Quest Controls, Inc.
Monitoring, Control & Test Solutions for Telecom

TELSEC® ESB

Programmer's Manual

Quest Controls, Inc.
208 9th Street Dr. West
Palmetto, FL 34221
www.questcontrols.com

Phone: (941) 729-4799
Fax: (941) 729-5480
Email: customerservice@questcontrols.com

Rev 1.0 02/06/08

Table Of Contents

Table Of Contents

Table Of Contents

Table Of Contents

Chapter 1 – Introduction

Congratulations on the purchase of your new TELSEC® ESB product! The TELSEC® ESB is a state-of-the-art electronic monitor and controller providing you with an integrated solution for the facility management of your remote sites such as CEVs, CUEs, Shelters/Huts, Cabinets, Customer Prem Sites, Central Offices, Switching & Data Centers, and Head-Ends. Acting as your smart eyes and ears at the remote site, the TELSEC® ESB is capable of performing your HVAC Control and Monitoring, Environmental Monitoring and Battery Monitoring, Generator Monitoring, Telephony Equipment Monitoring, Tower Light Monitoring, and Door Access Control --- all in ONE proven product.

This Programmer's Manual is intended to provide you with the information you need to completely program the TELSEC® ESB for your application. Please refer to the **User's Manual** and **Installation Manual** for information regarding operating and installing the system. Contact us at Quest if you have product questions or suggestions to improve this Manual.

Chapter 2 – Basic Programming Commands

2.1 General Information

The TELSEC® uses a control program called "Equation Language" for developing control strategies in addition to the standard ON/OFF discrete alarm monitoring which is handled with a single define statement. The building blocks for this system are called points. A point can be any input, output, or control unit. Each point has a name associated with it. A program is developed by defining these points and setting certain constraints around them. The TELSEC® has been designed to enable a person with no experience in programming to easily learn the system. This section will show how to define all the points in the system. Defining a point tells the TELSEC how the point is to be used when system uses it. I.e. reading inputs in a certain engineering units, how an output operates etc.

2.2 Command Syntax

This section outlines the syntax for defining the TELSEC's® points through the modem, local communication port or Telnet. This is a reference section. Each command will be listed and then each command component will be explained in detail. If defaults exist for a specific command or point they will be shown here. For the actual programming of the TELSEC®, see Chapter 3 – Equation Language.

2.2.1 Conventions:

Text shown in this TYPEFACE contain commands that are sent to the TELSEC®. Optional text is shown in brackets, `[text]`. If the user can enter one command from a list, greater-than/less-than symbols denote the list, `<list>`. These same conventions are used with the TELSEC®help prompts.

All programming must be ended by pressing the ENTER, RETURN or semicolon (;) key. If a mistake is made during input, use the backspace character to erase or press the escape (ESC) key to abort.

2.2.2 A Word about Names:

The TELSEC® associates an eight-character, alphanumeric name with every point. The name must start with an alpha (A-Z) character followed by up to seven alpha-numeric (A-Z,0-9) or special characters. The special characters are %, &, and _. The TELSEC® will not recognize a space within a name. The TELSEC® has default names for all points but we recommend users assign their own names. For example, the TELSEC® has default name of UIN001 for UIN.1. This name does not provide much information for this point. Let's say this point is a temperature sensor input for the outside air temperature. If this point is named OUTAIR, it will have much more meaning in your programming.

2.2.3 HELP Command:

Issuing the HELP or ? command alone will present the user with a list of available KEYWORD entries. A KEYWORD entry is defined as any command that starts a TELSEC® programming line. The available KEYWORD list is:

DEFINE	CLEAR	REVIEW	LIST	NAME	SET
REMOVE	BYPASS	COPY	HELP	HANGUP	SEARCH

A user must start a line of programming with one of these KEYWORDs. You can see specific help on a KEYWORD by entering the KEYWORD followed by the ENTER key. Further help levels can be seen by entering the KEYWORD followed by a point type.

2.2.4 Short Cuts

All commands can be abbreviated to the first three characters command from another. For example the REVIEW command can be shortened to REV and the DEFINE command can be shortened to DEF.

2.3 DEFINE Command:

The DEFINE command begins all point definition programming. By defining a point, the TELSEC® is programmed as to how that specific point will operate. Each of the following point types can be defined:

```
UIN  RLY  AOP  DAT  TOD  EQU  VAR  SPT  DOR  ANM  MSG  RTU  KEY
BUS  CARD
```

2.3.1 DEFINE ANM:

The TELSEC® has the ability to send alarms or page up to four phone numbers. When an alarm (or clear) occurs, the system will use all alarm numbers that are active at that time. Alarm numbers that are not active due to a TOD qualifier will not be used. Additionally in the UIN definition for digital alarms or equation alarming, you have the option to specify which alarm numbers to use. If you specify a number, that number will only be used if it is currently active.

Syntax: DEF [NAME=] ANM.# <'PH #'> [<TOD.#> <ON|OFF>] <PAGE|MODEM>
(when MODEM) <retry> [BACKUP] [ALMBAUD <rate>]

[<NAME> =]: A user-defined point name. This is optional and does not need to be entered if you are redefining the point, but what to keep the same name.

ANM.#: the schedule number from 1 to 4

<'PHONE NUMBER'>: The phone number that the TELSEC® will dial in ALARM instances. Valid AT command characters can be used in the phone number field for delay, pulse dialing etc The number must be enclosed in single quotation marks (') and has a maximum length of 39 characters.

[<TOD.#> <ON|OFF>]: An optional TOD qualifier can be used to make this ANM active when the TOD schedule is in either the ON or OFF state. If the TOD qualifier's state is the same as this state, the alarm will call out. If the states are not true, the alarm will not call out.
<PAGE|MODEM>: An ANM defined as PAGE will dial phone number including all pauses (,) etc. It will do this one time and is intended to connect to a digital pager and send the remaining digits after the pause character. This way the technician will now what site has paged them. An ANM defined as MODEM will attempt to connect to another modem and send the alarm message. Typically the receiving modem is attached to a PC and setup to receive alarms. The TELSEC® will continually attempt to call the modem number until successful in sending the alarm message.

<retry> : The retry delay is the amount of minutes the TELSEC® should wait between calls before making another attempt or moving to the next valid number. The value can be between 1 and 5 minutes.

[BACKUP]: The BACKUP option works in conjunction with the SET COM command (see section 2.4.3). If one COM port is defined as Network and the other as POTS, then the system will only use this alarm number if the NETWORK connection is down.

[ALMBAUD <rate>]: Some alarm receivers may require that you specify the baud rate used when transmitting the alarms. With this option you can specify the callout rate of 300, 1200, 2400 or 9600 baud. The TELSEC® will use the default rate as defined in the SET COM command if you do not specify a rate. Enter the word ALMBAUD plus the rate desired to use this function.

Default: None.

Example:
```
DEFINE HEADQRTS = ANM.1 '1-813-555-1000' MODEM 1
DEFINE NITEONLY = ANM.2 '5556637' TOD.1 ON MODEM 1 BACKUP ALMBAUD 2400
DEFINE PAGENUM = ANM.3 '555-3393,,,66558' PAGE
```

2.3.2 DEFINE AOP:

Analog outputs points (AOP) are used to control devices that can be in a range of positions or speed instead of either on/off or open/closed. The TELSEC® has two built in AOP's and can be expanded to 10 AOP's through expansion boards networked to the system. Each output can be 0-10 VDC or 4-20 mA for controlling such items as dampers, valves and variable frequency drives (VFD).

Syntax: DEF [NAME=] AOP.# <LOW> <HIGH> <INIT %>
[NAME =]: A user-defined eight character point name. This is optional and does not need to be entered if you are redefining the point, but what to keep the same name.

AOP.#: A valid Analog Output Point number (1-10)
<LOW>: A number representing the value of point being controlled when the TELSEC is at the minimum setting.

<HIGH>: A number representing the value of point being controlled when the TELSEC is at the maximum setting.

<INIT %>: A number from 0 to 100 representing 0 to 100% of the output. The system uses this value to set the initial setting of the ouput prior to any equations running to change the value of the output

Default: None

Examples:
```
DEFINE VALVE_1 = AOP.1 0 100 20
```
(Controls a valve from 0 to 100% open and has an initial value of 20% open)

```
DEFINE VENTFAN = AOP.2 1300 1800 0
```
(Controls a variable speed fan that runs at 1300 rpm at the minimum output and 1800 rpm at maximum output. The initial value will be 0%).

2.3.3 DEFINE BUS:

The DEF BUS command is used to assign expansion modules and RTUs to the TELSEC® system. By defining the bus, you are telling the TELSEC® the module is there and to begin communicating with the module.

Syntax: `DEFINE [<NAME>=] BUS.#`

`<NAME>:` A user-defined optional eight character point name.

`BUS #:` The bus number of the module you are assigning. Use the following chart to determine bus number:

BUS #	Address Switches
1	First Digital Output board (points 17-24)
2	Second Digital Output board (points 25-32)
3	Third Digital Output board (points 33-40)
4	Fourth Digital Output board (points 41-48)
5	Fifth Digital Output board (points 49-56)
6	Sixth Digital Output board (points 57-64)
7	First Universal Input board (points 17-32)
8	Second Universal Input board (points 33-48)
9	Third Universal Input board (points 49-64)
10	First Analog Output board (points 3-6)
11	Second Analog Output board (points 7-10)
12	RTU controller #1
13	RTU controller #2
14	RTU controller #3
15	RTU controller #4
16	RTU controller #5
17	RTU controller #6
18	RTU controller #7
19	RTU controller #8
20	RTU controller #9
21	RTU controller #10
22	RTU controller #11
23	RTU controller #12
24	RTU controller #13
25	RTU controller #14
26	RTU controller #15
27	RTU controller #16

Example:
```
DEF DOB_1 = BUS 1
DEF BUS 2
```

2.3.4 DEFINE CARD:

See Chapter 4 – Access Control

2.3.5 DEFINE DAT:

Use to define special date ranges or holidays to be used in programming. There are eight (8) schedules. Date schedules can be used inside of time of day (TOD) schedules or can be referenced within equations. They are used when you want action on a specific date(s) instead of a day of the week schedule.

Syntax: `DEFINE [<NAME> =] DAT.# <FIRST DATE> [<CONJUNCTION> <SECOND DATE>]`

[<NAME> =]: A user-defined point name. This is optional and does not need to be entered if you are redefining the point, but what to keep the same name.

<FIRST DATE>: Any valid date entry. A valid date can be in numeric format MM/DD or text format consisting of month name and numeric date.

<CONJUNCTION>:
AND Denotes two separate dates.
TO Denotes an inclusive range of dates.

<SECOND DATE>: Any valid date entry.

Default: None

Example:
```
DEFINE CHRISTMS = DAT.1 DEC 24
DEFINE JULY4TH = DAT.2 7/4
DEFINE HOLIDAY = DAT 3 12/25 AND 1/1
DEFINE WINTER = DAT.4 NOV 1 TO APR 30
```

2.3.6 DEFINE DOR:
See Chapter 4 – Access Control

2.3.7 DEFINE EQU:
See Chapter 3 – Equation Language

2.3.8 DEFINE KEY:
The DEF KEY can be used to program specific functions for the two yellow buttons on the TELSEC® keyboard. The key labeled "F1" is KEY.1 and KEY.2 is labeled "F2" The actual function of the keys is determined by the control strategy (equation or EQU) written to use them

Syntax: DEFINE [<NAME>=] KEY.#

<NAME>: A user-defined optional point name.

KEY#: A user-defined key number assignment.

Example:
```
DEF OCCUPIED = KEY 1
DEF LEAD_LAG = KEY 2
```

2.3.9 DEFINE MSG:
The TELSEC® can send the MSG point as an alarm message (through an equation or as part of an input definition, see DEFINE UIN), store the MSG in the freeform log, or send the MSG to the front panel display. See Chapter 3 – Equation Language for syntax using the SEND, LOG, and ALARM statements. The message must be enclosed in single quotation marks (') and have a maximum length of 32 characters. There are 64 MSGs available.

Syntax: DEFINE [<NAME> =] MSG.# <'ASCII MESSAGE'>

[<NAME> =]: A user-defined point name. This is optional and does not need to be entered if you are redefining the point, but what to keep the same name.

<'ASCII MESSAGE'>: A thirty two character message inside single quotation marks.

Default: None.

Example:
```
DEFINE TOOHOT = MSG.1 'TOO HOT IN SHELTER'
DEFINE SMOKEALM = MSG.2 'SMOKE OR FIRE IN SHELTER'
```

2.3.10 DEFINE RLY:

Digital outputs are the TELSEC®'s interface to the outside world. Countless different devices can be controlled using the digital outputs of the TELSEC®. In simple terms, the digital outputs turn a connected device ON or OFF according to programmed parameters.

Use the DEF RLY command to define all of your digital outputs.

Syntax: DEFINE [<NAME> =] RLY.# <FAIL STATE> <STAGING TYPE> <ENERGIZING TYPE> <[NOT] LOG>

[<NAME> =]: A user-defined point name. This is optional and does not need to be entered if you are redefining the point, but what to keep the same name.

