Software manual Active X Controls Signalog 6000 & PICAS

PEEKEL INSTRUMENTS

PEEKEL INSTRUMENTS B.V
INDUSTRIEWEG 161
3044 AS ROTTERDAM
TEL: (010)-415 27 22
FAX: (010)-437 68 26
EMAIL: sales@peekel.nl

PEEKEL INSTRUMENTS GMBH
BERGMANNSTRASSE 43
44809 BOCHUM
TEL: 0234/904 1603
FAX: 0234/904 1605
EMAIL: Peekel@t-online.de

Contents:

Version number	Version 1.1
Release date	October 2001
Author	J.H. Steeneveld

1 Introduction

The PB6000 processor card, which is used in PICAS and the Signalog 6000 system, is equipped with an RS232C communication port. Through this port, the measurement values of the input channels can be asked for. All the settings for the PB6000 and the CA2CF card(s) can be adjusted through this communication channel.
To read and/or adjust those settings in a sophisticated manner, a software package is available. This package contains the following parts:

- Signalog server
- PB6000 Active X control
- CA2CF Active X control

Those 3 modules offer the user a way to read and change the settings as the user wants.
The settings are made in the Active X Controls. The Signalog server will be activated by a Active X control, and will maintain the communication, through the RS232 channel, with the connected PICAS or Signalog 6000 system.
The PICAS and the Signalog 6000 system are build on the same hardware and software components. Therefore the functionality of both the systems is the same. The only difference is that the PICAS holds 2 CA2CF cards, with a total of 4 channels, and the Signalog 6000 system can hold 13 cards, with a total of 26 channels.
In the rest of this manual, where the term Signalog system is used, the subject is valid for both the PICAS and the Signalog 6000 system. Only the differences between the system are mentioned separately.

1.1 Channel numbering

One Signalog system can hold 52 channels at maximum (8 for PICAS). Each position, which can hold an amplifier card, is called a Slot. Those slots are numbered from 0 to 12 (0 to 1 for PICAS).
Each slot can address 4 input channels. In total a system can hold 4*13 = 52 channels (8 for PICAS). The channel are numbered from 0 to 51 (0 to 7 for PICAS), or through the slot number and slot-channel number. Channel number 6 can also be named as slot 1/channel 3.

2 PB6000 control

The PB6000 control takes care of the settings on the PB6000 processor card. Most of those settings can be accessed through the property box of the control. This property box has the following layout:

The actual functional meaning of the settings are explained in the PB6000 user manual.

2.1 Properties

The following properties are present in this control:
The type of variable is noted as done in Visual Basic.

Name	type	Description	
FWVersion	string	Firmware version number	Read only
SerialNumber	long	Serial number PB6000	Read only
DeviceNumber	integer	Device number. When changing the device number, the device number of the PB6000 will change also. Watch out in network configurations.	1 - 250
CFMaster	integer	CF master/slave setting	0 = slave 1 = master 2 = local
RS232Speed	integer	Speed RS232 port	0 = 4800 baud 1 = 9600 baud 2 = 19200 baud 3 = 38400 baud
RS485Speed	integer	Speed RS485 port	0 = 9600 baud 1 = 19200 baud

			2 = 38400 baud
MaxChannel	integer	Maximum number of channels	Read only 8 for PICAS 52 for Signalog
ValueUnit	integer	Determine in which unit the measurement value is displayed	0 =Output V 1 =Signal V/V 2 =Phys. Unit
DisplayTime	integer	Filter time for the actual measurement value	[0.1] second
Language	Integer	Language of the help file and the property box(es)	7 = German 9 = English 19 = Dutch
PeakValueActive	integer	Setting the peak value option	0 = no peakvalue detection 1 = peak value detection
Contrast*	integer	Picas display contrast setting	Read only
DatalogMode*	integer	Datalog interval multiplier setting	Read only 0 = 10 milliseconds 1 = 1 second
DatalogStarted*	integer	Datalog status information	Read only, sum of: 1 = datalog active + 2 = starttime available + 4 = buffer contains data

*) This setting is only available for devices with version 2.07 or higher firmware. Otherwise, the value of the property is undetermined.

2.2 Methods

The following methods are present in this control:

Name	**return**	**Parameter**	**Description**
CentralCalibrate			Calibrate all channels
CentralBalance			Balance all channels
StoreAllChannelsInFlash			Store all channel information in the FLASH memory
LoadAllChannelFromFlash			Load all channel information from the FLASH memory
CalibrateChannels		string	Calibrate those channels. The string holds the channel numbers, separated by a comma
BalanceChannels		string	Balance these channels. The string hold the channel numbers, separated by a comma
GetActualCardsInSlots	string		The string contains the type number for the card in each slot. When no card is present in a slot card type 255 will be returned for that slot. Commas separate the numbers.
SetSlotConfiguration		string	The string contains the desired card configuration of the system. For the slots

