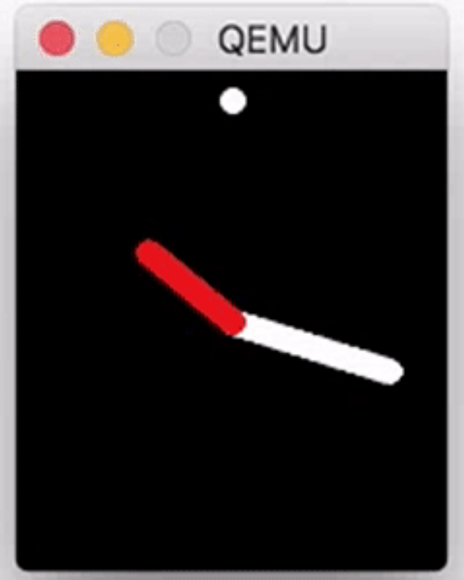
linkedin.com/in/francois-baldassari

linkedin.com/in/hbehrens

	Pebble Time	Pebble Time Round
CPU	ARM Cortex M4 @ 100 MHz 64KB RAM (incl. code and 2KB stack)	
Display	144x168px @ 64 colors (Rectangular)	180x180 @ 64 colors (Circular)
Peripheral	Bluetooth LE, 4 Buttons, Accelerometer, Magnetometer, Ambient Light Sensor, Vibrations, Microphone, SmartStrap	
OS	based on FreeRTOS with multiple tasks (system UI + unprivileged apps)	
3rd Party	appstore with >12,000 apps, >500 daily active developers CLI SDK & cloud-based environment	

https://developer.pebble.com/guides/tools-and-resources/hardware-information/



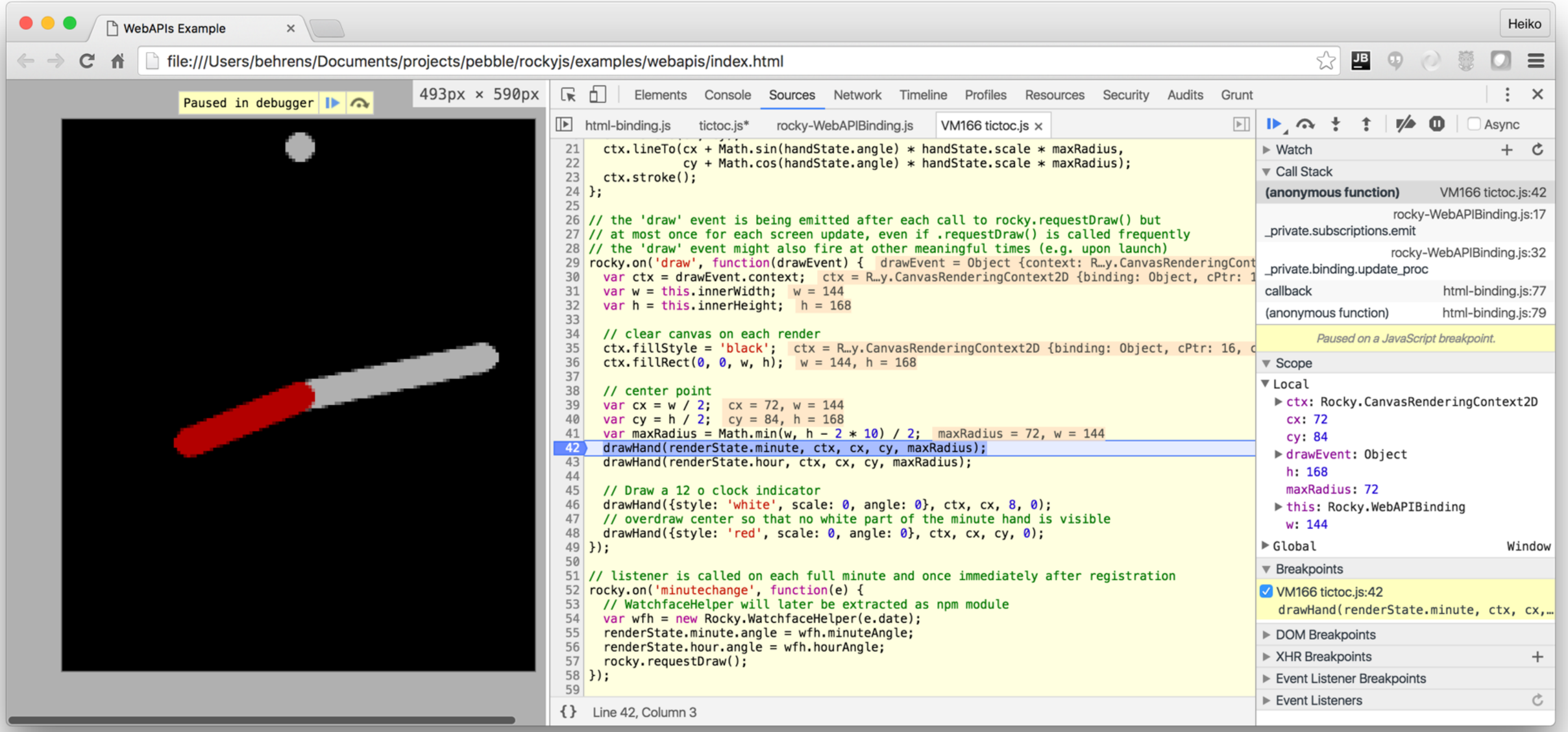
Pebble default watchface “TicToc” on firmware 3.12 is using JerryScript.

