JerryScript on RIOT
An ultra-lightweight JavaScript engine for the Internet of Things

Tilmann Scheller
Principal Compiler Engineer
t.scheller@samsung.com

Samsung Open Source Group
Samsung Research UK

RIOT Summit 2016
Berlin, Germany, July 15 – 16, 2016

Overview
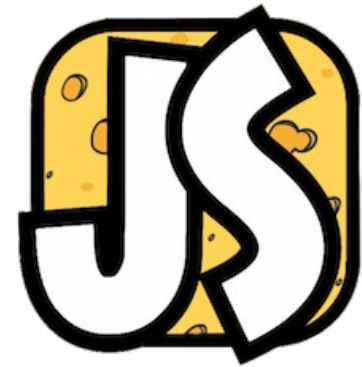
- Introduction

- JerryScript

- Memory consumption/Performance

- JerryScript on RIOT

- Demo

- Future work

- Summary

Introduction

What is JerryScript?

- A really lightweight JavaScript engine

- Has a base footprint of ~3KB of RAM

- Optimized for microcontrollers

- Developed from scratch by Samsung

- Actively developed on GitHub

- JerryScript is an open source project released under the Apache License 2.0

Why JavaScript on microcontrollers?

- There's a huge pool of JavaScript developers

- Opens up the possibility for web developers to easily write software for embedded devices

- Performance overhead of JavaScript less of an issue for control tasks

- Increased productivity, shorter time to market

- Ability to load code dynamically over the network

- Security: Executing JavaScript code is safer than executing arbitrary native code

JerryScript

JerryScript History

- Development started in June 2014

- Released as open source in June 2015

- JerryScript passed 100% of the test262 conformance test suite in August 2015

- Rewritten compact byte code implementation landed in January 2016

- Current focus on performance optimization

JerryScript
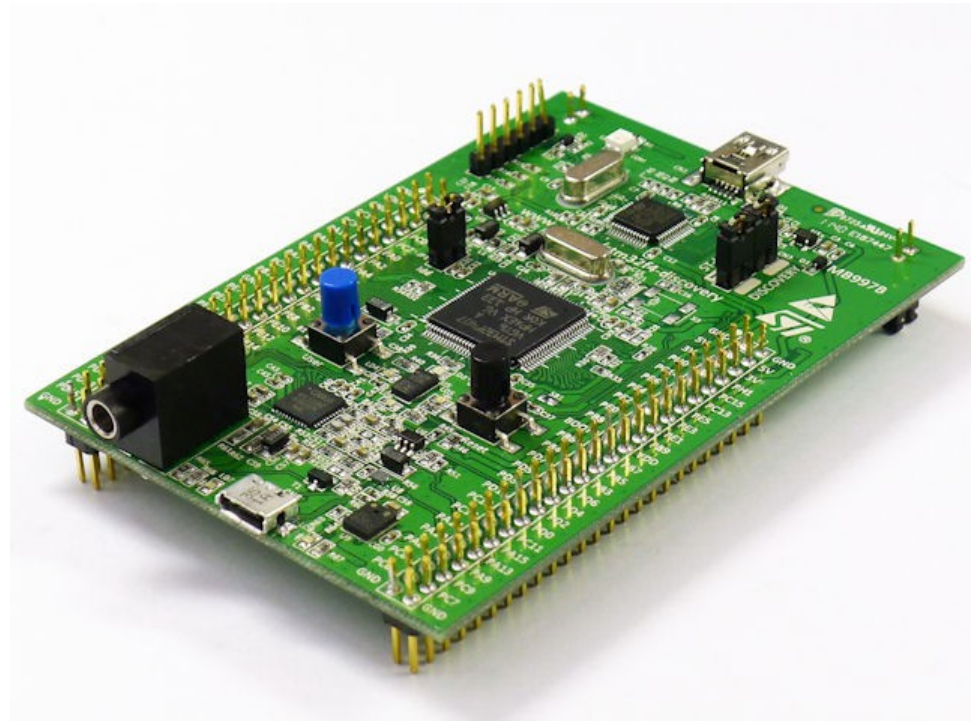
- Heavily optimized for a low memory footprint

