"The future of the 80x51 upgraded for interconnected things"

The Internet of Things (IoT) is the latest buzzword driving the industry for any number of low-power interconnected things. However, IoT encompasses an incredible number of different types of things ranging from edge objects, namely smart or wearable devices which are battery powered with sensors and wireless connectivity, through aggregation nodes, namely hubs, routers and gateways for data aggregation, up to information processing servers in the Cloud to handle the data pushed by edge objects.

Whereas data aggregation and information processing in the Cloud requires high speed and/or high performance computing, a whole range of processing capabilities is required to cover the needs of diverse edge objects. Optimizing for low-power consumption and for Return on Investment according to edge object development cost vs. selling price requires making the best compromises on each subsystem architecture. This implies making the appropriate choice of microcontroller (MCU) or microprocessor (CPU), of their subsystem modes and power management network, per application requirements.

Several segmentations of edge objects can be made depending on the angle of analysis. A first approach is to segment their field according to operating systems, thus to the processing power requirements, where three different levels can be distinguished:

1. Entry level needs with 8-16 bit MCUs running a dedicated application with no operating system,
2. Mid range needs with 16-32 bit MCUs running a Real-Time Operating System (RTOS) of even a Java OS,
3. High end needs with 32-64 bit Application Processors (AP) running an operating system such as Linux or Android to provide graphical data displays and intuitive user interactions.

A second approach is to classify according to increasing security requirements where three different levels can be distinguished:

1. Non-upgradable systems (closed),
2. Firmware upgradable, but non customizable systems (open but simple),
3. Firmware upgradable systems, which can be customized (open and complex), for instance with downloads from an application store.

While 8 or 16-bit MCUs, such as 8051 and 80251 microcontrollers, can be sufficient for entry level edge objects with minimum security, more and more IoT objects with embedded RTOS and upgradable firmware need the enhanced computing capabilities of 32-bit MCUs due to the increasing integration of sensors along with the necessary security-related processing.

It could have been an easy-going extension from the i80251 as the 32-bit extrapolation was readily noticeable in Intel's architecture. But times have changed since Intel's involvement in the 80s and 90s, and a new generation was needed.
To begin with, a codesign process is available nowadays for streamlining the development of the instruction code, i.e. maintaining its i51 legacy while minimizing its power-consumption through simultaneous upgrades of the new compiler and the upgraded processor architecture. The resulting pair, the i351 microcontroller core Zephyr and the compilation tool chain SmartCC, could not be sufficient as two other challenges must be faced: development tools and security.

This article details the architectural upgrade of i351 Zephyr from 80x51 legacy, with a focus on ultra low-power implementation and exceptionally high code-density tailored for edge devices running an RTOS or Java OS. It then describes the complete set of development tools, beyond the traditional compilers and Integrated Development Environments (IDE), namely SmartVision™, for early stage graphical verification of all the networks controlled by the MCU and for the matching between firmware development and hardware implementation with advanced low-power techniques. Finally, this article explains how Zephyr with enhanced "Armored" versions can help users implement custom security counter-measures in order to safely operate in the new interconnected world.

1. Architecture

As a gentle reminder, it is evoked here that 8-bit versus 16-bit versus 32-bit apply to 3 dimensions independently: instruction code, addressing space and word width… The Arithmetic Logical Unit (ALU) performs operations on the word width, which also determines the internal bus width. Thanks to an innovative i351 instruction set and core microarchitecture, Zephyr has the unique flexibility of dealing with 8, 16 and 32-bit words using dedicated instructions and minimum sufficient data path, in order to achieve low power consumption and small silicon area at subsystem-level (including program and data memories).

The 80x51 ecosystem with 32-bit RISC[1] capabilities

The i351 extends the 80x51 architecture with 32-bit addressing capability and with a larger data path to sustain the demands of increasing connectivity (USB, Bluetooth…) and sophisticated analog sensors, while keeping a majority of 8-bit and 16-bit instructions for preserving high code density.

To facilitate migration of existing 80x51 applications, the microarchitecture benefits from similar architectural characteristics:

- Special Function Register (SFR) interface compatible with all existing 80x51 compatible peripherals,
- A compatible memory mapping,
- An instruction set based on the 8051 foundation to facilitate assembly code migration and software developers' comfort.