<FAIL STATE> ON or OFF: The relay will take this state immediately after power up and before any equations can affect it.

<STAGING TYPE>:

STAGED	Three-second staging time active for this output.
IMMEDIATE	No staging time between digital output energizing.

<ENERGIZING TYPE>: On commands.

ENERGON	Energizes the relay when an ON command is given by an equation or when the user Bypasses the point ON. An OFF command by the equations or by the user will de energize the relay.
ENERGOFF	Energizes the relay when an OFF command is given by an equation or when the user Bypasses the point OFF. An ON command by the equations or by the user will de energize the relay.

<[NOT] LOG> Type:

LOG	RLY logs on change of state.
NOT LOG	RLY does not log on change of state.

Default: ON STAGED ENERGOFF LOG

Examples:
```
DEFINE COOL = RLY.1 ON STAGED ENERGON LOG
DEFINE VENTFAN = RLY.2 STAGED ENERGOFF LOG
```

2.3.11 DEFINE RTU:

The RTU controllers are intelligent stand alone controllers that can be networked to the TELSEC® ESB system. Up to 16 RTU controllers can communicate and receive commands from the TELSEC®. The Define RTU command tells the system what settings are to be passed to the controller. The TELSEC will continuously communicate these values to the RTU over the RS485 communications bus. The RTU will default to its local settings (see the RTU User Manual) in the event of a communications fault. Note when you define an RTU, the TELSEC® will automatically define the BUS address (DEF BUS).

Syntax: `DEF [NAME=] RTU.# <SETP> <HEAT D> <STG 2> <FAN MODE> <SHDN> <ECON> <AOP> <LOG INT> <ZALMH> <ZALML> <DIG1ALM> <DIG2ALM>`

Where:

`[NAME=];` A user-defined eight character point name. This is optional and does not need to be entered if you are redefining the point, but what to keep the same name.

`RTU.#:` is the RTU number 1-16.

`<SETP>:` is the setpoint value. This can be a SPT.#, VAR.# or a numeric value. The available range of values is 60 to 91 degree F inclusive. Anything outside of this range will cause the value in the RTU to be at 72.

`<HEAT D>:` is the heat differential subtracted from SETP to determine heat turn on point. This can be a SPT.#, VAR.# or a numeric value. The available range of values is 1 to 32 degree F inclusive. Anything outside of this range will cause the value in the RTU to be at 8.

`<STG 2>:` is the delta value to decide when to turn on stage 2. This is added to the SETP for COOL 2 and subtracted from (SETP - HEAT D). This can be a SPT.#, VAR.# or numeric value. The available range of values is 1 to 8 degree F inclusive. Anything outside of this range will cause the value in the RTU to be at 4.

`<FAN MODE>:` determines if the fan should run constant or only turn on with call for heat and cool. This can be a SPT or VAR with value of 0 or 1. Any none-zero value will call for continuous fan running.

`<SHDN>:` will shutdown all outputs when this point is ON. You can define a SPT or VAR with value of 1 or 0 or use the word OFF to disable the shutdown mode. Any none-zero value will enable the shutdown mode and turn off the HVAC system.

`<ECON>:` will enable the use of the economizer. This value can be a SPT, VAR or value of 1 or 0 or use the word OFF to disable the econ mode. A value of 1 will enable econ mode, which acts as stage 1 cooling.

`<AOP>:` is the analog output value. This can be a SPT.#, VAR.#, or a number from 0 to 100 and will be the percentage output of the AOP, i.e. 0 to 100% of the range. Use the word OFF in the definition if you are not using the AOP function.

`<LOG INT>:` is the logging interval in minutes between log entries. The available entries are from 0 to 120 minutes. If you enter 0, the logging function will be turned off. Each log entry will have zone temp, supply temp and current mode of RTU. To retrieve log use REV LOG RTU.

<ZALMH>: is the amount added to the <SETP> value to determine the high temp alarm. This value can be a numeric, VAR.#, or SPT.#. The delay time in minutes before alarming can also be set (ON SPT.23 20). The value can also be OFF for no alarm. Note the alarm will clear when the temperature retreats below the setting for one minute.

<ZALML>: is the amount subtracted from the <SETP> value to determine the low temp alarm. This value can be a numeric, VAR.#, or SPT.#. The delay time in minutes before alarming can also be set (ON SPT.24 20). The value can also be OFF for no alarm. Note the alarm will clear when the temperature rises above the setting for one minute.

<DIG1ALM>: is the Fan fail alarm. If set to ON, then when the fan is running and the input is off, the RTU will generate a major (MJ) alarm condition. This will clear when the fan is running and the feedback digital is on.

<DIG2ALM>: is the Filter clogged alarm. If set to ON, then when the fan is running and the input is on, the RTU will generate a minor (MN) alarm condition. This will clear when the fan is running and the feedback digital is off.

Default: None

Example:

```
DEFINE SETPOINT = VAR.13 72
DEFINE HEATDLTA = VAR.14 7
DEFINE STG2DLTA = VAR.15 2
DEFINE ECONMODE = VAR.16 0
DEFINE FAN_RUN  = SPT.22 1
DEFINE HITEMPZN = SPT.23 8
DEFINE LOTEMPZN = SPT.24 12
DEF HVAC_Z1 = RTU.1 VAR.13 VAR.14 VAR.15 SPT.22 OFF VAR.16 OFF 15 ON
SPT.23 20 ON SPT.24 20 ON ON
```

The RTU will use VAR.13 for the base setpoint, VAR.14 for the heat delta, VAR.15 for the stage delta, SPT.22 for the fan run (0,1), VAR.16 for economizer mode, OFF = AOP not being used, ON SPT.23 20 for setting the high temp alarm on with the value SPT.23 and a delay of 20 minutes, ON SPT.24 20 for setting the low temp alarm on with the value SPT.24 and an alarm after 20 minutes, ON for fan run and ON for filter alarm.

2.3.12 DEFINE SPT:

Use DEFINE SPT to define the setpoints used in equations. The difference between SPTs and VARs is that SPTs can be modified from the front panel. There are 32 available.

Syntax: DEFINE [<NAME> =] SPT.# <INITIAL VALUE>

[<NAME> =]: A user-defined point name. This is optional and does not need to be entered if you are redefining the point, but what to keep the same name.

<INITIAL VALUE>: The starting value of the variable. The range is -32767 to 32767.

Default: 0

Example: DEFINE ROOMSPT = SPT.1 70

2.3.13 DEFINE TOD:

Use the DEFINE TOD command to set up the 16 priorities of ON or OFF times for your TODs. The TODs can then be used in other TELSEC® program areas such as equations, input definitions for when logging is to occur, alarm numbers to activate the number and card access to determine when a card is valid. Note: You can only name TODs using the NAME command. There are sixteen (16) schedules.

Syntax: DEFINE TOD.# PRIORITY# <STATE> <TIME> <DAYLIST>

PRIORITY#: The priority of this TOD program entry (16 possible).

<STATE>: The digital state (ON or OFF) this TOD will take if the TIME and DAYLIST conditions are satisfied.

<TIME>: A time of day in the form HH:MM [AM,PM] when this TOD should become active. Time will be accepted in AM, PM or 24-hour military format.

<DAYLIST>: Days-of-the-week (D.O.W.) list or a date schedule (DAT.#). If the current date or D.O.W. agrees with the programmed list, the TOD priority will return the programmed STATE.

Examples:
DEFINE TOD.1 1 ON 8:00 AM M TU W TH F
(TOD.1 will be ON if the time is after 8:00 AM and the DOW is on a weekday.)
DEFINE TOD.1 2 ON 10:00 AM SA SU
(TOD.1 will be ON if the time is after 10:00 AM and the DOW is on a weekend.)
DEFINE TOD.1 3 OFF 5:01 PM M TU W TH F
DEFINE TOD.1 4 OFF 3:01 PM SA SU
DEFINE TOD.1 5 OFF 12:01 AM DAT.XMASDAY
(TOD.1 will be OFF it the time is after 12:01 AM and the DAT schedule XMASDAY is ON.)

Default: None

2.3.14 DEFINE UIN:

Inputs come in two types which are Digital and Analog. Digital inputs are dry contact closures and can be defined as normally open or normally closed. Analog inputs are any device that outputs 0-6 VDC or 0-20 mA. The TELSEC® provides built in conversion factors for various sensors as well as manual scaling factors for sensors with different ranges and engineering units. (See section 2.4.10)

Use the DEFINE UIN command to define your analog and digital inputs.

Format:
DEF [NAME=] UIN.# (ANA) <TEMPF|TEMPC|RH|FC|MV|THERMF|THERMC|SCALE #>
<OFFS> [MSG.#] <[NOT] LOG> <[NOT] AVG (1-120)>
(DIG) <DIG|INVDIG> [<CR|MJ|MN> <DLY (0-600)> <MSG.#|NONE> [ANM LIST]]
<[NOT] LOG> [THERMF]

[NAME =]: A user-defined point name. (ie. OUTAIR). The name is optional in the define command. You do not have to enter the NAME and equal sign if you are redefining a point.

2.3.14.1 Analog inputs (ANA):

TEMPF	Degrees Fahrenheit using AD592 temperature sensors.
TEMPC	Degrees Centigrade using AD592 temperature sensors.
RH	Relative Humidity conversion factor.
FC	Foot-Candle (light level) conversion factor.
MV	Milli volt conversion factor.
THERMF	Degrees Fahrenheit using 10k Type III Thermistor temperature sensors.
THERMC	Degrees Centigrade 10k Type III Thermistor temperature sensors..
SCALE #	Use a manual scaling factor for this input. (See section 2.4.10)

<OFFSET>: A number between -127 and 127 must be entered here. Only whole numbers will be accepted. This number is used to correct the sensor reading.

[MSG.#]: The message option (MSG) allows you to assign one of the 64 messages to the input. When the point is alarmed, the system will send the assigned message in the alarm message.

<[NOT] LOG>: The TELSEC® can be programmed for this input to automatically insert an entry into the log space for this point. Use the word LOG if you want to log the input or use NOT LOG to prevent automatic entry into the history log.

<[NOT] AVG>: This input can be programmed to have instantaneous data or averaged data sampled every minute for the log entry. NOT AVG will cause the system to wait the delay time and then enter the current reading into the history log. AVG will cause the system to average the sensor reading over the interval time and then enter the average reading once the interval time has been met.

<LOG INTERVAL>: Input the minute interval for log entries here. The range is 1 - 120 minutes.

2.3.14.2 Example Analog Define UIN:

```
DEF ROOMTEMP = UIN.1 TEMPF -1 MSG.1 LOG AVG 15
DEF ROOM_%RH = UIN.2 RH 0 LOG NOT AVG 16
DEF OUTAIR = UIN.7 THERMF 0 LOG TOD.1 AVG 30
DEF DC_AMPS = UIN.16 SCALE 1 0 LOG AVG 5
```

2.3.14.3 Digital inputs (DIG):

<DIG/INVDIG>: A point defined as DIG will show an ON or alarm value when a contact closure is made (normally open). A point defined as INVDIG will show an ON or alarm value when the input is in the open state (normally closed).

DIGITAL	Digital input point for normally open points.
INVDIG	Digital input point for normally closed points.

<CR/MJ/MN>: Alarm Condition – Use this option if you want to create automatic digital alarming. If the point is a monitor only point or requires additional logic for alarm using the equations (EQU) then omit this option.

CR	Critical alarm
MJ	Major alarm
MN	Minor alarm

<DELAY 0-600 SECONDS> Alarm Delay: 0-600 seconds that the system will wait prior to generating an alarm.

<MSG. #/NONE>: A 32-character message can be associated with this point. See Defining MSGs. There are 64 messages available. Messages can be added to digital inputs that are defined with the automatic alarming and for digital inputs that are monitor only points or will be alarmed through Equations.

[ANM LIST]: When digital inputs defined as alarm points <CR/MJ/MN>, you can specify which alarm phone number you want the system to dial when the point goes into alarm and then clears the alarm condition. The acronym for alarm number is ANM. Add the list of ANM's one at a time after the <MSG. #/NONE> field. Example ANM 1 ANM 2 etc. All defined alarm numbers will be dialed if you do not specify ANMs.
<[NOT] LOG>: The TELSEC® can be programmed for this input to automatically insert an entry into the log. Digital inputs log when the point changes state. Using the keyword NOT LOG will prevent the system from entering change of states in the history log.

[THERMF]: This is an optional parameter that tells the system to use the built in thermistor circuit for sensing the digital input. The thermistor circuit looks for voltage in the 0 to 5v range. This option is useful when piggybacking other alarm systems that are monitoring the same point.

Examples:
```
DEFINE SMOKE = UIN.4 INVDIG MJ 10 MSG.4 LOG
DEFINE FUSEPNL = UIN.5 DIG MJ 1 MSG.5 ANM 1 ANM 3 LOG
DEFINE LIGHT_SW = UIN.10 DIG LOG
DEFINE FIRETRBL = UIN.11 INVDIG MJ 10 MSG.11 LOG THERMF
DEFINE Vent_SW = UIN.12 DIG MSG.12 LOG
```

Default:
None

2.3.15 DEFINE VAR:

Use DEFINE VAR to define memory variables used in equations. There are 64 available. Variables cannot be changed from the front panel. Memory variables are useful to report status or to store numbers for equations such as the outcome of a mathematical equation (average of two sensors) or as a "flag" to tell other equations to be active based on the value.

