



Design of a Portable Digital Signal Processing Workstation

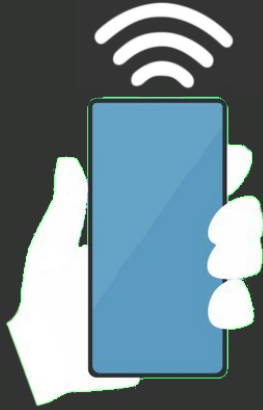
Clyne Sullivan

Mariusz Jankowski

What is digital signal processing (DSP)?

Used to transform and understand signals...

... for sending information



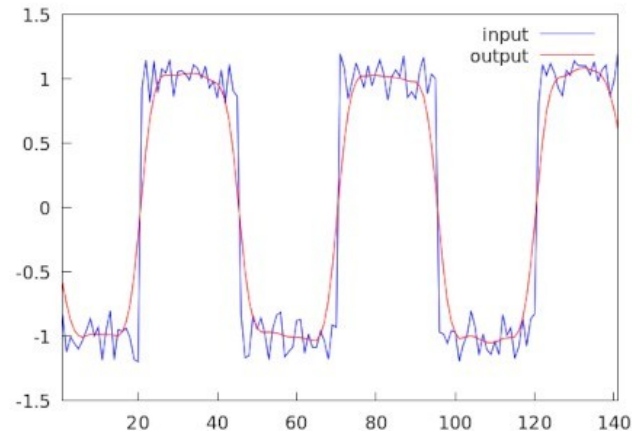
... for audio effects



DSP Education

- Course dedicated to DSP
- Use programs like Mathematica or Maxima to design and test algorithms
- No real-life lab component

```
wxdraw2d(  
  point_type=None,  
  points_joined=True,  
  key="input", color=blue, points(swave),  
  key="output", color=red, points(convolve(swave, h)),  
  yrange=[-1.5, 1.5]  
);
```



Possible Solutions

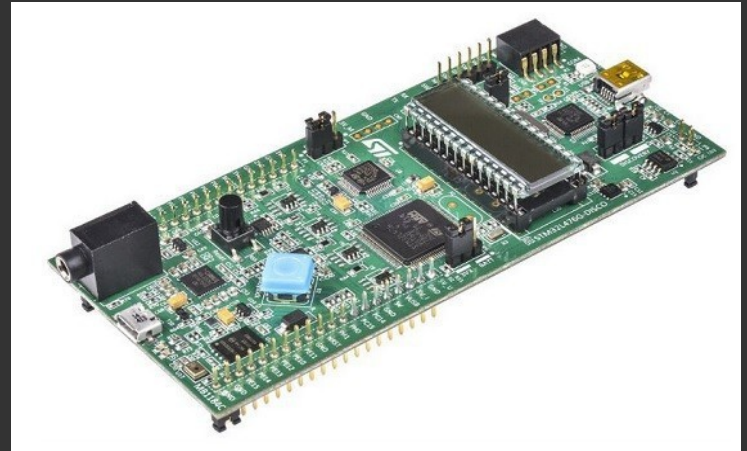
TI C6000 DSP Units

- Very capable
- Expensive



UMaine Orono's ECE486 Labs

- Achieves hands-on learning
- Lacks features and is too complex



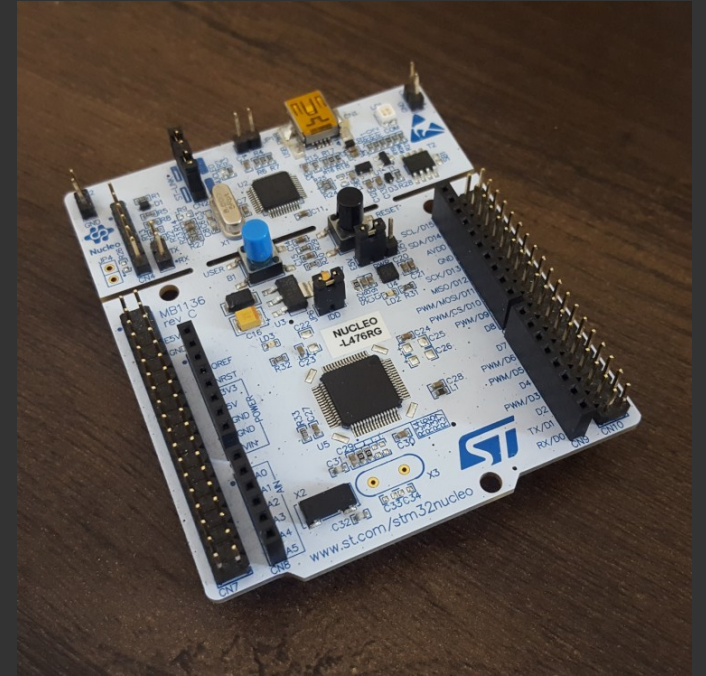


Our Solution

1. Focused on DSP algorithm design
2. Single-device solution (just add a computer)
3. Low-cost and accessible to students

