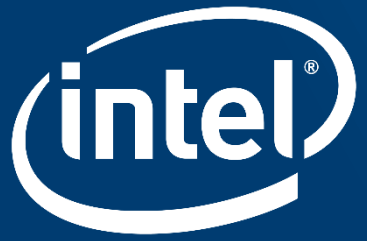


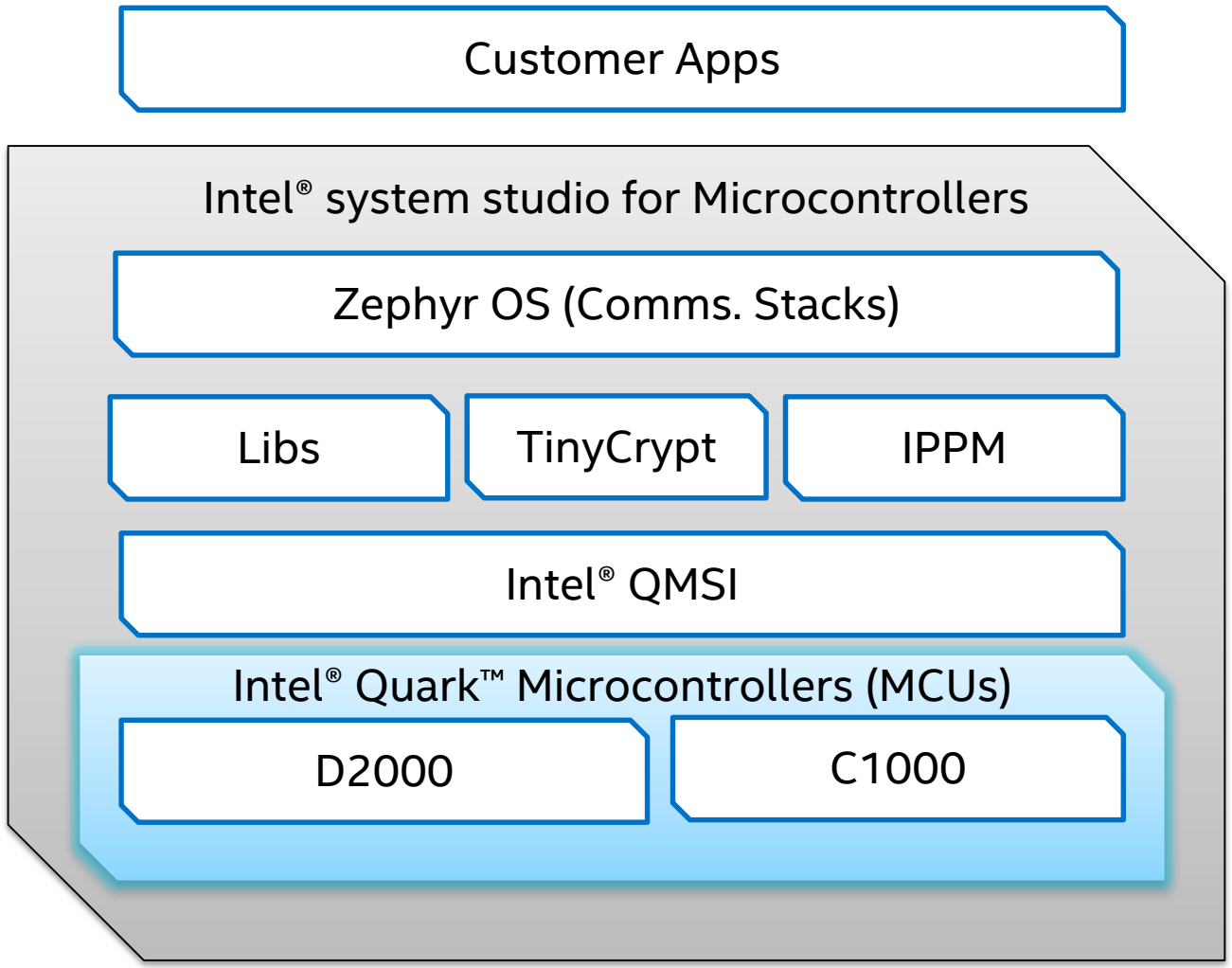
Intel® Quark™ Microcontroller Software Interface (Intel® QMSI)

Malcolm Prinn,

October 2016

Intel® QMSI quick facts

- Hardware Abstraction Layer written in C
- ~100k LoC
- Supports most Intel® Quark™ MCU SoCs
- Small code size / stack usage
- Low power functionality



Intel® Quark™ MCU family

The new Intel® Quark™ microcontroller D1000, Intel® Quark™ microcontroller D2000, and Intel® Quark™ SE microcontroller C1000 for IoT extend Intel's product roadmap to the very edge of the Internet of Things (IoT), enabling a consistent architecture from things to the cloud, with a broad portfolio of Intel products spanning from Intel® Quark™ to Intel® Xeon® processors.

Low Power

Optimized for low power consumption, such as battery-powered applications

Integrated Security

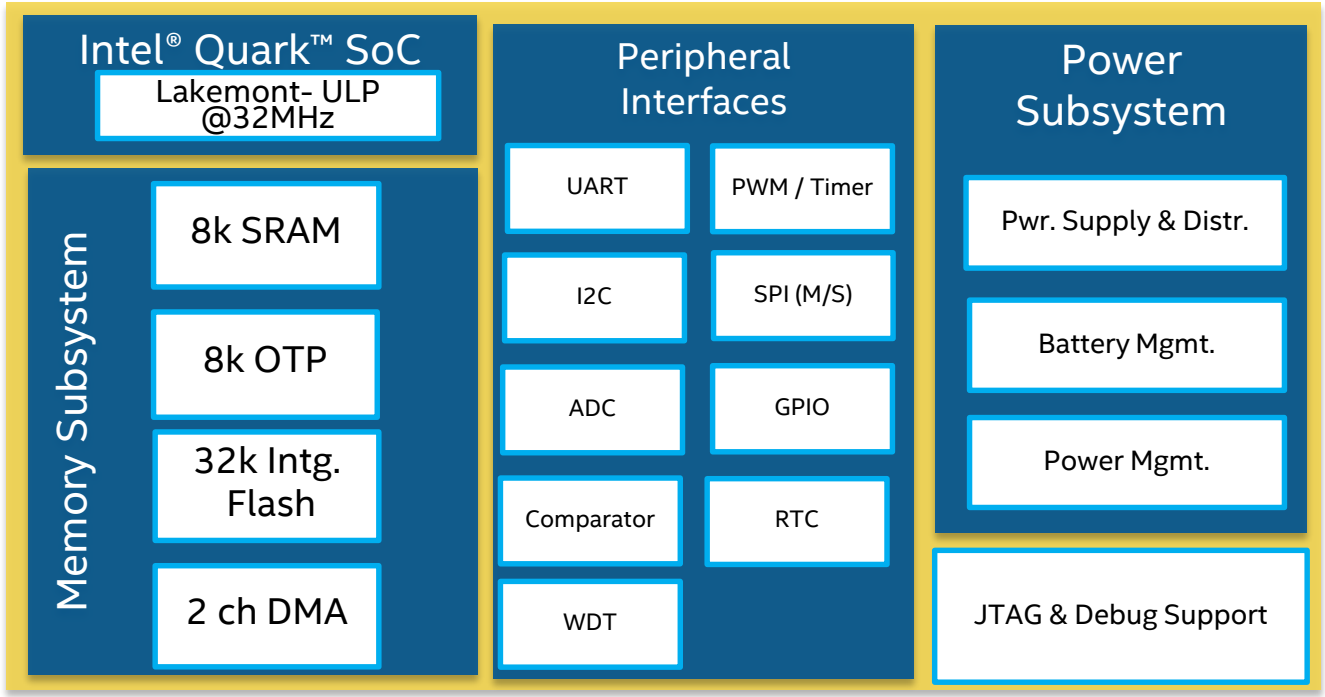
With manageability and connectivity to help protect your data at every endpoint

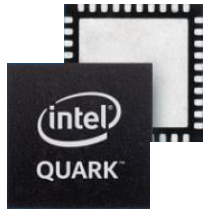
Scalable Architecture

Maximize investment by reusing software to scale up and down to any Intel processor

INTEL® QUARK™ MICROCONTROLLER D2000

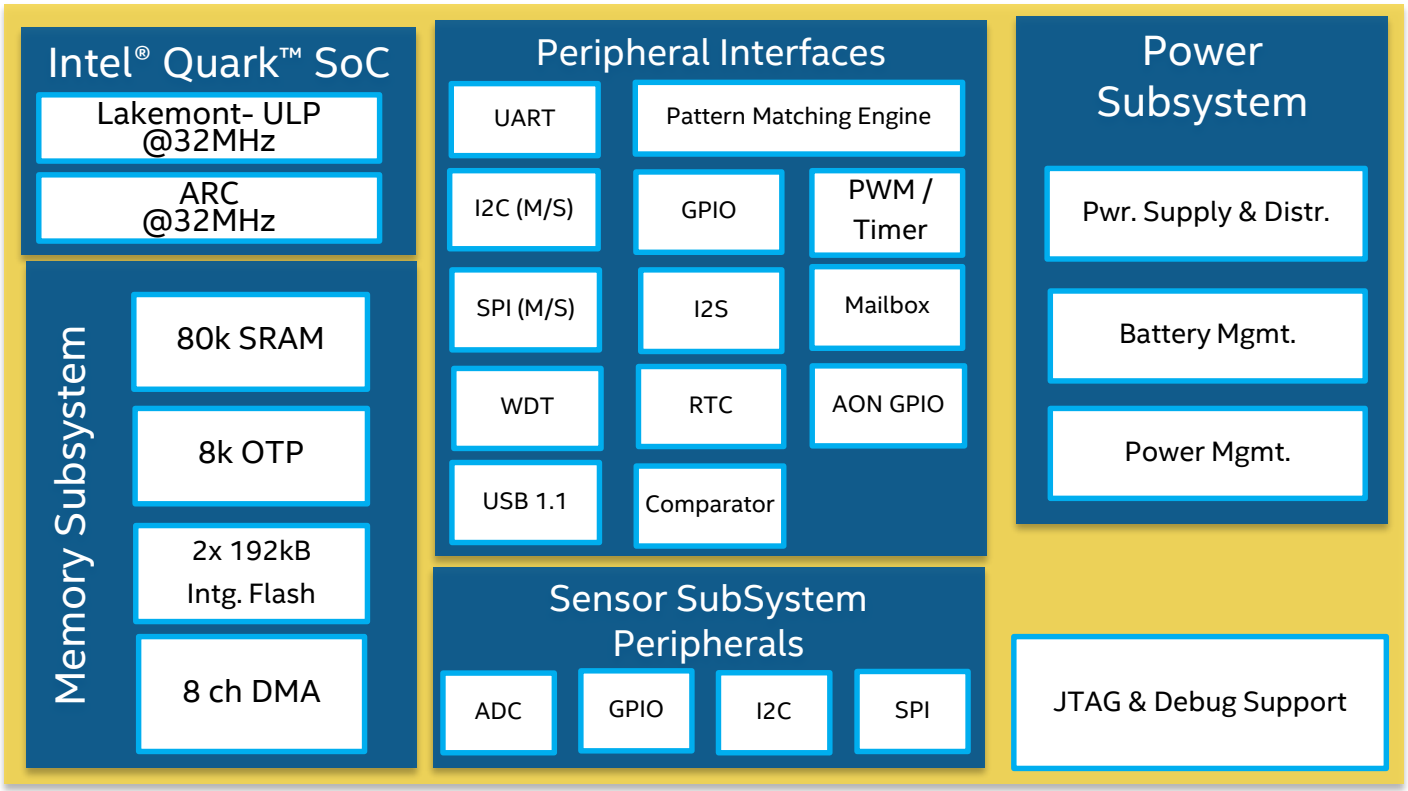
- Ultra low power, Entry Level
- 32MHz, 32-bit x86 Microcontroller, 32kB Flash, 8kB SRAM
- Scalable Software Development Kit, with sample apps and libraries
- Pre-validated comms and sensor modules
- Full Intel® x86 instruction set architecture for compatibility and scalability