			which contain no card, the type number 255 must be used. Commas separate the type numbers.
GetSlotConfiguration	string		This string contains the desired card configuration, which is present in the system. For a slot which contains no card the type number 255 is present. Commas separate the type numbers.
GetInfoFromDevice			After this command, all information will be retrieved from the system. The properties of the control are updated from those values.
SelectDevice		integer	With this command, the device number of the control is changed. The device number of the connected system remains unchanged. This function must be used to "connect" the device control to a signalog system.
ConfigDatalog	integer	string	With this string the datalog configuration is sent the to system. Detailed description below.
StartDatalog	integer		This command will start the datalog action. If no datalog configuration data is present, the value of "1" will be returned, and the datalog action will not be started.
StopDatalog	integer		This command will stop the datalog action.
AllowDatalogInfo	boolean		0 = no datalog information is send to the capplication 1 = datalog information is send to the application.
ResetPeakValues			The peakvalues in the device are reset.

The ConfigDatalog method in detail:
The string parameter holds the sample time for each channel. A comma separates the values for the channels.
A value of 0 means that this channel is not measured during datalog actions.
A value of 1 will measure the channel each 0.01 second.
A value of 100 will measure the channel each second. This is the maximum value for each channel. When a higher value is present here, this channel number will be returned, and the configuration will not be accepted.
A maximum of 100 sample in each second can be made. If the total requested samples in this message is too high, the number of the channel will be returned at which the maximum is reached.
If the configuration is accepted the return value will be –1. The message is then not accepted, and the datalogging can not be started.
For firmware version 2.07 and higher: the interval times can be higher than 100. If all specified values are dividable by 100, the device will set the interval for all channels as a multiple of 1 second, between 1 and 250 seconds. Higher intervals (eg. 300 seconds) are silently reduced to the maximum of 250 seconds.

If at least one value is not dividable by 100, the device will set the interval for all channels as a multiple of 10 milliseconds, between 0.01 and 2.5 seconds. Higher intervals (eg. 3 seconds) are silently reduced to the maximum of 2.5 seconds.

2.3 Events

This control generate the following events:

Name	Parameter	Description
NewDatalogInfo	string	New datalog information is received from the Signalog system This information is present in the string. No internal buffering is done, so the user must handle this message, to prevent that the datalog information is lost. See the description of the datalogging for the format of this string.
CardInSlotsChanged	string	A card configuration is received from the Signalog system, which is different than the configuration in the control. The received card configuration is present in the string.
NewDatalogSettings	String	New datalog settings are received. See below for details.
CommStatusChanges	integer	0 = there is no communication between de Signalogserver and the device 1 = the communication between the device and the signalogserver is ok.

The NewDatalogSettings event in detail:
The string parameter contains the datalog intervals as multiples of 0.01 seconds for each channel, separated by commas.
For firmware version 2.07 and higher: if the first value in the comma-separated string equals 255, the interpretation is different.
In this case, the second value contains extra datalog information. It is the sum of the following values:
1 if datalog is active, 0 otherwise, +
2 if the datalog buffer contains data, 0 otherwise +
4 if the measurement has an unknown starttime, 0 if the starttime is available +
8 if the intervals are multiples of 1 second, 0 if they are multiples of 0.01 seconds.
The remaining values (third and up) contain the intervals for each channel, which must be multiplied by either 1 second or 0.01 seconds.

2.4 Setting up the control

When the control is created in the main application the next sequence of action must be done, to get the control working properly:

1. Create the control
2. Perform the “SelectDevice” method with the device number you want to handle with this control

3. Perform the “GetInfoFromDevice” method, to get all the setting out of the connected device
4. Wait for the “CardInSlotsChanged” event. Now all the information is present in the control, and all properties can be read.

If you set a property to a desired value, the control and the Signalog Server will take care that this value is send to the device (if there is communication between the Signalog Server and the addressed device.

Example:
When a Signalog 6000 system contains a CA2CF card in slot 0 and slot 3 the string received with the “CardInSlotsChanged” event will be:

“13,0,255,255,0,255,255,255,255,255,255,255,255,255”

The SetSlotConfiguration and GetSlotConfiguration methods uses the same string layout.

2.5 Datalog configuration

The Signalog system is capable of automatic measuring inputs and storing the values in internal RAM memory. The Signalog system will sample 1 channel every 10 milliseconds. In total the system can make 100 samples every second.
With the “ConfigDatalog” method, a string is send to the device. This string contains the sample period for each channel, separated by a ‘comma’ sign. The sample period is a number which is a multiply factor for the 10 milliseconds interval.

