Setup Programming Environment Xcode, Eclipse & Pure Data

Version 0.1

Thomas Resch

thomas.resch@campus.tu-berlin.de

SETUP PROGRAMMING ENVIRONMENT

Eclipse Windows

Download: https://puredata.info/downloads
Install directly in the root directory (in C:\), not in Programs or Program Files
Download: https://github.com/pure-data/externals-howto to *workingDirectoryOfChoice*
Download Eclipse for C/C++ http://www.eclipse.org/
Download minGw-w64 https://sourceforge.net/projects/mingw-w64/
Install both in the proposed standard directories
Download the Pure Data sources https://github.com/pure-data/pure-data (for later)

In Eclipse go to File -> Import -> C/C++ -> Existing Code as Makefile Project
Select the downloaded example1 Folder in *workingDirectoryOfChoice/*externals-howto/
Choose GNU Autotools as Toolchain
Right Click on example1 in the Project Explorer
Select Properties in the Popup Menu
Go to C/C++ Build -> Environment and select the variable PATH or create it if it does not exist yet. Add the following two path-variables to PATH:
C:\Program Files (x86)\mingw-w64\i686-7.3.0-posix-dwarf-rt_v5-rev0;

C:\Program Files (x86)\mingw-w64\i686-7.3.0-posix-dwarf-rt_v5-rev0\mingw32\bin

(You can get them via clipboard by finding your mingw-w64 installation directory in the explorer. Right-click in the explorer text line on the top and select “Copy address as text”.)

In externals-howto-master/pd-lib-builder open the Makefile.pdlibbuilder with any Texteditor; with CTRL + f find the variables “pdincludepath” and “pdbinpath”; remove the “#” at the beginning and add the path to your Pd installation:

pdincludepath = C:\Pd\src
pdbinpath = C:\Pd\bin

(You can also drag the file into your eclipse project and edit it there..)

In the mingw-w64 installation folder, inside the **bin** directory, you should find a file called **mingw32-make.exe**. Copy this file in the same directory and change the name of the copy to **make.exe**.

Go to Project -> Build all

EDIT: Another guide for installing the IDE using VSCode is available on Github!

https://github.com/schlegelflegel/pd-vscode-wsl

Eclipse Linux

apt-get update
apt-get install puredata
apt-get git
apt-get gcc
apt-get install make
apt-get eclipse
apt-get ecplise-cdt

cd *workingDirectoryOfChoice*

git clone https://github.com/pure-data/externals-howto.git
chmod a+x=rwx example1

Start Eclipse, from the menu choose
New Project -> C/C++ -> Makefile Project with Existing Code
Choose Folder example1
Select Linux GCC for “Toolchain for Index Settings”
Click “Finish”

Or simply from the command line:

cd *workingDirectoryOfChoice/externals-howto/example1*
make

XCode OSX

Download/Install XCode https://developer.apple.com/xcode/
Download/Install Pure Data: https://puredata.info/downloads
Download puredata sources https://github.com/pure-data/pure-data (for later)
Open one of the provided example projects example1 or example 2
Click Command+b

Test in Pure Data

OSX:

- CTRL-Click on Product/helloworld from XCODE -> Show in finder
- CTRL-Click on helloworld -> Show original

  OR
- cd from command line to *workingDirectoryOfChoice*

- Copy helloworld.pd_darwin to Clipboard
- Go to Application/Pd-~~0.48-0~~ (latest release)
- Ctrl-click on Pd-~~0.48-0~~ (latest release)
- Select show Package Contents from Pop-up
- Go to Contents/Resources/extra
- Copy helloworld.pd_darwin from Clipboard to Contents/Resources/extra
- Start Pure Data
- Create a new Window (a patcher) with Command + n
- Create a new empty Object with Command + 1
- Type in your empty object box the object name (“helloworld” or “counter”)

Linux Command line:

- cd *workingDirectoryOfChoice/externals-howto/example1*
- puredata –lib helloworld
- Create a new Window (a patcher) with CTRL + n
- Create a new empty Object with CTRL + 1
- Type in your empty object box the object name (“helloworld” or “counter”)

Windows:

- copy helloworld.dll to C://pd/extra (should be in *workingDirectoryOfChoice/externals-howto/example1*)
- start puredata
- Create a new Window (a patcher) with CTRL + n
- Create a new empty Object with CTRL + 1
- Type in your empty object box the object name (“helloworld” or “counter”)

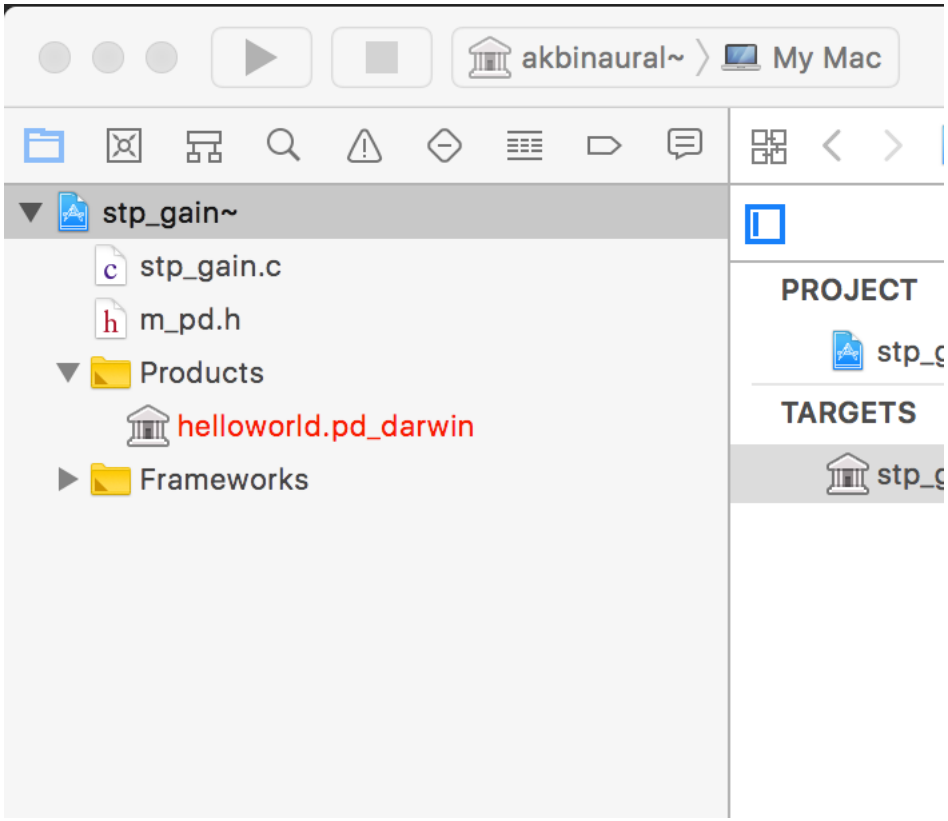
CREATING A NEW PROJECT

OSX:
Make a copy of the *stp_gain~* project folder (command+c; command+v)

