5 Gould Road, PO Box 2155
New London, NH 03257 USA
Voice: (603) 526-9800
info@canarysystems.com
www.canarysystems.com

Integrating the RXTX Telependulum with the CR800/CR1000

Sensor Application Note #17

Overview

The Application Note will provide all of the necessary hardware and software integration information so that RXTX Telependulum may be connected to a Campbell Scientific CR800 or CR1000 Control Module. This facilitates easier datalogging and management of the Telependulum network compared to using a terminal program and interrogating the pendulums manually.

The Telependulum is an electronic detection instrument designed to measure and record the movements of auscultation pendulums used in various Civil Engineering works: dams, hydro and nuclear power stations, bridges, buildings and rock foundations. The Telependulum can measure horizontal movements in the two axes (X and Y-axis) and in the case of pendulums equipped with an additional Invar levelling wire it can also measure vertical movements, or Z-axis.

Note: Measurement and storage of the vertical movement, or Z-axis, output is not supported at this time. Contact Canary Systems if you require storage of this output.

Refer to the Telependulum Installation and Configuration Manual for more information on the RXTX Telependulum.

Refer to the MultiLogger User's Guide for more information on the MultiLogger software.

Refer to the Campbell Scientific CR800 or CR1000 Operators Manual for more information on the Control Module.

Note: The gage types shown in the MultiLogger Configuration section are included in MultiLogger v5.1.6 and higher. The current version may be downloaded from the Support area at our website.

Hardware Interface

This Sensor Application Note will outline several connection devices and methods. The simplest is a direct RS-232 connection between the CR800 or CR1000 MCU and the TelePendulum. Wiring is below.

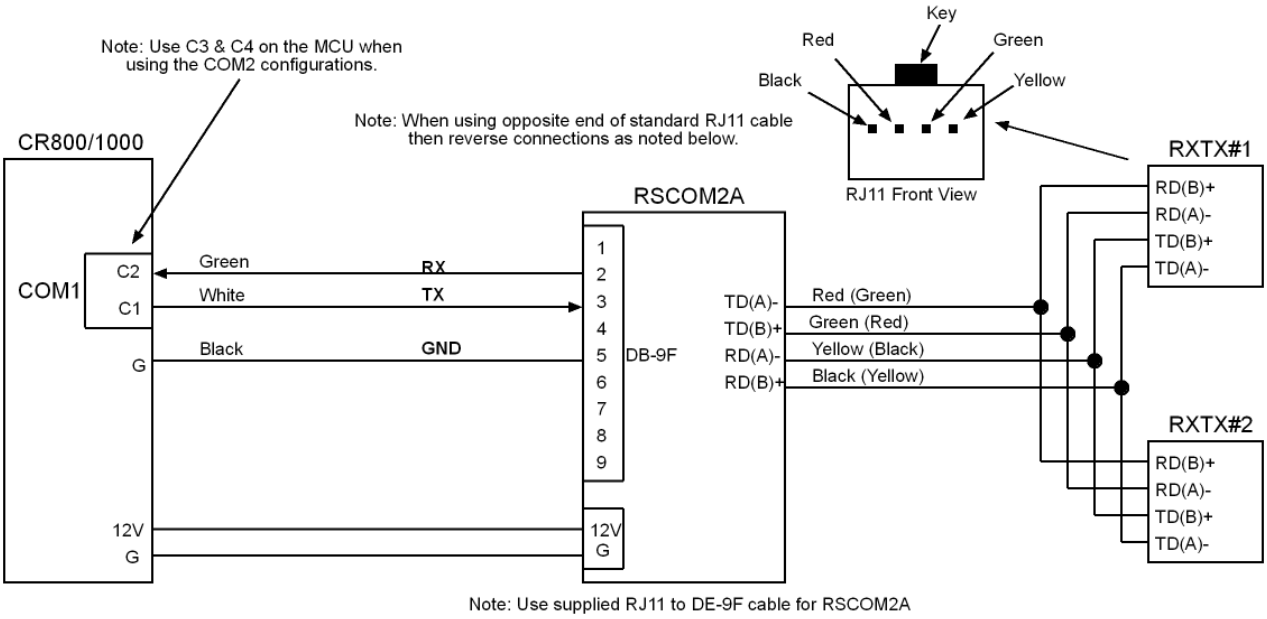
CR800/1000 COM1 Direct RS-232 Connection

CR800/1000	Description	RS-232 DB9	RS-232 DB25
C1	Transmit Data	3	2
C2	Receive Data	2	3
SW12V	Request to Send (RTS)	7	4
G	Ground	5	7

CR800/1000 COM2 Direct RS-232 Connection

CR800/1000	Description	RS-232 DB9	RS-232 DB25
C3	Transmit Data	3	2
C4	Receive Data	2	3
SW12V	Request to Send (RTS)	7	4
G	Ground	5	7

CR800/CR1000 COM1 RS-485 Connection using RESmith RSCOM2A Adaptor



RSCOM2A Configuration: Configure jumpers J7-J16 as needed for ground isolation/line termination. In most cases jumpers J8-J15 can be removed. All other jumpers should be in factory default configuration. See the RSCOM2A schematic for detail on the jumper locations and settings.

