

```

; - - - - -
; RFC-1/B script for ProComm Plus for Windows
; - - - - -

; The constants below should be changed to reflect your RFC-1/B.  The original
; settings will dial the Sine Systems demo transmitter and log its readings.

; TelNum is the telephone number to dial to reach the RFC-1/B.

; MainCode is the main security code of the RFC-1/B.

; SiteID is the site ID that is programmed in the RFC-1/B.  This is not
; usually case sensitive but it must be exact spaces, etc.

; Setup is a modem init string in addition to the one that the com program
; already sends.  This must be changed to one that works with your modem.  Use
; this to disable fax, compression, and automatic error correction.

; ChStart/ChStop are the start and stop channels to log readings.

; LogPath/LogFile are the path and filename of the file to log readings.

; ToDisk/ToPrint will toggle disk and printer logging.  1 is on, 0 is off.

; - - - - -
; Constants
; - - - - -
#define TelNum "1-615-777-7321"
#define MainCode "12345678"
#define SiteID "This is RFC1B"
#define Setup "AT\N"
#define ChStart 0
#define ChStop 13
#define LogPath "C:"
#define LogFile "RFC1LOG.TXT"
#define ToDisk 1
#define ToPrint 0

; - - - - -
; Global Variables
; - - - - -
string ErrMsg=""

; - - - - -
; Main Procedure
; - - - - -
proc main

; Setup file capture properties
set capture autostart off
set capture file LogFile
set capture overwrite off
set capture path LogPath
set capture query off
set capture recordmode screen

; Make three attempts at logon
clear
call DoLogon

if $CARRIER==0
    hangup
    statmsg "Making second attempt in 90 seconds"
    pause 90
    call DoLogon
endif

```

```

if $CARRIER==0
    hangup
    statmsg "Making third attempt in 90 seconds"
    pause 90
    call DoLogon
endif

; Clear before logging
clear

; Capture to disk enable
if ToDisk==1
    capture on
endif

; Capture to printer enable
if ToPrint==1
    printcapture on
endif

; Display header
call DoHeader

if $CARRIER==1
; Take readings
    call TakeReadings
else
; Log message
    termwrits "^M^J"
    termwrits "Could not establish connection"
    termwrits "^M^J"
    termwrits ErrMsg
    termwrits "^M^J"
endif

; Display footer
call DoFooter

; End capture
capture off
printcapture off

; Logoff and hangup
statmsg "Hanging up"
transmit "99"
hangup
pause 2
statmsg ""
endproc

; - - - - -
; Login Procedure
; - - - - -
proc DoLogon

; Send modem setup string
statmsg "Setting up modem"
transmit Setup
transmit "^M"
waitfor "OK" 5
pause 2
if FAILURE
    ErrMsg="Setup string is invalid for this modem"
    return
endif

```

```

; Send modem dial command and wait for connect message
statmsg "Dialing"
transmit "ATDT"
transmit TelNum
transmit "^M"
statmsg "Waiting for connect"
waitfor "CONNECT" 60
if FAILURE
    ErrMsg="No connection"
    return
endif

; Wait for login prompt
statmsg "Waiting for prompt"
waitfor "ENTER" 10
if FAILURE
    ErrMsg="No logon prompt"
    return
endif

; Send security code
statmsg "Logging on"
mspause 500
transmit MainCode
transmit "^M"
waitfor SiteID 5
if FAILURE
    ErrMsg="Incorrect security code"
    return
endif
endproc

; -----
; Readings Procedure
; -----
proc TakeReadings

; Local variables
integer Tens
integer Ones
integer LoopNum
string ChannelNum

; Take readings
statmsg "Taking readings"
for LoopNum = ChStart upto ChStop
    Tens=LoopNum/10
    Ones=LoopNum%10
    itoa Tens ChannelNum
    transmit ChannelNum
    itoa Ones ChannelNum
    transmit ChannelNum
    waitfor "^M" 3
    if FAILURE
        termwrites "Error: "
        termwrites ChannelNum
        termwrites "^M^J"
    else
        mspause 500
    endif
endfor
endproc

; -----
; Header Procedure

