST/RESPOND transmission service directed to the same slave device. Thus the master control unit keeps a copy of the Frame Count Bit per slave device. If an expected reply is timed-out (missing) or garbled, then the same SEND/CONFIRM or REQUEST/RESPOND service is repeated with the same FCB state.

The FCB bit is always zero for reset commands. After a reset command, the slave device will expect the next message received from the master control unit with the FCV bit set to have the FCB bit set also.

FCV: Frame Count Valid.
0 = alternating function of FCB bit is invalid.
1 = alternating function of FCB bit is valid.
SEND/NO REPLY service, broadcast messages and other transmission services that ignore the suppression of duplication or loss of information output do not alternate the FCB bit and indicate this by a cleared FCV bit.

DFC: Data Flow Control.
This bit is reserved and should be set to 0.

ACD: Access Demand.
This bit is reserved and should be set to 0.

Functions.
The only functions that are used in the K-Bus environment are shown below:

Function	Frame Type	Service Function	Use	FCV
0	Send-Confirm (class S2)	Reset Remote Link	Resetting and identifying slave devices.	0
4	Send-No reply (class S1)	User Data	Global Messages	0
11	Request-Respond (class S3)	Request User Data Class 2	Normal K-Bus Request	1

IEC870 function codes for request frames.

Function	Frame Type	Service Function	Use
0	Confirm (class S2)	ACK: Positive acknowledgement of Reset Remote Link	Indicates slave device is present.
8	Respond (class S3)	User Data	Normal K-Bus Response

IEC870 function codes for response frames.

Chapter 3: Automatic Data Retrieval

Courier is designed to operate using a polled system, which prevents a slave device from communicating directly to a master control unit when it needs to inform it that something has happened; it must wait until it is asked by the master control unit. If the master control unit continually asks a slave device for lots of information just in case something has happened, the communication system will be inefficient.

The status byte of a Courier message allows the slave device to inform the master control unit that it contains additional information to be extracted and to indicate changes of state of important data.

3.1. EVENT RECORDS

The Event record is a mechanism that has been designed to allow a master control unit to extract this type of unsolicited information from a slave device quickly and efficiently.

An event record is designed to be a short, concise description of an event which has occurred in a slave device which the master control unit should know about. It includes such occurrences as local setting changes, logic input changes, relay output changes and trips. When such an event occurs, the slave device prepares an appropriate message to be transmitted and sets the EVENT bit in the Courier status byte which is transmitted in each and every reply to the master control unit.

The extraction of an event record is considered a high priority task by a master control unit which it will perform as soon as it recognises the event bit of the slave device being set and the slave device has returned a valid response to the last request and not a busy message. This may occur at any time. The extraction of an event is a grouped transaction. It cannot be a blocked transaction because a blocked transaction may already be in progress and blocked transactions cannot be nested. It therefore follows that a complete event record must fit into one response message.

All event groups must begin with a standard packet format as outlined below:

Packet#	DTL	Use
1	46h	Menu cell reference
2	38h/3Ch	Time tag
3	18h+LL	ASCII Text description
4	xx	Menu cell value

This is described more fully in the Courier User Guide.

Once the event bit is seen as being set, the master control unit issues a Send Event Command. The slave device may respond with a busy message, an error code or a grouped event record. An error code 2 may occur if there are events waiting to be extracted but the oldest event is not currently accessible. The Send Event command should be re-issued until a valid event record is extracted. The Poll Buffer command is issued if a busy response is returned.

After a valid event has been extracted, the Send Event command would continue extracting the same event unless the slave device is informed that the master control unit has received the last event record. This is done by the master control unit issuing the Accept Event command once a valid event record has been extracted. On receipt of this command, the slave device will discard the last event that it sent and determine if any more events are stored. If there are no more events to be extracted, the slave device will reset its event bit. If there are still some events left, the event bit remains set. The slave device then returns a reply code of 0 (ERR_OK) to indicate that it has discarded the last event.