Syntax: DEFINE [<NAME> =] VAR.# <INITIAL VALUE>

[<NAME> =]: A user-defined point name. This is optional and does not need to be entered if you are redefining the point, but what to keep the same name.

VAR.# <INITIAL VALUE>: The starting value of the variable. The range is -32767 to 32767. Equations can change this value. The current value can be seen with the REV VAR command where the initial value can be seen with the LIST VAR command.

Default: 0

Example: DEFINE ROOMAVG = VAR.1 70

2.4 SET Command:

The SET command is used to configure items that are global to all the other functions such as the system clock, passwords and communications ports to name a few. The available items to SET are:

```
ID  CLOCK  PSWD  DLS  MAIL  SCALE  LIST  PROGRAM  COM  BUSALARM
```

2.4.1 SET BUSALARM:

The SET BUSALARM command is used to set the delay in minutes prior to generating a bus communications error alarm for a networked module.

Syntax: `SET BUSALARM <# (0-60)>`
`<# (0-60)>:` The number of delay minutes prior to generating a bus error alarm.

Example:
`SET BUSALARM 10`
Sets the bus alarm delay to 10 minutes

Default:
`SET BUSALARM 5`
5 minute bus error delay.

2.4.2 SET CLOCK:

Use the `SET CLOCK` command to set the system clock.

Syntax: `SET CLOCK <DATE FORMAT> <TIME FORMAT>`

`<DATE FORMAT>:` Enter the current MM/DD/YYYY. The system will accept the year with only the last two digits. Ie 07 instead of 2007

`<TIME FORMAT>:` Use HH:MM:SS with optional AM/PM or military time accepted. You do not need to specify the seconds. The system assumes 00 seconds if none are specified.

Example:
```
SET CLOCK 4/21/2007 3:15:20 PM
SET CLOCK 4/21/2007 3:15 PM
SET CLOCK 4/21/07 15:15
```

Leap Year Note: The TELSEC® automatically adjusts for leap year.

Daylight Savings Note: The system will adjust for daylight savings (DLS). This feature can be changed or turned-off using the SET DLS command.

2.4.3 SET COM:

The `SET COM` command sets the communications functions for the two COM ports. COM 1 is the primary communications port and will be automatically configured for a modem if the modem is present. COM 2 is the RS232 craft port interface. It is recommended that the settings be left with their default settings unless you have difficulty connecting with a remote modem.

Syntax: `SET COM <1|2> <POTS|DIRECT|NETWORK> <BAUD> <8|7> <2|1> <N|E|O>`
`<ON|OFF (ECHO)> ['AT STR']`

<COMNUMBER>: Enter 1 or 2.

<TYPE>:

POTS	Dial-up connection.
DIRECT	RS232 connection only.
NETWORK	Network connection. Works similar to DIRECT, but allows the BACKUP function in the DEF ANM to work when the port is down (see section 2.3.1).

<BAUD>: Enter the speed you want to use for communications from 300 to 9600 baud.

<8|7> Data Bits: Enter 7 or 8 for the data bits.

<2|1> Stop Bits: Enter 1 or 2 for the stop bits.

< N|E|O > Parity:

E	even parity
O	odd parity
N	no parity

<ON|OFF (ECHO)>:

ON	Shows characters typed.
OFF	Does not show characters typed.

['AT STR']: optional AT init string is available for ports defined as POTS. It is recommended that you do not change the init string unless you are familiar with AT command sets and require setting changes for proper connectivity.

Example:

```
SET COM 1 POTS 2400 7 1 E ON
```

(The existing AT init string will be retained if a new one is not specified.)

```
SET COM 2 DIRECT 2400 8 1 N OFF
```

Default:

```
SET COM 1 POTS 9600 8 1 N ON 'ATE0V0X1&S0&C1&D2S7=30S0=1'
SET COM 2 DIRECT 9600 8 1 N ON
```

NOTE: Default COM settings are dependant on how the system is configured at the factory.

2.4.4 SET DLS (Daylight Savings):

This is used to change the default daylight savings time.

Syntax: SET DLS <SPRING|FALL> < {<FIRST|SECOND|THIRD|FOURTH|LAST> <DOW> <MONTH>} | <NONE> >

<SPRING|FALL>:

SPRING	Clock moves ahead one hour.
FALL	Clock moves back one hour.

{<FIRST|SECOND|THIRD|FOURTH|LAST> <DOW> <MONTH>}: Specify the position of the month, the day of the week and the month in which you want the DLS to take effect.

<FIRST|SECOND|THIRD|FOURTH|LAST> To specify the position in the month.
<DOW> To specify which Day Of the Week DLS occurs.
<MONTH> Enter the month of daylight savings.

The word NONE can be entered for no DLS clock adjustment.

2.4.5 SET ID

The Set ID command is used to set the system identification. There are three lines available for the user to change.

Syntax: SET ID <IDNUMBER> <'ID STRING'>

SET ID <IDNUMBER>: A number from 1 to 3. The TELSEC® actually has four (4) ID strings but the fourth is unchangeable.

<'ID STRING'>: A string of alphanumeric characters used to identify this particular TELSEC® site. The ID strings are displayed during all call-ins and call-outs. The maximum length is 78 characters and the string must be enclosed within single quotes (').

Example:
```
SET ID 1 'CEV#1001'
SET ID 2 'PALMETTO, FLORIDA'
SET ID 3 'INSTALLED DECEMBER 15, 1999'
```

Default:
```
SET ID 1 'TELSEC ESB'
SET ID 2 'QUEST CONTROLS, INC.'
SET ID 3 'PALMETTO, FL'
SET ID 4 'REV X.X - RELEASE DATE'
```

2.4.6 SET LIST:

SET LIST establishes the format for how equations will be displayed when they are listed for viewing.

Syntax: SET LIST <NUMBER|NAME|NONE>

<NUMBER|NAME|NONE>:
NAME: Equations will list using the format PT.NAME. (Example: UIN.OUTAIR)
NUMBER: Equations will list using the format PT.NUMBER. (Example: UIN.3)
NONE: Equations will list using the format NAME. (Example: OUTAIR)

Default: NUMBER

2.4.7 SET MAIL:

SET MAIL is used to store information about the site or to communicate with other techs. This information is displayed with the LIST MAIL command. The MAIL field is also displayed after the system ID on modem dial in.

Syntax: SET MAIL <MAILNUMBER> <'MAIL STRING'>

<MAILNUMBER>: A number from one (1) to four (4).

<'MAIL STRING'>: A string of alphanumeric characters used for this particular mailbox. The MAIL strings are displayed during all call-ins after the ID strings. The maximum length is 80 characters and the string must be enclosed within single quotes (').

Example:
```
SET MAIL 1 'DISPATCH - CHANGE AIR FILTERS NOW'
SET MAIL 2 'KEN I TOLD YOU TO CHANGE THE LIGHT FIXTURE YESTERDAY!'
```

Default: None.

2.4.8 SET PROGRAM:

The SET PROGRAM command is used to receive application programs, store programs remove programs and perform upgrades.

Syntax: SET PROGRAM <TYPE>

<TYPE>:

DEFINE:	Takes the current program in RAM and writes it to the non-volatile flash memory.
REMOVE:	Removes the application program stored in flash. When the system is cold started, it will come back with no application program loaded.
PROGRAM:	Starts the Xmodem protocol to receive an application program using Xmodem transfer.
MAX:	Starts the Xmodem protocol to receive an operating system Upgrade via Xmodem transfer. Contact your Quest representative for available upgrades and further instruction.

2.4.9 SET PSWD:

SET PSWD is used to set the available access codes and level of access. These passwords are separate from the Ethernet passwords and are used for RS232 direct connection or modem dialup.

Syntax: SET PSWD <#> <READ|PROG|BYPASS|ACCESS| MASTER> <'PSWD'>

<PSWD NUMBER>: A number from 1 to 20

<ACCESS LEVELS>:

READ	Allows REVIEW, HELP, HANGUP
PROGRAM	Allows CLEAR, LIST, NAME, SET, REMOVE, COPY, SEARCH, DEFINE
BYPASS	Allows BYPASS commands
ACCESS	Allows DEFINE CARD, DEFINE DOR, REVIEW, LOG ACCESS, LIST CARD
MASTER	Allows SET PSWD, DEFINE EQU

<PSWD STRING>: The alphanumeric password code for the particular PSWD TYPE. Maximum length is eight characters. The password code must be enclosed within single quotes (').

The password levels associate specifically with commands. If you want access to a specific command you must specify a password with the corresponding level. A MASTER level alone would not have access to the REVIEW command. You need READ access for this command to function.

Example:
```
SET PSWD 1 READ 'AAA'
SET PSWD 2 READ PROGRAM BYPASS 'TECH'
```

```
SET PSWD 3 READ PROGRAM BYPASS ACCESS MASTER 'KAHUNA'
```

Default:
```
SET PSWD 20 READ PROGRAM BYPASS ACCESS MASTER 'MASTER'
```

Example:
```
SET DLS SPRING SECOND SUNDAY MARCH
SET DLS FALL FIRST SUNDAY NOVEMBER
SET DLS FALL NONE
SET DLS SPRING NONE
```

Default:
```
DLS SPRING: SUNDAY, MAR 11, 2007  2:00:00 AM
DLS FALL  : SUNDAY, NOV 4, 2007  2:00:00 AM
```

Once a date type is entered, the TELSEC® calculates the actual date of DLS. The LIST DLS command can then be used to see the actual date. The time adjustment occurs at 2:00 AM on the calculated date.

2.4.10 SET SCALE:

There are eight (8) user-definable scaling factors that can be used to create custom engineering units for inputs. Once you create a scale you can reference it with the DEF UIN command (see section 2.3.14.1)

Syntax: SET SCALE <1-8> <MIN> <MAX> <'3 CHARS'> [<blank>|V|A|T]

MIN: The minimum value of the sensor. This is the value the TELSEC® will display when the input is at zero (0) volts.

MAX: The maximum value of the sensor. This is the value the TELSEC® will display when the input returns a value of six (6) volts. Many sensors return a maximum of five (5) volts so the value must be calculated in this situation. Example: you have a 0-100 amp transducer that provides a proportional signal of 0-5 VDC. There is 20 amps per volt DC (100/5) so at 6 volts the sensor would read 120 amps. Enter 120 as the maximum and 0 as the minimum.

3-CHAR NAME: The three -character name that will display when any input defined with this SCALE # is REVIEWed.

[T]: By using the optional T on the end of the SCALE command, you tell the system to use the Thermistor (resistive) circuit instead of the normal 0-6v input. Use this function when you are scaling resistive input devices such as temperature sensors or setpoint adjuster slide switches.

[A]: The "A" option tells the system to use the current sensor circuitry on the expansion input boards when measuring the input. If you assign this scale to an input on the main unit, the "A" will be ignored.

[V] This option tells the system to use the 0-10 VDC circuitry on the expansion board when measuring the input. If you assign this scale to an input on the main unit, the "V" will be ignored.

Example:
`SET SCALE 1 0 60 'AMP'`
(scale for a 0-50 amp transducer with an output of 0-5 vdc)

`SET SCALE 2 -25 125 '%RH' A`
(scale for a 0-100% humidity sensor over 4-20 mA using the current sensing circuitry on the expansion boards).

`SET SCALE 3 -3 3 'ADJ' T`
(scale for a +/- 3 degree setpoint adjustment slider)

Default: None. There are eight (8) user-definable scaling factors.

2.5 LIST Command:

Use the `LIST` command to retrieve the TELSEC® program data. The list command will list back the program element in the exact format that the TELSEC® will accept command.

Syntax: `LIST < POINTTYPE >[.#]`

`<POINTTYPE>:` Any TELSEC® point type.

`UIN`	Definitions of the Inputs
`RLY`	Definitions of the Digital Outputs
`AOP`	Definitions of the Analog Outputs
`RTU`	Definitions of the networked RTU controllers
`KEY`	Definitions of the two yellow keys on the keypad
`DAT`	Alternate date schedule definitions
`TOD`	Time OF Day schedule definitions
`EQU`	Listing of an equation program.
`VAR`	The initial setting for memory variables
`SPT`	The initial setting for set points
`DOR`	Definition of the door access control points
`ANM`	Definition of the Alarm phone numbers.
`MSG`	The definition of all system messages
`ID`	The System Identification strings.
`PSWD`	The settings for the various available passwords.
`DLS`	Day Light Savings settings
`CARD`	The definitions of a CARD for the card access option.
`BUS`	Shows BUS addresses assigned and any communication errors.
`MAIL`	Shows the four mail box lines for user messages.
`SCALE`	Shows the scale factors for the manual scales
`LIST`	Shows how the equations will list back.
`COM`	The current settings for the communications ports.
`BUSALARM`	Shows the current bus error alarm delay

`[.#]`: An optional number list shows the specified point type. This can be used for all points that more than one entry.

