

Creating an amplifier module

An amplifier module provides G-clamp with the amplifier-specific scaling factors so that

- a) the raw voltage signals from the amplifier outputs can be correctly interpreted in units of mV for V_M and pA for I_M and
- b) the intended command to the amplifier is adequately scaled to its command input.

An amplifier module consists of three components:

- a list of scaling factors covering all recording configurations possible with a given amplifier. These are set up as individual tasks with National Instruments Measurement and Automation Explorer, a configuration utility for National Instruments hardware.
- a VI executing on the embedded controller that selects from the list of scaling factors the ones applicable to the current recording configuration. The recording configuration is either provided by user interaction on the front panel of G-clamp or determined from telegraph outputs of the amplifier.
- a VI executing on the host computer as part of the G-clamp front panel by which the user either specifies the current recording configuration or which serves as optical read-out of the amplifier telegraph outputs and their interpretation by G-clamp.

The following sections illustrate how these components work together by describing step-by-step how to create these components for a specific recording amplifier, the Axopatch 200B (Molecular Devices, formerly Axon Instruments).

The Axopatch 200B is a patch clamp amplifier designed for voltage clamp recordings. Although its “fast” current clamp recording mode might provide better current clamp capabilities than other patch clamp amplifiers we do NOT recommend the use of this amplifier for dynamic clamp recordings.

To build the amplifier module for the Axopatch 200B the following outputs and inputs will be used (connect to the analog-in and analog-out channels of the BNC-2090 breakout box as indicated in parentheses):

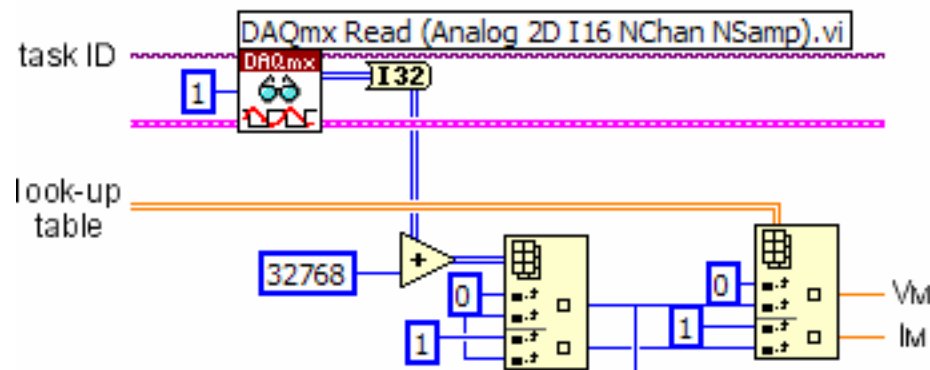
- the fixed output for V_M (analog-in 9),
- the fixed output for I_M (this output can be set to two different sensitivities via a toggle switch in the rear of the amplifier; analog-in 10),
- the variable (scaled) output for I_M (in voltage clamp mode) and V_M (in current clamp mode) which is scaled according to the setting of the output gain control (analog-in 8),
- the telegraph output for the setting of the output gain control (analog-in 14),
- the telegraph output for the recording mode (analog-in 13),
- the telegraph output for the setting of the output lowpass filter control (analog-in 12),

- the telegraph output for the setting of whole-cell capacitance control (analog-in 15),
- the front-switched input for external commands which is interpreted by the amplifier as voltage command in voltage clamp mode and as current command in current clamp mode (analog-out 0; the rear-switched command input will not be used).

In addition, a switch on the front panel of the Axopatch 200B selects one of the two feedback resistors in the CV 203BU headstage to provide a wide range of current-passing capacity in whole-cell mode.

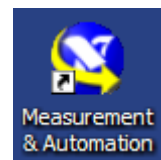
Setting up analog-in tasks for V_M and I_M

The analog input operations for reading the voltage and current output of the recording amplifier in the dynamic clamp feedback loop are set up to return raw, unscaled voltage signals in the form of a 2-D array of signed integers (I 16) ranging from -32768 to 32767¹. Adding 32768 (changing the range from 0 to 65536) allows to use these values to index a pre-determined look-up table that contains the scaled values (single precision numbers, SGL) for membrane potential (in mV) and membrane current (in pA).



Here we will first set up this analog input operation as DAQmx task, using National Instruments hardware configuration utility, Measurement and Automation Explorer (MAX).

(How the look-up tables are set-up will be detailed in the section dealing with the creation of the amplifier module VI executing on the embedded controller)

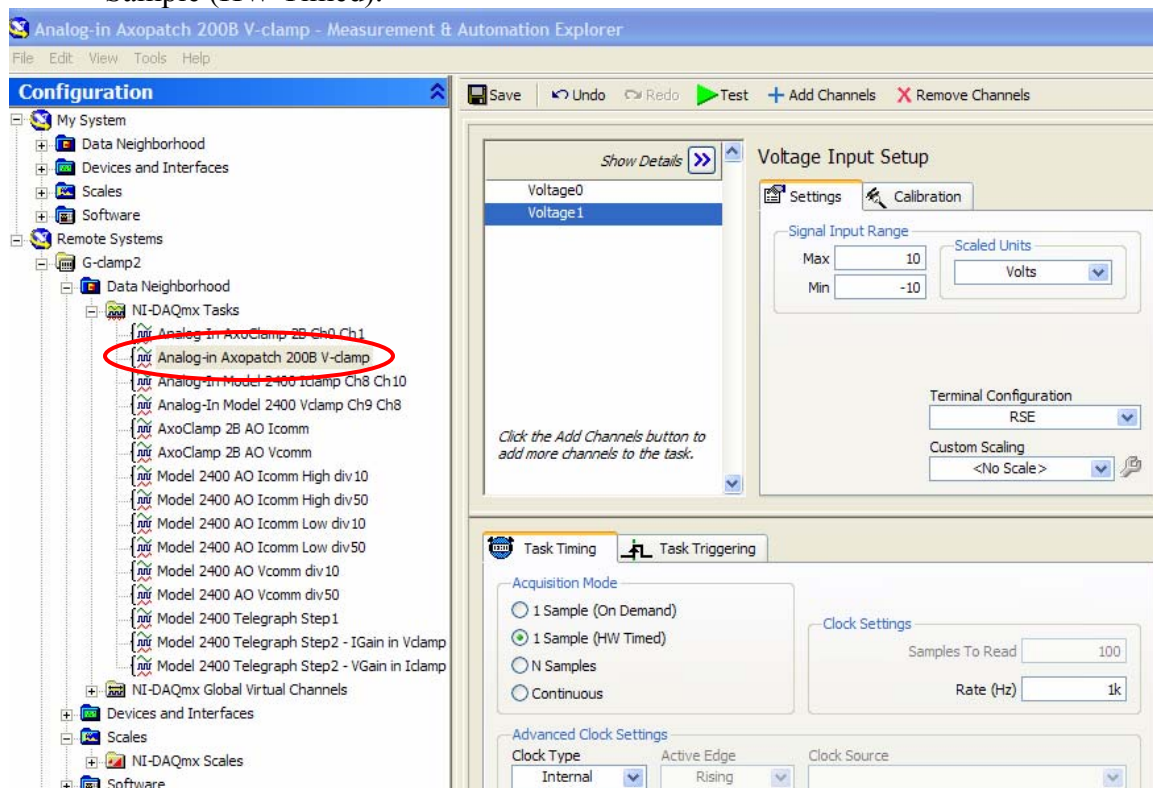


1. Boot the embedded controller and start MAX on the host computer.
2. Under *Remote Systems* double-click your embedded controller to access its configuration details.

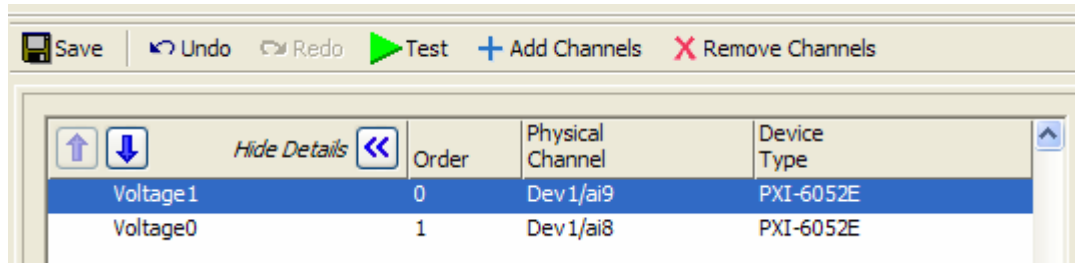
¹ This assumes that the data acquisition board uses 16-bit analog-to-digital converters.

Because the Axopatch 200B has a scaled output (V_M in current clamp and I_M in voltage clamp) which can be adjusted with the gain control to take advantage of the full ± 10 V working range of the data acquisition board we will set up two separate tasks: In voltage clamp G-clamp will use analog-in channel 9 for the V_M signal and analog-in channel 8 for the I_M signal. In current clamp the V_M signal will be taken from analog-in channel 8 and the I_M signal from analog-in channel 10.

3. In *Data Neighborhood* right-click *NI-DAQmx Tasks* and select *Create new NI-DAQmx Task...*
4. In the *Create New...* dialog box select *Analog Input > Voltage*. In the *Physical channels* tab select *ai8* and *ai9*. Next enter a name for this task, e.g. 'Analog-in Axopatch 200B V-clamp' and finish.
5. In the *Voltage Input Setup*, for each channel of the task (Voltage0, Voltage1), set the *Signal Input Range* to 10 and -10 Volts for Max and Min. *Terminal Configuration* for each channel should be RSE. In the *Task Timing* tab select 1 Sample (HW Timed).



6. Press *Show Details*. By using the up or down arrows change the order of the two analog-in channels so that Voltage1 (analog-in channel 9) precedes Voltage0 (analog-in channel 8) as G-clamp always interprets the first channel as V_M and the second channel as I_M .



7. Finally, press *Save* to add the new task permanently to the configuration of the embedded controller.
8. Repeat steps 4 and 5 to set up a task for the current clamp condition, using analog input channels 8 and 10.

Setting up analog-in tasks for telegraph outputs

For the telegraph outputs the analog input tasks will not be based on physical analog-in channels returning raw, unscaled voltage signals. Instead these tasks will use virtual channels which include pre-defined, custom scales. Using custom scales for the output of these analog input operations shifts the scaling of these signals to the hardware driver software and takes it out of the G-clamp code.

Because the scaling of the gain output generally depends on the recording mode, reading telegraph outputs will be done as two consecutive tasks: In the first task the mode, filter and capacitance telegraph outputs will be read. After the recording mode has been determined the second task will read the gain telegraph output.