[1] Reduced Instruction Set Computer

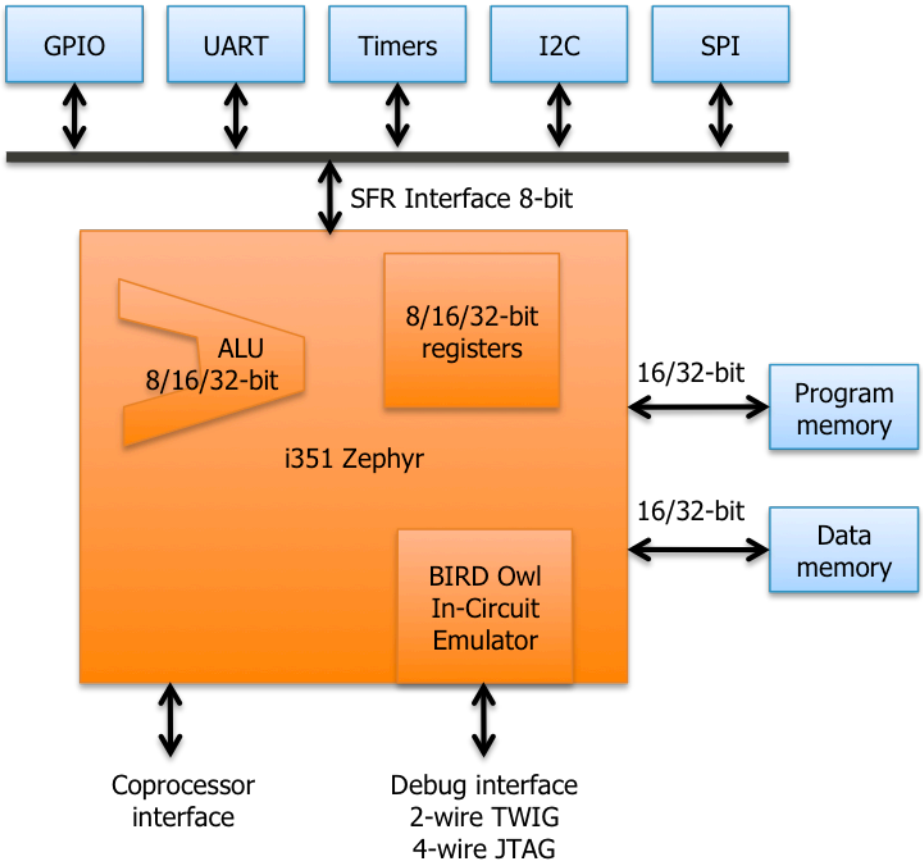

Figure1: A figure of i351 architecture

However, to address the needs for a new generation, the i351 architecture has been significantly enhanced for addressing the needs of diverse IoT applications:

- the Instruction Set Architecture (ISA) has been upgraded to a "RISC" style with more registers along with load/store instructions to obtain a fast and small footprint code when compiling C code with latest generation compilers,
- the pipeline has been optimized so that most instructions execute in a single cycle enabling to reach 1 DMIPS/MHz with consuming as low as 6 uW/MHz at 90 nm,
- memory addressing capabilities have been increased up to 4 GB with linear addressing to avoid complex and slow banking techniques even with very large memories,
- management of interrupts has been extended up to 256 interrupt sources with very low latency,
- a coprocessor interface enables extending the processing capabilities for fast and low consuming execution of application specific algorithms.

Low Power Optimization

IoT devices often spend most of their life time (up to 99.5 % of the time) in non-operating modes waiting to be awoken by external events. As most systems are battery powered, they require ultra low consumption in sleep mode but with fast wake up.
In addition to standard processor power modes which enable to control clock activity, the microarchitecture adds the capability to be switched-off while retaining only the registers needed to keep the execution context. With this unique "Retention Ready (RR)" feature, ultra low power figures (10 times less consuming than a traditional

approach with gated clock) can be reached in sleep mode as most of the registers and all the combinatory logic are switched off. This feature also delivers fast shut down and wake up times (<1 us) since the context never needs to be saved and restored as it is maintained in retention registers. Below is the RR feature with integrated sleep signal pins for reaching ultra low power consumption.