Example string:
When the next configuration is wanted:

Channel number	Sample time [seconds]
0	0.1
1	0.2
4	0.05
5	0
8	1
9	2

The string will be: “10,20,5,0,100,100”
In the above string, the sample time for channel 5 is set to ‘0’, This means that this channel will not be measured when the datalog action is started.
For channel 9 the sample time is set to 2 seconds, but the requested sample time is 1 seconds. The sample time for the channels not mentioned in this message will be set to zero. Those channels will not be measured during datalogging.
Firmware version lower than 2.07: The Signalog system will not accept sample time, which are longer then 1 second. If slower sampling is required, the main application must ask 1 second sampling from the device, and ignore the samples that are not required.
Firmware version 2.07 and higher: The Signalog system can either use 0.01 seconds intervals (between 0.01 seconds and 2.5 seconds) or 1-second intervals (between 1 second and 250 seconds). The 1-second interval is only used when all specified sample times are dividable by 100. Examples:
"10,20,5,0,100,200" is accepted by firmware v2.07 and higher for the example above. There is no need to request a sample time of 1 second instead of 2 for the last channel.
"100,300" sets the sample times for channels 0 and 1 to 1 and 3 seconds, respectively. It will use the new 1-second interval mode.
"10,300" sets the sample times for channels 0 and 1 to 0.1 and 2.5 seconds, respectively. The 0.01 seconds interval mode is used, and the 3 seconds interval is silently reduced to the maximum available 2.5 seconds for this mode.

The datalog action will not be started until a “StartDatalog” method is given. The datalog will stop, when a “StopDatalog” method is give, or when the Signalog system shuts down.

2.6 Datalog information

As explained before, every 10 millisecond 1 channel is measured. Because the interval time between two measurements of 1 channel is configurable, it must be clear at what

time a measurement is done. For this purpose a sample time index is stored with each measured value.
This index represents the number of 10 milliseconds intervals after the start time of the datalog action. To optimize storage, the index has a maximum value of 255. After this it gets the value 0 again.
When datalog information is received, the first field will be the start date/time of the information when the index was 0. Each value in the string has an index of measurement. The actual time of measurement must be calculated from the preceding date/time and this index. Each time the index overflows in one string, that is when the index is smaller than the proceeding index, 2.56 seconds must be added to the calculated measurement time.
Firmware version 2.07 and higher: If the device uses the new 1-second interval mode, the index represents the number of 1-second intervals after the start time of the datalog action. Each time the index overflows in one string, that is when the index is smaller than the proceeding index, 256 seconds must be added to the calculated measurement time.
In a string with the datalog information, all the fields are separated with a comma sign.

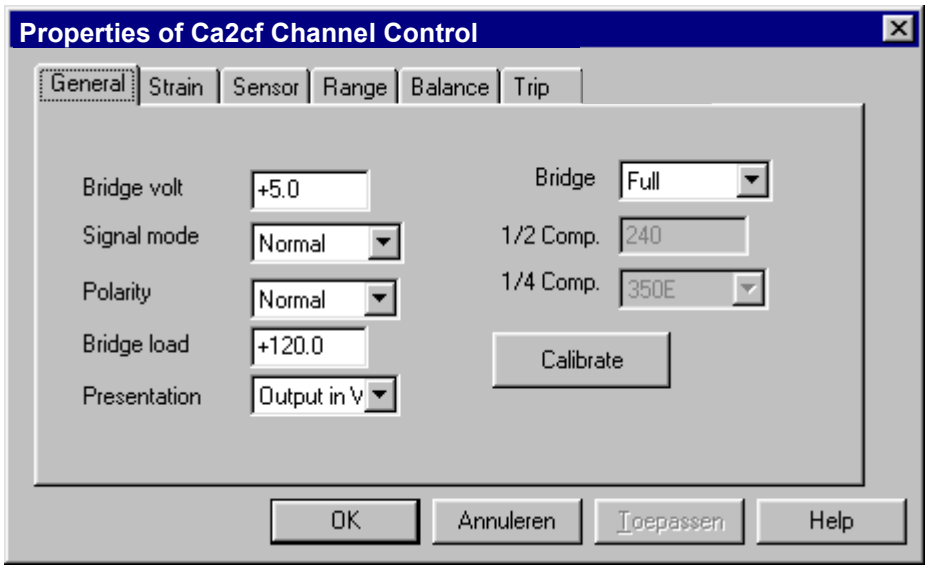
When datalog information is received from the device it will have the following format:

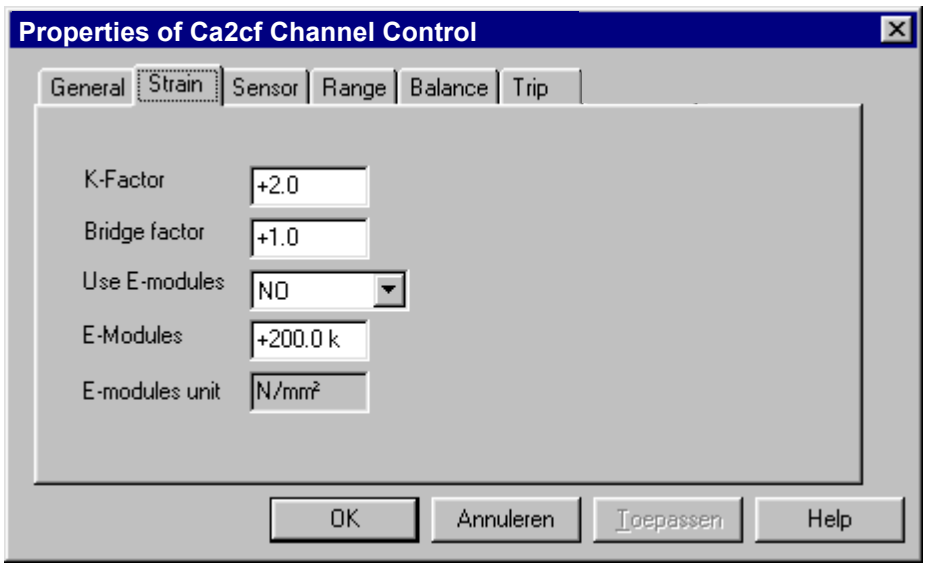
Start date and time, consist of:
- Start date/time, in seconds since 1 January 1970. This value equals 0 if the measurement was started on the device itself, because it doesn't know the actual date and time. <comma>
- Offset to start date/time in 10 milliseconds units (0…99). This value equals 255 if the measurement uses 1-second interval mode. <comma>