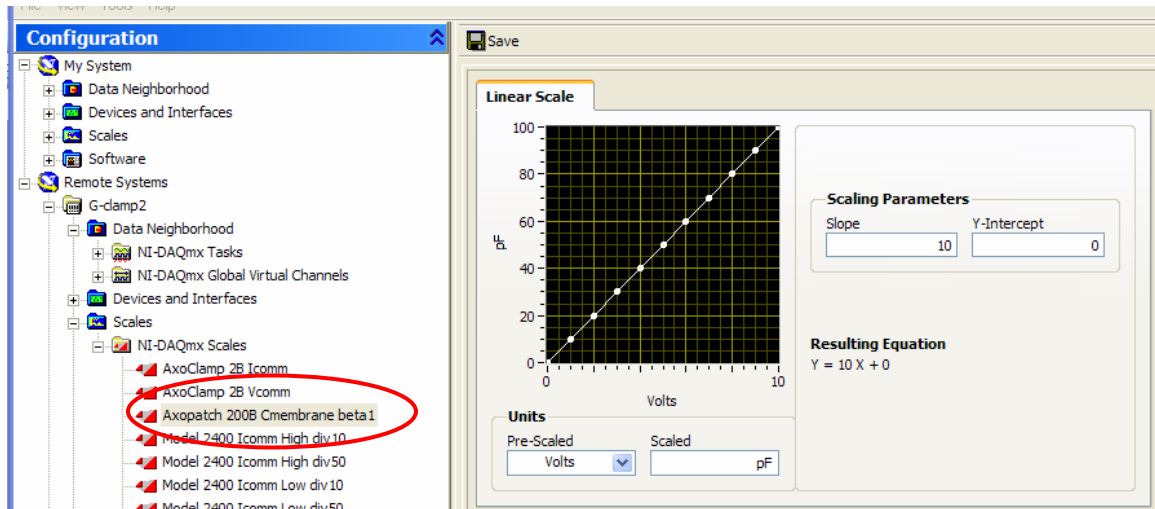
The general procedure for setting up one of these tasks is therefore first to define a NI-DAQmx Scale, second to use that scale when defining a NI-DAQmx Global Virtual Channel and third to use the virtual channel for defining a NI-DAQmx Task.

Cell capacitance

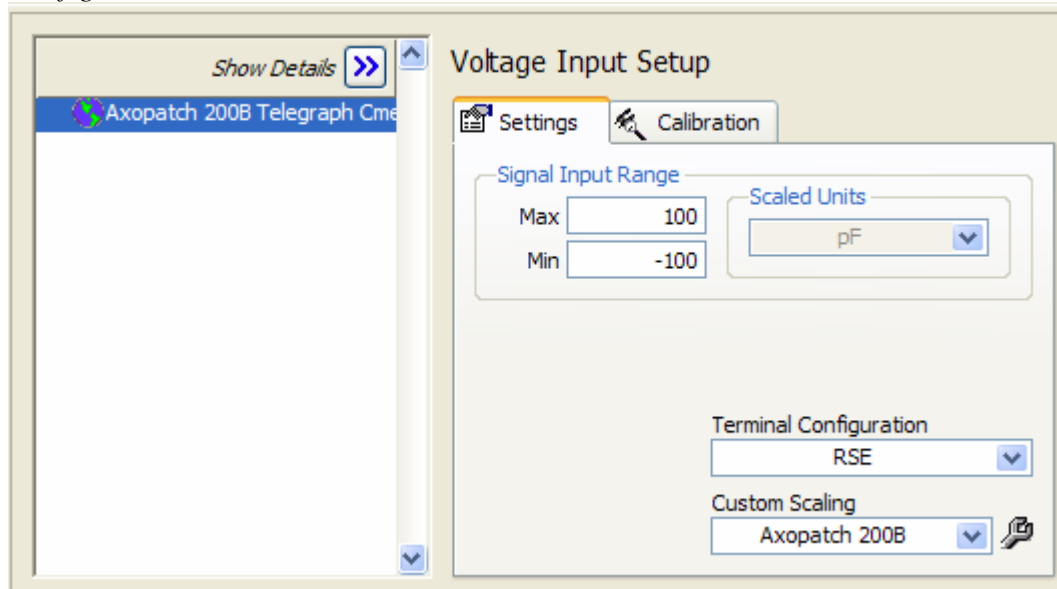
This telegraph output usually provides a voltage signal that is proportional to the setting of the whole-cell capacitance dial. In case of the Axopatch 200B amplifier: a voltage of 0 to 10 V corresponds to a setting of 0 to 100 pF (for $\beta = 1$; 0 to 1000 pF for $\beta = 0.1$) when whole-cell capacitance neutralization is active and a voltage of 0 to -10 V when whole-cell capacitance neutralization is off.

For the Axopatch 200B we therefore have to define two tasks corresponding to the setting for β .

1. In *Scales* right-click *NI-DAQmx Scales* and select *Create new NI-DAQmx Scale...*
2. In the *Create New...* dialog box select *Linear*. Enter a name for the scale, e.g. 'Axopatch 200B Cmembrane beta1' and finish.
3. In *Scaling Parameters* set the slope to 10 and in the *Scaled* field of *Units* enter 'pF'. *Resulting Equation* should read $Y = 10 X + 0$.

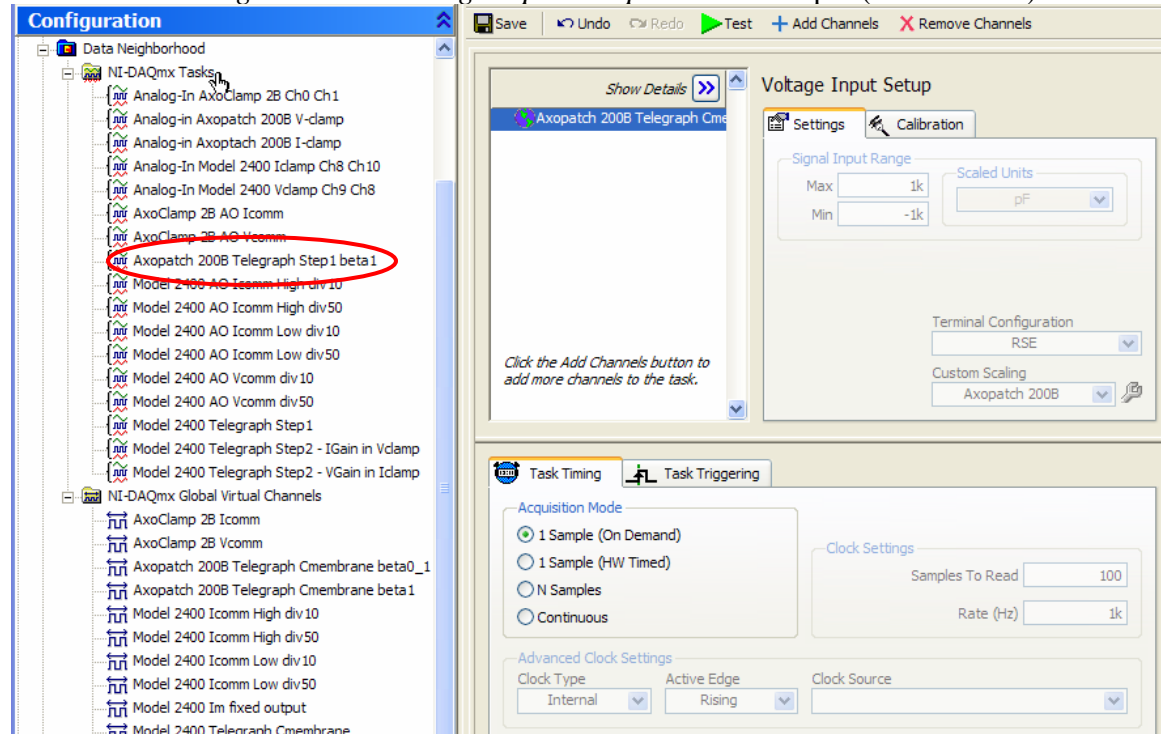


4. Press **Save** to add the new scale permanently to the configuration of the embedded controller.
5. Create another linear scale for $\beta = 0.1$ by repeating steps 2 to 4 (with *Slope* = 100). *Resulting Equation* should read $Y = 100 X + 0$.
6. In *Data Neighborhood* right-click *NI-DAQmx Global Virtual Channels* and select *Create new NI-DAQmx Channel...*
7. In the *Create New...* dialog box select *Analog Input > Voltage*. In the *Physical channels* tab select *ai15*. Enter a name for the virtual channel, e.g. 'Axopatch 200B Telegraph Cmembrane beta1' and finish.
8. In the *Voltage Input Setup*, set the *Signal Input Range* to 100 and -100 pF for Max and Min. In *Custom Scaling* select the scale defined in steps 2 to 4. 'Terminal Configuration' should be RSE.



9. Press **Save** to add the new virtual channel permanently to the configuration of the embedded controller.

10. Create another virtual channel for $\beta = 0.1$ by repeating steps 6 to 9 with the scale defined in step 5.
11. In *Data Neighborhood* right-click NI-DAQmx Tasks and select *Create new NI-DAQmx Task...*
12. In the *Create New...* dialog box select *Analog Input > Voltage*. In the *Virtual channels* tab select the virtual channel created in steps 6 to 9. Next enter a name for this task, e.g. 'Axopatch 200B Telegraph Step1 beta1' and finish.
13. In the *Task Timing* tab of the *Voltage Input Setup* select 1 Sample (On Demand).



14. Press *Save* to add the new task permanently to the configuration of the embedded controller.
15. Create another task for $\beta = 0.1$ by repeating steps 11 to 14 with the virtual channel defined in step 10.

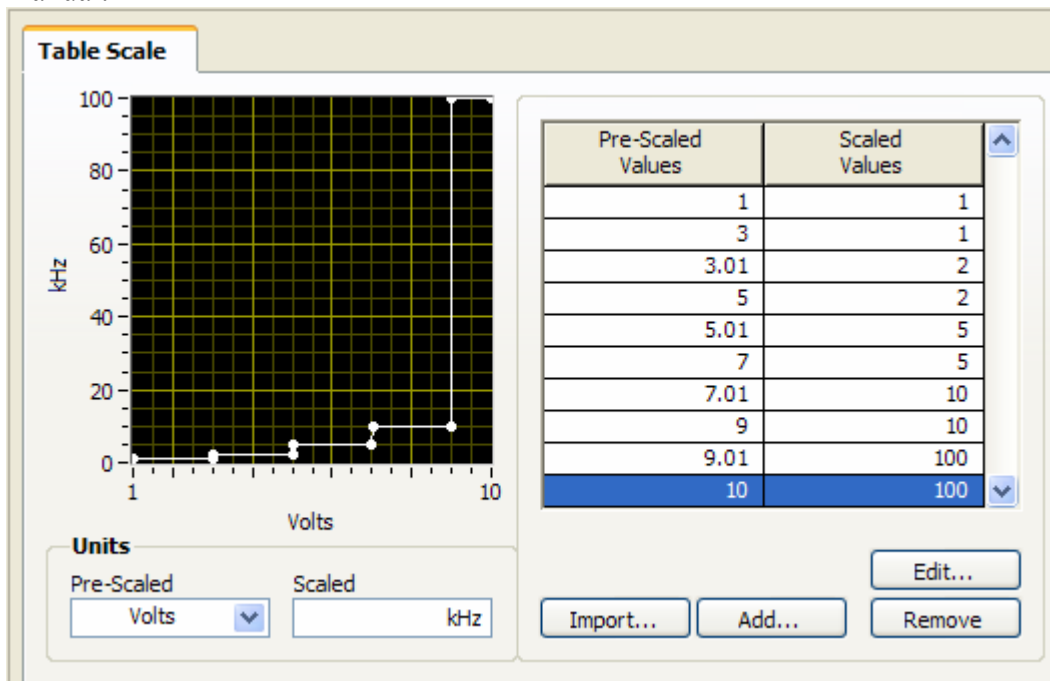
Filter frequency

This telegraph output usually provides a step-like signal where each step corresponds to one setting of the lowpass filter dial. In case of the Axopatch 200B amplifier:

Filter setting (kHz):	1	2	5	10	100
Telegraph Output (V):	2	4	6	8	10

16. In *Scales* right-click *NI-DAQmx Scales* and select *Create new NI-DAQmx Scale...*
17. In the *Create New...* dialog box select *Table*. Enter a name for the scale, e.g. 'Axopatch 200B FilterFreq' and finish.