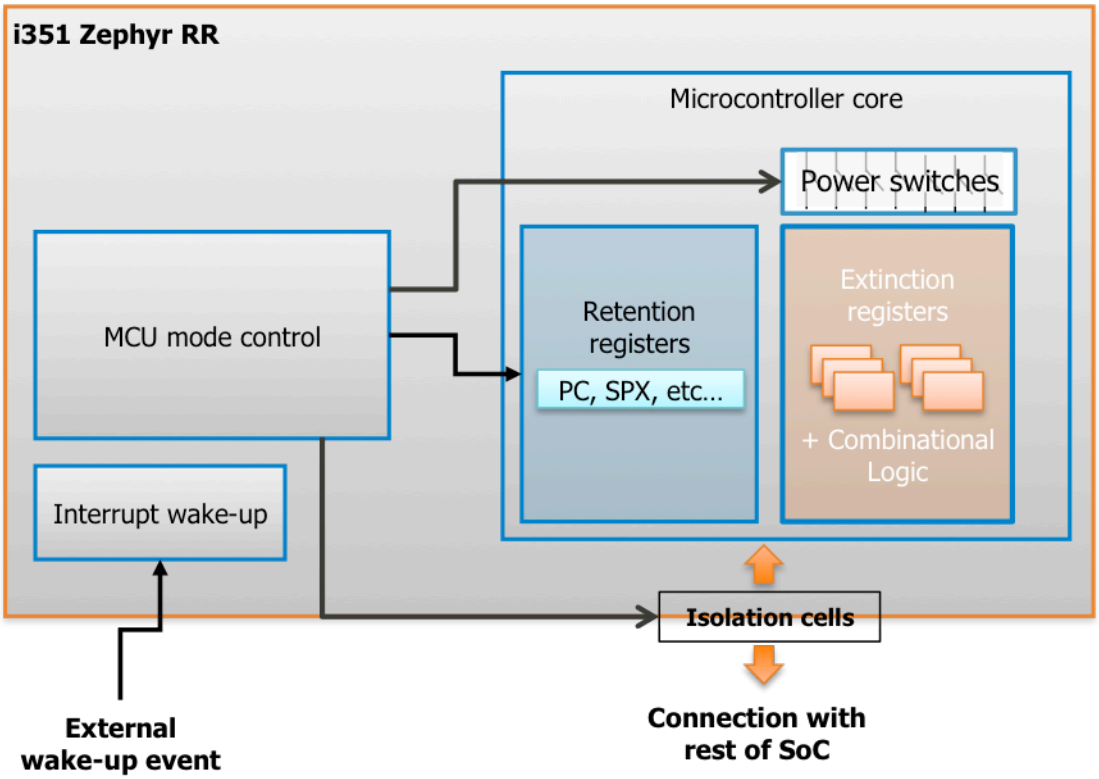

Figure 2: of RR implementation

As the power in operation is also critical, the microarchitecture benefits from architectural choices as well as specific features to limit dynamic power consumption of the core itself but, even more importantly, of the complete processing subsystem including its memories.

The micro-architecture has been carefully designed to reduce the rate of memory accesses which contribute for a large part to the power consumption of systems with such consuming memories as Flash, or with complex bus architectures. This is achieved thanks to:

- a low depth pipeline which limits the number of unnecessary memory fetches in case of a pipeline flush,
- variable size instruction words which contribute to smaller code size and thus to reduce the number of fetches.

A cache controller is also available as an add-on to Zephyr. It has been designed to reduce power consumption and improve performances of Flash or EEPROM based memory systems. When activated, it enables reducing power consumption due to Flash up to 6 times and accelerating memory accesses up to 3 times.

