Sending Text Messages from One DXM to Another

Overview

Communicating from one DXM Controller to another DXM Controller using SMS text messaging is possible with a ScriptBasic program. The data is sent using cellular text messaging at a cyclical rate but that can be changed to send data when something changes. This example uses one DXM Controller as the data originator (pitcher) and the other DXM Controller as the receiver (catcher), although this could be bi-directional.

However, because the communications between DXM Controllers uses SMS text messaging, verify your cellular wireless plan includes SMS messages to allow the text messaging rate programmed in the DXM system.

Some basic information must be set in the ScriptBasic program and XML configuration file for both devices to make texting communications work. That information is:

- Enter the cellular phone number to send text messages to into the ScriptBasic program.
- For this example, define the cyclical interval to send text messages.
- Open the software firewall on the receiver (catcher) DXM for the phone number of the transmitter DXM (pitcher).

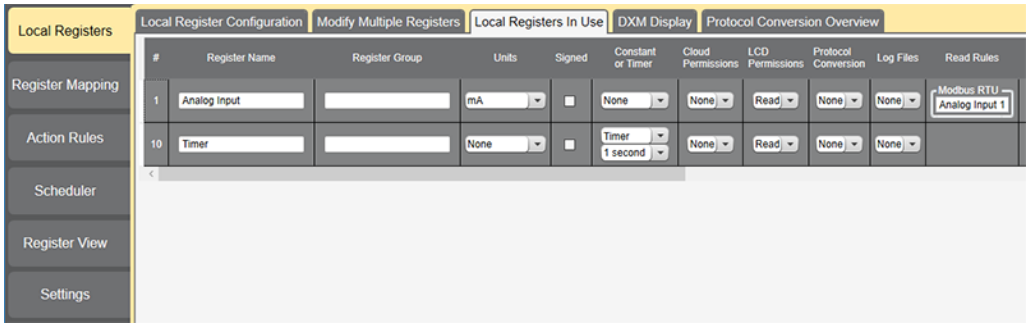
The two XML files, **Cellular_SMS_MappingPitcher.xml** and **Cellular_SMS_MappingCatcher.xml** can be loaded directly into two cellular DXM Controllers and should be functional as is. The ScriptBasic program must be changed to provide the phone number of the receiver (catcher) DXM Controller cell modem. Manually load the ScriptBasic program onto the transmitter (pitcher) DXM Controller using the DXM Configuration Tool, from the **Settings** > **Scripting** screen.

Configure the Transmitter DXM

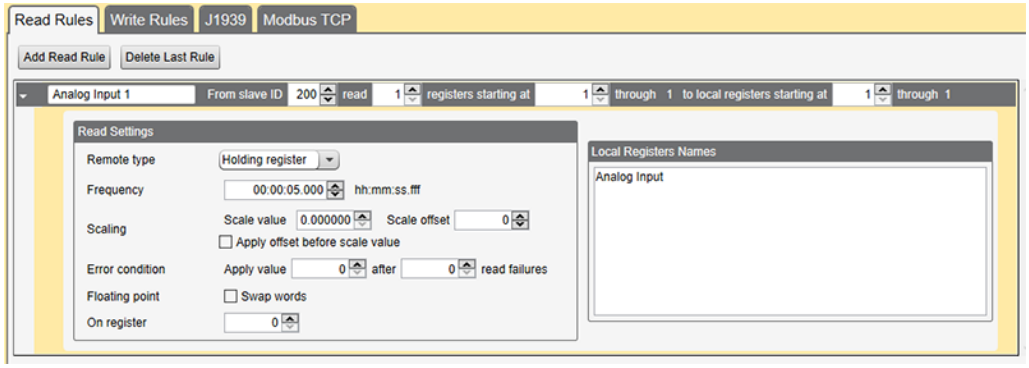
The configure the transmitter (pitcher) DXM Controller, follow these steps.

Upload the pre-configured configuration files to your DXM, or follow these instructions to update an existing xml file.

1. Launch the DXM Configuration Tool and go to the **Local Registers** > **Local Register Configuration** screen.
2. Define register 1 is the Analog Input and define register 10 is a timer used to schedule the SMS messages.