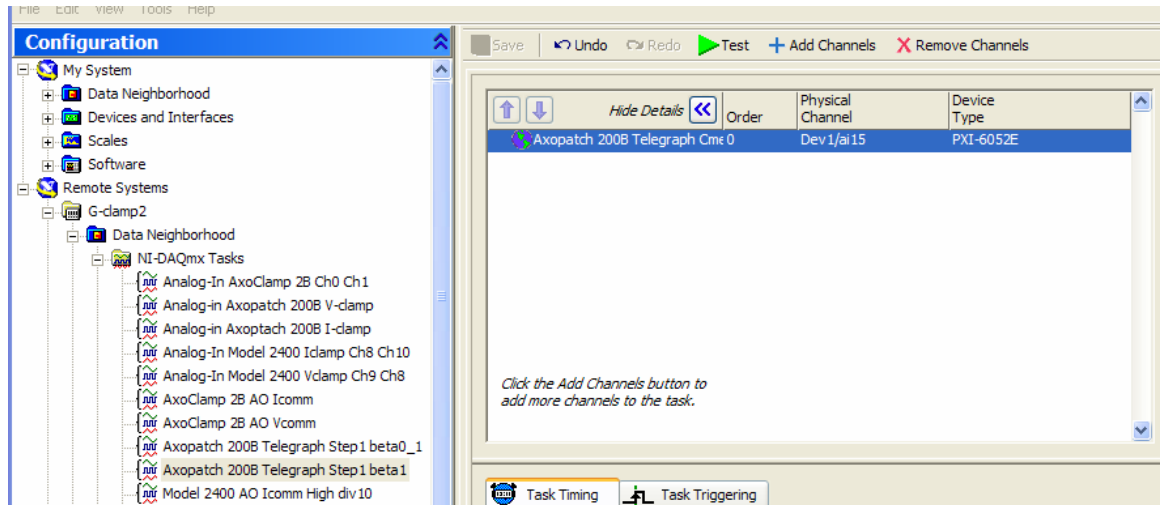
18. In *Table Scale* edit the table so that each filter frequency is represented by a voltage range centering on the nominal voltage value given in the manufacturer's manual.



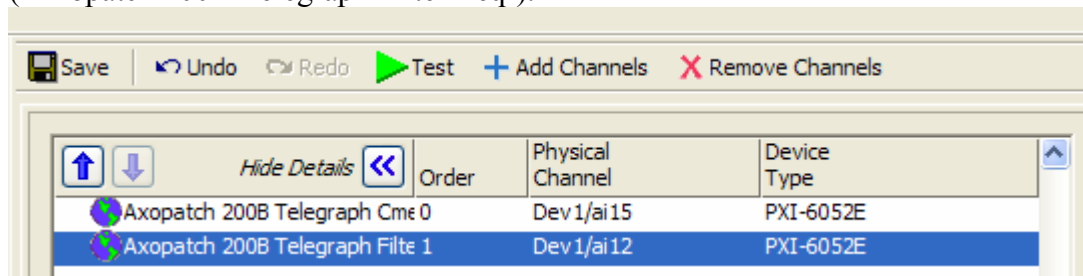
19. Press *Save* to add the new scale permanently to the configuration of the embedded controller.

20. In *Data Neighborhood* right-click *NI-DAQmx Global Virtual Channels* and select *Create new NI-DAQmx Channel...*
21. In the *Create New...* dialog box select *Analog Input > Voltage*. In the *Physical channels* tab select *ai12*. Enter a name for the virtual channel, e.g. 'Axopatch 200B Telegraph FilterFreq' and finish.
22. In the *Voltage Input Setup*, set the *Signal Input Range* to 100 and 1 kHz for Max and Min. In *Custom Scaling* select the scale defined in step 16 to 19. 'Terminal Configuration' should be RSE.
23. Press *Save* to add the new virtual channel permanently to the configuration of the embedded controller.

24. In *NI-DAQmx Tasks* of *Data Neighborhood* select the telegraph task created in the Cell capacitance section above ('Axopatch 200B Telegraph Step1 beta1').



25. Press **Add Channels** and select **Voltage**. In the **Add Channels To Task** dialog box select the virtual channel created in steps 20 to 23 from the **Virtual channels** tab ('Axopatch 200B Telegraph FilterFreq').



26. Press **Save** to make the changes in the task permanent.
27. For the Axopatch 200B amplifier repeat steps 24 to 26 with the NI-DAQmx task 'Axopatch 200B Telegraph Step1 beta0_1' for $\beta = 0.1$.

Recording Mode

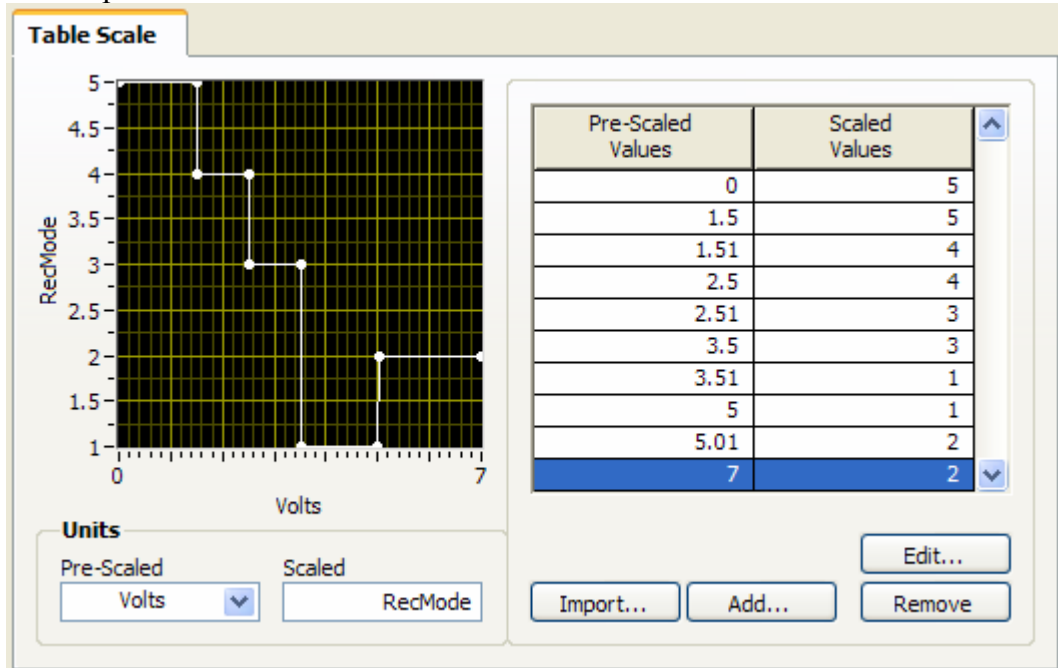
This telegraph output usually provides a step-like signal where each step corresponds to one possible recording mode. In case of the Axopatch 200B amplifier:

Mode:		I-clamp fast	I-clamp normal	I = 0	V- clamp	Track
Telegraph output (V):		1	2	3	6	4

28. In **Scales** right-click **NI-DAQmx Scales** and select **Create new NI-DAQmx Scale...**
29. In the **Create New...** dialog box select **Table**. Enter a name for the scale, e.g. 'Axopatch 200B Mode' and finish.
30. In **Table Scale** edit the table so that each recording mode (coded as a numerical value) is represented by a voltage range centering on the nominal voltage value given in the manufacturer's manual. The numerical coding for the recording modes used here is as follows:

I-clamp fast 5
I-clamp normal 4

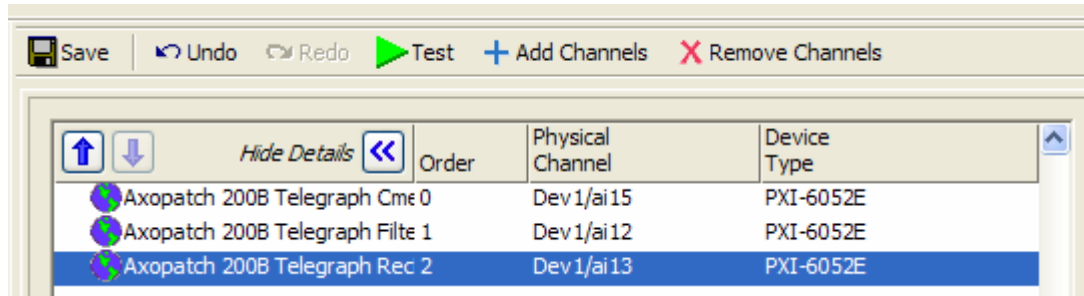
I = 0 3
 Track 1
 V-clamp 2



31. Press *Save* to add the new scale permanently to the configuration of the embedded controller.

32. In *Data Neighborhood* right-click *NI-DAQmx Global Virtual Channels* and select *Create new NI-DAQmx Channel...*
33. In the *Create New...* dialog box select *Analog Input > Voltage*. In the *Physical channels* tab select *ai13*. Enter a name for the virtual channel, e.g. 'Axopatch 200B Telegraph RecMode' and finish.
34. In the *Voltage Input Setup*, set the *Signal Input Range* to 5 and 1 for Max and Min. In *Custom Scaling* select the scale defined in step 2 to 4 ('Axopatch 200B Mode'). *Terminal Configuration* should be RSE.
35. Press *Save* to add the new virtual channel permanently to the configuration of the embedded controller.

36. In *NI-DAQmx Tasks* of *Data Neighborhood* select the telegraph task created in the Cell capacitance section above ('Axopatch 200B Telegraph Step1 beta1').
37. Press *Add Channels* and select *Voltage*. In the *Add Channels To Task* dialog box select the virtual channel created in steps 28 to 31 from the *Virtual channels* tab ('Axopatch 200B Telegraph FilterFreq').



38. Press *Save* to make the changes in the task permanent.
39. For the Axopatch 200B amplifier repeat steps 32 to 35 with the NI-DAQmx task 'Axopatch 200B Telegraph Step1 beta0_1' for $\beta = 0.1$.

This completes the first task for reading the telegraph outputs.

