Atmel Studio 7
New features and Atmel Tools

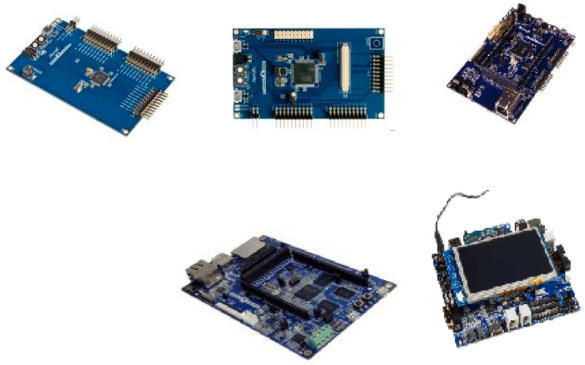
Contents

Atmel Product Introduction
Atmel Hardware tools
Atmel SW Development Tools
Development Cycle with Atmel Device
Improvement on Atmel Studio 7
- Ease of Use & Performance improvement
- Atmel Start integration & ASFv4.0
- Help System Improvement
- Visualization update. (Atmel Data Visualizer)
-

Demo
- Atmel Start
- Help Feature
- Data Visualizer
- External Tools (Performance Analyzer(Zigbee/BLE)

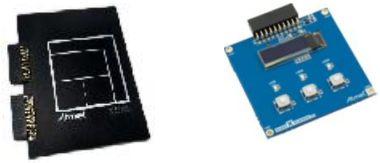
Atmel Product Introduction

- **MCU**
  - ARM Based : Cortex-M0,M3,M4,M7 (ex. SAMD2x,SAMLx,SAM4x)
  - AVR Based : AVR 8/16/32 bit MCU Series (ex. AVR, Atmegar, TinyAVR)
- **MPU**
  - ARM Based : Cortex-A5 Series (ex. SAMDA5Dx)



- **RF and Connectivity**
  - Zigbee (ex. SAMR21/AT86RF233)
  - BLE (ex. BTLC100/SAMB11)
  - Wifi (ex. WINC1500/WILC1000/SAMW25)
  - Sigfox (ex. ATA8520D)

- **Touch Key (PTC)**
  - QT6
  - Max Touch



- **ETC**
  - EEPROM/CRYPTO (ex. AT24C16D /ECC508)
  - Arduino / Sensors

Atmel Hardware tools

Low cost hardware platforms for evaluation and rapid prototyping

Xplained eval board platform

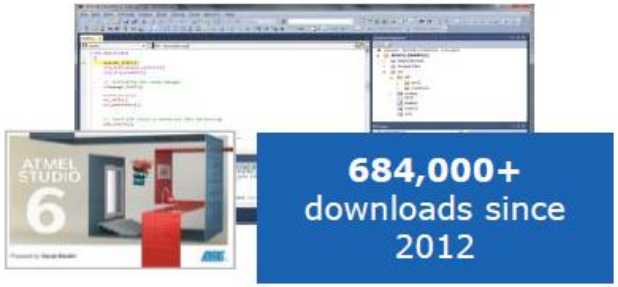
- Extendable using Atmel extension boards or Arduino shields
- All boards supported in Atmel Studio with example projects
- All design files available. Easy to reuse in customer projects
- On-board debugger on all boards

- **Debuggers & Programmers**
- Atmel-ICE Stand-alone debugging & programming tool for all ARM/AVR cores
- Atmel EDBG on-board debug solution for integration
- power debugger for high accuracy power measurement suppo

Atmel SW Development tools

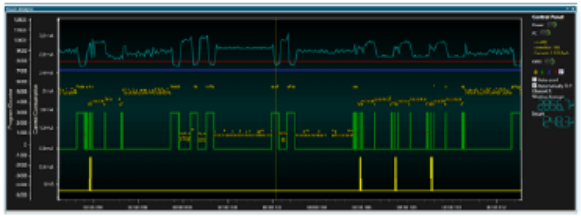
Strongest MCU tools portfolio on the market

- **Atmel Studio**
- Based on Visual Studio frontend, supports 8/32-bit AVR and ARM development & advanced debugging



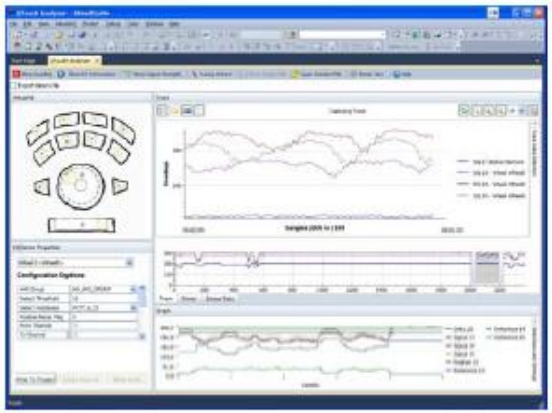
- Studio 7 adds migration path for makers to production ready tools **684,000+** downloads since 2012

- **Atmel Data Visualizer**
- Oscilloscope view allows you to Monitor application behavior, display data & interact with running applications



- Renders power data while debugging.
  Real time power graphs that visually illustrate power modes

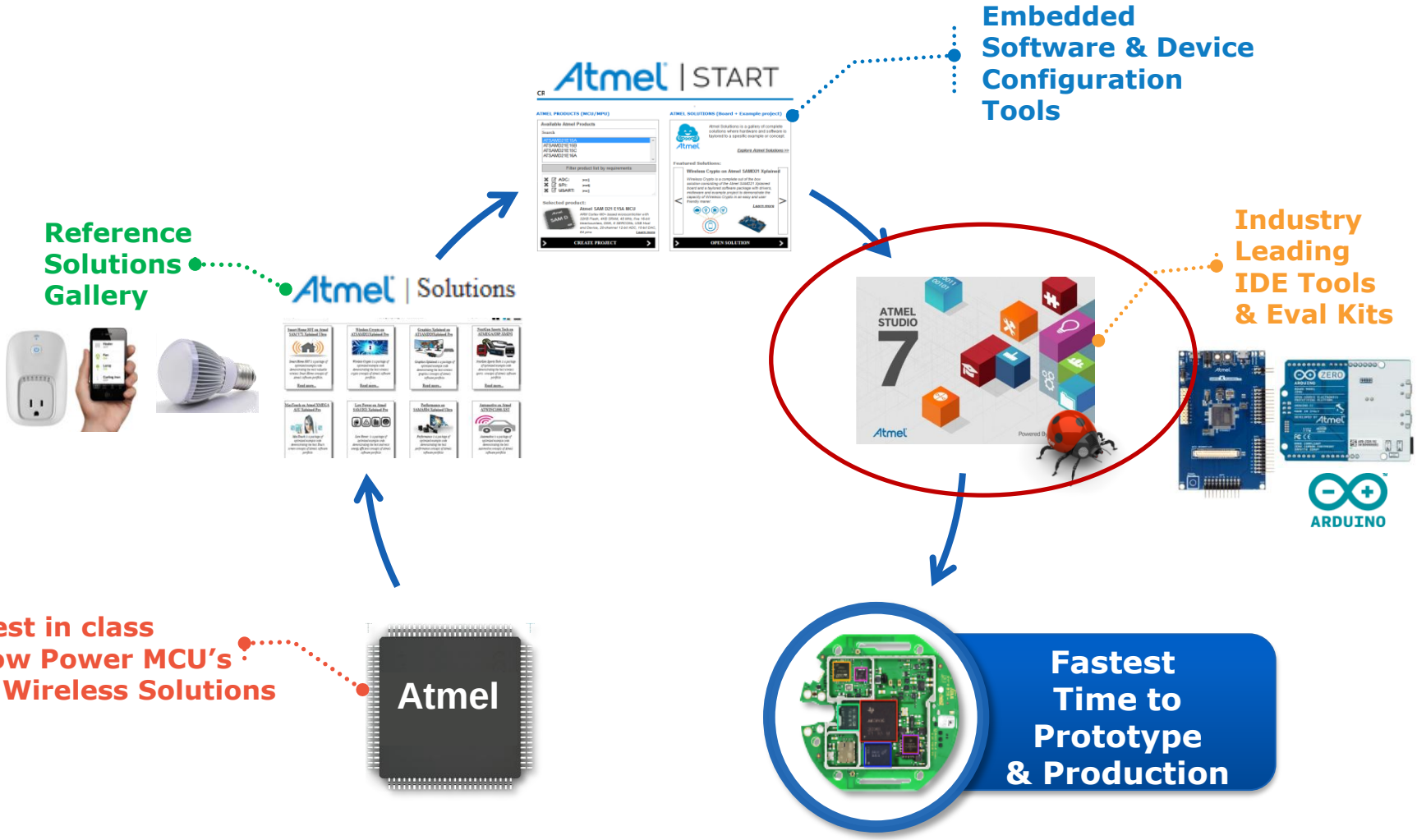
- **Q-Touch Studio & Wireless composer**
- Studio plug-ins for step-by-step guidance for setting up and testing touch & RF systems.



- **Available for free at atmel.com**

Development Cycle with Atmel Device

Addressing the complete product development cycle

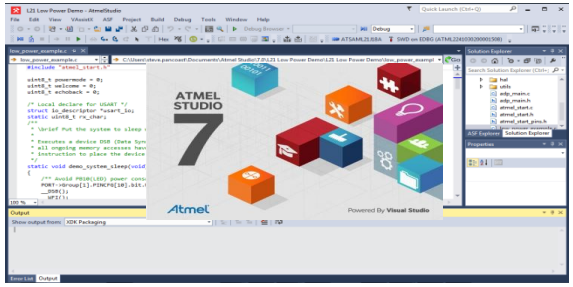


All backed by Atmel's leading Customer Support !

Improvement on Atmel Studio7

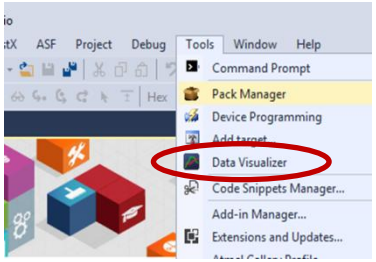
Improvement on Atmel Studio 7

Focus is on Ease of Use & Performance

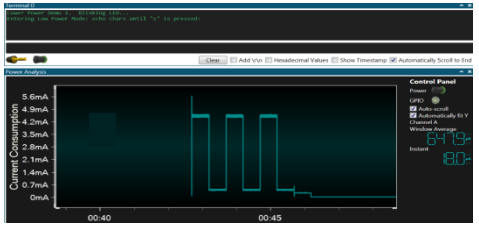


- **Latest Visual Studio Isolated Shell**
  - Improved IDE responsiveness and startup times
  - Modern user interface with improved editing and

- **Source code control features**
  - Windows 10 support
  - Modularized installer to reduce size
  - 2MB Web Installer, selected components downloaded on demand
  - Reduces average download requirement by 50%
  - All selections done at start of installation, enables unatended installs.