```

```
; - - - - -
proc DoHeader
; Local variables
  string DateCap
  string TimeCap

; Header routine
  ltimestrs $LTIME DateCap TimeCap
  termwrites "-----^M^J"
  termwrites "Site:  "
  termwrites SiteID
  termwrites "^M^J"
  termwrites "Date:  "
  termwrites DateCap
  termwrites "^M^J"
  termwrites "Time:  "
  termwrites TimeCap
  termwrites "^M^J"
  termwrites "-----^M^J"
endproc

; - - - - -
; Footer Procedure
; - - - - -
proc DoFooter
  termwrites "End of set"
  termwrites "^M^J^M^J"
endproc

; - - - - -
; End of file
; - - - - -
```

```

; - - - - -
; RFC-1/B script for ProComm Plus for DOS
; - - - - -

; The constants below should be changed to reflect your RFC-1/B.  The original
; settings will dial the Sine Systems demo transmitter and log its readings.

; TelNum is the telephone number to dial to reach the RFC-1/B.

; MainCode is the main security code of the RFC-1/B.

; SiteID is the site ID that is programmed in the RFC-1/B.  This is not
; usually case sensitive but it must be exact spaces, etc.

; Setup is a modem init string in addition to the one that the com program
; already sends.  This must be changed to one that works with your modem.  Use
; this to disable fax, compression, and automatic error correction.

; ChStart/ChStop are the start and stop channels to log readings.

; LogFileName is the path and filename of the file to log readings.

; ToDisk/ToPrint will toggle disk and printer logging.  1 is on, 0 is off.

; - - - - -
; Constants
; - - - - -
define TelNum "1-615-777-7321"
define MainCode "12345678"
define SiteID "This is RFC1B"
define Setup "AT\N"
define ChStart 0
define ChStop 13
define LogFileName "C:RFC1LOG.TXT"
define ToDisk 1
define ToPrint 0

; - - - - -
; Global Variables
; - - - - -
string ErrMsg=""

; - - - - -
; Main Procedure
; - - - - -
proc main

; Turn auto CRLF off
set MSG_CRLF off

; Make three attempts at logon
clear
call DoLogon

if not connected
hangup
statmsg "Making second attempt in 90 seconds"
pause 90
call DoLogon
endif

if not connected
hangup
statmsg "Making third attempt in 90 seconds"
pause 90
call DoLogon

```

```

endif

; Clear before logging
clear

; Capture to disk enable
if ToDisk==1
    log open LogFileName
endif

; Capture to printer enable
if ToPrint==1
    printer on
endif

; Display header
call DoHeader

if connected
; Take readings
    call TakeReadings
else
; Log message
    message "^M^J"
    message "Could not establish connection"
    message "^M^J"
    message ErrMsg
    message "^M^J"
endif

; Display footer
call DoFooter

; Stop capture
printer off
log close

; Logoff and hangup
statmsg "Hanging up"
transmit "99"
hangup
pause 2
statmsg ""
endproc

; - - - - -
; Login Procedure
; - - - - -
proc DoLogon

; Send modem setup string
statmsg "Setting up modem"
transmit Setup
transmit "^M"
waitfor "OK" 5
pause 2
if FAILURE
    ErrMsg="Setup string is invalid for this modem"
    return
endif

; Send modem dial command and wait for connect message
statmsg "Dialing"
transmit "ATDT"
transmit TelNum
transmit "^M"

