Crystal
pressure

30 Series LabVIEW™ Driver Library

Instruction Manual

AMETEK®
TEST & CALIBRATION INSTRUMENTS

Contents

Overview

INTRODUCTION

The Model 30 LabVIEW Driver Library is a set of VIs that implements the serial commands of the Model 30. The library also provides parsing of the streaming data output, placing it into a simple cluster for easy access. A sample test panel is also provided to demonstrate the use of each of the VIs, as well as a simple data plotting example.

This documentation and the associated driver library assume a reasonable working understanding of LabVIEW, and are not meant to be a tutorial or trainer on developing applications within the LabVIEW development environment.

DATA COLLECTION AND THE MODEL 30

An important aspect of the Model 30 is to understand how it outputs data. The Model 30 is a streaming instrument, which means data is sent continuously across the serial port once streaming is turned on. Data is sent in packets, with each packet containing different information about the instrument, such as the pressure reading, milliamp reading, and analog-to-digital conversion values. This packetization of data means that a specific measurement (the pressure on the low pressure port, for example) could be the next packet, or it could be 3 packets away. Because of this, the simplest approach is to create a cluster of values, and fill that cluster in as the data is received from the unit. It is this approach that is used in the library, with values being updated as new packets arrive.

The Model 30 uses a round-robin approach to sending packets, although it cannot be assumed that the mA reading packet is every 5th packet, for example, because the unit asynchronously outputs A/D calibration packets that would skew any fixed packet count assumptions.

It is important to understand that because of this packetization, it is not necessarily predictable when a specific value is going to be received.

WAITING FOR A SPECIFIC MEASUREMENT

In order to store data for a specific measurement (for example, the pressure measurement of the low pressure port), there are two approaches that can be taken: one, simply wait long enough to allow the unit to update all of the measurements within the cluster, which is typically less than 400 milliseconds; or two, look for a value change of the cluster variable that contains the measurement.

In the first approach, simply maintain a value that specifies when the next sample is to be taken, and retrieve the data element from the cluster on those intervals. This is the easiest approach and is typically sufficient for most applications. This is also the approach that is taken in the sample test panel provided with the library.

With the second approach, initialize the data value within the cluster to something the unit cannot possibly report, such as -100 for the pressure reading. When that reading changes to something other than -100, you can immediately take action on the received value. This works well when attempting to provide the quickest response time to measurement changes, or to maximize the measurement rate of the unit.

USING THE LIBRARY

The library is made up of two basic parts: the command VIs that implement the Model 30 commands such as Zero, Units, and Reset; and the data collecting and parsing VIs that handle the reception and parsing of data packets. In addition, there is a sample *M30 Test Panel.vi* that demonstrates the use of all commands and the data collection components.

Note that the VISA handle must be opened by using *M30 Initialize.vi*, as the serial port requires a special, non-standard setup to allow communication with the Model 30. This handle should then be passed to all VIs. While each of the VIs also provides a VISA resource name (dup) output, it's not required to use this output, as the handle properties are not modified within any of the VIs. That standard LabVIEW VISA Close can be used to close the serial port.

In order to receive and parse the data, *M30 Collect Data.vi* must be placed inside a loop structure, as the data is continuously streamed and therefore must be read and parsed to populate the M30 cluster. A cluster with elements named as in the test panel (or copy of the cluster from the test panel) should be wired into the inputs and outputs of the VI. The example test panel demonstrates the data collection process in the first three frames of the stacked sequence structure inside the main while loop.

Note that when collecting data, the Model 30 only outputs the values displayed on the LCD display. In order to receive a specific sensor's information, either manually select the units from the button on the front panel, or use *M30 Change Units Command.vi* or *M30 mA Command.vi* to programmatically change the units.

All commands can be executed asynchronously from the data collection loop.