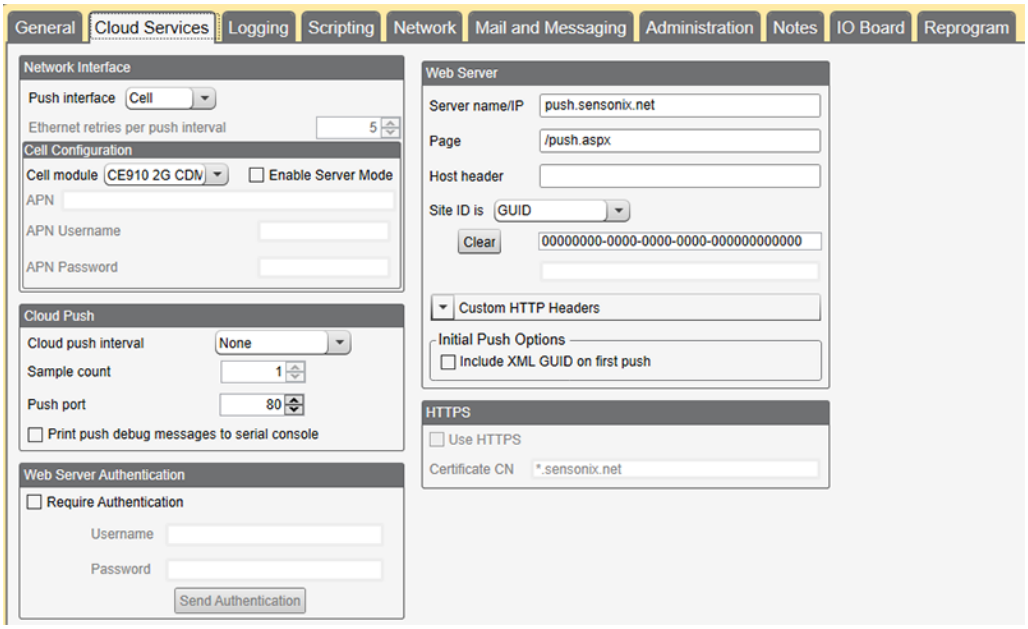


3. Define a Read Rule to read Universal input 1 and write the value into Local Register 1.



4. Set the Network's **Push interface** to Cell.

5. Set the **Cloud push interval** to None.



6. From the **Settings** > **Scripting** menu, manually load the ScriptBasic program (Cellular_SMS_Mapping.sb) to the transmitter (pitcher) DXM Controller by clicking **Upload File**.

   The ScriptBasic program needs to be loaded manually, separate from the XML configuration file. The setting of the ScriptBasic program to run at start up time is already configured in the XML file.

ScriptBasic Program Contents

The ScriptBasic program running on the transmitter (pitcher) DXM Controller creates a SMS text message to the receiver (catcher) DXM Controller.

The first section of the program defines constants and variables for the follow-on program. The constant that should be adjusted is the phone number, **PhoneNumberA**. The reset of the constants and variables can be left alone for basic operation.

SMS_Interval

Defines how often the DXM Controller creates the test message.

Currently set to 30 seconds.

PhoneNumberA

Constant that represents the cellular phone number of the receiver (catcher) DXM Controller.

Defined within the ScriptBasic program.

The beginning of the program starts with a SETREG command that writes a configuration register in the I/O board to make Universal input 1 a 4–20mA input. Not all applications will require this step.

The main program loop starts with the WHILE command. The first GETREG command reads the analog input value from Local Register 1, the next GETREG command reads the TIMER register.

The IF statement checks to see if the timer has reached the maximum value. If so, the program sends the SMS text message.

The **Message** variable is loaded with "sr1,xxxx". The value xxxx is the analog value read from Local Register 1.

The TIMER register is written to zero, then the text message is sent using the FILEOUT command.

The PRINT commands throughout the program will show up on the console output in the DXM Configuration Tool.

The last command, SLEEP(1) causes the program to wait 1 second and then repeat the process. The WEND command defines the end of the WHILE loop, so that all programming statements between the WHILE and the WEND are executed forever.

```
 ' ScriptBasic program to send a SMS text message every minute to send and analog input
' to an analog output on another DXM using a cellular connection.

CONST In1_reg = 1
CONST In1Type_reg = 3306
CONST AnalogType4_20 = 2
CONST Timer_reg = 10
CONST LED_regs = 1102
CONST SMS_Interval = 30
CONST PhoneNumberA = "4403180566"
CONST SMS_Out = 3
'
```

```
CONST LocalReg = 199
CONST IOBoardSID = 200
CONST DisplaySID = 201
CONST HoldingReg = 0
'
AnalogInputData = 0
TimerValue = 0
Message = 0
Result = 0
WrErr = 0
'
' Set the universal input 1 type to 0-20ma (type 2)
' Not all applications will need to setup a universal input type, this example does…
WrErr = SETREG(In1Type_reg, AnalogType4_20, IOBoardSID, HoldingReg)
'
' MAIN LOOP
WHILE(1)
        'Read I/O to detect change.
        AnalogInputData = GETREG(In1_reg, LocalReg, HoldingReg)
        'Timer to indicate when to send a message.
        TimerValue = GETREG(Timer_reg, LocalReg, HoldingReg)
        '
        'Send SMS message if timer has reached SMS_Interval
        IF TimerValue >= SMS_Interval THEN
                PRINT"Input data read: ",AnalogInputData, "\n"
                PRINT"Send SMS message...\n"
                Message = "sr1," &AnalogInputData
                ' Reset the timer register to zero
                WrErr = SETREG(Timer_reg, 0, LocalReg, HoldingReg)
                ' Send SMS message to PhoneNumberA
                Result = FILEOUT(SMS_Out, 0, 0, PhoneNumberA, Message )
                IF Result = 0 THEN
                        PRINT"SMS message: ", Message, " Phone#: ",PhoneNumberA, "\n"
                ELSE
                        PRINT" *** ERROR - Cellular \n"
                END IF
        END IF
        SLEEP(1)
WEND
END
```

Configure the Receiver DXM

The receiver (catcher) DXM requires little configuration. The SMS text message it receives from the transmitter DXM is a standard format that it understands and knows what to do with it. The receive string, sr1,xxxx indicates to Set Register 1 with value xxxx.

Upload the pre-configured configuration files to your DXM, or follow these instructions to update an existing xml file.

1. On the DXM Configuration Tool and go to the **Local Registers** > **Local Register Configuration** screen.
2. Set the Network's **Push interface** to Cell.
3. Set the **Cloud push interval** to None.
4. On the **Settings** > **Network** screen, go the **Cell Software Firewall** section and select **Open software firewall**.
5. Define Local Register 1 as the analog output. The SMS message updates this register when the text message is received.
6. Define a **Write Rule** to read Local Register 1 and write it to the I/O base board analog output 1.