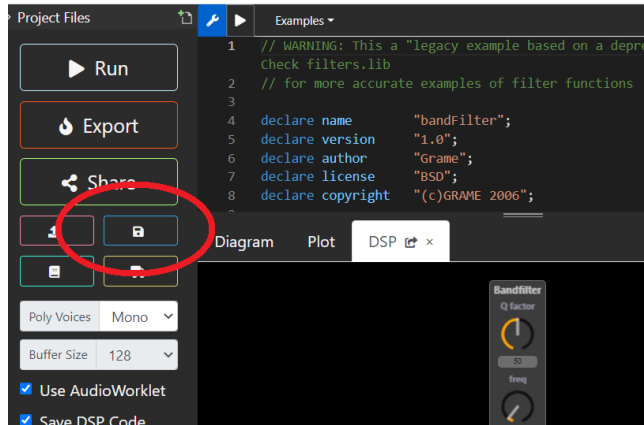


Phase 1: Part export a DSP file in C language

From an FaustIDE exportation button



From the Faust Command line interface

```
Faust.exe \  
-lang c -ct 1 -es 1 -mcd 16 -single -ftz 0  
./faust_echo.dsp  
>faust_echo.c
```

The screenshot shows the generated C code for the bandpass filter. The code includes standard headers, defines macros for C++ and Apple, and declares a structure for the filter parameters. The code is as follows:

```
15  
16  
17 #ifdef __cplusplus  
18 extern "C" {  
19 #endif  
20  
21 #if defined(_WIN32)  
22 #define RESTRICT __restrict  
23 #else  
24 #define RESTRICT __restrict__  
25 #endif  
26  
27 #include <math.h>  
28 #include <stdint.h>  
29 #include <stdlib.h>  
30  
31  
32 #ifndef FAUSTCLASS  
33 #define FAUSTCLASS mydsp  
34 #endif  
35  
36 #ifdef __APPLE__  
37 #define exp10f __exp10f  
38 #define exp10 __exp10  
39 #endif  
40  
41 typedef struct {  
42     FAUSTFLOAT fHslider0;  
43     FAUSTFLOAT fHslider1;  
44     int IOTA0;  
45     float Over0[8192];  
46 }
```

Phase 2: Generate the Audio Chain source code

```
python Faust2AC.py generate_ac  
--name "faust-echo" --group  
"sound_fx" faust_echo.c
```

Generated

Name	Date modified	Type	Size
audio_chain_faust_echo.c	04/07/2023 13:07	C File	11 KB
audio_chain_faust_echo.h	04/07/2023 13:07	H File	2 KB
audio_chain_faust_echo_factory.c	04/07/2023 13:07	C File	3 KB
audio_faust_echo.h	04/07/2023 13:07	H File	2 KB

What does **Faust2AC.py** ?

- Add a « **static** » before each function to prevent namespace collision
- Disable by an ifdef, useless functions for audio chain
- Generate the AC Designer user control description and factory
- Generate extra source code for the audio chain integration

The code generated can be immediately built and Tested in IAR or CubeID