CATTRON™

Messenger 600

User Manual

9M02-7000-A301-EN

CONNECT. CONTROL. PROTECT.

Revision History

VERSION	DATE	NOTES
1.0.0	05/2020	
B	03/2021	Document rebranded and contact information updated

Contents

Introduction

This User Manual describes installation and setup of the Messenger 600 product.

The Messenger 600 is a cellular, cloud-based monitoring, alarm notification and telemetry platform. It supports monitoring and reporting of up to 600 data values from the industry standard Modbus RTU serial protocol, as well as support of custom serial communications to external devices.

Features

The hardware feature set of this platform includes the following:

- Messenger 600 with LTE modem running on ATT or Verizon networks
- 12 VDC battery backup
- 120 VAC to 12 VDC power supply battery charger with main power fail signal
- RS485 port for Modbus RTU communication
- RS232 port for programming using the internal menu system
- Terminal blocks for field connections

1 Description

1.1 Capabilities

The Messenger 600 is a highly configurable platform for remote monitoring and control applications. Some of the capabilities are listed below.

- Virtual real-time transfer of monitored conditions
- Local computations from monitored conditions
- Time stamping of monitored data and events
- Battery-backed historical data/event buffers
- Automatic monitoring of max/min for analog values
- Event and data logging
- Telemetry of monitored conditions to host server-based applications via cellular
- OTA programming and diagnostics

1.2 Monitoring

All monitored values can be transmitted via cellular to the RemoteIQ™ cloud-based server. Monitored values are transmitted based on time or events. Notification events are based on rules set by the user and each event can generate an immediate report. Telemetry includes cellular connectivity and GPS for asset location.

The Messenger 600 reads values using the Modbus RTU protocol over an RS485 serial connection. The user programs each of the 600 values that are to be read into Messenger 600 channels, which are numbered from 301 to 900.

1.3 Host Server Communications

The Messenger 600 utilizes a proprietary protocol to send notifications and to receive OTA commands from the RemoteIQ server. Each notification sent typically consists of location, date/time, an event code and associated data. An event code provides a unique identifier to indicate the reason that the notification is being sent – for example, normal scheduled update or an engine diagnostic message received. A description of the protocol, format of messages and definition of event codes is available on request (reference protocol document “M09-PRTCLxxx”).

Some of the conditions on which notifications can be sent to the host server are listed below:

- Any monitored value exceeding a predefined or user-defined limit
- Any diagnostic message received from the engine bus
- A digital input changing state (on/off)
- A digital output changing state
- An analog input exceeding a user-defined delta for a user-defined period of time
- A scheduled update
- System faults
- Power on or reset

1.3.1 Alarm Notification

The RemoteIQ backend provides notification to designated personnel via one of the following:

- Text message
- Email
- Voice

Each method allows the user to acknowledge the message received.

1.3.2 Event Codes

Every message sent by the Messenger 600 to a host-based server application is triggered by an event. The event generates a message and the message contains an Event Code. The Event Code uniquely identifies to the server the reason the message is being sent. Some of the messages generated contain data; others serve simply as notification that a particular event has occurred. Protocol document “M09-PRTCLxxx” contains a list of all event codes.

1.3.3 Positive Acknowledgement

The Messenger 600 is configured to require a message acknowledgement from the host server providing complete end-to-end verified communication. Message acknowledgement provides a verifiable mechanism that a message was delivered, even during poor network conditions.

1.3.4 Store and Forward Data Queue

There are several scenarios where a message may not be deliverable – network down, host server down and poor connectivity, to name a few. In the event that a message cannot be delivered, it is stored in memory and is continually re-sent until it is properly acknowledged. This store and forward memory is non-volatile and remains intact during power off.

1.3.5 Real-Time Clock (RTC)

The RTC is used to time stamp data records and events. All messages sent to the host server contain a time stamp to provide a chronology of data/events to the end user. The RTC is battery-backed to provide time keeping during power off. If the RTC is configured to be automatically set, the Messenger 600 will set the time after every power on and it will perform a time check every midnight. If the RTC time differs from the actual time by more than 30 seconds, the RTC is adjusted.

The RTC can be set in one of the following ways:

Method	Description
Automatic via Cell	This is the default setting. The Messenger 600 will set the RTC from an internet NIST time server.
Automatic via GPS	The Messenger 600 will set the RTC from the date/time read from the GPS module.
Manually	The time is set via the Debug port through the Maintenance menu.
OTA/SMS	The RTC is set from an OTA config command or an SMS config command.