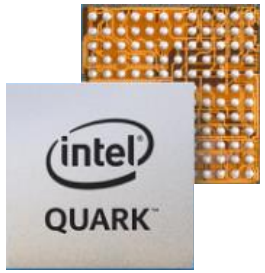


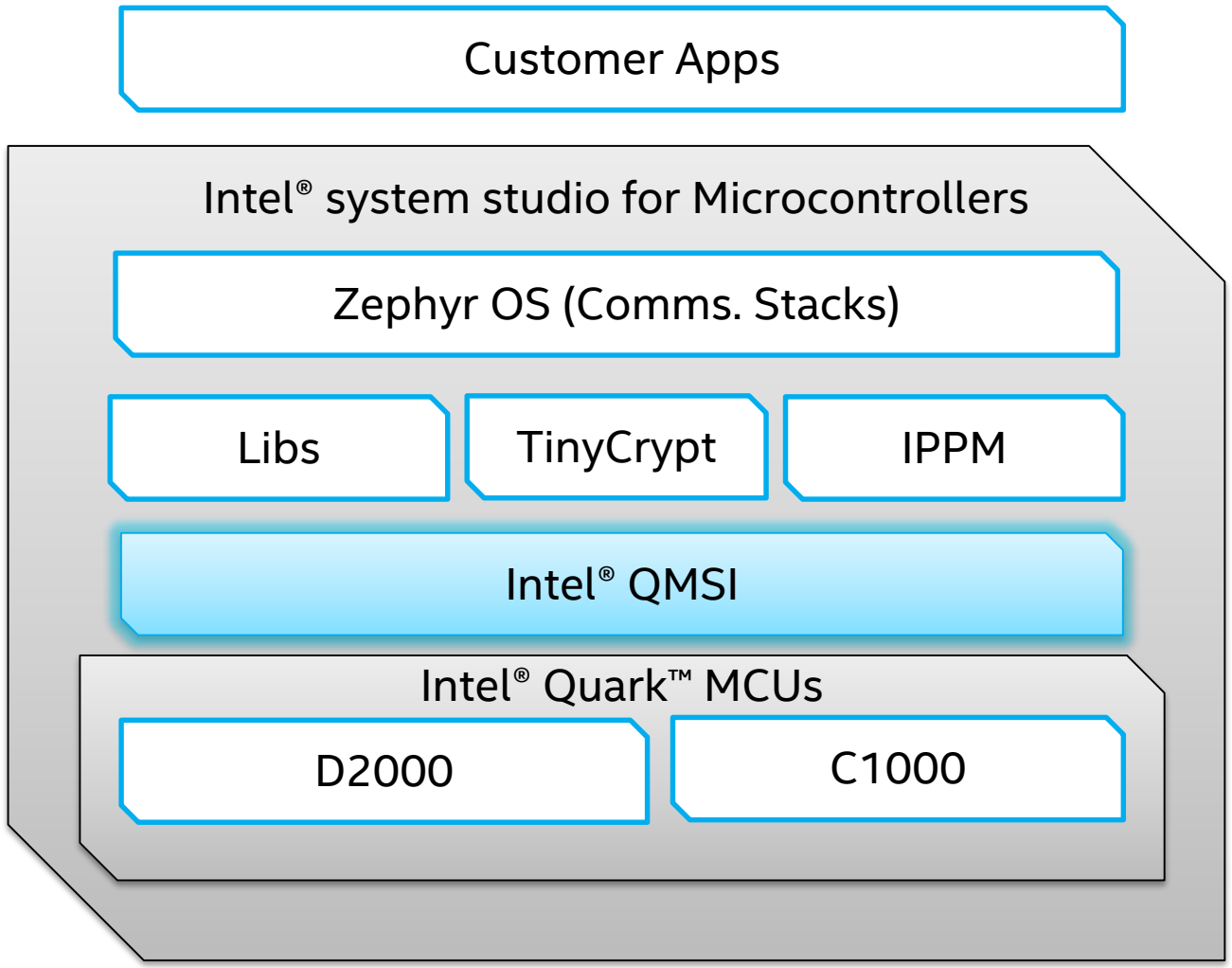


INTEL® QUARK™ SE MICROCONTROLLER C1000

- High-efficiency power consumption
- 32MHz, 32-bit x86 Microcontroller
- 384Kb Flash, 80kB SRAM
- Scalable Software Development Kit, with sample apps and libraries
- Pre-validated comms and sensor modules
- Full Intel® x86 instruction set architecture for compatibility and scalability
- Always sensing: always-listening Internal Sensor Hub
- Intelligent: Pattern Matching Engine