Our Solution

1. Selected a target processor



Our Solution

2. Wrote device firmware

- Designed so student only needs to focus on DSP algorithm design
- Withstands algorithm coding errors

```
msg_t message;
asm("svc 0; mov %0, r0" : "=r" (message)); // sleep until next
if (message != 0) {
    auto samples = MSG_FOR_FIRST(message) ? samplesIn.data() :
    auto size = samplesIn.size() / 2;

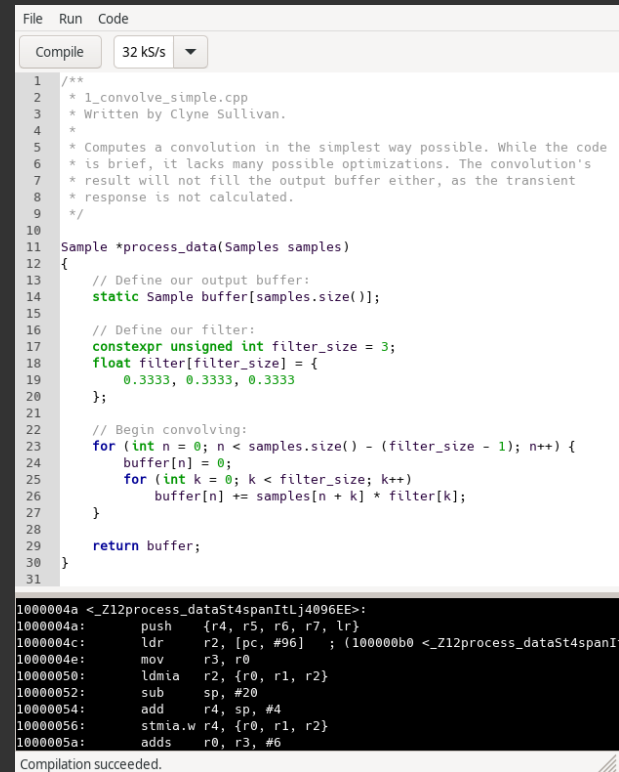
    if (elf_entry) {
        if (!MSG_FOR_MEASURE(message)) {
            // Remember the stack pointer in case the algorithm
            uint32_t sp;
            asm("mov %0, sp" : "=r" (sp));
            samples = elf_entry(samples, size);
            asm("mov sp, %0" :: "r" (sp));
        } else {
            uint32_t sp;
            asm("mov %0, sp; eor r0, r0; svc 2" : "=r" (sp));
            samples = elf_entry(samples, size);
            asm("mov r0, #1; svc 2; mov sp, %0" :: "r" (sp));
        }
    }

    if (samples != nullptr) {
        if (MSG_FOR_FIRST(message))
            samplesOut.modify(samples, size);
        else
            samplesOut.midmodify(samples, size);
    }
}
```

