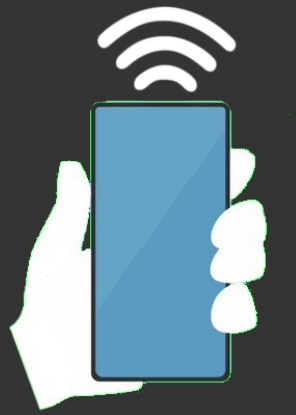
Design of a Portable Digital Signal Processing Workstation

Clyne Sullivan
Mariusz Jankowski

What is digital signal processing (DSP)?

Used to transform and understand signals…

… for sending information

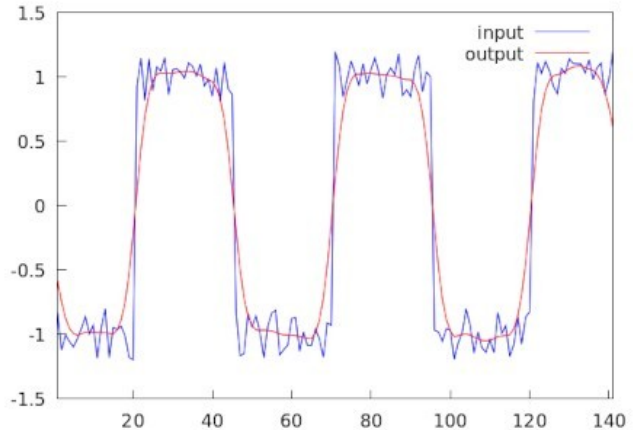


… for audio effects

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]
);
```



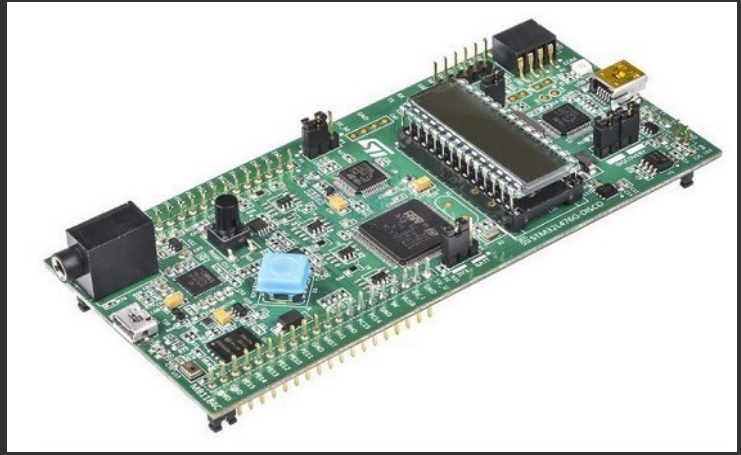
Possible Solutions

TI C6000 DSP Units

- Very capable
- Expensive

UMaine Orono's ECE486 Labs

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

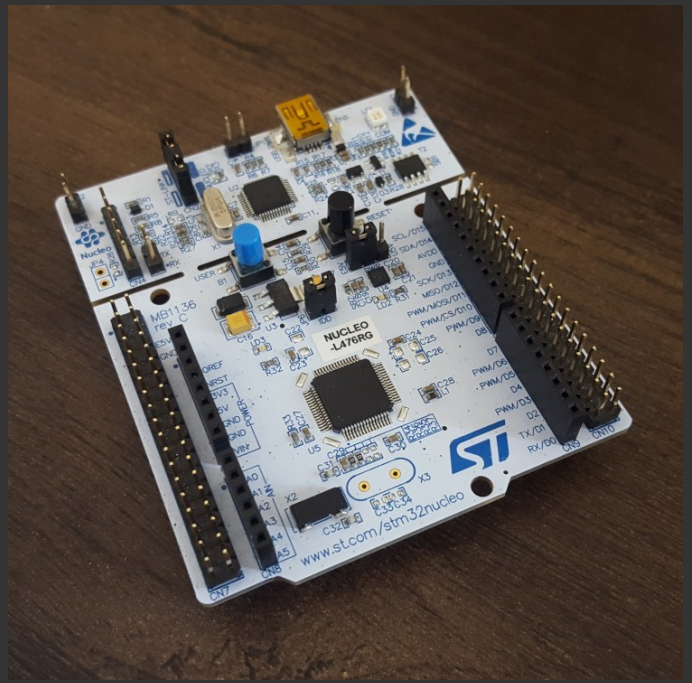


Our Solution

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

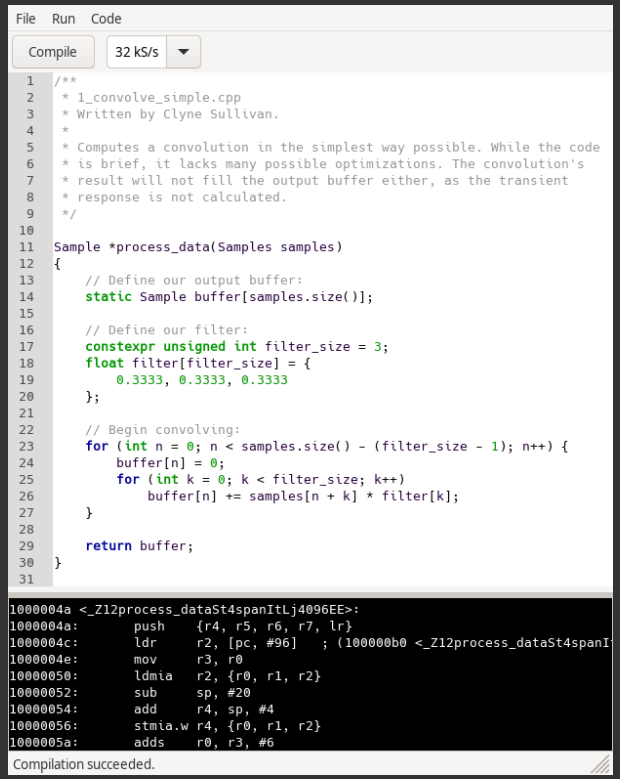
- Designed so student only needs to focus on DSP algorithm design
- Withstands algorithm coding errors

