

Contents

- I. Introduction 2
 - Compatible Products 2
 - Supported Protocols and Features 3
- II. Device Hardware and Setup 4
 - A** Lantronix/Gridconnect Enhanced XPort Ethernet Controller 4
 - Critical Device Settings (TCP/IP) 4
 - B** Serial RS-485 interface (7-segment control mode only) 4
 - Critical Device Settings (Serial) 5
 - Device Wiring (Serial) 5
- III. 7-Segment Control Mode (Ethernet or RS-485) 6**
 - a) “Legacy” 7-Segment Protocol 6
 - Example displays: Legacy 7-Segment Protocol..... 6
 - b) “Extended” 7-Segment Protocol..... 7
 - Font size flag: <ESC> + “F” (0x1B 0x46) 8
 - Text color flag: <ESC> + “T” (0x1B 0x54) 9
 - Background color flag: <ESC> + “B” (0x1B 0x42)..... 10
 - c) “Extended” 7-Segment Protocol: Character Maps 11
- IV. Advanced Sign Control + Bitmap Mode (Ethernet Only)..... 13**
 - Protocol Structure..... 13
 - Request..... 13
 - Response 13
 - Sign Commands (Ethernet Only)..... 14
 - Command 0x01: GET Sign Info 14
 - Command 0x02: GET Sign Image..... 15
 - Command 0x04: GET Sign Brightness..... 15
 - Command 0x05: SET Sign Brightness 15
 - Command 0x06: GET Message Status 16
 - Command 0x08: SET Blank Message 16
 - Command 0x13: SET Bitmap Message 16

I. Introduction

This document outlines the accepted protocols and communication modes for Signal-Tech’s SA Flex (SAF) products.

Compatible Products

A compatible sign is indicated in its **Product Number** as “SAF”.

While there may be other compatible variants, these are the **standard** configurations:

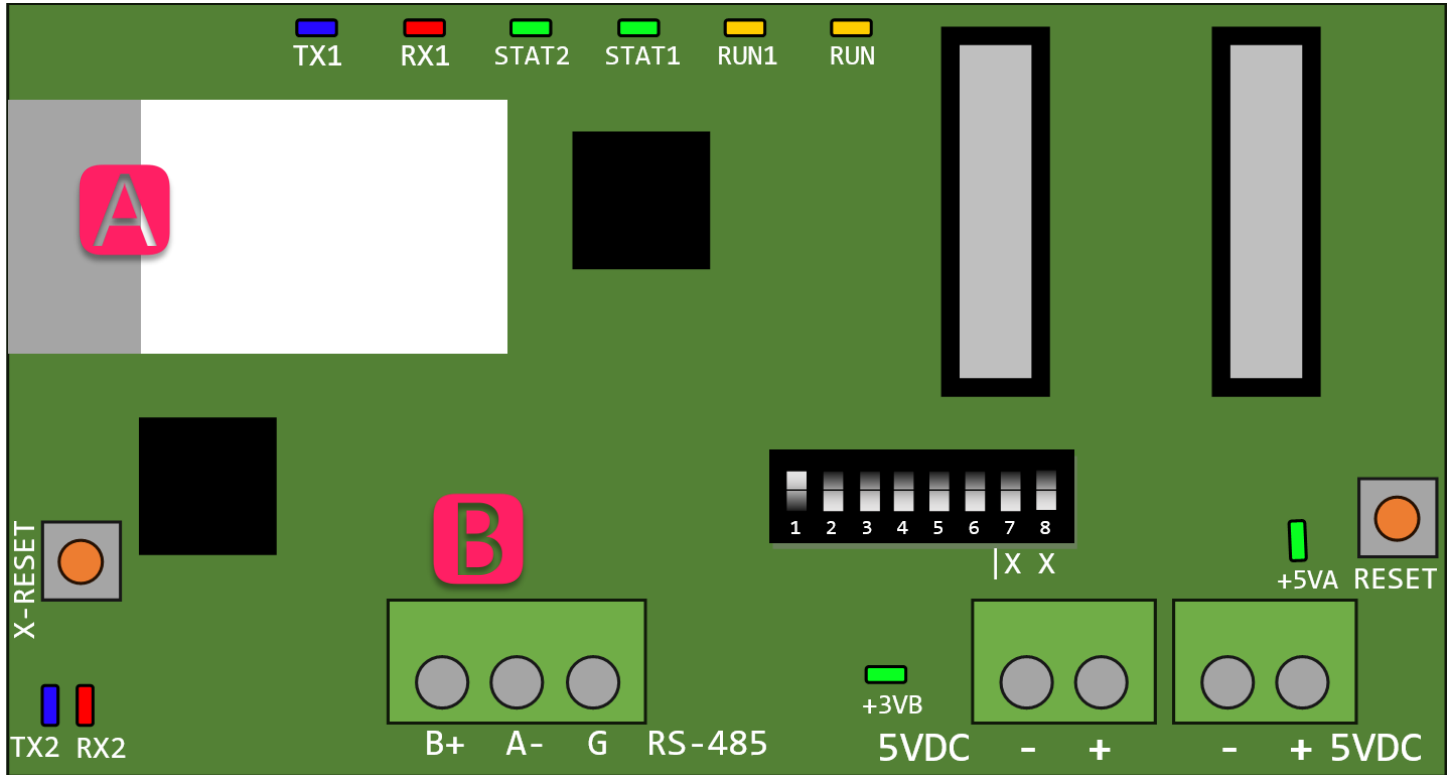
Product ID	Resolution (HxW)	Size class (HxW)	Sample displays
69113	16x64 px	7”x 26”	
69151	16x96 px	7”x 39”	
69152	16x128 px	7”x 51”	
69153	32x64 px	14”x 26”	
69143	32x96 px	14”x 39”	
68007	32x128 px	14”x 51”	