- **Ease of use Visualization Tools**
  - Support for Power visualization and Data visualization
  - Real time graph power - visually illustrating MCU power modes
  - Power Debugging

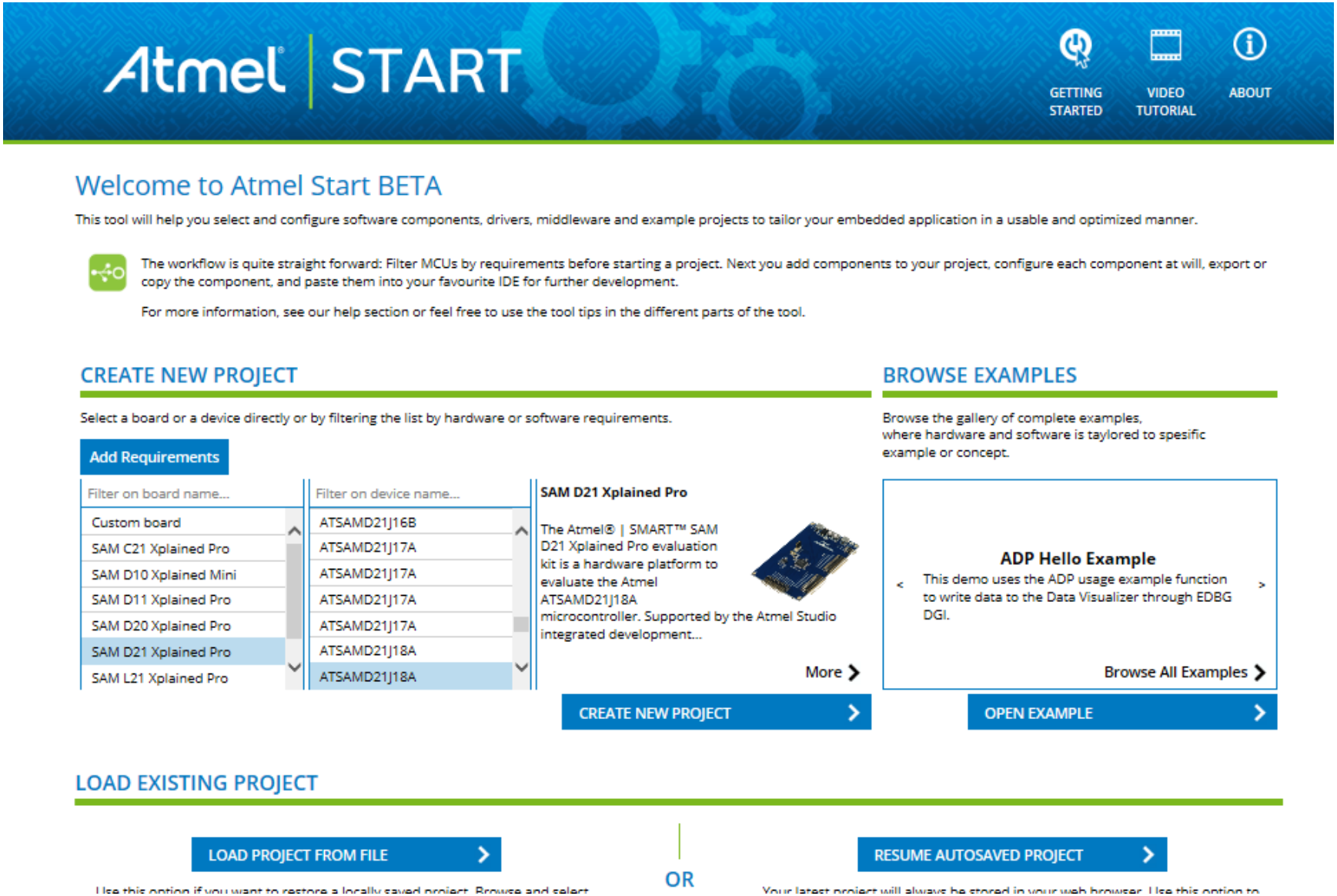


- **Arduino project import**
  - Ability to import Arduino sketches as C++ projects, creating a migration path for users that wish to go pro.

- **Coming in Q4 - MPU support in Studio (SAMA5)**
  - Support for SAMA5 devices(non MMU mode) will be added to Studio, as well as to the Atmel-ICE probe.

Improvement on Atmel Studio 7

New tool - Atmel Start (1) – http://start.atmel.com



Improvement on Atmel Studio 7

New tool - Atmel Start (2) – http://start.atmel.com

Web based SW, no installation required.

- Connected with My Atmel, always up to date.

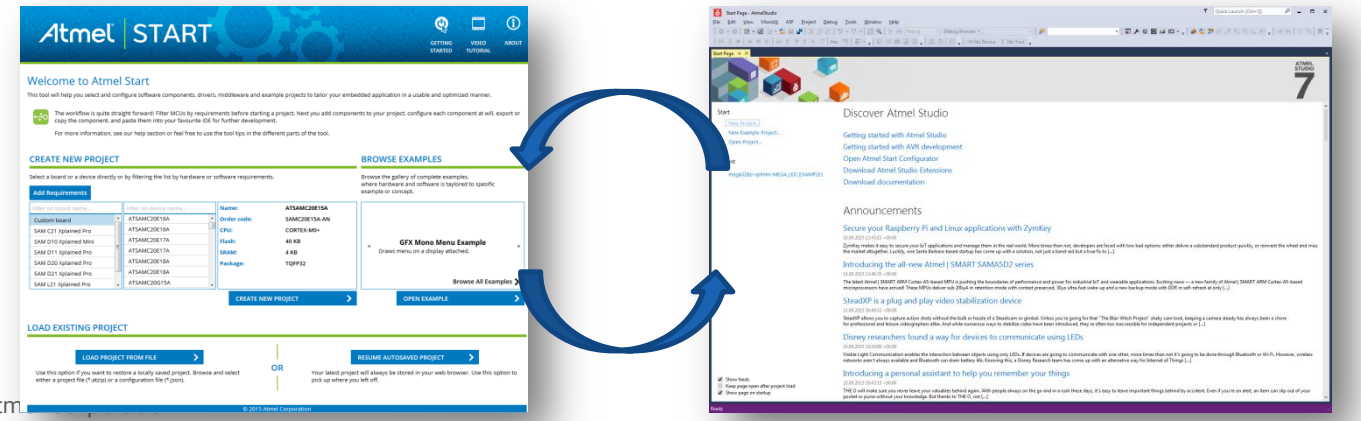
Explore software components, examples and complete reference solutions

- Explore Atmel kits and examples / solutions
- Repository of software components
- Atmel SW, 3rd party SW and Open Source community SW

Visual software & device configuration

- Configure middleware and software drivers
- MCU Pin mux, Peripherals and System clock
- Automatic generation of clean, readable code

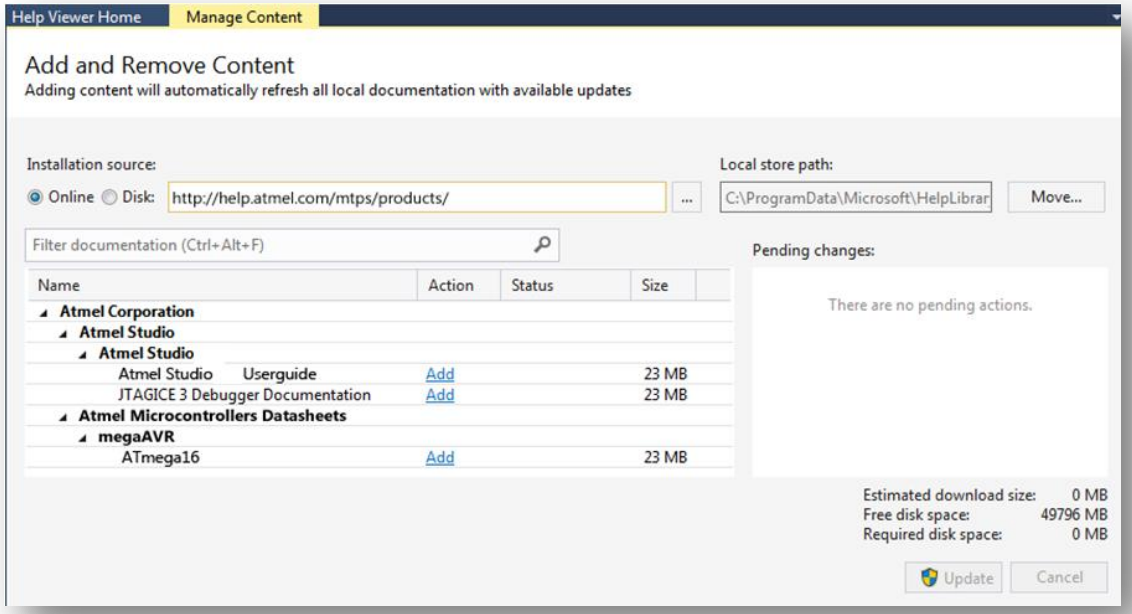
IDE agnostic - supports code/project generation for Studio, IAR and Keil



Improvement on Atmel Studio 7

Help System Improvement (1)

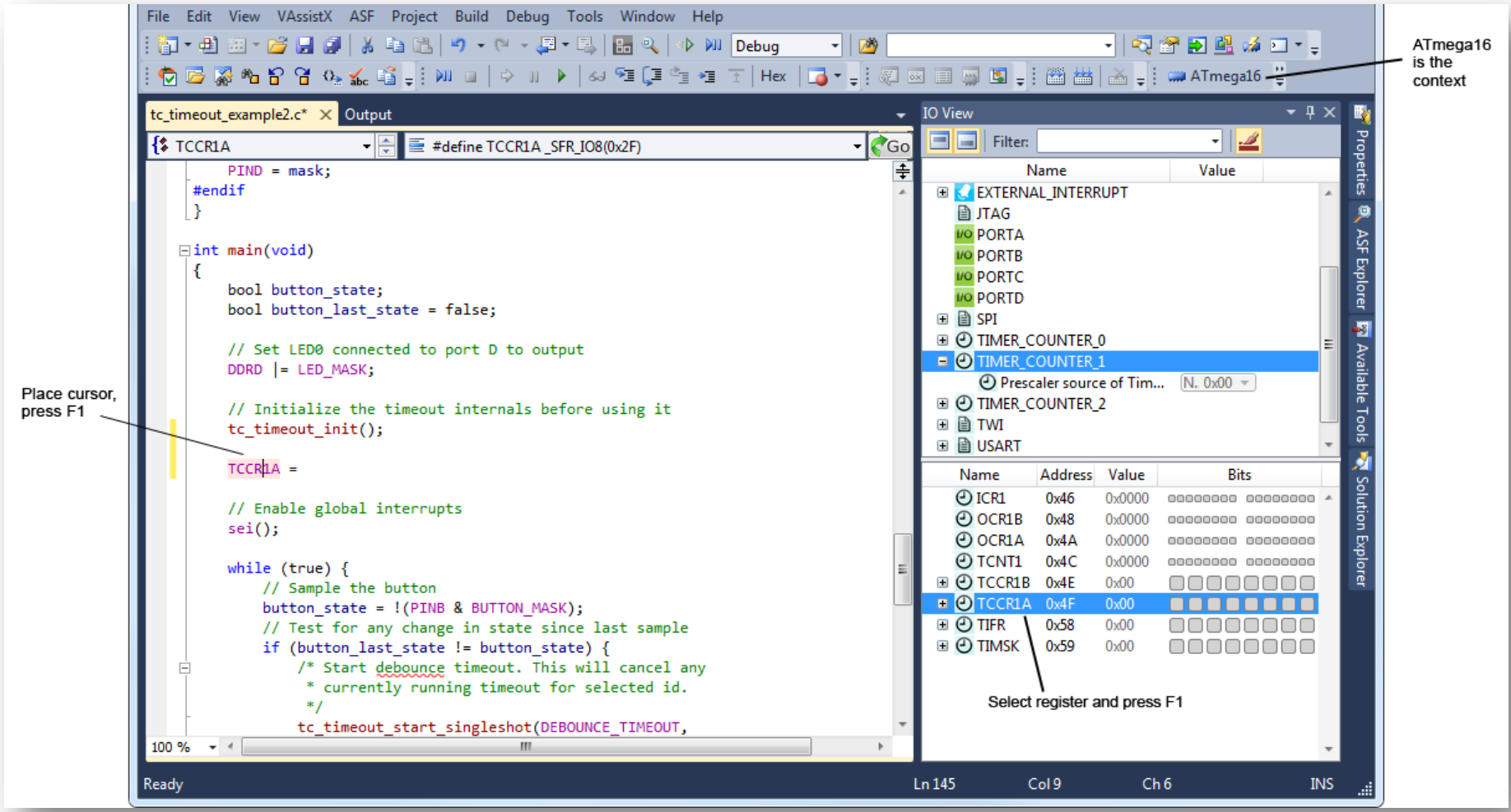
- Manage help content to download
- Have up-to-date help
  - Users are notified of available updates
- Help can be updated independent of Atmel Studio releases
- Reduces download size of installer
- Data sheets will be made available for search