```

```

statmsg "Waiting for connect"
waitfor "CONNECT" 60
if FAILURE
    ErrMsg="No connection"
    return
endif

; Wait for login prompt
statmsg "Waiting for prompt"
waitfor "ENTER" 10
if FAILURE
    ErrMsg="No logon prompt"
    return
endif

; Send security code
statmsg "Logging on"
mspause 500
transmit MainCode
transmit "^M"
waitfor SiteID 5
if FAILURE
    ErrMsg="Incorrect security code"
    return
endif
endproc

; - - - - -
; Readings Procedure
; - - - - -
proc TakeReadings

; Local variables
integer Tens
integer Ones
integer LoopNum
string ChannelNum

; Take readings
rflush
statmsg "Taking readings"
for LoopNum = ChStart upto ChStop
    Tens=LoopNum/10
    Ones=LoopNum%10
    itoa Tens ChannelNum
    transmit ChannelNum
    itoa Ones ChannelNum
    transmit ChannelNum
    waitfor "^M" 3
    if FAILURE
        message "Error: "
        message ChannelNum
        message "^M^J"
    else
        mspause 500
    endif
endfor
endproc

; - - - - -
; Header Procedure
; - - - - -
proc DoHeader

; Header routine
message "-----^M^J"

```

```
message "Site:  "
message SiteID
message "^M^J"
message "Date:  "
message $DATE
message "^M^J"
message "Time:  "
message $TIME0
message "^M^J"
message "-----^M^J"
endproc
```

```
; -----
; Footer Procedure
; -----
```

```
proc DoFooter
  message "End of set"
  message "^M^J^M^J"
endproc
```

```
; -----
; End of file
; -----
```



```

/* -----
RFC-1/B script for Telix for Windows (SALT)
----- */

/* The constants below should be changed to reflect your RFC-1/B. The original
settings will dial the Sine Systems demo transmitter and log its readings.

TelNum is the telephone number to dial to reach the RFC-1/B.

MainCode is the main security code of the RFC-1/B.

SiteID is the site ID that is programmed in the RFC-1/B. This is not
usually case sensitive but it must be exact spaces, etc.

Setup is a modem init string in addition to the one that the com program
already sends. This must be changed to one that works with your modem.
Use this to disable fax, compression, and automatic error correction.

LogPath/LogFile are the path and filename of the file to log readings.

ChStart/ChStop are the start and stop channels to log readings.

LogPath/LogFile are the path and filename of the file to log readings.

ToDisk/ToPrint will toggle disk and printer logging. 1 is on, 0 is off. */

/* -----
User Global Variables
----- */

str TelNum = "1-615-777-7321" ;
str MainCode = "12345678";
str SiteID = "This is RFC1B";
str Setup = "AT\n";
str LogFile = "RFC1LOG.TXT";
int ChStart = 0;
int ChStop = 13;
int ToDisk = 1;
int ToPrint = 0;

/* -----
Main Procedure
----- */

main()
{

/* Setup */
_add_lf = 0;

/* Local Variables */
str ErrMsg[40];

/* Make three attempts at logon */
clear_scr();
ErrMsg = DoLogon();

if (not(carrier()))
{
hangup();
printS ("Making second attempt in 90 seconds");
delay (900);
ErrMsg = DoLogon();
}
}

```

```

if (not(carrier()))
{
    hangup();
    printS ("Making third attempt in 90 seconds");
    delay (900);
    ErrMsg = DoLogon();
}

/* Clear before logging */
clear_scr();

/* Capture to disk enable */
if (ToDisk == 1)
    capture (LogFile);

/* Capture to printer enable */
if (ToPrint == 1)
    printer (1);

/* Display header */
DoHeader();

if (carrier())
/* Take readings */
    TakeReadings();
else
/* Log message */
{
    printS ("^M^J");
    printS ("No readings available");
    printSC ("Error: ");
    printS (ErrMsg);
}

/* Display footer */
DoFooter();

/* End capture */
capture ("*CLOSE*");
printer (0);

/* Logoff and hangup */
printS ("Hanging up");
cPutS ("99");
hangup();
}

/* -----
Login Procedure
----- */

DoLogon()
{
/* Local Variables */
int Rcvd;

/* Send modem setup str */
printS ("Setting up modem");
cPutS (Setup);
cPutS ("^M");
Rcvd = (waitfor("OK", 5));
if (not Rcvd)
    return ("Setup str is invalid for this modem");
else