Telependulum Programming

When using the RXTX TelePendulum with RS-485 networking the Logical RS-485 ID's must be configured as follows:

First Unit: TP_01
Second Unit: TP_02
.
.
.
Last Unit: TP_16

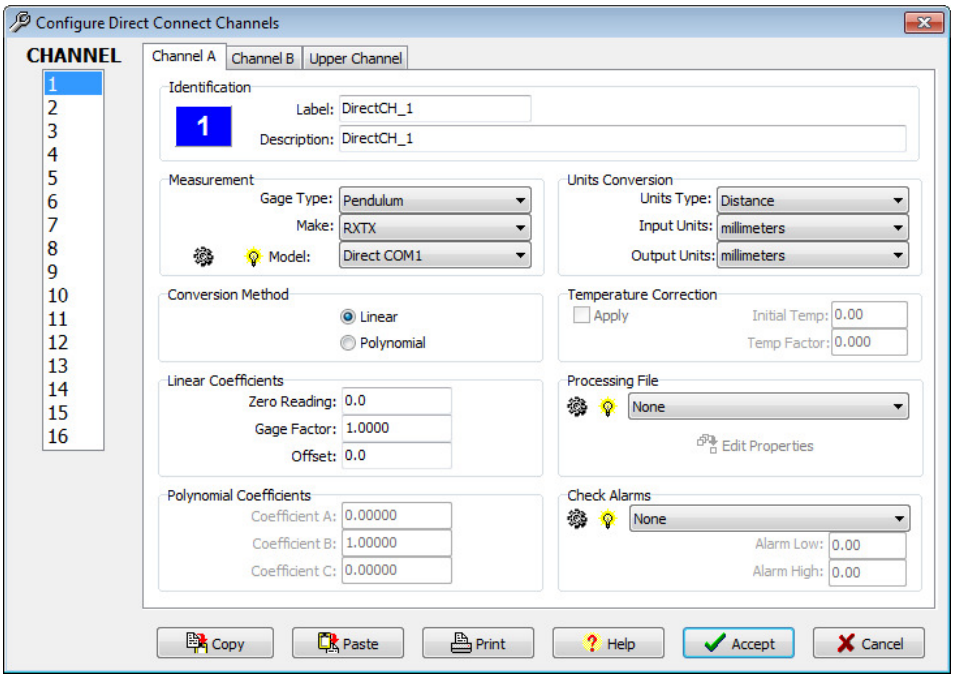
The Telependulum's are programmed using the DB-25 Console Port, a standard serial cable connection to a PC (with 25 to 9-pin changer) and the software Hyperterminal. Hyperterminal should be configured for 9600 baud, 8 data bits, 1 stop bit and no parity. Press **<ENTER>** on the PC after powering up the Telependulum and making the cable connections. You should see the console menu returned by the Telependulum. Use the **PW** command to enter the password, the default is **RXTX**. Use the **CF** command to enter the configuration menu, then select **CH** to change the Logical RS-485 ID. After changing the RS-485 ID press <ESC> to return to the menu, then <ESC> again to return to the main menu. Settings are saved automatically. See the Telependulum Installation and Configuration Manual for more information.

Note: There are also 2 sets of DIP switches in the Telependulum which configure various hardware and software options. This Sensor Application Note assumes the default switch settings.