Improvement on Atmel Studio 7

Help System Improvement (2)

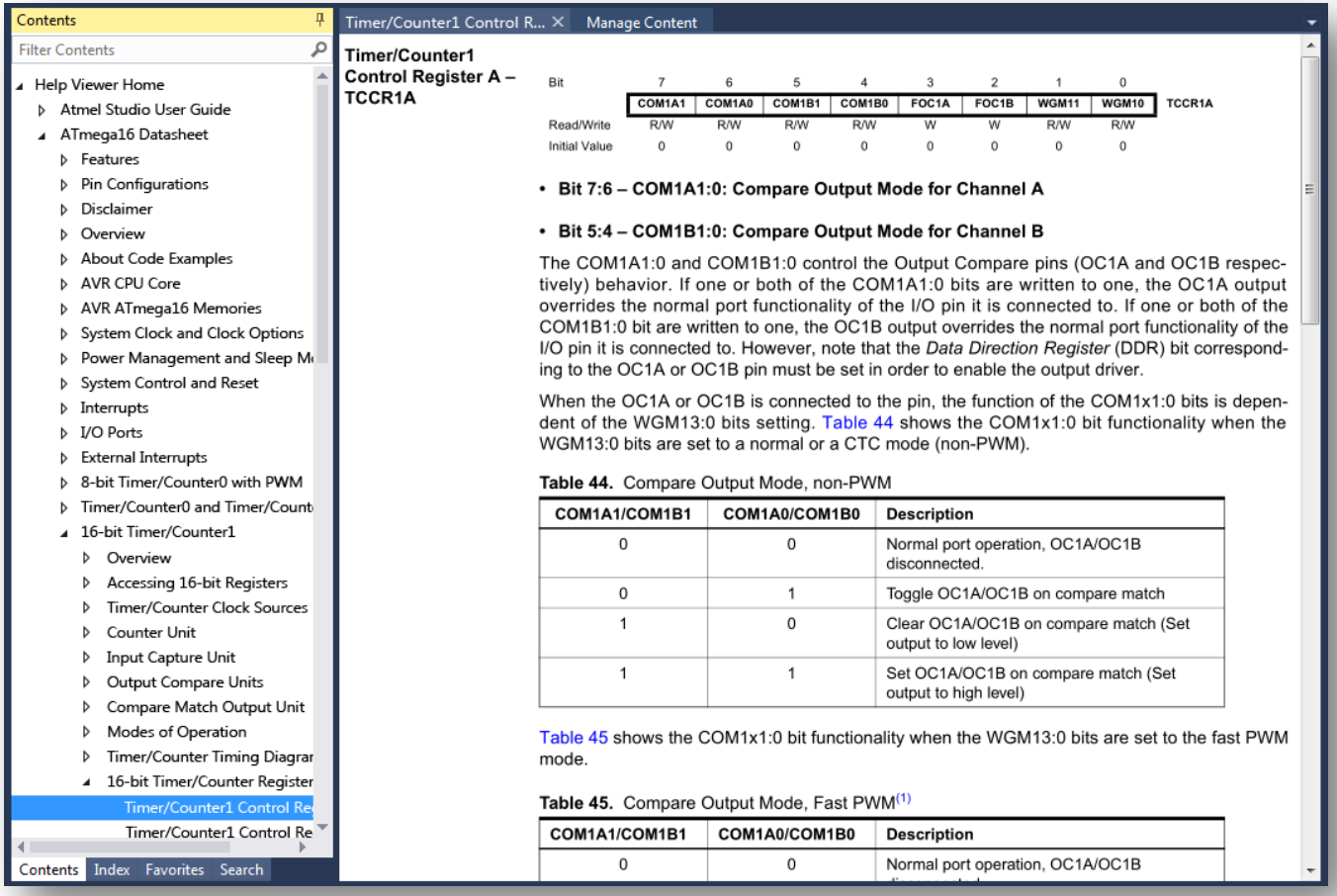
- In editor and IO view: lookup datasheet & other prog. references



Improvement on Atmel Studio 7

Help System Improvement (3)

In editor and IO view: lookup datasheet & other prog. references

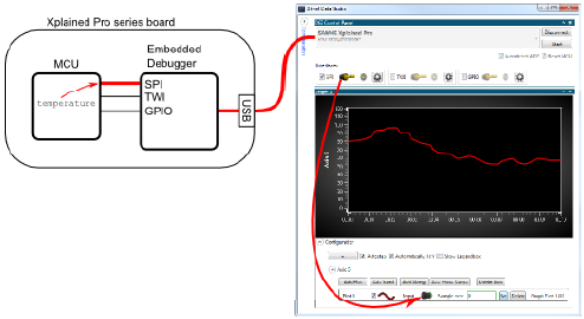


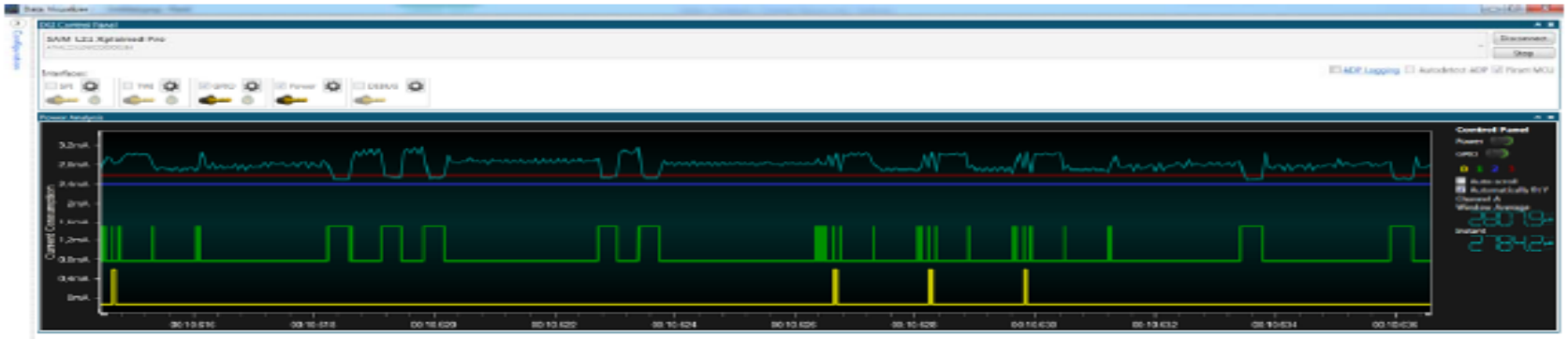
Atmel Data Visualizer

Visualize power consumption during debugging

Available in Atmel gallery

- Processes and renders data collected while debugging
- Uses EDBG Data Gateway Interface, or serial port on custom boards
- Renders GPIO, SPI, TWI, USART.. Today.
- Roadmap covers touch, motor feedback, wireless...
- Power data supported starting with SAML21-XPRO
- **Can run as stand alone application**
- Pluggable view in Studio 7, stand alone for other IDEs