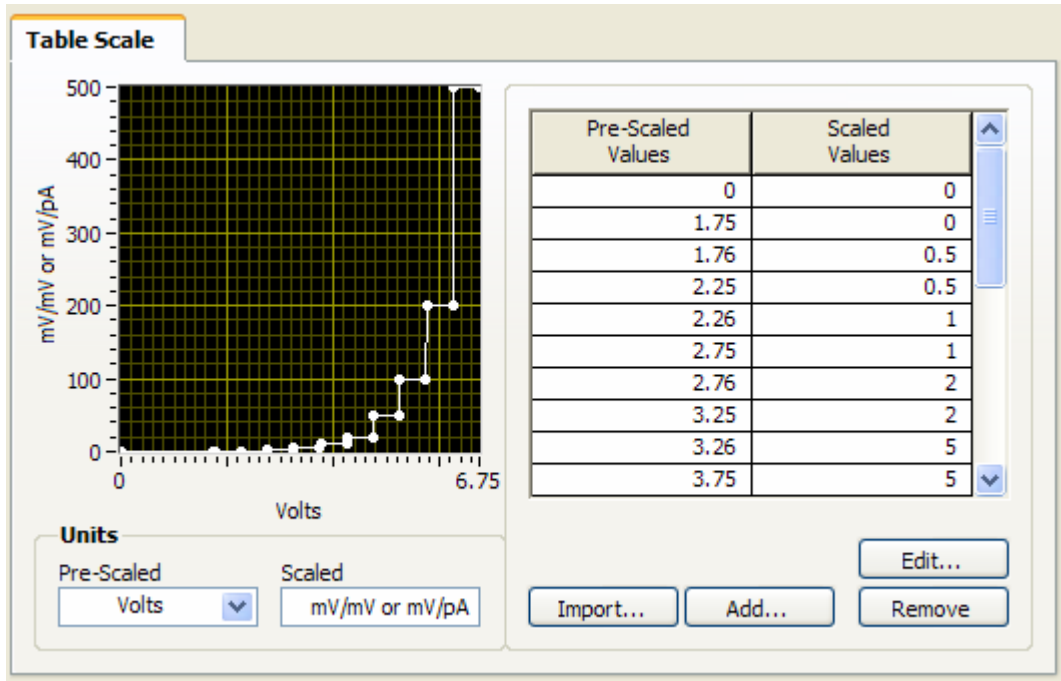
Gain

This telegraph output usually provides a step-like signal where each step corresponds to one setting of the gain dial. In case of the Axopatch 200B amplifier:

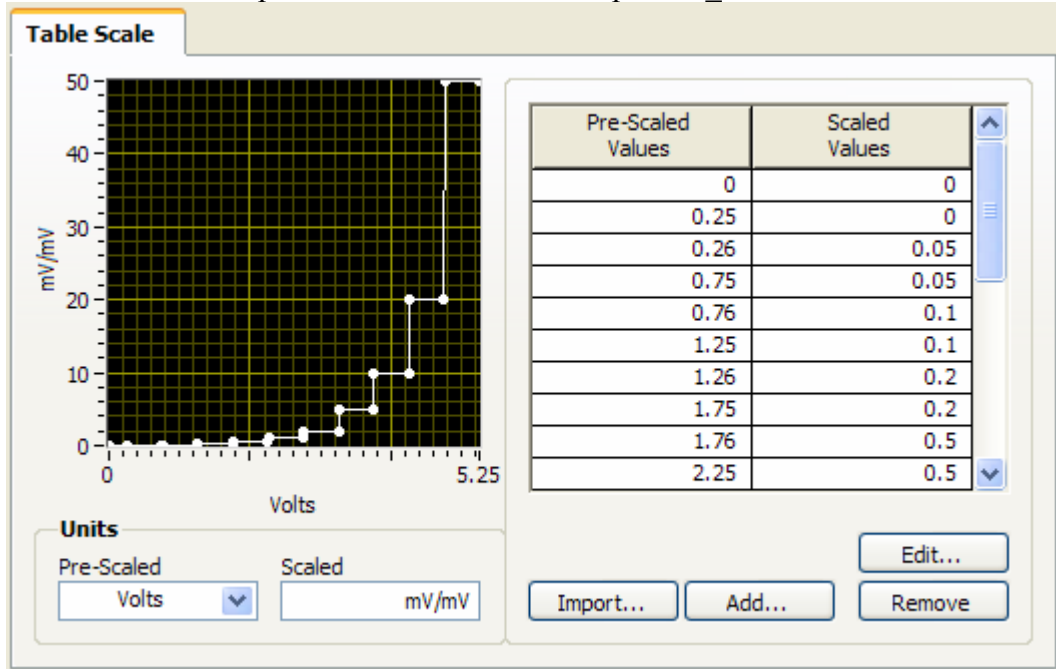
I ($\beta = 1$) (mV/pA):				0.5	1	2	5	10	20	50	100	200	500
I ($\beta = 0.1$) (mV/pA):	0.0	0.1	0.2	0.5	1	2	5	10	20	50			
V-clamp (mV/mV)				0.5	1	2	5	10	20	50	100	200	500
Telegraph output (V):	0.5	1	1.2	2	2.5	3	3.5	4	4.5	5	5.5	6	6.5

Because the scaling for I ($\beta = 1$) and V-clamp is numerically identical, these two operations can be performed using the same custom scale and virtual channel and thus can be set up as one task. A second task will be defined for the case I ($\beta = 0.1$).

1. Create a scale 'Axopatch 200B VGain in Iclamp beta1 and IGain in Vclamp'.



2. Create a scale 'Axopatch 200B VGain in Iclamp beta0_1'.



3. Create a virtual channel 'Axopatch 200B Telegraph VGain in Iclamp beta1 and IGain in Vclamp' using physical channel ai14 with the previously defined custom scale 'Axopatch 200B VGain in Iclamp beta1 and IGain in Vclamp'. Set the *Signal Input Range* to 0 to 500 for Min and Max.
4. Create a virtual channel 'Axopatch 200B Telegraph VGain in Iclamp beta0_1' using physical channel ai14 with the previously defined custom scale 'Axopatch 200B VGain in Iclamp beta0_1'. Set the *Signal Input Range* to 0 to 50 for Min and Max.

5. Create a task 'Axopatch 200B Telegraph Step2 VGain in Iclamp beta1 and IGain in Vclamp' using the previously defined virtual channel 'Axopatch 200B Telegraph VGain in Iclamp beta1 and IGain in Vclamp'. Set *Acquisition Mode* to '1 Sample (On Demand)'.
6. Create a task 'Axopatch 200B Telegraph Step2 VGain in Iclamp beta0_1' using the previously defined virtual channel 'Axopatch 200B Telegraph VGain in Iclamp beta0_1'. Set *Acquisition Mode* to '1 Sample (On Demand)'.

Setting up analog-out tasks

Setting up amplifier command operations requires definition of at least two different tasks: a current command task (current clamp) and a voltage command task (voltage command). These commands might feed into the amplifier via a single input (e.g. the Axopatch 200B) and thus can use the same analog-out channel of the data acquisition board or they might use separate inputs in which case each task can use a separate analog-out channel (e.g. Axoclamp 2B). In addition, some amplifiers allow different settings for the command sensitivity (Model 2400) or the headstage can be switched between different feedback resistors changing the scaling of the amplifier command (Model 2400, Axopatch 200B).

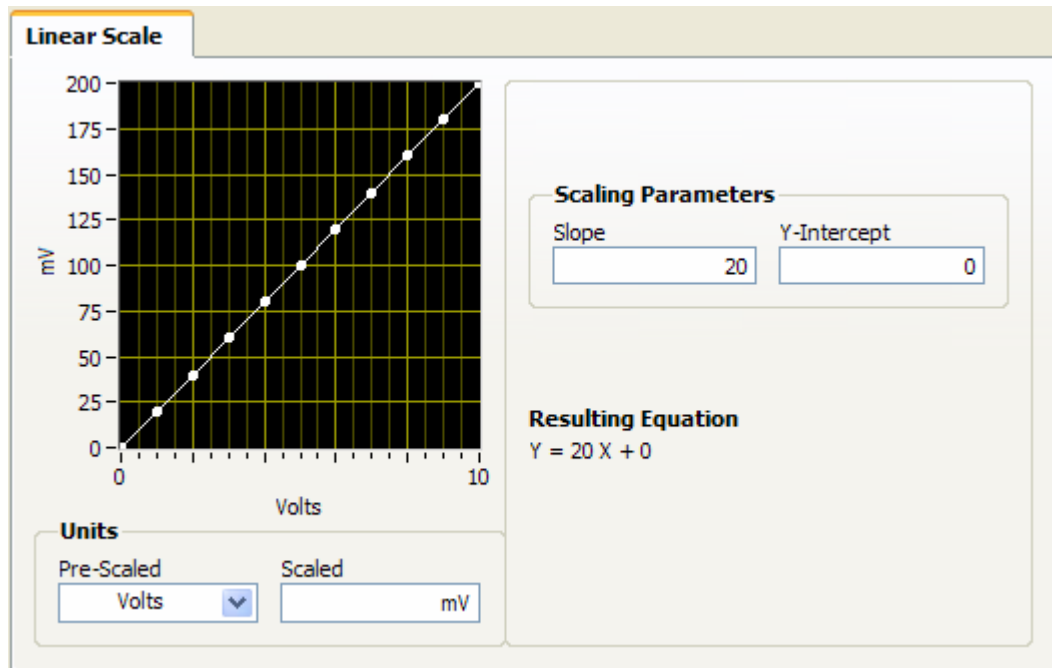
For the Axopatch 200B three tasks are needed:

- a voltage command task for voltage clamp,
- a current command task for $\beta = 1$ and
- a current command task for $\beta = 0.1$

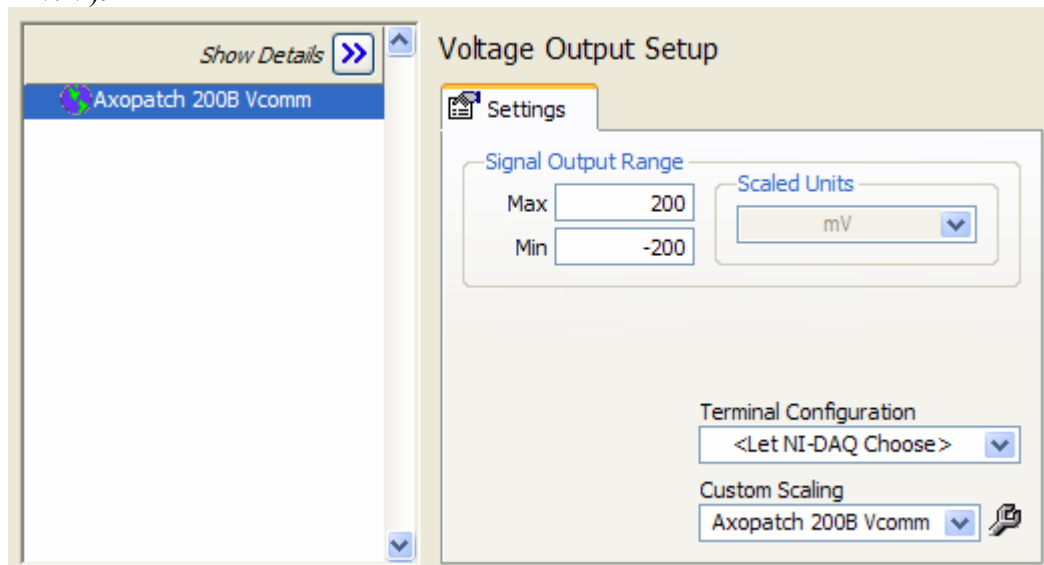
Similar to the telegraph readings these tasks will be set up using custom scales and virtual channels, thus shifting the scaling of the command signal from the dynamic clamp feedback loop to the hardware driver software.

