APPLICATION NOTE

AN_2901_010 **JULY 2005**

73M2901CE Quasi-Synchronous Operation

The Teridian 73M2901CE modem supports the use of a host-based software module for V.42 / MNP 2-4 Error correction Protocol and SDLC operation. These protocol operations require a special mode from the 73M2901CE called quasi-synchronous. This mode of operation is enabled via the ATY4 command.

Quasi-synchronous operation (sometimes called the pseudo-synchronous) is the ability of the modem to perform an asynchronous to synchronous conversion (and vice versa) to the data going over the DTE interface to and from the Host system. The following discussion details the operation of this mode.

HDLC, as well as V.42 and MNP Error correcting protocols, use a synchronous data format. The data is sent via 8-bit octets synchronized by the Transmit and Receive Clocks. A problem arises when sending data to and from the host system because most hosts do not have a synchronous connection to the modem data pump. These systems generally only have an asynchronous UART for sending and receiving serial data. A UART does not use the Transmit and Receive Clocks from the modem; instead it transfers data as characters using extra bits called Start and Stop bits to transfer timing information. If the UART is not sending data, the Transmit pin is held in a high, or in a “Marking” state. If the UART wishes to transmit a byte (Octet) of data, a “Start Bit” is transmitted first. A Start Bit or low state is sent for one bit time before the first bit of data is transmitted. After the 8 bits of data are transmitted, the UART returns to the high, or “Marking” state for an additional bit time before more data can be transmitted. This trailing “Marking” time is called the “Stop Bit”. These extra bits added to the data stream represent 20% more data than what would normally be transmitted in a synchronous fashion.

If the modem needs to send and receive synchronous data, there is a fundamental compatibility problem communicating with a UART interface. This is where the quasi-synchronous mode becomes useful. In quasi-synchronous mode, after the initial V.22 or V.22bis connection is established, the modem will automatically remove these extra start and stop bits from the transmitted data stream. This allows the Host system to use the more commonly found UART to connect to the modem, yet still send and receive synchronous data with the other modem. A visual representation of the data transfer across the UART to the modem is depicted on the following page.

The sending process is easy to comprehend, but the receiving side is somewhat harder. The problem here is that the receiving modem no long can look for start bit to tell where the Octet data boundaries were in the transmitted data

Quasi-Synchronous Process for Transmission

Data To Send

X0	X1	X2	X3	X4	X5	X6	X7
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7

Data After Start and Stop Bits Are Added

ST	X0	X1	X2	X3	X4	X5	X6	X7	SP
ST	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	SP

Asynchronous Serial Data That Is Sent to the TXD Pin of the Modem

ST	X0	X1	X2	X3	X4	X5	X6	X7	SP	ST	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	SP

Data that is Transmitted

X0	X1	X2	X3	X4	X5	X6	X7	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7

Quasi-Synchronous Process for Reception

Data that is Received By the Modem

W6	W7	X0	X1	X2	X3	X4	X5	X6	X7	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Z0	Z1

Received Data Sent to Host From Modem

ST	W6	W7	X0	X1	X2	X3	X4	X5	SP
ST	X6	X7	Y0	Y1	Y2	Y3	Y4	Y5	SP
ST	Y6	Y7	Z0	Z1	Z2	Z3	Z4	Z5	SP

Data After Host Processing

X0	X1	X2	X3	X4	X5	X6	X7
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7

A modem receiving data in the quasi-synchronous needs to do the opposite actions of the transmitting channel by re-inserting the missing start and stop bits so the data can be send over the UART receive path. In the example for the receive path, the data is no longer aligned with the original data boundaries. Synchronous data has unique data patterns called “flags” (7E hex) that mark the boundaries of the HDLC frames and this is what the host-based HDLC protocol will be looking for when it processes the data. Once the flags have been identified the rest of the data can be aligned to the original boundaries. This illustrates that the received data may no longer appear to be the same as the octets that were sent since the receive sync to async conversion process does not care where the beginnings of the octets were that were sent. This all looks a little strange at the serial interface, but can easily be recognized by the protocol. This flag pattern is not allowed in the data fields and “zero insertion” is used after any sequences of 5 ones in any field except the flags so a “fake flag” cannot be transmitted. These extra zeros are removed after the flags are detected in the receiver. The extra zeros are not used in the CRC computation.

There are a couple of things the system designer must know about in order to implement the Quasi-synchronous mode using the 73M2901CE.

1. Quasi-synchronous mode should only be used with Teridian Host Based Protocol software or software that can perform the same functions. Contact your Teridian representative for information about receiving and using this code module.
2. The TIES Escape Sequence (+++) does not work in Quasi Synchronous mode. For this reason, the designer should use the DTR signal in conjunction with the R2 command in order to terminate a connection.

A typical command string to enable the Quasi-synchronous mode of operation using the Host Based Protocol Engine, and DTR control of the modem is as follows:

```
ATR2Y4<CR><LF>
```

Please see your Teridian representative for more information on the use of the Host Based Protocol Engine.

Maxim cannot assume responsibility for use of any circuitry other than circuitry entirely embodied in a Maxim product. No circuit patent licenses are implied. Maxim reserves the right to change the circuitry and specifications without notice at any time.

Maxim Integrated Products, 120 San Gabriel Drive, Sunnyvale, CA 94086 408-737-7600