Virtual Instruments

LIBRARY VIS

The Model 30 LabVIEW library contains all the VIs that control and collect data from the instrument. Each of the library's VIs are documented below. Each of the VIs has VISA handles or error clusters as inputs and outputs, and more information on these elements can be found in the LabVIEW documentation. Inputs or outputs specific to this library are documented below.

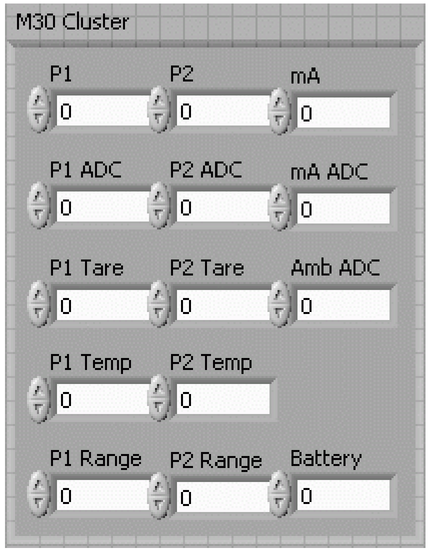
In addition, there are two VIs that are not considered part of the library, but are provided as example VIs that uses the library. These VIs are documented in the Sample VIs section below.

The VIs *M30 Test & Parse Data.vi*, *M30 Parse & Assign Data.vi*, and *M30 Parse 31-Byte Block.vi* are internal functions used by *M30 Collect Data.vi*, and are not documented here. However, their block diagrams are included if you wish to browse through them.

M30 CLUSTER

The M30 Cluster has the following elements:

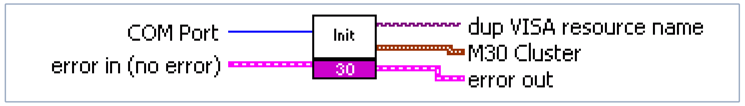
Value	Format	Description
P1	Double	P1 pressure value as displayed on the front panel
P1 ADC	32-bit signed integer	The raw P1 pressure value from the analog-to-digital converter
P1 Tare	32-bit signed integer	The value subtracted (tared) from the P1 ADC value prior to a conversion to pressure.
P1 Temp	32-bit signed integer	The raw P1 temperature value from the analog-to-digital converter
P1 Range	16-bit signed integer	The selected range (units) of the P1 sensor
P2	Double	P2 pressure value as displayed on the front panel
P2 ADC	32-bit signed integer	The raw P2 pressure value from the analog-to-digital converter
P2 Tare	32-bit signed integer	The value subtracted (tared) from the P2 ADC value prior to a conversion to pressure.
P2 Temp	32-bit signed integer	The raw P2 temperature value from the analog-to-digital converter
P2 Range	16-bit signed integer	The selected range (units) of the P2 sensor
mA	Double	mA value as displayed on the front panel
mA ADC	32-bit signed integer	The raw mA value from the analog-to-digital converter
Battery	16-bit signed integer	Three values: -1 = dead, 0 = low, 1 = good



VI DESCRIPTIONS

M30 Initialize.vi

The M30 Initialize VI opens the serial open and correctly initializes it for operation with the Model 30. It also passes an initialized M30 Cluster, which can then be used as inputs to other M30 VIs.



▶ Inputs and Outputs

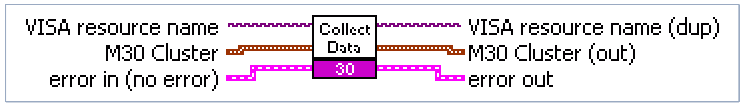
M30 Cluster	Cluster of data values that are updated by M30 Collect Data.vi.

M30 Collect Data.vi

M30 Collect Data handles the reception and parsing of the serial data being streamed from the Model 30. The serial port must be opened using the *M30 Initialize.vi*, and streaming must be enabled by calling *M30 Start Streaming.vi* before this VI is used. It only processes data already sent, and will not wait for the next data packet, so it is possible that none of the elements in the M30 Cluster will be updated within the execution of a single call.