1.3.6 Packaging

The Messenger 600 comes mounted on a DINrail for easy mounting in panels, or it is available in a NEMA 4X enclosure.

2 Installation and Setup

This section provides information on installing the Messenger 600 and confirming its initial operation.

⚠	**WARNING**
	IT IS RECOMMENDED THAT YOU READ THIS ENTIRE SECTION BEFORE STARTING THE INSTALLATION.

2.1 Installation Steps

Installation consists of the following steps:

1. Unpack the Equipment.
2. Mount the Equipment.

   The Messenger 600 should be mounted in a vertical position to try and minimize the chance of water entering through the antenna connection. The antenna wire should have a service loop just below the antenna connectors.
3. Connect Main Power.

2.2 Mounting the Equipment

The Messenger 600 is shipped mounted to a DINrail as shown in Figure 1: Messenger 600 DINrail Mount.

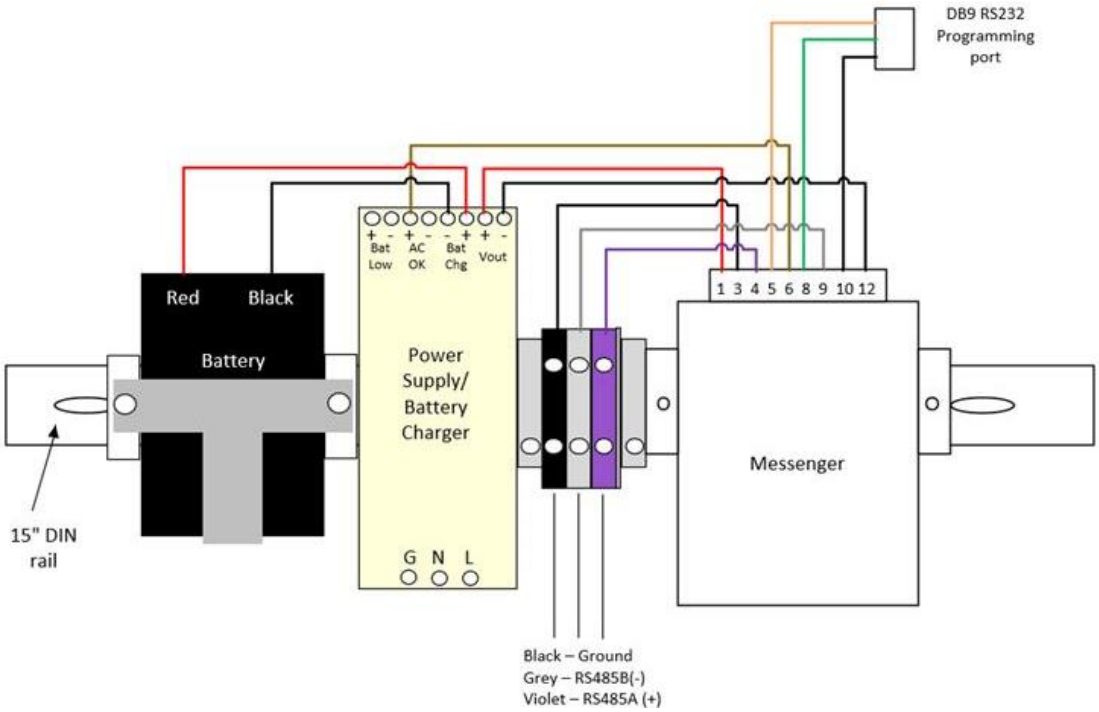

Figure 1: Messenger 600 DINrail Mount

2.3 Mounting the Antenna

In general, the antenna should be mounted with an unobstructed view of the sky. The GPS side works best when it can see the horizon. If the antenna is mounted outside and may be subject to lightning, a surge arrestor can be inserted between the Messenger 600 and the antenna. If the antenna is mounted inside, it should be located near a window.

Note: Service loops should be provided for the antenna cabling, near the antenna connections, to minimize water ingress through the RF COAX connections.

2.4 DIP Switch/Jumper Settings

The Messenger 600 uses an on-board DIP switch and jumpers to configure application specific IO and set operational modes. DIP SW3 is used to set operational modes. See Figure 2 for switch and jumper locations.