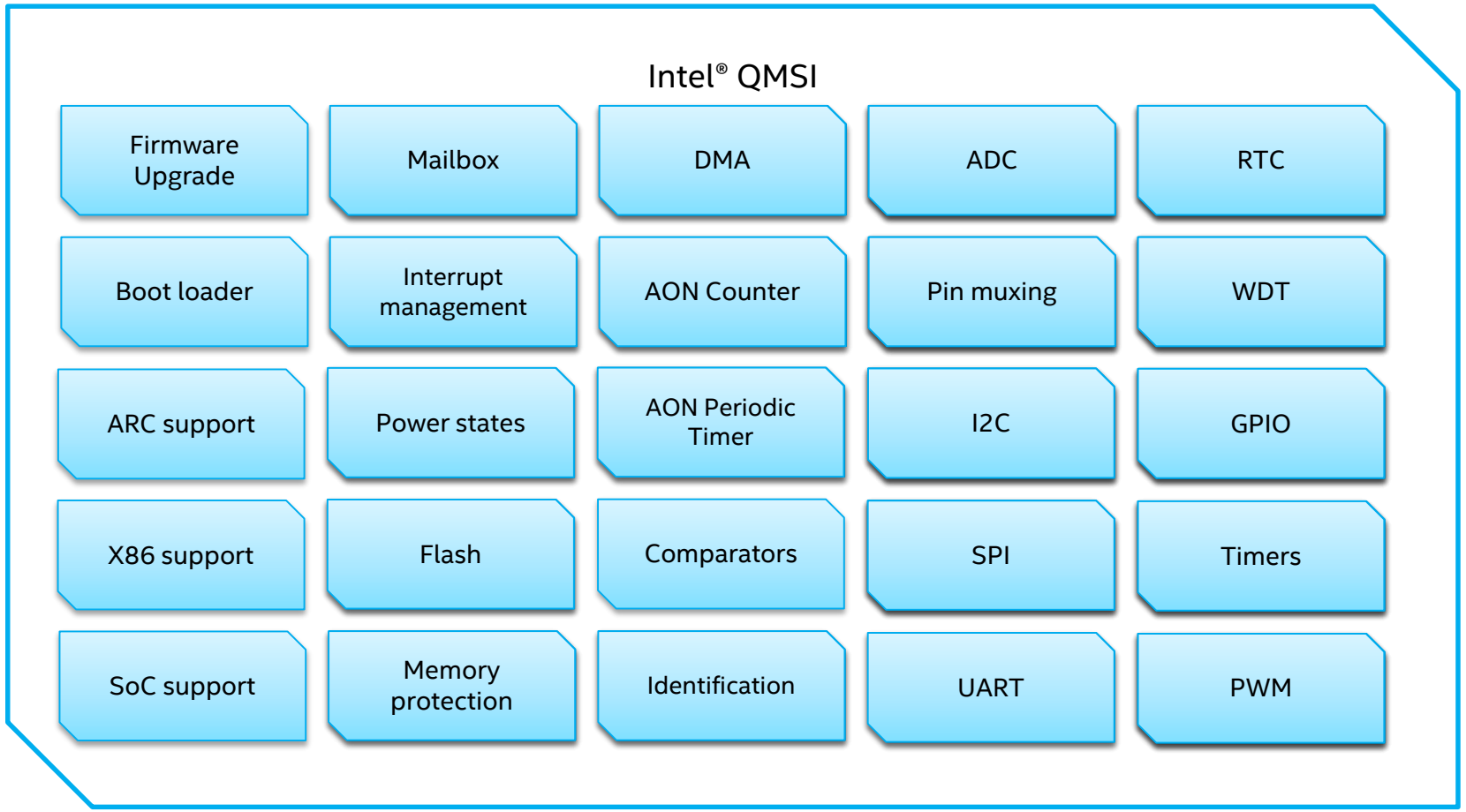






Intel® Quark™ Microcontroller Software Interface

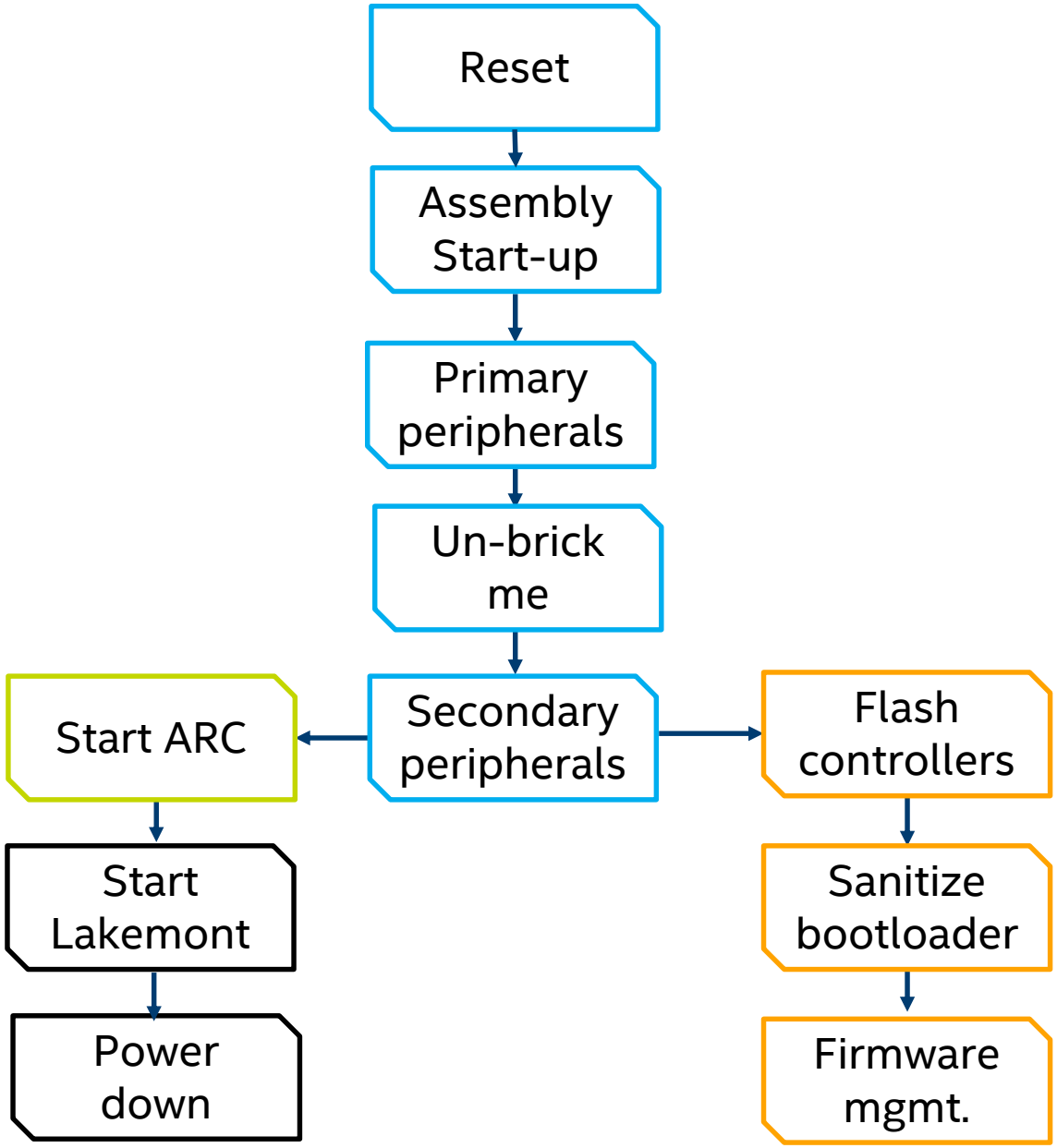
- Intel® Quark™ Microcontrollers Hardware Abstraction Layer (HAL)
- APIs provide a standard interface to all functionality in Intel® Quark™ microcontrollers.
- APIs are consistent across the Intel® Quark™ microcontroller family of devices.
- Included with this API are a collection of sample applications to enable users to get started quickly.
- Embedded devices with a limited amount of memory,
  - Object code size & stack size minimization is a primary design objective.
  - Everything accessed as MMIO.
  - Nothing is dynamic, all the heavy work is done at compile time.
  - Garbage collection on unused functionality.



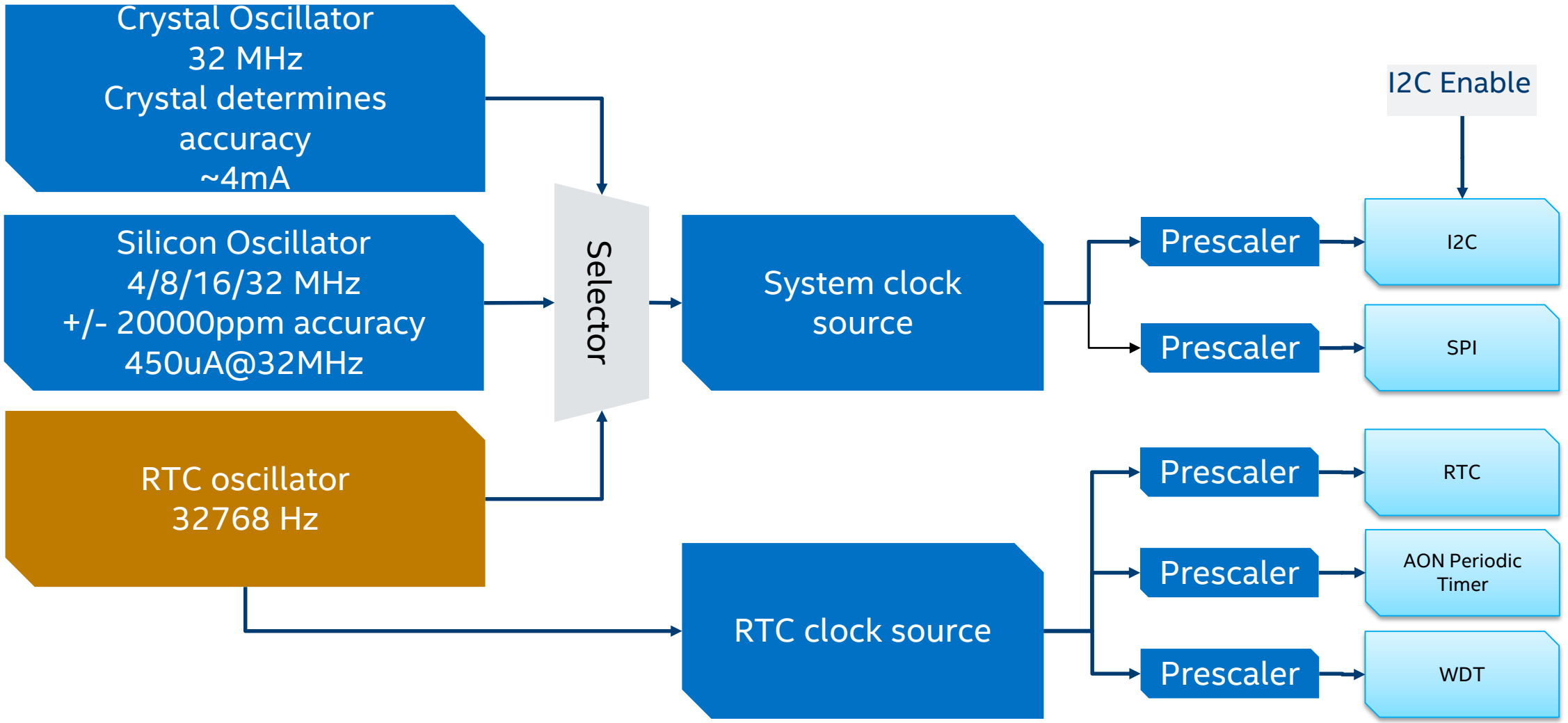
Bootloader / ROM

- Bootstrap
- TRIM code calculation
- Firmware update management
  - Host tools also provided (fork of DFU utils)
- Un-brick me