- Interpreter-only

- Compact object representation

- Compressed pointers

- No AST, directly creating byte code

- Compact byte code heavily optimized for low memory consumption

JerryScript Portability

- Extremely portable

- Self-contained

- Small C library

- Can run bare-metal

- Supports the STM32F4, ESP8266, Arduino 101, FRDM-K64F boards

- OS support: NuttX, Zephyr, mbed, RIOT

- Runs on Linux/macOS as well

Target hardware

- STM32F4 developer board

- Cortex-M4F clocked at 168 MHz

- 192KB of RAM

- 1MB of flash memory
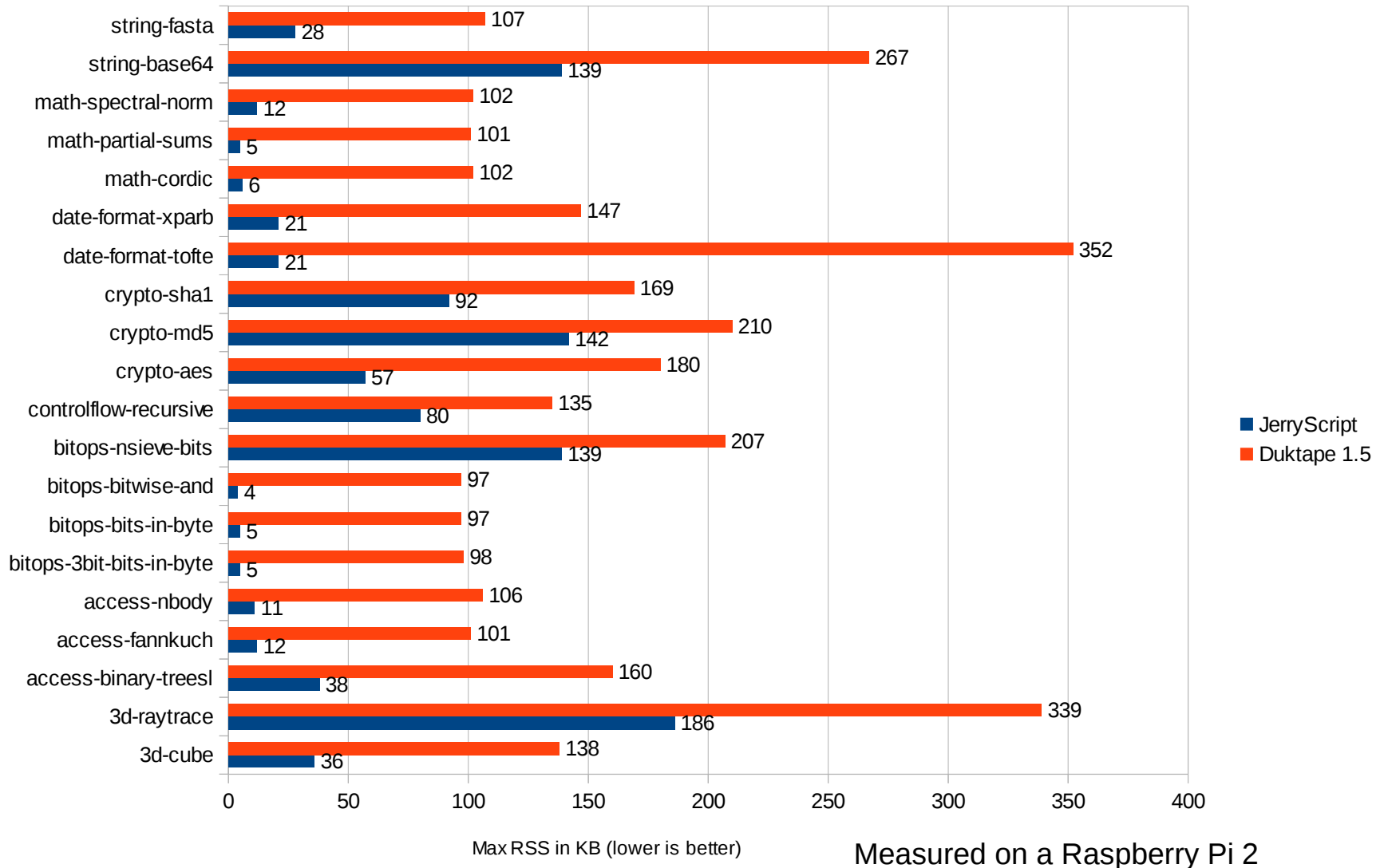
JerryScript

- Written in C99

- About 82KLOC

- Code size 156KB when compiled with GCC in LTO mode for ARM Thumb-2

- Implements the entire ECMAScript 5.1 standard, passes 100% of the test262 conformance test suite