Supported Protocols and Features

The SA Flex product line supports two message protocols (click header to jump to section):

7-Segment Control Mode (Ethernet or RS-485)	Advanced Sign Control + Bitmap Mode (Ethernet Only)
<ul style="list-style-type: none">• Uses Signal-Tech’s 7-segment/LED Count Display protocol• Requires no changes to control software (if 7-segment protocol is already used)• Also compatible with SA- and S-SA signs <p>Additional sign commands (Jump to: “Extended” 7-Segment Protocol):</p> <ul style="list-style-type: none">• Text/background color control• Font size control• A full symbol library	<ul style="list-style-type: none">• Uses Signal-Tech’s RGB Protocol as a container• Allows bitmap images to be sent to the display once per second <p>Additional sign commands (Jump to: Sign Commands (Ethernet Only)):</p> <ul style="list-style-type: none">• Brightness control• Retrieval of hardware info: product ID, serial number, product image, manufacture date• Retrieve current message status (checksum)

II. Device Hardware and Setup



The SA Flex Controller has **two communication interfaces** (**A** and **B**):

For instructions on using the DIP switch bank for addressing, see **7-Segment Control Mode (Ethernet or RS-485)**.

A Lantronix/Gridconnect Enhanced XPort Ethernet Controller

The embedded “XPort” module provides a wired Ethernet interface to the sign controller.

All sign commands—bitmap, 7-segment, etc.—are supported via Ethernet.

The Ethernet controller has HTTP GUI (port 80) and telnet (port 9999) interfaces that can be used to configure a static IP address, a different TCP port, and/or a device password.

Critical Device Settings (TCP/IP)

The sign will receive the message payload over TCP/IP on port 10001.

By default, the XPort is configured to use DHCP. Use a DHCP router or download Lantronix [DeviceInstaller](#) to discover the device, then set a static IP if desired.

B Serial RS-485 interface (7-segment control mode only)

The SA Flex controller also features an RS-485 port, making it easy to replace an older 7-segment display.

The serial interface is limited to accept “Legacy” and “Extended” 7-segment commands only.