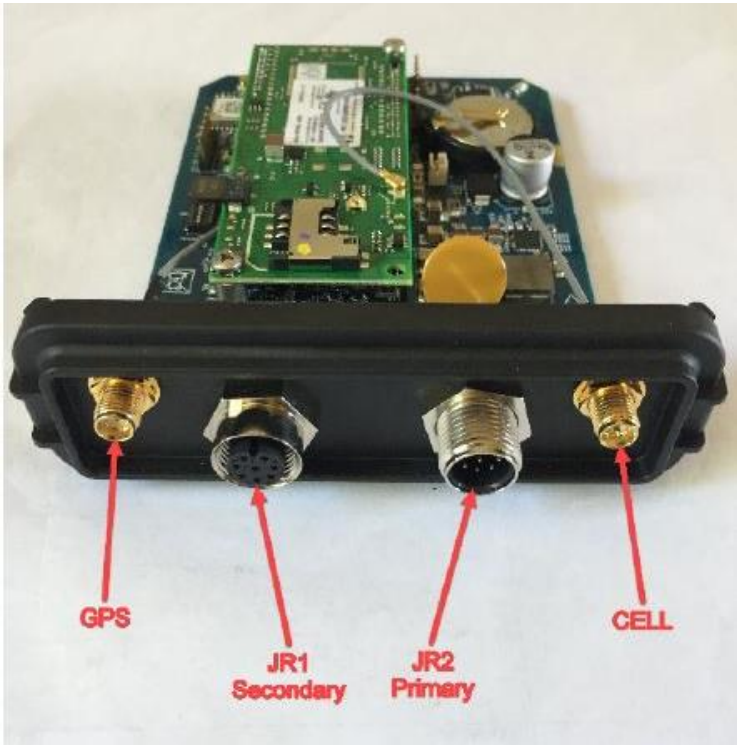

Figure 2: Messenger 600 IO Connectors

2.5 DIP Switch 3 Decode

DIP Switch 3 is a 4-position dip switch located on the left side of the board; refer to Figure 3 to see the location. If any position on switch 3 is changed, the power must be cycled for the new switch positions to be read. Table 1 describes the position of the dip switches.

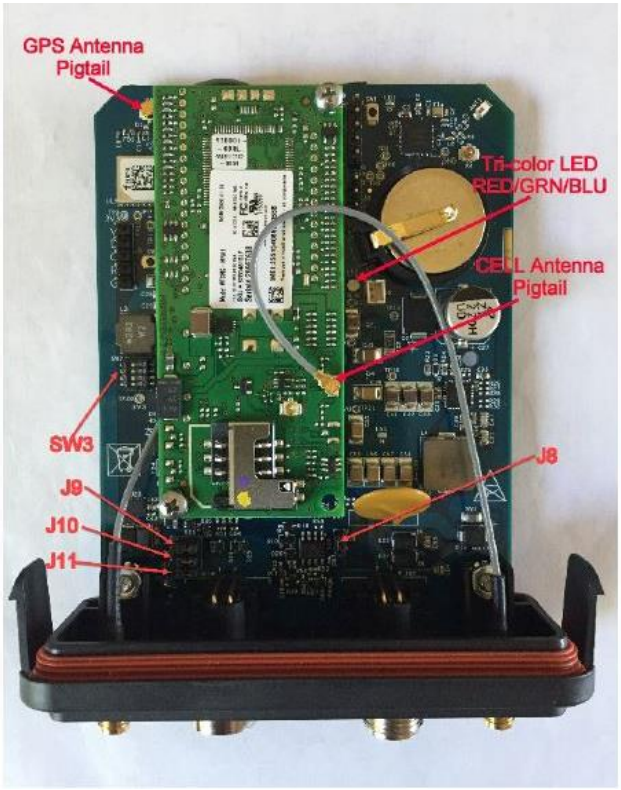

Figure 3: Messenger with Enclosure End-Cap Attached

Table 1: DIP Switch 3 Decode

Position	1	2	3	4	Description
Function					
Enter BOOT Loader	↑	↑	↑	↑	Used for code download via internal debug port - RS232, 57600,8,1,N
Factory Default Settings	↓	↓	↓	↓	

Key: ↓ = switch in “OFF” position, ↑ = switch in “ON” position

Note: Switch positions 1 and 2 will override any other settings for Ports 2, 3, or 4.

3 LED States

There is one tri-color LED visible to the user to indicate various system conditions, whose location is shown in Figure 4: Status LED Location.

Figure 4: Status LED Location

These conditions are conveyed to the user via LED color and blink patterns. On power-up, an LED test is performed by blinking all LEDs every second for 3 seconds. Following the LED test, the LEDs blink based on the conditions described below.