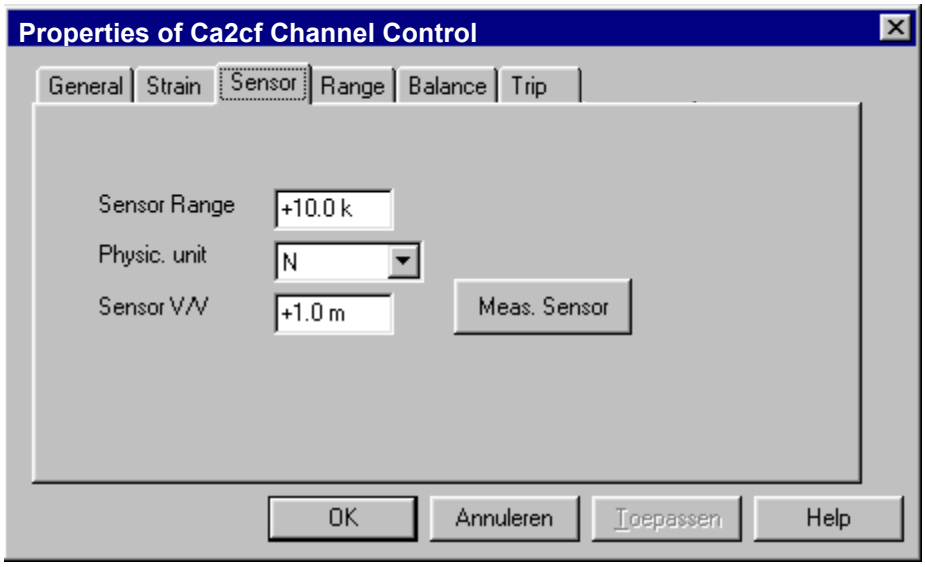
Measured value consist of:
- Index <comma>
- Channel number <comma>
- Value <comma>

