

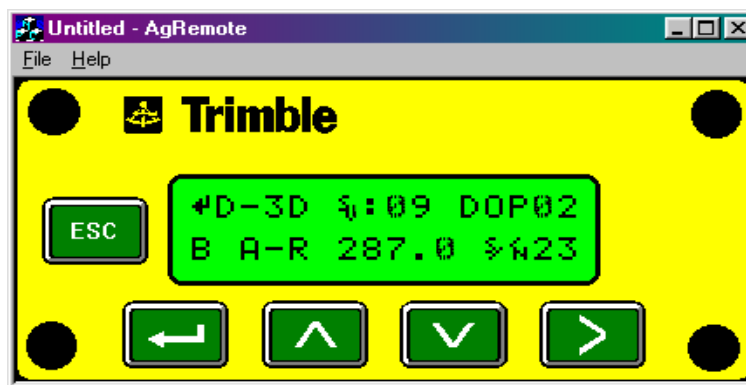
1 Remote Display Interface

1.1 Introduction

The remote display interface allows a third-party developer to easily take advantage of the configuration and status functionality provided by an AgGPS 124/132 front panel display in their own application using an AgGPS receiver that may not have its own display (i.e. AgGPS boardsets). If the developer is using an AgGPS receiver that has a display, the remote display interface allows the developer more flexibility in where the receiver information is displayed so that the receiver itself does not have to be mounted in the operator's field of view. The developer's system, whether it is a yield monitor with a display and keypad, a touch-screen task controller, or a laptop or palmtop PC, for example, will be referred to as the user terminal in this document.

A sample display from the AgGPS 124/132 front panel is show in Figure 1-1. This sample display is taken from the AgRemote program which uses the remote display interface to provide receiver configuration and status capability in a Windows 95/98/NT environment.

Figure 1-1 Sample Display from AgRemote Program



It is important to keep in mind the distinction between the remote display interface and the PC program, AgRemote. AgRemote is an example of how the remote display interface can be implemented by a developer. However, developers are not limited to using the AgRemote program if they would rather write their own or integrate the functionality into an existing product using the interface defined in this document. The remote display interface itself can be applied to any platform while the PC program AgRemote runs on Windows 95/98/NT.

1.1.1 Design Philosophy

The remote display interface is modeled after the virtual terminal concept where the user terminal does not need to know about the functionality available in the remote receiver. It only needs to know how to perform a limited set of display functions and how to pass on key presses from the user to the remote receiver. In the case of the remote display interface, the display functions are limited to displaying two text lines that are 16 characters long and displaying the cursor at a specified location and with a specified mode (i.e. solid cursor, blinking, etc.).

1.1.2 Benefits

- **Simpler interface to implement.** The user terminal software only needs to implement a small set of commands and reports (either in TSIP for serial port applications or TPCAN for CAN bus applications) in order to implement a large set of control and status functionality. Implementing this functionality using standard TSIP packets to configure port parameters, GPS masks, satellite frequencies, swath widths, etc. is a much larger development effort.
- **Automatic compatibility with latest firmware.** Since all of the display content is controlled by the receiver, any new feature additions that require control or status information are automatically available without having to change the user terminal software.
- **Manuals updated automatically.** The Ag124/132 manual is updated to contain the latest screen information available in a new firmware release, so documentation of the latest remote display screens is already available in this manual.

1.1.3 Implementation Overview

The following sequence of events needs to be implemented, either via TSIP or TPCAN messages described below, in order to support remote display functionality.

1. Establish connection with receiver either over serial link or CAN bus. Querying for the firmware version and receiver type may be useful in determining whether the unit supports the remote panel interface. The AgGPS receiver line, except for Beast-board based products, support the remote panel interface via TSIP in versions 1.31 and later. (Beast-boards are identifiable by their TSIP machine ID in TSIP packet 0x4B or by hardware type in TSIP packet 0x8F/0x7F. The Beast machine ID is 97 and the Beast hardware type is 1.)
2. Request remote panel configuration to determine if remote display is enabled. Send the command to enable automatic remote display contents updates.
3. Parse incoming remote display contents messages and display them on the user terminal.
4. Send key press command messages to the receiver indicating when the appropriate key presses have been made.
5. Once per minute, the request screen contents message must be sent to the receiver to maintain the automatic remote display updates. This provides a timeout mechanism if the receiver is disconnected from the user terminal without disabling the remote display automatic reporting. This functionality will be supported in versions 1.40 and later.
6. Before terminating the remote display session, disable the automatic remote display reporting.

1.2 TSIP Implementation

The TSIP packets listed in Table 1-1 must be implemented to support the remote display interface. .