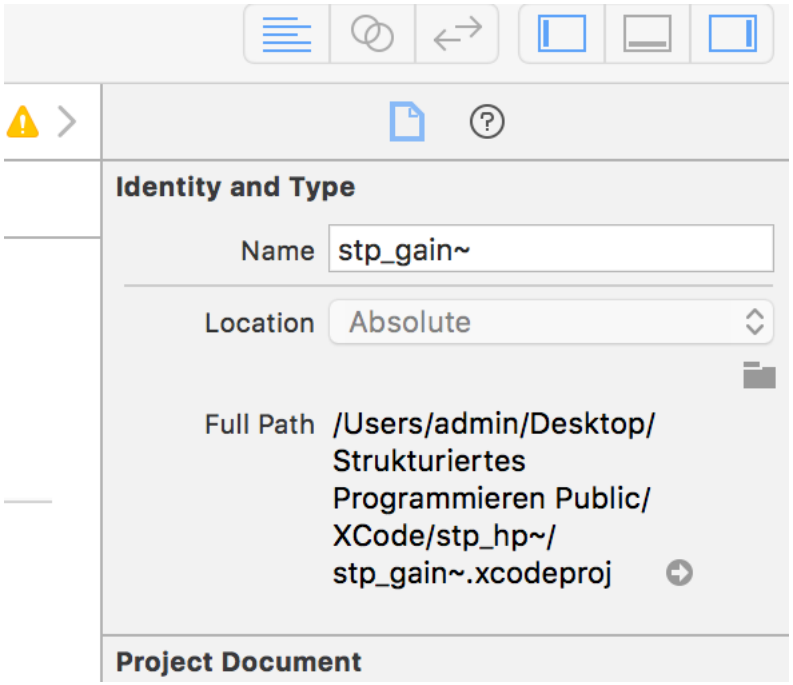
Rename it to whatever you are planning to implement.
But: A lot of objects already exists inside Pure Date. So always use an abbreviation in front of your project name/object name. For example, hp~ and lp~ are already taken, so do not name your lowpass project/object lp~, instead use your initials in the front (tr_lp~).

Inside your project folder open the xcode project (still named *stp_gain~)*

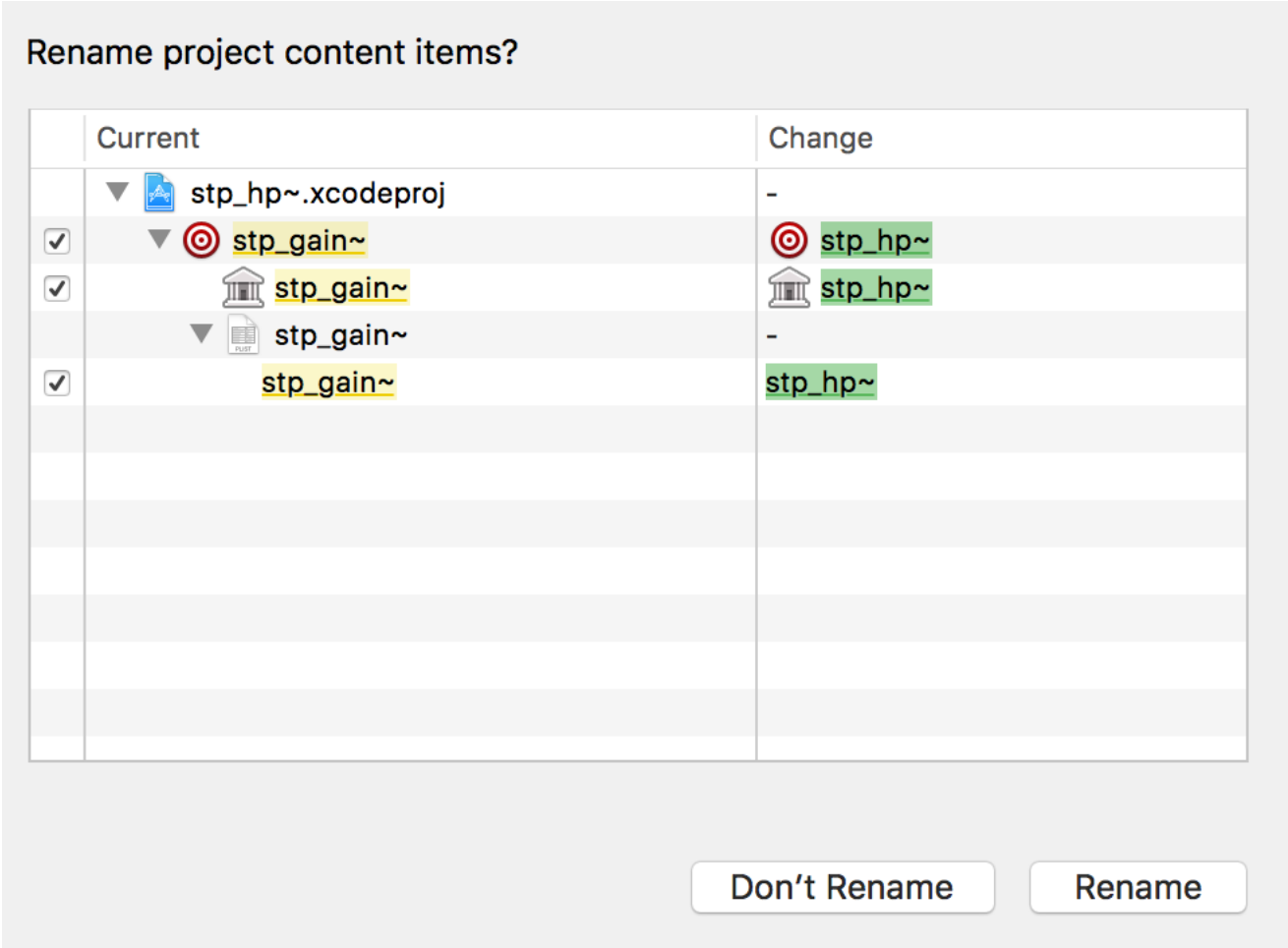
Select your project file in the upper left corner



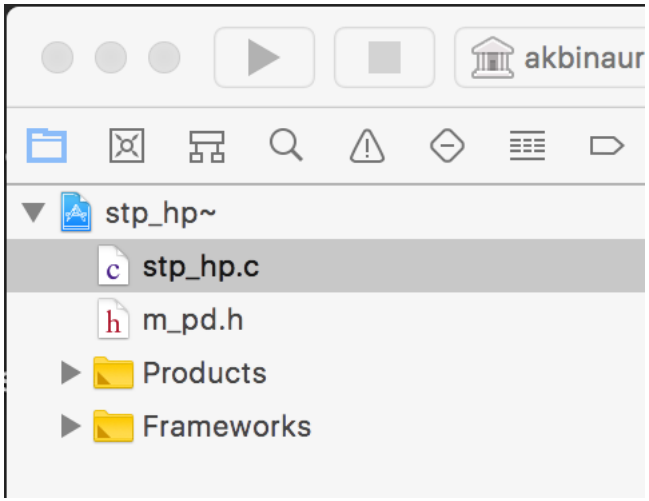
In the upper right corner, rename the project, for example from stp_gain~ to stp_hp~ if you want to implement a new highpass~ filter and hit return.