Critical Device Settings (Serial)

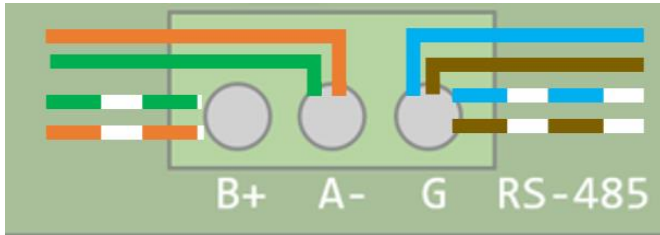
The settings below are not configurable on the controller. The host device/server should be configured for the following:

- **Protocol:** RS-485
- **Baud Rate:** 9600
- **Data Bits:** 8
- **Stop Bits:** 1
- **Parity:** None

Device Wiring (Serial)

Wiring diagram (CAT6 shown)

Note: Other twisted-pair cable, or shielded, RS-485-specific cable should perform as well as CAT6



B+	White/Orange White/Green
A-	Solid Orange Solid Green
G	(All others)

III. 7-Segment Control Mode (Ethernet or RS-485)

Go back to the [Device Hardware and Setup](#) section for configuration settings.

Additional hardware settings: When using 7-segment control—either over RS-485 or Ethernet—the Sign Address (SA) must be set using the controller’s DIP switch bank (**addresses 1-63**):



a) “Legacy” 7-Segment Protocol

Hex	16	16	02	[SA]	[CM]	[CD]	X1	X2	X3	X4	[CS]	03
Def	SYN	SYN	STX	Sign address	Command mode	Enable response	Digit 1	Digit 2	Digit 3	Digit 4	XOR Checksum	ETX

Following Signal-Tech’s proprietary LED Count Display Protocol, existing systems can control SA Flex signs without modifying the host software.

The 7-Segment/LED Count Display Protocol can be found here:
<https://www.signal-tech.com/downloads/led-count-display-protocol.pdf>

Notes for “Legacy” 7-Segment Protocol:

- Font will be 15px high and be right-justified
- Leading 0s will be removed
- “FULL” (<CM> 0x01) and “CLSD” (<CM> 0x03) will appear in red
- All other characters will appear in green

Example displays: Legacy 7-Segment Protocol

Hex sent:	16 16 02 01 01 01 30 31 32 33 01 03
Packet info:	Sign address = 1; <CM> = 1; displays FULL
Display (shown on 16x48 px sign):	

Hex sent:	16 16 02 3A 06 01 00 00 32 33 3C 03
Packet info:	Sign address = 58; <CM> = 06; displays 23
Display (shown on 16x48 px sign):	

b) "Extended" 7-Segment Protocol

Hex	16	16	02	[SA]	[CM]	[CD]	X1	X2	...	XN	[CS]	03
Def	SYN	SYN	STX	Sign address	Command mode	Enable response	Char 1	Char 2	...	Char <i>N</i>	XOR Checksum	ETX

Within the same protocol structure, the control software can also add the following to the character stream (X1,...XN):

1. <ESC> flags (0x1b) to control:
 - a. **Font size** (Default: 15px)
 - b. **Text color** (Default: Green)
 - c. **Background color** (Default: Black)
2. Upper ASCII values to represent **arrows** and other **common symbols** (Jump to: **CHARACTER MAP**)

Notes:

- Like the "Legacy" 7-segment mode, all text will be right-justified and begin on the top row
- Refer to the original [protocol document](#) for the checksum calculation
- The examples below **do not include complete data packets** unless otherwise noted
- Maximum number of bytes in character stream = **255**




Flags are defined on pages 8-10...

Font size flag: <ESC> + “F” (0x1B 0x46)

Insert this flag to select one of **three** font sizes. The default value is 0x01 (“Medium” 15px).

Hex	1B	46	NN
Def	<ESC>	F	Font index (defined below)

Note: Only **one font size** is allowed **per line**, i.e. a [CR] (0x0A) is required before the next font is selected.

Example: Font size flag (32x64px display shown)	Font	Hex in character stream
	Small (7px height) <ESC> + “F” + 00	0x1B 0x46 0x00
	Medium (15px height) <ESC> + “F” + 01 <i>(Default—no flag needed)</i>	0x1B 0x46 0x01
	Large (30px height) <ESC> + “F” + 02	0x1B 0x46 0x02

Text color flag: <ESC> + "T" (0x1B 0x54)

The text color flag may be used to interrupt the current foreground color at any time.