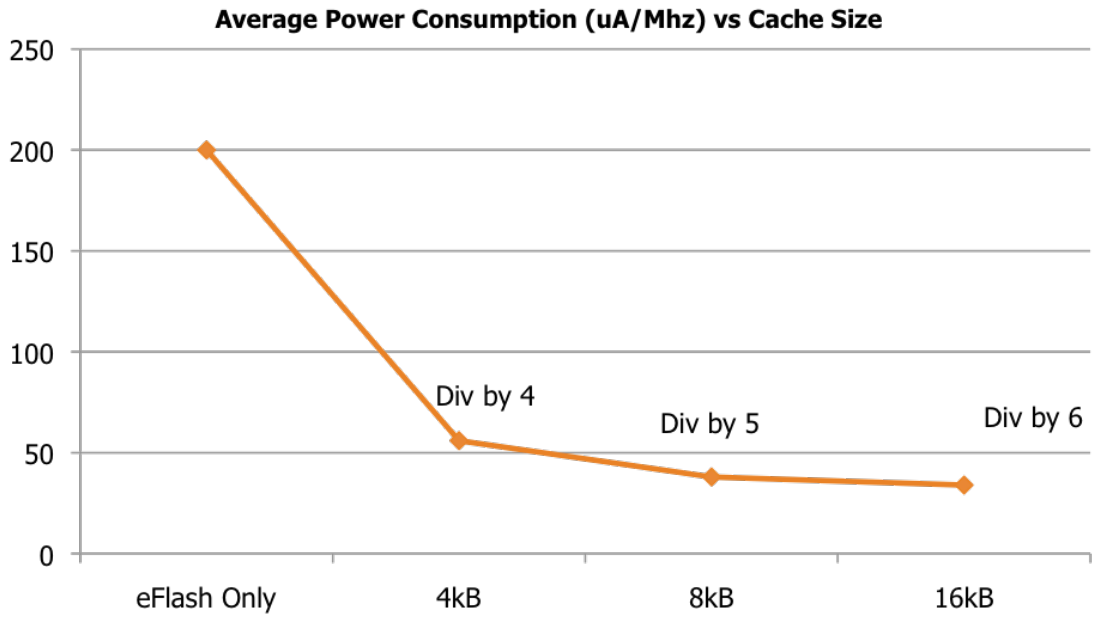

Figure 3: Optimization using a cache controller

To lower the power consumption due to transactions on busses, Zephyr benefits from a dedicated peripheral interface avoiding to go through a hierarchy of consuming bridges (such as an AHB to APB[2] bridge) to access peripherals.
Accesses to this bus can be managed by the processor but also by an autonomous Direct Memory Access (DMA) controller which can perform many transfer operations while leaving the processor and its program memory in sleep mode most of the time. This can lead to significant power consumption reductions as the DMA is more than 10 times less consuming than a processor and its memory.

Power optimization is also available at core level: the instruction set includes dedicated instruction to manage 8-bit, 16-bit and 32-bit words. When executing these instructions, only the necessary registers and logic gates of the data path are activated. Zephyr will thus automatically adapt to the type of data on which operations are performed, resulting in impressive power reduction. For example, when executing a sub program performing operations on characters, which are 8 bit data, only a quarter of the data path will be activated.

With the increased processing power, the microcontroller has the capacity to handle complex tasks with a minimum sequence of instructions. This enables lowering power consumption by reducing the operating frequency of the system or leaving the processor more often in sleep mode.
For optimizing the execution of specific algorithms such as cryptographic or data processing algorithms, the coprocessor interface of Zephyr enables enriching the i351 instruction set with custom implementations of specialized instructions. This approach enables improving processing power and reducing power consumption in comparison to pure software implementations while maintaining flexibility and a low area as compared to pure hardware implementations.

[2] Advanced Peripheral Bus (APB) and Advanced High-performance Bus (AHB) are bus protocols described in the Advanced Microcontroller Bus Architecture (AMBA)

Small Subsystem Silicon Area

Flash memories are now used in most connected devices as program memory, and they often represent a large amount of the die area. Minimizing code size is thus critical to reduce die size and thereby save cost per device.

The i351 instruction set is optimized for ultra high code density to enable the use of small Flash memories thanks to variable size instruction words.
Contrarily to current 32 bit microcontrollers which provide 32 bit instructions or a mix of 32 bit and 16 bit instructions, i351 instructions can be 8, 16, 24 or 32 bits. This high granularity of instruction size enables each instruction to always be encoded with the smallest possible number of bits, avoiding filling code memories with unused information. As shown in Figure 4, the processing subsystem of Zephyr reaches the smallest silicon area and beats typical 32, 16 and 8-bit counterparts!