It is expected that this VI is called within a While Loop, as it must be continuously called to parse the streaming data from the Model 30. For performance improvement reasons, it is recommended that a 25 to 50 millisecond delay be added to the While Loop, but is not required. Adding this delay will lower the CPU load the program places on the computer but will not adversely affect throughput.

To detect a new reading, simply look for changes between the input cluster and the output cluster. As stated above, it is possible no values will change, depending on where the Model 30 is in its streaming output.

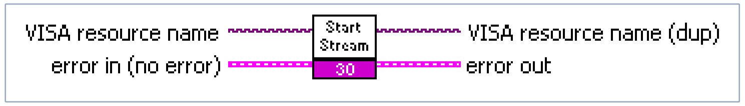


▶ Inputs and Outputs

M30 Cluster	Copied to M30 Cluster (out). Any new readings will supercede the input values.
M30 Cluster (out)	Output cluster, contains the values from the M30 Cluster plus any new readings parsed during the call.

M30 Start Streaming Command.vi

Starts streaming data over the serial interface. Streaming must be enabled before any data collection can occur.

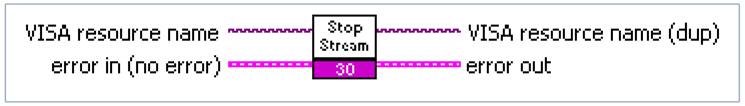


▶ Inputs and Outputs

Only standard VISA handles and error clusters.

M30 Stop Streaming Command.vi

Stops streaming data over the serial interface.

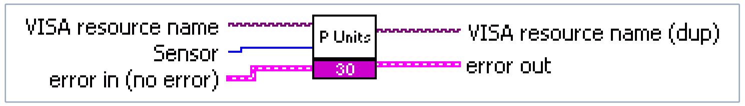


▶ Inputs and Outputs

Only standard VISA handles and error clusters.

M30 Change Units Command.vi

This will cause the Model 30 to change the displayed and reported pressure unit for the given sensor. Effect is identical to pressing the corresponding **(units)** button on the Model 30.

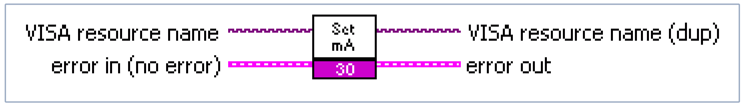


▶ Inputs and Outputs

Sensor.	Use 1 for P1 or 2 for P2.

M30 mA Command.vi

The mA Command VI will cause the mA readings to be displayed on the front panel or change the units of the mA reading, if it's already displayed. Has the same effect as pressing the mA button on the Model 30.

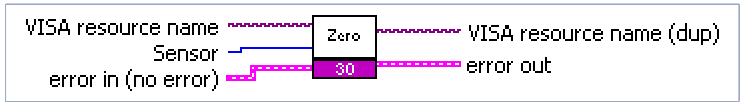


▶ Inputs and Outputs

Only standard VISA handles and error clusters.

M30 Zero Command.vi

This VI will zero (or tare) the pressure reading by subtracting the current value from future readings. Using this function has the same effect as pressing the corresponding ZERO button on the Model 30.

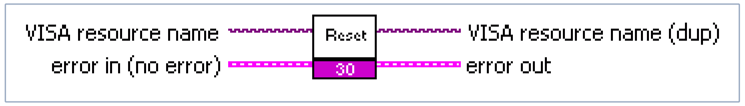


▶ Inputs and Outputs

Sensor.	Use 1 for P1 or 2 for P2.

M30 Reset Command.vi

This VI will cause the unit to reset, as if the batteries were removed and re-inserted. A reset will cause the unit to clear the zero (tare) value for all pressure sensors, and the display will should P1 and mA readings. This function can be useful to return the unit to a known state prior to sending additional commands.



▶ Inputs and Outputs

Only standard VISA handles and error clusters.

