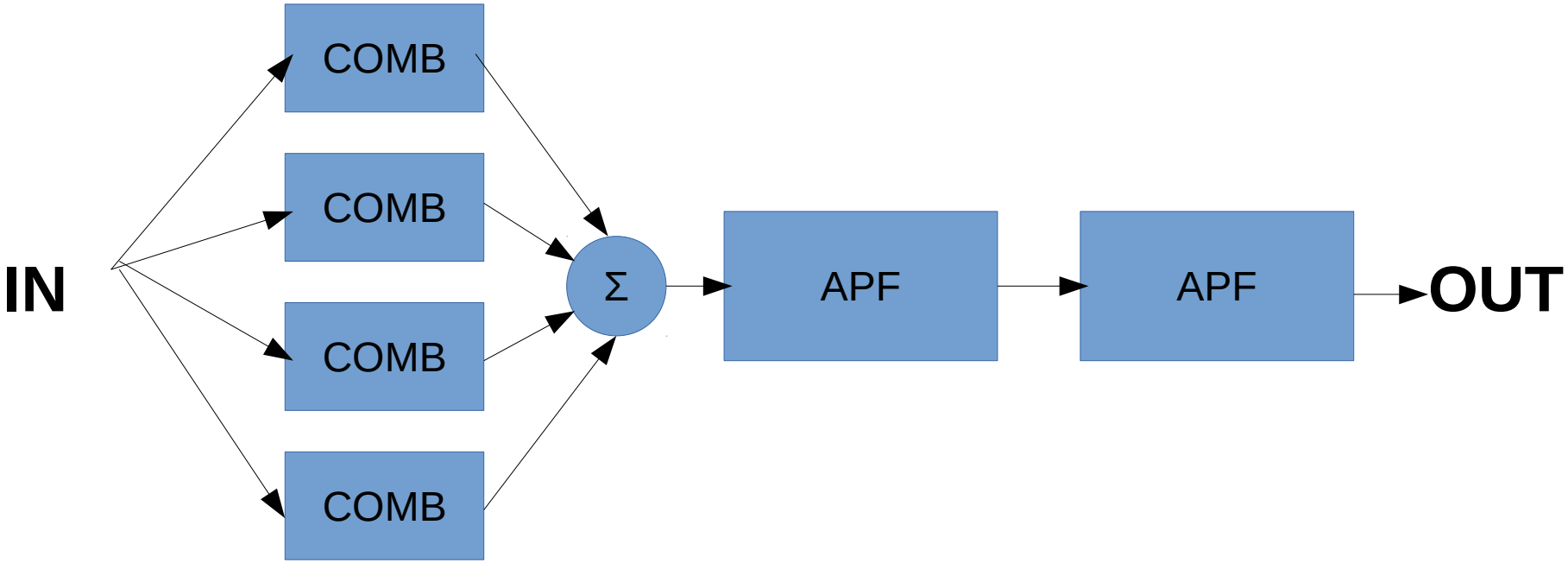
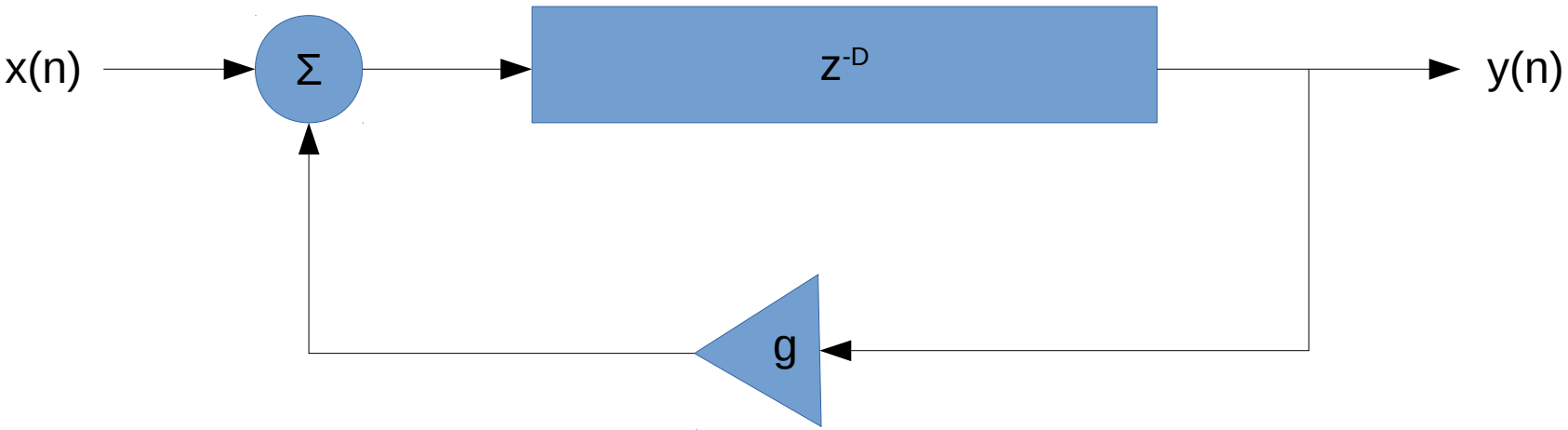
Schroeder’s Reverberator