```

```

    delay (20);

/* Send modem dial command and wait for connect message */
cPutS ("ATDT");
cPutS (TelNum);
cPutS ("^M");
cPutS ("Waiting for connect");
Rcvd = (waitfor("CONNECT", "NO CONNECT", "NO CARRIER", "BUSY", "NO DIALTONE", 60));
if (Rcvd == 2)
    return "No connection";
else if (Rcvd == 3)
    return ("No carrier detected");
else if (Rcvd == 4)
    return ("Line busy");
else if (Rcvd == 5)
    return ("No dialtone");

/* Wait for login prompt */
Rcvd = (waitfor("ENTER", 10));
if (not Rcvd)
    return ("No logon prompt");
else
    delay (10);

/* Send security code */
cPutS (MainCode);
cPutS ("^M");
Rcvd = (waitfor(SiteID, 5));
if (not Rcvd)
    return ("Incorrect security code");

/* Default Message */
return ("");
}

/* -----
Readings Procedure
----- */

TakeReadings()
{

/* Local variables */
int Tens;
int Ones;
int LoopNum;
int Rcvd;
str ChTemp[2];
str ChannelNum[3];

/* Take readings */
flushbuf();
for (LoopNum=ChStart; LoopNum <= ChStop; LoopNum = LoopNum + 1)
{
    Tens = (LoopNum / 10);
    Ones = (LoopNum % 10);
    ItoS (Tens, ChannelNum);
    cPutC (ChannelNum);
    ItoS (Ones, ChannelNum);
    cPutC (ChannelNum);
    Rcvd = (waitfor("^M", 3));
    if (Rcvd)
        delay (5);
    else
    {
        ItoS (LoopNum, ChannelNum);

```

```

        printSC ("Error on channel: ");
        printS  (ChannelNum);
    }
}
}

/* -----
Header Procedure
----- */

DoHeader()
{

/* Local variables */
str DateCap[9];
str TimeCap[9];

/* Header routine */
date(curtime(), DateCap);
time(curtime(), TimeCap);
printS  ("-----");
printSC ("Site: ");
printS  (SiteID);
printSC ("Date: ");
printS  (DateCap);
printSC ("Time: ");
printS  (TimeCap);
printS  ("-----");
}

/* -----
Footer Procedure
----- */

DoFooter()
{
    printS ("End of set");
    printS ("^M^J");
}

/* -----
End of file
----- */

```

```

// - - - - -
// RFC-1/B script for Telix for DOS (SALT)
// - - - - -

// The constants below should be changed to reflect your RFC-1/B.  The original
// settings will dial the Sine Systems demo transmitter and log its readings.

// TelNum is the telephone number to dial to reach the RFC-1/B.

// MainCode is the main security code of the RFC-1/B.

// SiteID is the site ID that is programmed in the RFC-1/B.  This is not
// usually case sensitive but it must be exact spaces, etc.

// Setup is a modem init string in addition to the one that the com program
// already sends.  This must be changed to one that works with your modem.
// Use this to disable fax, compression, and automatic error correction.

// LogPath/LogFile are the path and filename of the file to log readings.

// ChStart/ChStop are the start and stop channels to log readings.

// LogPath/LogFile are the path and filename of the file to log readings.

// ToDisk/ToPrint will toggle disk and printer logging.  1 is on, 0 is off.

// - - - - -
// User Global Variables
// - - - - -

str TelNum[] = "1-615-777-7321" ;
str MainCode[] = "12345678";
str SiteID[] = "This is RFC1B";
str Setup[] = "AT\N";
str LogFile[] = "d:\temp\RFC1LOG.TXT";
int ChStart = 0;
int ChStop = 13;
int ToDisk = 1;
int ToPrint = 0;

// - - - - -
// Main Procedure
// - - - - -

main()
{
// _add_lf = 0;

// Local Variables
str ErrMsg[40];

// Make three attempts at logon
clear_scr();
ErrMsg = DoLogon();

if (not(carrier()))
{
hangup();
printS ("Making second attempt in 90 seconds");
delay (900);
ErrMsg = DoLogon();
}

if (not(carrier()))