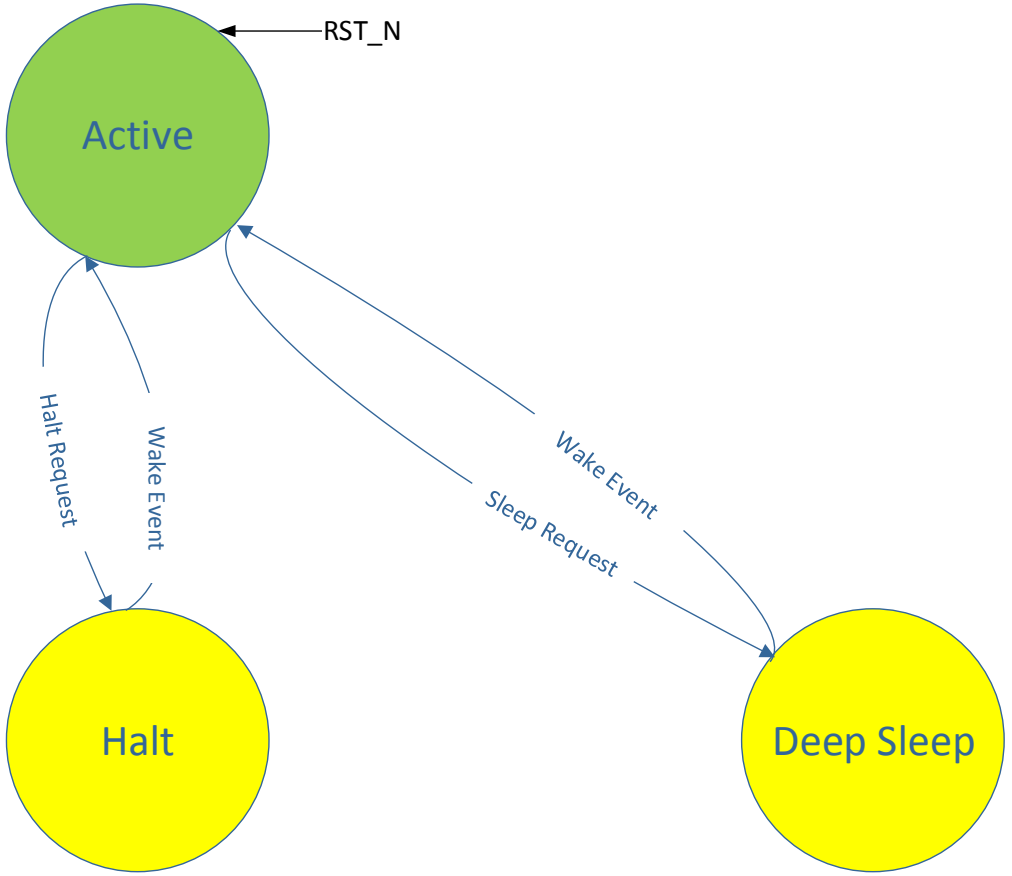
Boot flow



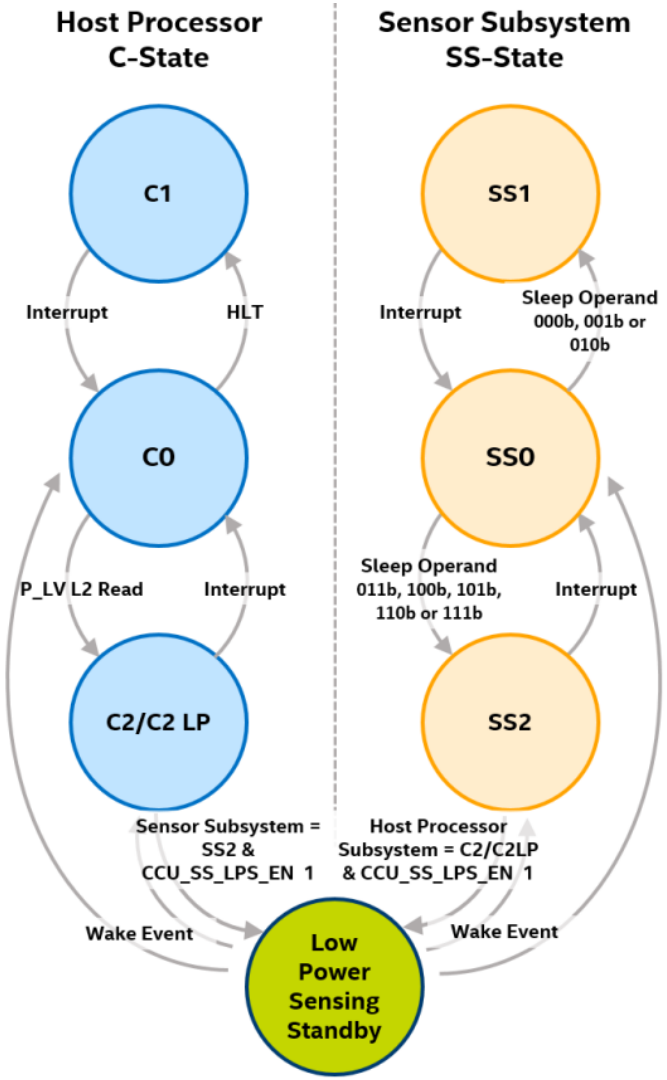
Clocking



Power states - D2000



Power states - C1000



Sample Real Time Clock driver usage

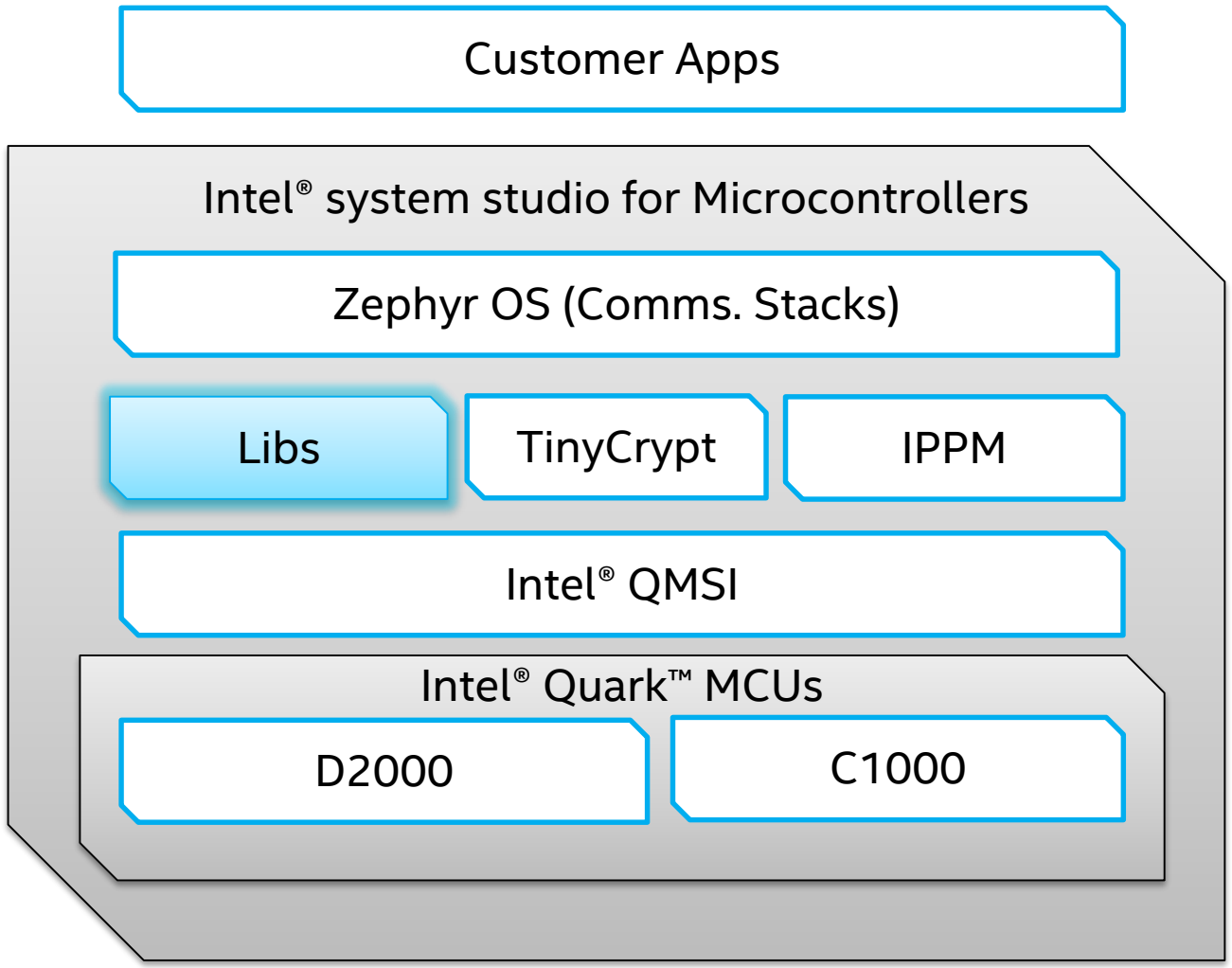
```
qm_rtc_config_t cfg;

/* Configure RTC and request the interrupt. */
cfg.init_val = 0;
cfg.alarm_en = true;
cfg.alarm_val = ALARM_INTERVAL;
cfg.callback = rtc_example_callback;
cfg.callback_data = NULL;
cfg.prescaler = CLK_RTC_DIV_1;

qm_irq_request(QM_IRQ_RTC_0, qm_rtc_isr_0);

/* Enable RTC. */
clk_periph_enable(CLK_PERIPH_RTC_REGISTER);

/* RTC actually starts here. */
qm_rtc_set_config(QM_RTC_0, &cfg);
```

RTC configuration structure

Register / request interrupt

Enable clocking to RTC

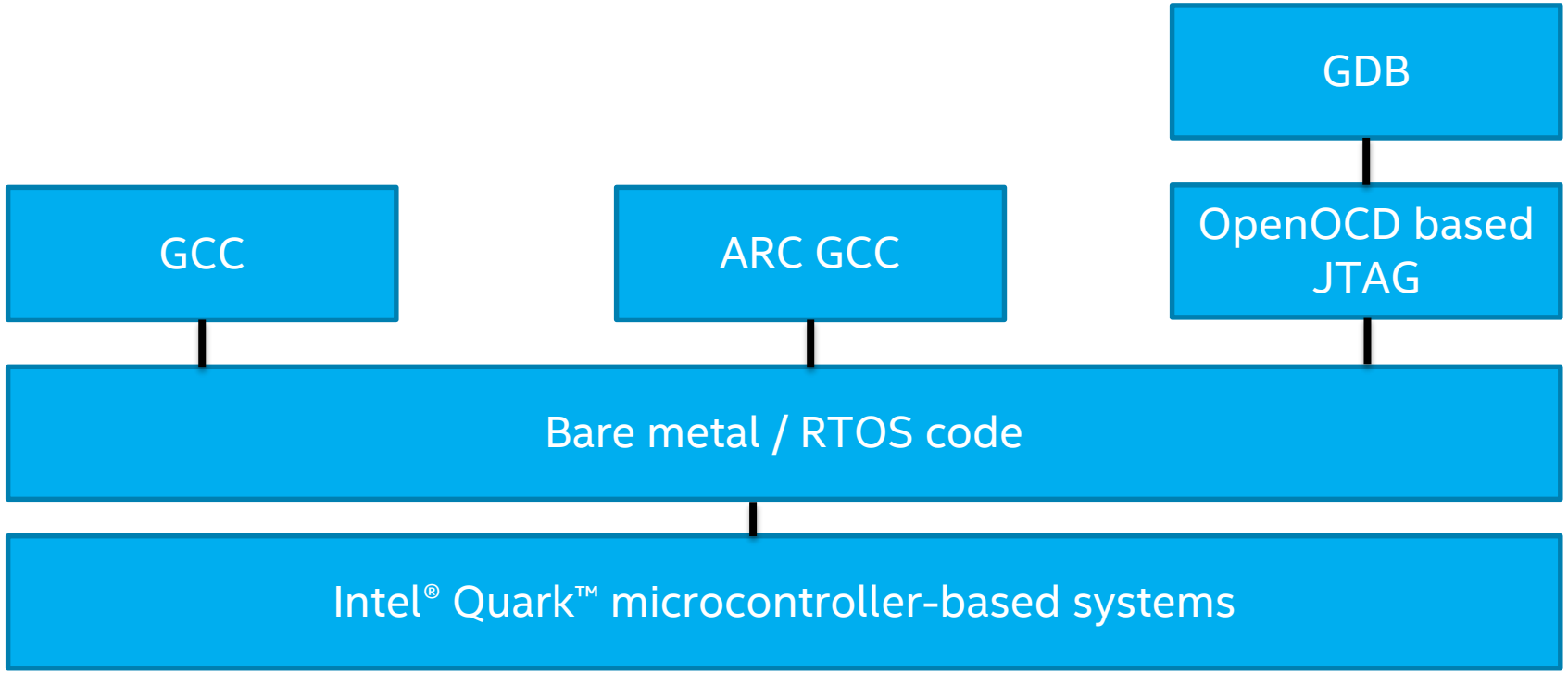
Set configuration and start RTC



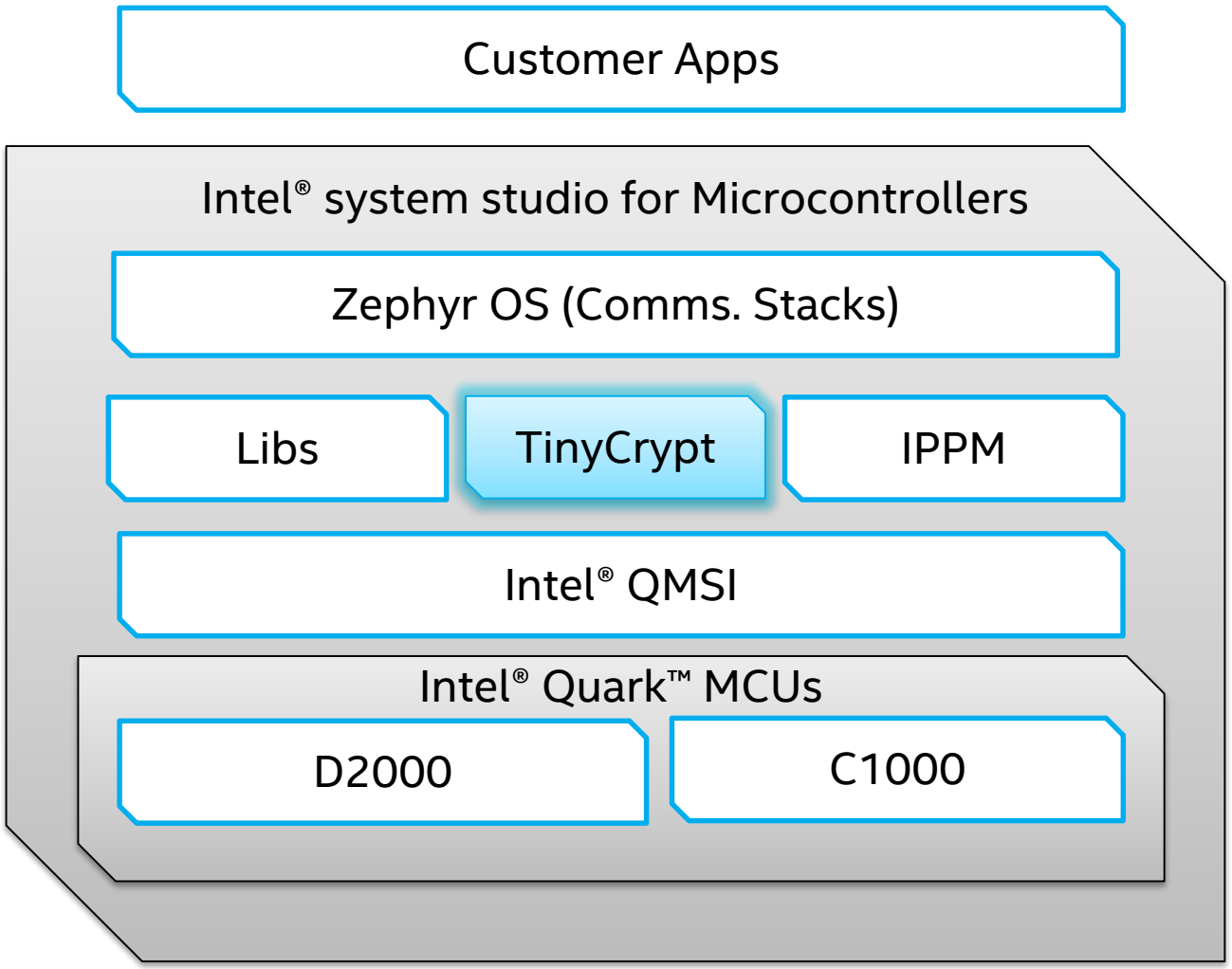
Newlib syscalls

- Pico printf
  - Modular, support for formats can be disabled at compile time
  - Supports ‘d’, ‘u’, ‘x’, ‘X’ and ‘s’
- Puts
- Malloc / free
- Assert