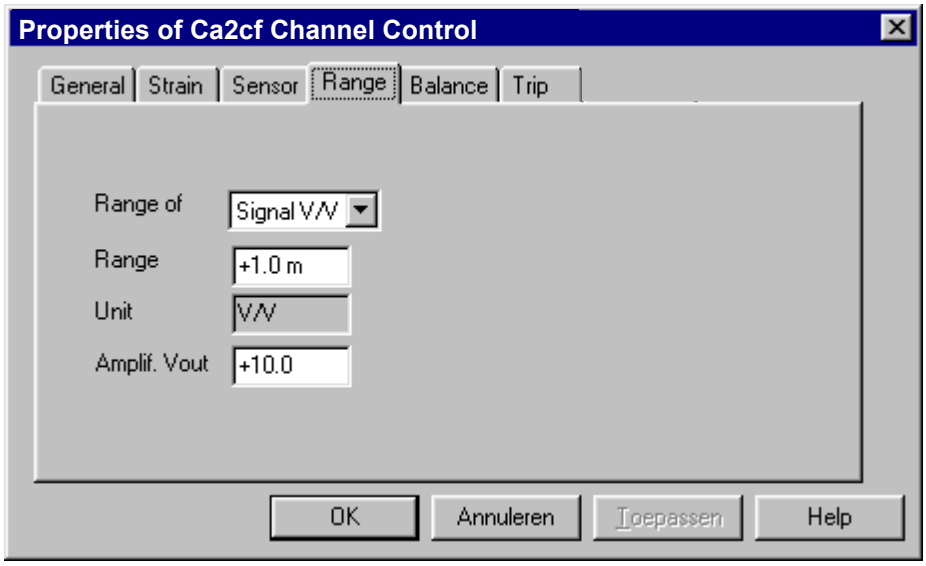
3 CA2CF control

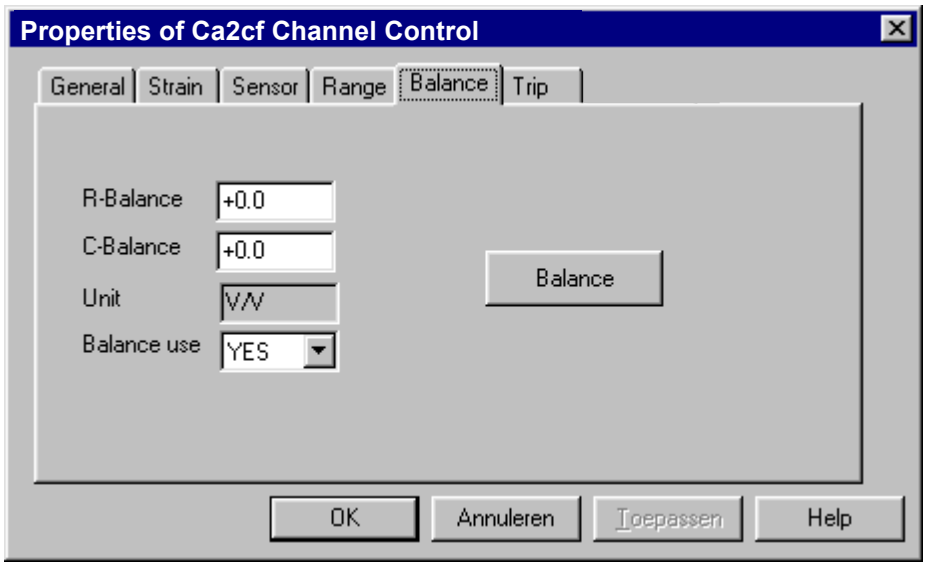
The CA2CF control takes care of the settings of the CA2CF input card. Most of these settings can be adjusted through the property box of the control. This property box has the following layout:

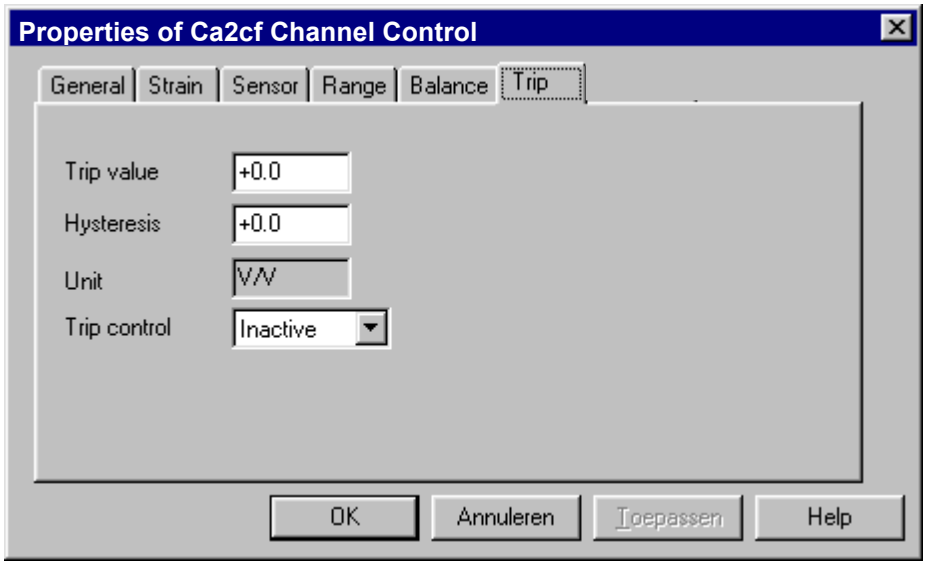





Properties of Ca2cf Channel Control
General
Strain
Sensor
Range
Balance
Trip
Sensor Range
+10.0 k
Physic. unit
N
Sensor V/V
+1.0 m
Meas. Sensor
OK
Annuleren
Toepassen
Help


Properties of Ca2cf Channel Control
General
Strain
Sensor
Range
Balance
Trip
Range of
Signal V/V
Range
+1.0 m
Unit
V/V
Amplif. Vout
+10.0
OK
Annuleren
Toepassen
Help


Properties of Ca2cf Channel Control
General
Strain
Sensor
Range
Balance
Trip
R-Balance
+0.0
C-Balance
+0.0
Unit
V/V
Balance use
YES
Balance
OK
Annuleren
Toepassen
Help



3.1 Properties

The following properties are present in this control:
The type off variable is noted as done in Visual Basic..

Name	type	Description	
BridgeSupply	single	Bridge supply	0.5 ⇔ 5.5V
Polarity	integer	Polarity of amplifier	0 = normal 1 = inverted
SignalMode	integer	Signal mode of amplifier	0 = normal 1 = capacitive
BridgeLoad	single	Bridge impedance	60 Ω ⇔ 3000Ω
RBalanceValue	single	R Balance value in V/V	- 65 mV/V ⇔ 65 mV/V
CalcRBalance	single	R Balance value in the unit of the range	Set this value only if all the other properties are set. This is a calculated value from RBalanceValue
CBalanceValue	single	C Balance value in V/V	-10 mV/V ⇔ 10 mV/V
CalcCBalance	single	C Balance value in the unit of the range	Set this value only if all the other properties are set. This is a calculated value from CBalanceValue
BalanceUse	integer	Balance in use	0 = not used 1= used
MaxVOut	single	Maximum output voltage	1 ⇔ 10 V
Range	single	Amplifier range. This range depends on the setting of the “range of” property.	100uV/V ⇔ 1V/V for amplifier range
RangeOf	integer		0=Amplifier 1=Sensor

