

TOTAL ENGINEERING SERVICES TEAM, INC.

RTU / SCADA SYSTEMS

MULTI DROP COMMUNICATIONS

Document No. 1170-01

P.O. Drawer 1760
671 Whitney Ave.
Gretna, La. 70054
(504) 368-6792

Contact Arthur Zatarain, PE
via www.artzat.com
For information on this document

*This document is (C) copyright 1993 by
Total Engineering Services Team, Inc., Gretna, La. USA.
All rights are reserved.*

TOTAL ENGINEERING SERVICES TEAM, INC.

RTU / SCADA SYSTEMS

MULTI DROP COMMUNICATIONS

- 1.0 - INTRODUCTION
- 2.0 - DCD AND ONLINE STATUS
- 3.0 - SOFTWARE SETUP
- 4.0 - MODEM MONITOR
- 5.0 - MULTI DROP IDs
- 6.0 - CONFIGURING A LINK
- 7.0 - MULTI DROP SESSIONS
- 8.0 - BROADCAST MODE
- 9.0 - SET MD DELAY COMMAND
- 10.0 - TERMINAL MODE
- 11.0 - CHECKSUM ERRORS
- 12.0 - REFERENCE SECTION
 - 12.1 - FILE SEGMENTS
 - 12.2 - MULTI DROP SETTINGS
 - 12.3 - LINK SETTINGS

1.0 INTRODUCTION

SCADA systems require the use of some type of communications medium to send data from one unit to another. TEST's SCADA system supports the use of telephones, radios, and direct wire connections to transmit data. The normal process of communication between two units involves a dial out by one unit, a connection with another unit, the transmission of data, and finally a disconnect. This process describes a point-to-point mode of operation where only two devices at a time can communicate. Once a connection is made between two units the communications path becomes blocked and can not be used by other units. Not until the current session is over will other units be able to transmit data.

TEST's RTU/SCADA system also supports a multi drop mode of operation where several units share a single communications path. In this mode any unit can communicate with any other unit as long as the communications path is clear. Unlike the point-to-point mode, the communications path does not remain blocked for an entire session. Instead, the communications path is blocked only when a unit is transmitting. At the end of each spurt of data the path becomes clear. At that time any unit can gain control of the path and transmit its own message to any other unit.

Multi drop communications represent a peer-to-peer relationship between units rather than a master/slave relationship. By this we mean that each unit attached to the communications path is equally capable of communicating with any other unit attached to the path. There is no master unit responsible for communication with remote units. Instead, communications can occur at any level, whether it be between a Host and a Host, a Host and an RTU, or simply between two RTUs.

2.0 DCD AND ONLINE STATUS

The TEST RTU/SCADA system uses the serial port of a PC to connect a modem or other data transmission device so that it can send information to a distant unit. The unit on the other end must have a similar device to receive the information and feed it into the computer. For point-to-point communications the Data Carrier Detect (DCD or CD) line is used to determine the online status of the serial communications. Nothing can be transmitted unless the DCD is present. However, for multi drop communications the presence of a DCD will prohibit a unit from transmitting. An explanation of how the DCD is used for all types of communications is given below.

For telephone communications in point-to-point mode a HAYES type modem is used. The DCD is used to reflect the connected state of a modem with another modem. When a modem answers a call from another modem the DCD becomes true at the receiving modem. The calling modem will then detect the carrier from the other end and turn on its DCD. As long as each modem detects an audio tone carrier from the other modem the DCD will be true and the two computers can communicate.

For radio communications in point-to-point mode a special type of modem called a Packet Radio Controller (PRC) is used. The PRC is a little more

complicated than a phone modem because of the "half-duplex" nature of radio transmissions. The PRC does not send out a constant carrier tone like the HAYES modem. In the case of the PRC, the DCD status represents a logical connection to another PRC rather than a physical connection. The status of the connection is determined by radio tone handshaking from the modems themselves. When one PRC calls another they exchange codes and the DCD is established. Until the PRCs disconnect, the DCD will be true even though no actual carrier is being transmitted. The disconnect will occur after a specified period without any activity or when the computer commands the PRC to "hang up".