Examples:

`LIST UIN.1,2,5-7`	Programming will list for `UIN's 1,2,5,6,7.`
`LIST RLY`	Programming will list for all `RLY` points.
`LIST TOD.1`	Programming will list for `TOD.1`.

The `LIST` command supports the key word ALL, which will cause the system to list back all of the TELSEC® programming with the exception of the card access (CARD) database. This is useful for retrieving the program for storage on a local computer.

Example:
`LIST ALL` (lists all programming except CARDs)

2.6 CLEAR Command:

The CLEAR command is used to reset a point. For timer points that mean resetting the timer back to zero. For other points it means to reset the value back to the original starting value.

Syntax: `CLEAR < POINTTYPE >[.#]`

Where `< POINTTYPE > is:`

`UIN`	Resets all timers associated with the point to zero
`DAC`	Resets the Accumulated On timer to 0
`DNO`	Resets the Interval On timer to 0
`DNF`	Resets the Interval Off timer to 0
`DTM`	Resets the Total Time elapsed to 0
`DEC`	Resets the Event Counter to 0
`RLY`	Resets all timers associated with the point to zero
`RAC`	Resets the Accumulated On timer to 0
`RLO`	Resets the Interval On timer to 0
`RLF`	Resets the Interval Off timer to 0
`RTM`	Resets the Total Time elapsed to 0
`REC`	Resets the Event Counter to 0
`AOP`	Sets the output to the initial value
`TOD`	Causes the TOD schedule to reprocess and run through all priorities
`VAR`	Sets the value of the memory variable (VAR) to the initial defined setting
`SPT`	Sets the value of the set point (SPT) to the initial defined setting
`RTU`	Clears the error count to zero if a board is in error.
`BUS`	Does nothing - for future use
`LOG`	Erases all log entries in the main UIN & RLY log.

`[.#]`: An optional number list shows the specified point type. This can be used for all points that more than one entry.

Examples:

`CLEAR UIN.1,2,5-7`	Resets timers for `UIN's 1,2,5,6,7.`
`CLEAR SPT 1`	Clears set point one to the defined initial value.
`Clear AOP.1`	Sets analog output (AOP) 1 to its defined initial percentage output.

2.7 REMOVE Command:

Use the REMOVE command to delete a point(s) from the TELSEC® programming memory. Some items may not be removed and should be re-defined rather than removed. The points you removed only affects RAM memory and doesn't affect the program stored in flash unless you do a SET PRO DEF command to store the new settings from RAM memory

Syntax: `REMOVE <SOURCE POINTTYPE>.#`

`<SOURCE POINTTYPE>:` Available point types are:

`RLY DAT TOD EQU DOR ANM MSG RTU KEY BUS PSWD CARD`

Example:

`REMOVE PSWD. 1`	Deletes `PSWD.1` from the system.
`REMOVE EQU.1-4`	Removes equation `1 through 4.`
`REMOVE TOD.1`	Removes programming for all priorities of `TOD.1`

!!CAUTION!! The `REMOVE` command will wipe out programming for the TELSEC®. Use it with caution.

2.8 NAME Command:

Use the `NAME` command to set names for any TELSEC® system points. The name assignments can be as many as eight (8) characters long and must start with an alpha letter (A - Z). The characters %, _, and & can also be used within the name.

Syntax: `NAME <STRING> = <POINT.#>`

`<STRING>:` A string up to as many as eight characters.

`<POINT.#>:` Any of the TELSEC® system points.
Available system points are:

`UIN RLY AOP RTU KEY DAT TOD EQU VAR SPT DOR ANM MSG`

Example: `NAME FAN_FAIL=UIN.5`

2.9 COPY Command:

The `COPY` command provides a quick and easy way of copying point definitions. After programming one point, you can use the copy command to write that programming to one or a range of specified points. The name of the point is NOT copied. You must name your points after the `COPY` procedure.

Syntax: `COPY <POINT.#> <RANGELIST>`

`<POINT.#>:` Available point for the copy command are:

`UIN RLY KEY DAT TOD VAR SPT ANM MSG BAT`

<RANGELIST>>: A single or list of numbers. A list must be delimited by commas and a range uses the dash symbol.

Example:
`COPY UIN.1 3`
(Copies programming from `UIN.1 to UIN.3`.)
`COPY RLY.1 2,4,5-8`

(Copies programming from `RLY.1 to RLY 2, 4, 5, 6, 7`, and 8.)

2.10 TELSEC® Point Acronyms and Quantities

The numbers in parenthesis show the maxim number available for the TELSEC®.

ANM Alarm Phone Number (4)
AOP Analog Output Point (2) expandable to (10)
BUS Communication points for expansion modules (27)
CARD Access Control Cards (600)
DAT Alternate Date Schedules (8)
DOR Door access control point (4)
EQU Equations or control strategies (64)
KEY Yellow Keys on Front Panel (2)
MSG User definable 32 character messages (64)
RLY Digital Output Point (16) expandable to (64)
- RAC Relay Accumulating ON Timer
- RLO Relay Interval ON Timer (1 per RLY)
- RLF Relay Interval OFF Timer (1 per RLY)
- RTM Relay Event Timer (1 per RLY)
- REC Relay Event Counter (1 per RLY)

RTU Roof Top Unit controllers (16)
SPT Setpoints. Used to store and reference setting for control strategies. (32)
TOD Time Of Day Schedule Point (16 schedules with 16 priorities each)
UIN Universal Input Point (16) expandable to (64)
- DAC Digital Input Accumulating ON Timer (1 per Digital UIN)
- DNO Digital Input Interval ON Timer (1 per Digital UIN)
- DNF Digital Input Interval OFF Timer (1 per Digital UIN)
- DTM Digital Input Event Timer (1 per Digital UIN)
- DEC Digital Input Event Counter (1 per Digital UIN)

VAR Memory Variables used for status and storing the outcome of equations (64)

Chapter 3 – Equation Language

3.1 General Remarks

Equations are the heart of the TELSEC®'s programming. Chapter 2 – Basic Programming Commands tells about the various program elements. Before writing equations, these elements should be defined to give them names and outline how they should work. When writing equations, defined program elements are combined. An equation is a sequence of activities, directed toward a specific goal. This goal might be: computing degree days, logging abnormal temperature readings, operating `RLY.3` as desired, or some other function of your choosing.

As many as 64 equations can be programmed into the TELSEC®. Each equation has its own goal. One equation may compute a number value and convey it to another equation for its use. Together, these equations control the TELSEC® and the equipment attached to its relays.

The TELSEC® operates all equations at the same time. For instance, if an equation tells the TELSEC® to do something in any situation, the TELSEC® does that thing repeatedly, and also does everything it is told to do by any other equation. An equation can tell the TELSEC®, "Wait for ten minutes." Such a statement doesn't bring the entire TELSEC® to a halt, but only that equation. When any equation is waiting, the TELSEC® recognizes it, and recognizes what the equation is waiting for. The TELSEC® continually checks to see if the equation can resume operation. (Section 3.9.3 gives more detailed information about the exact sequence in which the TELSEC® runs equations.)

3.2 The Components of Equations

3.2.1 Formulas:

Formulas tell the TELSEC® to do arithmetic. Formulas combine program elements that have numeric values, by adding, multiplying, taking remainders of division, and other operations. When a formula appears in an equation, the TELSEC® does the computation and uses the resulting number in place of the formula. (See Section 3.3.6)

3.2.2 Assignments:

Assignments look like equations in mathematics, because they use an equal sign. However, equations in the TELSEC® mean something different. The TELSEC® computes the value of the formula on the right side of the equal sign and assigns it to the object on the left side. So you can write seemingly impossible math equations, such as: `VAR.4 = VAR.4 + 1.`

3.2.3 Statements:

Statements take actions, like turning on a relay, logging data, or making a phone call. Each statement has a different form and requires entry of a different combination of formulas or program elements. Assignments are a form of statement. This chapter will present each type of statement and provide examples of how they are used.

3.3 The Form of Equations

Every equation consists of its define line and one or more statements. The statements are separated by commas. This usually does not look like an equation from mathematics. If the equation doesn't have an assignment or a comparison in it, it may not even have an equal sign. In the TELSEC®, "equation" means a separate, goal-directed sequence of steps.

3.3.1 General Format:

The general format for writing equations is as follows:
DEFINE <EQUNAME> = EQU.#<cr>
<STATEMENT>,<cr>
<STATEMENT><cr>
<cr>

<EQUNAME>: EQUNAME can be any unique 8-character name. The # symbol can be any number from 1 to a maximum of 64. STATEMENTS are entered on successive lines after the DEFINE line. If more than one STATEMENT is to be entered, separate them with commas. When the equation is completed, terminate the entry with two successive carriage returns. The TELSEC® will then know to process the equation and will report any errors or accept what was sent with an "OK" followed by the amount of memory the equation occupies.

3.3.2 Conditional Equations:

Conditional equations (see Section 3.3.9) are an especially useful form. They use the words IF, THEN and ELSE. The TELSEC® performs the statements only IF the specified condition is TRUE. Otherwise, an ELSE condition statement can be executed. This is the way to program the TELSEC® to take different actions at different times or in different situations.

Follows is a typical equation:

```
DEFINE TIMESCHD = EQU.18
IF TOD.1 = ON THEN TURN ON RLY.1
ELSE TURN OFF RLY.1
```

3.3.3 One-time Equations:

One-time only equations can be programmed using the DO command. Simply enter the word DO [ENTER] when at the semicolon (;) prompt and enter an equation. The equation will run one time and then destroy itself. This is an easy way to make quick changes to the system or to test the alarming function.

Examples:

```
DO <enter>
ALARM UIN 1 MJ<enter><enter>
```

(The system will alarm input 1 with Major severity. This will generate the alarm, put it in the active alarm list and alarm history log)

```
DO <enter>
ALARM UIN 1 CLEAR<enter><enter>
```

(The system will CLEAR the alarm condition. This will clear the alarm from the active list and put a clear alarm in the history log)

NOTE: make sure you CLEAR any alarms that you manually created once you have completed testing.

```
DO <enter>
SEND MSG 2<enter><enter>
```
(sends the contents of MSG 2 to the front display buffer)

```
DO <enter>
SEND MSG 2 CLEAR <enter><enter>
```
(Clears MSG 2 from the front display buffer)

```
DO <enter>
SPT 1 = 50<enter><enter>
```
(temporarily sets the value of SPT 1 to 50)

3.3.4 Typing numbers:

When typing a number, type only the series of digits. Commas and/or decimal points can not be used. If typing a negative number, start the number with a minus sign.

For example: 15000
-25
0

3.3.5 Typing intervals:

One way to specify an interval is to simply type a counting number, as just described. A number that represents an interval cannot be negative. The TELSEC® interprets this number as a number of seconds. You can also specify an interval in the form hh:mm:ss (hours, minutes, and seconds). For example, 1:00:00 represents one hour. Typing 1:00 represents 1 minute and typing 0:15 represents a fifteen-second interval.

3.3.6 Arithmetic:

The TELSEC® uses formulas to perform arithmetic. Formulas combine program elements discussed in Chapter 2 – Basic Programming Commands. Number arithmetic combines number items such as variables, analog inputs, event counters, and intervals (which are numbers of seconds). In the TELSEC®, all numeric program elements have values which are counting numbers, such as 0, 72, or -20. If you write a formula that uses division or takes a percentage of something, the result will be a fraction. But before you can store this number anywhere, the TELSEC® truncates the number. For example, 18.5 would be 18. If you need greater accuracy multiply the numerator by 10, to move the decimal point, before dividing.

3.3.7 Operators:

Using "m" and "n" to represent any number or system point that has a numeric value (e.g. UIN, RYL, VAR, SPT etc.) elements, two elements can be combined by typing one of these symbols:

```
m + n   add two numbers
m - n   subtract the second number from the first
m * n   multiply two numbers
m / n   divide the second number into the first
m % n   take m percent of n.  This is (m / n) * 100
m MOD n       find the remainder of the division m / n
```

Parentheses tell the TELSEC® which operators to perform first. The TELSEC® evaluates everything inside the parentheses before combining the resulting value with anything outside the parentheses. For example:

(3 * 4) + 5 has the value 17
3 * (4 + 5) has the value 27

When the TELSEC® lists an equation, it supplies parentheses if the programmer did not type them originally. This shows exactly how the TELSEC® interpreted the formulas entered. The TELSEC® follows normal rules of precedence; multiply/divide operations first, then addition/subtraction operations.

3.3.8 Functions:

Functions also combine numeric elements. Enter the name of the function, an open parenthesis, the element or elements to which the function will be applied, and a closed parenthesis. If a function will be applied to more than one number, separate the numbers by commas. The TELSEC® provides these functions:

ABS(m)	Absolute value: remove any minus sign that 'm' may have.
MIN(m,n)	Find the minimum (lowest) number in a list of up to ten numbers.
MAX(m,n)	Find the maximum (highest) number in a list of up to ten numbers.
AVG(m,n)	Find the average of a list of up to ten numbers. The TELSEC® adds each element and divides the sum by the count of elements in the list.