Table 1-1 TSIP Remote Display Packet Summary

ID	Name	Direction
0x8E 0x8C 0x00	Display Key Press Command	to receiver
0x8E 0x8C 0x01	Remote Display Screen Contents Request	to receiver
0x8E 0x8C 0x03	Remote Display Configuration Command	to receiver
0x8E 0x8C 0x04	Extended Remote Display Configuration Command	to receiver
0x8E 0x8C 0x05	Remote Event in line state change report	to receiver
0x8F 0x8C	Remote Display Information Acknowledgment	from receiver
0x8F 0x8C 0x01	Remote Display Screen Contents Report	from receiver
0x8F 0x8C 0x02	Remote Output Line Control	from receiver
0x8F 0x8C 0x03	Remote Display Configuration Report	from receiver
0x8F 0x8C 0x04	Extended Remote Display Configuration Report	from receiver
0x8F 0x8C 0x05	Remote Event in line state request	from receiver

In addition, the developer may find it useful to implement a few additional packets to help in establishing the TSIP connection and troubleshooting connection problems:

- 0x13 Unparseable Packet Received Report which will be sent by the receiver if it receives a packet it cannot parse or is not supported by the firmware version.
- 0x1F Request Version Information Command and the corresponding 0x45 Version Report which are a useful way of determining that the TSIP connection has been established and information about the receiver (i.e. type, version of firmware) that may be necessary to determine if the remote display interface is available.

These additional packet definitions are available in the TSIP manual that can be found on the Trimble FTP site at ftp://ftp.trimble.com/pub/agriculture/Diagnostic_Tools.

1.2.1 TSIP Communication To the Remote Receiver

The term “command packet” defines communication to the remote receiver from the user terminal.

0x8E 0x8C Command Packet 0x8E 0x8C Remote Display Communication to Receiver (Commands)

The Command Packets under 0x8E 0x8C control the remote display functionality. The content of the packet varies depending on the Data Type (byte 1).

0x8E 0x8C 0x00 Command Packet 0x8E 0x8C 0x00 Display Key Press Command

The key press command packet simulates sending a key press to the display. Whenever the end user application wants to move around the remote display (i.e. in response to a key press on its own terminal, touch screen, etc.), this packet is sent to the receiver to indicate that the receiver should process the specified key press action. The receiver will acknowledge the key press command by sending Report Packet 0x8F 0x8C.

Table 1-2 Send Remote Display Key Press

Byte #	Item	Type	Value/Units	Meaning
0	Subpacket ID	BYTE	0x8C	Remote Display Information
1	Data type	BYTE	0x00	Command a key press
2	Key Pressed	BYTE	<i>flag</i>	Key pressed: 0 Enter key pressed 1 Up arrow pressed 2 Down arrow pressed 3 Right arrow pressed 4 Escape key pressed 6 Enter key held long 7 Enter key released after long hold 8 Left arrow pressed 9 Function key 1 pressed 10 Function key 2 pressed 11 Function key 3 pressed 12 Function key 4 pressed 13 Function key 5 pressed 14 Function key 6 pressed
3-6	Reserved	BYTE	0x00	Reserved (set to zero)
7-8	Checksum	INTEGER	<i>checksum</i>	Checksum

For receivers providing both beacon and satellite differential capability, the “home screen” on the receiver responds to a long enter key press (i.e. holding down the enter key for several seconds) differently than a regular enter key press in order to switch from beacon to satellite differential operation. To indicate that the enter key is being held down for a long time, send the “Enter key held long” key press and then send the “Enter key released after long hold” to indicate that the key press has ended.

0x8E 0x8C 0x01 Command Packet 0x8E 0x8C 0x01**Remote Display Screen Contents Request**

This packet is used to request the remote display screen contents once. The receiver will respond by sending Report Packet 0x8F 0x8C 0x01. If automatic reporting of the screen contents is configured, this packet must be sent once per minute to maintain that configuration (version 1.40 and later). This provides a timeout mechanism if the receiver is disconnected from the user terminal so the automatic reporting does not continue indefinitely if not needed.

Table 1-3 Request Remote Display Screen Contents Request

Byte #	Item	Type	Value/Units	Meaning
0	Subpacket ID	BYTE	0x8C	Remote Display Information
1	Data Type	BYTE	0x01	Request current screen contents
2-3	Checksum	INTEGER	<i>checksum</i>	Checksum

0x8E 0x8C 0x03 Command Packet 0x8E 0x8C 0x03**Remote Display Configuration Command**

The packet shown in Table 1-4 is used to request the remote display configuration. The receiver will respond by sending Report Packet 0x8F 0x8C 0x03. Table 1-5 shows the packet used to change the remote display configuration (i.e. activate or deactivate the remote display). The receiver will acknowledge the configuration change by sending Report Packet 0x8F 0x8C.