For the voltage command task:

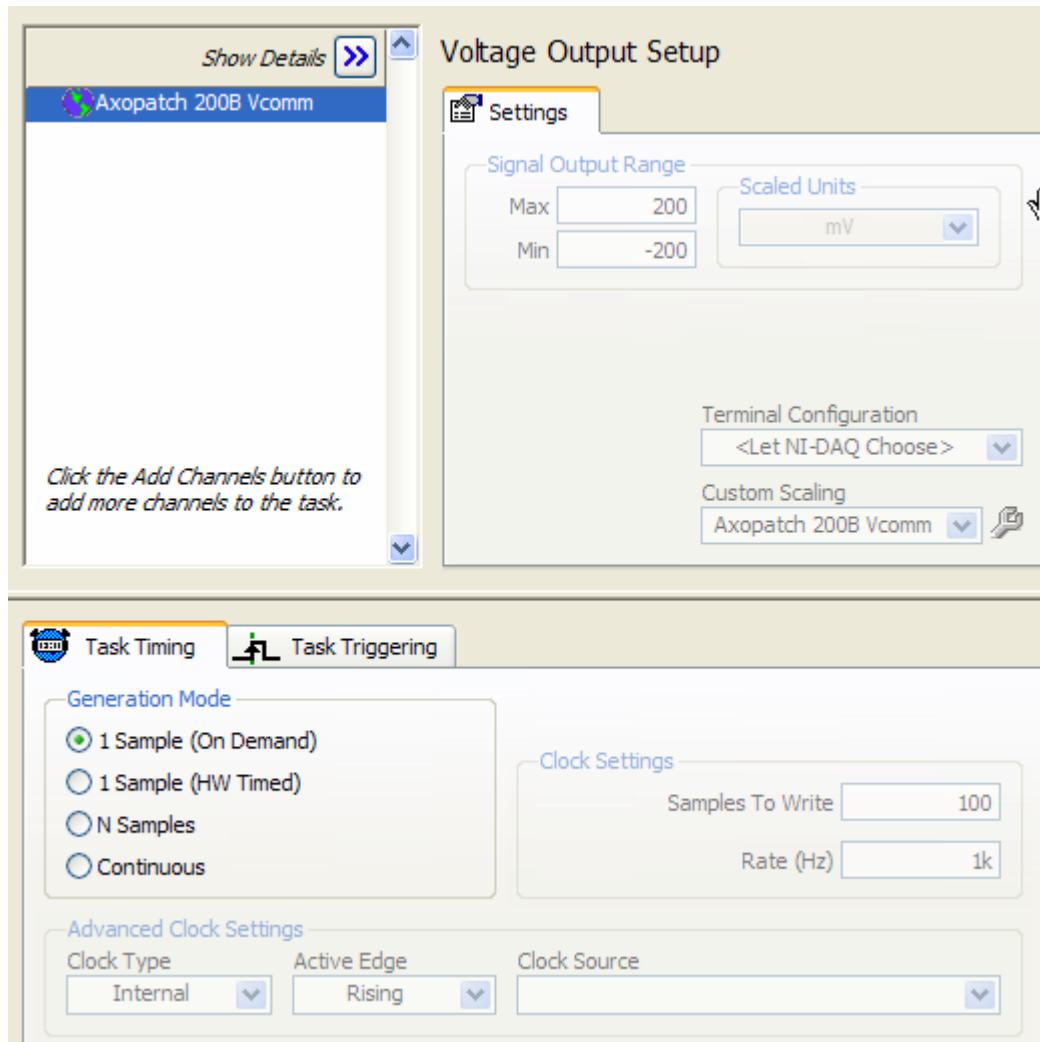
1. Create a custom linear scale 'Axopatch 200B Vcomm'. With a scale factor of 20 mV/mV the *Resulting Equation* should read: $Y = 20 X + 0$.



2. Create a new virtual channel 'Axopatch 200B Vcomm' for *Analog Output* > *Voltage* with *Physical* channel ao0. Select the scale defined in the previous step in *Custom Scaling* and set the *Signal Output Range* to -200 to 200 mV (set by the ± 10 V working range of the digital-to-analog converter and the scale factor 20 mV/V).



3. Create a new task 'Axopatch 200B AO Vcomm' using the virtual channel defined in step 2. In *Task Timing* set *Generation Mode* to 1 Sample (On Demand).



For the current command task with $\beta = 1$:

4. Create a custom linear scale 'Axopatch 200B Icomm beta1'. With a scale factor of $2 / \beta \text{ nA/V} = 2 / 1 \text{ nA/V} = 2 \text{ nA/V} = 2000 \text{ pA/V}$ the *Resulting Equation* should read: $Y = 2000 X + 0$.
5. Create a new virtual channel 'Axopatch 200B Icomm beta1' for *Analog Output > Voltage* with *Physical* channel ao0. Select the scale defined in the previous step in *Custom Scaling* and set the *Signal Output Range* to -20000 to 20000 pA (set by the $\pm 10 \text{ V}$ working range of the digital-to-analog converter and the scale factor 2000 pA/V).
6. Create a new task 'Axopatch 200B AO Icomm beta1' using the virtual channel defined in step 2. In *Task Timing* set *Generation Mode* to 1 Sample (On Demand).

For the current command task with $\beta = 0.1$:

7. Create a custom linear scale 'Axopatch 200B Icomm beta0_1'. With a scale factor of $2 / \beta \text{ nA/V} = 2 / 0.1 \text{ nA/V} = 20 \text{ nA/V} = 20000 \text{ pA/V}$ the *Resulting Equation* should read: $Y = 20000 X + 0$.

8. Create a new virtual channel 'Axopatch 200B Icomm beta0_1' for *Analog Output* > *Voltage* with *Physical* channel ao0. Select the scale defined in the previous step in *Custom Scaling* and set the *Signal Output Range* to -200000 to 200000 pA (set by the ± 10 V working range of the digital-to-analog converter and the scale factor 20000 pA/V).
9. Create a new task 'Axopatch 200B AO Icomm beta0_1' using the virtual channel defined in step 2. In *Task Timing* set *Generation Mode* to 1 Sample (On Demand).

After all tasks have been defined and saved to the hardware configuration of the embedded controller it is strongly recommended to save a copy of this configuration file on the host computer. This can be done using *File > Export...* in MAX.

2

Amplifier VIs

The two amplifier VIs (one executing on the embedded controller, the other executing on the host computer with its front panel integrated into the main G-clamp front panel) are designed as plug-in components of G-clamp. Thus new VIs simply have to be placed into specific directories to be usable by G-clamp, provided that some features characteristic for these amplifier VIs are maintained. It is therefore suggested to use an existing amplifier VI as a template instead of starting with a blank new VI.

Amplifier VI executing on the host computer

The amplifier VI executing on the host computer serves two purposes:

- It contains representations of those amplifier-specific controls that change the scaling of the V_M or I_M signal or the amplifier command and which are not telegraphed from the amplifier to the computer. Thus any change in the setting of one of these controls on the amplifier has to be repeated with the corresponding control on the amplifier VI front panel.
- For all amplifier-specific controls that change the scaling of the V_M or I_M signal or the amplifier command and which are telegraphed from the amplifier to the computer the front panel of the amplifier VI contains corresponding representations. These indicators are automatically updated whenever an experiment is started and thus serve as a visual reminder of the current amplifier settings on the front panel of G-clamp.

For the Axopatch 200B this means that the front panel will contain:

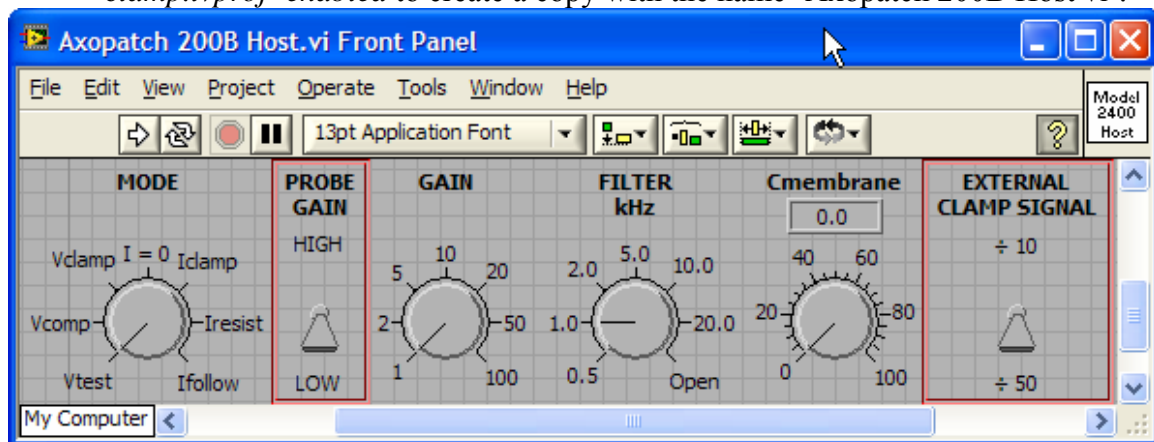
- a control CONFIG which determines the feedback resistor in use ($\beta = 1$ or $\beta = 0.1$)

² A copy of our configuration file (G-clamp v2.nce) is included in the G-clamp download as part of G-clamp.lvproj.

- a control I OUTPUT which determines the sensitivity of the I_M output (100 β mV/pA or β mV/pA; this switch is located in the rear of the Axopatch 200B)
- an indicator WHOLE CELL CAP.
- an indicator LOWPASS BESSEL FILTER
- an indicator OUTPUT GAIN
- an indicator MODE

This configuration resembles very much the configuration of the amplifier VI for the Model 2400 amplifier. Therefore the Model 2400 amplifier VI will be used as a template. This maintains the amplifier VI-specific input/output characteristics and also allows to re-use – with modifications – the controls and indicators instead of creating them from scratch.