Combinations are legal, since formulas and functions are themselves number elements. For instance, you can put a function inside another function. This example returns the lowest of three temperature readings, but never returns a number lower than 10:

```
MAX(10, MIN(UIN.TEMP1,UIN.TEMP2, UIN.TEMP3))
```

When a function is placed within a function, be sure to type matching left and right parentheses. Notice in the above example there two left parentheses and two right parenthesis. The TELSEC will evaluate the MIN function first and then evaluate the MAX function.

3.3.9 Conditionals:

Conditional statements can be utilized in equations to link the functions of any system point to a corresponding action. They always contain a condition (IF) followed by an action (THEN).

The keyword 'IF' appears in an equation to make one or more statements after it conditional. The conditional statements only take effect if the specified condition is. You type IF, followed by condition you want to test, followed by THEN, followed by the action statements:

```
IF fact THEN statement, statement, ...
```

If there are statements you want to take effect only if the specified fact is FALSE, then use the word 'ELSE.' Although several statements may have been entered after THEN, separated by commas, do not type a comma immediately before the word 'ELSE':

```
IF fact THEN statement, statement, ..., statement
ELSE
IF fact THEN statement, statement, ..., statement
ELSE
statement, statement, ..., statement
```

The IF/THEN/ELSE technique enables the TELSEC® do perform different functions in different situations. It is the primary way to link physical points and program points to create a control sequence.

The word 'THEN' must be used after every use of the word 'IF.' If the only relevant case is the case where the fact is FALSE, then test the opposite fact by using the word 'NOT' as follows:

```
IF NOT (fact) THEN statement, statement...
```

It's common to use IF/THEN/ELSE where it is desirable for only one group of statements to take effect:

```
IF fact THEN statement, statement...
ELSE
IF fact THEN statement, statement...
ELSE
IF fact THEN statement, statement...
```

If an entire equation follows the form shown above, then the statements on only one line take effect at a given time. The first line where the fact is TRUE is the line from which statements take effect. After carrying out the statements, the TELSEC® proceeds to the next equation. At other times, if some of the facts switch between TRUE and FALSE, the statements from different lines may take effect instead. The TELSEC always process the equations from the beginning (top) going left to right. Once it finds a TRUE statement, it will do the corresponding action statements (after the THEN). Think of each IF statement as priorities where the first IF statement will be the highest priority.

3.3.10 Comparisons:

The operators and functions in Section 3.3.7 combine numbers and produce a number. Comparisons are also operators, but they produce a value of TRUE or FALSE. The most common place for comparisons is between the words IF and THEN. The TELSEC® will perform a function only if the comparison is TRUE; this is how the TELSEC® tests its points.

Once again using "m" and "n" to stand for any number element, the TELSEC® provides six comparisons:

m=n	TRUE if m equals n (FALSE otherwise).
m<n	TRUE if m is less than n.
m>n	TRUE if m is greater than n.
m<=n	TRUE if m is less than or equal to n.
m>=n	TRUE if m is greater than or equal to n.
m<>n	TRUE if m is not equal to (less or greater than) n.

As well as comparing number elements, digital elements can be compared. For example, you can see whether a digital input is ON by writing:

```
IF UIN.SWITCH4 = ON
```

In fact, digital elements can be compared and combined with number elements by assuming OFF = 0 and ON = 1.

Conjunctions combine elements (such as digital inputs and the results of comparisons). Conjunctions are operators, but they take the form of separate words. The words 'AND' and

'NOT' are conjunctions. They mean exactly the same thing as they do in English. For example, use 'AND' to conjoin two comparisons:

```
(m = 12) AND (n = 19)
```

This formula is TRUE only if both sides are TRUE; otherwise, it is FALSE. You could use the word NOT to reverse this state:

```
NOT ((m = 12) AND (n = 19))
```

This formula is FALSE only if both the comparisons are TRUE.

You can use the conjunction OR just like AND. But this OR is "inclusive," not an either/or, as you usually mean in English. For example, you could write the following:

```
(m = 12) OR (n = 19)
```

The meaning of this is obvious except for one thing: If both halves are TRUE, the total formula is still TRUE.

3.3.11 The FOR keyword:

Any comparison or other TRUE/FALSE element can be followed with the word 'FOR' and a time interval. The time interval can be a constant or any numeric element, representing a number of seconds. For example: `IF UIN.TEMPSENS > 85) FOR 0:10:00`

This expression asks the TELSEC® to see if the input is greater than the number 85 for ten minutes running. When the TELSEC® reaches a comparison of this form, it sets an internal timer to 00:00. The TELSEC® continually tests the element. If it is TRUE, the timer runs. If the TELSEC® ever finds it FALSE, the timer goes back to 00:00. Only if the timer reaches the specified interval (in this example, ten minutes) does the equation proceed. So if a comparison with the word 'FOR' is entered, it takes the TELSEC® at least the specified interval, and possibly longer, to produce a result.

3.3.12 Switching Relays:

The TURN statement sets a specified relay to the ON or OFF state. Section 2.3.11discusses relays and explains what 'ON' and 'OFF' means in the real world. There are two forms of the statement; both require exactly one relay to be specified:

```
TURN ON rly <#>
TURN OFF rly <#>
```

A sequence of TURN statements separated by commas can be used in an equation to switch more than one relay. The TURN statement has no effect if the relay was already ON or OFF; it simply stays in the desired state.

Examples: `TURN ON RLY.14, TURN OFF RLY.COOL_1`

3.3.13 Analog Outputs:

Equations are used to control the analog outputs by assigning a value of 0 to 100 equaling 0 to 100% of the output. This will allow one hundred steps over the 0-10V or 4-20mA output. The AOP is controlled by assigning the AOP point to a specific number or to a Set point (SPT) or memory variable (VAR) to pass a value to the AOP.

Examples: `AOP.1 = 50, AOP.2 = VAR.2, AOP.3 = (UIN.1 – SPT.1)*10`

3.3.14 Bus Status:

The TELSEC® monitors the communications of all expansion boards and RTUs assigned on the network and will automatically generate alarms when a module ceases communications. Additionally when the module begins communicating again, the TELSEC® will generate a clear alarm. In some cases you may want to compare the bus alarm status to a set point or variable to generate additional control logic such as turning on a relay connected to a light or horn to indicate the problem or to do an automatic reset of the bus by cycling power through a relay. The point name is BUS (See section 2.3.3)followed by the BUS number (See section 2.3.3). The value returned is the status of the error counter.

Example: `IF BUS.1 = VAR.2 THEN . . .`

3.3.15 Waiting:

The `WAIT` statement indicates that any remaining statements in the equation should not run until some time in the future. When the TELSEC® reaches a `WAIT` statement, it suspends work on that equation for some number of seconds that you specify. When typing a `WAIT` statement, you must specify a number of seconds, either by typing an interval or by specifying a numeric element: `WAIT` interval.

If a numeric element is used (for example, a variable) to specify a number of seconds and another equation changes the element's value during the wait, it can change the length of the wait.

Examples:
```
WAIT 1:00:00
WAIT VAR.DELAY
```

3.3.16 WAIT UNTIL:

The `WAIT UNTIL` statement is a more complex `WAIT` statement. After the words `WAIT UNTIL`, you can type any element: a comparison, a digital input, or a conjunction of several of these. When the TELSEC® reaches `WAIT UNTIL`, it suspends work on the equation if the element has the value `FALSE`. The TELSEC® will continue to evaluate the element in case its value should change to `TRUE`. When this happens, the equation proceeds; statements following `WAIT UNTIL` will then take effect.

`WAIT UNTIL` fact: In an equation that tests for a problem condition, the last statement in the list is often `WAIT UNTIL`, to ensure that the problem has gone away. This makes sure the TELSEC® doesn't start the equation over again until the next time the problem occurs.

Examples:
```
WAIT UNTIL NOT UIN.ALARM
WAIT UNTIL (UIN.12 < VAR.SETPOINT)
WAIT UNTIL (UIN.TEMP4 > 80) FOR 0:03:00
```

3.3.17 Assignment:

Assignment means changing the value of something. To form an assignment statement, specify what you want to change, type the equal sign, then type a formula. Whenever the TELSEC® encounters an assignment statement, it computes the current value of the formula and stores that value in the element you specified. (E.g. element = formula.)

Example: `RLY.1=UIN.4`

In the example above, the RLY.1 will go ON and OFF as the UIN.4 goes ON and OFF. The SET statement is also an assignment statement; it has the same effect as the form shown above.

3.3.18 SET element TO formula:

You can assign values to many of the elements presented in Section 2.3 DEFINE Command:. Inputs cannot be assigned a new value. Their value is always a number or ON/OFF that is the signal the TELSEC® currently reads at that input.

Relays can be assigned a value of ON or OFF. Doing so turns the relay ON or OFF, just as the TURN statement does (see Section 3.11). These statements are equivalent:

```
RLY.1 = ON
SET RLY.1 TO ON
TURN ON RLY.1
```

3.3.19 Variables:

Variables exist for the purpose of receiving values in assignments. Instead of having a complicated formula in a single assignment statement, you can use several assignment statements with shorter formulas. Temporary variables hold the partial results.

Two equations can use variables to communicate. For example, one equation can put a certain value in a variable (e.g. VAR.GOWILD = 100) to tell another to start working. The other equation tests the variable using the IF/THEN technique discussed in Section 3.3.9. It typically resets the variable once it has sensed the value it was looking for...

`IF (VAR.GOWILD = 100) THEN VAR.GOWILD = 0,` ...and then continues with other statements.

3.3.20 Setpoints:

Setpoints act exactly the same way variables do with one exception: a user can modify a setpoint's value using the TELSEC® front panel. Setpoints can be used to allow the user to modify his environment easily. Suppose you wanted to control an air conditioning unit connected to RLY.1. The standard setpoint for the room is 70 degrees Fahrenheit. You also have a temperature sensor connected to UIN.1 that monitors the room temperature. Your setpoint definition and equation might look like this:

```
DEFINE AC1SPT = SPT.1 70
DEFINE DELTA = SPT.1 2
DEFINE AC1CTL = EQU.10
IF  UIN.SPACETMP > SPT.AC1SPT THEN TURN ON RLY.1
ELSE
IF  UIN.SPACETMP < SPT.AC1SPT – SPT.DELTA
THEN TURN OFF RLY.1
```

In this simple form, you can see that the AC will turn on if the temperature is greater than the set point. Now suppose it is an extremely hot day and the people in the room wish the air to go on at a lower temperature. Instead of calling up the TELSEC® and making a change to EQU.10, they can go up to the front panel and modify SPT.AC1SPT slightly. One other point to remember is that the formulas have no control over what values users may enter through the front panel. Suppose someone modifies the setpoint to 30 degrees F. The room will get extremely cold! Therefore, this potential must be taken into account when writing equations to impose limits on the setpoints. This must be done before the setpoint is used in another equation. Here is the new example:

```
DEFINE AC1VARH = VAR.1 75 (High setting)
DEFINE AC1VARL = VAR.2 65 (Low setting)
DEFINE AC1SPT = SPT.1 70
DEFINE AC1LMT = EQU.9
SPT.1 = MIN(VAR.1,(MAX(VAR.2,SPT.1)))

DEFINE AC1CTL = EQU.10
IF  UIN.ROOMTEMP > SPT.AC1SPT THEN TURN ON RLY.1
ELSE TURN OFF RLY.1
```

Notice that equation 9 limits the setpoint value between the two variables. This method controls the range that a user can modify a setpoint. The TELSEC® will then run equation 10 with the corrected setpoint value.

3.3.21 Additional ways to change values:

The values of specified numeric elements can be changed using the `INCREMENT` or `DECREMENT` statements.

3.3.21.1 Increment Statement

The `INCREMENT` statement increases the value of a specified numeric element. The two statement forms below have an identical effect; the `INCREMENT` form is legal only where the `SET` form would be legal:

```
INCREMENT element1 BY element2
SET element1 TO element1 + element2
```

If the word '`BY`' and the second element are omitted, the `INCREMENT` statement simply adds one (1) to the value of the number element specified.

3.3.21.2 Decrement Statement

The `DECREMENT` statement decreases the value of a specified numeric element. The two statement forms below have an identical effect; the `DECREMENT` form is legal only where the `SET` form would be legal:

```
DECREMENT element1 BY element2
SET element1 TO element1 - element2
```

Examples:
INCREMENT VAR.1
DECREMENT SPT.1 BY 2
SET VAR.TEMPSPT TO 70

If word '`BY`' and the second element are omitted, the `DECREMENT` statement simply subtracts one (1) from the value of the numeric element specified.

3.3.21.3 Clear Statement

The `CLEAR` statement sets an element's value back to zero. The exceptions are variables and setpoints where the command resets the variable to its initial defined value. The two statement forms below have an identical effect; the `CLEAR` form is legal only where the `SET` form would be legal:

```
CLEAR element
SET element TO 0
```

The CLEAR statement is typically used to reset the values of point statistics. If you CLEAR RLY.# or UIN.# all associated timers and counters are set to zero.