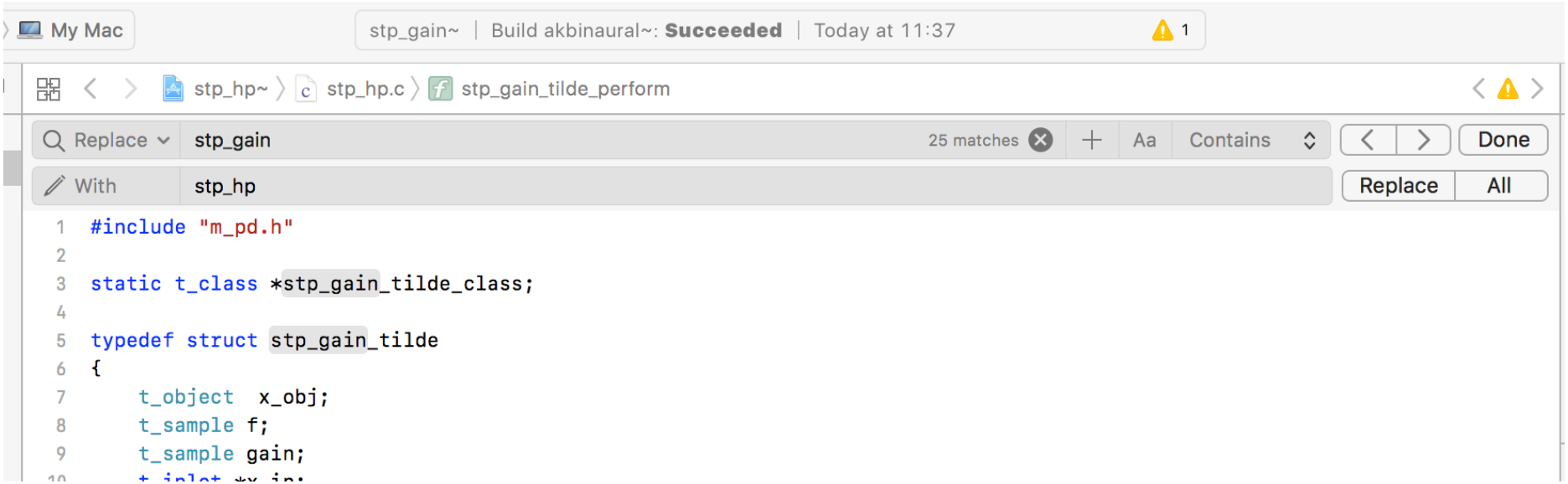
In the popup window select *rename:*



Now select your source file, still named stp_gain and rename it and press return.



Press Command+f for Find, select *Replace* instead of *Find* (click on the word *Find*), and enter stp_gain next to *Replace* and the new name next to *With;*
Then push the *All* Button on the right side.
Restart XCode, and Build your new object.



ECLIPSE (Windows & Linux)
(Under Linux skip modifying the PATH var)

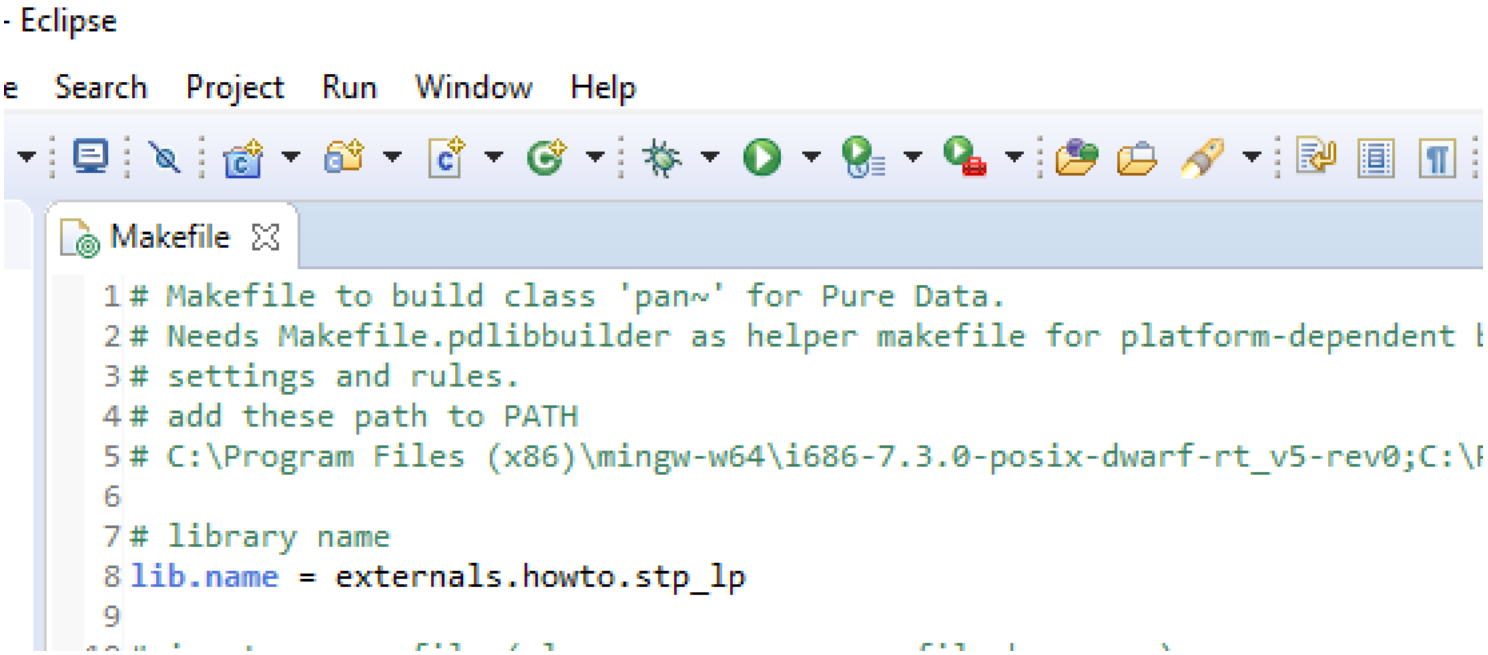
Copy the *stp_gain* Project to your already existing *externals-howto-master* Folder

Make a copy (CTRL+c CTRL+v) and rename it (for example to stp_hp)

Open Eclipse and go to the Menu File -> Import -> Existing Code as Makefile Project
Select GNU Autotools as Toolchain

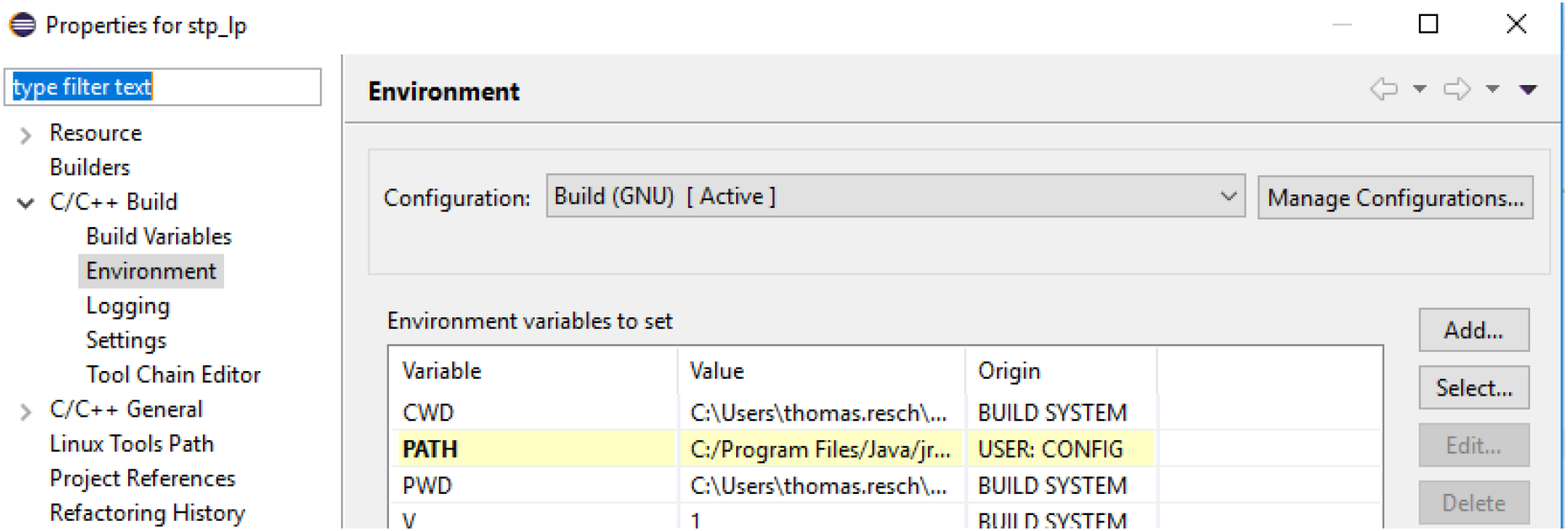
(As already described in the Beginning)