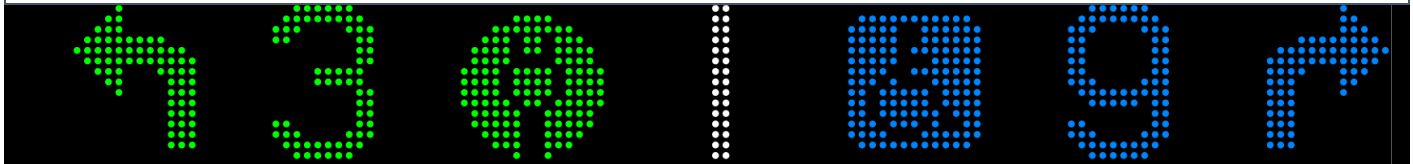
Hex	1B	54	[RR]	[GG]	[BB]
Def	<ESC>	T	Red value (00-FF)	Green value (00-FF)	Blue value (00-FF)

Note: The text color may be changed at any point (even within the same line).

Example: Text color flag (16x128px display shown):

Complete packet shown (ads 1):

16 16 02 01 06 01 **AA 20 33 20 B1 20** 1B 54 FF FF FF 7C 20 1B 54 00 00 FF **B3 20 39 20 AB** 73 03



Bytes	.	AA	20	33	20	B1	20	.	7C	20	.	B3	20	39	20	AB	
Def		[Sym]	[Sp]	"3"	[Sp]	[Sym]	[Sp]		" "	[Sp]		[Sym]	[Sp]	"9"	[Sp]	[Sym]	
Flags	Default size + color (no flag required)								Color Flag: 1B 54 FF FF FF		Color Flag: 1B 54 00 00 FF						

Background color flag: <ESC> + "B" (0x1B 0x42)

Insert this flag to change the background color. The default is 00-00-00 (black).

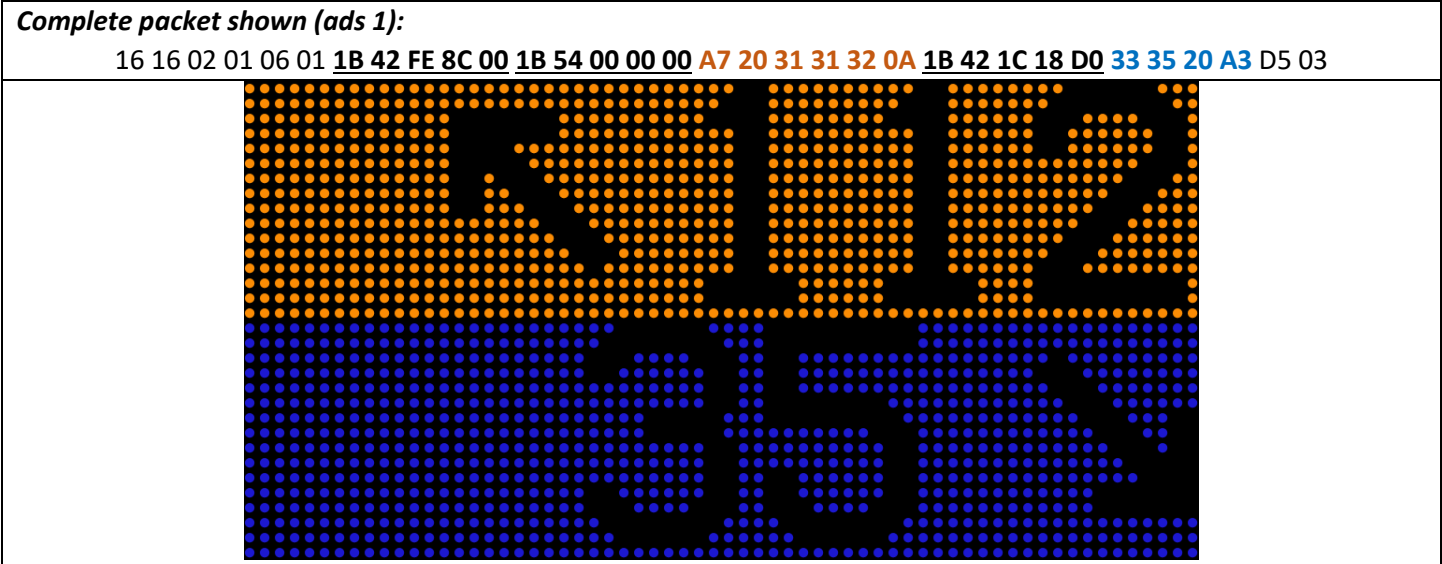
Hex	1B	42	[RR]	[GG]	[BB]
Def	<ESC>	B	Red value (00-FF)	Green value (00-FF)	Blue value (00-FF)

Note: Only **one background color** is allowed **per line**, i.e. a CR (0x0A) is required before the next background color is selected.