Toolchain

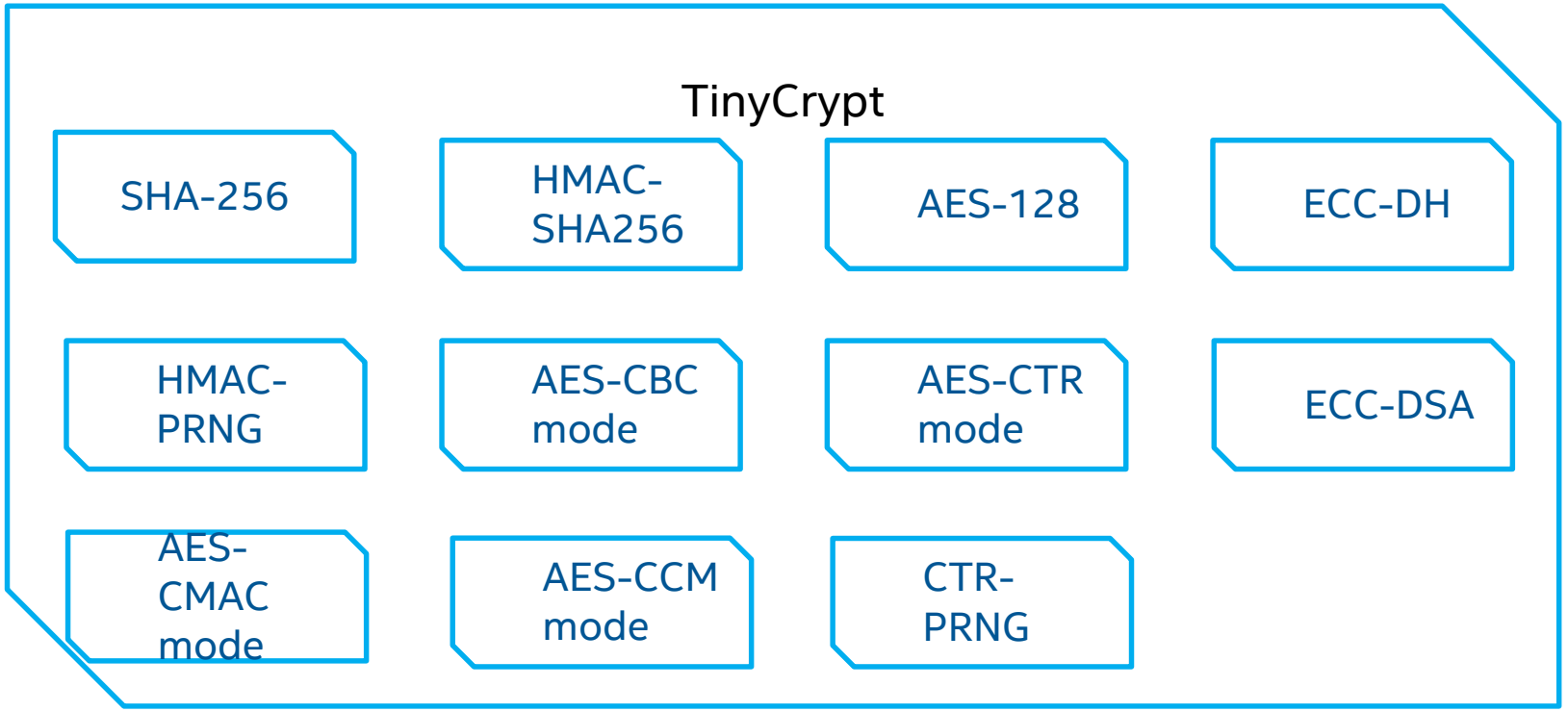


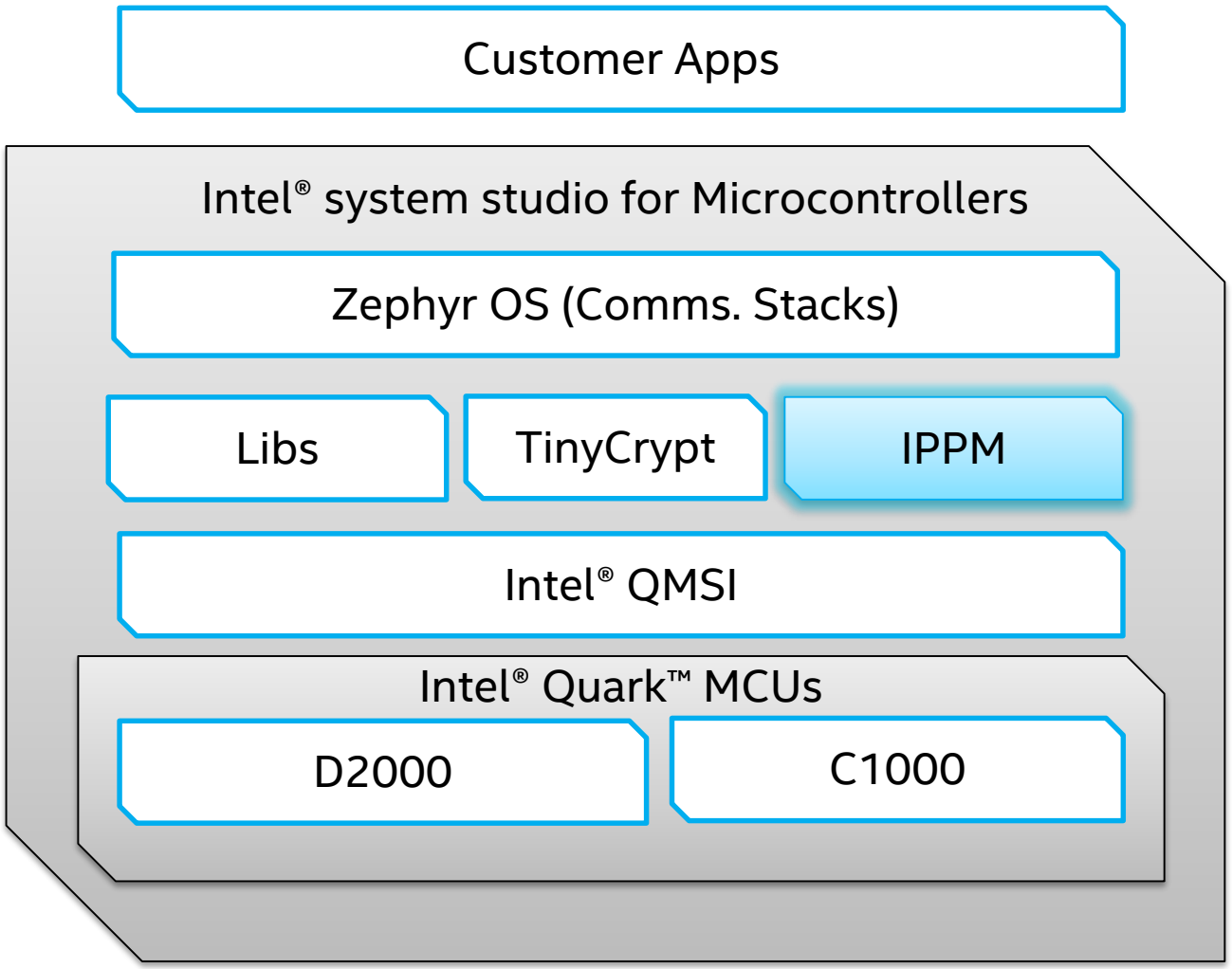
https://software.intel.com/en-us/articles/issm-toolchain-only-download



TinyCrypt Library

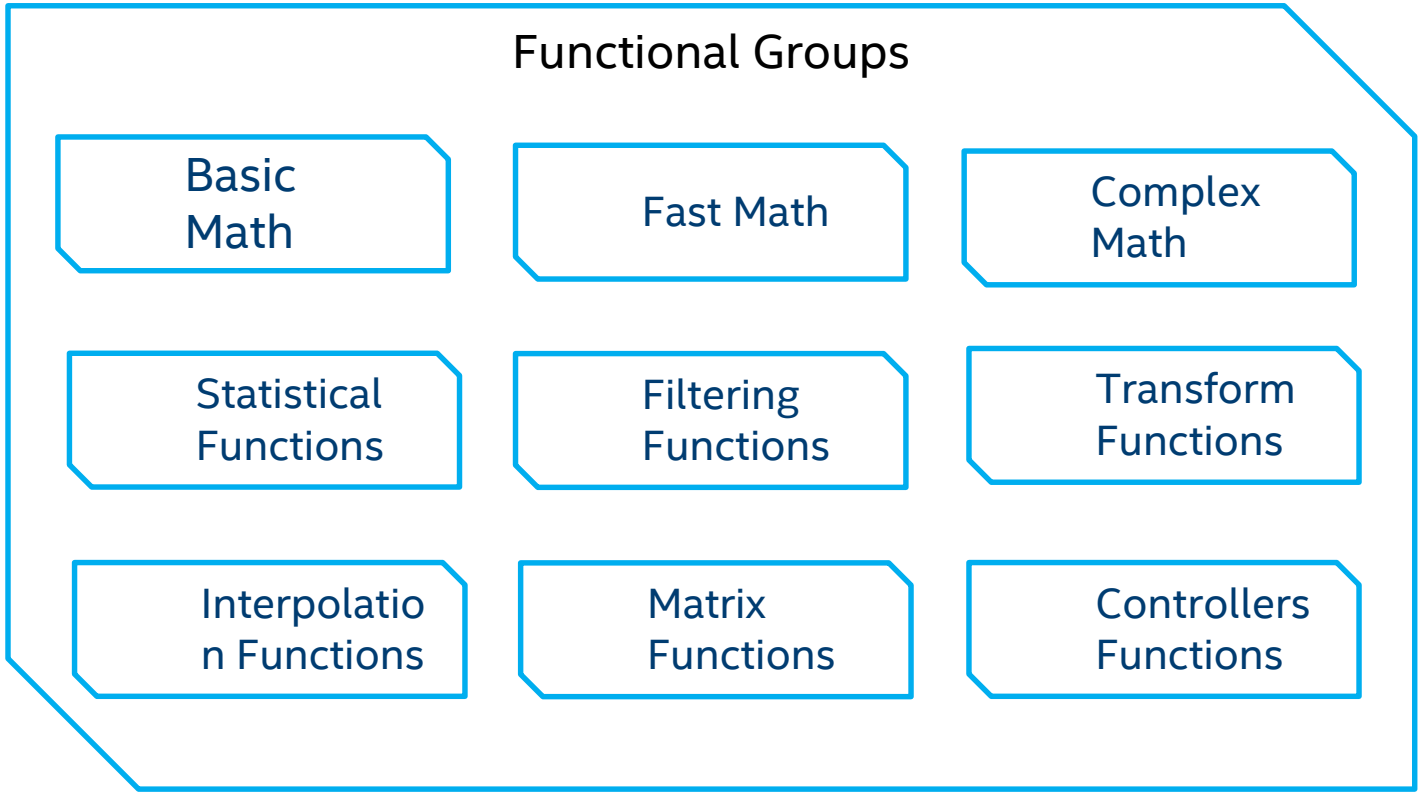
- The TinyCrypt Library provides an implementation for constrained devices of a minimal set of standard cryptography primitives.