Double Click on the makefile inside the Project explorer and copy the paths in line 5 without the # to clipboard

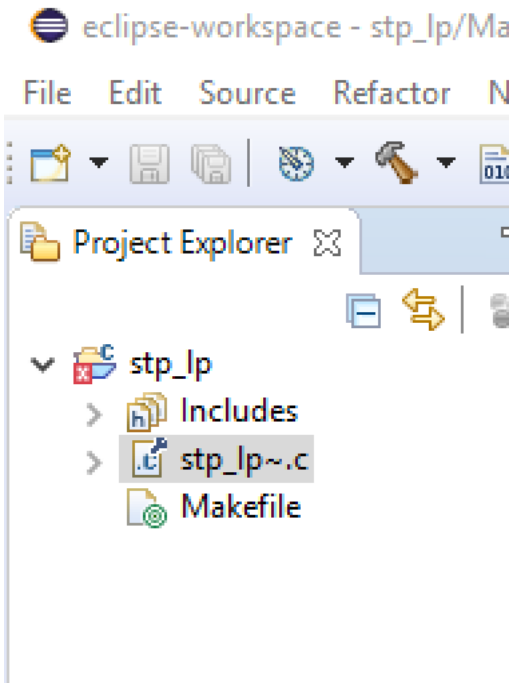


Right-Click on your Project Folder and select Properties in the popup menu (As described already in the beginning)

Select *Environment* and by clicking the *Add* Button create a new variable named PATH and copy the paths from clipboard into the *value* field



Right-click on the source file (stp_gain~.c) in the Project Explorer and select *Rename*
Rename it (for example to stp_lp~.c)



Double click on the makefile again and edit *lib.name* and *class.sources,* save the makefile, Eclipse does not autosave before building.

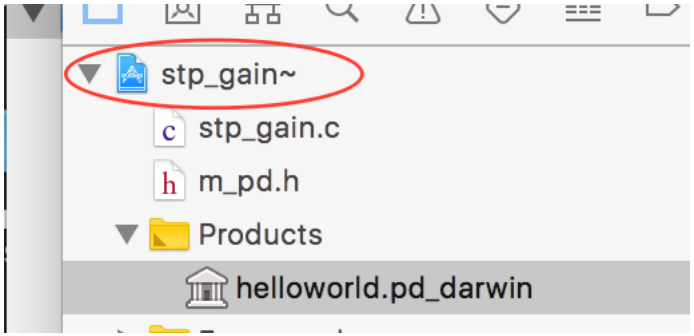
Makefile

```
# Makefile to build class 'pan~' for Pure Data.
# Needs Makefile.pdlibbuilder as helper makefile for platform-deper
# settings and rules.
# add these path to PATH
# C:\Program Files (x86)\mingw-w64\i686-7.3.0-posix-dwarf-rt_v5-rev

# library name
lib.name = externals.howto.stp_lp

# input source file (class name == source file basename)
class.sources = stp_lp~.c

# all extra files to be included in binary distribution of the libr
datafiles =

# include Makefile.pdlibbuilder from submodule directory 'pd-lib-bu
PDLIBBUILDER_DIR=../pd-lib-builder/
include $(PDLIBBUILDER_DIR)/Makefile.pdlibbuilder

```

ADDING SOURCE FILES TO YOUR PROJECT

XCode

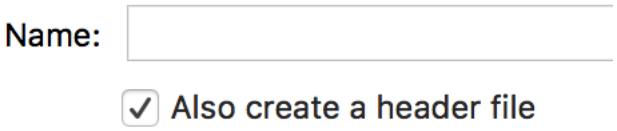
CTRL+Click on your xcode project



Select *New File* from the Pop-up menu

Choose *C-File* and click next

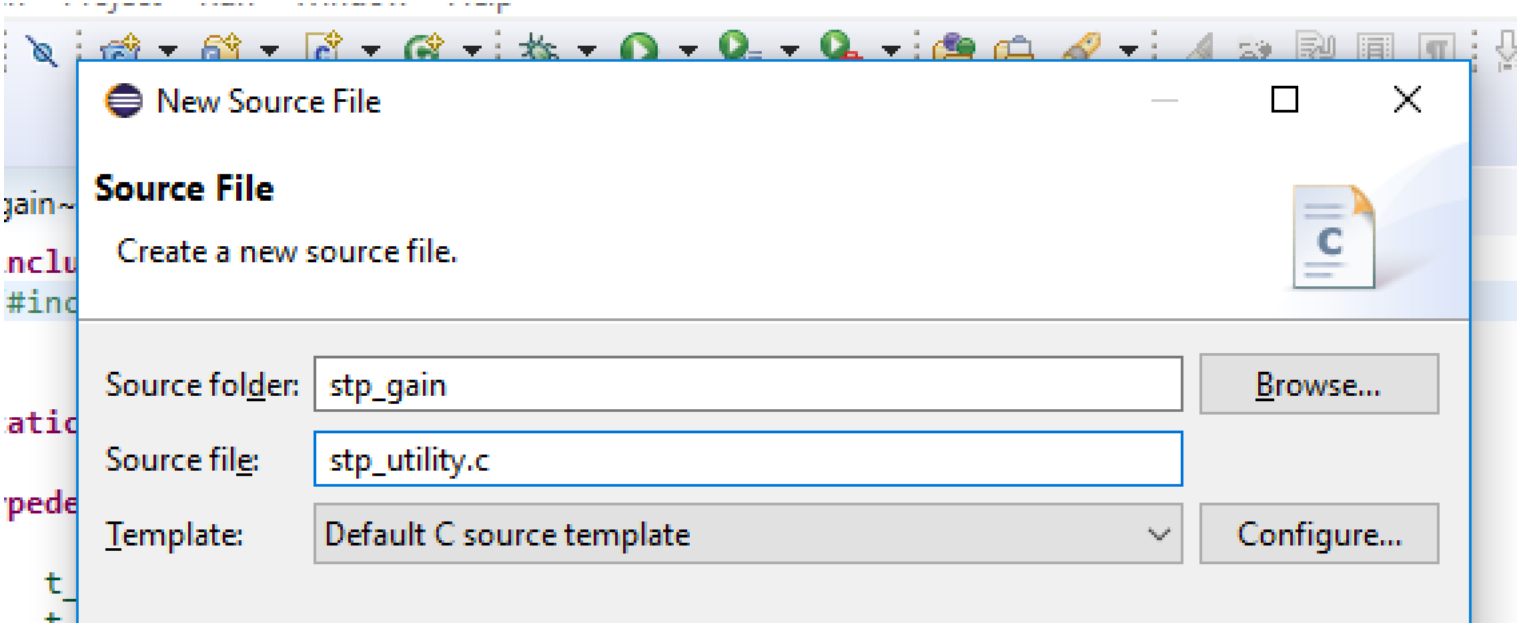
Name it and create a header file too



In the next dialogue click *create*

Eclipse

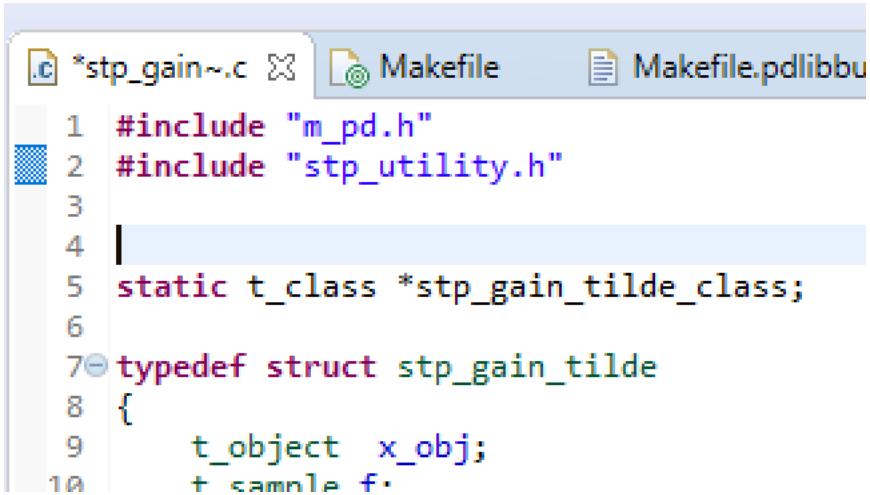
Right click on your project in the Project Explorer
In the Pop-up Menu select New -> Source File