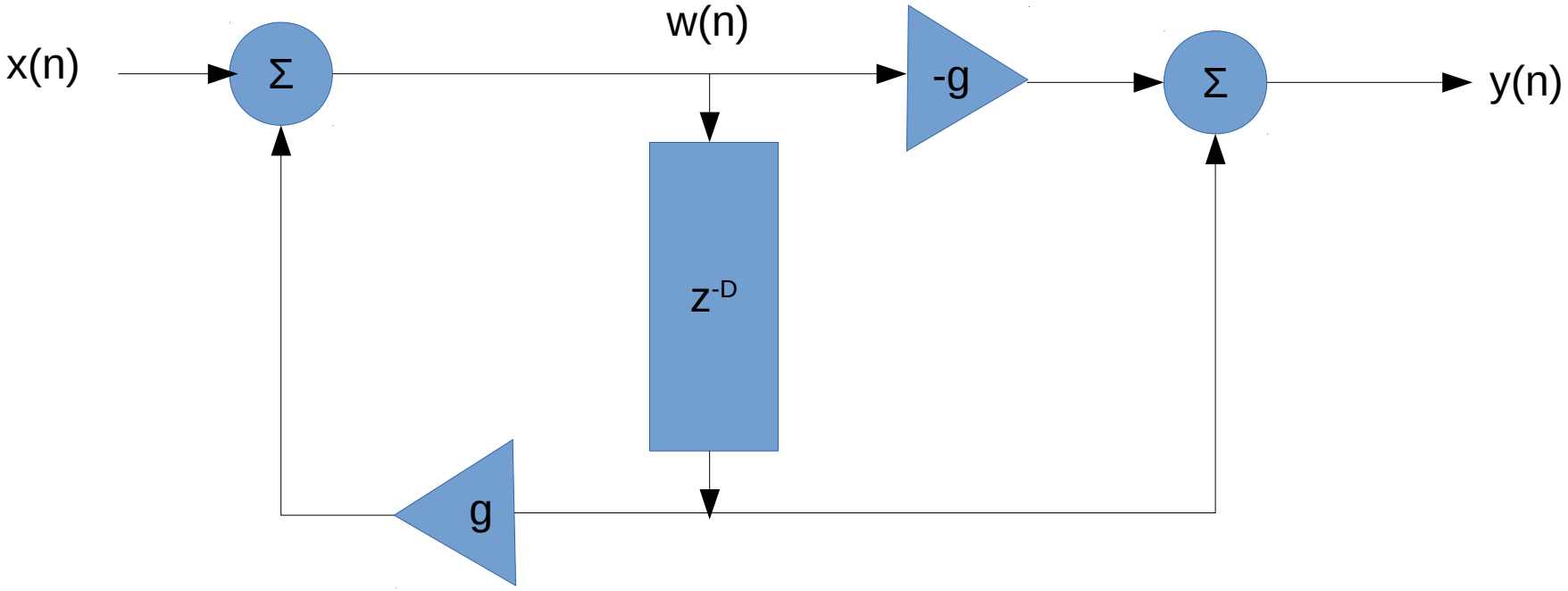


Comb Filter



$$y(n) = x(n - 1) + g * y(n - 1)$$

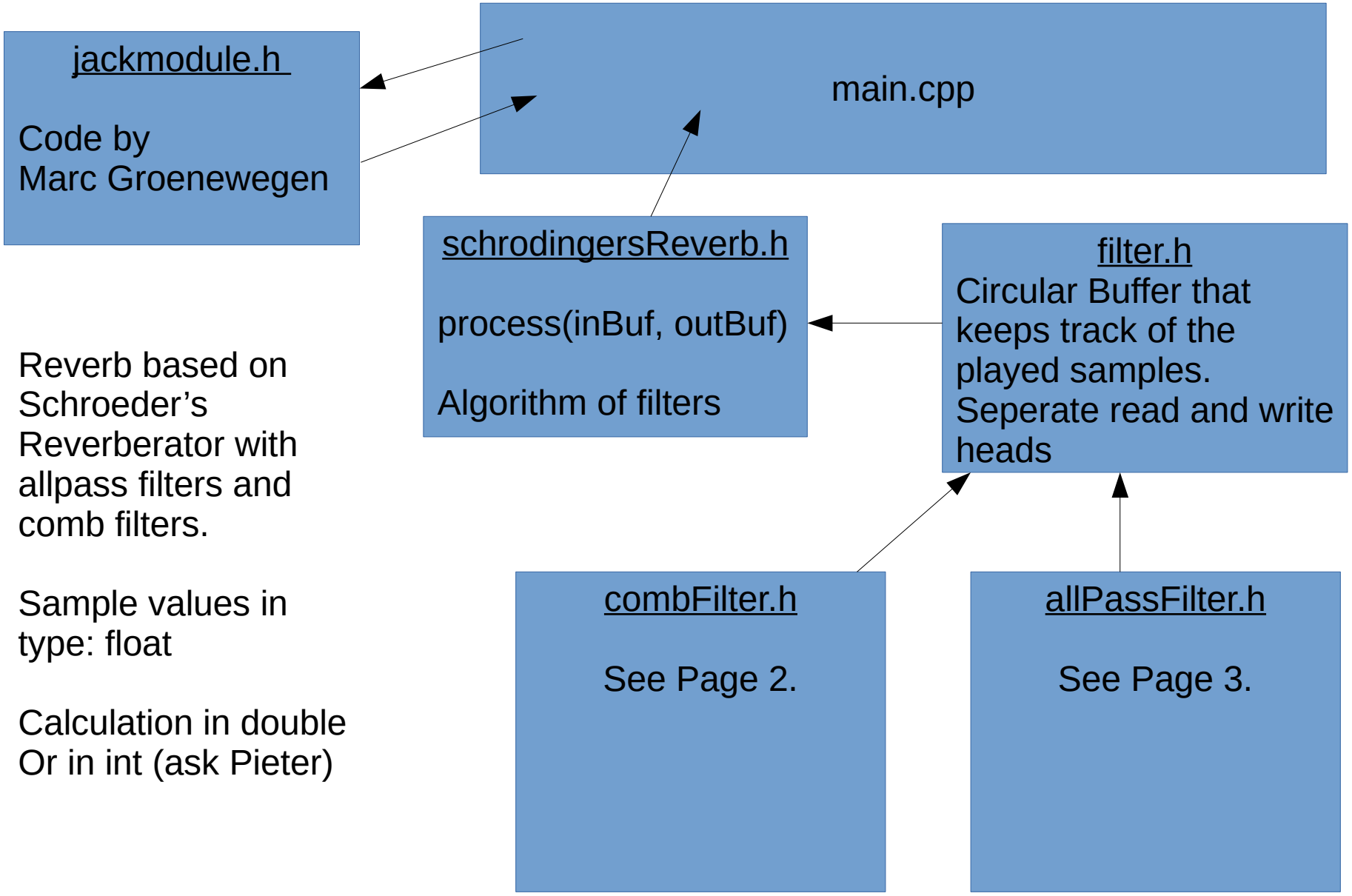
All Pass Filter



$$w(n) = x(n) + g * w(n - D)$$
$$y(n) = -g * w(n) + w(n - D)$$

$$y(n) = -g * x(n) + x(n - D) + g * y(n - D)$$

Class Diagram



Reverb based on Schroeder's Reverberator with allpass filters and comb filters.

Sample values in type: float

Calculation in double
Or in int (ask Pieter)

Schrödinger’s Reverb

Functionaliteit:
- Raspberry Pi
- Realtime input/output
- Reverb effect based on Schroeder’s reverberator.
- Use filter.h base class from synthesizer assignment.

Extra:
- tweaking the parameters
- changing the algorithm
- add potmeters to change the parameters.
- DIY DAC (2channels) and ADC (8 channels(audio in and potmeter in).

Schedule

Week 1: - Allpass and comb filter algorithms done.
- Research DIY ADC/DAC.

Week 2: - Reverb algorithm done.
- Parts ordered DIY ADC/DAC.

Week 3: - DIY ADC/DAC build.
- Tweak sound.

End result

What went right?
- The file structure I tried using for the first time worked really nice and I will be using it many more times.

-The book *“Designing Audio Effect Plug-Ins in C++”* worked really well. It had a lot of useful information.

- The idea to make a base class that keeps track of previous played sample so you can use it for your filter worked well.

Moments of enlightenment

Code :
- First make a basic Makefile and expand it as your project grows.

- Keep checking your code!

- Always give your variables logical names.

- Make small soundcheck files so you can check if your Raspberry Pi works correctly.

- Work on one feature at a time. Multitasking may feel more efficient, but it really isn’t.

Questions:

- Can you better use full length or abbreviated variable names?

Memo’s

Diy ADC & DAC
SPI ADC 12 bit 8 channel (2 audio, 6 parameters)
Or I2C ADC 12 bit 2 channels for audio and SPI ADC 10 bit 4 channel for parameters

To Do:
Choose better names for filterBuffer[ ], readIndex, writeIndex, readIndex2, writeIndex2, delay and delay2.
Think about x(n) and y(n).
Needs to be easy to read.