ZLP Electronics

23rd November 2012.

Application programming notes for the DigiMaster microFSK interface.

microFSK is a “TNC” type interface developed by G4ZLP / ZLP Electronics to enable serial data from a host to be presented to any device at a user selectable serial baudrate. The data stream presented to the destination equipment can be at any baudrate, the data stream to the device can be presented slower or can be presented faster than the original data stream. The presented data stream can be with or without translation from one format to another.

In this case the interface is used to convert serial data from a host in to a format suitable for use in the transmission of FSK signals in several common formats. The baudrate can be any whole number (45.45baud is close enough (1%) to be considered the same as 45 baud).

ALL common baudrates between 10 and 300 baud have been used with the interface as have baudrates that are “user defined”, the interface will work correctly regardless of the baudrate value specified. (The interface will operate with any baudrate, any value between 10-300 is valid). Of course, once you have FSK you have access to all the digital modes that use it (not just FSK RTTY).

Serial comms baudrate from / to the PC should be 19,200 baud.
Handshaking: Is via hardware CTS flow control.

The interface will return all characters sent to it for transmission but only if the interface has been "Opened", otherwise data will be ignored. The return characters will be returned exactly as they were sent to the interface and in the same order, even if they are invalid for the mode being used (the interface will translate or act on the characters as appropriate but the character returned to the PC will always be the same as it was received - commands are not returned. (This is the default action, which can be turned off)
With the exception of the Open Command, and the "GET_STATUS" command, the interface will not return anything in response to any command. The status LEDs on the interface will show the interfaces status. HOWEVER, you can send the "GET STATUS" to retrieve the values of parameters that are set in the interface. It is perfectly valid to issue a command then issue a status command to check that the command has been actioned.

In software it is envisaged that the user be allowed to edit / save several configuration profiles so that he can select one profile (consisting of one or more commands) to configure the interface for contest use at different baudrates etc... Along with the normal extra functions such as "brag" files and selection of CQ calls etc. etc..

microFSK Interface commands [V1.0]

1st byte is always; 0x01 = Command "announcer".
2nd byte is the command.
3rd and 4th byte are parameters which should be set to 0xFF if not used.

ALL commands to the interface are 4 bytes long and ALL start with the "announcer" 0x01 followed by the command and then by 2 parameters.
Any unused parameter should be set to 0xFF but will be ignored.
Commands sent to the interface are actioned but most commands return nothing.
However, after issuing a command, a "GET STATUS" command can be sent to retrieve the interfaces status as a check that a command has been actioned.
At present, there are NO user controls on the interface to affect any FSK parameters and so there is no need for any "polling". HOWEVER, it is possible that future versions of the interface be provided with controls and in response to the user making changes to controls on the interface, the status block may be returned to the PC at any time.

COMMAND: Open FSK-RTTY module: 0xF0

Command 0xF0. This is needed prior to doing anything as the interface has several functions (only one of which is FSK).
If this command is not issued the interface will NOT respond in any way and will ignore any data.
0x01,0xF0,Param[0],Param[1].

Param[0]=0x01 = OPEN FSK MODULE.
Param[0]=0x00 = CLOSE FSK MODULE.

If Param[1] = 0x01 then echo = ON // Data characters are returned to host.1
If Param[1] = 0x00 then echo = OFF // Data characters are NOT returned to the host.

[Note the similarity to the “GetSatatus” command; 0x01,0xF0,0xFF,0xFF]

The "Open" command returns a string of characters for information to be displayed on the screen to show user that the interface has been opened successfully;
eg,
Returns;-
Interface Open (or similar)
Firmware version 1.0 (or similar)

On exiting a control program the application should issue the CLOSE command.
The closed command returns a string of characters to be displayed on the screen to show the user that the interface has been closed.

The interface will always power up with default parameters set as follows;

[1] 5 bit mode (baudot)
[2] Normal polarity
[3] 1.5 stop bits
[4] UnShiftOnSpace ON.
[5] 45 baud

COMMAND: Get Status 0xF0

You can get all the parameters set on the interface by sending the command;-
0x01,0xF0,Param[0],Param[1].
Param[0] = 0xFF.
Param[1] = 0xFF. Unused.
The interface will respond with 12 bytes (fixed length) as follows;