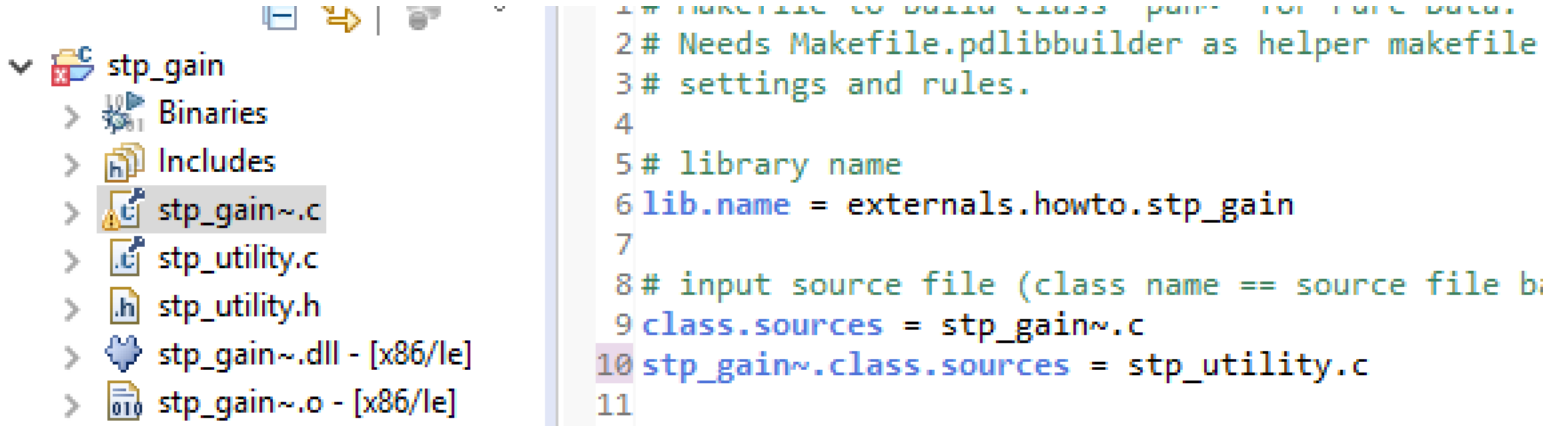
Enter a name including file extension *.c*
Select *Default C source template* and click finish

Repeat the process but instead of source choose *Header File*
Enter the same name, change the file extension to *.h*

In the C file stp_gain.c add the line *#include stp_utility.h*

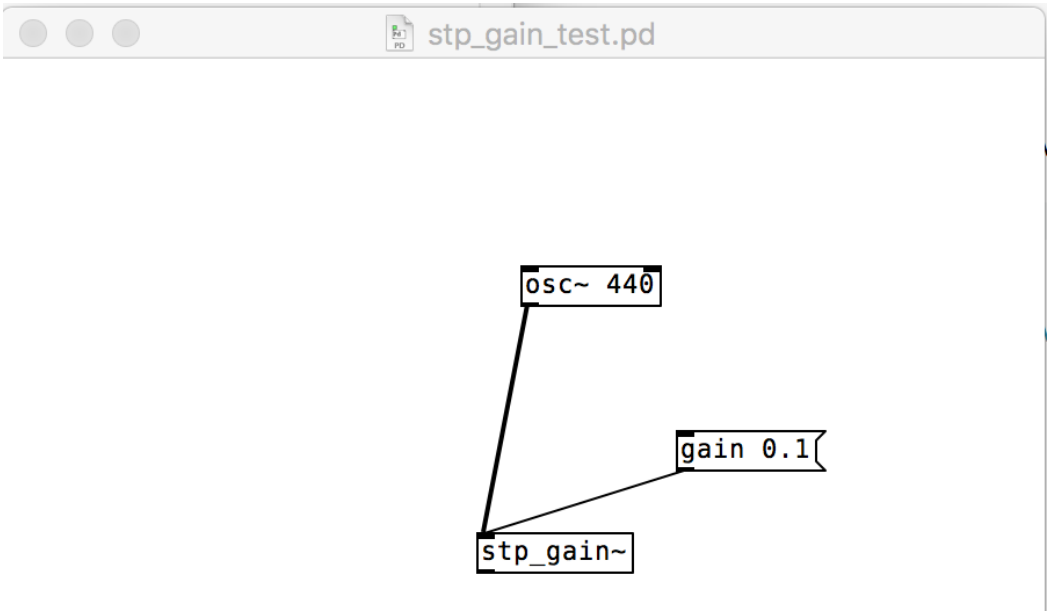


Add the line *stp_gain~.class.sources = stp_utility.c* to the makefile



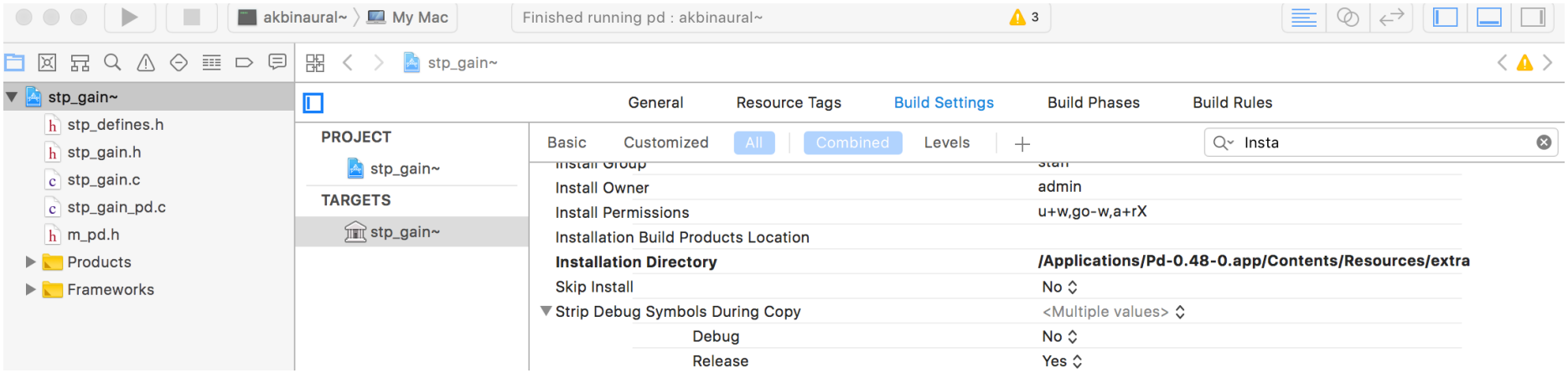
DEBUGGING PURE DATA OBJECTS

Create a debugging patch for your external in Pure Data which will be automatically loaded.