```
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 algorith
            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

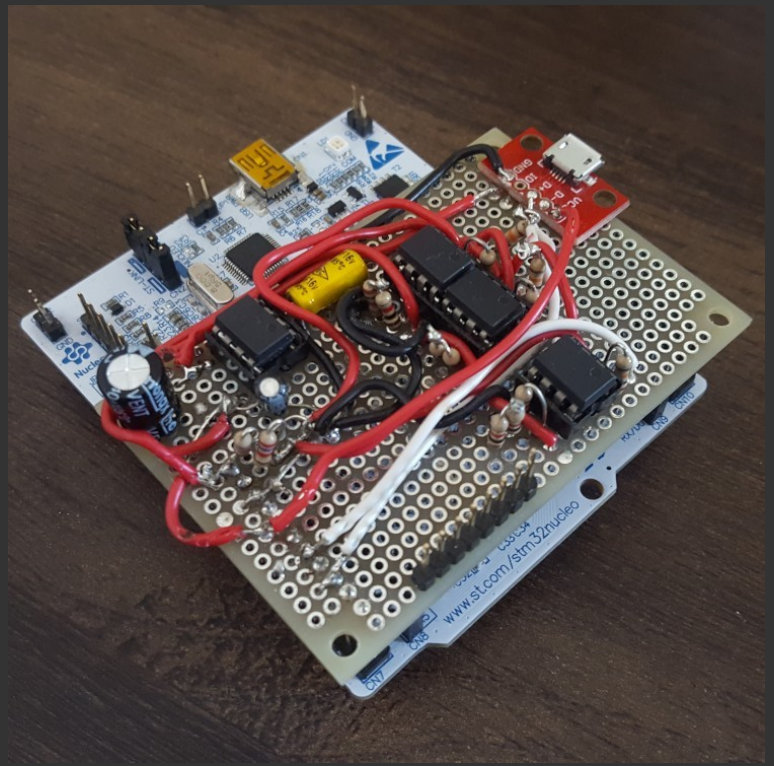
- Styled similar to the Arduino IDE
- Includes useful analysis features



Our Solution

4. Designed additional hardware

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

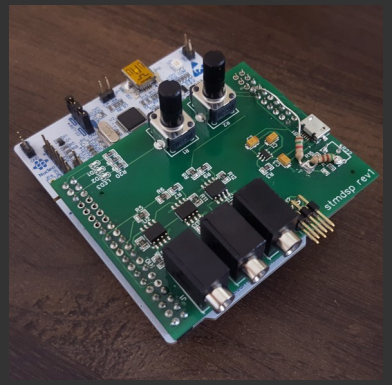


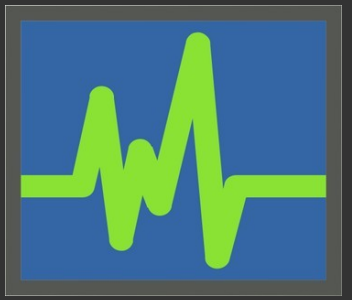
Our Solution

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/stmdspguiSSfwrR.o
```

Ready.

The future

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