Courier specifies an automatic retry of commands which have not resulted in a reply within a specified time-out period. This may result in repeated Accept Event or Send Event commands if the replies are not received correctly. To prevent these repetitions from discarding events which the master control unit has not received correctly, these commands must be interlaced with each other. Repeated Accept Event commands will therefore only discard one event. Repeated Send Event commands will result in the same event being transmitted.

The actual implementation of event records in the slave device is application dependent. The extraction procedure outlined above assumes that event records are lost on loss of power to the slave device and that once discarded they may not be retrieved again. This in fact may not be the case as some slave devices may decide to store all or some of their event records in non-volatile memory. In this instance, additional commands may be sent to the slave device (e.g. changing the value of a setting cell) to make previously discarded event records available again.

3.2. PLANT STATUS WORD

The plant status word is a pre-defined binary word containing information about the state of various pre-defined items of plant which may be controlled by the slave device. It can be found at a reserved location in the system data column of the slave device's database. Each item of plant uses 2 bits in this word, the meaning of which is shown below:

High bit	Low bit	Meaning
0	0	Not fitted
0	1	Open
1	0	Closed
1	1	Not fitted

On average there are three controllable isolators per circuit breaker, so the bit pairs are arranged as 1 circuit breaker and 3 isolators per byte as follows:

```
bit 31                                bit 0
---- ---- ---- ---- ---- ---- ---- --XX  Circuit Breaker 1
---- ---- ---- ---- ---- ---- ---- XX--  Isolator 1
---- ---- ---- ---- ---- ---- --XX ----  Isolator 2
---- ---- ---- ---- ---- ---- XX-- ----  Isolator 3
---- ---- ---- ---- ---- --XX ---- ----  Circuit Breaker 2
---- ---- ---- ---- ---- XX-- ---- ----  Isolator 4
---- ---- ---- ---- --XX ---- ---- ----  Isolator 5
---- ---- ---- ---- XX-- ---- ---- ----  Isolator 6
---- ---- ---- --XX ---- ---- ---- ----  Circuit Breaker 3
---- ---- ---- XX-- ---- ---- ---- ----  Isolator 7
---- ---- --XX ---- ---- ---- ---- ----  Isolator 8
---- ---- XX-- ---- ---- ---- ---- ----  Isolator 9
---- --XX ---- ---- ---- ---- ---- ----  Circuit Breaker 4
---- XX-- ---- ---- ---- ---- ---- ----  Isolator 10
--XX ---- ---- ---- ---- ---- ---- ----  Isolator 11
XX-- ---- ---- ---- ---- ---- ---- ----  Isolator 12
```

The length of this word is variable depending on the number of circuit breakers and is identified when the data is requested. However, 2 bytes will usually suffice.

3.3. CONTROL STATUS WORD

The control status word is a binary word used to interchange control information with the master control unit and can be found at a reserved location in the system data column of the slave device's database. There are no specific definitions for the uses of the bits within this word and each slave device is free to define them for its own uses. The master control unit would need to be programmable to perform different actions depending on the state of these bits for each slave device or type of slave device.

3.4. TRIP INDICATIONS

The trip flag is used to mimic a trip indication on the front of the slave device for annunciation purposes. It cannot be used to extract fault information from a slave device because it cannot be guaranteed to be set for a long enough period in order to be identified. Also it is not automatically reset, and therefore may cause the same fault record to be read twice (or some to be missed if the master control unit does not see it clear and set again for a second trip). Fault information is sent using event records.

3.5. ALARM INDICATIONS

The alarm flag is used to mimic an alarm indication on the front of the slave device for annunciation purposes. It cannot be used to extract alarm information from a slave device because it cannot be guaranteed to be set for a long enough period in order to be identified. Also it is not automatically reset, and therefore may cause the same alarm to be read twice (or some to be missed if the master control unit does not see it clear and set again for a second alarm). The alarm indication may flash on and off causing the alarm flag to set and clear repeatedly. Alarm information should be sent via the event record mechanism.