SDK for 3rd-party developers on the road map for this year.

```
var renderState = {
    minute: {style: 'white', scale: 0.80, angle: 0},
    hour:   {style: 'red',   scale: 0.51, angle: 0}
};

var drawHand = function(handState, ctx, cx, cy, maxRadius) {
    ctx.lineWidth = 8;
    ctx.strokeStyle = handState.style;
    ctx.beginPath();
    ctx.moveTo(cx, cy);
    ctx.lineTo(cx + Math.sin(handState.angle) * handState.scale * maxRadius,
               cy + Math.cos(handState.angle) * handState.scale * maxRadius);
    ctx.stroke();
};

rocky.on('draw', function(drawEvent) {
    var ctx = drawEvent.context;
    var w = this.innerWidth;
    var h = this.innerHeight;

    // clear canvas on each render
    ctx.fillStyle = 'black';
    ctx.fillRect(0, 0, w, h);

    // center point
    var cx = w / 2;
    var cy = h / 2;
    var maxRadius = Math.min(w, h - 2 * 10) / 2;
    drawHand(renderState.minute, ctx, cx, cy, maxRadius);
    drawHand(renderState.hour, ctx, cx, cy, maxRadius);

    // Draw a 12 o clock indicator
    drawHand({style: 'white', scale: 0, angle: 0}, ctx, cx, 8, 0);
});

// listener is called on each full minute and once immediately after registration
rocky.on('minutechange', function(e) {
   var wfh = new WatchfaceHelper(e.date);
    renderState.minute.angle = wfh.minuteAngle;
    renderState.hour.angle = wfh.hourAngle;
    rocky.requestDraw();
});
```

Findings (based off JerryScript mid Feb)

- **global state introduces challenges**
  (multi tasking, uninitialized state)
- **launch performance** (due to parsing, execution)
  - **parsing** (10KB JS ~5s)
  - **execution** (below 10 FPS)
- **stack** (TicToc.js requires ~8KB)
- **development is difficult**
  - **debugging** (poor stack traces)
  - **education around embedded** (low-memory)



http://pebble.github.io/rockyjs/

Cross-compiled our firmware's application layer with Emscripten to develop apps in the browser.

Near-Term Challenges

- stable snapshot format
  - rare changes (or forward-compatible)
  - compatible across platforms
  - ideal: executes & stores heap of root-level script
- lower stack requirements (e.g. storing as part of `mem_heap_t`)
- education / low memory strategies

Medium-Term Challenges

- performance
- multi-instance
- debugging & tooling