Direct connect systems for point-to-point communications are fairly simple. Direct connections can be made using a 5 wire null-modem cable. This simply crosses the TD and RD lines from each of the computers and also switches the RTS and DCD lines to provide flow control. The fifth wire is used for the Signal Ground. When a direct connect system "dials" out it simply raises its RTS line and keeps it on for the duration of the session. The other unit will "see" the signal and its DCD input which will cause it to turn on its own RTS line. Therefore, direct connect systems use their RTS lines to activate the DCD inputs of the other computers. As long as the DCDs are present the systems can communicate. Again, a disconnect will occur after a specified period without any activity or when the computer performs a controlled "hang up".

For multi drop communications the DCD line is not used to determine the online status of a unit. The DCD is only used to indicate that a unit is transmitting data and that the communications path is temporarily blocked for use by other units. The online status of a multi drop system is determined by a logical connection rather than a physical one. This is similar to the logical status used by Packet Radio Controllers.

There are two ways that a unit's task can go online when operating in multi drop mode. First of all, a task will logically be placed online whenever an attempt is made to call out. Second, a task will go online whenever it receives a message from the communications line that is intended for that unit. Once online, a task will remain online until a specified amount of time has elapsed without any activity for that unit or a BYE command is processed which forces a "hang up".

In addition to being online, a unit must have a clear communications path in order to transmit. All tasks operating in multi drop mode are continuously monitoring the DCD line of the communications path. If a task wishes to transmit and a DCD is present, the task will have to wait. The amount of time that a task will wait for the DCD to clear is controllable for each task by using the

SET MD DELAY xx xx

command. If the DCD is not cleared before the wait period expires the attempt to transmit will fail and the task will generate the message

COULD NOT SEND

A failure can also occur if the DCD is cleared but another task gains control of the communications path and begins to transmit. The original period that a task will wait for the DCD to clear will not be reset when the DCD first clears. It will continue to count down until it either expires or the task finally gains control and

begins to transmit.

When the communications path is clear (no DCD is present) and a task wishes to transmit it will begin by turning on the RTS line. This will cause the modem to turn on a carrier tone that will be detected by the DCD of all other units on the line. A slight delay will occur between the time the RTS line is turned on and the transmission starts. This allows modems and radios to get their carriers turned on before the actual data starts down the line. This also allows the squelch to clear at the receiving radio or modem. This delay is also controllable for each task by using the

```
SET MD DELAY xx xx
```

command. (The SET MD DELAY command is explained below.) At the end of the data burst the RTS line is kept on for one additional system tick. This will allow all characters to get transmitted out of the UART without chopping any off.

3.0 SOFTWARE SETUP

Setting up a system for multi drop communications is not much different from the setup for point-to-point communications. The main difference is that MULTIDROP must be specified as the communications media. The media to be used by each communications port must be defined when the SCADA program is started. The COMM statement in the main configuration (DAT) file is used to define the type of media used by each port. For example, the line

```
COMM 1 1200 MULTIDROP N 8 2
```

could be used in the main configuration file to setup comm port 1 for multi drop communications with a baud rate of 1200, no parity, 8 data bits, and 2 stop bits.

Normally, the media is setup one time during startup and never needs to be changed. However, in the event that the media does need to be changed once the program is running the SET MEDIA command could be used. The available keywords to be used with this command are PHONE, RADIO, KEYBOARD, MULTIDROP, and NONE. For example, the command

```
SET MEDIA MULTIDROP
```

could be used to change the communications media of a comm port to multi drop at any time.

Other considerations for using multi drop communications can be found under the sections called MODEM MONITOR and CONFIGURING A LINK. Another point worth mentioning is that the HAYES and PRC commands required for phone and radio communications in point-to-point mode are not necessary for multi drop communications. The modems used for multi drop communications are not programmable and do not perform independent processing like the more intelligent HAYES and PRC devices. Multi drop modems only serve to convert and detect serial data to and from audio tones. All actions by these modems are controlled by the RTS line of the serial port.

4.0 MODEM MONITOR

The Modem Monitor device is a TEST INC. designed and manufactured device intended to assist in unattended operations. This device is used to monitor system activity and perform an automatic reset of the computer if a malfunction occurs in the processor. Although not a communications device itself, it is included here because it ties into the serial port.

The TEST SCADA program uses either the RTS (pin 4) or DTR (pin 20) as a control line for the Modem Monitor device (default is RTS). The software toggles one of these lines once per second to activate the watchdog timer in the Modem Monitor. If the line does not toggle within a specified time period, the Modem Monitor will reset the computer. Refer to the SYSTEM DESIGN manual for more information about the Modem Monitor.