Xcode

For debugging (and for convenience) objects must be installed automatically into the extra folder of Pure Data:

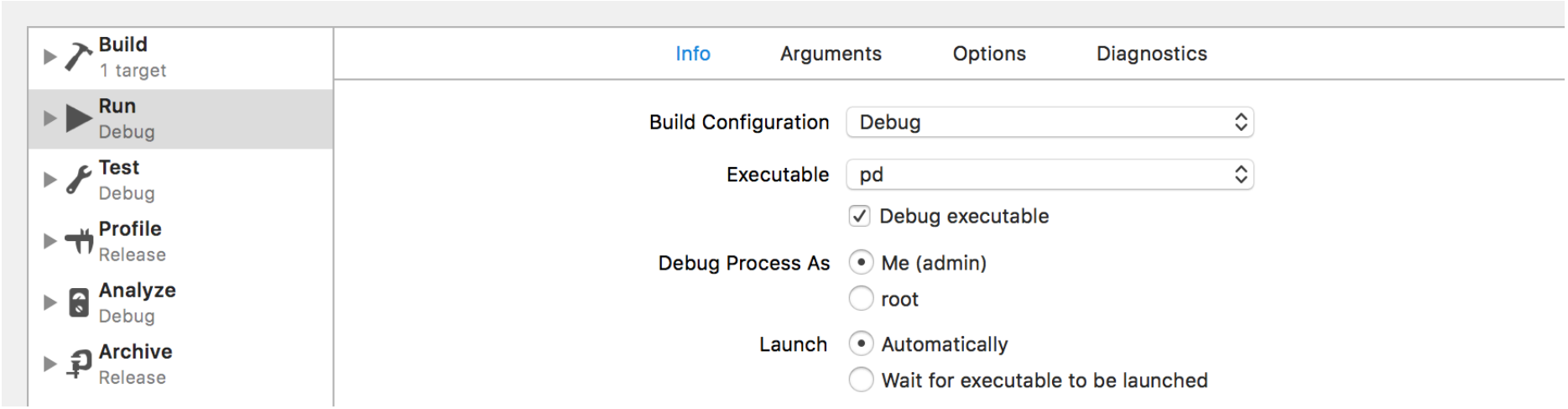


Select your project in the upper left corner and then enter into the search box on the right screen side *installation*. Add the path to the Pure data *extra* folder in the line *Installation Directory.*
Skip Install must be set to *No.*

Now go to Applications, CTRL + Click on Pure Data and select *Show Package Contents.* Goto *Contents/Resources,* CTRL + Click on *Bin* and select *Create Alias.* Move the Alias to your Desktop or wherever you want.

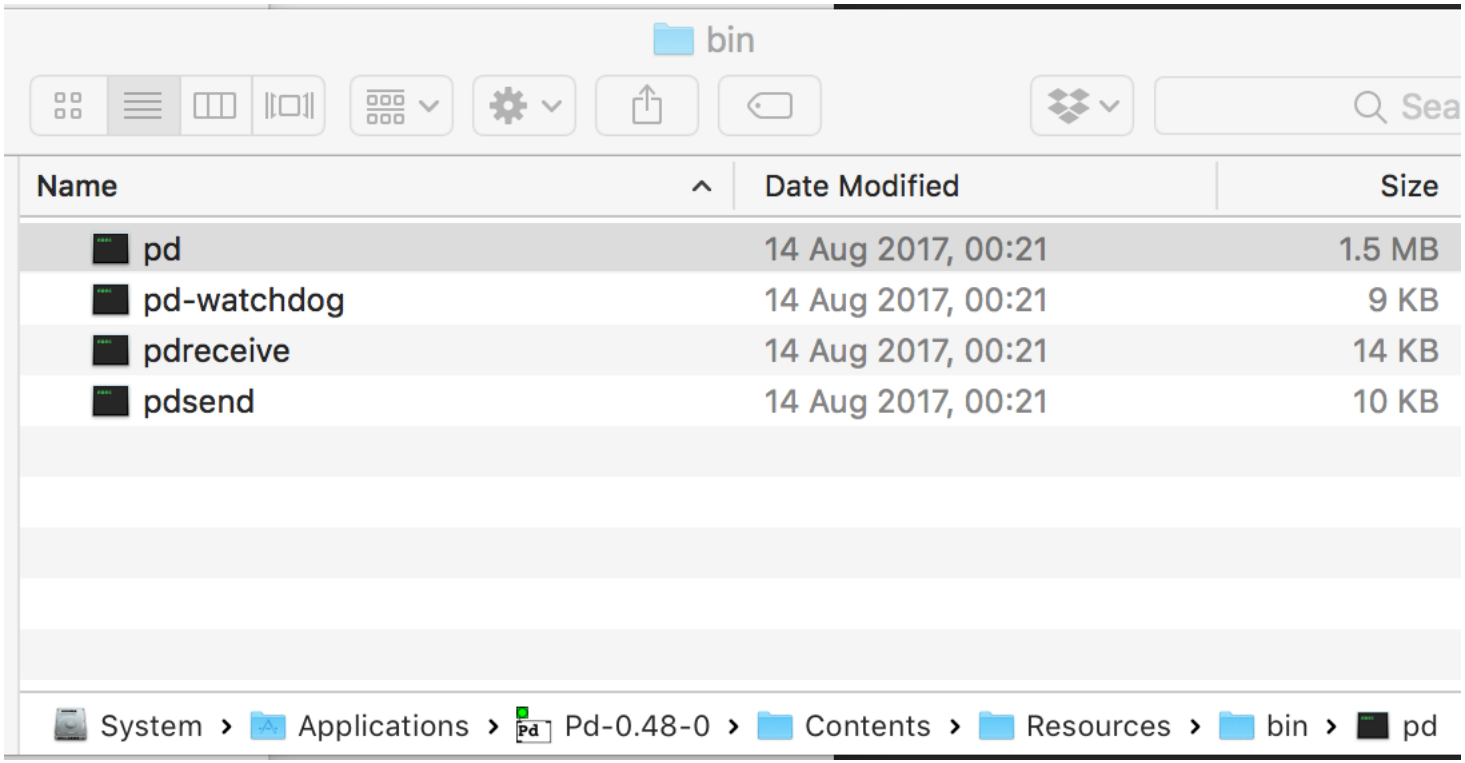
Create a debugging patch for your external in Pure Data which will be automatically loaded from Xcode.

Now select from the xCode Top Menu *Product->Scheme->Edit Scheme.*



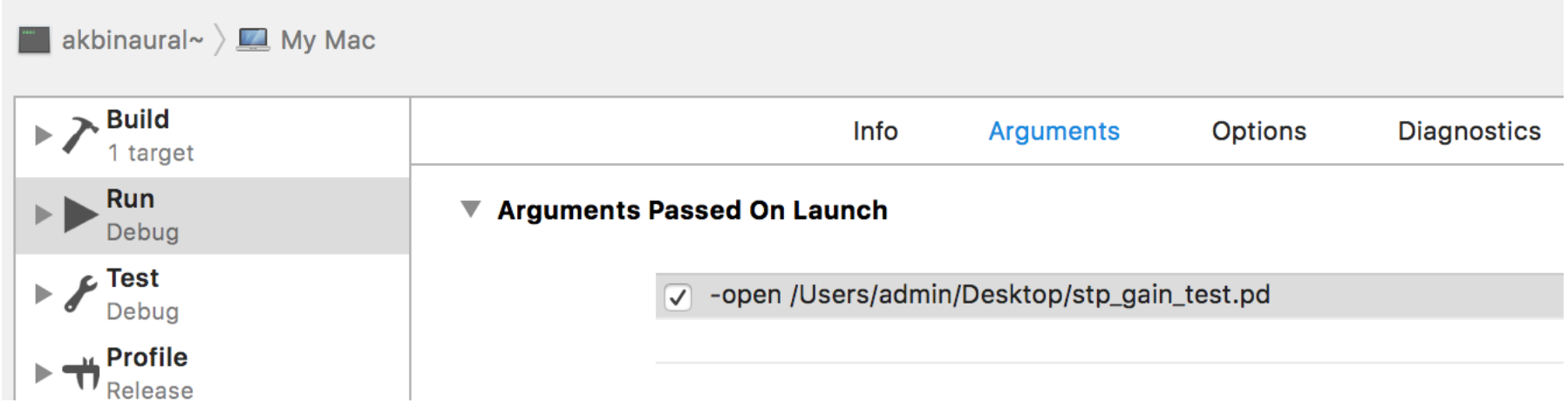
On the right side select *Run.*

For Build Configuration choose *Debug* and for Executable choose the *pd* unix executable inside the alias of the Bin folder we created earlier.