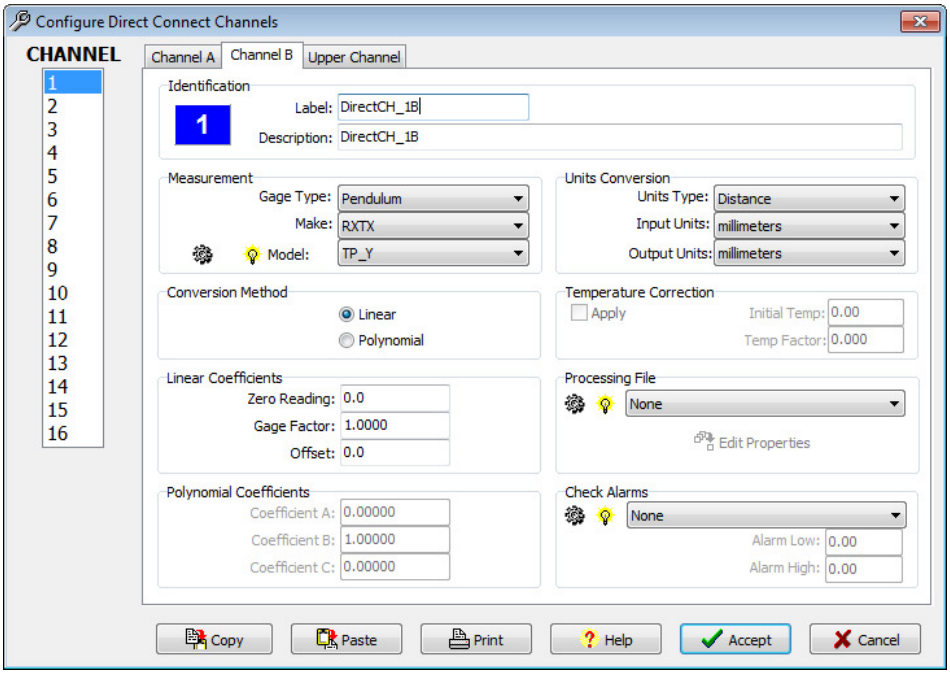
MultiLogger Configuration

Select the menu item **Configure** | **Direct Connect Channels** to display the channel configuration form (or use the Direct Connect button on the Program tab). Up to 16 Pendulums may be configured with X & Y. The following screenshots provide example configurations.

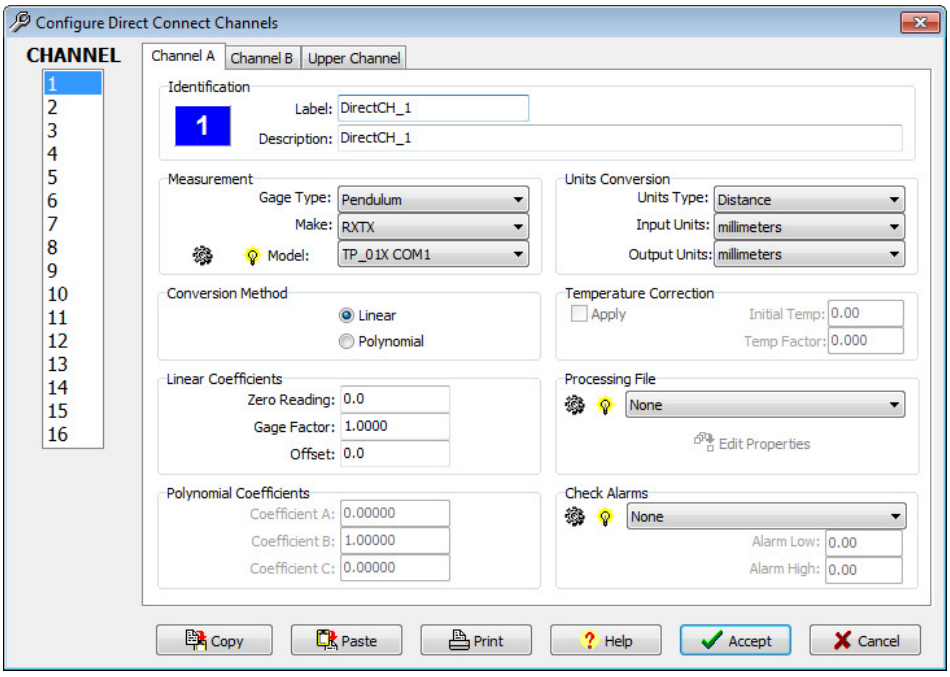
Read X with Direct Connection via COM1 (COM2 also supported)



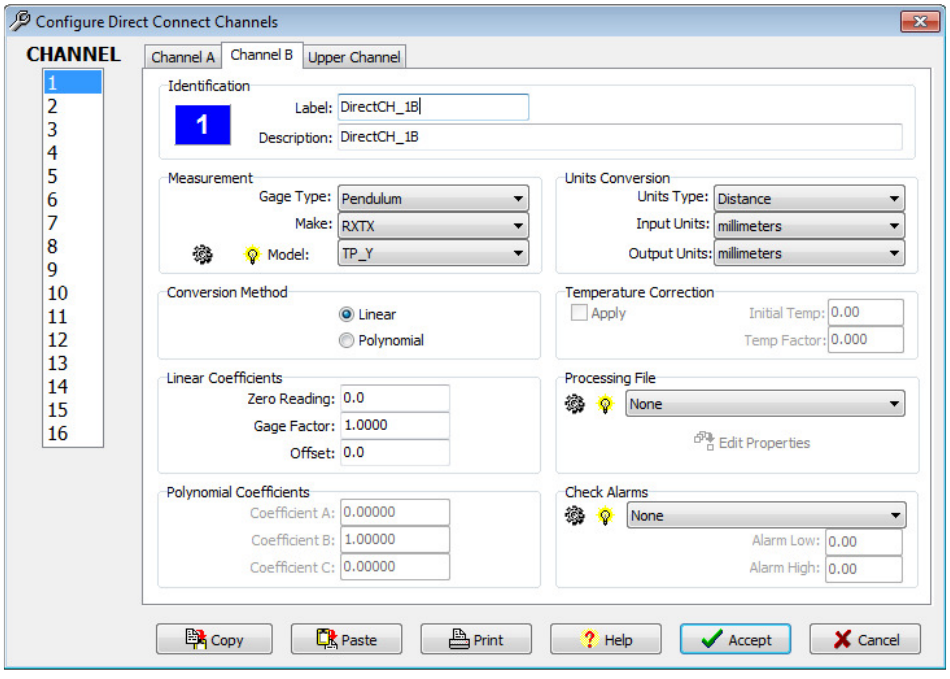
Read X with Direct Connection via COM1 (COM2 also supported)