On multi drop systems the RTS line is used as a special signal to control the carrier output of a modem. The software will turn this signal on prior to transmission and will turn it off when the transmission is complete. Because the RTS line is used to control the CD (or DCD) of the modem and the watchdog timer of the Modem Monitor, a conflict results. This conflict can be resolved in one of two ways.

First of all, the RTS can be used to control both functions as long as the computer will transmit often enough to prevent the watchdog timer from timing out. In this case the

MONITOR OFF

command must be used to stop the program from toggling the RTS line once per second. This will allow the Modem Monitor to still use the RTS line without interfering with CD of the modem. Also, the turning on and off of the RTS line during data transmissions will provide the necessary toggling to prevent the modem monitor from resetting the computer.

Second, the control line for the Modem Monitor could be changed to the DTR line. This option should be used on systems where transmissions will not occur often enough to prevent the watchdog from timing out. Changing the control line for the Modem Monitor can be done using the

MONITOR DTR

command. The MONITOR OFF or MONITOR DTR command can be included in the START0.RTU file which automatically gets processed when the program is started.

5.0 MULTI DROP IDs

In multi drop operation several units share a common communications path. This means that every message transmitted by every unit will be "heard" by all other units. However, each message transmitted is usually intended for only one particular unit. This means that one unit should process the message and all other

units should ignore it. Multi drop IDs are used to keep track of which units should process which messages.

Each task that is used for multi drop communications has two IDs associated with it. These IDs are included as part of a message header that is attached to each line transmitted by the task. The first ID is used to identify the intended destination of the message. The second ID is used to identify the source of the message. The format of the header is

DestinationID/SourceID/

This header is automatically inserted in the message packet of the sending unit between the leading Control-A character and the command sequence number. For example, consider a task with the multi drop ID GA-HOST that is transmitting a message to another unit whose communications task has the multi drop ID GA343A. The actual message package that is sent might appear as follows:

GA343A/GA-HOST/1A0messagechecksum

where 1 is the number of the block being sent and 0 is the number of the block being acknowledged.

By default, each task's multi drop ID is set to the name of the system RTU. This name is defined during startup by using the NAME statement in the main configuration (DAT) file. If this command is not used in the main configuration file the name SYSTEM will be used. Any task can change its multi drop ID at any time by using the

SET MD ID xxx

command where xxx is the name of the new ID. The maximum length of a multi drop ID is 8 characters. The

DUMP TASK x

command can be used to display a task's multi drop ID. The x is replaced with the number of the task for which to display a list of information.

When a task transmits a message the source ID in the multi drop header is simply taken from the task's multi drop ID. However, the destination ID can come from one of two places, depending on what caused the message to be transmitted. If the message being transmitted is a response to a message from another unit, the destination ID will contain the ID of the unit that sent the original message. On the other hand, if the message being transmitted is the result of an activated link, the destination ID will contain the phone number or radio call sign of the link. In the case of multi drop communications, the phone number or radio call sign of a link is actually the multi drop ID of the unit to receive the outgoing messages.

6.0 CONFIGURING A LINK

Setting up a link for multi drop communications is very similar to setting up a link for point-to-point communications. All link parameters can be programmed using the `CONFIG LINK x` or the `PROG LINK x` command. The only difference for setting up a multi drop link is specifying the number to call. Normally, a phone number or radio call sign must be given. This is the actual number that must be called in order for two units to establish a connection and communicate.

However, in multi drop mode there is no actual call out and connect process. Since all units are connected to a common communications path, it is more a matter of identifying the unit to "talk" to rather than calling it up and establishing a connection. For multi drop communications, the phone number or radio call sign that must be specified is actually the multi drop ID of another unit's communications task.

For example, consider a Host and two RTUs that are setup to use multi drop communications. Assume that the Host system is called GA-HOST and the two RTUs are called GA343A and GA343B. Each unit would require a `comm` command in its main configuration (DAT) file to set up a comm port for multi drop communications. Each unit would also require an RTU task to be defined which would use the comm port for communications. By default, the multi drop ID for each of these RTU tasks would be the system name.