Examples:
```
CLEAR VAR.HOWMANY
CLEAR SPT.COUNTER
CLEAR REC.4
CLEAR RTM.4
CLEAR RLY.1
CLEAR UIN.DIGSWTCH
```

3.4 Timers and Counters

The TELSEC® has a few special point types. These are the digital timer and counter points. Each digital output and digitally defined input carries these points. You may use these points in your equations to calculate various things: run time for equipment, pulse accumulation, equipment maintenance, etc. The following sections 3.4.1 to 3.4.2 provide descriptions for each point.

3.4.1 Digital Output Points (RLY):

RAC Accumulated ON time (counts total on time).
RNO Interval ON time (resets to 0 when RLY goes on and starts counting).
RNF Interval OFF time (resets to 0 when RLY goes off and starts counting).
RTM RLY event timer (time since timers were cleared).
REC RLY event counter (increments when RLY changes State).

3.4.2 Digitally defined Input Points (UIN):

DAC Accumulated ON time (counts total on time).
DNO Interval ON time (resets to 0 when UIN goes on and counts on time).
DNF Interval OFF time (resets to 0 when UIN goes off and counts on time).
DTM Digital event timer (time since timers were cleared).
DEC Digital event counter (increments when UIN changes State).

3.5 Send Command:

The SEND statement sends point values to the front panel display. When you enter a SEND statement, you specify what to send using this form:

```
SEND <point.#>
```

You may specify any TELSEC® point type except ANM. The TELSEC® will update the front panel with a new message. If no new message exists, the current message will continue on the display. If there is another message to be displayed, the current message will be displaced by the new message. The front display has a thirty two message buffer and will round robin each message or point to the screen. Once you send a point to the screen, it will continue to be displayed until you issue a SEND <point> CLEAR statement.

If you SEND MSG to the front panel, the name of the MSG will not appear. The actual MSG text will appear on the front panel display. The first sixteen characters show on the first line and characters 17-32 show on the second line.

Example:
DEF DISPLAY = EQU 1
SEND UIN 1, SEND UIN 2, SEND UIN 3,
IF UIN.4 = ON THEN SEND MSG.4
ELSE
SEND MSG.4 CLEAR

3.6 Alarm Equations:

The ALARM statement places point values into the ALARM log and causes an alarm message to be generated. When an ALARM statement is entered, specify what to send using this form:

```
ALARM <point.#> <severity> [list of ANM]
```

<point.#>: Any point within the TELSEC® such as UIN, RLY, SPT, MSG, ETC.

<severity>:
CR Critical alarm.
MJ Major alarm.
MN Minor alarm.
CLEAR Alarm condition has cleared.
NONE Alarm condition is status only.

[list of ANM]: Optional - Like the digital alarms in section 2.3.14.3, you can specify which alarm numbers to dial when the alarm is generated. All active numbers will be dialed if you do not specify.

Once an entry has been placed into the alarm log, the TELSEC® will take appropriate action. If alarm phone numbers (ANMs) have been defined, the TELSEC® will wait for the modem to become available and then attempt to call out the alarm. Once the TELSEC® makes the alarm callout connection, it will dump all alarms not yet sent in the alarm log. The information contained in an alarm callout includes system TID, the current date and time and the point information specified to alarm. This information will be sent in a TL1 formatted message.

Example of an alarm equation:

```
DEF HITEMP = EQU 1
IF UIN.TEMP2 > VAR.SETPOINT THEN
ALARM UIN.TEMP2 MJ, WAIT UNTIL
UIN.TEMP2 < VAR.SETPOINT, ALARM UIN.2 CLEAR
```

Notice that a WAIT UNTIL statement ends this equation block. This will keep this equation from continuing to enter ALARM statements in the alarm log each time this equation is processed. The equation will now only process once and WAIT UNTIL the alarm condition has gone away before it processes the rest of the equation and goes back to the beginning again.

Example of an alarm equation with specify which ANM's to use:

```
DEF HITEMP = EQU 1
IF UIN.TEMP2 > VAR.SETPOINT THEN
ALARM UIN.TEMP2 MJ ANM 1 ANM 3, WAIT UNTIL
UIN.TEMP2 < VAR.SETPOINT, ALARM UIN.2 CLEAR ANM 1 ANM 3
```

If the TELSEC® can not complete the alarm callout, it will wait the delay time defined in the ANM and then attempt the call again. This ensures alarms are not missed due to busy or noisy phone lines.

3.7 Freeform Logging:

The LOG statement makes an entry into the freeform log. When you type a LOG statement, you must specify the element to store using this form:

```
LOG <point.#>
```

The TELSEC® records the current value of the specified element in the log, noting the current date and time. The log also keeps an indication of the name of the element you logged. You may review this information using the REVIEW LOG FREE command (see section **Error! Reference source not found.**). The freeform log contains approximately 800 entries arranged in a circular queue. If the log is full when the LOG statement processes, the current entry causes the oldest entry to scroll out of the log.

Example:

```
DEF LOGAVG = EQU 1
VAR.1 = (AVG(UIN.1,UIN.5),LOG VAR.1, WAIT 10:00
```

This equation sets variable 1 equal to the average reading of input 1 and 5 and then logs variable 1. The equation then waits 10 minutes before running again.

3.8 Advanced Equation Functions:

3.8.1 Enable/Disable:

When an equation is first defined it becomes enabled. This means it is set to operate continually. (Section 3.9.3 discusses the exact sequence of activities.) An equation can be disabled or enabled. Disabling an equation takes it out of service. The TELSEC® suspends all work on the disabled equation for as long as it is disabled.

One equation can disable or enable another equation, or disable itself. A restart or power failure always re-enables all equations. In addition, a restart or power failure re-starts all equations at the beginning.

The DISABLE statement disables an equation. After typing DISABLE, specify the equation to disable:

```
DISABLE equation
```

The equation is out of service and has no further effect on the TELSEC® until the next time an equation or an operator ENABLEs it (see below), restarts the TELSEC®, or if the power fails. If the specified equation was already out of service, the DISABLE statement has no effect.

Having an equation disable itself is a useful programming technique. For instance, equation number 1 can specify a power failure recovery sequence. The equation ends by disabling itself.

The ENABLE statement enables an equation. ENABLE follows the same form as DISABLE:

```
ENABLE equation
```

The equation resumes operation starting where it left off when you disabled it. If the equation was in a WAIT statement when you disabled it, it resumes its wait until the specified time is up or

the specified condition is TRUE. The equation disregards time that passed while it was disabled, or conditions it is waiting for that were TRUE only while the equation was disabled. Using ENABLE on an equation that was already enabled has no effect.

Suppose equation number 1 specifies a power failure recovery sequence, as mentioned above. Then any other equation could make this sequence happen at any time, using this statement:
`ENABLE EQU.1`

3.8.2 Returning Values:

Section 3.3.17 gives an example of an equation assigning a value to a variable so that another equation will see the value and do something. This is an example of communication between equations.

Each equation has a variable associated with it. You specify it by typing the symbol EQU. followed by the equation's number. This can be used anywhere that a numeric element is legal.

The RETURN statement is used by an equation to specify a value for that equation's variable. Any equation can then read the specified value, using the symbol EQU. as described above. Type RETURN and then a numeric formula:

RETURN [(number) or (value of a formula)]
The TELSEC® computes the current value of that formula and makes it the value of the equation.

Unlike many other programming languages, the RETURN statement on the TELSEC® does not change the order in which the TELSEC® performs statements; it does not keep the statement following RETURN from being reached; and there is no limit on the number of RETURN statements you can use in a single equation. Whenever another equation uses the EQU. symbol, it sees the number value most recently computed by a RETURN statement inside the specified equation.

For example, say equation number 20 wants to pass a number value for use inside equation number 23. One of the statements inside equation 20 is: `RETURN 100`

Equation 23 can make some number of statements conditional, so they won't run until equation 20 gives this signal. Inside equation 23, you might type:

```
IF EQU.20 = 100 THEN ...
```

3.8.3 Multiple Conditionals:

In the sequence of statements that follows the word 'THEN,' there can be another IF test. When several IF/THEN pairs are used in a single equation, pay careful attention to the exact outcome. Unless the word 'ELSE' is used, the second IF/THEN test becomes just one of the statements in the list. Therefore, the TELSEC® only makes the second test if the first one was true. Consider this equation:

```
IF UIN.1 > 72 THEN TURN ON RLY.1,
IF UIN.2 > 72 THEN TURN ON RLY.2
```

This looks like a case where two relays turn on independently based on two analog inputs (say, temperatures). But this is not how the equation works. The first test controls the entire equation; the TELSEC® doesn't even compare `UIN.2 > 72` unless it found `UIN.1 > 72` was TRUE and turned on RLY.1.

The conditionals presented so far are unbounded. That is, the first conditional used in an equation makes the rest of the equation conditional. The section below, 3.8.4, presents bounded conditionals. Use them to limit the range of an IF statement's effects, so several unrelated events in a single equation can be controlled.

3.8.4 Statement Blocks:

A statement block is a sequence of statements preceded by DO and followed by END. Use a statement block anywhere a single statement can be used (except inside another statement block). Follows is an example of a statement block:

```
DO TURN ON RLY.1, TURN ON RLY.2, SEND MSG.1 END
```

There may be a series of statements between DO and END, separated by commas. But do not type a comma immediately before the word END. A comma may be needed before the word 'DO' or after the word 'END.' Imagine that the entire DO...END range were replaced with a single statement. Supply commas if that single statement would need them before or after it—for instance, if other statements or statement blocks immediately precede or follow the DO...END range.

3.8.5 Nested IFs:

Statement blocks are also useful in the same way parentheses are, when one IF/THEN/ELSE group is typed within another. For example:

```
DEF ECONMZER = EQU.30
IF UIN.OUTAIR  < 70 THEN DO
IF UIN.ROOMTEMP > SPT.COOL FOR 1:00 THEN TURN ON RLY.FREECOOL
ELSE
IF UIN.ROOMTEMP< SPT.C00L FOR 1:00 THEN TURN OFF RLY.FREECOOL
END
```

In this example, the equation will first check to see if the outside air is less than 70. If it is TRUE then it will process the IF statements after the DO command and pick the first one that is TRUE.

The spacing makes it clear to a reader what this equation is meant to do. However, without the DO/END, it would not be clear if the word 'ELSE' applies to the first IF or to the second IF. Nested DO's (DO statements within DO statements) are illegal. The word 'END' must follow the word 'DO' before the word 'DO' can appear again to start another statement block.

3.8.6 RATE and AVERAGE RATE function:

The RATE function provides a very rough idea of how quickly the value of a formula is changing over time. There are three parameters inside parentheses: the formula to test, a time interval, and a number value.

```
RATE (formula, interval, number)
AVERAGE RATE (formula, interval, number)
```

Basically, the value the RATE function produces is a number that tells you how quickly the formula is changing over the specified interval. The third parameter is a "standard value." The RATE function simply provides this number if the specified interval has not yet occurred.

How the RATE function works:

Step One When 'RATE' is first typed into an equation, the TELSEC® sets its value equal to the "standard value." The TELSEC® computes the current value of the formula and remembers it.

Step Two When a specified interval has passed, the TELSEC® again computes the formula's value. The value of the RATE function becomes the difference between this and the remembered value of the formula. The TELSEC® remembers the formula's new value for future use.

Step Three Step Two repeats at the specified interval.

The TELSEC® conducts the computations above even in an equation where the RATE function is not currently being reached. (However, if and equation is DISABLEd, these computations end.)

Follows is an example measuring the rate at which a temperature is changing:

```
RATE(UIN.TEMP2, 0:10:00, 0)
```

What the example above actually measures is the net change of the specified temperature during a recent period of ten minutes. In this example, during the first ten-minute period, RATE has the standard value of 0, claiming there was no change in temperature.

The AVERAGE RATE will compute the average rate of change of the formula's value. Each time the RATE function evaluates, the stored value will be the average between the last calculated value and the new calculated value.

3.8.7 Pulse Command:

The pulse command allows the TELSEC® to turn on or off a relay quickly with very detailed timing. The Pulses are accurate to the nearest 10th of a second for the first 16 relays. Do not use the pulse command with any expansion board output since the timing will be affected by the amount of traffic on the bus. An example of using this function would be for connecting to pulse to analog transducers where sending contact closure for a specific period will change the amount of output that occurs such as with variable speed motors or variable position dampers.

The command format is `PULSE <ON|OFF> RLY <#> FOR <time in 10ths>`
Where:
`<ON|OFF>:` Turn the Relay either ON or OFF
`<#>:` The relay #
`<time in 10ths>:` Time in tenths. A 1 = 1/10, 10 = 1 second etc.

Example:
`IF <statement> THEN PULSE ON RLY.1 FOR 1,WAIT 5:00`
If the `<statement>` is true then the relay will pulse on for 1/10 of a second and then the equation will wait 5 minutes.

`IF <statement> THEN PULSE ON RLY.1 FOR VAR.1,WAIT 5:00`
If the `<statement>` is true then the relay will pulse on for the value of variable 1 and then the equation will wait 5 minutes. Another equation can be used to change the value of variable 1.