Chapter 4: Time Synchronisation

One of the important functions of existing SCADA systems is sequence of event recording. This gives the system engineer valuable insight into the order in which events on the system occur. The typical accuracy of existing systems ranges from ±10ms to ±1ms. This function is currently carried out by the RTU's of the SCADA system which monitor system events using digital inputs. These events are generally time tagged using a system synchronising pulse distributed around the communication network. Often of more importance than the absolute time of an event is its time relative to other events across the system.

Two possible methods of time tagging are provided in the Courier Protocol. The first allows each slave device to have its own real time clock synchronised to the rest of the system. Time tagged events are logged by recording this time in IEC time & date format and transmitting this information with the event. However, using multiple real time clocks can be expensive and requires them all to be synchronised externally.

An alternative method of time synchronisation has been devised which removes the need for separate clock synchronisation wiring. Rather than trying to synchronise the clocks within each individual slave device, the clocks are allowed to free run. Events within each slave device are time tagged relative to the internal free running clock. This is derived from the microprocessors clock and resolutions of ±1ms are easily obtained.

Clearly when these event records are transmitted to the master control unit, events from different slave devices will be out of step. This problem is solved by also transmitting the current value of the slave device's clock. This is compared with the master control unit's clock and the difference used to calculate the actual time of the event.

Consider a system with a master control unit whose clock T_c is synchronised to real time. A slave device on the system has a clock T_r which is not synchronised. At any particular instance in time we can say that:

$$T_c = T_r + \Delta t$$

where Δt is the difference between the two timers. Thus if an event is seen by the slave device at time T_r1, then the synchronised time at which it actually occurred T_c1 is given by:

$$T_c1 = T_r1 + \Delta t1 \qquad (1)$$

where $\Delta t1$ is the difference between the timers at the instance of the fault. At some time later, T_r2 (equivalent to $T_c2-\Delta t2$) the slave device transmits the time of the event (T_r1) together with its current time T_r2. This is received by the master control unit at time $T_c2 + t_p$, where t_p is the propagation delay down the communications channel. At this point we can say:

$$T_c2 = T_r2 + \Delta t2 - t_p \qquad (2)$$

The time t_p can be calculated from a knowledge of the message length and the baud rate, however, using a KBUS based network it is insignificant compared with the 1ms accuracy of T_c and T_r. The difference between $\Delta t1$ and $\Delta t2$ is the rate of drift between the two clocks. Both of these clocks are derived from quartz crystals or better and hence even over several hours we can say:

$$\Delta t1 = \Delta t2$$

Equation 2 can therefore be simplified and rearranged to give:

$$\Delta t1 = \Delta t2 = T_c2 - T_r2$$

This knowledge of $\Delta t1$ is then used in equation 1 to calculate the actual time of the event T_c1:

$$T_c1 = T_r1 + T_c2 - T_r2$$

This time synchronisation method can only be used on systems where the speed of transmission is fast compared to the timing resolution required and where the transmission times are consistent (such as KBUS) such that t_p can be eliminated. On an IEC870 based system, this cannot be guaranteed and therefore the slave devices would be better having their own real time clocks.

Where a KBUS network is interfaced directly to an IEC870 network, the point of inter-connection will be a protocol converter. This would require its own real time clock to time stamp KBUS messages as they are received. This time is inserted into the message after the slave device's own clock value as an IEC time & date formatted packet. The time synchronisation at the master control unit will use the protocol converter's time rather than its own in order to align all time tagged events. The protocol converter's real time clock will need external synchronisation with all other real time clocks in the system.

DTL time Tag (38h)+(04h)	4 byte millisecond timer count value

↑ This data packet is only required if slave device has no real time clock

DTL IEC870 time & date	Real-time time-tag

↑ This data packet optionally inserted by a protocol converter.

The time synchronisation process generally produces a real time value adjusted by an offset for each slave device, which when added to a time tag in the Courier response message, converts the relative time to a time synchronised real time value. This offset real time value is calculated in layers 1 and 5 of the master control unit and passed to the application by a separate path so that it can be used when required.

AMENDMENTS

Issue	Date	Changes
A	06/05/93	First Issue
B	05/04/95	Changed fonts
	12/04/95	CB position reversed.