Table 1-4 Request Remote Display Configuration

Byte #	Item	Type	Value/Units	Meaning
0	Subpacket ID	BYTE	0x8C	Remote Display Information
1	Data Type	BYTE	0x03	Configuration
2-3	Checksum	INTEGER	<i>checksum</i>	Checksum

Table 1-5 Set Remote Display Configuration

Byte #	Item	Type	Value/Units	Meaning
0	Subpacket ID	BYTE	0x8C	Remote Display Information
1	Data Type	BYTE	0x03	Configuration
2	Display Location	BYTE	Table 1-6	Determines whether information is displayed on receiver front panel or remote front panel
3-6	Reserved	BYTE	0x00	Reserved (set to zero)
7-8	Checksum	INTEGER	<i>checksum</i>	Checksum

Table 1-6 Byte 2, Display Location

Bit	Meaning
0	Display information to front panel: 0: Disable 1: Enable (<i>default</i>)
1	Display information to remote display via serial port. Enables automatic output of screen content and cursor information packets 0: Disable (<i>default</i>) 1: Enable

0x8E 0x8C 0x04 Command Packet 0x8E 0x8C 0x04**Extended Remote Display Configuration Command**

The packet shown in Table 1-7 is used to request the **extended** remote display configuration. The receiver will respond by sending Report Packet 0x8F 0x8C 0x04. Table 1-8 shows the packet used to change the remote display configuration (i.e. activate or deactivate the remote display). The receiver will acknowledge the configuration change by sending Report Packet 0x8F 0x8C.

Table 1-7 Request Extended Remote Display Configuration

Byte #	Item	Type	Value/Units	Meaning
0	Subpacket ID	BYTE	0x8C	Remote Display Information
1	Data Type	BYTE	0x04	Configuration
2-3	Checksum	INTEGER	<i>checksum</i>	Checksum

Table 1-8 Set Remote Display Configuration

Byte #	Item	Type	Value/Units	Meaning
0	Subpacket ID	BYTE	0x8C	Remote Display Information
1	Data Type	BYTE	0x04	Configuration
2	Display Location	BYTE	Table 1-6	Determines whether information is displayed on receiver front panel or remote front panel
3-6	Reserved	BYTE	0x00	Reserved (set to zero)
7-8	Checksum	INTEGER	<i>checksum</i>	Checksum

Table 1-9 Byte 2, Display Location

Bit	Meaning
0	Display information to front panel: 0: Disable 1: Enable (<i>default</i>)
1	Display information to remote display via serial port. Enables automatic output of screen content and cursor information packets 0: Disable (<i>default</i>) 1: Enable

0x8E 0x8C 0x05 Command Packet 0x8E 0x8C 0x05**Remote Event Line State Change Report**

This packet, shown in Table 2-7, is sent to the receiver by the remote display to state that one of the event lines into the remote display has changed state.

Table 1-10 Remote Event Line State Change Report

Byte #	Item	Type	Value/Units	Meaning
0	Subpacket ID	BYTE	0x8C	Remote display information
1	Data Type	BYTE	0x05	Remote Event line state change report
2	Event port	BYTE		Event line (port) number
3	Line state	BYTE	0 or 1	Event line state
4	Reserved	BYTE	0x00	Reserved (set to zero)

Table 1-10 Remote Event Line State Change Report

Byte #	Item	Type	Value/Units	Meaning
5	Reserved	BYTE	0x00	Reserved (set to zero)
6	Reserved	BYTE	0x00	Reserved (set to zero)
7	Reserved	BYTE	0x00	Reserved (set to zero)
8-9	Checksum	INTEGER	<i>checksum</i>	Checksum

1.2.2 TSIP Communications From the Remote Receiver

The term “report packet” defines communication from the remote receiver to the user terminal.

**0x8F 0x8C Report Packet 0x8F 0x8C
Remote Display Information Report**

Report Packet 0x8F 0x8C is used to report and control the remote display functionality. The contents of the reports varies, depending on the request for data included in Command Packet 0x8E 0x8C.

Remote Display Information Acknowledgment

This form of the packet acknowledges when the remote display is enabled or disabled, or when a key press message is received.

Table 1-11 Remote Display Acknowledgment

Byte #	Item	Type	Value/Units	Meaning
0	Subpacket ID	BYTE	0x8C	Remote display information
1-2	Checksum	INTEGER	<i>checksum</i>	Checksum

0x8F 0x8C 0x01 Report Packet 0x8F 0x8C 0x01**Remote Display Screen Contents Report**

This form of the packet, shown in Table 1-12, reports the data currently appearing on the display in ASCII format (except for a few special characters) and the cursor position and mode. This packet can be requested for a single output using Command Packet 0x8E 0x8C 0x01 or it can be configured to be output automatically whenever the screen contents changes using Command Packet 0x8E 0x8C 0x03.