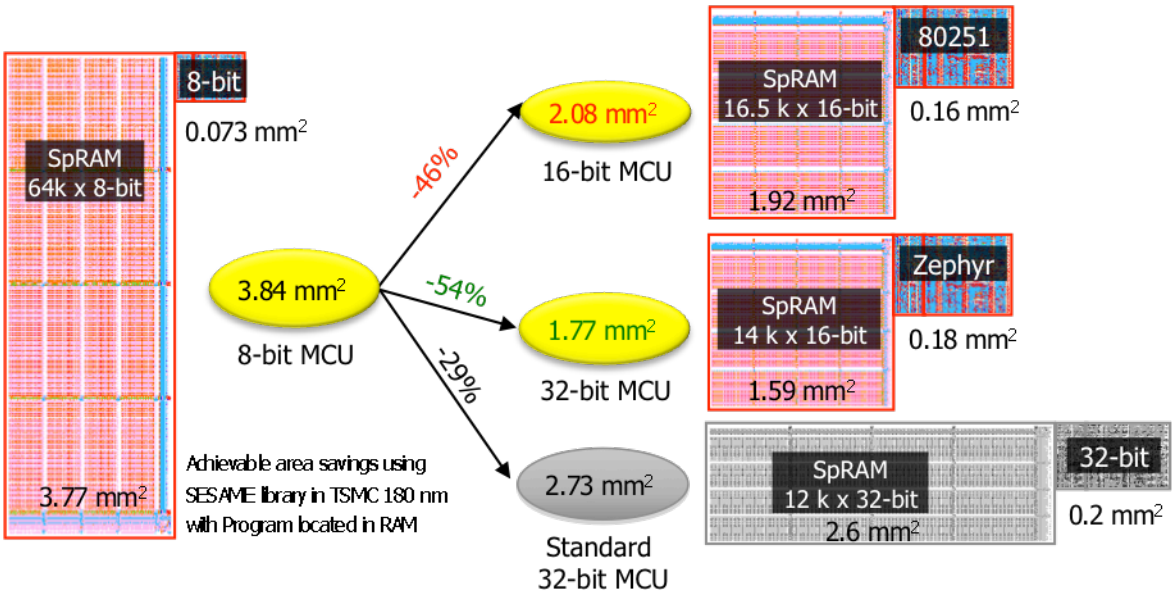

Figure 4: Processing subsystem silicon area comparison

2. Development Tools

Besides MCU architectural enhancements, the first challenge to be faced is how to provide a development environment geared to address low-power consumption software required for the IoT era. Whereas traditional Integrated Development Environments (IDE) are aplenty, granting new software development productivity is mandatory, both for debugging features and for security controls, while facilitating the migration of previously developed application software with the upgrades for lesser power consumption and higher computing power.

For software developer productivity, the SmartCC compiler benefits from LLVM [3] powerful potential for compiling in C, C++ or Java, proposing corrections with source code context aware suggestions displayed in the IDE, debugging with full access to program symbols, up to static analysis of the source code to automatically find bugs in the application program.

[3] The LLVM Project is a collection of modular and reusable compiler and tool chain technologies.

Integrated Development Environments for microcontroller subsystems provide built-in configured subsystems, embedding some configurable memory banks up to specific analog peripherals, along with the capability to program additional peripherals using specific software development kits. Customizing the built-in configured subsystems is limited, with little flexibility on the configuration of the memory subsystem, and tedious when integration of additional peripherals is required.
However, it is necessary to consider how the software code impacts the hardware in order to be able to use the hardware more efficiently. Therefore, a more explicit and user-friendly approach to the configuration of the subsystem is required to enable engineers to develop and validate efficiently the embedded software in the context of the target IoT application. Modeling of the hardware enables finding bugs that would only appear later during the prototyping phase. Indeed, such bugs could not have been detected earlier with a limited subsystem in simulation.

Addressing the challenge of software development tooling took the form of a brand new Interactive Development Environment: SmartVision™ is a timely replacement of the IDE of the 90s, enabling the modeling of the whole peripheral organization for a complex SoC, so as to display the impact of I/O instructions on the network of power regulators as well as modern peripherals, whether from a power consumption or an execution speed point-of-view, or even for security testing.

SmartVision™ provides users with the unique capability for visualizing the networks controlled by the microprocessor (as shown in Figure 5): bussed networks of peripherals or memories and power islands, thereby enabling to assess the impact on the power-regulator network and clock paths. This enables identifying whether the application software is well suited for the control network that has been defined by the hardware team. For example, it allows identifying if a power island is never turned off to save power consumption although the network has been designed to put this island in retention.

Whatever the ultimate choice of microprocessor, SmartVision™ with an 80x51 core enables to develop the C program which will serve to dynamically control the power modes with dual voltage and frequency stepping for optimization of power for performance.