The number of blinks will range from one to three. The general philosophy when deciding behavior will be as follows:

- One blink will convey a state that the module is expected to be in most often (the “OK” state)
- Three blinks are used to convey that there is an issue that may need attention
- Two blinks are used as needed to convey a state that may be of interest to the user
- No blinks (LED solid on or off) indicates that the system is no longer functioning; contact Cattron at www.cattron.com/contact for assistance

Status LED Behavior

The Status LED will cycle through blinking each color for the appropriate number of times to convey the state of the corresponding module, as shown in Figure 5: LED Pattern Definition and Table 2*: LED Color/Blink Patterns*. There is a pause between color changes.

The “Comm” signal refers to the RS485 communication from the Messenger 600 to an external device. The “On” blink pattern means that communication is configured and is working without errors. The “Off” blink pattern means that no communication is configured. The “Fault” blink pattern means that communication is configured but there are errors in the communication. This is typically caused by invalid Modbus ID or Modbus registers numbers, incorrect baud rate, data length, or parity. This fault can also occur when the wiring is not correct (usually the RS485 wires are reversed).

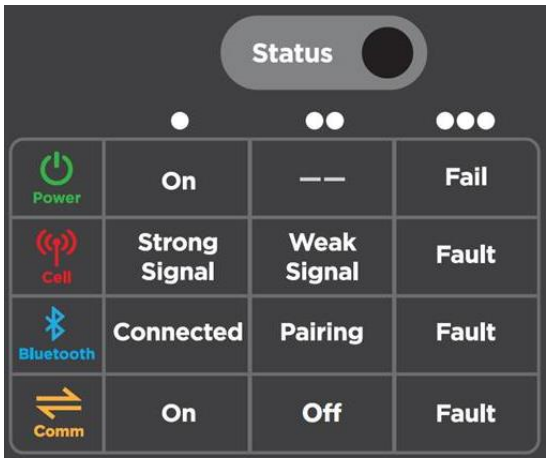

Figure 5: LED Pattern Definition

Table 2: LED Color/Blink Patterns

Module	Color	1 Blink	2 Blinks	3 Blinks
Power	Green	On	----	Fail
Cell	Red	Strong Signal	Weak Signal	Fault
Bluetooth	Blue	Connected	Pairing	Fault
Comm	Amber	On	Off	Fault

4 Channels

The Messenger 600 maps all monitored conditions into channels. Each channel has data storage and configuration parameters. Data storage holds current value, max/min values, and other run time data values. Configuration consists of user-settable parameters that define rules on how the data values are to be processed (as shown in the Channel Configuration – Type 9 section).

There is a set of predefined channels (numbers 1-4) and a set of Modbus channels (numbers 301-900). The Modbus channels are user configurable for reading/writing digital or analog values from a Modbus RTU slave device.

4.1 Defined Channels

The following channels are predefined in the Messenger 600. The wiring for some of these channels can be found in the figures in the **Error! Reference source not found.** section.

Predefined Channels

Channel Number	Channel Name	Type	Description
Start Digital Channels (data values are '0' or '1')			
1	Cell	Digital (System)	Status of cellular modem operation 1 = fault 0 = normal
2	Comm	Digital (System)	Status of any serial port used for communications with external device (MODBUS Master, Slave, or proprietary) 1 = fault 0 = normal
4	User DIN1	Digital	0 = On AC power 1 = On Battery power
Start MODBUS Channels			
301-900	Modbus	User Defined	Analog or Digital channels read from Modbus RTU Slaves

4.2 Channel Data

All values read from a Modbus slave device are continually updated and tested as defined by the configuration parameters. For each channel, based on its type (analog or digital), there is a basic set of data collected. For purposes of discussion, the term “not normal” is used to indicate an analog value that has violated a limit threshold or a digital value that does not match its configured ‘normal’ state.

Note: The RemoteIQ configuration should be made to match the configuration in the Messenger 600 hardware. Alarm notification is only configured in the RemoteIQ system.

5 Modbus

The Messenger 600 can be configured as a Modbus Master, Modbus Slave or both. This is done by setting the Mode of one of the available serial ports, 2 or 3. Port 2 supports RS232 and RS485; port 3 only supports RS232.

5.1 RTU Master

When configured as a Modbus RTU Master device, the Messenger 600 reads register values from Modbus Slave devices.

In addition to setting up a Modbus Master serial port, channels in the Messenger 600 must be configured to receive the values read from the slave device. As shown in the Channel Configuration – Type 9 section, define the Modbus Slave address, the register data type to read, the register number, how to interpret or scale the data being read, and any alarm limits.