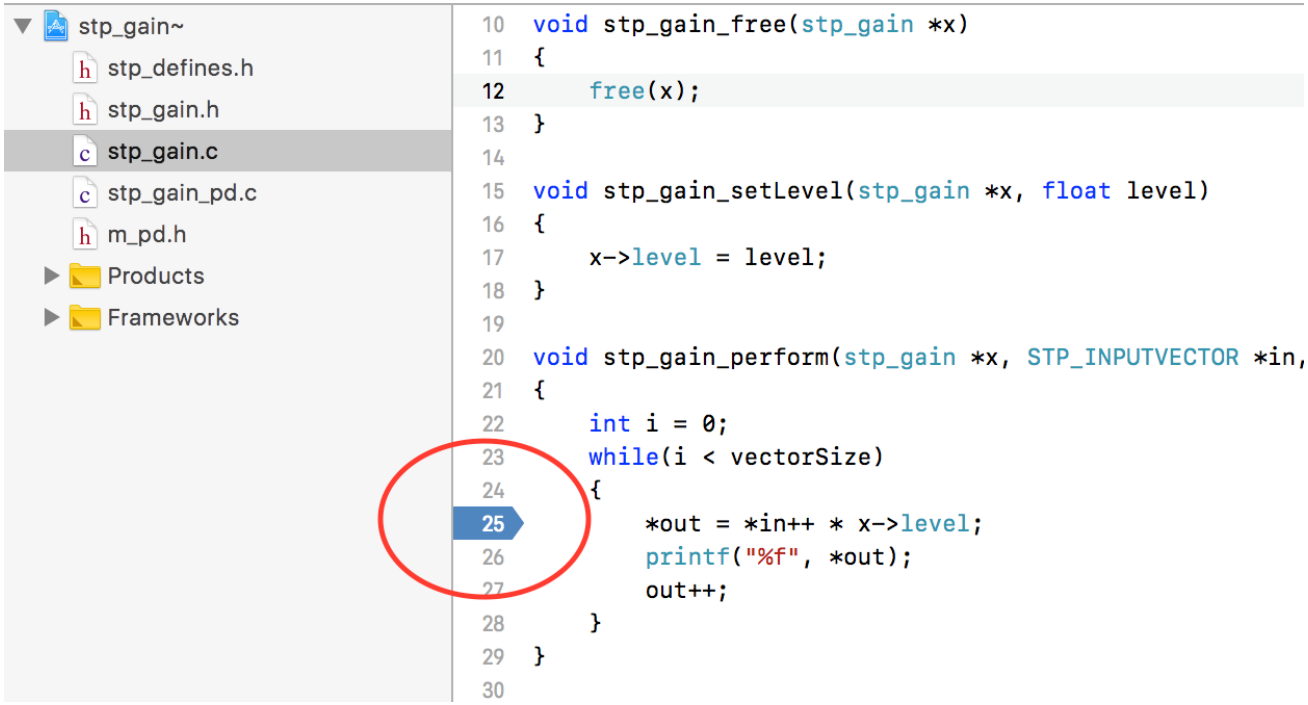


Now select the *Arguments* tab and under *Arguments Passed On Launch* enter

-open /PATH/TO/YOUR/DEBUGGINGPATCH



With a double click left to your code you can create breakpoints.

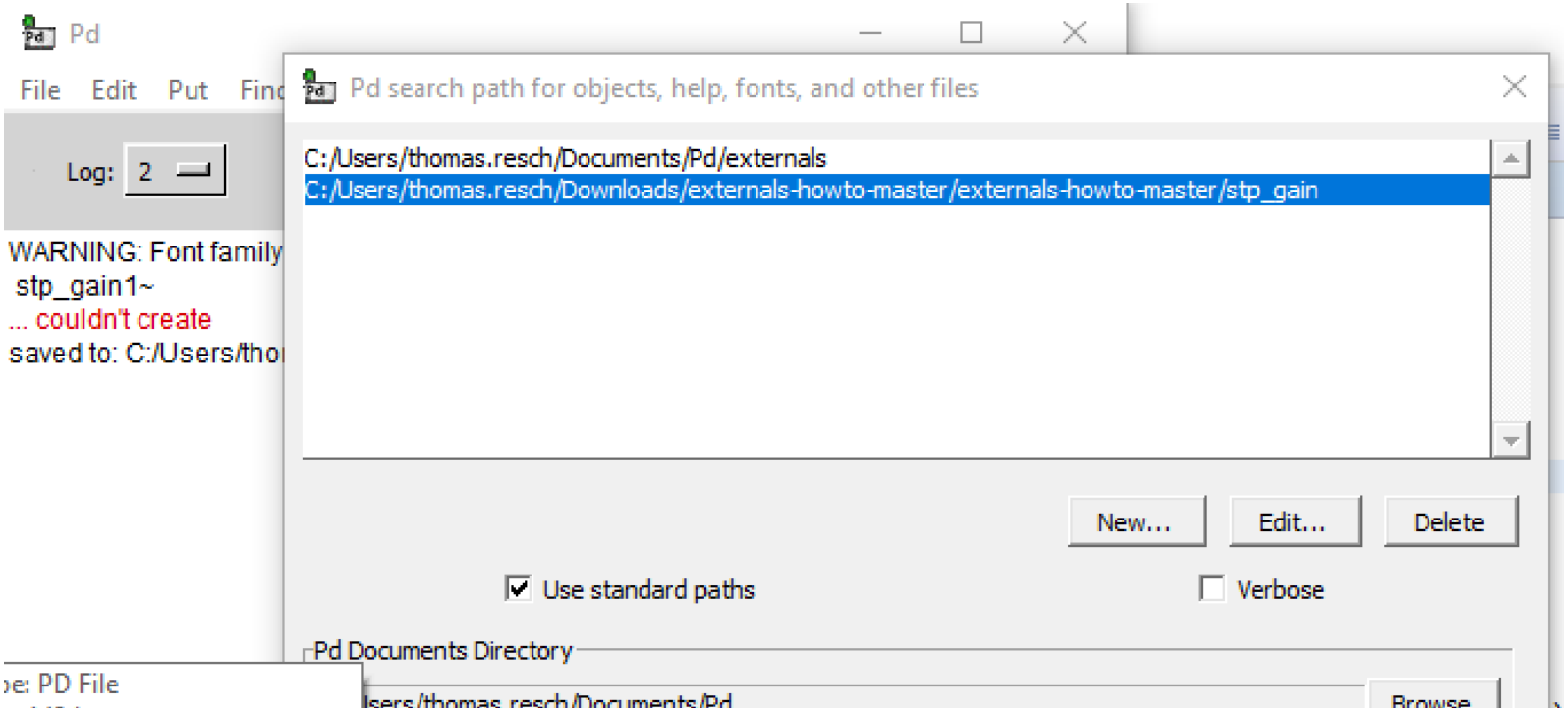


Now Press CMD + R (Run), Xcode will automatically launch Pd and open the testpatch for the Debugging. Under Debug from the XCode Menu you can step into/over from breakpoint to breakpoint.