At the Host two links could be setup to communicate with each RTU. The phone numbers specified for each link would be GA343A and GA343B. At the GA343A RTU, two links could be setup to communicate with the Host and the other RTU. The phone numbers for each link would be GA-HOST and GA343B. The GA343B RTU could also contain two links setup in the same manner. This would allow any unit to be able to activate a link and communicate with any other unit. For example, the first link at the Host could be setup as follows and used to communicate with the GA343A RTU.

LINK 1 OF 4

Link Description **MDrop Link to GA343A**
Number of Task Using This Link 1
Phone Number or Radio Call Sign **GA343A**
Seconds to Wait For a Connect Before Giving Up 60
Maximum Number of Tries to Call Out When an Attempt Fails 5
Number of Tries to Rapidly Call Out When an Attempt Fails 2
Seconds to Wait Between Call Out Attempts After Exceeding Rapid Tries 120
Maximum Number of Tries to Call Out Without Resetting Counter to Zero 0
Is This a Host? Y
Alternate Link -1

<F2> Accept • <PgUp> Previous Link • <PgDn> Next Link • <Esc> Cancel

NOTE: If a task's multi drop ID is changed using the `SET MD ID xxx` command, any link set up on another unit which is used to communicate with that task will also have to be changed to the new multi drop ID. The phone number of a link must specify the multi drop ID of the task it will communicate with, which is not necessarily the name of the unit it will communicate with.

7.0 MULTI DROP SESSIONS

To begin communications in multi drop mode a link must be activated. This can be done by pressing the **F5** key, by using the `LINK NOW` command, or by using some other link action. When a link is first activated the task that is responsible for processing the link will immediately go online. If a `CONNECT` file exists for that task it will be processed at that time. The program will then process a `LINK` file for that link if one exists. If the link file does not exist, the computer will process its own default commands depending on whether the unit that initiated the communications is a Host or an RTU. The default command for a Host unit is `BLOCK READ DOWNLOAD` and the default command for an RTU is `READ DOWNLOAD`. Either the `LINK` file or the `DOWNLOAD` file will be responsible for processing the commands necessary to transfer the data from one unit to the other.

Most commands that are transmitted between two units require an acknowledgement of some kind. This can be either a response to a command or a simple `ACK`. For example, a `SCAN` command requires a response in the form of a `DATA` command. Other commands simply require an `ACK` which informs the

sending unit that the message was properly received by the receiving unit.

When a message is transmitted that requires an acknowledgement the task that sent the message will remain online and wait for the acknowledgement from the other unit. Although the acknowledgement should be almost instantaneous, the DCD will temporarily be cleared between the time the message is sent and the time the acknowledgement is returned. During that time any unit sharing the communications path is free to transmit.

While waiting for an acknowledgement a task might see messages go by that are transmitted by other units. Some of those messages could contain the multi drop ID of the task waiting for the acknowledgement. This means that other units could be trying to communicate with a task while it is waiting for an acknowledgement from another unit. The task waiting for the acknowledgement will not be able to process any incoming messages other than a message from the unit it is waiting to hear from. Once the acknowledgement is received the task will be able to process other incoming commands that contain its multi drop ID.

What this means is that it is possible for a third unit (or more) to intervene during the communications of two other units. There are only two limitations to processing messages coming from more than one unit at a time. First of all, as already mentioned, no unit can communicate with a task that is in the process of waiting for an acknowledgement or attempting to send an acknowledgement to another unit.

Second, each incoming message must contain a higher block number than previously processed message. If an incoming message contains a block number less than the last processed message, a simple ACK will be returned because the task will assume that it has already received and processed that message. The reason for this is that each task only keeps a single sequential block counter for all incoming messages. The source of the messages does not matter because a separate block counter is not kept for each unit sharing the communications path.

An additional point to take note of regarding incoming block numbers is the special case of block number 1. Except for the first incoming message of a session, a task will reset its incoming and outgoing block counters to zero when it receives an incoming message with a block number of 1. Normally, incoming messages start with a block number of 1 and are incremented by 1 for each additional incoming message. When a message comes in with a block number less than the current block count the message is ignored and an ACK is returned. However, if the incoming block number is 1, the message will get processed and the block counters will get reset. This feature allows a task to automatically reset the block counters of another unit without having to process a `BLOCK RESYNC` command. This is common for situations where a task processes a file that continuously loops sending constant updates to another unit. At the end of the loop the file could contain a `RESYNC` command to reset the block counters locally. When the next message is sent to the other unit it will contain the block number 1. When the other unit receives a message with the block number 1 it will reset its own block counters.