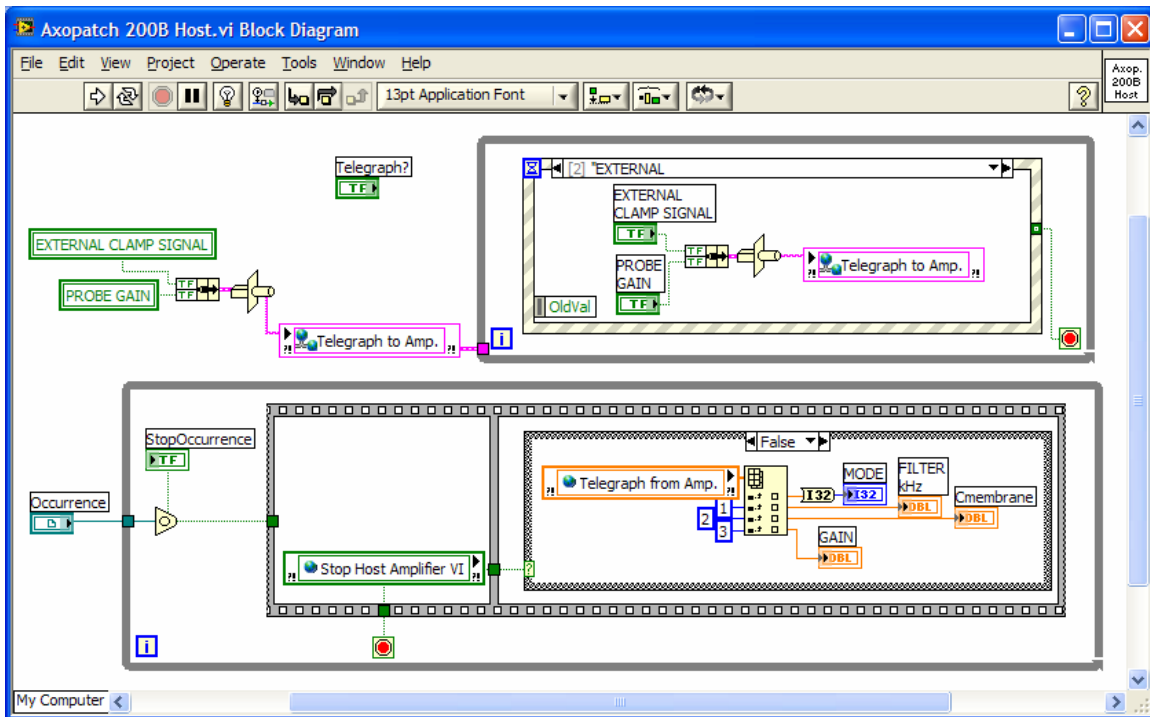
1. Start LabVIEW and load 'G-clamp.lvproj'.
2. Under *My Computer > Host > Amplifiers* open 'Model 2400 Host.vi'.
3. Use *File > Save As... > Open additional copy* with the option *Add copy to G-clamp.lvproj* enabled to create a copy with the name 'Axopatch 200B Host vi'.



4. Close 'Model 2400 Host.vi' and in the Project Explorer window move 'Axopatch 200B Host vi' into the folder *My Computer > Host > Amplifiers*.
5. Right-click on the VI icon and select *Edit Icon...* to change the VI icon.



6. Select *Window > Show Block Diagram* to display the wiring diagram.



The two functions of the VI – transmitting the setting of non-telegraphed amplifier controls to the amplifier VI on the embedded controller and displaying the setting of telegraphed amplifier controls – are done in two separate *For Loops*. The first task is done in the upper *For Loop*. It contains an *Event Structure* and thus sits idle until one of the events defined for the structure actually occurs. The specific case of the Event Structure shown above is set to execute whenever one of the two controls EXTERNAL CLAMP SIGNAL and PROBE GAIN on the front panel is changed by the user. In that case both controls are read out, the information bundled into a cluster, the cluster formatted into a string which is then used to update the network-shared variable Telegraph to Amp. (Telegraph to Amp. is read out by the amplifier VI on the embedded controller when an experiment is started) Notice that EXTERNAL CLAMP SIGNAL and PROBE GAIN are also read out via local variables to initialize the network-shared variable Telegraph to Amp. when the VI is started and before the *For Loop* with the *Event Structure* starts executing.

The second task is done in the lower *For Loop*. It contains a *Wait on Occurrence* function which lets the loop sit idle until the event referenced by the control OCCURRENCE actually occurs. This event can be set by several processes in the main G-clamp VI and therefore can have multiple meanings. For the purpose of the amplifier VI any event different from stopping G-clamp or changing the amplifier module by the user (resulting in setting the shared variable Stop Host Amplifier VI to TRUE by the main G-clamp VI) results in reading the shared variable Telegraph from Amp. and updating the indicators MODE, FILTER kHz, Cmembrane and GAIN on the amplifier VI front panel. (Telegraph from Amp. is set by the main G-clamp VI after the amplifier VI on the embedded controller executed and transmitted this information to the host computer)

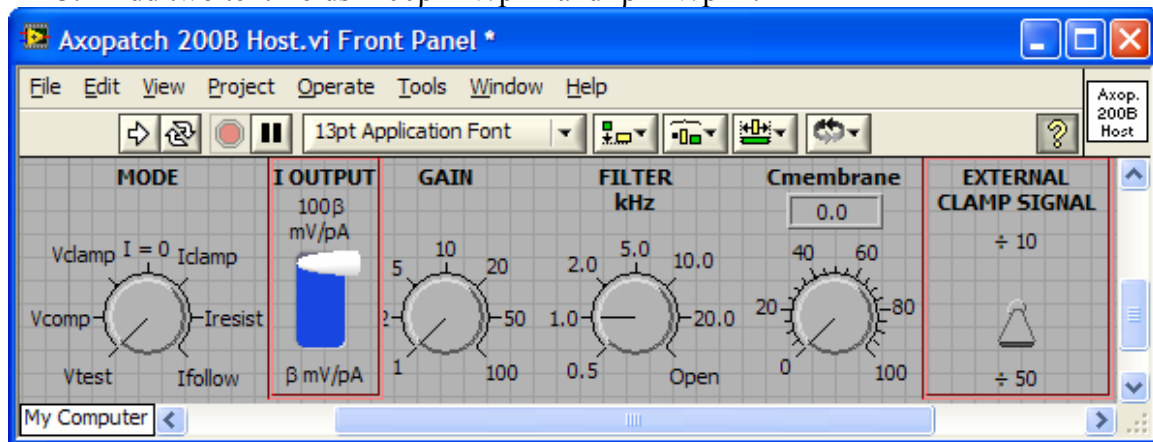
(The purpose of the control Telegraph? is not quite clear to me right now. However, it is used by referencing it in the main G-clamp VI and therefore must have some function. Don't delete!)

For the Axopatch 200B this wiring scheme can be completely re-used, requiring only cosmetic changes to make the controls and indicators look like their real Axopatch 200B counterparts.

I OUTPUT

For this control the PROBE GAIN control will be modified:

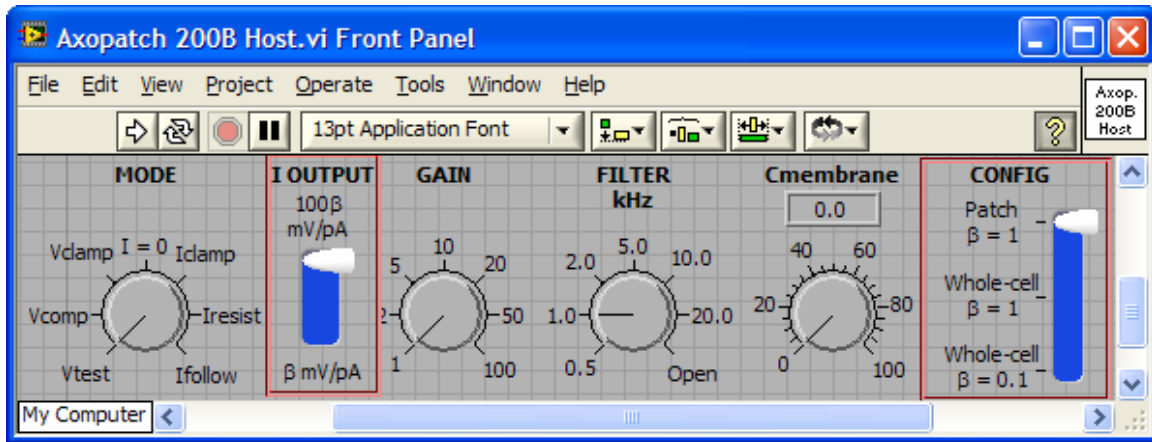
1. Right-click the control and select *Replace* to change this Boolean control into a numeric control, e.g. Vertical Pointer Slide, and rename the control I OUTPUT. Note that on the block diagram the local variable automatically adapted to this change.
2. Right-click the control and select *Properties*.
 - a. In the *Scale* tab change the *Scale Range Minimum* to 1 and the *Maximum* to 100. Also change the *Scale Style* to no-ticks.
 - b. In the *Data Range* tab de-activate *Use Default Range*. Set *Representation* to I16, *Minimum* to 1, *Maximum* to 100 and *Increment* to 99. Thus the control can have only two values, 1 and 100.
3. Add two text fields '100 β mV/pA' and ' β mV/pA'.



CONFIG

For this control the EXTERNAL CLAMP SIGNAL control will be modified:

1. Right-click the control and select *Replace* to change this Boolean control into a numeric control, e.g. Vertical Pointer Slide, and rename the control CONFIG.
2. Right-click the control and select *Properties*.
 - a. In the *Scale* tab change the *Scale Range Minimum* to 1 and the *Maximum* to 2. Also change the *Scale Style* to no-minor-ticks and no-numbers.
 - b. In the *Data Range* tab set *Representation* to I16.
3. Add three text fields 'Patch $\beta=1$ ', 'Whole-cell $\beta=1$ ' and 'Whole-cell $\beta=0.1$ ' for the numeric values 2, 1 and 0.



WHOLE CELL CAP., LOWPASS BESSEL FILTER, OUTPUT GAIN, MODE

Amplifier-specific indicators are saved as custom controls in G-clamp.lvproj. Thus before modifying the indicators of the Model 2400 amplifier module, copies of these controls should be made and named for the Axopatch 200B amplifier in a way identical to creating Axopatch 200B Host.VI from Model 2400 Host.VI (steps 3 and 4 in the section 'Amplifier VI executing on the host computer' above). Next all Model 2400 indicators on the front panel of Axopatch 200B Host.VI have to be replaced with the newly created indicators: Right-click each indicator, select *Replace > Select a control...*

Modifying these indicators to more closely resemble the real Axopatch 200B controls is now done by modifying the custom control: Right-click an indicator on the front-panel of Axopatch 200B Host.VI and select *Open Type Def..*