6 Debug Menu

The debug menu is a text based menu system accessible via terminal emulation software running on a PC (such as Tera Term or HyperTerminal) or via a PC based utility that connects to the Messenger 600 OTA via TCP. Debug menus allow the user to view or change configuration parameters, view data values and history logs, see communications between CPU and attached peripherals, and to clear accumulated data values or logs.

The Debug Menu can be accessed via a serial USB or RS232 connection using a USB-RS232 converter and an RS232-RS485 converter. The default port settings are as follows: **57600 baud**, **8 databits**, **1 stop bit**, **no parity**.

Note: Once connected, press the 'Enter' key to see the Main Menu (example below).

```
________MessengerAB________
Site Name: Site Name
Model    : CP750/ANTX LVW3 MBUS J1939
Version  : AB v8.1.8D  02/26/20
Date/Time: 05/13/20  14:34:09
Asset ID : 357766090587705
GPRS Reg : Home
Server   : Primary
MSISDN   : 15772170946
Local IP : 10.208.242.239
Network  : LTE
CELL RSSI: 16 (18,15)
GPS      : No Fix
DataQue  : 9
PushQue  : 0
Output 1 : Off
Output 2 : Off
Output 5 : Off

1) View Config
2) Timers
3) Data
4) Events
5) Reports
6) Setup
7) Maint
8) J1939 Diags

Cmd =>
```

Most menu navigation commands are single alpha-numeric entries, no carriage return (Enter) required. However, when modifying configuration parameters, the configuration string entered must be followed by a carriage return (Enter).

7 Configuration

The Messenger 600 is a highly configurable platform with several methods in place to allow a user to read and/or modify the whole configuration or any part of it. All configuration parameters are stored in non-volatile memory.

Configuration parameters can be changed using the following methods:

- User modify via debug menu
- Transfer configuration file, using “XMODEM-1K”
- Transfer configuration file via FTP, initiated via debug menu

All configuration parameters are read/written using text strings, with commas to separate values in the string. This basic format is the same for all of the methods listed above. Some of the methods put a “wrapper” around the configuration line to aid in transport and decoding on the receiving end.

The basic configuration line format (referenced as CL) is described below:

CL = x,<i,>y,zzzz,y,zzzz,y,zzzz,…,y,zzzz

where:

- The commas are required as the delimiter between value fields
- *x* is a unique number identifying the configuration line type (e.g., Site=1, Serial=6, Channel=9, etc.)
- *<i,>* is an optional index used with specific configuration line types that have more than one element (e.g., there is more than one Serial Port and more than one Channel)
- *y* is a parameter code identifying the configuration parameter that follows
- *zzzz* is the configuration parameter (this parameter could be an integer, a floating point number or a text string, as defined by the parameter code *y*)

Table 3*: Configuration Line Types* identifies the available configuration line types and their type code.

Table 3**: Configuration Line Types**

Type Code	Reference	Description	Optional Index
1	Site	Messenger 600 global settings	None
6	Ports	Serial port parameters	Serial Port Number Range 1-4
9	Channels	Set scaling and alarm parameters for channels	Channel Number Range 301-900
12	Report Flags	Defines data types to include in reports to the host server	Channel Number Range 301-900
16	Date/Time	Set the Date/Time externally	None

7.1 Debug Configuration Commands

Users can manage the current configuration via the built-in Debug Menu system. From the main menu, select *Site Setup* (6). From *Site Setup*, select *User Input* (1). From the *User Input* prompt, the following read/modify commands are applicable.

7.1.1 Read Command

This is the read configuration command format:

255,x<,i><CR>

where:

- *255* is the read command
- *x* is the configuration line type code
- *<,i>* is an optional index that is a function of the line type code (see the Table 3: *Configuration Line Types* table)
- *<CR>* is a line-terminating carriage return

Some examples using the READ command:

Examples:	
255,1<CR>	Prints Site config
255,6,2<CR>	Prints Serial Port 2 config
255,6,255<CR>	Prints all Serial Port configs
255,255<CR>	Prints a Full System config

By issuing the “255,255” command, the user can capture a complete system configuration to a file. This file can then be used as the master configuration file. This master config file can then be modified and loaded back into the system or any configuration segment that needs to be reconfigured.

7.1.2 Reset Command

To reset a configuration to factory defaults, use the following command format:

256,x<,i><CR>

where:

- *256* is the reset command
- *x* is the configuration line type code
- *<,i>* is an optional index that is a function of the line type code (see the Table 3: *Configuration Line Types* table)
- *<CR>* is a line-terminating carriage return

Some examples using the RESET command:

Examples:	
256,1<CR>	Reset Site config
256,6,2<CR>	Reset Serial Port 2 config
256,256<CR>	Reset System config

	CAUTION
	THE RESET ALL COMMAND (256,256) SHOULD BE USED WITH CAUTION. THIS MAY RESET THE SERIAL PORT IN USE. COMMUNICATIONS WITH THE UNIT COULD BE LOST.