3.8.8 COM Port Status:

The status of the two communications ports can be monitored by equations and then additional alarming or control functions can be accomplished based on the status. The point names are COM1 for the modem and COM2 for the serial port. The portsl have the following value based on the status of the Data Carrier Detect (DCD) and Data Set Ready (DSR) signals:

Value	DSR BIT	DCD BIT	Meaning
0	0	0	DCD & DSR are not present
1	0	1	DCD present, but not DSR
2	1	0	DSR present, but not DCD
3	1	1	DSR and DCD present

Example:

```
DEF CK_COM1 = EQU 1
IF COM1 = 0 THEN ALARM MSG.1 EQPTSA MJ,WAIT UNTIL COM1 = 3
```

This equation will see if the COM1 has lost connectivity and then alarm a message saying the connection to COM1 is down. Once COM1 is active again, the equation will send a Clear alarm message.

3.9 Shortcuts:

The TELSEC®'s large number of operators, functions, and statements provide many different ways of solving a problem or specifying programmed action. In fact, there are usually several ways to write something that will have an identical effect. There is no one "right" way to write an equation. Different ways to write the same thing can be compared by asking these questions:

- Does the equation work as desired? (In every situation?)
- Is it as readable as it could be?
- Are there any wasted steps?
- How much of the TELSEC®'s memory does it occupy? (That is, could the equation be written more briefly?)

An efficient equation makes the TELSEC® do no more computing than necessary, it has the shortest possible form, and it is readable. This last attribute produces efficiency by saving time when you or someone else must change it.

Trade-offs must sometimes be made. For example, an equation that is longer than necessary may be written to emphasize what its function is or make it easier to change. An example of this is to have all conditionals bounded.

Use the power of digital elements to make an equation shorter and more elegant. Digital elements are TRUE/FALSE elements, such as comparisons, schedules, or digital inputs or relays, which can be ON or OFF. In the TELSEC®, TRUE is equivalent to ON (and to the number value 1); FALSE is the same as OFF (or 0).

Consider the following equation:

```
IF UIN.OCCUPIED THEN TURN ON RLY.LIGHTS
ELSE TURN OFF RLY.LIGHTS
```

The equation above checks whether UIN.OCCUPIED is TRUE (ON) and moves ON to RLY.LIGHTS. If UIN.OCCUPIED is FALSE (OFF), it moves OFF to RLY.LIGHTS. In both cases, the desired effect is to move the value of UIN.OCCUPIED directly to RLY.LIGHTS.

Write this directly:
```
RLY.LIGHTS = UIN.OCCUPIED
```

This puts the relay in sync with the input.

A more general statement of this idea is: Write a formula that is true in all cases, instead of using `IF/THEN` to test each case at a time.

Follows is an equation that computes the lower of two temperatures:

```
IF UIN.TEMP1 < UIN.TEMP2 THEN VAR.LOWER = UIN.TEMP1
ELSE VAR.LOWER = UIN.TEMP2
```

A much easier way to do the same thing is to use the built-in `MIN` function to get the minimum temperature:

```
VAR.LOWER = MIN(UIN.TEMP1, UIN.TEMP2)
```

Once it has been written in this way, the variable may not need to be used at all; the `MIN` function itself can be used in place of the variable.

3.9.1 Avoiding Repeat Effects:

Keep in mind that the TELSEC® runs each equation continually. If a WAIT statement is reached, the TELSEC® will suspend operation on that equation until the conditions are achieved. You must consider whether an equation will produce one effect or many effects, and whether these effects are desired.

3.9.2 Repeat effects may not matter.

Suppose the goal of a certain equation is to put `RLY.LIGHTS` in the correct state. This equation can be written so that the TELSEC® will either reach `TURN ON RLY.LIGHTS` or `TURN OFF RLY.LIGHTS` every time. In this case, it does not matter how often the equation runs, as long as it does the right thing each time. This is because turning on the lights has no effect if they're already on.

If the equation is written to complete a task such as pulsing a relay, making a phone call, logging, sending messages, etc, it is important to ensure the action only occurs once per occurrence.

The `WAIT UNTIL FALSE` statement is a typical way to produce a single effect. If an equation starts with an `IF/THEN` test, it may end with the same test, preceded by `WAIT UNTIL NOT`. This keeps the TELSEC® from starting the equation again until the situation that made the TELSEC® initiate the equation has ceased. Follows is an example of this form:

```
IF UIN.ALARMED THEN statement, statement,
statement, statement,
WAIT UNTIL NOT UIN.ALARMED
```

This equation does four things if an alarm button is pressed. The `WAIT` statement at the end waits until the button is released. If you pushed the button again, the four statements would run again. But if you didn't include the `WAIT` statement, the TELSEC® would do the four statements as many times as it could until you let up on the button.

Using a DISABLE statement is another way to produce a one-time equation. The last statement in the equation is a DISABLE statement that refers to the equation it is in. This means that, when the TELSEC® runs completely through the equation, it finishes by taking the equation out of service.

Another equation could use the ENABLE statement to make the first equation run again, one time. Normally, equations that disable themselves run once after a restart or power failure, since the TELSEC® re-enables all equations on startup.

3.9.3 Detailed Timing:

The TELSEC® runs any equation that isn't disabled by obeying each of the statements in the sequence in which they appear. The IF statement makes the TELSEC® skip over some statements in certain cases. Some statements, such as WAIT, make the TELSEC® stop working on that equation, though it continues to run other equations.

If it's time for the TELSEC® to run an equation (see below), but that equation is waiting, then instead of going to the start of the equation, the TELSEC® goes to the point where it left off and checks whether the equation can stop waiting.

When the TELSEC® reaches the end of an equation (or the last statement it's allowed to obey, because of IF statements), then it is done with that equation and goes on to another one (see below). The next time it runs the original equation, it goes back to the start.

If an equation starts with an IF statement detecting an unusual condition, the TELSEC® makes the specified test every time it runs the equation. Typically, it gets the value FALSE, decides there's nothing else it can do in this equation, and stops running it until the next time.

Including the RATE function in an equation makes certain computations occur every time the TELSEC® runs a specific equation, even if the equation is waiting and can't continue. These computations will not occur, however, if the equation is disabled.

After a cold start, such as that during the TELSEC® installation, there are no equations, so none will run. When a new equation is defined, it becomes enabled. This means the TELSEC® runs it at least once. The equation may take itself out of service. This would still produce a one-time effect, unless another equation disabled it before the TELSEC® reached it.

After a reset, such as restoration of power, to the TELSEC®, the TELSEC® automatically enables all equations. The equations are then processed and run in order expeditiously.

3.9.4 Use of Memory:

When an equation is defined, the TELSEC® will report how many "bytes" of memory the equation requires. No equation is allowed fill more than 256 bytes. When an equation that is too complex is entered, the TELSEC® will alert the programmer. The equation can be simplified or variables can be used to pass information to other equations, so that some of the computation can take place there. The TELSEC® "byte" report will alert the programmer when an equation being entered is getting close to the 256-byte limit. Equations obtain memory in 32-byte sections. If the TELSEC® reports an equation used 37 bytes, you should recognize that the equation actually used two 32-byte sections, and actually removed 64 bytes from the total available memory in the TELSEC®.

3.9.5 Checksums:

The TELSEC® system will do a check sum test on all equations when the unit comes up from a power failure or when the system is reset. This test ensures the integrity of programmed control strategies. Any equation that does not have the same check sum as that prior to the power fail or reset, will be disabled and will not perform any control or monitoring functions. The system will automatically insert an alarm in the alarm log stating that a failure for an equation has occurred. If an alarm phone number is programmed, the system will call out the alarm alerting the user to a failure. The equation will also return a value of -1 which can be seen with the `REVIEW EQU` command. You can also write strategies to monitor critical equations for failure and take a corresponding action. An example would be as follows:

```
IF MIN (EQU.1, EQU.2, EQU.3, EQU.4, EQU.5, EQU.6, EQU.7, EQU.8, EQU.9,
EQU.10) = -1 THEN TURN ON RLY.ALARM
ELSE
TURN OFF RLY.ALARM
```

If an equation has a check sum failure, you can correct the strategy by transmitting the equation. The entire controller does not have to be reprogrammed.

3.10 Establishing Criteria to Write Equations:

The parameters of how the facility is to be controlled and monitored must be established. The decisions make up the "Criteria" or "Sequence of Events" for your facility. Writing equations is taking your criteria and putting into a syntax that the TELSEC® can interpret. Use the following steps to program your system. Although the program can be loaded in any order, it is best to start with defining the physical inputs and outputs, and then complete the rest of the programming. This will establish a logical progression when writing the program.

Inputs: We have generated the following definitions. Note the names chosen for each of the inputs.

```
DEFINE ROOMTEMP = UIN.1 TEMPF 0 MSG.1 LOG AVG 30
DEFINE SMOKE  = UIN.3 DIG CR 5 MSG.3 LOG
DEFINE TOXIC  = UIN.4 DIG MJ 60 MSG.4 LOG
```

These definition lines will configure the TELSEC® for our application. Of course, the individual sensor wires must be terminated at the proper TELSEC® input terminal block location.

Outputs:

```
DEFINE VENT_FAN = RLY.1 OFF STAGED ENERGON
DEFINE COOL_1  = RLY.2 OFF STAGED ENERGON
DEFINE COOL_2  = RLY.3 ON STAGED ENERGON
DEFINE HEATING = RLY.4 ON STAGED ENERGON
```

We chose STAGED for all RLYs because none of our devices need critical ON/OFF timing.

Control: We have the inputs and outputs defined, so the next step is to provide the control interface.

We now define a few setpoints to use in our equations. Using setpoints (SPTs) allows us to easily change our operating parameters later. It also gives a front panel user the opportunity to change the settings.

```
DEFINE COOL_SP = SPT.1 78
DEFINE CL2DELTA = SPT.2 5
DEFINE HEAT_SP  = SPT.3 65
DEFINE HIGHTEMP = SPT 4 90
```

A message will be defined so it can be sent to the display when the alarm occurs.

```
DEF HIGHTEMP = MSG.1 'HIGH TEMP ALARM IN ROOM'
```

Equation One: Its purpose is to control RLY.2 which is the air conditioner Stage 1.

```
DEFINE COOL__1 = EQU.1
IF UIN 3 = ON THEN TURN OFF RLY.2
ELSE
IF UIN.1 > SPT 1 FOR 2:00 THEN TURN ON RLY.2
ELSE
IF UIN.1 < SPT 1 FOR 2:00 THEN TURN OFF RLY.2
```

Notice there are three (3) IF..THEN..ELSE statements in the equation. This corresponds to the amount of tasks this equation must handle. The three tasks, in priority, are as follow:

Task 1 Turn off the air conditioning if the smoke detector is on.

Task 2 Turn on the AC if the temperature is greater than the current setpoint for two minutes.

Task 3 Turn off the AC if the temperature is less than the current setpoint for two minutes.

Equation Two: The control strategy for the second air conditioner can be programmed as follows:

```
DEFINE COOL__2 = EQU.2
IF UIN 3 = ON THEN TURN OFF RLY.3
ELSE
IF UIN.1 > SPT 1 + SPT 2 FOR 2:00 THEN TURN ON RLY.3
ELSE
IF UIN.1 < SPT 1 FOR 2:00 THEN TURN OFF RLY.3
```

Again, there are three (3) `IF..THEN..ELSE` statements in the equation. This corresponds to the number of tasks this equation must handle. The three tasks are as follow:

Task 1 Turn off the air conditioning if the smoke detector is on.

Task 2 Turn on the AC if the temperature is greater than the current setpoint plus the stage 2 delta for two minutes.

Task 3 Turn off the AC if the temperature is less than the current setpoint for two minutes.

Equation Three: The heating can be programmed as follows:

```
DEFINE HEATER = EQU.3
IF UIN 3 = ON THEN TURN OFF RLY.3
ELSE
IF UIN.1 < SPT 1 FOR 2:00 THEN TURN ON RLY.3
ELSE
IF UIN.1 > SPT 1 FOR 2:00 THEN TURN OFF RLY.3
```

Notice there are three (3) `IF..THEN..ELSE` statements in the equation. This corresponds to the number of tasks this equation must process. The three tasks, in priority, are as follow:

Task 1 Turn off the heater if the smoke detector is on.

Task 2 Turn on the heater if the temperature is less than the current setpoint for two minutes.

Task 3 Turn off the heater if the temperature is greater than the current setpoint for two minutes.

Equation Four: We have completed controlling the AC and heating units for the facility and can now work on alarm equations:

```
DEFINE HI_TEMP = EQU 4
IF UIN.1 > SPT 4 FOR 5:00 THEN ALARM UIN.1 MJ,
SEND MSG.1,WAIT UNTIL UIN.1 < SPT.4,
ALARM UIN.1 CLEAR, SEND MSG.1 CLEAR
```

Here the System generates an alarm message for high temperature when the temperature is greater than the high temperature alarm setpoint and sends the message to the front display. The system will then send a clear message once the input is below the alarm setpoint and stop (clear) sending the message to the front display.