Mode

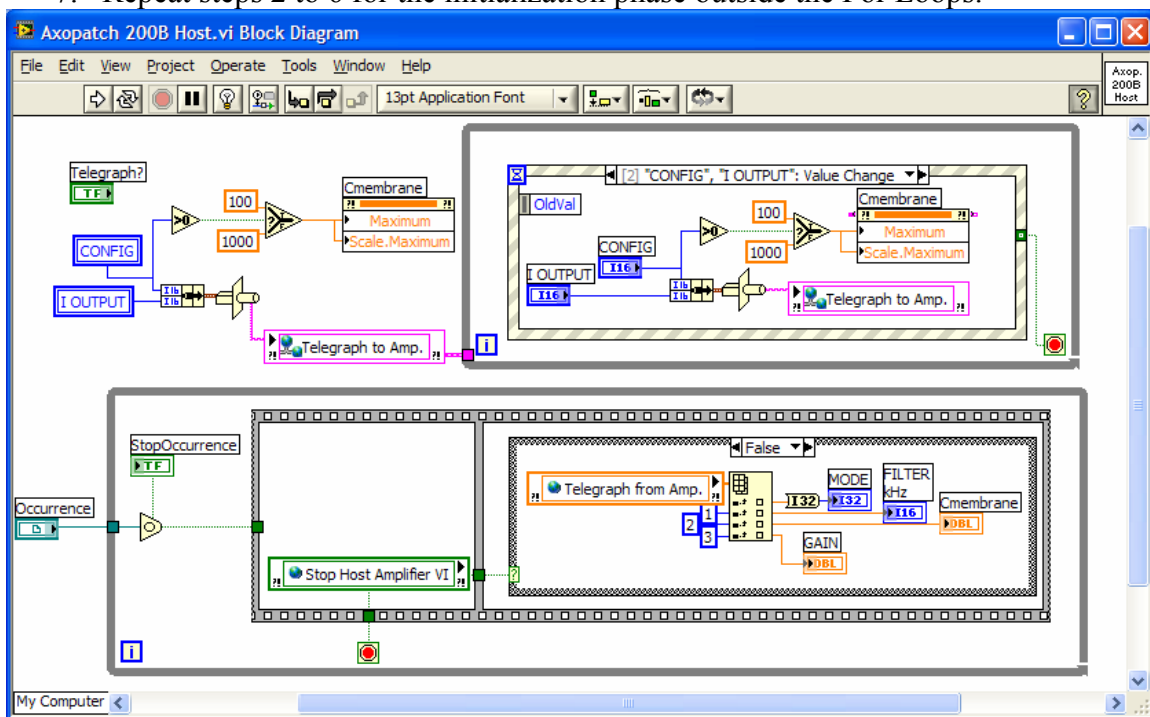
1. Right-click the control and select *Properties*.
 - a. In the *Scale* and in the *Data Range* tabs set *Minimum* and *Maximum* to 0 and 4.
 - b. In the *Text Labels* tab de-activate *Sequential values* and delete two rows of values and text labels. Label value 1 'Track', value 2 'Vclamp', value 3 'I = 0', value 4 'Idamp normal' and value 5 'Idamp fast'. (The values in this list have to correspond to the scaled values that were defined in MAX for the NI-DAQmx Scale 'Axopatch 200B Mode'!!!)
2. Save the control (Axopatch 200B MODE.ctl).

Whole cell cap.

This indicator does not need any modifications (except maybe its label). For $\beta = 1$ the indicator has the same range (0 to 100 pF) as in the Model 2400 amplifier. To adjust the indicator to the larger range (0 to 1000 pF) for $\beta = 0.1$ will be done by changing the block

diagram of Axopatch 200B Host.VI so that it changes the control properties *Data Range* and *Scale Range* programmatically depending on the setting for CONFIG:

1. In order to be able not only to read this property during run-time but to set it the indicator has to be disconnected from its type defining control. On the front panel right-click on the Cmembrane indicator and select *Disconnect from Type Def.*
2. In the block diagram of Axopatch 200B Host.VI right-click on the terminal for the Cmembrane indicator and select *Create > Property Node > Data Range > Maximum*. Place the property node in the case of the Event Structure that updates the network shared variable Telegraph to Amp. upon changes in CONFIG or I OUTPUT.
3. In the block diagram right-click the property node created in step 1 and select *Change To Write*.
4. Expand the node to show another property. While pointing on this additional property right-click and select *Properties > Scale > Range > Maximum*.
5. Create two numerical DBL constants of value 100 and 1000.
6. Wire a *Greater than 0?* comparison function to the CONFIG terminal, followed by a *Select* comparison function and wire the output of *Select* to the two properties.
7. Repeat steps 2 to 6 for the initialization phase outside the For Loops.



Lowpass Bessel filter

1. Change the label of the control to 'LOWPASS BESSEL FILTER'.
2. Right-click the control and select *Properties*.
 - a. In the *Data Range* tab activate *Use Default Range*.
 - b. In the *Scale* tab set *Minimum* and *Maximum* to 1 and 100.

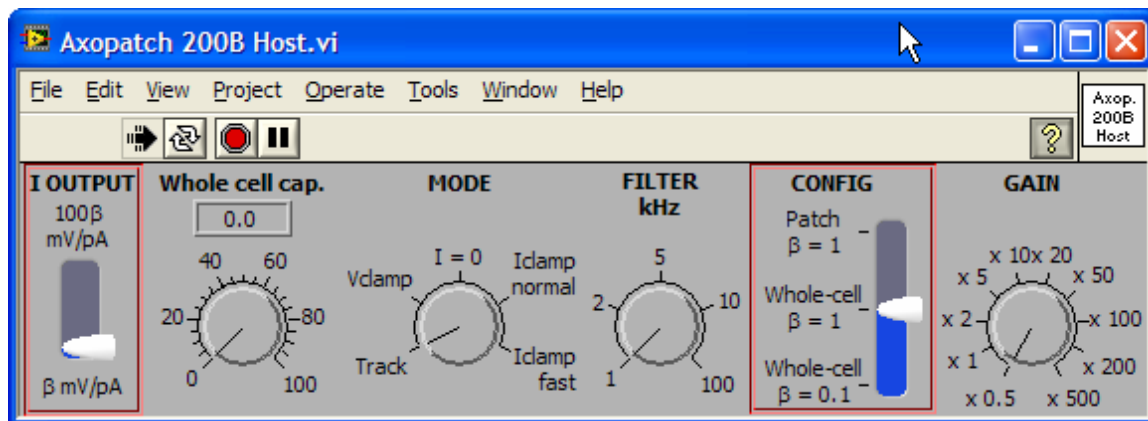
- c. In the *Text Labels* tab de-activate *Sequential values*. Label value 1 '1', value 2 '2', value 5 '5', value 10 '10' and value 100 '100'. (The values in this list have to correspond to the scaled values that were defined in MAX for the NI-DAQmx Scale 'Axopatch 200B FilterFreq'!!!)
3. Save the control (Axopatch 200B FILTER kHz.ctl).

Gain

4. Change the label of the control to 'OUTPUT GAIN (α)'.
5. Right-click the control and select *Properties*.
 - a. In the *Data Range* tab activate *Use Default Range*.
 - b. In the *Scale* tab set *Minimum* and *Maximum* to 0.05 and 500.
 - c. In the *Text Labels* tab de-activate *Sequential values*. Label value 0.05 'x 0.05', value 0.1 'x 0.1', value 0.2 'x 0.2', value 0.5 'x 0.5', value 1 'x 1', value 2 'x 2', value 5 'x 5', value 10 'x 10', value 20 'x 20', value 50 'x 50', value 100 'x 100', value 200 'x 200' and value 500 'x 500'. (For the Gain two NI-DAQmx Scales were defined ('Axopatch 200B VGain in Iclamp beta0_1' and 'Axopatch 200B VGain in Iclamp beta1 and IGain in Vclamp') in which the scaled values correspond to the actual scale factors for the corresponding signal. The amplifier VI executing on the embedded controller will have to match these scaled values with the values set up here for the Gain control.)
6. Save the control (Axopatch 200B GAIN.ctl).

If any of the changed indicators appear grayed-out on the front panel of Axopatch 200B Host.VI right-click the indicator and select *Update from Type Def.* to re-link the indicator to the updated control.

Finally, move the controls/indicators on the front panel of Axopatch 200B Host.VI so that their arrangement resembles somewhat their arrangement on the actual amplifier:



Amplifier VI executing on the embedded controller

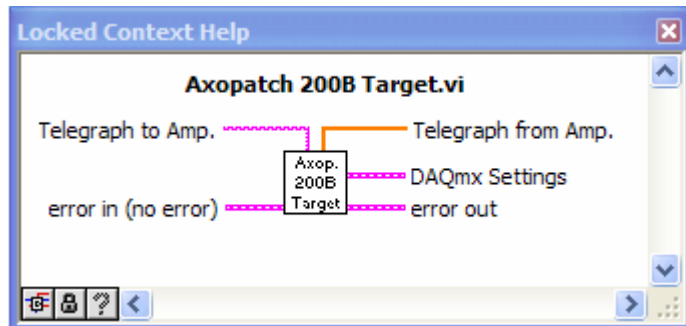
Development of the amplifier VI executing on the embedded controller requires access to the tasks that had been defined for the various analog-in and analog-out operations. As these tasks are stored in the hardware configuration file of the embedded controller it is necessary that the host computer is connected to the running embedded controller.

Again because of the similarities between the Axopatch 200B and the Model 2400 controls we will use the Model 2400 amplifier VI as template for the development of the Axopatch 200B VI.

1. Start LabVIEW and load 'G-clamp.lvproj'.
2. Under *My Computer > Target > Amplifiers* open 'Model 2400 Target.vi'.
3. Use *File > Save As... > Open additional copy* with the option *Add copy to G-clamp.lvproj* enabled to create a copy with the name 'Axopatch 200B Target vi'.
4. Close 'Model 2400 Target.vi' and 'Axopatch 200B Target vi' and in the Project Explorer window move 'Axopatch 200B Target vi' from *My Computer* to your embedded controller.
5. Open 'Axopatch 200B Target vi' from its location under your embedded controller. With the controller running access to all tasks defined on the embedded controller is possible.
6. Right-click on the VI icon and select *Edit Icon...* to change the VI icon.



Notice that the amplifier VI executing on the embedded controller has a number of defined inputs and outputs:



These should neither be deleted nor modified nor should their placement in the connector pane be changed!

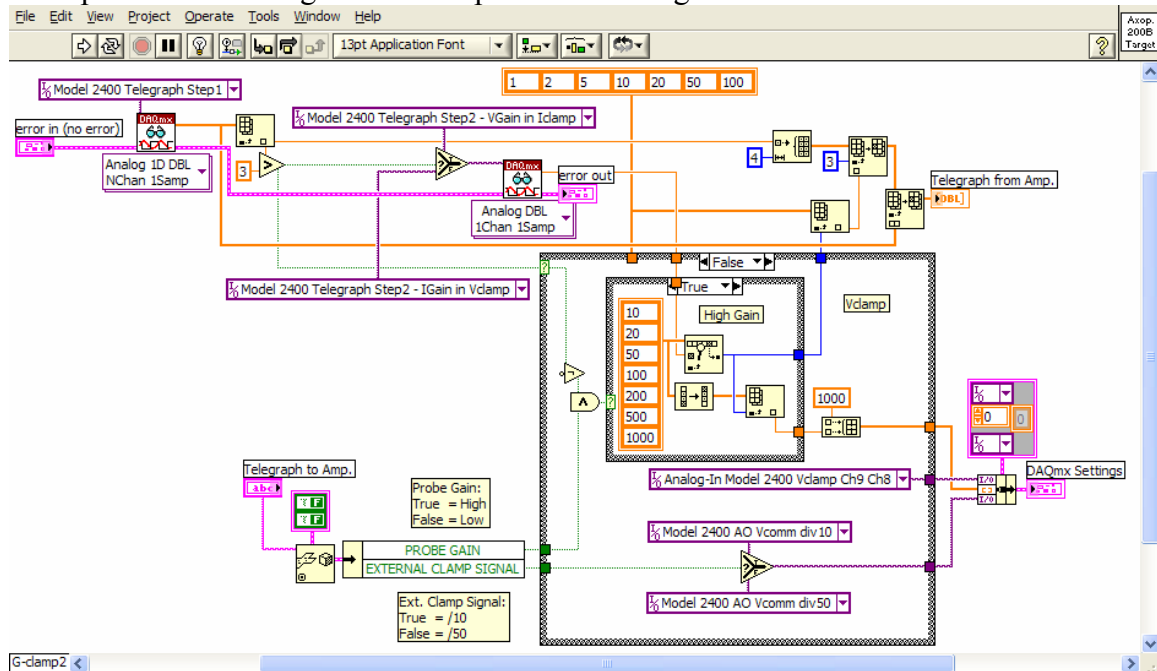
Telegraph to Amp. is a string provided by the main G-clamp VI on the embedded computer after it received this string upon experiment start from the main VI executing on the host computer containing information about amplifier settings not provided by amplifier telegraph outputs required to choose the correct analog-in and analog-out tasks.