Block counters are also reset for a task each time the task performs a disconnect. Upon going offline a task's block counters are reset to 0. Each time the task goes online again it will start its outgoing block count at 1.

8.0 BROADCAST MODE

Communications in multi drop mode begin when a link is activated and one unit attempts to transmit data to and receive data from another unit. The unit to be contacted is determined by the multi drop ID that is specified for the phone number or radio call sign of the link that was activated.

During normal multi drop communications a message is sent out by one unit. That message is seen by all units sharing the communications path. However, only one task monitoring the path should have the same multi drop ID that is contained in the message. That task is responsible for processing the message and returning some type of response to the sending unit.

The RTU/SCADA program also supports another mode of operation for multi drop communications. This mode, known as BROADCAST MODE, occurs when the multi drop ID (phone number or radio call sign) of an activated link is 0. By specifying 0 as the multi drop ID for a link, instead of an actual task's multi drop ID, a single unit is able to communicate to all other units on the path at the same time.

Normally, each unit monitoring the communications path will only process a message that contains its multi drop ID in the message header. However, each task will also process any incoming message that has a multi drop header of 0 as the destination ID. This allows a single command to be sent out by one unit that will get processed by all other units.

Although this seems simple enough, there is one catch to broadcast mode. When a task receives a message in broadcast mode it will process the message but never return any type of reply to the sending unit. This means that the sending unit is never sure of what units received the message properly and processed it and which ones did not.

This may seem like an unreliable mode of communication at first, but when used properly it poses no real problem. Broadcast mode is designed to be used in situations where one unit sends the same group of commands to all other units over and over again in a continuous loop. Therefore, if a message does not properly get transmitted to some units it shouldn't matter. It should only be a matter of seconds before the same command will be transmitted again.

9.0 SET MD DELAY COMMAND

The SET MD DELAY command is used to set the amount of time that a task will wait for the DCD to clear in order to transmit. This command is also used to set the amount of time that a task will wait between the time it turns on its RTS and the time it begins to transmit. The format of the command is

```
SET MD DELAY dcd_wait keyup_delay
```

The time to wait for the DCD to clear is specified in seconds and the default value is 5 seconds. The keyup delay is specified in system ticks and the default

value is 18. At 36 system ticks per second, the default value of 18 is equivalent to a half of a second. As an example, the command

```
SET MD DELAY 8 36
```

could be used to set a task's DCD wait period to be 8 seconds and its delay between keyup and data transmission to be 1 second.

The DCD wait period and the keyup delay period are independently set for each task. This means that the SET MD DELAY command will affect only the task that processes the command. A convenient place to use this command (if it is necessary at all) is in a task's START file (START1.RTU, START2.RTU, etc).

The SET MD DELAY command can be used without any parameters to display the current settings for a task. Again, the task that processes the command will be the task that the settings are displayed for. The "watch" will have to be turned on to actually see the current settings displayed for any task other than task 0 (the local CRT). For example, the following sequence of commands could be used to display the current multi drop delays for task 1.

```
WATCH 1  
FORCE 1 SET MD DELAY  
WATCH OFF
```

10.0 TERMINAL MODE

In normal point-to-point communications the program can operate in two modes, RTU mode and Terminal (Human) mode. In terminal mode simple text messages are transmitted from one computer to another. For phone and direct connect communications characters are sent one at a time from one computer to the other. When the receiving computer receives a carriage return it knows that the line is complete and ready for processing. For radio communications the characters are buffered in the Packet Radio Controller and sent as a single block when a carriage return is entered. The PRC is responsible for keying up the radio and transmitting the message.

In RTU mode the same type of messages are transmitted as in Terminal mode but each message is automatically packaged with special codes and sent as a single block. These special codes are used to control message sequencing and detect errors in data transmissions. When the receiving computer receives a message it will perform all the proper error checking before processing the command. This is the normal mode of operation for all systems.

When using multi drop communications only the RTU mode of operation is available. In RTU mode the program will automatically package a message with all of the special error checking codes, wait for the DCD to clear, turn on the RTS, pause for the keyup delay, transmit the message, wait an additional system tick, and finally clear the RTS. In this case the program is responsible for keying up the modem prior to transmitting. The modem does not take care of keying itself up as it does for radio communications in point-to-point mode.