			2=Strain
RangeUnit	string	Unit of the selected range	
SensorUnit	integer	A selection can be made for the Unit of sensor measurement	0=% 1= bar 2= øC 3= g 4= g/mm2 5= G 6= Hz 7= inch 8= K 9= lbs 10= m 11= m/m 12= m/s 13= m/s2 14= N 15= Nm 16= N/mm2 17= Pa 18= ppm 19= psi 20= t 21= V 22= V/V
SensorRange	single	Range of sensor measurement	
SensorInput	single	Input signal from sensor	
EModules	single	Emodules value	
KFactor	single	K factor of the bridge	
BridgeFactor	single	Bridge factor of the bridge	
EModUse	integer	Use of E modules	0 = Not used 1 = in use
EModUnit		Fixed Unit of force	N/mm^2
DeviceNumber	integer	Number of the device to which this control belongs	1-253
SlotNumber	integer	Number of slot to which this control belongs	0-12
ChannelNumber	integer	Number of channel in the slot to which this control belongs	0-3
MessValue	single	The measurement value	
FullHalfBridge	integer	Setting for half bridge completion	0 = Full bridge 1 = Half bridge
KW120E350E	integer	Setting for quarter bridge completion	0 = 120 Ω 1 = 350 Ω
UnitofValue	string	Unit of measurement value	
ValueUnit	integer	Unit of value at the MessValue property When default is selected, the	0=Default 1=Vout 2=Vin

		unit will be used as selected in the PB6000 control	3=Phys.Unit
Language	Integer	Language of the help file and the property box(es)	7 = German 9 = English 19 = Dutch
TripValue	single	Value for trip action in [V]	
CalcTripValue	single	Value for trip action in the unit of the range	Set this value only if all the other properties are set. This is a calculated value from TripValue
TripHysteresis	single	Hysteresis for trip action in [V]	
CalcTripHysteresis	single	Hysteresis value in the unit of the range	Set this value only if all the other properties are set. This is a calculated value from TripHysteresis
TripControl	integer	Selection for trip action	0= inactive 1= trip on high value 2= trip on low value
MaximumValue	single	This value is the maximum value of the measurement.	

3.2 Dependence of properties

Several properties depend on each other. This means that a property can change, when a property, from which it depends, changes.
The next dependencies exist:

Property to be changed	changed in:	Related property	changed in:
FullHalfBridge	Full bridge	KW120E350E	120 Ω
ValueUnit	Vout	UnitofValue	'V'
ValueUnit	Vin	UnitofValue	'V/V'
ValueUnit	Phys.Unit	UnitofValue	RangeUnit
RangeOf	Amplifier	RangeUnit	'V/V'
		Range	Amp. range
		MaximumValue	New value
RangeOf	Sensor	RangeUnit	SensorUnit
		Range	SensorRange
		MaximumValue	New value
RangeOf	Strain	RangeUnit	Strain unit
		Range	
		MaximumValue	New value
EModUse	Not used	Strain unit	"m/m"
EModUse	Used	Strain unit	EModUnit
CalcTripValue	value	TripValue	New value
TripValue	value	CalcTripValue	New value
TripHysteresis	value	CalcTripHysteresis	New value
CalcTripHysteresis	value	TripHysteresis	New value

3.3 Methods

The following methods are present in this control:

Name	return	Parameter	Description
Calibrate			Calibrate the bridge supply
AutoBalance			Balance the channel
MeasureSensor			Measure the sensor, and calculate the belonging range setting.
StoreSettingsinFLash			Store the channel settings in the FLASH memory
LoadSettingsFromFlash			Load the channel settings from FLASH memory
LoadDefaultSettings			Set the channel setting to default
GetInfoFromDevice			Get all the settings from the channel, and put them in the control
SetLocalParam			After this command the properties will only be changed in the control, and not passed to the device.
SendAllParam			All changed settings will be send to the device at once.

When a property is changed in the control, this change will also direct be send to the device, so that this change will directly be in effect. When al lot of properties must be changed, it will take some time before all the changes are transmitted to the device. To speed up this process, the methods SetLocalParam and SendAllParam are added. First activate the SetLocalParam method. Now the changes will be made only in the control. Now change all properties you want to change, and then activate the SendAllParam method. Only one message will be send to the device, which contains all the changes.