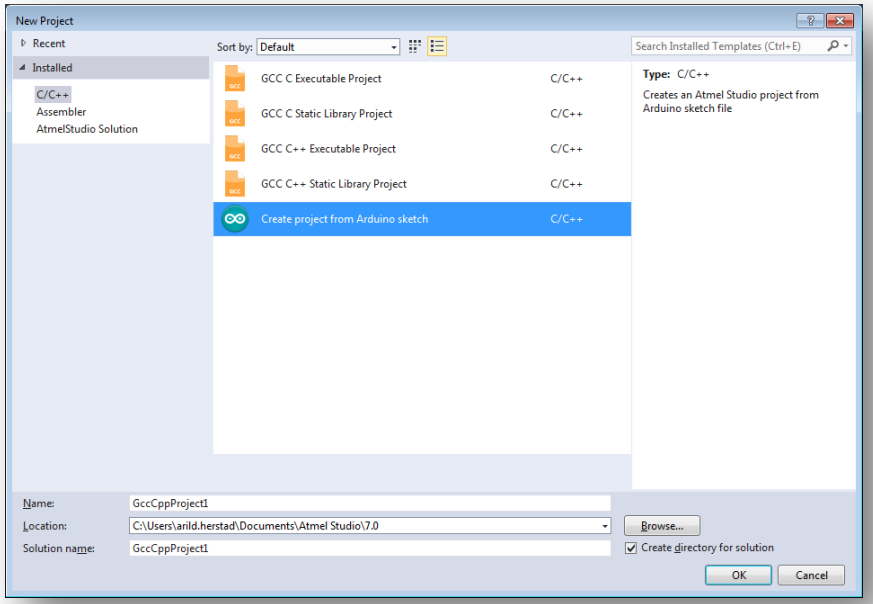




From Maker to Market

Ease path for Arduino users to move to Atmel Studio

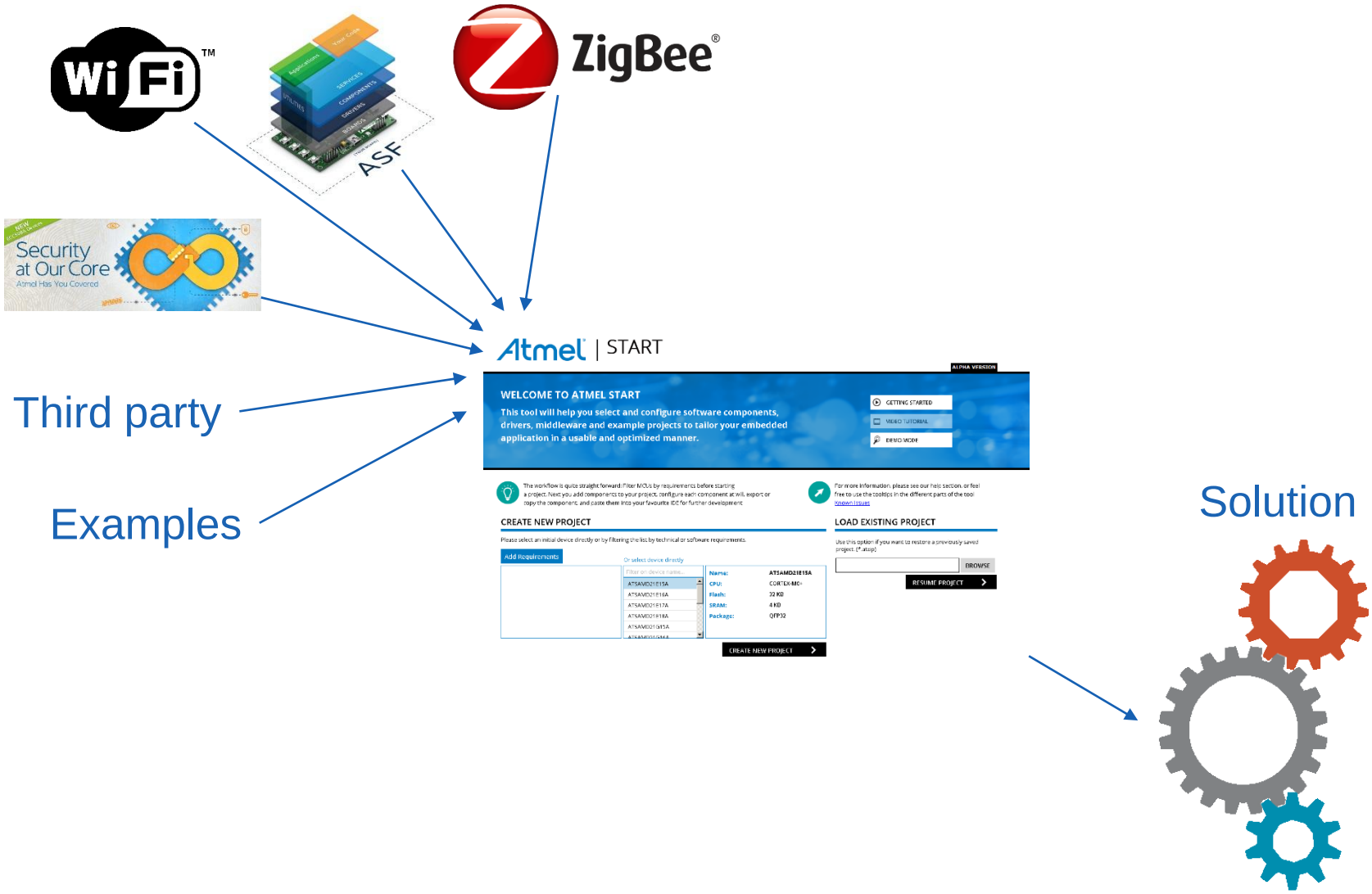
- Import Arduino project into Studio
- Opened as a C++ project with include paths to the Arduino library source code



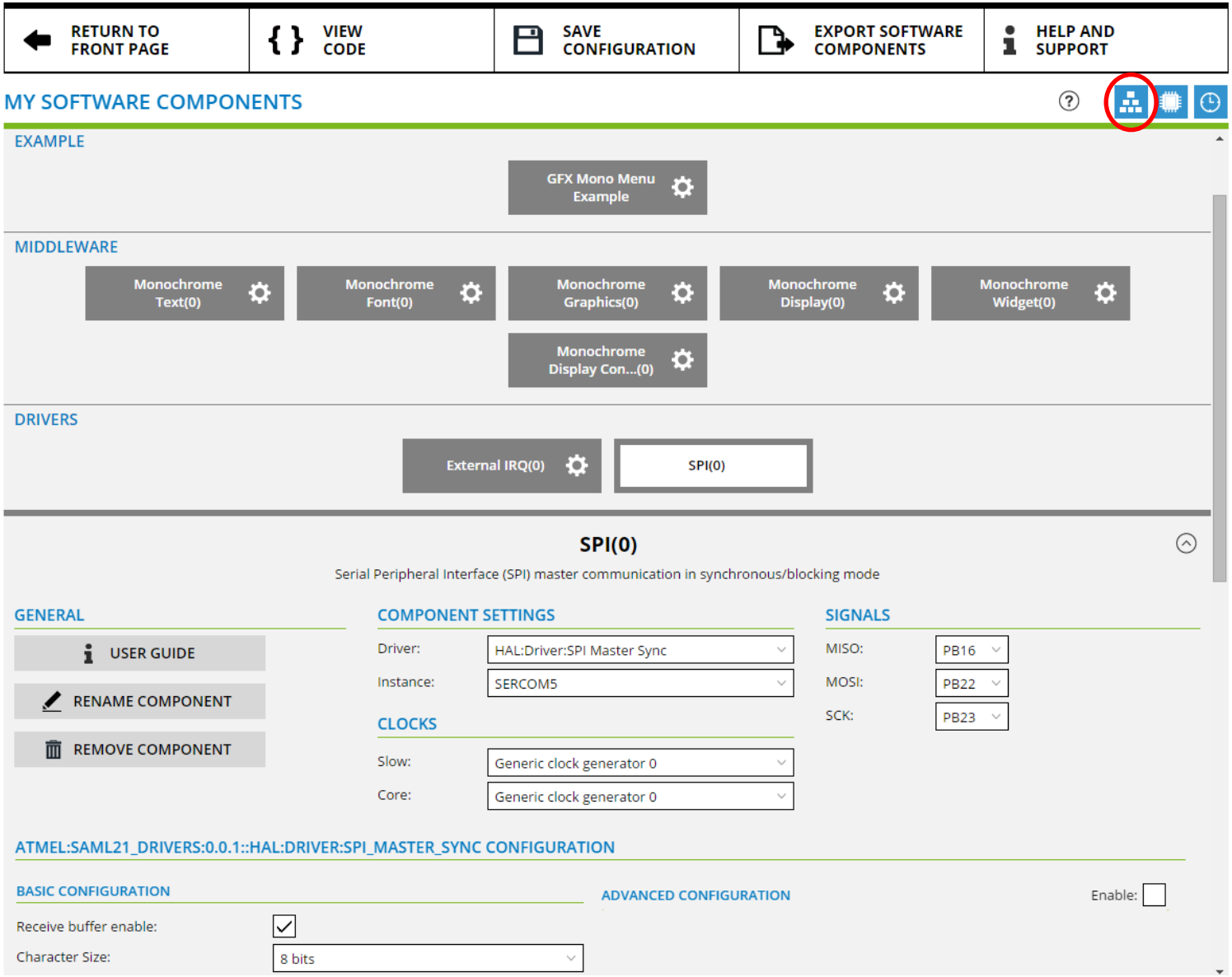
Demo

Atmel Start

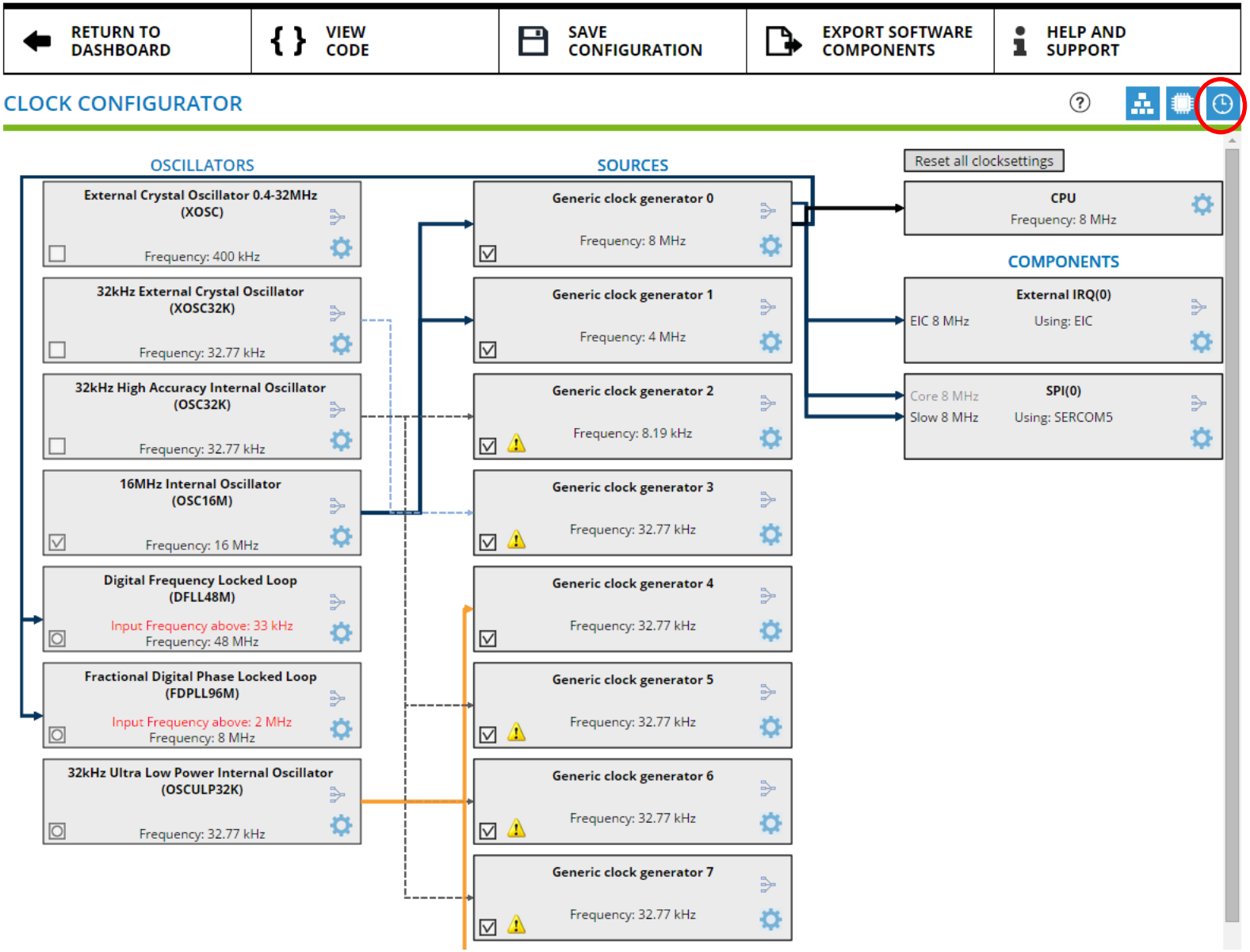
A hub for software examples and solutions