7.1.3 Global Command

The global command can be used to set the same parameter to the same value, within the same configuration type, for multiple indexes. The global command only works with Channel and Report Flags configuration types.

The global command format is:

257,x,i-j,y,zzzz<CR>

where:

- *257* is the global command
- *x* is the configuration line type code
- *i* is a required starting index
- *j* is a required ending index, greater than *i*

Note: The range of indexes from *i* to *j* is inclusive and by definition are sequential.

- *y* is a parameter code identifying the configuration parameter that follows
- *zzzz* is the configuration parameter (this parameter could be an integer, a floating point number or a text string, as defined by the parameter code *y*)

Some examples using the GLOBAL command:

Examples:	
257,9,301-320,6,1<CR>	For channels (9) 301-320 inclusive, change the decimal position (6) to 1
257,12,352-388,4,2<CR>	Change the report flags (12), for channels 352-388 inclusive, in the end of day report (4), to include data type 2 (2)

7.2 Site Configuration – Type 1

Parameters for Site Configuration			
Parameter Code	**Reference**	**Description**	**Default**
1	Daylight Saving	Used to adjust local time for daylight saving. Local time is used for timestamp of events and display of date/time in debug menu. 0 = disabled 1 = enabled	Enabled [1]
2	Time Zone	Defines local time zone for the Messenger 600. For display and event log timestamps. Range 0-7: 0 = UTC 1 = Atlantic 2 = Eastern 3 = Central 4 = Mountain 5 = Pacific 6 = Alaska 7 = Hawaii	Central [3]
3	Next Call Delay	This delay is enforced between successive attempts to connect and send data to host server. Range 1-32000 seconds	20
4-8		Reserved	
11	Modbus Poll Mode	This defines how the unit polls a MODBUS slave for registers. 0 = single register per poll 1 = multiple registers per poll	1
12	Modbus Scan Rate	This defines how often the Messenger 600 polls MODBUS slave devices. Range 0-3600 seconds 0 = no delay between successive polls	5

7.3 Serial Port Configuration – Type 6

The Messenger 600 has four serial ports (1-4). Port 1 is dedicated to the on-board cellular modem

⚠	**CAUTION**
	DO NOT MODIFY THE PARAMETERS OF PORT 1.

Port 2 is set to RS485 by the factory. Port 3 is set to RS232 by the factory.

Note: Not all parameters shown below apply to every port.

Parameters for Serial Port Configuration			
Parameter Code	**Reference**	**Description**	**Default**
1	Mode	Range 0-7: 0 = none **1 = debug (port 3)** 2 = MODBUS RTU Slave **4 = MODBUS RTU Master (port 2)**	Port Specific
3	Modbus Slave ID	This sets the Slave ID of the unit when the port mode is set to MODBUS RTU Slave. Range 1-247	126
4	Baud	Defines the port baud rate. Valid baud rates: 1200 2400 4800 9600 19200 38400 57600 115200	57600
5	Max Idle	Defines the period of inactivity before the active receive buffer is closed. Range 1-32000 ms	5
6	Response Timeout	Maximum time to wait for a response Range 1-60 seconds	2

Parameters for Serial Port Configuration			
Parameter Code	**Reference**	**Description**	**Default**
7	RS485 pre-Tx delay	Defines the duration of time between enabling the RS485 transmitter and starting transmission. Range 0-20 ms	2
8	RS485 post-Tx delay	Defines the duration of time after transmission of the last character to disabling the RS485 transmitter. Range 0-20 ms	4
9	Data Bits	Defines the number of data bits in the serial stream. Range 7-8	8
10	Stop Bits	Defines the number of stop bits in the serial stream. Range 1-2 **Note:** Due to errata in UART device, 2 stop bits are required for ports 3 and 4.	1
11	Parity	Defines parity for the serial stream. Range 0-2: 0 = none 1 = odd 2 = even	None [0]