3.4 Properties and Methods

If a method is called, it is possible that certain properties are changed. The next dependencies are present:

Method	property	Changed in:
AutoBalance	RBalanceValue	Value measured by balance command.
	CBalanceValue	Value measured by balance command
MeasureSensor	SensorInput	Value measured by measure command
LoadSettingsFromFlash	All properties	Properties are set to the values, which are last stored in the FLASH memory
LoadDefaultSettings	Alle properties	Properties are set to the default value
GetInfoFromDevice	Alle properties	The properties in the control are set to the values in the device

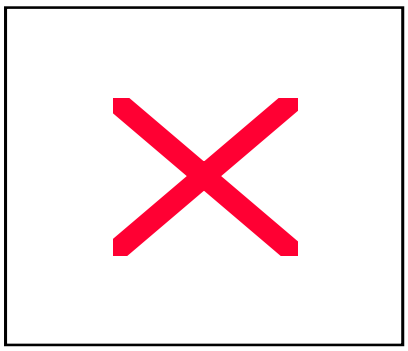
3.5 Events

The control generates the following events:

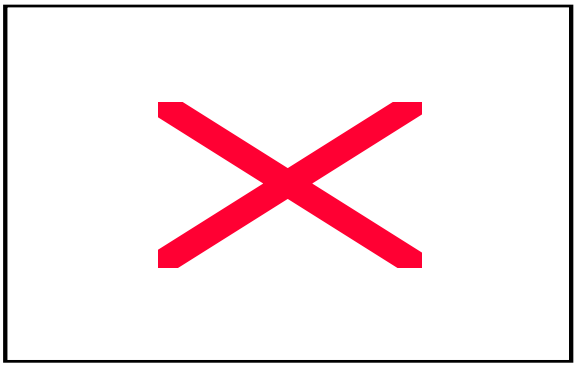
Name	**Parameter**	**Description**
NewMessValue	single	The newest measured value
UnitOfValueChanged	string	The unit of the measured value is changed
MaximumValueChanged	single	The maximum value of the measurement is changed
RBalanceChanged	single	The R Balance value is changed
CBalanceChanged	single	The C Balance value is changed
NewPeakValue	single	A new Peak value is received
NewTripStatus	single	A new trip status is received

4 Signalogserver

The Signalog server has only one setting. This setting defines the serial port which is used to communicate with the device.
This setting is made in the next Signalog Server menu:



After this command, the next dialog box appears:



Because this setting is stored in the window registry, it has to be made only once.
At default the com1 port is used.
To make this setting, the Signalog Server program has to be started manually, with the explorer, or from the desktop. When it starts on demand of an Active X Control, the window of the program will not been shown and the command cannot be given.

If desired the setting can also be made by hand in the window registry. The next entry must be adjusted:

HKEY_CURRENT_USER\Software\Peekel Instruments\SignalogServer\Comm poort setting Poort nummer=2

4.1 User the server on a remote PC

Ordinarily, the Signasoft application, which uses the Actice X Controls, runs on the same PC to which the Signalog and/or Picas devices are connected. It is possible, however, to use one PC to connect the Signalog/Picas device(s) to and another PC to run application, for example Signasoft.
For this setup to work, the PC's must be connected by a Windows network. On the ‘Application” PC, you must install the Active X Controls as usual.
On the remote PC, where the devices are connected to, the Active X Controls must also be installed.
Now, you must configure the DCOM communication protocol, which is used to communicate between application and the SignalogServer on the remote PC. Below, DCOM configuration is described in general terms. Where ever the word 'server' is used, you should substitute 'SignalogServer' or 'device PC', and where the word 'client' appears, you can read it as ‘Application’.

4.1.1 DCOM configuration

DCOM (Distributed Component Object Model) allows Windows applications to execute code on remote computers, using a Windows network and DCOM support software. On a default Windows PC, this type of 'remote execution' is not allowed. To enable DCOM on a PC, you must use a configuration tool called 'dcomcnfg'. Start this tool by selecting 'Run…' from the Start menu and typing 'dcomcnfg'.
If you are using Windows 95/98, it is likely that this tool is not installed. If so, Windows will indicate that 'dcomcnfg' cannot be found.
To install DCOM and dcomcnfg on Windows 95, run the following applications:

dcom95.exe
dcm95cfg.exe

To install DCOM and dcomcnfg on Windows 98, run the following applications:

dcom98.exe
dcm95cfg.exe

On Windows NT, DCOM and dcomcnfg are installed by default.

4.1.2 Dcomcnfg on Windows 95/98

To configure DCOM on Windows 95/98, you must now whether the PC is part of a Windows NT Domain. If this is the case, the PC will use a Windows NT Domain Controller in the network to verify username and password before allowing anyone access to the network. Consult your network administrator if you are unsure about this. If your PC is not part of a Windows NT Domain, you must disable the DCOM access security to be able to communicate with another PC using DCOM.

4.1.3 Windows 95/98 in a Windows NT Domain

If you are running Windows 95/98 in a Windows NT Domain, the DCOM configuration is basically the same as the Windows NT DCOM configuration as described below under 'Dcomcnfg on Windows NT'.