- C API for embedding JerryScript

- Byte code snapshot feature
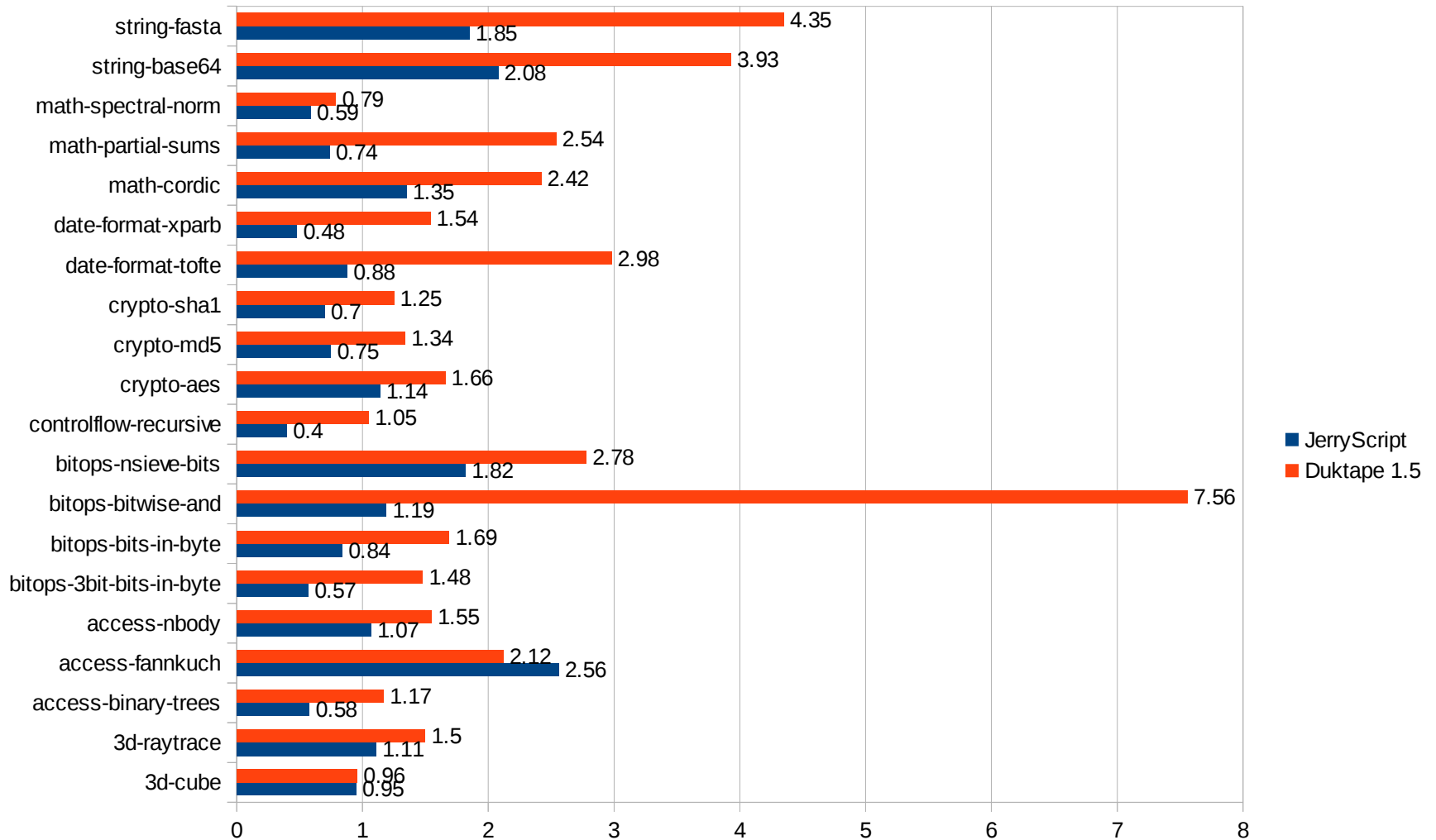
Memory consumption/ Performance