7.4 Channel Configuration – Type 9

Parameters for Analog Channel Configuration (channel numbers 51-300 are predefined analog channels)			
Parameter Code	**Reference**	**Description**	**Default**
1	Name	A representative name by which to reference the channel ASCII Text, 30 characters max	Channel Specific
2	Mode	Defines operating mode of channel. 0 = disabled 3 = report digital on change of state, analog on delta violation	Disabled [0]
3	Alarm Delay	The continuous time the value must exceed a limit before it is considered to be in violation of that limit. Range 0-65535 seconds 0 = no delay time, immediate alarm	10
6	Precision	Digits to the right of the decimal point. Affects precision of value displayed and precision of value reported to host server. Range 0-8	0
7	Low Warning Limit	Low warning limit Floating point value – e.g., 15.2 Range -999999.0 to 999999.0 -1.0 disables limit	-1.0
8	Low Alarm Limit	Low alarm limit (should be less than low warning limit) Floating point value – e.g., 12.8 Range -999999.0 to 999999.0 -1.0 disables limit	-1.0
9	High Warning Limit	High warning limit Floating point value – e.g., 26.4 Range -999999.0 to 999999.0 -1.0 disables limit	-1.0
10	High Alarm Limit	High alarm limit (should be greater than high warning limit) Floating point value – e.g., 28.1 Range -999999.0 to 999999.0 -1.0 disables limit	-1.0

Parameters for Analog Channel Configuration (channel numbers 51-300 are predefined analog channels)			
Parameter Code	**Reference**	**Description**	**Default**
16	Alarm Output	A digital output channel that can be activated when this channel goes into alarm (referenced as channel number). 0 = disabled xxx = any MODBUS channel configured as a Write Coil channel	Disabled [0]
Channel numbers 301-900 are user-configurable MODBUS channels. The following parameters are specific to those channels when configured as type analog.			
20	Slave ID	Modbus ID of slave unit Range -1 to 255 -1,0 = disabled 1-227 = for direct connect slave devices 228-239 = reserved 240-255 = for peer to peer devices	-1
21	Register Number	Register number in slave to read/write Range 1 – 65535	1
22	Modbus Function Code	Defines the type of data register to read/write in the slave. **Note:** A single register is 2-bytes (16-bit); a double register is 4-bytes (32-bit).	None [0]

Parameters for Analog Channel Configuration (channel numbers 51-300 are predefined analog channels)			
Parameter Code	**Reference**	**Description**	**Default**
		0 – None 3 – Read Holding (16-bit) 4 – Read Input (16-bit) 6 – Write Holding (16-bit) 8 – Read Holding Double (32-bit) 10 – Read Holding as Float (32-bit) 11 – Read Packed Bits in Holding (16-bit, leave as aggregate) 12 – Read Packed Bits in Holding Double (32-bit, leave as aggregate) 16 – Write Multiple Holding (16-bit) 25 – Write Holding Double (32-bit) 26 – Write Float (32-bit) 27 – Read Double Precision Holding (32-bit) 31 – Read Input Double (32-bit) 33 – Read Input as Float (32-bit) 34 – Read Packed Bits in Input (16-bit, leave as aggregate) 35 – Read Packed Bits in Input Double (32-bit, leave as aggregate) 36 – Read Double Precision in Input (32-bit)	
23	Weight	Used to scale register value when not in engineering units Scaled = (weight * register value) + offset Floating point value – e.g., 0.25 Range -999999.0 to 999999.0	1.0
24	Offset	Used to scale register value when not in engineering units Scaled = (weight * register value) + offset Floating point value – e.g., -25.0 Range -999999.0 to 999999.0	0.0
25	Endian	When reading double registers, this defines the byte ordering in the register pair. Example: Double register value (hex) = 0x12345678 If order is Big Endian: Register x = 1234 Register x+1 = 5678 If order is Little Endian: Register x = 5678 Register x+1 = 1234	Big Endian [1]

Parameters for Analog Channel Configuration (channel numbers 51-300 are predefined analog channels)			
Parameter Code	**Reference**	**Description**	**Default**
		0 = Little Endian 1 = Big Endian	
26	Aggregate Mask	When reading bit packed registers (function codes 11, 12, 34 and 35), use this value to mask unwanted bits. A 0 in a bit position clears that bit in the value read. For the remaining bits, when a change from 0 to 1 is detected, an alarm event is generated. If a change from 1 to 0 is detected, a return to normal event is generated. This makes it possible for an alarm and normal event to be generated for the same channel at the same time. Hexadecimal value – e.g., ffff1afc Range 0-ffffffff	ffffffff
27	Signed	Indicates that the register contains a signed value 1 = register signed 0 = register not signed	Not Signed [0]