Dashboard and software configuration



Configure the clocks

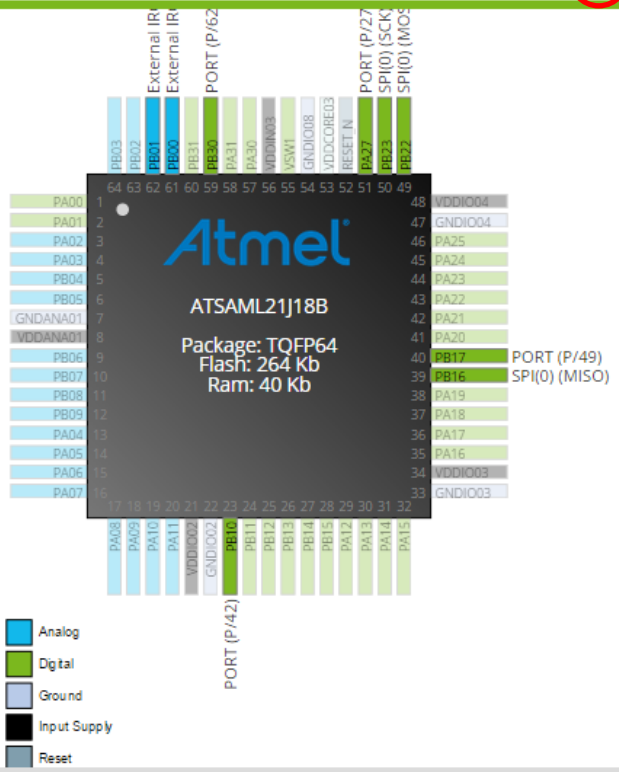


Configure the pinmux

RETURN TO DASHBOARD | VIEW CODE | SAVE CONFIGURATION | EXPORT SOFTWARE COMPONENTS | HELP AND SUPPORT

PINMUX CONFIGURATOR

#	Pin label		Board label		Mode	Signal	
	Pad	User	Header	Pin		Label	Mode
External IRQ(0)							
61	PB00	BUTT...	EXT3	ADC+	Digital input	EXTIN...	Enabled
62	PB01	BUTT...	EXT3,DG...	ADC...	Digital input	EXTIN...	Enabled
PORT							
23	PB10	LED1	LEDs,EXT3	Yell...	Digital output	P/42	
40	PB17	MON...	EXT3	SPI_...	Digital output	P/49	
51	PA27	MON...	EXT3	SPI_...	Digital output	P/27	
59	PB30	MON...	EXT3	GPI...	Digital output	P/62	
SPI(0)							
39	PB16	MISO	EXT3,EX...	SPI_...	Digital input	MISO	
49	PB22	MOSI	EXT3,EX...	SPI_...	Digital output	MOSI	
50	PB23	SCK	EXT3,EX...	SPI_...	Digital output	SCK	
No software components							



PIN CONFIGURATION SETTINGS

Pin 49 (PB22) is used as **MOSI** with **SPI(0)**.

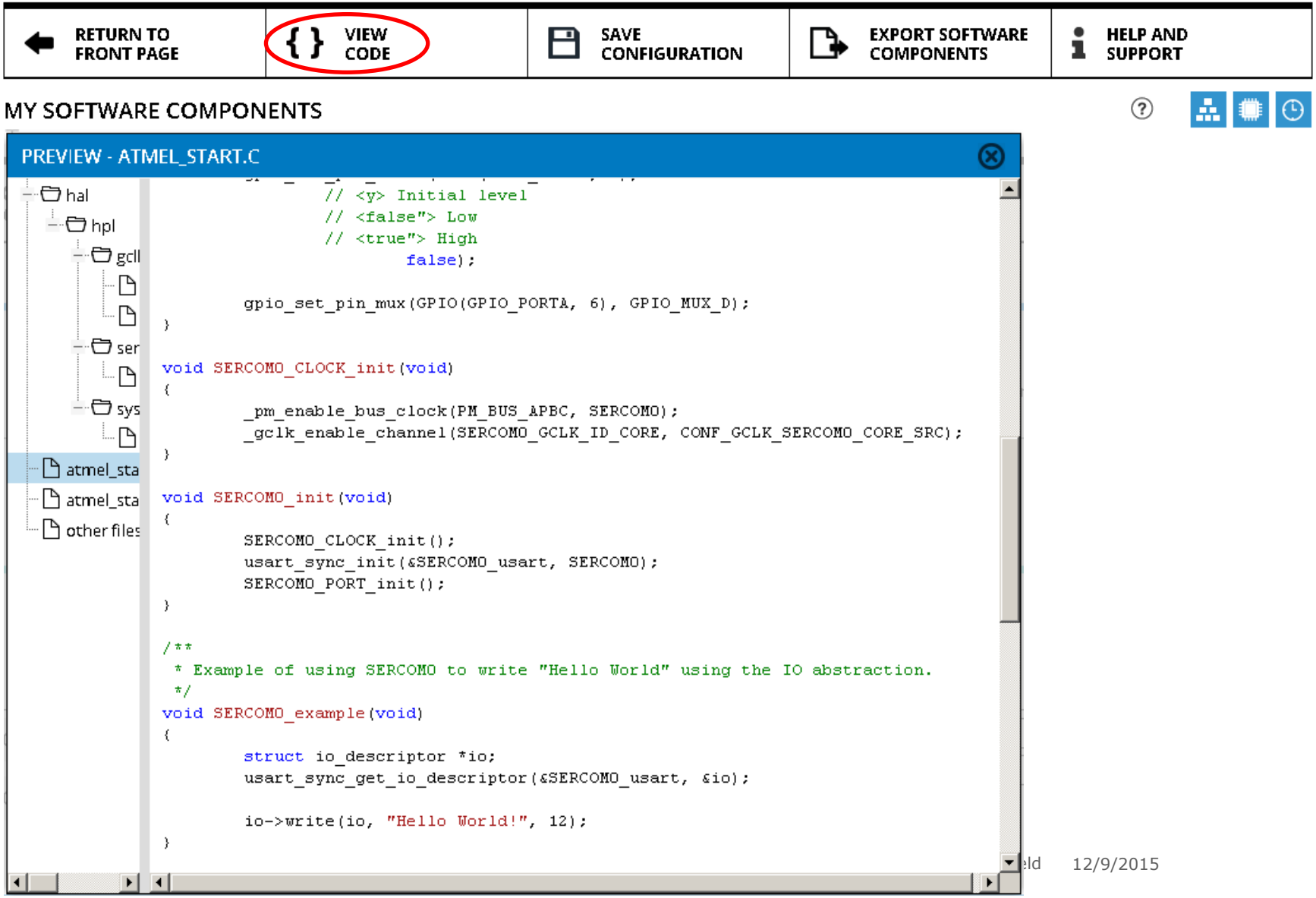
User label: MOSI

Pin mode: Digital output

Initial level: Low

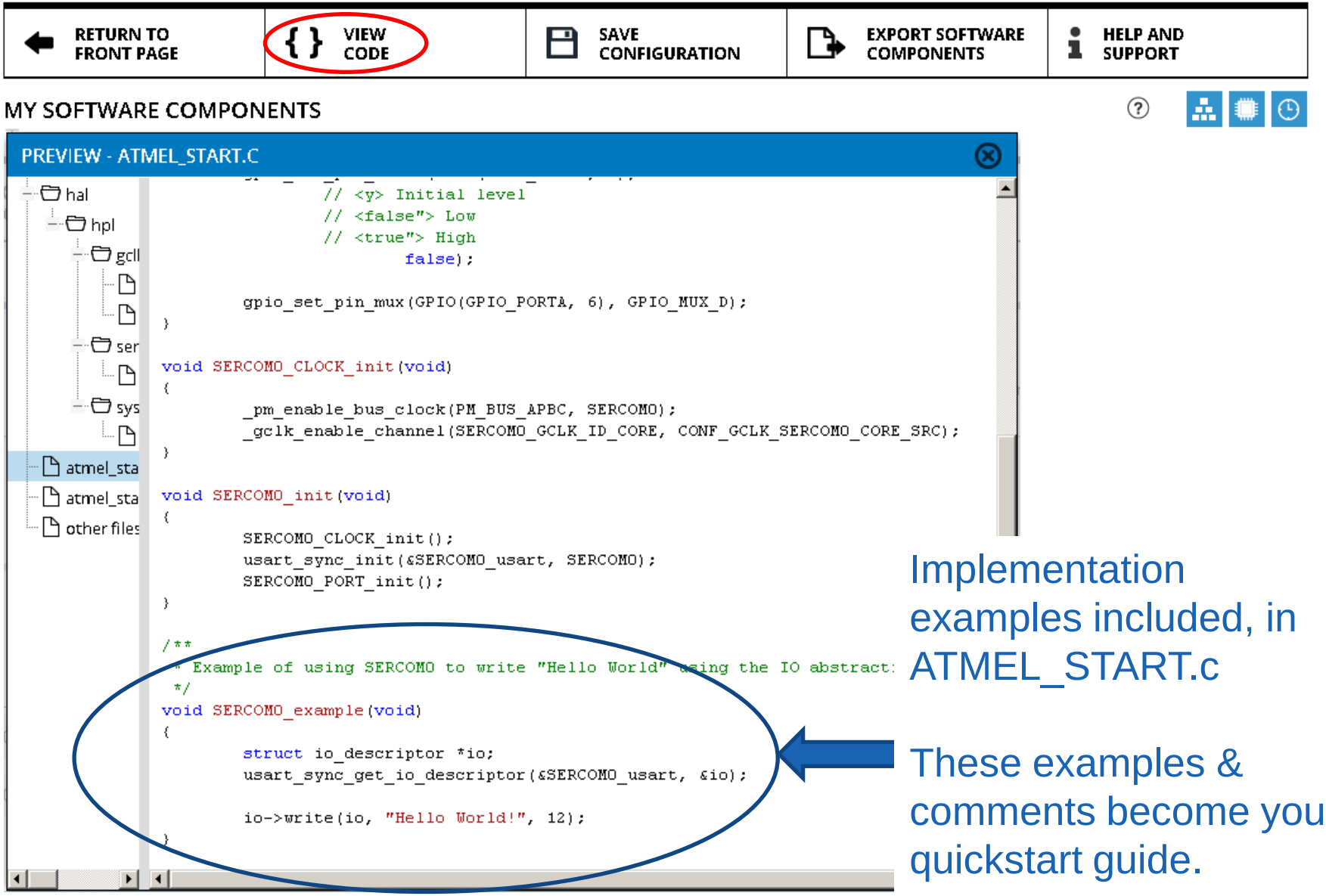
Project preview and exporting

Project preview and exporting



Project preview and exporting

Project preview and exporting



Implementation examples included, in ATMEL_START.c

These examples & comments become your quickstart guide.