Read X with RS-485 Connection via COM1 (COM2 also supported)



Read Y with RS-485 Connection via COM1 (COM2 also supported)



Note: Use the above Gage Type | Make | Model for all Y outputs.

RXTX Instruction Files

The instruction files can be loaded into the editor by clicking the gear button located to the left of the selected **Model** option. Following are 2 example files.

Read X and Y Output of RXTX TelePendulum with Direct COM1 Connection

```
'Read the X and Y output of RXTX TelePendulum connected via RS-232 to COM1 (C1
& C2)

'Turn on RTS
SW12 (1)

'Com up delay
Delay (0,1,Sec)

'Open our serial port
SerialOpen (Com1,9600,0,1000,255)

'Send <CR> to get menu
SerialOut (Com1,CHR(13),"",0,0)
Delay (0,2,Sec)

'Make sure we clear buffer
SerialFlush(Com1)

'Wait after clearing buffer
Delay (0,1,Sec)

'Send Read command
SerialOut (Com1,CHR(13),"",0,0)

'Get up to 350 characters
SerialIn (sInBuf,Com1,500,"",350)

'Check if we have enough characters
if Len(sInBuf) > 250 then
      'Remove header before 'DZ'
      Splitstr(sOutBuf,sInBuf,"DZ",1,4)
      'Split out 10 values (includes Z)
      Splitstr(ScratchLoc(2),sOutBuf,"",10,0)
      'get our X value
      mlReading = ScratchLoc(8)
Else
      mlReading = -99999
EndIf

'Close our serial port
SerialClose (Com1)

'Reset TelePendulum
SW12 (0)
```

Read X and Y Output of RXTX TelePendulum with RS-485 COM1 Connection

```
'Read the X and Y output of RXTX TelePendulum connected to COM1 (C1&C2) at Address 01

'Clear our retries counter
ScratchLoc(1) = 0

'Repeat this sequence 5 times to improve reliability
Do
	'Send our BRK signal
	SerialBrk (Com1,200)

	'Open our port
	SerialOpen (Com1,9600,0,1000,255)

	'Send speed command to RXTX, where: 5=1200,10=2400,12=9600,16=19200
	SerialOut (Com1,"12"+CHR(13)+CHR(13),"",0,0)
	'Wait 3 seconds
	Delay (0,3,Sec)

	'Send Address
	SerialOut (Com1,"TP_01"+CHR(13),"",0,0)
	'Wait 3 seconds
	Delay (0,3,Sec)

	'Send Attention command
	SerialOut (Com1,"&&&"+CHR(13),"",0,0)
	'Wait 3 seconds
	Delay (0,3,Sec)
	'Flush our receive buffer
	SerialFlush(Com1)

	'Send our Read command
	SerialOut (Com1,"P"+CHR(13),"",0,0)
	'Get our response
	SerialIn(sInBuf,Com1,500,"",350)
	'Check for enough characters
	if Len(sInBuf) > 250 then
		'Remove header before 'DZ'
		Splitstr(sOutBuf,sInBuf,"DZ",1,4)
		'Split out 10 values (includes Z)
		Splitstr(ScratchLoc(2),sOutBuf,"",10,0)
	'No valid response
	Else
		ScratchLoc(8) = -99999
	EndIf
	'Increment our counter
	ScratchLoc(1) = ScratchLoc(1) + 1

'End measurement loop
Loop Until (ScratchLoc(1) >= 5) OR (ScratchLoc(8) > -99999)

'Copy our reading whatever it is
mlReading = ScratchLoc(8)

'Close our serial port
SerialClose (Com1)

'Send our BRK signal
SerialBrk (Com1,200)
```