BYTE [1] [1 byte] Start of Status: 0x01;
BYTE [2] [1 byte] Mode, 1 byte;
where value corresponds to;
0x05: baudot
0X07: 7 bit ascii
0X08: 8 bit ascii.
BYTE [3] [1 byte] Polarity, 1 byte;
Where value correspond to;
0x00 = normal
0x01 = reversed
BYTE [4] [1 byte] StopBits, 1 byte;
Where value correspond to;
0x01 = 1 stop bit
0x02 = 1.5 stop bit
0x03 = 2 stop bit
BYTE [5] [1 byte] USOS, 1 byte;
Where value correspond to;
0x00 = OFF
0x01, = ON
BYTE [6+7] [2 byte] Baudrate, 2 bytes:
Where values correspond to;
Hibyte
Lobyte
BaudRate=256*HiByte+LoByte.
BYTE [8] [1 byte] Interface OPEN, 1 byte;
Where value correspond to;
0x00 = FSK module is Closed
0x01 = FSK module is Open with echo OFF.
0x02 = FSK module is Open with echo ON.
BYTE[9] [1 Byte] 0=Interface in Rx, 1=Interface in Tx
BYTES[10] - [12] reserved.

COMMAND: Mode: 0xF1
eg. 0x01, 0xF1, Param[0], Param[1].
Param[0] = 0x05, 0x07, 0x08.
Param[1]=0xFF. Unused.

Where Param[0] Will be = 0x05 (Baudot) or 0x07 (7 bit ASCII) or 0x08 (8 bit ASCII).
Return=nothing.

COMMAND: Polarity: 0xF2
eg. 0x01, 0xF2, Param[0], Param[1].
Param[0] = 0x00 or 0x01
Param[1]=0xFF. Unused.

Where Param[0] = 0x00 (normal polarity), or 0x01 (reverse polarity)
Return=nothing.

COMMAND: StopBits: 0xF3
eg. 0x01, 0xF3,Param[0],Param[1].
Param[0]=0x00 or 0x01 or 0x02.
Param[1]=0xFF. Unused.

Where Param[0];
0x01 = 1 stop bit
0x02 = 1.5 stop bits
0x03 = 2 stop bits.
Return=nothing. Unused.

COMMAND: UnshiftOnSpace: 0xF4
eg. 0x01, 0xF4,Param[0], Param[1].
Param[0]=0x00 or 0x01
Param[1]=0xFF. Unused.

Where Param[0];
0x00 = 0 means Off
0x01 = 1 means On.
Return=nothing.

COMMAND: Baudrate: 0xF5
eg. 0x01, 0xF5, Param[0], Param[1].
Param[0]=hibyte
Param[1]=lobyte.

Where first 0x?? =hi byte of speed and 2nd byte = low value of speed.
eg, 110 baud command = 0x00,0x89,0x00,0xC8,
and 300 baud would be 0x00,0x89,0x01,0x2C.
Return=nothing.
Baudrate = param[1]*256 + param[2]

The interface will transmit at any baudrate and you can select the baudrate to be between 10 and 300 (and more) baud. The interface will adjust to transmit at ANY baudrate specified that is between 10-300(+) baud.
ANY baudrate is valid for the interface and the interface will transmit at ANY baudrate specified... 33 baud is fine, as is 67 baud, or 132 baud, or 180 baud, whatever you like.

COMMAND: Tx/Rx 0xFF
0x01,0xFF,Param[0],Param[1].
Param[0] = 0x00 = Rx
Param[0] = 0x01 = Tx

Param[1] = 0xFF. Unused.

Return nothing

A typical sequence of events could be;

[1] Open serial port
[2] Start Read thread
[3] Start Write thread
[4] Send the OPEN command to the interface, and issue commands to ensure interface is in its default state.
[5] Display all responses from Read thread
[6] Send commands to get the status and / or set up the interface parameters if not using the default settings
[7] Issue the Tx command when appropriate
[7] Send DATA for transmission to the interface
[8] Receive the same DATA returned from the interface as the interface transmits it and display it as a "transmission progress indicator".
[9] Issue the Rx command.
[10] Repeat 7 through 9 as required
[11] Before exiting the software issue the Close command to close the interface.
Also issue commands to ensure interface is in its default state.

Interface parameters can be changed at any time, though it possible it is not advisable to change parameters during transmission.

This document is provided on the understanding that this interface is in the final stages of development. The information provided is accurate and applies to the interface at this developmental stage. It is possible that future changes need to be made to the interface that require changes to this information.

microFSK (or alternatively mFSK) is the intellectual property of ZLP Electronics.