Example: Background color flag (32x64px display shown):

Complete packet shown (ads 1):

16 16 02 01 06 01 **1B 42 FE 8C 00 1B 54 00 00 00 A7 20 31 31 32 0A 1B 42 1C 18 D0 33 35 20 A3** D5 03



c) "Extended" 7-Segment Protocol: Character Maps
























8-px height

HEX	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_a	_b	_c	_d	_e	_f
0_																
1_																
2_	SP	!	"	#	\$	%	&	'	()	*	+	,		.	/	
3_	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4_	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5_	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6_	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7_	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
8_																
9_																
a_																
...																
f_																

16-px height

HEX	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_a	_b	_c	_d	_e	_f
0_																
1_																
2_	SP	!	"	#	\$	%	&	'	()	*	+	,		.	/	
3_	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4_	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5_	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6_	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7_	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
8_																
9_																
a_																
b_																
...																
f_																

32-px height

HEX	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_a	_b	_c	_d	_e	_f
0_																
1_																
2_	SP	!	“	#	\$	%	&	'	()	*	+	,	.	/	
3_	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4_	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5_	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6_	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7_	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
8_																
9_																
a_																
b_																
...																
f_		±							o							

End of “7-Segment Control Mode”

IV. Advanced Sign Control + Bitmap Mode (Ethernet Only)

Protocol Structure

Request

	Length	Description
<start>	1 byte	Always 0x09
<length>	4 bytes	The count of bytes in <data>
<command>	1 byte	The command byte (see Sign Commands (Ethernet Only))
<data>	variable	The sent data related to the command, if required, may be 0 bytes long (see “Request sent <data>” for each command)
<checksum>	8 bytes	The checksum calculated by adding the bytes in <data> and using the 64 least significant bits
<end>	1 byte	Always 0x03

Response

	Length	Description
<start>	1 byte	Always 0x10
<length>	4 bytes	The count of bytes in <data>
<command>	1 byte	The echoed command byte
<data>	variable	The sent data related to the command, if required, may be 0 bytes long (see “Response received <data>” for each command)
<checksum>	8 bytes	The checksum calculated by adding the bytes in <data> and using the 64 least significant bits
<end>	1 byte	Always 0x03

Important: These commands are only supported via TCP/IP (*not* over the serial port)

Hex	Name (link to section)	Modes	Description
0x01	Get Sign Info	Read	Returns XML encoded sign information, such as product ID and serial number
0x02	Get Sign Image	Read	Returns the sign's PNG primary image
0x04	Get Brightness	Read	Returns the brightness level of the sign (0=auto, 1=lowest, 15=highest)
0x05	Set Brightness	Set	Sets the brightness level of the sign (0=auto, 1=lowest, 15=highest)
0x06	Get Message Status	Read	Returns the last message status and checksum
0x08	Set Blank	Set	Tells the sign to blank the display
0x13	Set Bitmap Message	Set	Send .bmp data to the sign (up to once per second)

Each request's data format is explained in its own section below, along with examples of the request and response structure.

Command 0x01: GET Sign Info

Each sign controller is preprogrammed with XML configuration data that describes the messages on the sign, as well as some global sign data. The XML format is described in a later section of this document.