SAMPLE VIS

Two VIs are provided as examples of how to use the Model 30 library. They are *M30 and Test Panel.vi* and *M30 Collect Pressure Point.vi.*

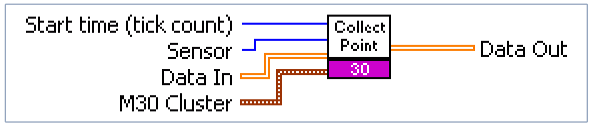
M30 Test Panel.vi

This top level VI demonstrates the use of each of the VIs within the library. It also demonstrates a sample data collecting loop, including the graphing of the pressure or mA readings from the unit. As it is a top level VI, it has no inputs or outputs.

The COM Port numeric control should be set to the proper COM port prior to executing the VI. To stop executing, press the Stop button on the VI. Do not use the LabVIEW stop command, as it will leave the serial port open until LabVIEW is exited, or the VI is rerun and its stop button is pressed.

M30 Collect Pressure Point.vi

This VI is a simple data collection VI that takes the M30 Cluster as an input and adds the appropriate sensor reading to the Data array. Time stamps are based off of the Start time (tick count) input.



▶ Inputs and Outputs

Sensor	Use 1 for P1 or 2 for P2.
Data In	Input 2xN array of data elements.
Data Out	Output 2xN array of data elements. The appropriate sensor data is added to the end, with column one equal to the current tick count minus the Start time (tick count) input, and column two equal to the sensor reading.
M30 Cluster	The data source for the array addition above.

Support

SERVICE AND SUPPORT

How to contact us:

Phone	(805) 595-5477
Toll-Free	(800) 444-1850
Fax	(805) 595-5466
Email	service@crystalengineering.net
Web	www.crystalengineering.net

Send your comments to: feedback@crystalengineering.net

DISCLAIMER OF WARRANTY

LabVIEW AND THE CORRESPONDING PRESSURE GAUGE (COLLECTIVELY "PRODUCT") IS PROVIDED "AS IS." CRYSTAL ENGINEERING CORPORATION DOES NOT AND CANNOT WARRANT THE PERFORMANCE OR RESULTS PURCHASER MAY OBTAIN BY USING THE PRODUCT. CRYSTAL ENGINEERING CORPORATION MAKES NO PROMISES, REPRESENTATIONS, OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE PRODUCT'S CONDITION, ITS CONFORMITY TO ANY REPRESENTATION OR DESCRIPTION, THE EXISTENCE OF ANY LATENT OR PATENT DEFECTS, ANY NEGLIGENCE, AND ITS MERCHANTABILITY OR FITNESS FOR A PARTICULAR USE.

Good data processing procedure dictates that any Product be thoroughly tested with non-critical data before relying on it. The purchaser must assume the entire risk of using the Product. ANY LIABILITY OF CRYSTAL ENGINEERING CORPORATION FOR A DEFECTIVE PRODUCT WILL BE LIMITED EXCLUSIVELY TO REPLACEMENT OF PURCHASER'S COPY OF THE PRODUCT WITH ANOTHER COPY OR, IF SO ELECTED BY CRYSTAL ENGINEERING CORPORATION IN ITS SOLE DISCRETION, REFUND OF THE INITIAL AMOUNT PAID BY PURCHASER TO CRYSTAL ENGINEERING CORPORATION FOR THE PRODUCT. IN NO EVENT WILL CRYSTAL ENGINEERING CORPORATION OR ITS SUPPLIERS BE LIABLE TO PURCHASER FOR ANY CONSEQUENTIAL, INCIDENTAL OR SPECIAL DAMAGES, INCLUDING ANY LOST PROFITS OR LOST SAVINGS, EVEN IF CRYSTAL ENGINEERING CORPORATION WAS ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY THIRD PARTY.

Crystal pressure

AMETEK®
TEST & CALIBRATION
INSTRUMENTS

3349.A

ISO 9001:2008
17025:2005