```

```

    {
        hangup();
        printS ("Making third attempt in 90 seconds");
        delay (900);
        ErrMsg = DoLogon();
    }

// Clear before logging
clear_scr();

// Capture to disk enable
if (ToDisk == 1)
    capture (LogFile);

// Capture to printer enable
if (ToPrint == 1)
    printer (1);

// Display header
DoHeader();

if (carrier())
// Take readings
    TakeReadings();
else
// Log message
{
    printS ("^M^J");
    printS ("No readings available");
    printSC ("Error: ");
    printS (ErrMsg);
}

// Display footer
DoFooter();

// End capture
capture ("*CLOSE*");
printer (0);

// Logoff and hangup
printS ("Hanging up");
cPutS ("99");
hangup();
}

// - - - - -
// Login Procedure
// - - - - -

DoLogon()
{

// Local Variables
int Rcvd;

// Send modem setup str
printS ("Setting up modem");
cPutS (Setup);
cPutS ("^M");
Rcvd = (waitfor("OK", 5));
if (not Rcvd)
    return ("Setup str is invalid for this modem");
else
    delay (20);
}

```

```

// Send modem dial command and wait for connect message
cPutS ("ATDT");
cPutS (TelNum);
cPutS ("^M");
cPutS ("Waiting for connect");
Rcvd = (waitfor("CONNECT", "NO CONNECT", "NO CARRIER", "BUSY", "NO DIALTONE", 60));
if (Rcvd == 2)
    return "No connection";
else if (Rcvd == 3)
    return ("No carrier detected");
else if (Rcvd == 4)
    return ("Line busy");
else if (Rcvd == 5)
    return ("No dialtone");

// Wait for login prompt
Rcvd = (waitfor("ENTER", 10));
if (not Rcvd)
    return ("No logon prompt");
else
    delay (10);

// Send security code
cPutS (MainCode);
cPutS ("^M");
Rcvd = (waitfor(SiteID, 5));
if (not Rcvd)
    return ("Incorrect security code");

// Default Message
return "";
}

// -----
// Readings Procedure
// -----

TakeReadings()
{
// Local variables
int Tens;
int Ones;
int LoopNum;
int Rcvd;
str ChannelNum[3];

// Take readings
flushbuf();
delay(5);
for (LoopNum=ChStart; LoopNum <= ChStop; LoopNum = LoopNum + 1)
{
    Tens = (LoopNum / 10);
    Tens = Tens + 48;
    Ones = (LoopNum % 10);
    Ones = Ones + 48;
    cPutC (Tens);
    cPutC (Ones);
    Rcvd = (waitfor("^M", 3));
    if (Rcvd)
        delay (5);
    else
    {
        ItoS (LoopNum, ChannelNum);
        printSC ("Error on channel: ");
    }
}
}

```

```

        printS (ChannelNum);
    }
}

// -----
// Header Procedure
// -----

DoHeader()
{
// Local variables
str DateCap[9];
str TimeCap[9];

// Header routine
date(curtime(), DateCap);
time(curtime(), TimeCap);
printS ("-----");
printSC ("Site: ");
printS (SiteID);
printSC ("Date: ");
printS (DateCap);
printSC ("Time: ");
printS (TimeCap);
printS ("-----");
}

// -----
// Footer Procedure
// -----

DoFooter()
{
    printS ("End of set");
    printS ("^M^J");
}

// -----
// End of file
// -----

```