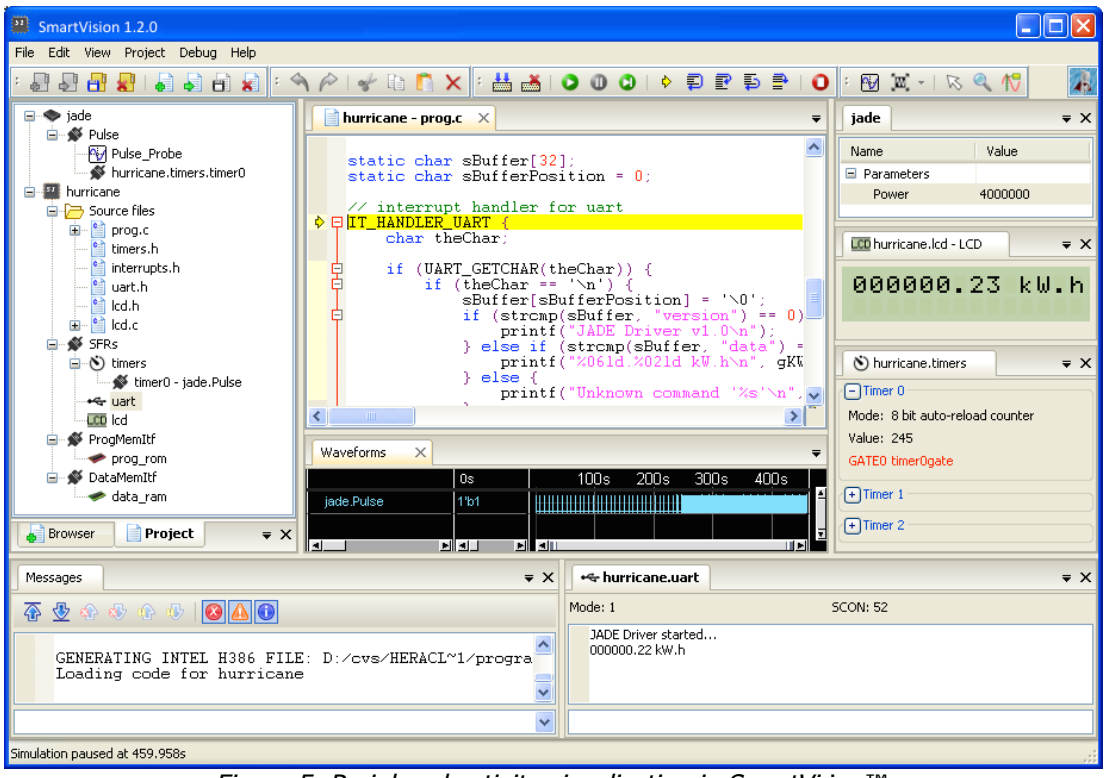

Figure 5: Peripheral activity visualization in SmartVision™

3. Security

Last, but to be hushed, even though it is even more vital for interconnected objects than for Smartcards, the challenge of security controls must be addressed with an open mind, while keeping in mind that all security measures can be broken into given enough time and money.

The objectives of security are primarily to control access to critical data of the application, i.e. to ensure asset confidentiality, and to defend against modifications, i.e. to ensure data integrity. For example, unauthorized users must not be able to gain access to the crypto-keys or to modify the boot memory.

Whether the firmware is upgradable or not, counter-measures are needed to protect against different types of attacks which can range from non-intrusive observation, such as advanced power consumption and electromagnetic analysis, through fault injection to cause errors in operation of the device, up to destructive attacks.

To counter non-intrusive observations, the microcontroller must be designed to balance operations both from a timing and from a power consumption point of view, thereby limiting relevance of observations, whether by design or with additional counter-measures.
To counter fault injection attacks, the microcontroller can handle exceptions such as the execution of non-existing instructions, jumping to an undefined address…

Firmware upgradable systems, whose embedded software can be upgraded, need to properly manage access rights to the memory system. For that purpose, Zephyr supports two levels of privileges. System tasks like critical OS routines, cryptography,

bootstrapping… will be run in privileged mode whereas all other tasks will be run in unprivileged mode with restricted access to memories.

For applications requiring high security controls, a pool of defenses is required from which the "Armored" versions of Zephyr can be equipped. This library includes a confidential set of redundant counter-measures, based on which the security experts of each customer can request a custom configuration related to its security needs. This approach is mandatory to avoid any domino effect, in case a specific custom subsystem has been broken into, since each subsystem will benefit from its specific counter-measure configuration.