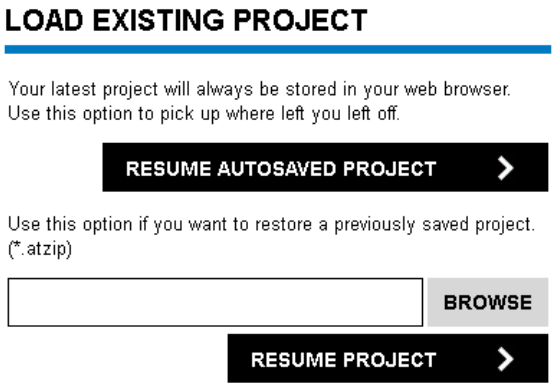
Project preview and exporting

Project preview and exporting

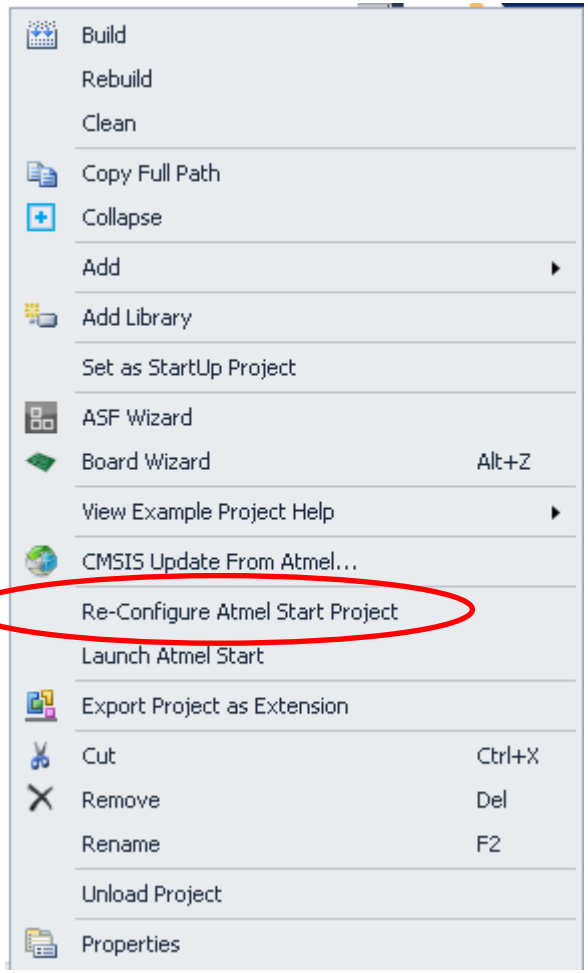
Project preview and exporting

Project preview and exporting

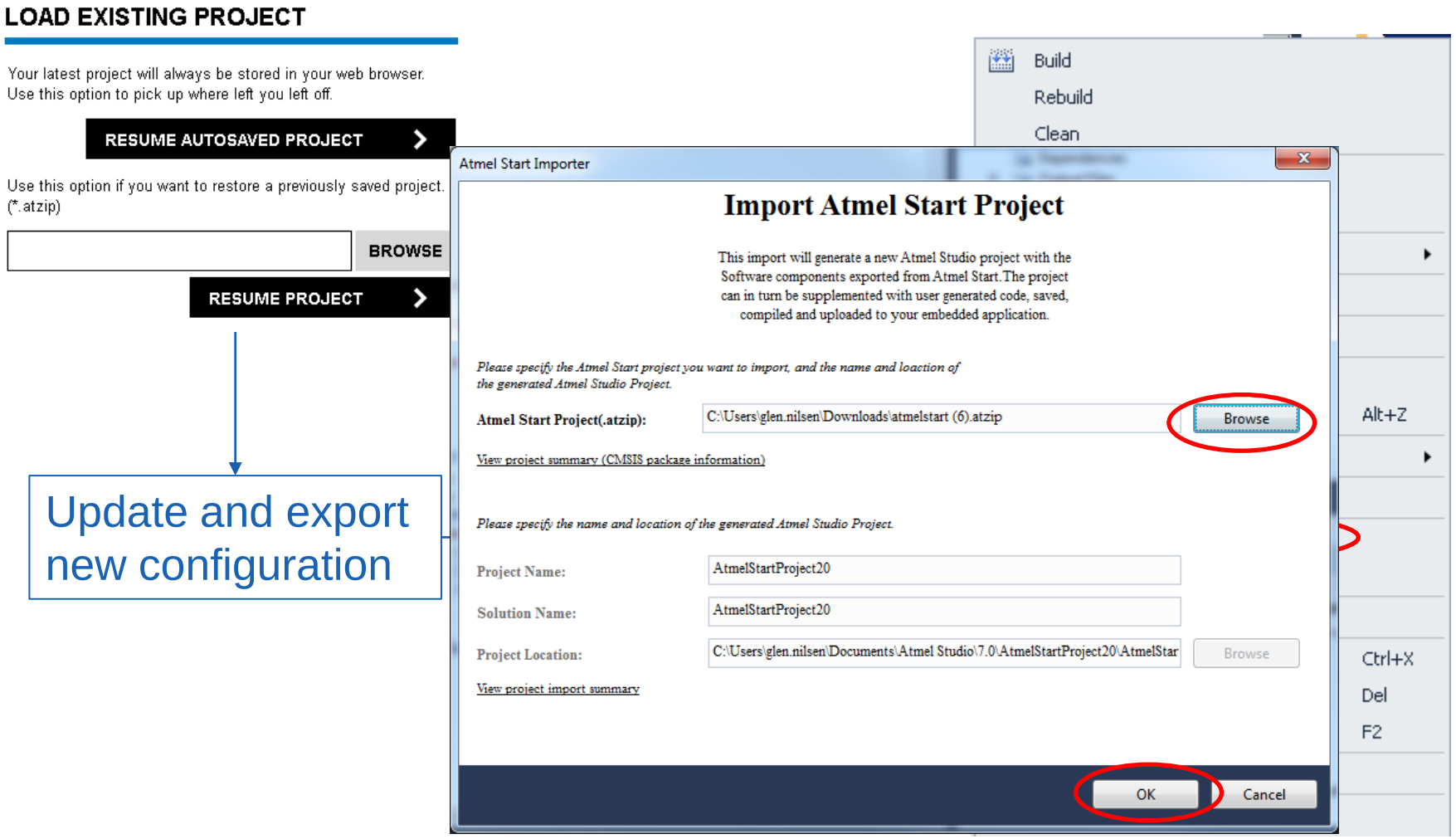
Re-configure project



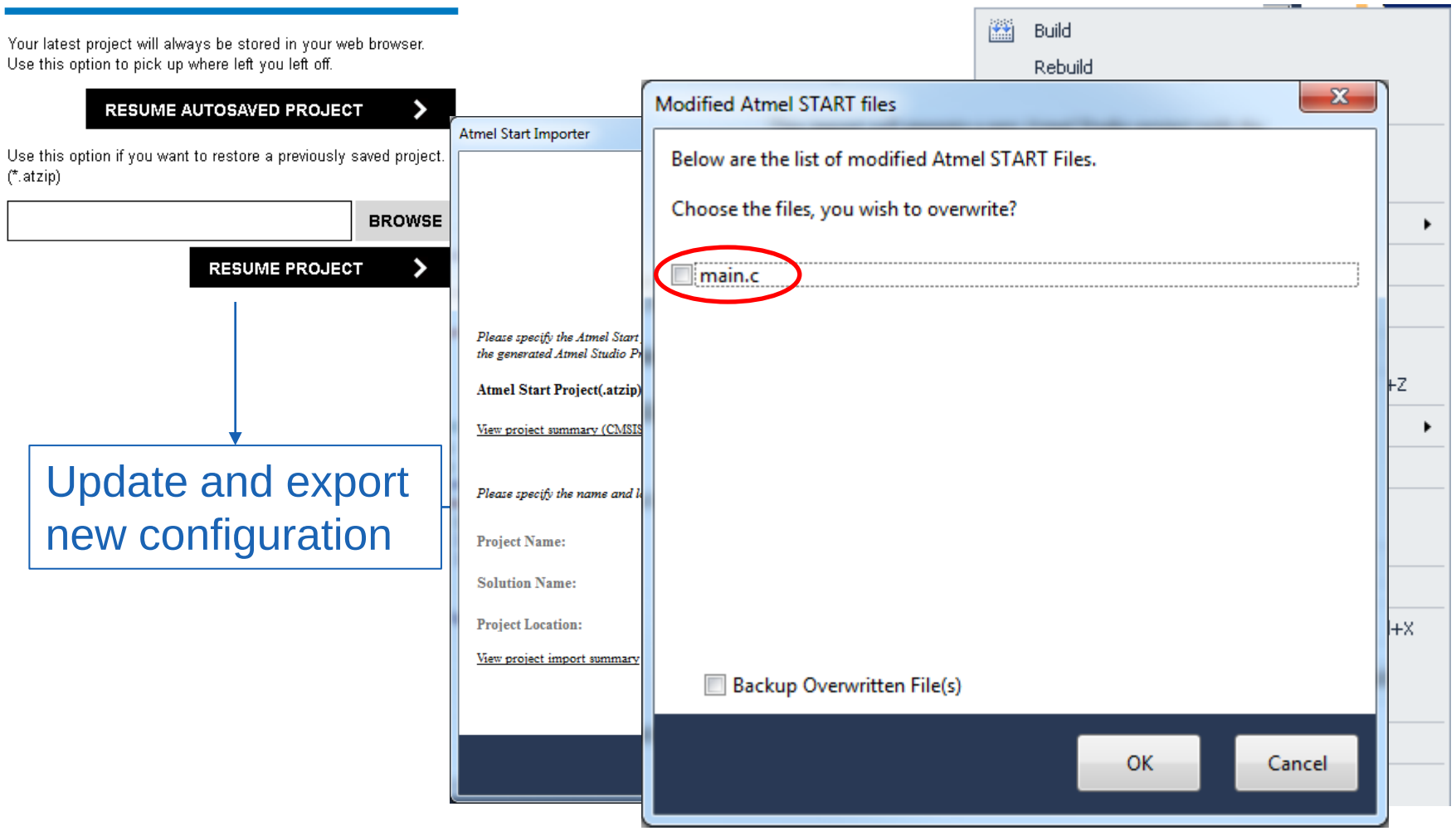
Update and export new configuration



Re-configure project

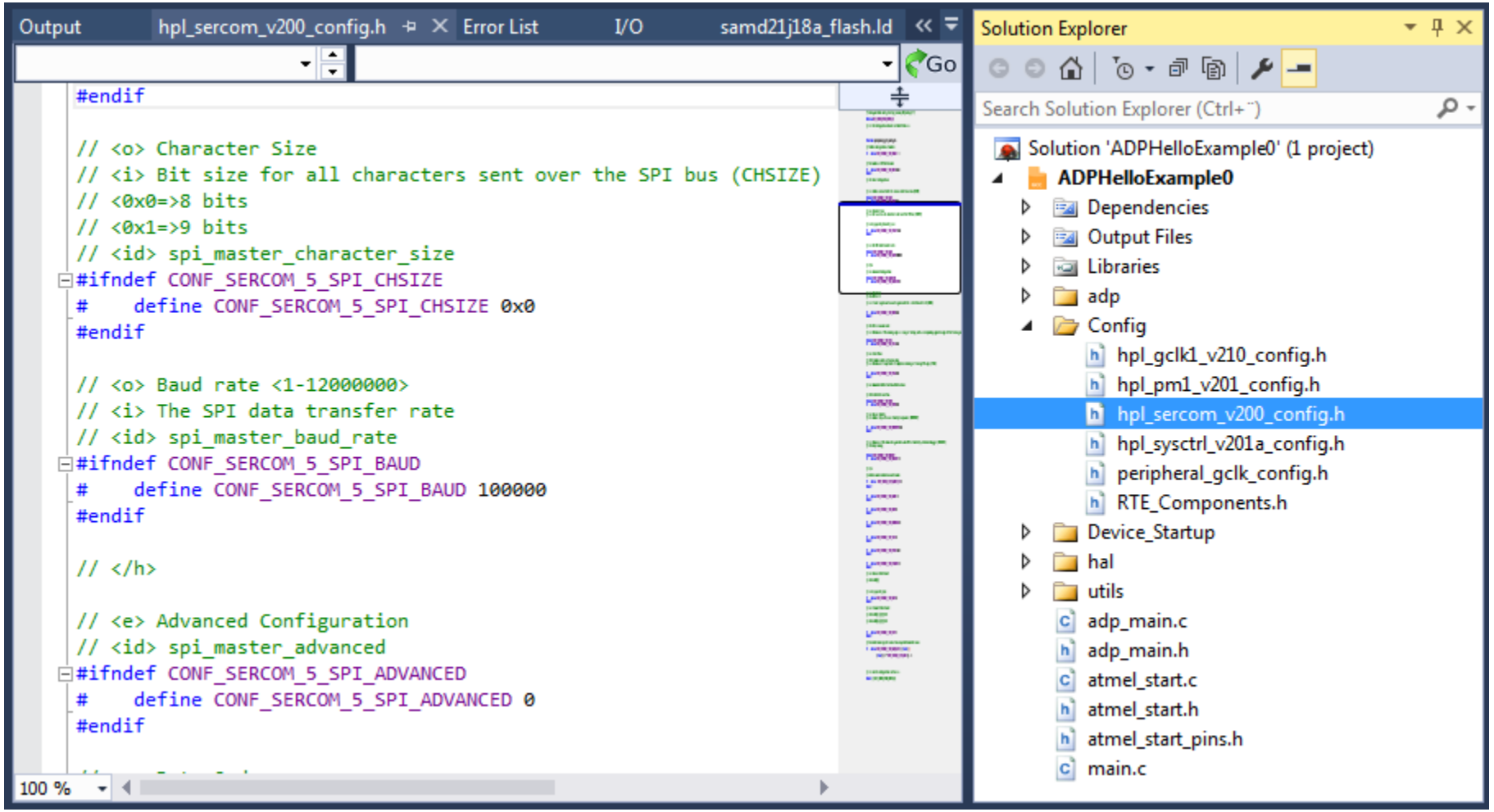


Re-configure project



CMSIS Configuration Anotation Notation

Config files: Can test new config (but will be overwritten)



Atmel Start and ASFv4

Agenda

- Introduction to Atmel Start
- **Atmel START** & **ASF4 vs. ASF Wizzard & ASF3**
  - Positioning & Messaging
  - Roadmap
  - Introduction to ASFv4
- Current work/Future plans

Messaging to existing customers

Why ASF4?

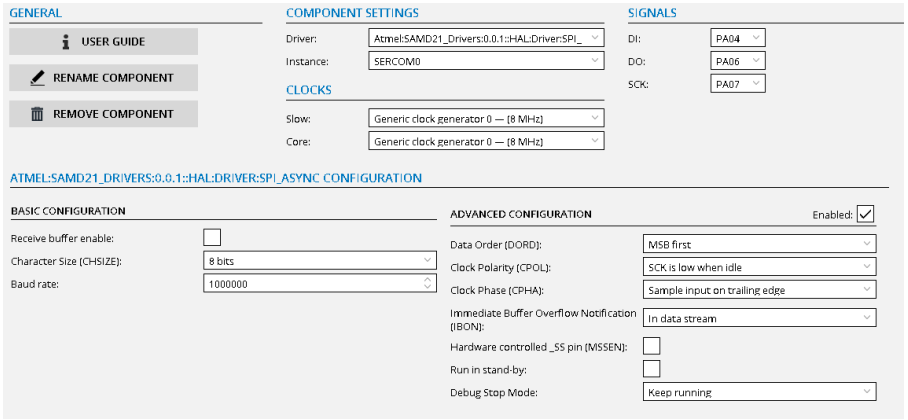
- The Atmel START Configuration tool replaces the ASF Wizard, the code generated is ASFv4.
- ASFv4 is a major upgrade to the Atmel Software framework, re-architected for low code footprint and low run-time performance overhead
- ASFv4 is not 100% compatible, though the code style will be familiar for users of ASFv3 API, supporting the Atmel SAMD/L/C MCU's.
- ASFv4 will be the main SW platform for new NPI work going forward.
  - Major new device platforms will only be supported on ASFv4.
  - Derivatives of devices on existing platforms may also be supported on ASFv3.
- Maintenance will continue on ASFv3, in line with longevity commitments for supported parts, but the platform will not be developed further.

ASF 3 vs. Atmel Start + ASF4

ASFv3

```
#include <asf.h>

struct spi_module spi;

int main (void)
{
    struct spi_config config;
    uint8_t example_text[] = "Hello World!";

    system_init();

    spi_get_config_defaults(&config);

    config.mux_setting = SPI_SIGNAL_MUX_SETTING_E;
    config.pinmux_pad0 = PINMUX_PA04D_SERCOM0_PAD0;
    config.pinmux_pad1 = PINMUX_PA05D_SERCOM0_PAD1;
    config.pinmux_pad2 = PINMUX_PA06D_SERCOM0_PAD2;
    config.pinmux_pad3 = PINMUX_PA07D_SERCOM0_PAD3;
    config.run_in_standby = false;
    config.mode_specific.master.baudrate = 1000000;
    config.select_slave_low_detect_enable = false;

    spi_init(&spi, SERCOM0, &config);
    spi_enable(&spi);
