Juniper: A Functional Reactive Programming Language for the Arduino

Hands-on Sound Visualization Project
Please go to
http://www.juniper-lang.org/

Caleb Helbling
Tufts University

Samuel Z. Guyer
Tufts University

Workshop on Functional Art, Music, Modelling and Design (FARM)
September 2016
Parts List

Nano microcontroller
Sound sensor
100 ohm resistor
8 LED Bargraph
Bridged 8 pin header
8 pin jumper wire
3 pin jumper wire
Micro-usb cable
Internal Breadboard Layout
Putting the "work" back in workshop

VCC: 2.4-5.5V
Adjustable gain
OUT GND VCC

3V3 A1 GND
D2 - D9
module SoundBar
open (Prelude)

let microphonePin = 15
let barPins = [9, 8, 7, 6, 5, 4, 3, 2]
let numBarPins = 8

fun setup() = (  
  Io:setPinMode(microphonePin, Io:input());  
  for i in 0 to numBarPins - 1 do  
    Io:setPinMode(barPins[i], Io:output())  
  end
)

...
fun drawBar(level : uint16) = (  
  for i in 0 to level do
      Io:digWrite(barPins[i], Io:high())
  end;
  for i in level + 1 to numBarPins - 1 do
      Io:digWrite(barPins[i], Io:low())
  end
)
let state = ref List:replicate<uint16; 5>(0, 0)
fun main() = (  
    setup();  
    while true do (  
        let micSig = Io:digIn(microphonePin);  
        let barSig = Signal:map(  
            fn (digVal) ->  
                case digVal of  
                    | Io:low() => 7u16  
                    | Io:high() => 0u16  
                end  
            end,  
            micSig);  
        let pastBarSig = Signal:record(barSig, state);  
        let meanBarSig = Signal:map(List:average, pastBarSig);  
        Signal:sink(drawBar, meanBarSig)  
    ) end  
)
Thank you!

http://www.juniper-lang.org/