The "Armored" versions of Zephyr are equipped to counter non-intrusive observation as well as fault injection attacks, thus making them suitable for IoT edge objects with open but simple security schemes.

Bright future of 80x51 microcontroller

The IoT era is fostering technical innovations for providing the best tradeoffs between low power consumption, low silicon area and small footprint and high performances such as computing power.
Due to the increasing number of connected devices and amount of data analyzed in the Cloud, the security challenge is unprecedented for both personal privacy and public safety.
The new generation i351 Zephyr is carefully evolved from 80x51 legacy to match the diverse processing and security requirements of IoT edge devices. Naturally strict compatibility demands are opposed to innovation and adaptation to the future: evolution is essential for the survivors in such a new era. Zephyr is designed for power optimization with dedicated retention registers and ultra low power modes, minimum memory accesses for dynamic power reduction, together with peripherals such as cache controller and direct memory access (DMA) for further power minimization. Processing subsystems based on Zephyr offer considerable savings in terms of power consumption and silicon area, while providing higher computing power with low system latency. Together with the advanced development environment SmartVision™ which brings the IDE to a higher level and Armored library for dealing with custom security implementations, i351 Zephyr, from 80x251 legacy, is the natural candidate for mid-range IoT edge objects with embedded RTOS.

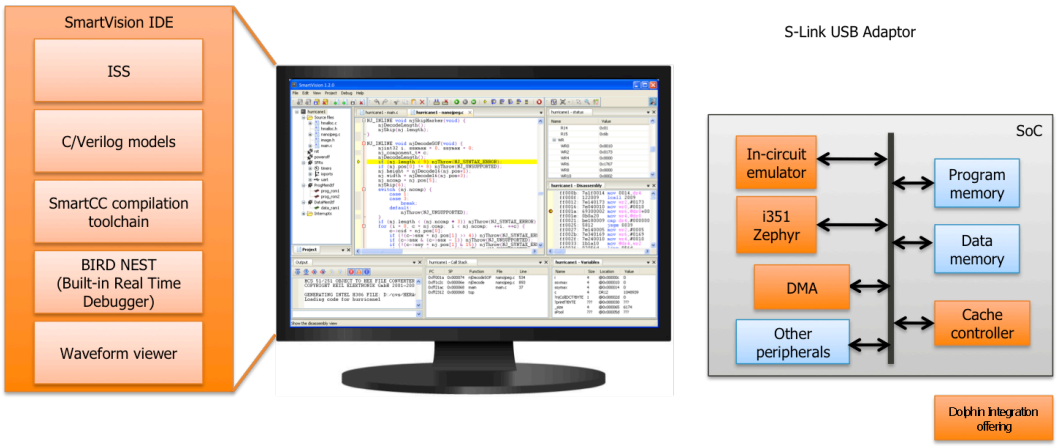

Figure 6: Visual representation of complete Zephyr offering

For further information on Zephyr and SmartVision offering, please contact logic@dolphin.fr

About Dolphin Integration

Dolphin Integration contributes to "enabling mixed signal Systems-on-Chip" for worldwide customers - up to the major actors of the semiconductor industry - with Silicon IP components best at low-power consumption.
This wide offering is based on innovative libraries of standard cells, register files, memory generators and power regulators. Complete networks for power supply can be flexibly assembled together with their loads: from high-resolution converters for audio and measurement applications to power-optimized micro-controllers of 8 or 16 and 32 bits.
Over 30 years of diverse experiences in the integration of silicon IP components and providing services for ASIC/SoC design and fabrication, with its own EDA solutions solving unaddressed challenges, make Dolphin Integration a genuine one-stop shop covering all customers' needs for specific requests.

The company striving to incessantly innovate for its customers' success has led to two strong differentiators:

- state-of-the-art "configured subsystems" for high-performance applications securing the most competitive SoC architectural solutions,
- a team of Central and Field Application Engineers supporting each user's need for optimal application schematics, demonstrated through EDA solutions enabling early performance assessments

Its social responsibility has been from the start focused on the design of integrated circuits with low-power consumption, placing the company in the best position to now contribute to new applications for general power savings through the emergence of the Internet of Things.

Please visit our website at www.dolphin-integration.com.