    spi_write_buffer_job(&spi, example_text, 12);

    while (1)
    {
        //TODO:: Please write your application code
    }
}
```

Atmel Start + ASFv4




```
#include "atmel_start.h"
#include "atmel_start_pins.h"

int main(void)
{
    uint8_t example_test[] = "Hello World!";
    struct io_descriptor *spi_io;

    system_init();

    spi_m_sync_enable(&SPI_0);
    spi_m_sync_get_io_descriptor(&SPI_0, &spi_io);

    io_write(spi_io, example_test, 12);

    while(1) {
    }
}
```

ASF 3 vs. Atmel Start + ASF4

ASFv3

Runtime configuration: In the project

- Configuration structs must be populated
- Configuration dependencies resolved and calculation done **run-time** which add code to the project

Generic drivers

- Tries to support **all features** in a module
- Feature list changes with target device
- #ifdefs used to enable/disable code depending on features supported by the device
- One TC driver

Common peripheral APIs

- Usually wrappers on top of other drivers
- Different function signatures

ASFv4

Pre-compile configuration: Atmel Start

- Configuration structs populated **compile time**
- Configuration dependencies resolved and calculations done **compile time**

Use-case driven drivers

- Supports a subset of features
- Feature list is the same on any device
- Driver code much more **tailored to application** target
- Separate drivers for PWM, Timer, Input capture

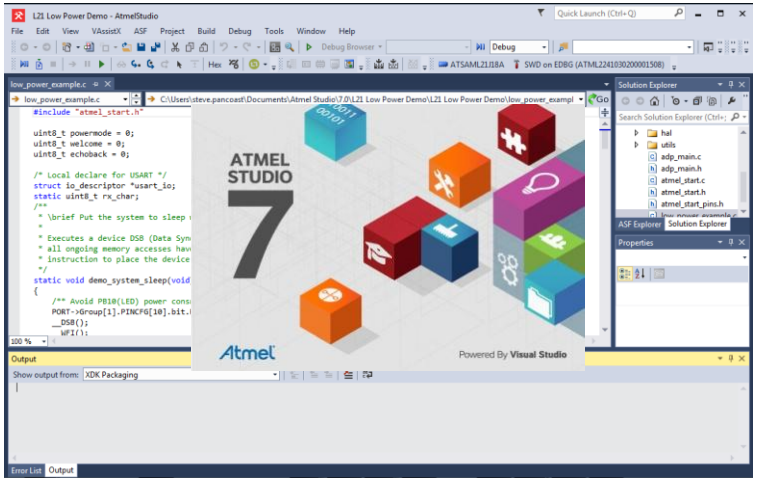
Common peripheral APIs

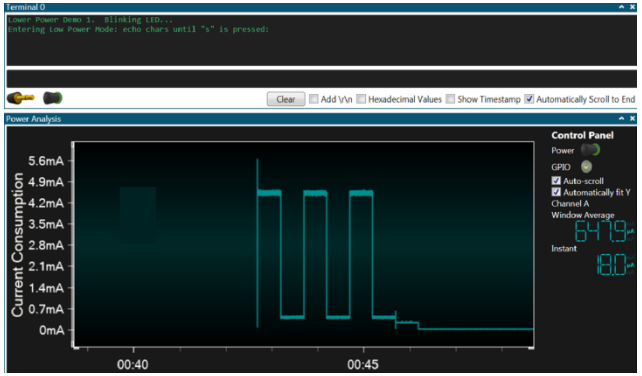
- All HAL APIs designed to be device agnostic
- HAL implementation not changed for any device
- (What common APIs should have been)

Atmel Studio 7

Focus is on Ease of Use & Performance

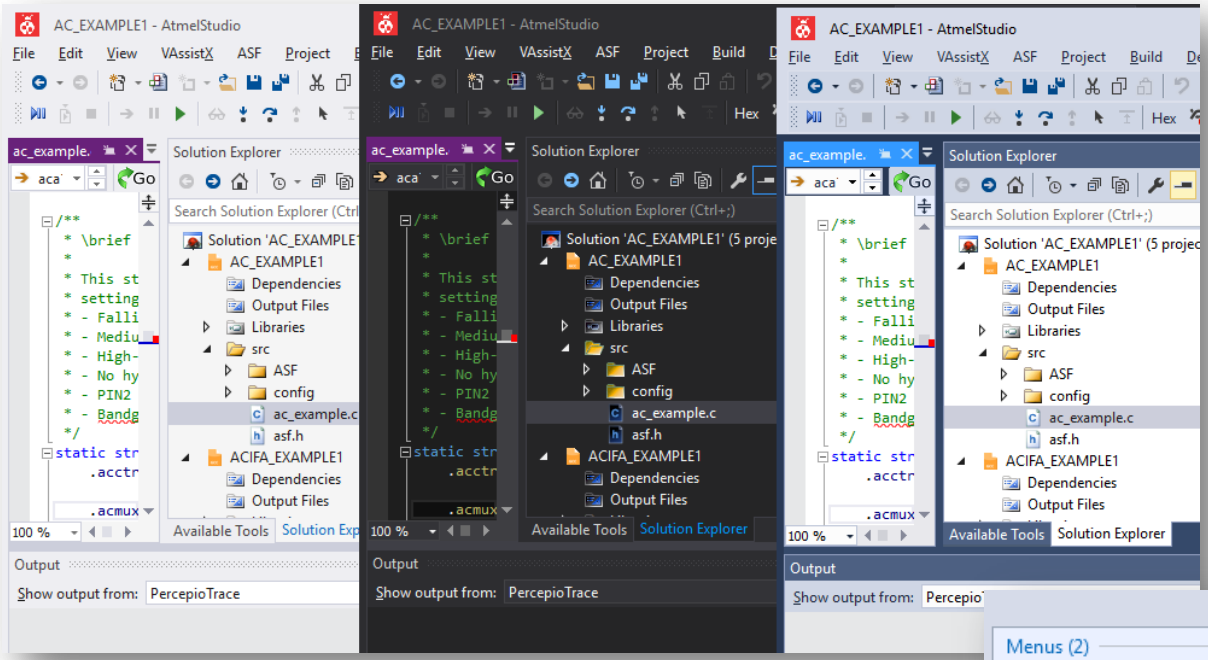
- **Ease of use Visualization Tools**
  - Support for Power visualization and Data visualization
  - Real time graph power - visually illustrating MCU power modes
  - Power Debugging
- **Smaller Download and Faster**
  - Modular download reduces download, web based updates and features
  - Latest MS Shell 2015 improves performance and startup times
- **Full Arduino Support "Maker to Market"**
  - Import Arduino sketches as C++ projects
  - Supports Arduino HW boards with EDBG
  - (Arduino IDE will also start supporting Atmel Explained Kits)