Likewise, Telegraph from Amp. is a 1-D array of DBLs containing information from amplifier telegraph outputs. This information is read out by the main VI on the embedded controller, transmitted to the main VI on the host computer which will provide it to the amplifier VI on the host computer via a shared variable.

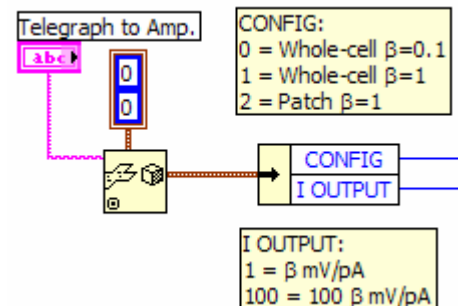
DAQmx Settings is a cluster containing an analog-in task, an analog-out task and a 1-D array of DBLs containing the input limits for the two channels in the

analog-in task. The input limits will be used by the experiment module VI to build the look-up tables which are used to derive the scaled mV and pA values from the unscaled raw voltage output of the analog input operation.

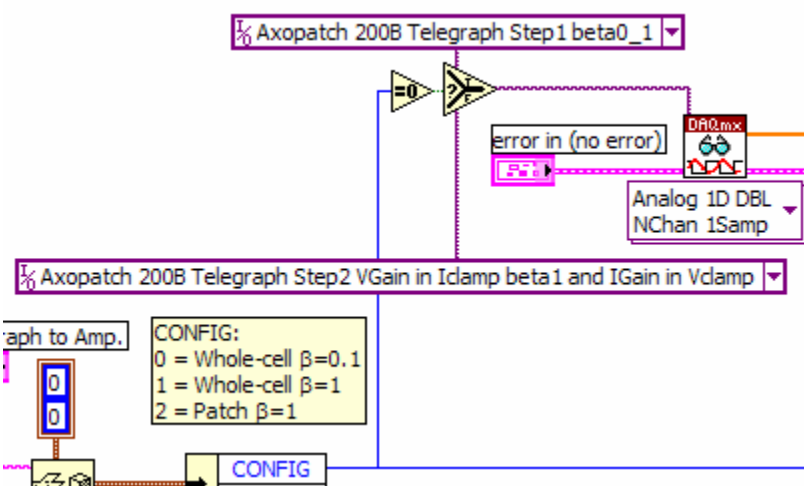
7. Open the block diagram of 'Axopatch 200B Target vi'.



- a. The 'Telegraph to Amp.' string for the Axopatch 200B was derived from a cluster built of the outputs of the CONFIG and I OUTPUT controls. To retrieve this information from the string, the *Unflatten From String* function has to be provided with the corresponding data type at its type input. Open the block diagram of 'Axopatch 200B Host.vi', right-click on the wire connecting the *Bundle* and the *Type Cast* function, select *Create > Constant* and copy the constant into 'Axopatch 200B Target.vi'. Connect to the type input of *Unflatten From String*. Successively right-click the items of the *Unbundle by Name* function and select *Select Item* to retrieve CONFIG and I OUTPUT.

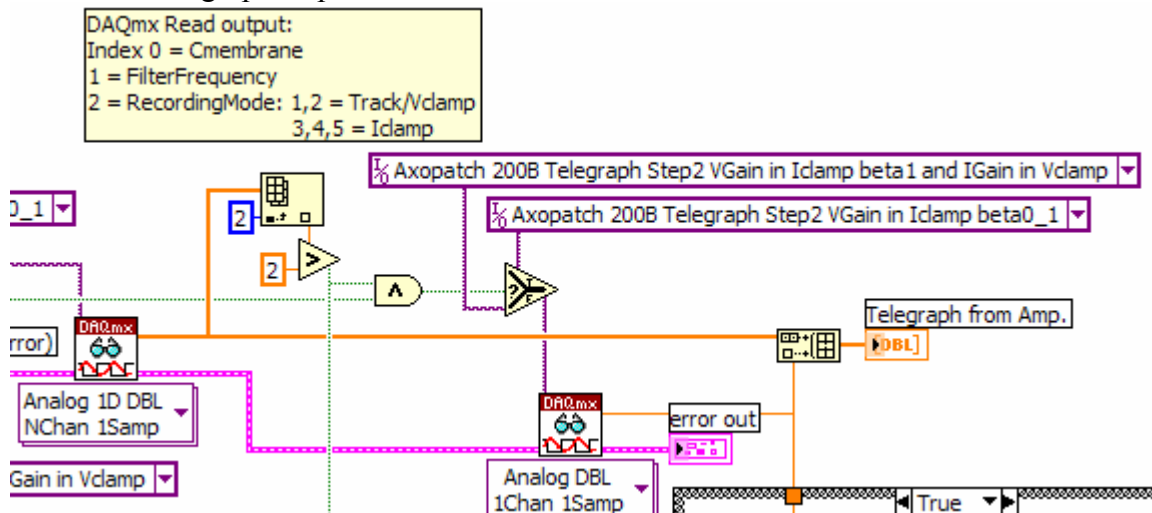


- b. For the Axopatch 200B, knowing β is a prerequisite

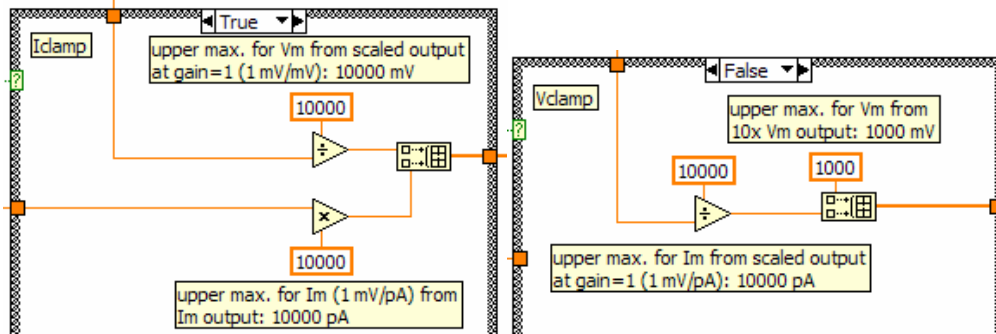


- for selection of the first telegraph task, either 'Axopatch 200B VGain in Idamp beta0_1' or 'Axopatch 200B VGain in Idamp beta1 and IGain in Vclamp'.

- c. The output of DAQmx Read is a 1-D array of DBLs containing three elements. The third element is the scaled output of the Mode telegraph. Any value larger than 2 indicates a current clamp mode. Together with the setting for β this information is then used to select one of the two possible telegraph step2 tasks.

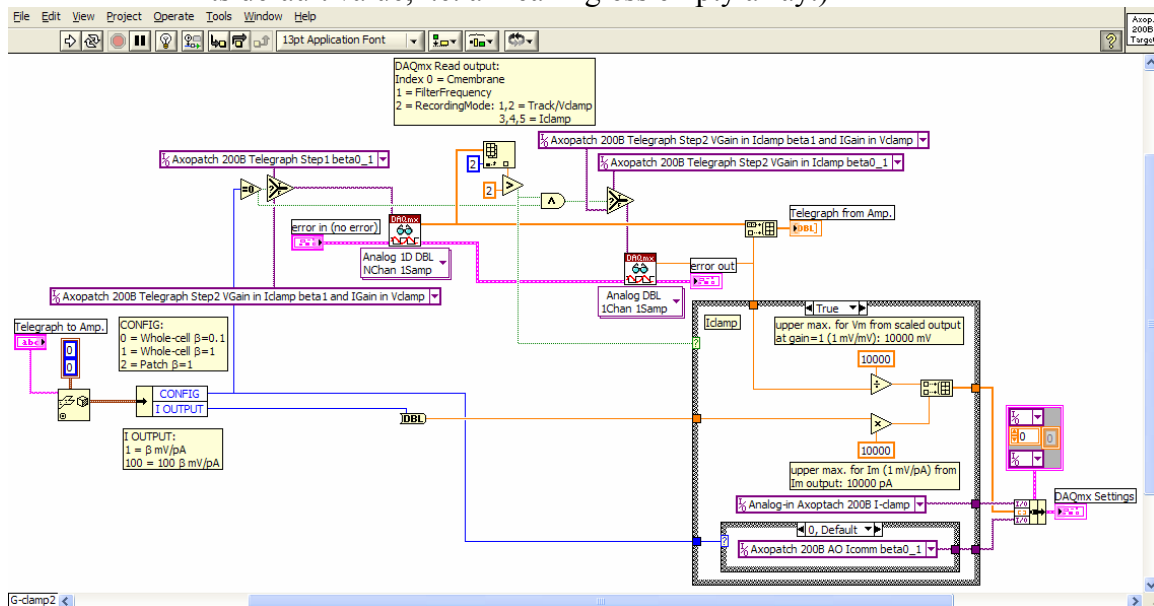


- d. The scaled output of the second telegraph step is the scale factor for the scaled amplifier output. By dividing the theoretically possible maximum value for V_M (current clamp) or I_M (voltage clamp) at gain = 1 by this scale factor the upper limit is derived. For each recording condition the upper limit of the other analog-in channel is based on the ± 10 V working range of the analog-to-digital converter and the scaling factors of the fixed amplifier outputs. (In case of the I_M output the setting of the sensitivity switch in the rear of the amplifier is also taken into account.)



The two upper limits for V_M and I_M are combined to a 1-D array and together with the analog input task and analog output task bundled into a cluster which represents the output of the VI (DAQmx Settings) that is then provided to the experiment module VI executing on the embedded controller.

The second output of the VI (Telegraph from Amp.) is a 1-D array combining the outputs of both telegraph tasks. (For an amplifier without telegraphing capabilities this output would simply be undefined and keep its default value, i.e. a meaningless empty array.)



- After saving 'Axopatch 200B Target.vi' move the VI in the Project Explorer Window from its location under your embedded controller to *My Computer > Target > Amplifiers*. Save the project.
- In the Project Explorer Window under *My Computer > Build Specifications* right-click 'Target Dynamic VIs' and select *Build* to create a new compilation of G-clamp dynamic VIs executing on the embedded controller that now includes the new amplifier VI.
- Copy the contents of the compilation folder (C:\ni-rt\startup\Dynamic VIs) into the corresponding folder on the embedded controller.
- In the Project Explorer Window under *My Computer > Host > G-clamp v2 Host.lvlib* double-click 'G-clamp v2 Host.cfg'. This is a configuration file

read at program start by the main G-clamp VI executing on the host computer. In the section [Available] add 'Axopatch 200B' to the amplifier section. Use exactly the same name that was given to the amplifier VIs (without ... Host.vi or ...Target.vi) and separate amplifiers with a semicolon only (no other characters or spaces).

- i. Run G-clamp, select the new amplifier and – using the model cell – test out whether all functions of the new amplifier module work and all the scales were set up correctly.