Intel® Performance Primitives for Microcontrollers

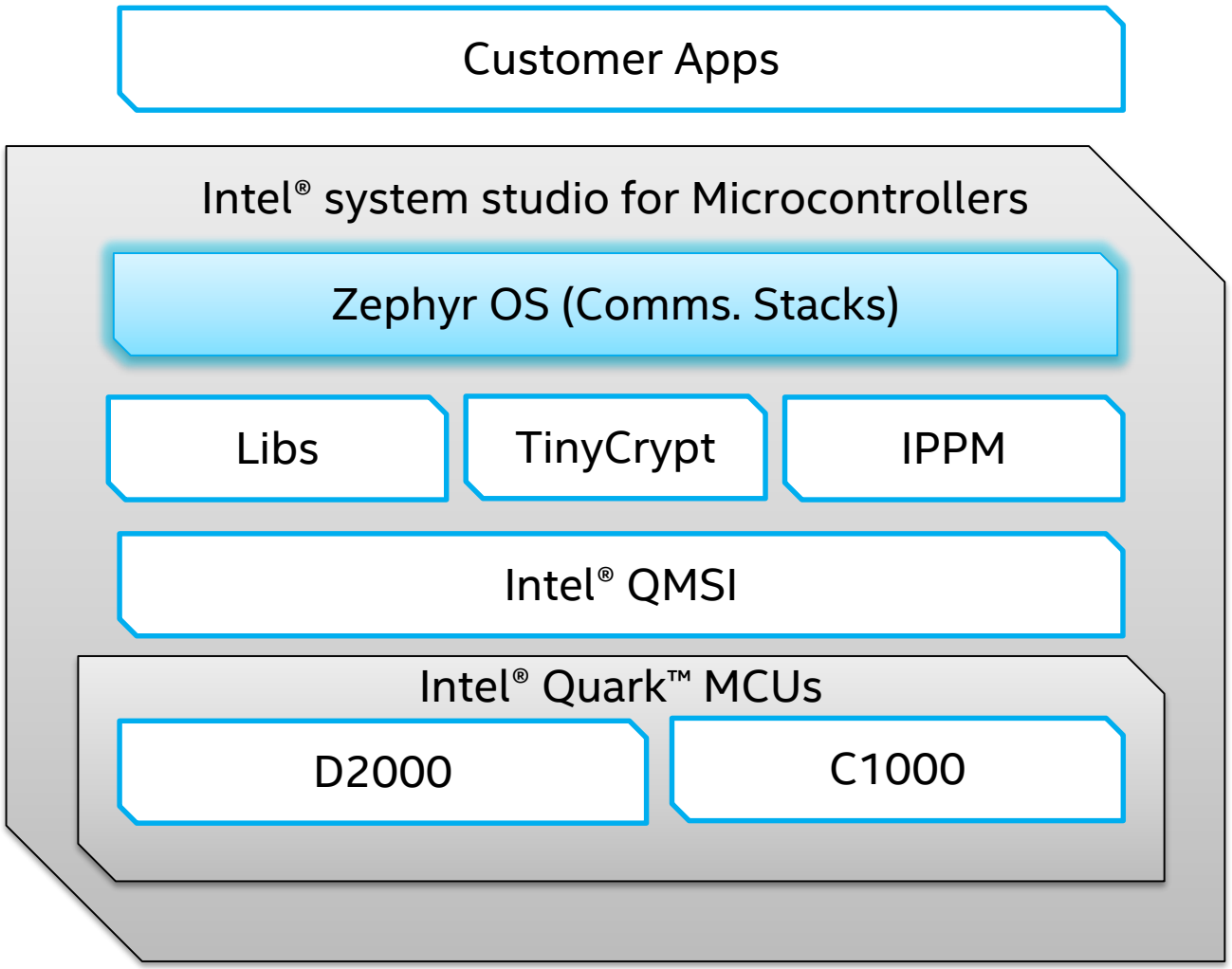
- Supported data types
  - Fixed point: q15, q31
  - Floating point: 32f (using floating point simulation)
- Optimization criteria
  - Size (~1 KB per function)
  - Performance
  - Accuracy
  - Power consumption



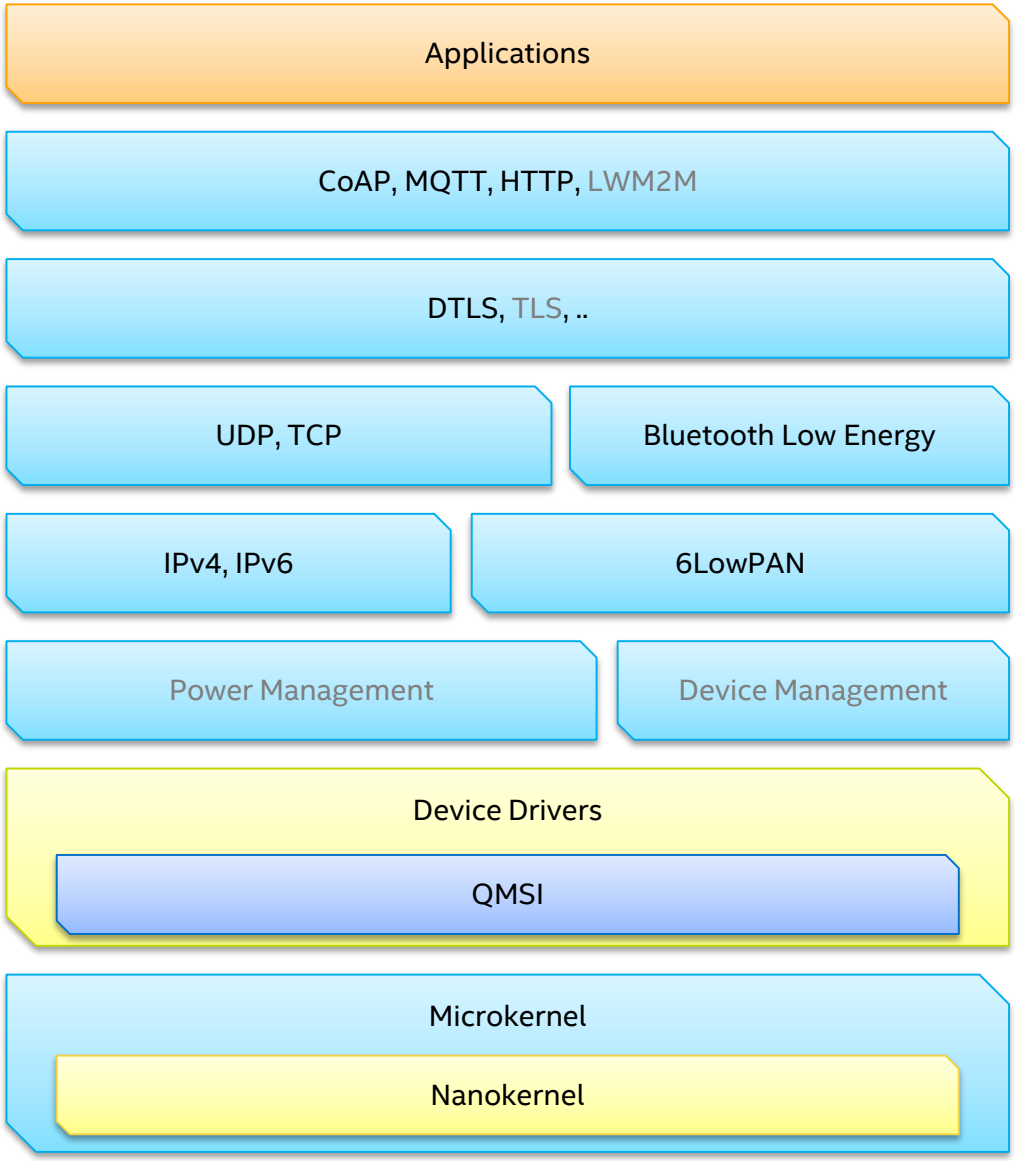
Floating Point Library

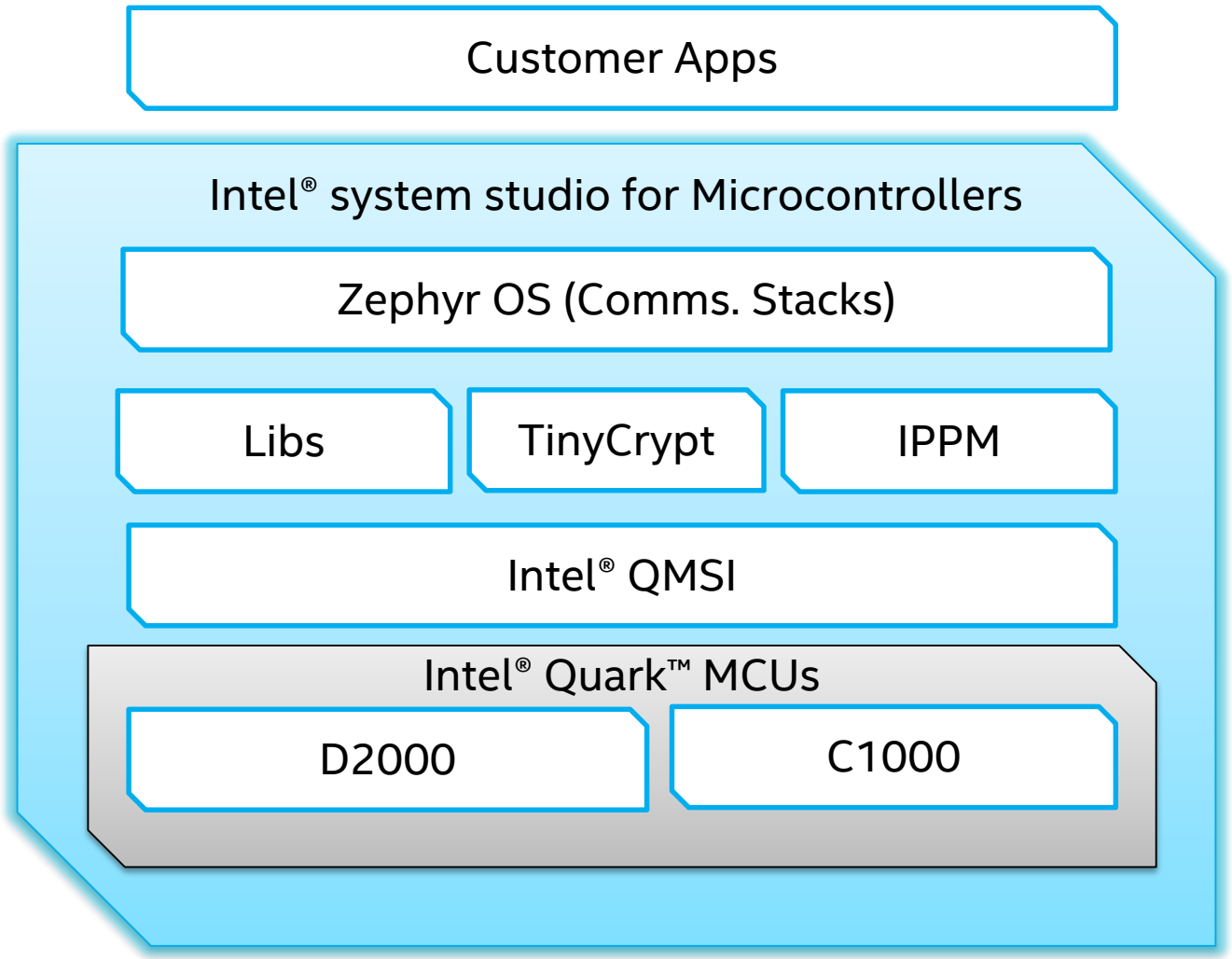
- Only applicable to Intel® Quark™ Lakemont processor cores