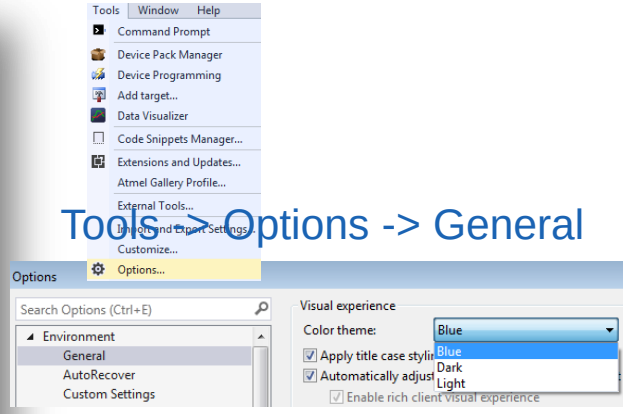


Platform update

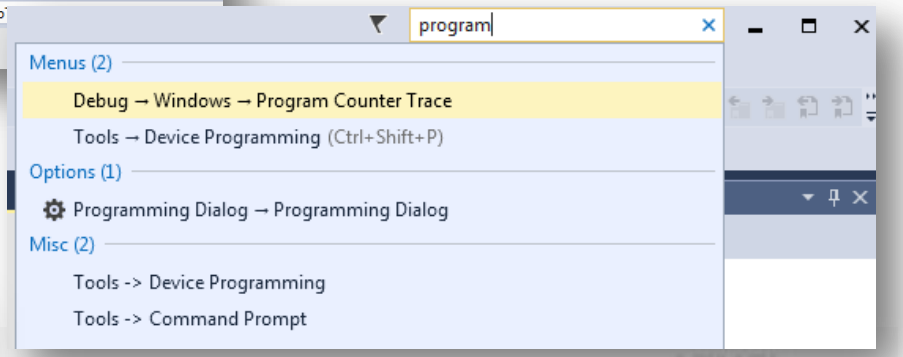
Chose theme: Dark, blue or light



Tools -> Options -> General

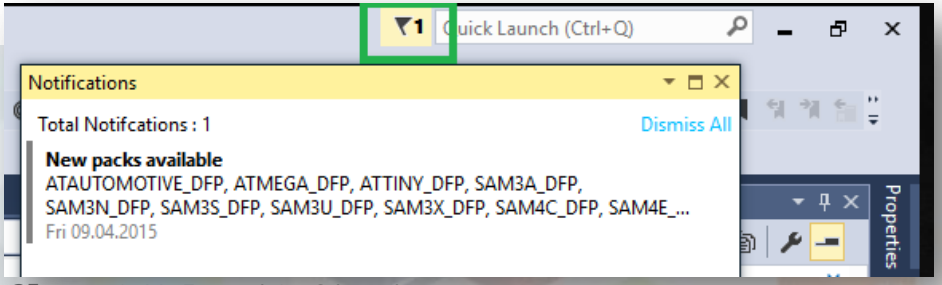


Quick launch



Get update notifications

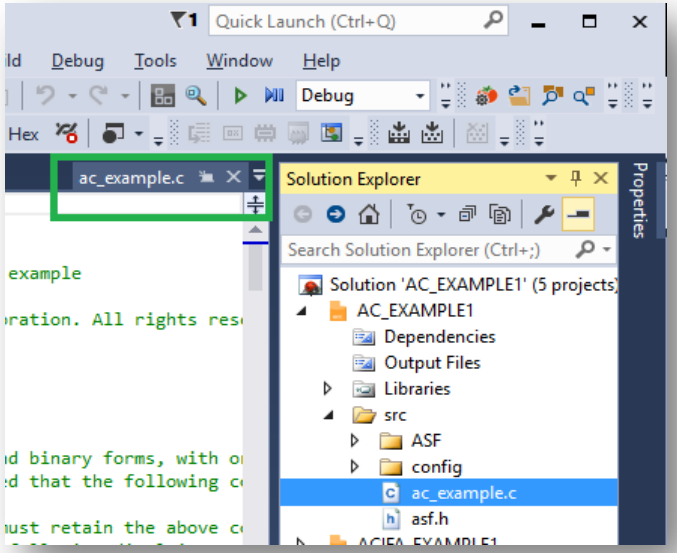
- New part packs availble



Platform update

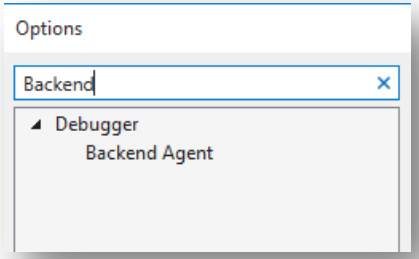
Preview selected item

- single click file item to preview in solution explorer

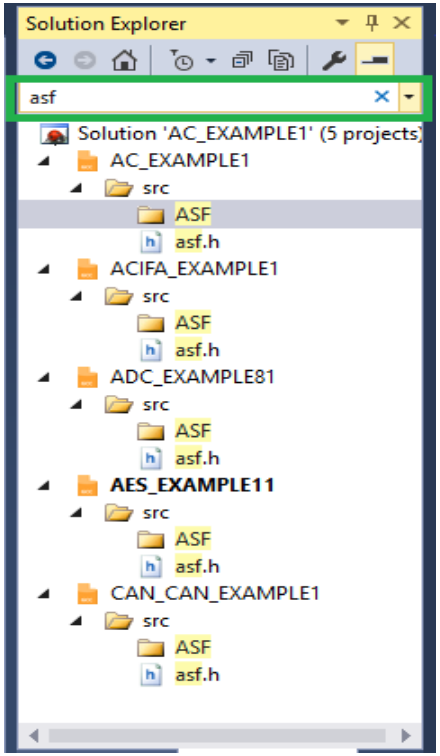


Search in:

- Tools->Options
- More easily find advanced config options

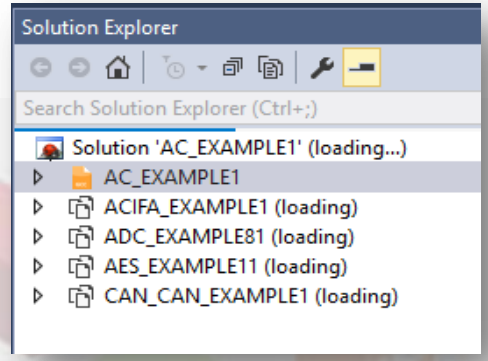


- Solution explorer



Asynchronous project load

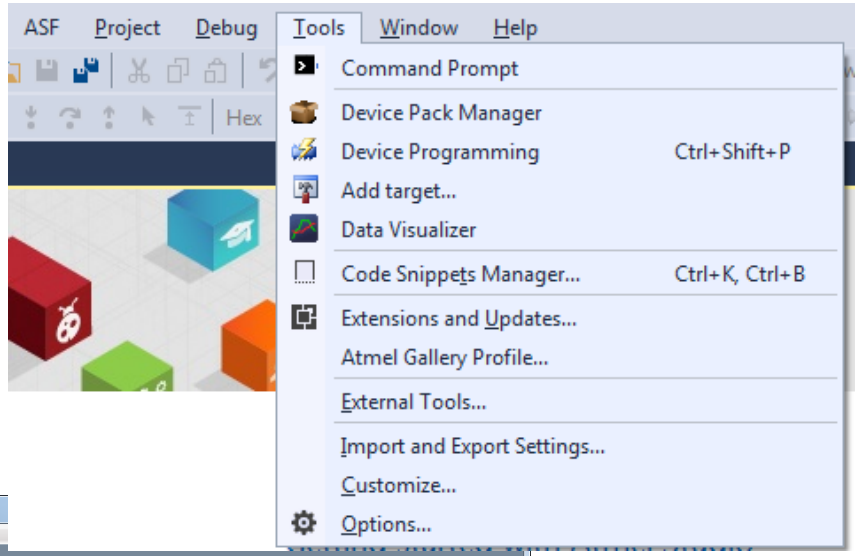
- When loading many projects part of a solution

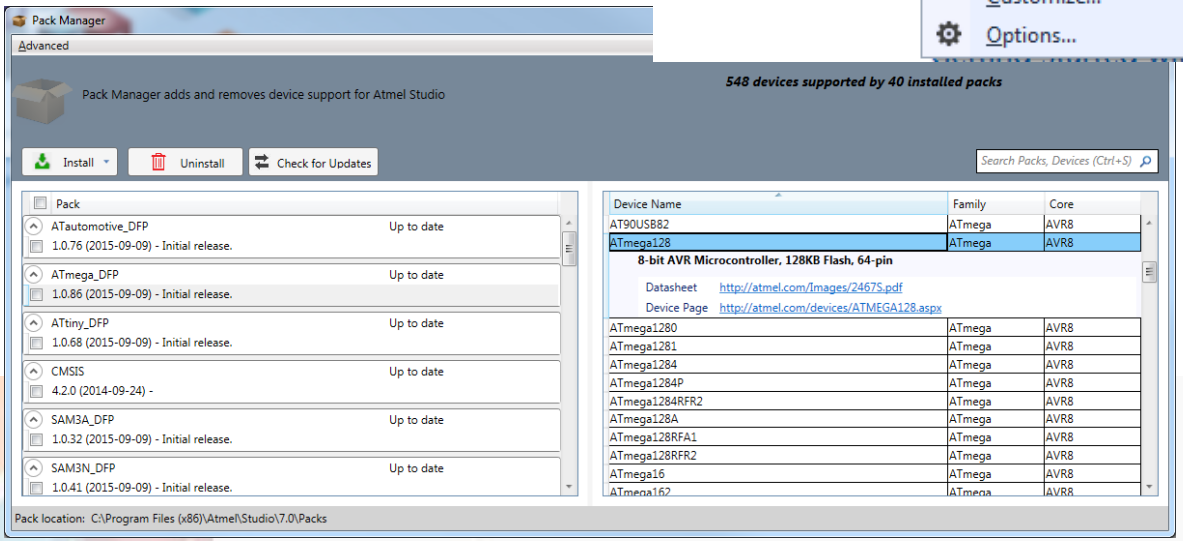


Pack Manager - New device support mechanism

Get support for the latest devices in Atmel Studio

- Pack Manager lists installed devices
- Offline support
  - Local packs can be added using the Pack Manager UI or the CLI
- Studio device selection dialogs lists only installed devices
- Check for updates





From Maker to Market

Ease path for Arduino users to move to Atmel Studio

- Import Arduino project into Studio
- Opened as a C++ project with include paths to the Arduino library source code