Request sent <data>: n/a

Response received <data>: <ASCII XML Sign Info>

XML Format:

```
<?xml version='1.0' encoding='UTF-8'?>
  <sign>
    <product_number>SAF16x64-10mm</product_number>
    <product_id>69113</product_id>
    <height>7.299</height>
    <width>26.197</width>
    <serial_number>0000-0000-0000</serial_number>
    <manufacture_date>1970-01-01</manufacture_date>
    <has_photo_eye>N</has_photo_eye>
    <pixel_height>16</pixel_height>
    <pixel_width>64</pixel_width>
    <pixels_per_module_height>16</pixels_per_module_height>
    <pixels_per_module_width>32</pixels_per_module_width>
  </sign>
```

Example:

Hex Sent	09	00 00 00 00	01	(omit)	00 00 00 00 00 00 00 00	03
Def	<start>	<length>	<get sign info>	<data>	<checksum>	<end>
Hex Received	10	00 00 00 01	01	[ASCII XML data]	NN NN NN NN NN NN NN NN (8-byte checksum)	03

Command 0x02: GET Sign Image

Each sign controller stores a transparent PNG image of the sign, which can be shown in the control software.

Request sent <data>: n/a

Response received <data>: <Binary PNG image data>

Example:

Hex Sent	09	00 00 00 00	02	(omit)	00 00 00 00 00 00 00 00	03
Def	<start>	<length>	<get sign image>	<data>	<checksum>	<end>
Hex Received	10	00 00 00 01	02	[Binary PNG data]	NN NN NN NN NN NN NN NN (8-byte checksum)	03

Command 0x04: GET Sign Brightness

Request sent <data>: n/a

Response received <data>: 0x01-0x0F (1-15)*

*Note: if value is 0, auto-dimming is enabled (not currently implemented)

Example:

Hex Sent	09	00 00 00 00	04	(omit)	00 00 00 00 00 00 00 00	03
Def	<start>	<length>	<get brightness>	<data>	<checksum>	<end>
Hex Received	10	00 00 00 01	04	0F	00 00 00 00 00 00 00 0F	03

Command 0x05: SET Sign Brightness

Request sent <data>: 0x01-0x0F (1-15)*

Response received <data>: 0x01-0x0F (1-15)*

*Note: 0x00 will enable full brightness, as auto-dimming is not currently implemented

Example:

Hex Sent	09	00 00 00 01	05	0F	00 00 00 00 00 00 00 0F	03
Def	<start>	<length>	<set brightness>	<data>	<checksum>	<end>
Hex Received	10	00 00 00 01	05	0F	00 00 00 00 00 00 00 0F	03

Command 0x06: GET Message Status

This command will get the <status> and <checksum> of the message currently on display.

<status> **0x00** means the .png file was properly displayed

<status> **0x01** indicates a problem with the received .png file.

Request sent <data>: n/a

Response received <data>: <status> <checksum>

Example:

Hex Sent	09	00 00 00 00	06	n/a	00 00 00 00 00 00 00 00	03
Def	<start>	<length>	<get message>	<data>	<checksum>	<end>
Hex Received	10	00 00 00 09	06	00 00 00 00 00 00 00 00 C8	00 00 00 00 00 00 00 C8	03

Command 0x08: SET Blank Message

Request sent <data>: N/A

Response received <data>: N/A

Hex Sent	09	00 00 00 00	08	n/a	00 00 00 00 00 00 00 00	03
Def	<start>	<length>	<get message>	<data>	<checksum>	<end>
Hex Received	10	00 00 00 00	08	n/a	00 00 00 00 00 00 00 C8	03

Command 0x13: SET Bitmap Message

The SA Flex display will accept BMP files embedded in the protocol's <data> field. This may be refreshed up to once per second (1FPS).

Request sent <data>: .bmp file, starting with header "BM" or "0x42 0x4D" (see below)

Response received <data>: Checksum of the request sent

Critical Bitmap file parameters

Make sure that the bitmap file meets the specifications below.

Reference: https://en.wikipedia.org/wiki/BMP_file_format

Supported file types	.bmp
Supported header types	BM
Supported color depths	RGB24 (8R-8G-8B) – 16M colors RGB565 (5R-6G-5B) – 65K colors RGB8 – 256 colors

Example:

Hex Sent	09	NN NN NN NN	13	42 4D ... NN	NN NN NN NN NN NN NN NN NN	03
Def	<start>	<length>	<Set msg>	<data>	<checksum>	<end>
Hex Received	10	00 00 00 08	13	NN NN NN NN NN NN NN NN NN	NN NN NN NN NN NN NN NN NN	03

Questions/feedback? Send an email to integrations@signal-tech.com or call 814-835-3000