SunSpider 1.0.2 - Memory consumption

Benchmark	JerryScript	Duktape 1.5
string-fasta	28	107
string-base64	139	267
math-spectral-norm	12	102
math-partial-sums	5	101
math-cordic	6	102
date-format-xparb	21	147
date-format-tofte	21	352
crypto-sha1	92	169
crypto-md5	142	210
crypto-aes	57	180
controlflow-recursive	80	135
bitops-nsieve-bits	139	207
bitops-bitwise-and	4	97
bitops-bits-in-byte	5	97
bitops-3bit-bits-in-byte	5	98
access-nbody	11	106
access-fannkuch	12	101
access-binary-treesl	38	160
3d-raytrace	186	339
3d-cube	36	138

■ JerryScript
■ Duktape 1.5

Max RSS in KB (lower is better)

Measured on a Raspberry Pi 2

SunSpider 1.0.2 - Performance

Execution time in seconds (lower is better)

Measured on a Raspberry Pi 2 (-Os -flto)

Legend: ■ JerryScript ■ Duktape 1.5

Benchmark	JerryScript	Duktape 1.5
string-fasta	1.85	4.35
string-base64	2.08	3.93
math-spectral-norm	0.59	0.79
math-partial-sums	0.74	2.54
math-cordic	1.35	2.42
date-format-xparb	0.48	1.54
date-format-tofte	0.88	2.98
crypto-sha1	0.7	1.25
crypto-md5	0.75	1.34
crypto-aes	1.14	1.66
controlflow-recursive	0.4	1.05
bitops-nsieve-bits	1.82	2.78
bitops-bitwise-and	1.19	7.56
bitops-bits-in-byte	0.84	1.69
bitops-3bit-bits-in-byte	0.57	1.48
access-nbody	1.07	1.55
access-fannkuch	2.56	2.12
access-binary-trees	0.58	1.17
3d-raytrace	1.11	1.5
3d-cube	0.95	0.96

JerryScript on RIOT

JerryScript on RIOT
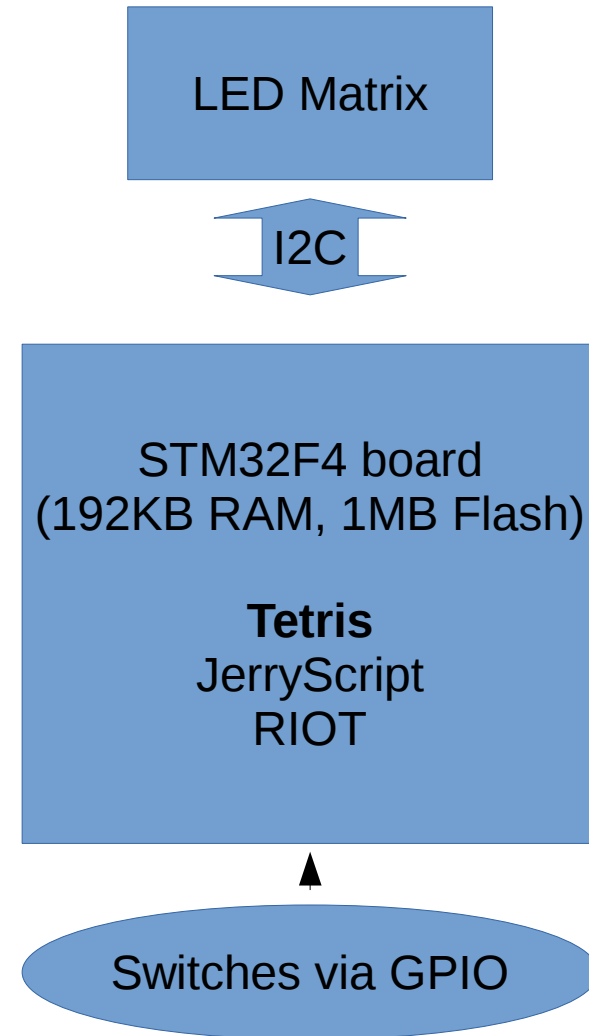
- Ported JerryScript over to RIOT