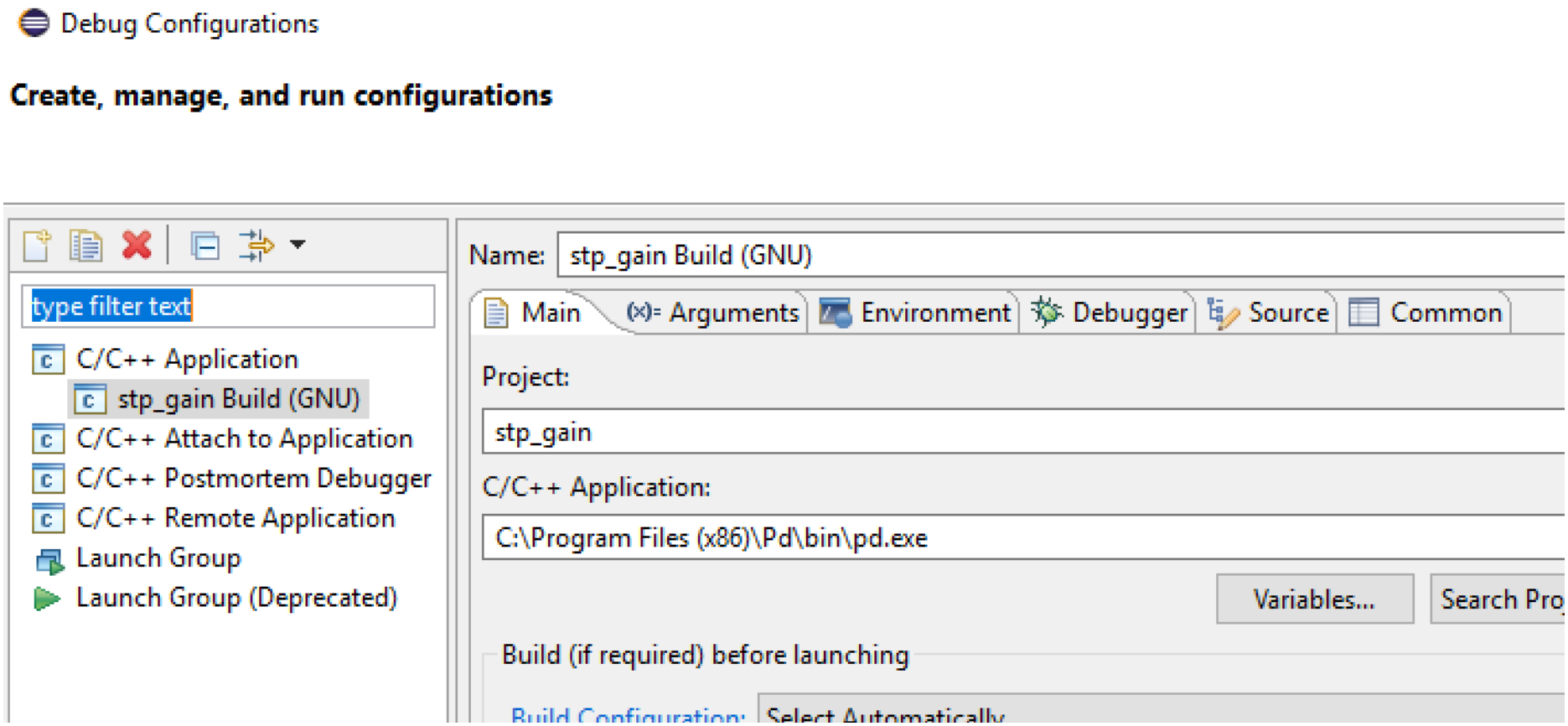
Eclipse

Unfortunately, under Windows the Debugger is not working correctly with Pure Data. We can optimize the Workflow but Breakpoints will not work.

Add your Pure Data Object Directory to Pure Data's search path, so you can skip the copying step to the Extra Folder (Pd -> Preferences -> Path)

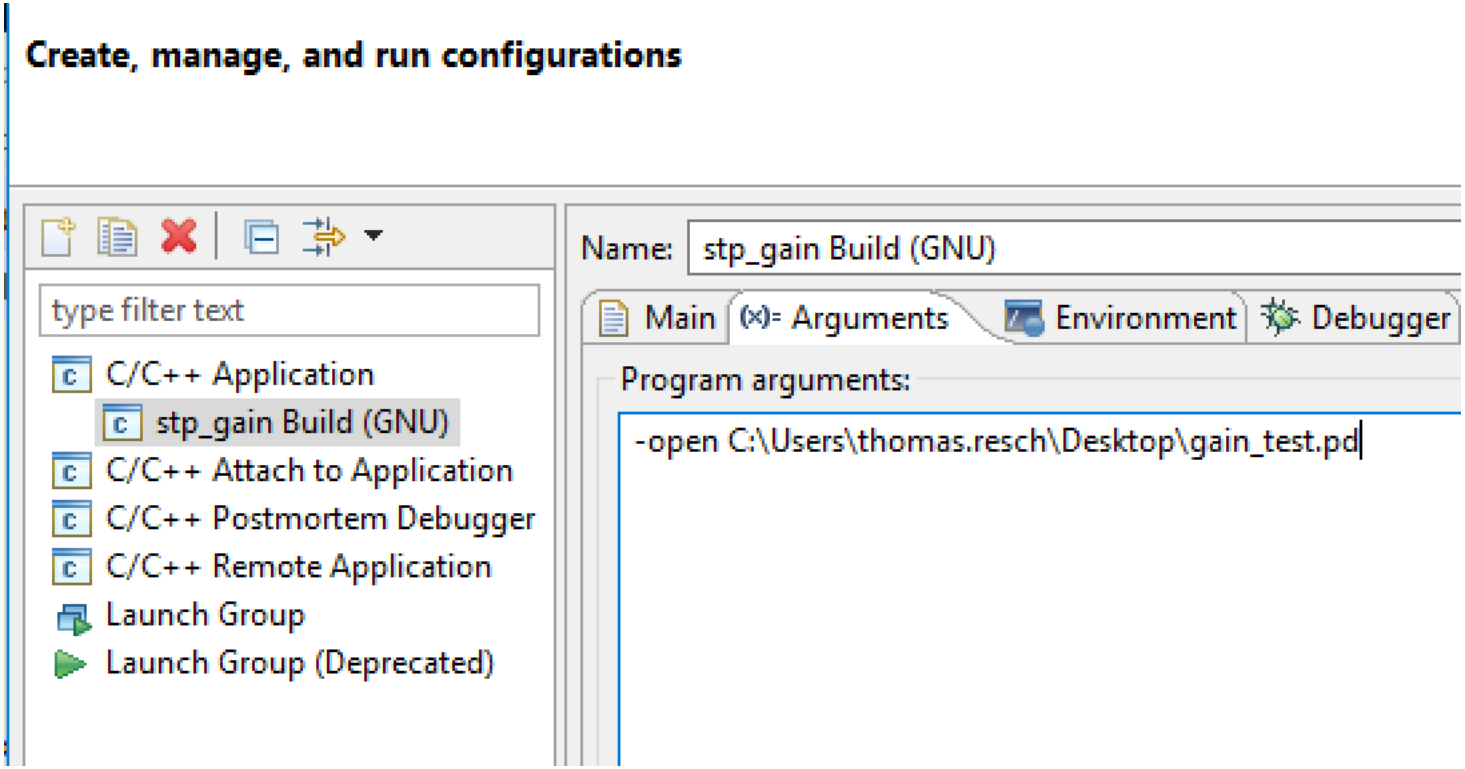


In Eclipse select Run->Debug Configurations from the Menu and create a new Configuration by double clicking on C/C++ Application on the left side on the screen.



Below *C/C++ Application:* enter the Path the pd.exe Binary.

Click on the Arguments Tab.



Under *Program arguments* type

-open C:\PATH\TO\YOUR\DEBUGGING\PATCH.

By pressing F11, Eclipse will automatically start Pure Data with the Debugging Patch. Use the Pd post() instruction for Debugging. Syntax is similar to printf() but printing will occur in the Pure Data Console Window.