Table 1-12 Remote Display Screen Contents Report

Byte #	Item	Type	Value/Units	Meaning
0	Subpacket ID	BYTE	0x8C	Remote display information
1	Data type	BYTE	0x01	Screen contents
2	Horizontal Cursor Position	BYTE	0-15	Current horizontal coordinate of cursor on display, i.e. character position on a line
3	Vertical Cursor Position	BYTE	0-1	Current vertical coordinate of cursor on display, i.e. line number Note that this line number may not match the line number in byte 5 if the cursor is on the other display line.
4	Cursor Mode	BYTE	<i>flag</i>	Current cursor mode: 0 No cursor 1 Blinking cursor 2 Underline cursor 3 Underline and blinking cursor
5	Line number	BYTE	0-1	Packet contains data for this line number of the 2-line display
6-22	Line contents	BYTE		16-display characters plus a termination character that may or may not be null. It is recommended that developers insert a null in byte 22. A description of the character set for the line contents will be added to the document.
23-28	Reserved	BYTE	0x00	Reserved (set to zero)
29-30	Checksum	INTEGER		Checksum

0x8F 0x8C 0x02 Report Packet 0x8F 0x8C 0x02**Remote Output Line Control**

This packet, shown in Table 1-14, is sent by the receiver to control the state of a remote output line, i.e. an alarm. It is sent automatically if the receiver has been configured to drive output lines. For example, the PSO interface allows the user to set criteria for tripping an audible alarm. To control the output line to this alarm remotely (i.e. the output line is not directly connected to the receiver), this packet is sent.

Table 1-13 Remote Output Line Control Packet

Byte #	Item	Type	Value/Units	Meaning
0	Subpacket ID	BYTE	0x8C	Remote display information
1	Data Type	BYTE	0x02	Remote output line control
2-5	Set Hi Mask	BYTE	<i>Table 1-15</i>	Set Hi for 32 lines(1 = Set Hi, 0 = No change)
6-9	Set Low Mask	BYTE	<i>Table 1-15</i>	Set Low for 32 lines (1 = Set Hi, 0 = No change)
10-12	Reserved	BYTE	0x00	Reserved (set to zero)
13-14	Checksum	INTEGER	<i>checksum</i>	Checksum

0x8F 0x8C 0x03 Report Packet 0x8F 0x8C 0x03
Remote Display Configuration Report

This packet, whose form is shown in Table 1-14, shows the receiver configuration to display data on its own display or a remote display (i.e. a PC) connected to a serial port. It is sent in response to a configuration request via Command Packet 0x8E 0x8C 0x03.

Table 1-14 Remote Display Configuration Packet

Byte #	Item	Type	Value/Units	Meaning
0	Subpacket ID	BYTE	0x8C	Remote display information
1	Data Type	BYTE	0x03	Remote display configuration
2	Display Location	BYTE	<i>Table 1-15</i>	Remote display options
3-6	Reserved	BYTE	0x00	Reserved (set to zero)
7-8	Checksum	INTEGER	<i>checksum</i>	Checksum

Table 1-15 Byte 2 Remote Display Options

Bit #	Meaning
0 (LSB)	Display information to receiver front panel 0: No 1: Yes
1	Display information to remote display via serial port. Enables automatic output of screen content and cursor information packet 0: No 1: Yes
2-7	Reserved (set to zero)

0x8F 0x8C 0x04 Report Packet 0x8F 0x8C 0x04***Extended Remote Display Configuration Report***

This packet, whose form is shown in Table 1-14, shows the receiver configuration to display data on its own display or a remote display (i.e. a PC) connected to a serial port. It is sent in response to a configuration request via Command Packet 0x8E 0x8C 0x04 .

Table 1-16 Extended Remote Display Configuration Packet

Byte #	Item	Type	Value/Units	Meaning
0	Subpacket ID	BYTE	0x8C	Remote display information
1	Data Type	BYTE	0x04	Remote display configuration
2	Display Location	BYTE	<i>Table 1-15</i>	Remote display options
3-8	ID	BYTE		unit ID (5 character Null terminated string)
9-12	reserved	BYTE	0x00	Reserved, Set to 0
13-14	Checksum	INTEGER	<i>checksum</i>	Checksum

Table 1-17 Byte 2 Remote Display Options

Bit #	Meaning
0 (LSB)	Display information to receiver front panel 0: No 1: Yes
1	Display information to remote display via serial port. Enables automatic output of screen content and cursor information packet 0: No 1: Yes
2-7	Reserved (set to zero)

1.3 Revision Notes

Table 1-18. Remote Display Interface Revisions

Rev	Description	Date	Approved
Pre-A	Initial Release	4/23/99	
A	Clarified some field descriptors	5/6/99	
B	Addition of introductory information and implementation outline	5/21/99	
C	Added extended configuration packets, event input packets	6/5/00	

