



Applied Systems Engineering, Inc.

Technical Note #33 Switched COMM circuits

Technical Note 33: Switched COMM circuits

Switched COMM Circuits	3
GPT Properties	3
COMM Properties	3
COMM APIs	4
COMM Processing	4

Switched COMM Circuits

This note discusses how the COMM layer manages serial connections that are switched (e.g. dial-up circuits). Generally most circuits are dedicated lines. Once the connection between the local device and the remote device is opened the connection is always available. It is possible to create non-dedicated circuits between the local station and the remote station. This switched link can be opened and closed dynamically by either the local or remote station. Switched serial links are handled in a similar way to the way the GPT handles networked connections. The main point to understand is that the user manages the circuit switching in the physical layer the COMM. The operation of the GPT is independent of whether the physical link is serial (switched or dedicated) or networked.

GPT Properties

Several properties are defined to support switched serial links. At a high level the GPT property `IPROP_GPT_SWITCHED` is used to indicate whether the serial link is a switched or dedicated link. This property if `TRUE` indicates that the physical circuit is a switched circuit. The main use of the property is in the configuration of the COMM in `DACCFG.C`. The GPT does not use this property at run-time.

COMM Properties

The following COMM properties are used to support switched serial circuits.

- **IOC_TRANSLATION.** This property describes the translation mode the COMM is operating in either RTU (`GPT_RTU`) or Master (`GPT_MASTER`). The translation mode can be used by the COMM to decide which station initiates the connection on the switched circuit. The station running as Master typically does this, although an RTU running in balanced mode could initiate a connection to a remote device.
- **IOC_SWITCHED.** This property is true if the connection on the COMM circuit can be connected and disconnected dynamically.
- **IOC_ROUTE.** This property is used to describe how a connection between the local station and a remote station can be established. The value of this property is a pointer to a structure `LPCOMROUTE`. This structure is defined as follows:

```
typedef struct {  
  
    GPTDWORD    Destination;  
    GPTBYTE     Type;  
    GPTBYTE     Path[COM_MAX_PATH];  
  
} COMROUTE, *LPCOMROUTE;
```

Where:

Destination: Identifies a path between the remote and local device. This is typically the link address of the remote device.

Type: Contains any additional user supported information to describe the connection.

Technical Note 33: Switched COMM circuits

Path: The path used to connect to the remote device. This path can be a network IP address or a phone number.

When running as master the GPT will configure a route that connects the local device (Master) with every remote RTU.

COMM APIs

If `IOC_SWITCHED = TRUE`, the user must also provide additional services with the COMM. The services are local to the COMM and are not accessible by the GPT. These services include:

- **Accept.** Accept a connection request from a remote station. This might include answering an incoming phone call to establish a connection using a modem.
- **Disconnect.** This service allows the local station to disconnect the circuit and re-use it to establish a connection with other remote devices.
- **Connect.** This service allows a connection to be established with a remote device. This could involve dialing out on a phone line used a phone number configured with the `IOC_ROUTE` property.

COMM Processing

With the above services in place this section describes how they are used in COMM to establish a connection between two stations.

CommOpen

CommOpen initializes all properties of COMM. Any additional resources required when the circuit is switched can be allocated here.

CommClose

Release all resources allocated in CommOpen.

SetComm

SetComm processes the `IOC_ROUTE` property. The GPT test software uses the `IOC_ROUTE` property to configure paths used to connect to remote stations. The user can replace this property with any other COMM configuration that provides equivalent functionality.

WaitComm

WaitComm is called periodically by the GPT to read physical frames from the physical circuit. If the circuit is switched WaitComm must use the Accept COMM service to accept connection requests from a remote device. Conditions which require the accept are:

- **Data Server.** The GPT is running as a data server (RTU). The COMM must accept connection requests from the remote master before any physical frames can be read from the circuit.
- **Data Client.** The GPT is running as a client (Master), but the remote RTU device is permitted to initiate connection requests because the two sides are running in balanced mode.

Technical Note 33: Switched COMM circuits

When a remote connection request is accepted the circuit is flagged as connection and the reading of frames proceeds the same as if a dedicated circuit is being used.

WriteComm

WriteComm is called to transmit protocol data link blocks over the physical circuit. If the circuit is switched WriteComm will use the Connect and Disconnect services to establish a connection with a remote device. WriteComm needs to look at the type of write service requested by the GPT. The service type is contained in the control field the GPTLSCU structure supplied on the call. If the service request is LSDU_REQUEST (local station master is initiating a new request with a remote RTU), or LSDU_INDICATION (local station RTU is initiating a spontaneous transmission to a remote master) WriteComm must perform the following steps:

- 1. Disconnect.** WriteComm must check to see if a connection to the remote device already exists. If a connection is up it can be used for the write. A connection to a remote device might exist, but it might not be the destination device specified in the write requests. WriteComm should check the destination device in the request LSDU.Destination and verify that it is the same destination used to open the current connection. If it is not WriteComm should disconnect the current connection and mark the circuit as disconnected.
- 2. Connect.** WriteComm must check to see if the circuit is connected. If it is it can be used for the write. If the circuit is not connected WriteComm should call the COMM connect service to establish a connection. LSDU.Destination should be stored as a property of the new connection.