- Without any prior knowledge of RIOT the port took only a couple of days to complete

- Running RIOT on the STM32F4 board

- Ran into issue #4488 which causes crashes when writing to the serial console

- RIOT now officially supported as a target OS in upstream JerryScript

Demo
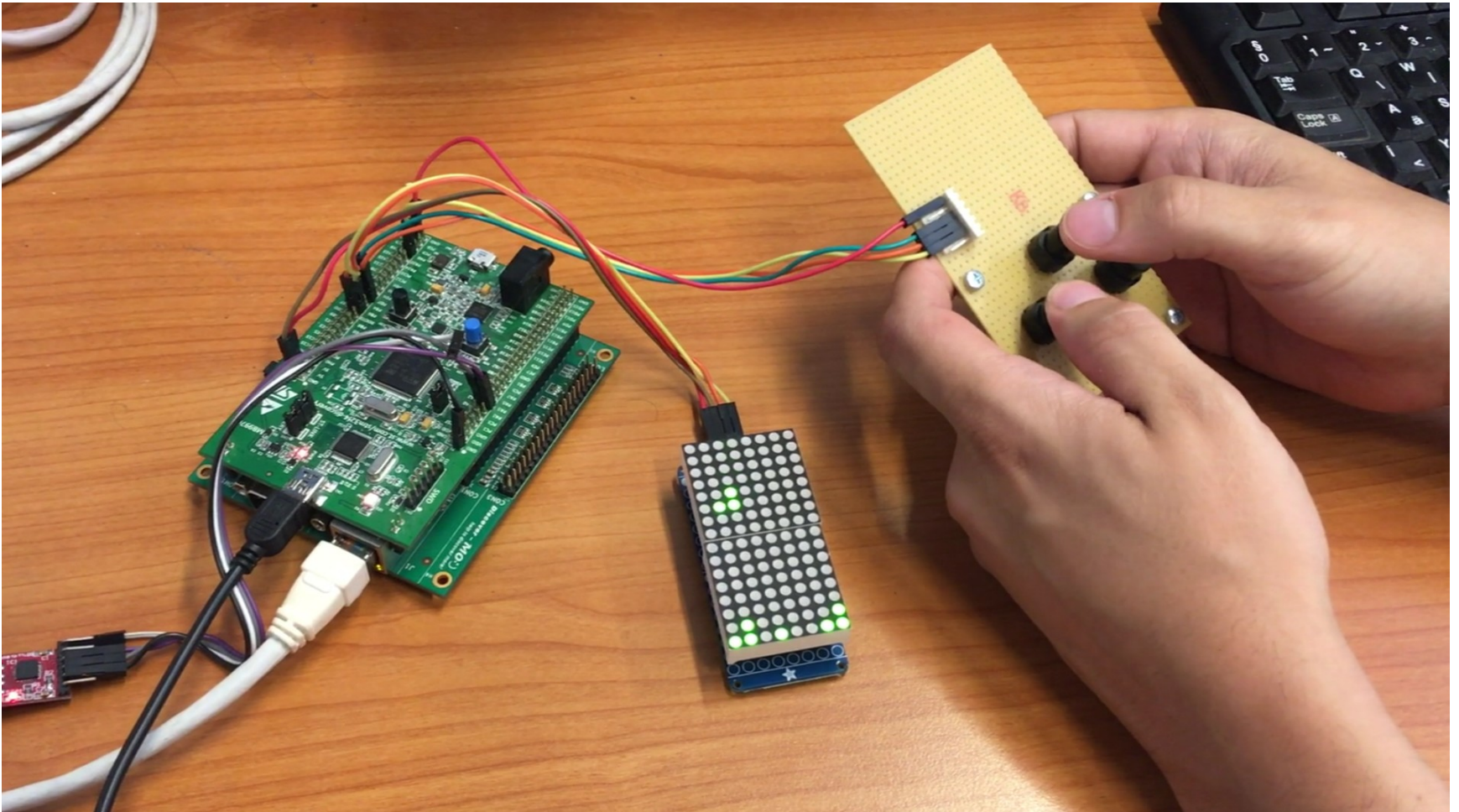
Tetris Demo

- Implementation of the classic Tetris game

- An LED matrix is used as a display

- Written in JavaScript

LED Matrix

I2C

STM32F4 board
(192KB RAM, 1MB Flash)