Our Solution

3. Built computer program

- Styled similar to the Arduino IDE
- Includes useful analysis features

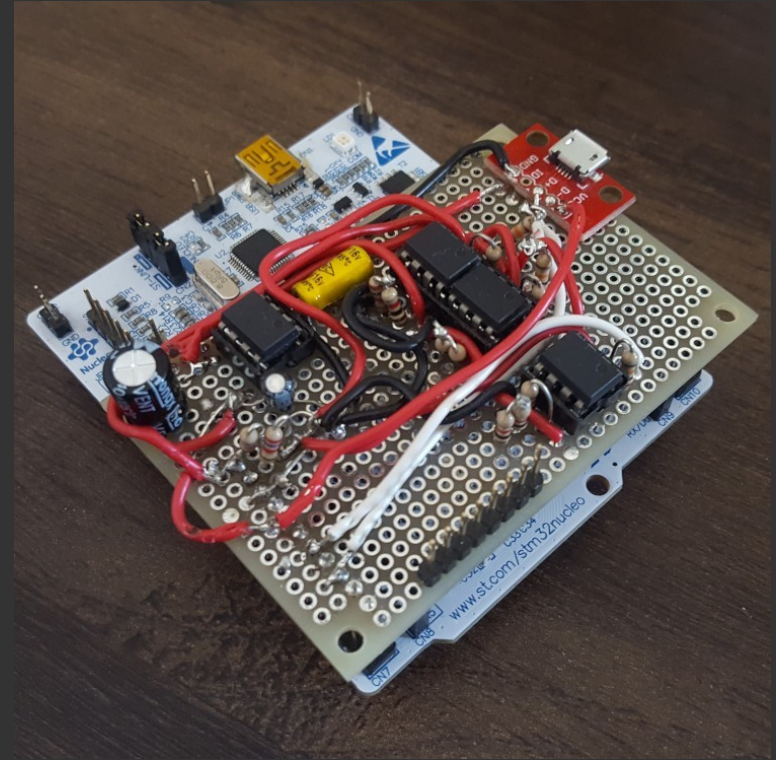


```
File Run Code
Compile 32 k/s
1 /**
2  * 1_convolve_simple.cpp
3  * Written by Clyne Sullivan.
4  *
5  * Computes a convolution in the simplest way possible. While the code
6  * is brief, it lacks many possible optimizations. The convolution's
7  * result will not fill the output buffer either, as the transient
8  * response is not calculated.
9  */
10
11 Sample *process_data(Samples samples)
12 {
13     // Define our output buffer:
14     static Sample buffer[samples.size()];
15
16     // Define our filter:
17     constexpr unsigned int filter_size = 3;
18     float filter[filter_size] = {
19         0.3333, 0.3333, 0.3333
20     };
21
22     // Begin convolving:
23     for (int n = 0; n < samples.size() - (filter_size - 1); n++) {
24         buffer[n] = 0;
25         for (int k = 0; k < filter_size; k++)
26             buffer[n] += samples[n + k] * filter[k];
27     }
28
29     return buffer;
30 }
31
1000004a <_Z12process_dataSt4spanItLj4096EE>:
1000004a: push    {r4, r5, r6, r7, lr}
1000004c: ldr    r2, [pc, #96] ; (100000b0 <_Z12process_dataSt4spanI
1000004e: mov    r3, r0
10000050: ldmia r2, {r0, r1, r2}
10000052: sub    sp, #20
10000054: add    r4, sp, #4
10000056: stmia.w r4, {r0, r1, r2}
1000005a: adds  r0, r3, #6
Compilation succeeded.
```


Our Solution

4. Designed additional hardware

- Allows working with a wide variety of signals, including audio



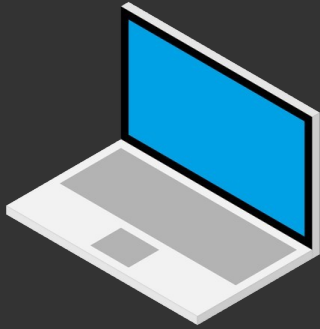
Our Solution

4. Designed additional hardware

- Allows working with a wide variety of signals, including audio



How it works



- Write algorithms
- Configure device
- Analyze results



- Signal input/output
- Algorithm application



Signal generators,
oscilloscopes,
speakers, ...



Demonstration

Audio echo effect

File Run Code

Compile

48 kS/s



```
1 Sample *process_data(Samples samples)
2 {
3     constexpr unsigned int D = 2000;
4
5     static Sample output[samples.size()];
6     static Sample prev[D]; // prev[0] = output[0 - D]
7
8     float alpha = readpot1() / 4095.;
9
10    // Do calculations with previous output
11    for (unsigned int i = 0; i < D; i++)
12        output[i] = samples[i] + alpha * (prev[i] - 2048);
13
14    // Do calculations with current samples
15    for (unsigned int i = D; i < samples.size(); i++)
16        output[i] = samples[i] + alpha * (output[i - D] - 2048);
17
18    // Save outputs for next computation
19    for (unsigned int i = 0; i < D; i++)
20        prev[i] = output[samples.size() - (D - i)];
21
22    return output;
23 }
24
```

text	data	bss	dec	hex	filename
2684	0	12192	14876	3a1c	/tmp/stmbspGUISSfwrR.o

Ready.



The future

- Prepare for student use
- Add additional features
- Other applications?