Equation Five: Equation five is a special equation. By using SPTs in our equations, we have allowed front panel users to modify the setpoints. We could have used variables (VARs) and not allow the front panel user access to our equations. We opted not to do so in this case to demonstrate front panel access within limits. The TELSEC® built-in functions of MIN and MAX are utilized here to limit the range a front panel user can modify a setpoint.

```
DEFINE STPTLMTS = EQU.5
SPT.1 = (MAX 70, (MIN (SPT.1, 80))),
SPT.2 = (MAX 0, (MIN (SPT.2, 5))),
SPT.3 = (MAX 55, (MIN (SPT.3, 70)))
```

Equation 5 recalculates each setpoint based on the two programmed limits. In this way, if a front panel user decides to try to change the temperature setpoint (`SPT.1`) to 64 degrees because he is too hot, the TELSEC® will reset the setpoint to a value within the limits. In our example, the two limits for SPT.1 are 70 and 80. The front panel user can move SPT.1 freely between these values giving him the flexibility of modifying the setpoint somewhat.

Let's say he tries to bring `SPT.1` up to 82 degrees because he is too cold. The TELSEC® will evaluate the MIN (`SPT.1, 80`) part of the equation first because of the parenthesis inserted. The function will compare the current value of `SPT.1` (or 82) with a limit of 80. The function will return the minimum (`MIN`) of these two values or 80. `SPT.1` has now been limited to 80. Personnel can be prevented from wasting energy in this way.

Equation Six: Equation six performs a simple function. It simply scrolls the analog values across the front panel display to be seen by anyone wanting to know the current conditions.

```
DEFINE DISPLAY  = EQU.6
SEND UIN.1SEND UIN.2,SEND UIN.3
```

3.11 Uploading Programs:

Once all TELSEC® programming and equation definitions are complete, the information must be uploaded to the TELSEC®. Entering programs manually is very time consuming. Most users write all programs using a computer and word processing software first. The programming is entered in ASCII text format. Be sure to save the program file in a non-document mode (in .TXT mode). You can use the forward slash (/) as the first character on a line to create comments in your program file. Once the programs are complete, communication software is used to quickly upload the program. Prior to uploading the program, it is recommended that you first set the system clock using the SET CLOCK command (refer to section 2.4.2). The TELSEC® supports two methods for uploading programs the first and preferred method is using Xmodem protocol:

3.11.1 Xmodem File Transfer

Xmodem is the preferred method of sending programs since it provides error checking and will notify you of any errors in your program. Most communications programs support Xmodem transfer. In the setup for Xmodem on your communications program choose "used relaxed timing" if you have that option. To start an upload, first issue the command SET PRO PRO to the TELSEC®. The system will respond with: **START XMODEM XFR..**. you have up to 1 minute to start sending the program or the system will time out and abort the transfer process. Send the program to the TELSEC® using the Xmodem protocol. Once the file is received, the system will start outputting progress dots (….) as it processes each line of your program. If the program is accept completely then the system will respond with a message saying "NO ERRORS". Otherwise the system will respond with an error message listing the line numbers where the error(s) was found

Example:
`COULD NOT SYNC –` The TELSEC® could not sync with your Telnet client or terminal program.

```
ERRORS FOR LINES:
3 5 10 20
```
Use your Text editor to correct problems with lines 3,5,10 and 20

```
NO ERRORS
```
No errors were found in your program and it was accepted successfully.

3.11.2 ASCII Text Transfer

The TELSEC® system accepts ASCII downloads using software flow control. All of your provisioning commands can be saved to an ASCII Text file and then loaded via ProComm to the controller. Make sure your ASCII download settings are set up as follows:

1. STRIP LF on upload
2. 5 millisecond character delay
3. Don't expand blank lines (ProComm default is to expand blank lines)
4. Use software flow control (XON/XOFF)

Note: you must be logged on with a password in order for the system to take your program files.

The system will respond with OK after each program line. An error message will be displayed if a line is not accepted. When this occurs, correct the line and then either retransmit the file or copy and paste the correction to the system by using the Windows copy and paste commands.

3.11.3 Saving Programs to Non-Volatile Memory

All programming when loaded to the system is stored in battery backed up RAM. The program can be written to the non-volatile FLASH memory once you have completed your programming. To store all programs to FLASH, Type the command **SET PROGRAM DEFINE** and the system will respond with a message stating it is writing to the FLASH.

Chapter 4 – Access Control

4.1 Overview

The TELSEC® has an integrated Access Control port for key/card code access control of the facility. The hardware supports any card swipe or proximity reader using the industry standard Wiegand format. With a maximum database of 600 cards, the TELSEC® can handle large personnel requirements. The hardware also supports digital feedback from the door to alert during illegal entry and door ajar conditions. Quest also offers custom card formats tailored to specific applications. Contact your authorized Quest representative for more details. Quest also offers a peripheral module that will allow up to four card readers and control of four doors.

4.2 Using the Access Control System

We will discuss the software portion of the TELSEC® Access Control system here. For Hardware installation, see the Installation Manual. Once the card reader and door have been wired into the TELSEC®, you are ready for programming.

4.2.1 Define the feedback digital point

(Optional): If the door closure is wired to the TELSEC® to provide feedback, this input must be defined as `DIGITAL`. This is done using the following command line (see section 2.3.14.3):

```
DEFINE DOORSTAT = UIN.# DIGITAL LOG
```

The # is the `UIN` point where feedback digital is landed. The name 'DOORSTAT' can be any eight-character name.

4.2.2 Define your Digital Output:

Typically you will define your relay output with this command line (section 2.3.9):

```
DEFINE DOORRLY = RLY.# OFF IMMEDIATE ENERGON
```

Where # is the digital output point where the door opening circuit is landed. If the output is required to have inverted logic, refer to the `DEFINE RLY` portion of your TELSEC® User's Manual for more information.

4.2.3 Define the DOR point:

Define the DOR point to correlate a valid card presented to a reader with the control of a particular output that is actuating the door mechanism. This point returns an `ON` or `OFF` state using the `CARD` information which will be defined later. The syntax for this command is:

```
DEFINE [<NAME> =] DOR.<#> <RLY.#> <SECONDS OPEN> [<DIGITAL FEEDBACK>
<SECONDS TO CLOSE>]
```

`<NAME>:` A unique, user defined point name.
`<#>:` Select DOR point 1-4. Note you must have the 4 port door peripheral present to support more than one door.
`<RLY.#>:` The digital output controlling the door solenoid.

`<SECONDS OPEN>:` This field contains the time in seconds that the door digital output will energize during a valid access condition. Once a valid card is recognized, the output relay will energize for this time allowing the cardholder to open the door. Valid seconds are 1 to 59.

<DIGITAL FEEDBACK> (optional): This field contains the UIN.# of the feedback digital used to sense actual door opening and closure. If a feedback digital was not used for installation, leave this field blank. The feedback digital will cancel the DOR ON command once the door is open so it can turn off quicker that the value in the <SECONDS OPEN> field.

<SECONDS TO CLOSE> (optional):
This field contains an amount of time in seconds that the TELSEC® waits before alarming the door is ajar. If you have defined a feedback digital for your door, you must enter a number between 1 and 120 in this field. If no digital was defined, leave this field blank. A message will be entered in the access control log if the door is left ajar for longer than this time delay.

Note: An ACCESS-level password is necessary to use the DEFINE DOR command.

4.2.4 Define valid access cards:

The syntax to define your cards is:

```
DEFINE CARD <# | NEXT> <CARD NUMBER> [<TOD.#>]
DEFINE CARD <CARD ID> [TOD.#] [DOR.1,DOR.2,DOR.3,DOR.4] ["<NAME>"]
```

<CARD ID>: This field contains the card number (sometimes referred to as 'Badge ID' or 'Card ID') of the card you wish to have access to your door. The TELSEC® will accept numbers from 1 to 1048575 but the maximum number of cards in the system is 600.

[TOD.#] (optional): This field contains an optional time schedule number used to grant access only during valid time periods. If the application requires this option, refer to the DEFINE TOD section 2.3.13 of the TELSEC® User's Manual and input the special time period criteria. The TOD.# may now be entered in this field and this particular card number will only be given access during an ON state of the TOD.

[DOR.1, DOR.2, DOR.3, DOR.4]: you can specify which DOR point will work with this card if you have more than one door wired into your system. If you do not specify a DOR point, then the TELSEC® system will automatically make the card valid for all DOR points.

["<NAME>"]: The TELSEC® will allow you to attach a 16 character name to the card. The name can contain spaces and must be defined between quotation marks.

Example:
DEF CARD 300 "JOHN Q TECH"
Card code 300 is entered in the database and assigned the name JOHN Q TECH. There is not limit based on time of day and this card has access to all available doors.

DEF CARD 1050 TOD 1 DOR 1 "CLEANING CREW"
Card code 1050 is entered into the database and assigned the name CLEANING CREW. This code is limited to when time of day schedule 1 is active and will only work on the first door.

After completing and changes you will need to save your changes to flash with the SET PRO DEF command.

4.2.5 Setting site code and bit format:

Determine the site code and the bit format for the cards. You will need to know the bit positions for the following parameters: facility code start, facility code end, card id start, card id end, total bits. (Call Quest if these parameters are not known.)

Enter the following command line from a logged-on terminal:

```
DEFINE DOR.1 RLY.1 10 ENERGON <SITE CODE> <FACILITY START> <FACILITY
END> <CARD ID START> <CARD ID END> <TOTAL BITS> <KS>
```

<SITE CODE>:	The facility or customer code programmed for the card
<FACILITY START>:	The position of the first bit is for the facility code.
<FACILITY END>:	The position of the last bit for the facility code.
<CARD ID START>:	the position of the first bit for the unique cards code.
<CARD ID END>:	The position of the last bit for the unique cards code.
<TOTAL BITS>:	The total number of bits to be expected from the card.
<KS>:	Optional this is the facility code for the keypad if it is different from the cards being used. Note keypad codes will always be 26 bit.

Substitute the proper numeric values for the parameters.

Example:

```
DEFINE DOR.1 RLY.1 10 ENERGON 8 1 8 9 24 26 1
```

(This is a standard setup for a 26-bit Wiegand card with a facility code of 8 for the cards and a keypad facility code of 1)

The TELSEC® will reply 'DONE' and display the help message. The previous DOR definition will not be affected. The format will be stored in non-volatile memory when you save your program to flash with the SET PRO DEF command.

4.2.6 Using Keypad Codes:

The TELSEC® can support the use of keypad entries and proximity cards. It does this by mimicking a card being swiped when you enter keys on the keypad. Six numbers are required for each code to be entered in to the keypad. The first two numbers are the facility code for the site followed by four unique digits for the user. The keys entered are in a hexadecimal format and are transmitted to the controller as a hexadecimal number. The controller will automatically convert the hexadecimal number to a decimal number and compare it to the defined cards to see if there is a match. All codes are entered into the system in decimal (DEC) format providing a level of encryption for security.

Create a list of passwords to be assigned. File this list in a secure location.

User Name	Keypad Code (Hex)	Card Code (Decimal)
John Doe	011234	4660

Next use the calculator that comes with windows. Set it up for scientific format and click on HEX numbers. Enter the unique 4 digits for the user and click on the DEC. The calculator will convert the number for you. For example a four digit key code of 1234 will be entered into the system as 4660. The command to enter the code will then look like the following:

```
DEFINE CARD 4660 "JOHN DOE"<enter>
```

When John wants access to the facility, he will enter 01 (value in the <KS> field) followed by 1234.

After completing and changes you will need to save your changes to flash with the `SET PRO DEF` command.

4.3 System Messages

The system will log one of the following statements in the access log when a card is flashed to the system:

`NOTICE, IN, 09/25/07 14:21:00, DOR.1 , CARD 345 (OSCAR GRAHAM), ACCESS GRANTED#:` A valid card has been received and the door relay has been energized.

`NOTICE, IN, 09/25/07 14:20:00, DOR.1 , CARD 8 (DICK BRUTIS), ILLEGAL ATTEMPT:` Access was attempted and denied to card #. The card # was not found in the database. This message will show you if anyone with the correct site code on their card does not have access to the door.

`NOTICE, IN, 09/25/07 14:20:00, DOR.1 , CARD NONE (), ILLEGAL SITE CODE:` A card with a site code differing from the one defined using the `DEFINE DOR` command was found. No access was given.

4.3.1 Reviewing Access Control Info

All access control system transactions are stored in a log in the TELSEC® memory.

To review this information, enter the command:

```
REVIEW LOG ACCESS
```

The log data will be output listed with the most recent transactions first to the oldest entries. There are approximately 800 entries available in the log.

4.4 Listing Access Control Information

When the programming for the access control system is complete, the data can be listed back for storage or reference using the `LIST` command.

To list the door definition use:
`LIST DOR`

To list the valid card data use:

`LIST CARD`	(will show all cards in the database. . If nothing is displayed then there are no cards in the data base.)
`LIST CARD 3100`	(will list the card with access code 3100. If nothing is displayed then the card number is not in the data base.)

4.5 Removing Cards

The `REM CARD <card#>` command will delete a specific card from the data base.

Format:
`REM CARD <card#>`

Where `<card#>` = the access code number of the card.

Example:
```
REM CARD 3050
REM CARD 4095
```

CAUTION: If you send the command `REM CARD 0` to the system, it will delete the entire card database. Use this command carefully.

After completing all changes you will need to save your changes to flash with the `SET PRO DEF` command.