Tetris
JerryScript
RIOT

Switches via GPIO

Tetris Demo

Future work

Future work

- Further performance and memory optimizations

- 1.0 release of JerryScript in the coming weeks

- Debugging support

- Memory profiling

- Support for more boards
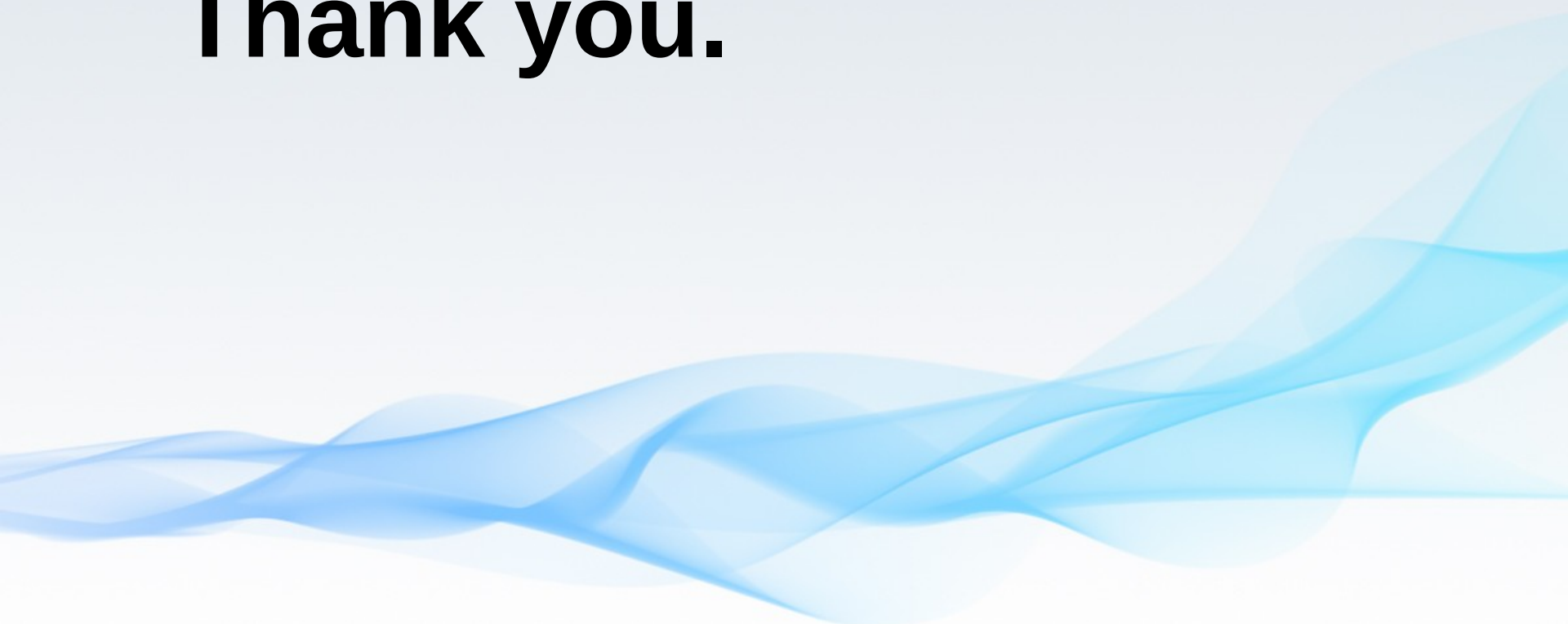
Summary

Summary

- Significantly lowers barrier of entry for JavaScript development targeting heavily constrained embedded devices

- Speeds up development

- Active community

- More information on http://samsung.github.io/jerryscript

- Looking for bug reports and feedback

Thank you.

Contact Information:

Tilmann Scheller
t.scheller@samsung.com

Samsung Open Source Group
Samsung Research UK