Parameters for Digital Channel Configuration (channel numbers 1-50 are predefined digital channels)			
Parameter Code	**Reference**	**Description**	**Default**
1	Name	A representative name by which to reference the channel ASCII Text, 30 characters max	Channel Specific
2	Mode	Defines operating mode of channel. 0 = disabled 1 = Status Only (value is monitored, no testing of value against limits) 2 = Report on Alarm (value is monitored and tested against limits)	Status Only [1]
3	Alarm Delay	The continuous time the value must exceed a limit before it is considered to be in violation of that limit. Range 0-65535 seconds 0 = no delay time, immediate alarm	10
6	Normal (Idle) State	State of input when condition being monitored is normal 0 – open or > 3.0 V 1 – closed to ground	Normally Open [0]

Parameters for Digital Channel Configuration (channel numbers 1-50 are predefined digital channels)			
Parameter Code	**Reference**	**Description**	**Default**
16	Alarm Output	A digital output channel that can be activated when this channel goes into alarm (referenced as channel number). 0 = disabled xxx = any MODBUS channel configured as a Write Coil channel	Disabled [0]
Channel numbers 301-900 are user-configurable MODBUS channels. The following parameters are specific to those channels when configured as type digital.			
20	Slave ID	Modbus ID of slave unit Range -1 to 255 -1,0 = disabled 1-227 = for direct connect slave devices 228-239 = reserved 240-255 = for peer to peer devices	-1
21	Register Number	Register number in slave to read/write Range 1 – 65535	1
22	Modbus Function Code	Defines the type of data register to read/write in the slave. 0 – None 1 – Read Coil 2 – Read Status 5 – Write Coil 7 – Read Packed digitals and select individual bit (16-bit) 9 – Read Packed digitals and select individual bit (32-bit) 30 – Read Packed Bit in Input (16-bit) 32 – Read Packed Bit in Input Double (32-bit)	None [0]
23	Bit Number	When reading holding registers as packed digitals, this defines the bit number to monitor. Range 1-32 where 1 is LSB	1

7.5 Report Flag Configuration – Type 12

There are several report types that can be generated by the Messenger 600, each with a unique trigger mechanism. Report flags are used to enable specific channel data to be included in specific report types.

Parameters for Report Flag Configuration			
Parameter Code	**Report Type**	**Description**	**Trigger**
3	Standard (PER)	Include all channels that have a PER report flag set. By default, all channels.	Standard Report Interval (time based only)
4	End of Day (EOD)	Include all channels that have an EOD report flag set. By default, no channels.	Midnight (UTC Time)
5	Demand (DEM)	Include all channels that have a DEM report flag set.	Request from the web server

Defining which channels to be included in which report is step 1. Step 2 is to decide what type of data from that channel to include. Each channel maintains a basic set of data based on the channel type, either analog or digital (see the Channel Data section).

Channel Type		Report Type	Data Types Reported (i.e., Report Flag Setting)
Modbus	Analog	Standard (PER) Demand (DEM)	0 = none 1 = current value 2 = current value + current max/mix
		End of Day (EOD)	0 = none 1 = current value 2 = current value + daily max/mix
	Digital	Standard (PER) End of Day (EOD) Demand (DEM)	0 = none 1 = current value 2 = current value
	Aggr	Standard (PER) Demand (DEM)	0 = none 1 = aggregate value (hex) 2 = aggregate value (hex)

7.6 Date/Time Read/Write – Type 16

This method of setting the RTC in the system is always available but should be used with caution. When GPS or cellular is available, setting time this way can generate unwanted side effects. There is not a real Date/Time configuration. This provides an alternate method for setting date/time OTA.

Parameters for Date/Time Configuration		
Parameter Code	**Reference**	**Description**
1	Time	Enter as hours, minutes and seconds. Format = hhmmss Use 2 digits for each
2	Date	Enter as month, day and year. Format = mmddyy Use 2 digits for each
3	Time Zone	Local time zone Range 0-7: 0 = UTC 1 = Atlantic 2 = Eastern 3 = Central 4 = Mountain 5 = Pacific 6 = Alaska 7 = Hawaii
4	Daylight Saving	To correct local time if daylight saving is observed. Range 0-1: 0 = daylight saving not active 1 = active

8 Technical Support

For remote and communication control systems support, parts and repair, or technical support, visit us online at: www.cattron.com/contact.

Due to continuous product improvement, the information provided in this document is subject to change without notice.

Cattron Support
For remote and communication control systems support, parts and repair, or technical support, visit us online at:
www.cattron.com/contact

Cattron North America Inc., 655 N River Rd NW, Suite A, Warren, OH 44483

9M02-7000-A301-EN
Version B