4.1.4 Windows 95/98 outside a Windows NT Domain

A Windows PC without access to a Windows NT Domain has no way to verify the username/password of users on other PC's. Therefore, it cannot use the builtin security of DCOM. To be able to use DCOM on this type of machine, you must disable the DCOM security.
You can do this by running 'regedit', and opening the key:
'HKEY_LOCAL_MACHINE\Software\Microsoft\Ole'
Add or replace a DWORD-value 'LegacyAuthenticationLevel' in this key and put in the value 1. You must reboot the PC for this change to take effect.
After this, DCOM access security is switched off. This means that DCOM applications on this PC can be started remotely by anyone within the Windows network!
If you make this change, it must be made on both the client and the server-PC, even if one of them is a Windows NT machine.
After making this change, select 'Run…' from the Start menu and type 'dcomcnfg' to configure DCOM. Answer any 'DCOM Configuration Warning' dialogs, if they appear, with 'Yes'. Now, click on the 'Default Properties' tab and make sure the 'Enable Distributed COM' checkbox is checked. On the 'Default Security' tab, check the 'Enable remote connection' box.
To get the client PC to run a server application on the server PC, go to the 'Applications' tab and find the server application. Select the application and click 'Properties'. Now, click on the 'Location' tab and check 'Run application on the following computer:'.
Type the name of the server PC. Depending on the type of network, you may use the 'Computer Name' which can be found on the 'Identification' tab of the 'Network' settings in the 'Control Panel', or use a TCP/IP address, or consult your network administrator.

4.1.5 Dcomcnfg on Windows NT

Select 'Run…' from the Start menu and type 'dcomcnfg' to configure DCOM. Answer any 'DCOM Configuration Warning' dialogs, if they appear, with 'Yes'. Now, click on the 'Default Properties' tab and make sure the 'Enable Distributed COM' checkbox is checked.
To get the client PC to run a server application on the server PC, go to the 'Applications' tab and find the server application. Select the application and click 'Properties'. Now, click on the 'Location' tab and check 'Run application on the following computer:'.
Type the name of the server PC. Depending on the type of network, you may use the 'Computer Name' which can be found on the 'Identification' tab of the 'Network' settings in the 'Control Panel', or use a TCP/IP address, or consult your network administrator.
On the 'Security' tab for the application, you can select 'Use custom access permissions' and click 'Edit…' to determine which users are allowed to access the server.

4.1.6 DCOM Error Codes

During the configuration it is possible that an error will occure. The possible errors, with an short explanation are mentioned here:

1722 = RPC_S_SERVER_UNAVAILABLE
Possible causes:
'Enable Distributed COM' in dcomcnfg not checked on the server.

'Enable remote connection' in dcomcnfg not checked on the server (Windows 95/98 only).
'Launch permissions' in dcomcnfg do not allow the user access (Windows NT only).
An old version of DCOM (v1.0) is used (Windows 95 only). Update by reinstalling DCOM as described above.

2148007941 = CO_E_SERVER_EXEC_FAILURE
Possible causes:
'Enable remote connection' in dcomcnfg not checked on the server (Windows 95/98 only).
'Launch permissions' in dcomcnfg do not allow the user access (Windows NT only).

2147549466 = RPC_E_NO_GOOD_SECURITY_PACKAGES
Possible causes:
An old version of DCOM (v1.0) is used (Windows 95 only). Update by reinstalling DCOM as described above.

1747 = RPC_S_UNKNOWN_AUTHN_SERVICE
Possible causes:
You are trying to access a Windows 95/98 machine from a Windows NT machine using a security level not available to Windows 95. If you are using Windows 95/98 in a Windows NT Domain, select 'Default Authentication Level' = 'Connect' on the 'Default Properties' tab of dcomcnfg on the Windows NT machine. If you are using Windows 95/98 without a domain, make sure you disable the DCOM security on both server and client PC (DWORD value 'HKEY_LOCAL_MACHINE\Software\Microsoft\Ole\LegacyAuthenticationLevel' = 1)

1723 = RPC_S_SERVER_TOO_BUSY
Possible causes:
You are trying to access a Windows 95/98 machine from a Windows NT machine using a security level not available to Windows 95. If you are using Windows 95/98 in a Windows NT Domain, select 'Default Authentication Level' = 'Connect' on the 'Default Properties' tab of dcomcnfg on the Windows NT machine.
If you are using Windows 95/98 without a domain, make sure you disable the DCOM security on both server and client PC (DWORD value 'HKEY_LOCAL_MACHINE\Software\Microsoft\Ole\LegacyAuthenticationLevel' = 1)

2147942405 = E_ACCESSDENIED
Possible causes:
The remote machine does not accept calls from the current user. Make sure the security configuration is correct.

5 Register the Active X controls

Before the active x controls can be used, they must be registered on the PC. With a normal installation of the software those registrations will be done automatically.

How the manual registration must be done, is explained in the next section

Registration Signalog server:
Before the Signalog server can be registered, the file Atl.dll must be present in the system directory and be registered on the PC.

To register the Signalog server execute it with the following command:

Signalogserver.exe /RegServer

The Signalog server will register itself.

The Active X controls must be registered with the following commands:

regsvr32 /s /c ca2cf.ocx
regsvr32 /s /c pb6000.ocx

The program **regsvr32.exe** and **Atl.dll** are present in the delivered software package.