The Terminal mode of operation will not work for multi drop communications because the program does not key up the communications device each time a character is entered nor does it buffer all characters until a carriage return is entered. When attached to a task that is used for multi drop communications the *`TERMINAL` command can be used to put the task into Terminal mode. As each character is typed it will immediately be sent to the communications device to be transmitted. Since there is no buffering and no key up the characters will simply be lost rather than transmitted.

11.0 CHECKSUM ERRORS

As already mentioned, each message that is transmitted is automatically packaged with special codes before it is sent. These codes are used to check the sequence of incoming and outgoing messages as well as detecting any problems with the actual transmission. To check the validity of a data transmission a special checksum code is attached to the end of every outgoing message. This checksum is calculated from the actual character string that is being transmitted.

When a task receives an incoming message it will calculate its own checksum based on the character string it received. If the character string is scrambled during the transmission and what is received is not actually what was sent, the checksum will catch the error.

Any time a task receives a message and detects a checksum error the message will not get processed. In normal point-to-point mode communications the receiving task will send a NAK to the sending task to inform it of the error. The sending task will immediately attempt to send the message again.

For multi drop communications the process is a little different. First of all, if the message gets scrambled during transmission it might contain an invalid multi drop ID when it is seen by the tasks monitoring the communications path. In this case no task will process the message, the sending task will time out waiting for an acknowledgement, and finally the sending task will attempt to resend the message.

Another possibility is that the message gets scrambled during transmission but the multi drop ID of the message still matches the ID of one of the monitoring tasks. Again, no task will process the message and no NAK will be sent back to the sending task. This will also cause the sending task to time out waiting for an acknowledgement and finally attempt to resend the message. The reason the task does not return a NAK (as is done in point-to-point mode to force an immediate resend) is because there is no way to be sure that the message was originally intended for the task whose multi drop ID the message contained.

12.0 REFERENCE SECTION

This section provides a quick reference to things that should be checked when using multi drop communications. This reference is actually a summary of things that must be properly set to have multi drop communications operate correctly.

12.1 FILE SEGMENTS

A `COMM` statement must be included in the main configuration (`DAT`) file to setup a port for multi drop communications. To set up comm port 1 for multi drop communications the following command could be used:

```
COMM 1 1200 MULTIDROP N 8 2
```

The `SET MEDIA` command could be used to guarantee that a task is setup for multi drop communications. This command is usually included in the `START` file for each task that is used for communications. For example, the command

```
SET MEDIA MULTIDROP
```

could be included in the `START1.RTU` file to make sure task 1 is setup for multi drop communications. The `START` file for each task is automatically processed when the task is started.

12.2 MULTI DROP SETTINGS

The only real settings for multi drop communications are the amount of time that a task will wait for the DCD to clear in order to transmit and the amount of time that a task will wait between the time it turns on its RTS and the time it begins to transmit. Both of these settings are controlled by the `SET MD DELAY` command. Although the default settings a 5 seconds (DCD wait) and 1/2 second (keyup delay) are probably fine, it might sometimes be necessary for these values to be changed.

For example, if it is common for the communications path to be very busy at times because of several units trying to transmit simultaneously, it might be necessary to increase the DCD wait period. This will give a task a little more time to wait for the DCD to clear before giving up. It might also be necessary in some situations to increase the amount of time that a task will wait between turning on its RTS and transmitting. To double the default values for these delays the command

```
SET MD DELAY 10 36
```

could be used. A common place to use this command is in the `START` file of the task whose delays you want to change. This will allow the command to automatically be processed each time the task is started.

12.3 LINK SETTINGS

When setting up a link for multi drop communications you must be certain that the phone number (or radio call sign) of the link contains the multi drop ID of the task that you wish to communicate with. Remember, each task has its own multi drop ID that is set to the name of the system RTU during startup. Although this is probably fine, it might sometimes be desirable to change the ID of a task from the default value. This can be done by using the

`SET MD ID xxx`

command. Again, the common place to use this command is in a task's START file because this command must be processed by the task whose ID you want to change.

If the multi drop ID of a task is changed at one unit all links at other units that are used to "call up" and communicate with that task must be changed. A communications session will be unsuccessful if the activated link's phone number does not match (don't worry about upper or lower case) the multi drop ID of the task at the other end.