- Floating Point
  - The Floating Point Library emulates basic floating point operations with hardware integer instructions.
  - Compatible with the Intel® MCU Architecture ; supports Intel® Pentium® processor instruction set minus instructions for x87 floating point unit.
- Fixed point data format
  - Intel® IPP for Microcontrollers functions operate on fixed-point data in Q*n* format.
  - Example : *X* in the Q15 format is $X*2^{-15}$, with the range of supported values for the Ipp16s data type equal to $[-1, 1-2^{-15}]$.



Zephyr

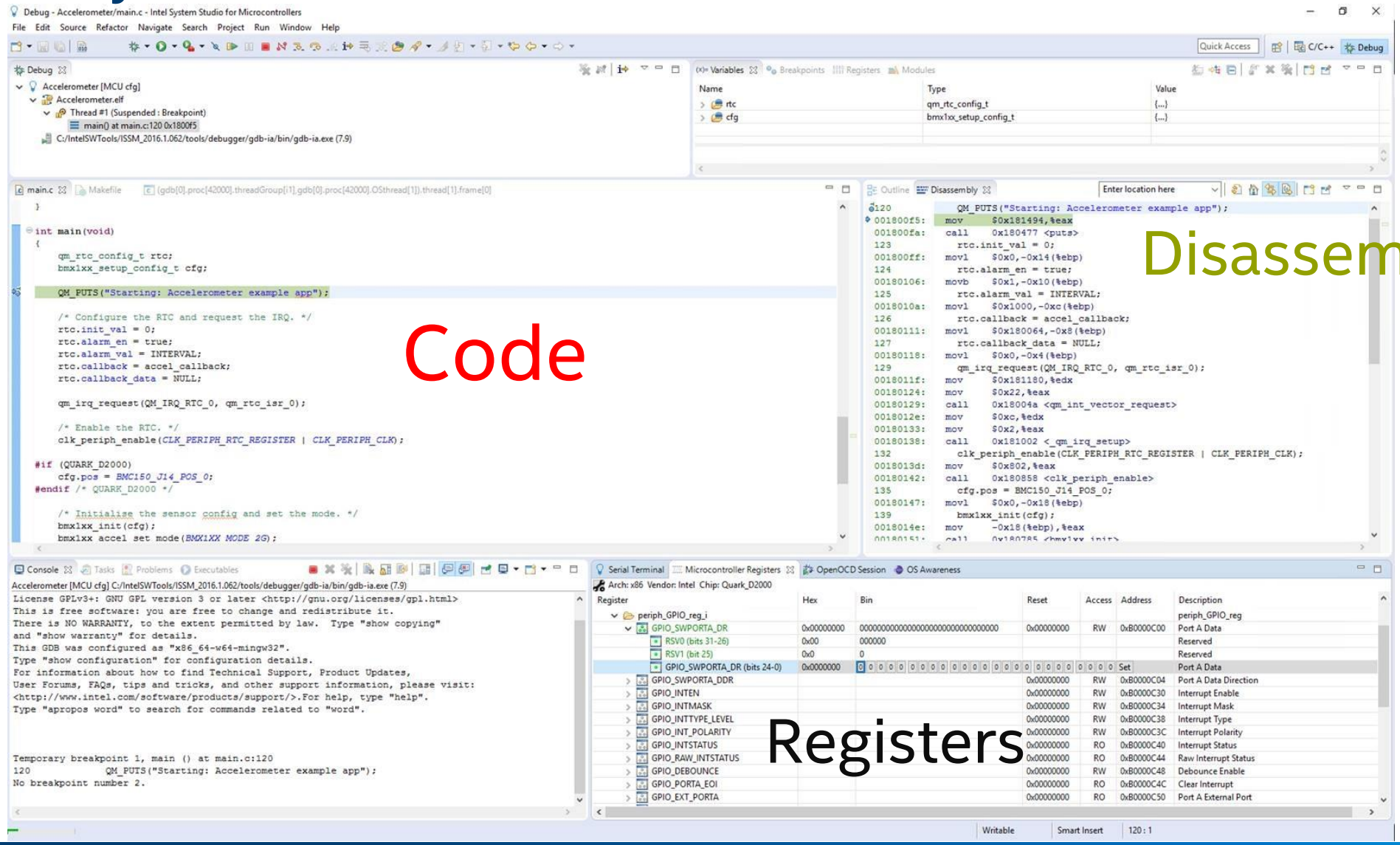


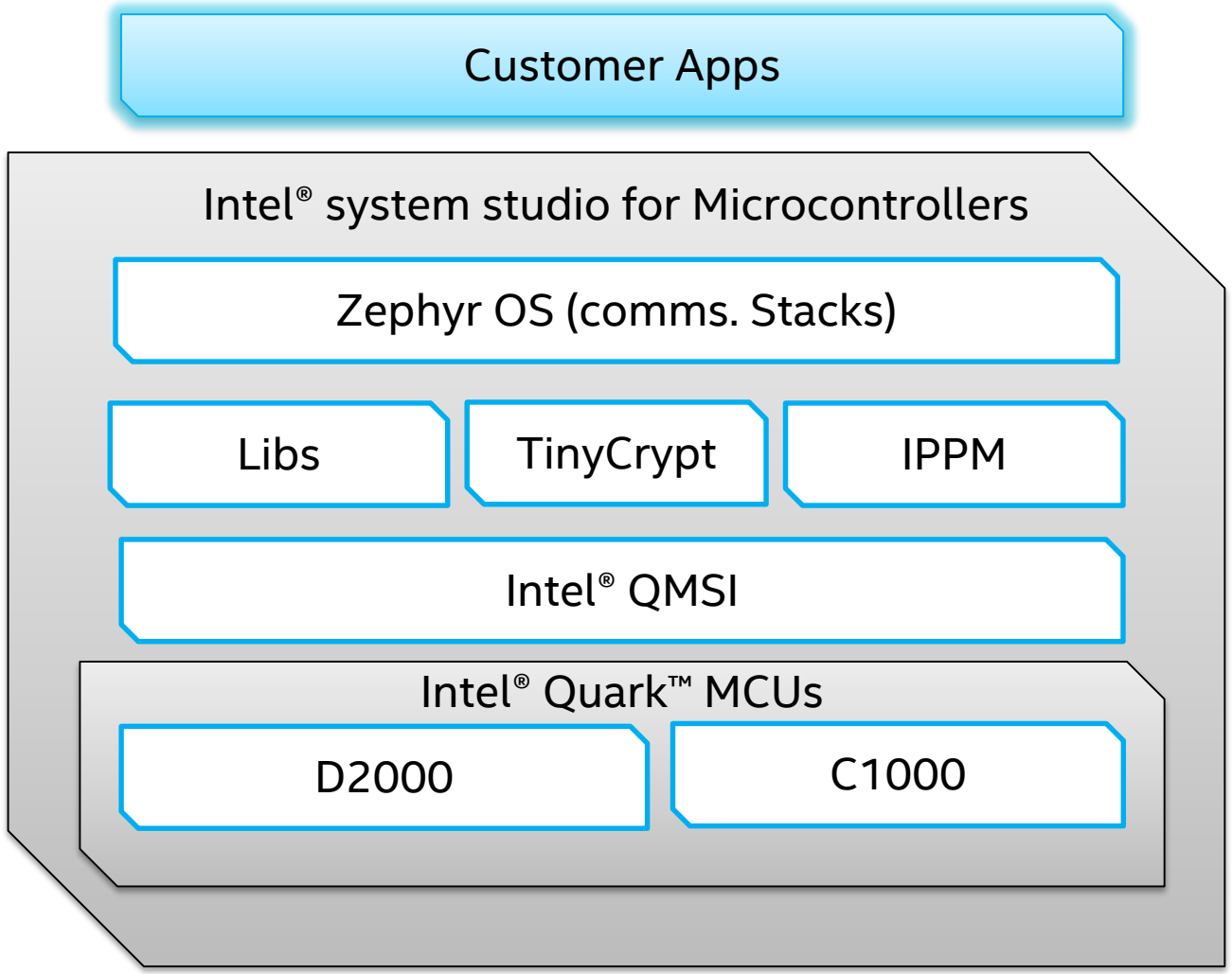


Intel® System Studio for Microcontrollers

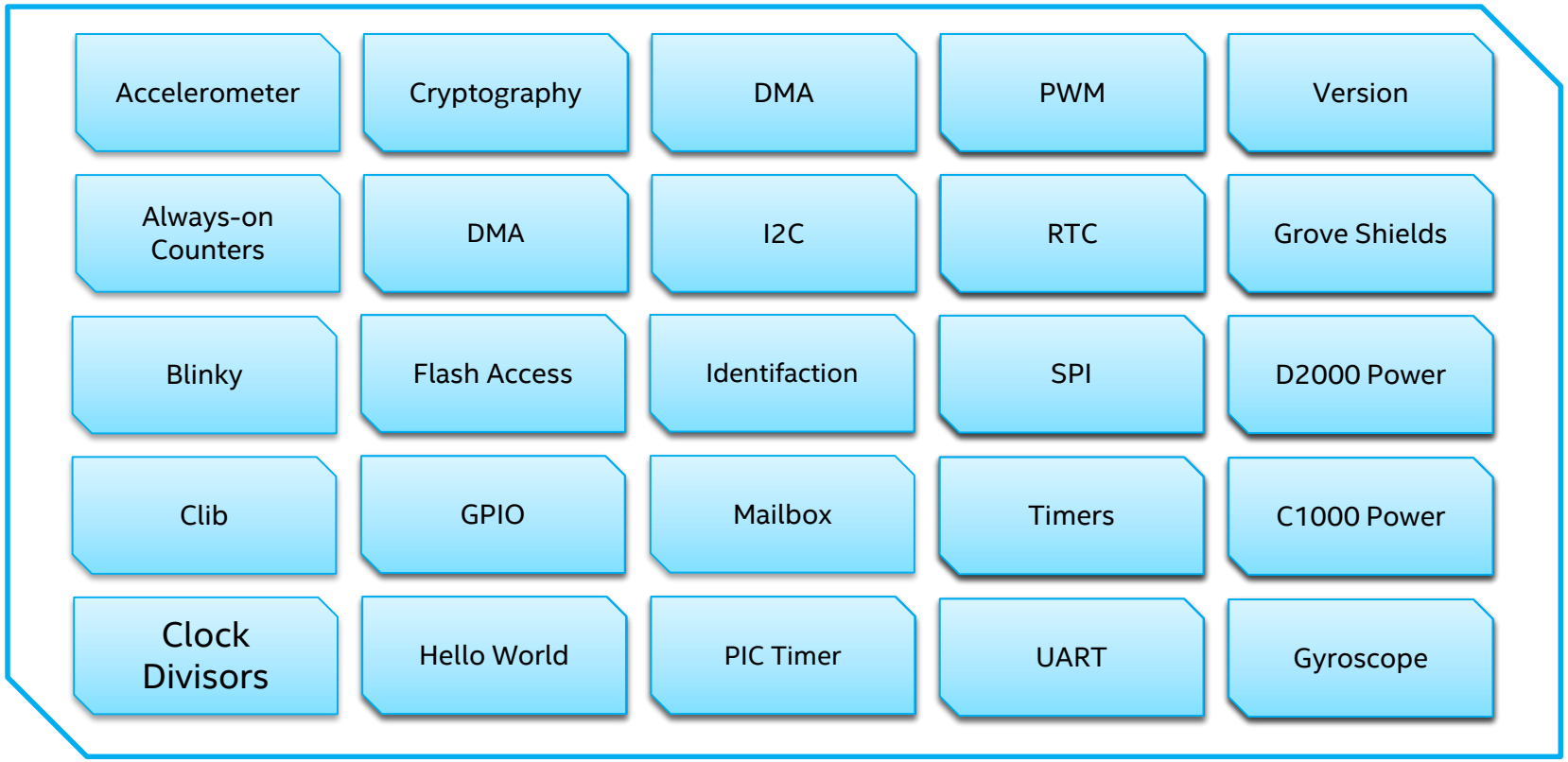
- Bundles components into a Software Development Kit (SDK)
- Integrated Development Environment (IDE)
  - Based on Eclipse
- Intel® System Studio for Microcontrollers plugins
- SoC debugger integration
- Windows USB Driver for:
  - Intel® Quark™ Microcontroller D2000 Development Board
  - Intel® Quark™ SE C1000 Development Board

Intel® system studio for Microcontrollers in action





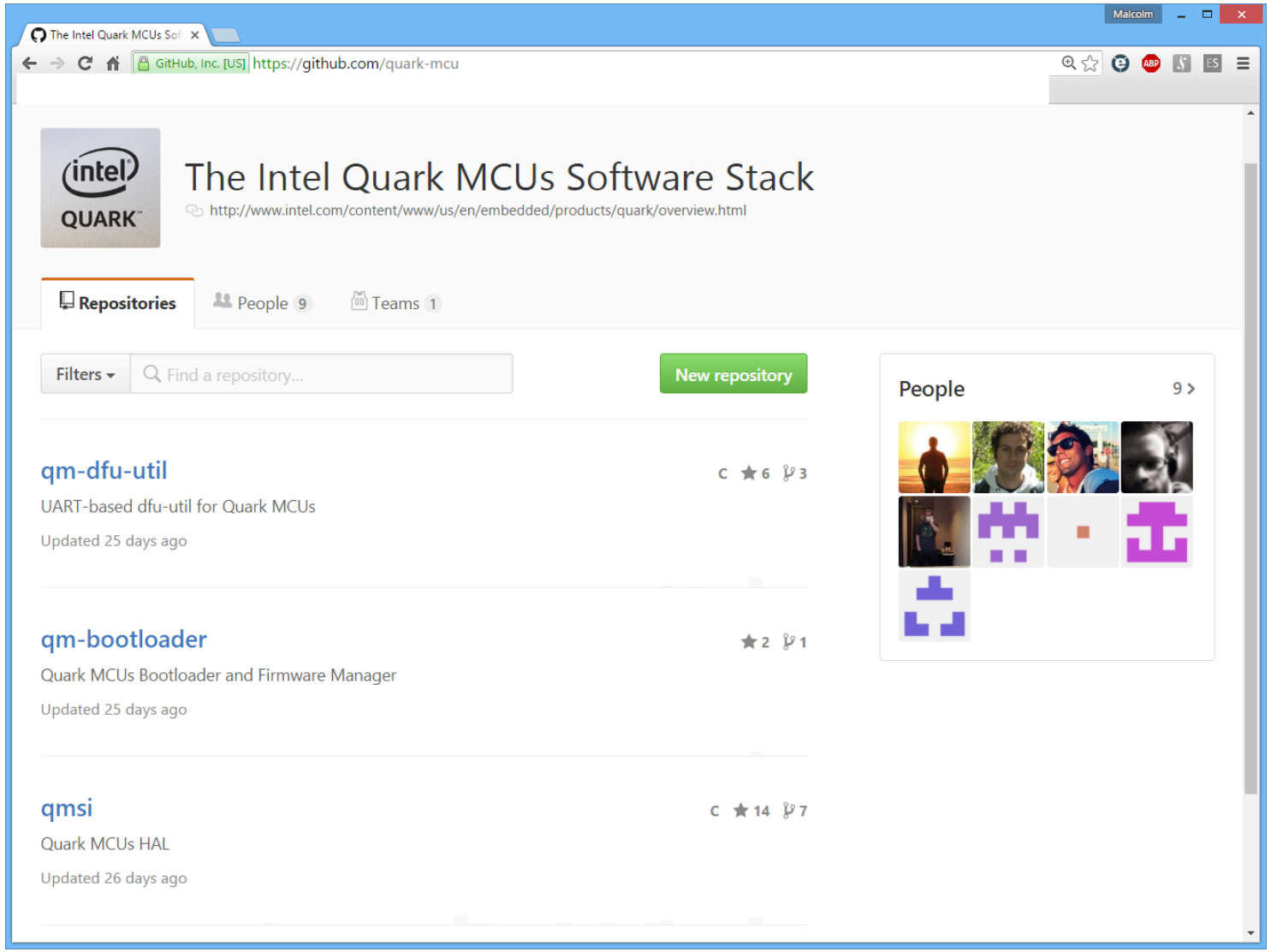
Build upon already existing example applications



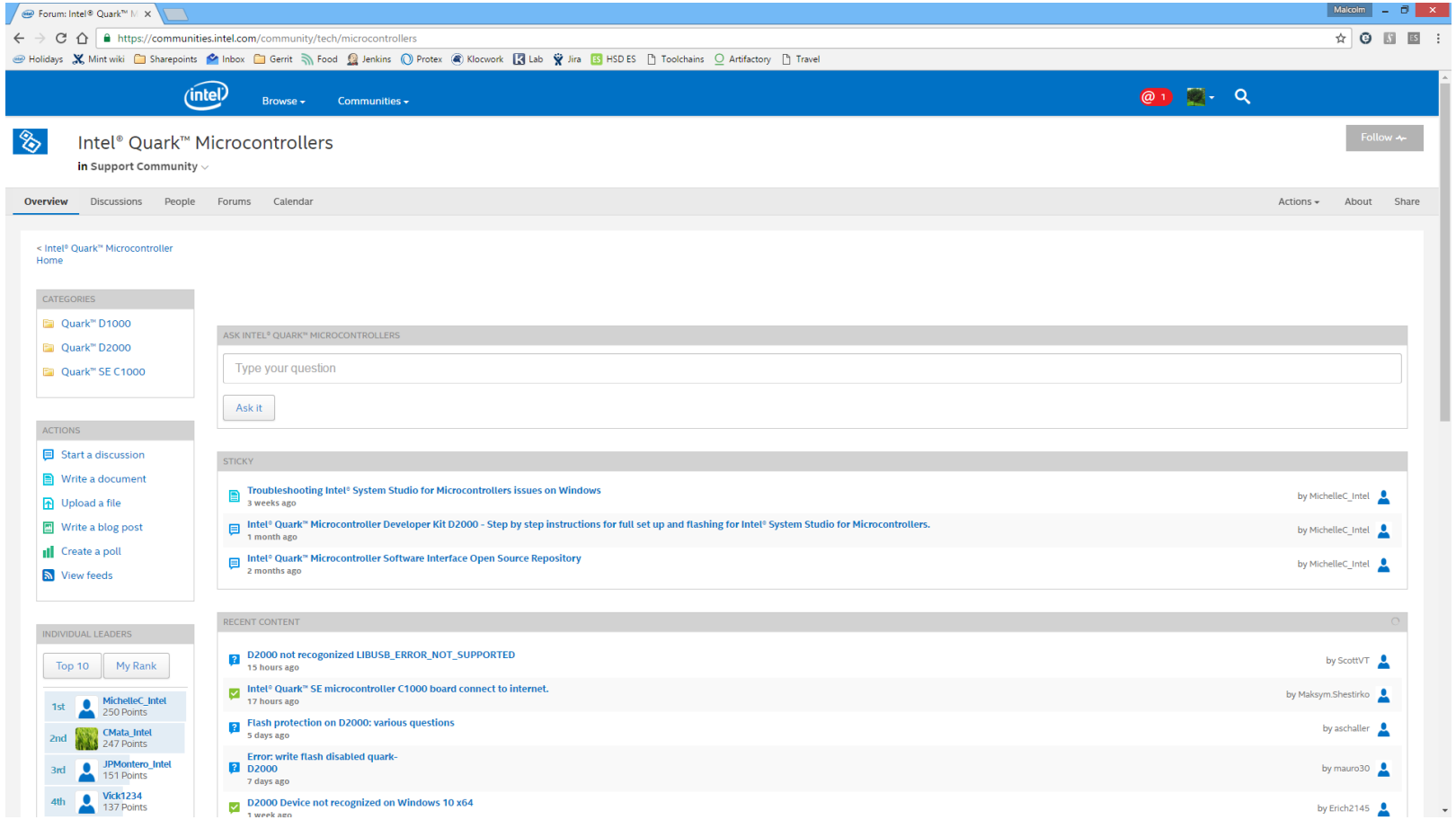
Where to get Intel® QMSI™

Intel® Developer Zone: https://software.intel.com/en-us/intel-system-studio-microcontrollers

Github.com/quark-mcu



Intel® Community



Intel® Quark™ Microcontroller Forum: https://communities.intel.com/community/tech/microcontrollers

Legal Disclaimer & Optimization Notice

INFORMATION IN THIS DOCUMENT IS PROVIDED “AS IS”. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO THIS INFORMATION INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.

Optimization Notice

Intel’s compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Notice revision #20110804


intel
experience
what's inside™