

Atmel[®]

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)

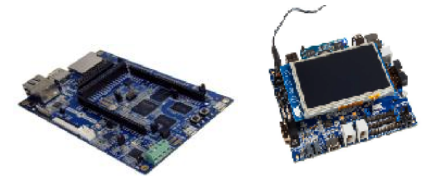
QT6

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 support



Atmel SW Development tools

Strongest MCU tools portfolio on the market

- **Atmel Studio**

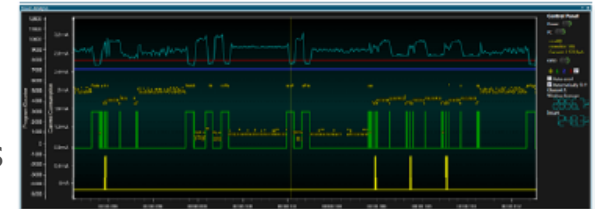
- Based on Visual Studio frontend, supports 8/32-bit AVR and ARM development & advanced debugging

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



- **Atmel Data Visualizer**

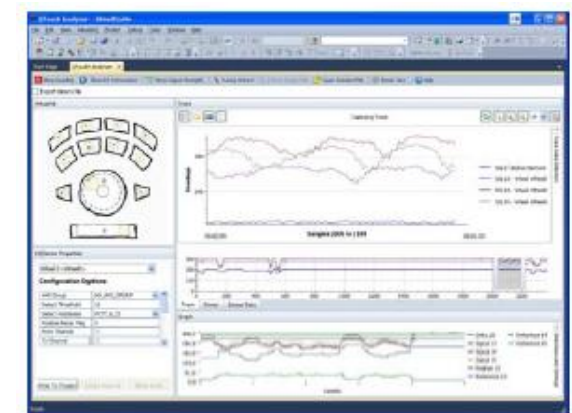
- Oscilloscope view allows you to Monitor application behavior, display data & interact with running applications



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

- **Q-Touch Studio & Wireless composer**

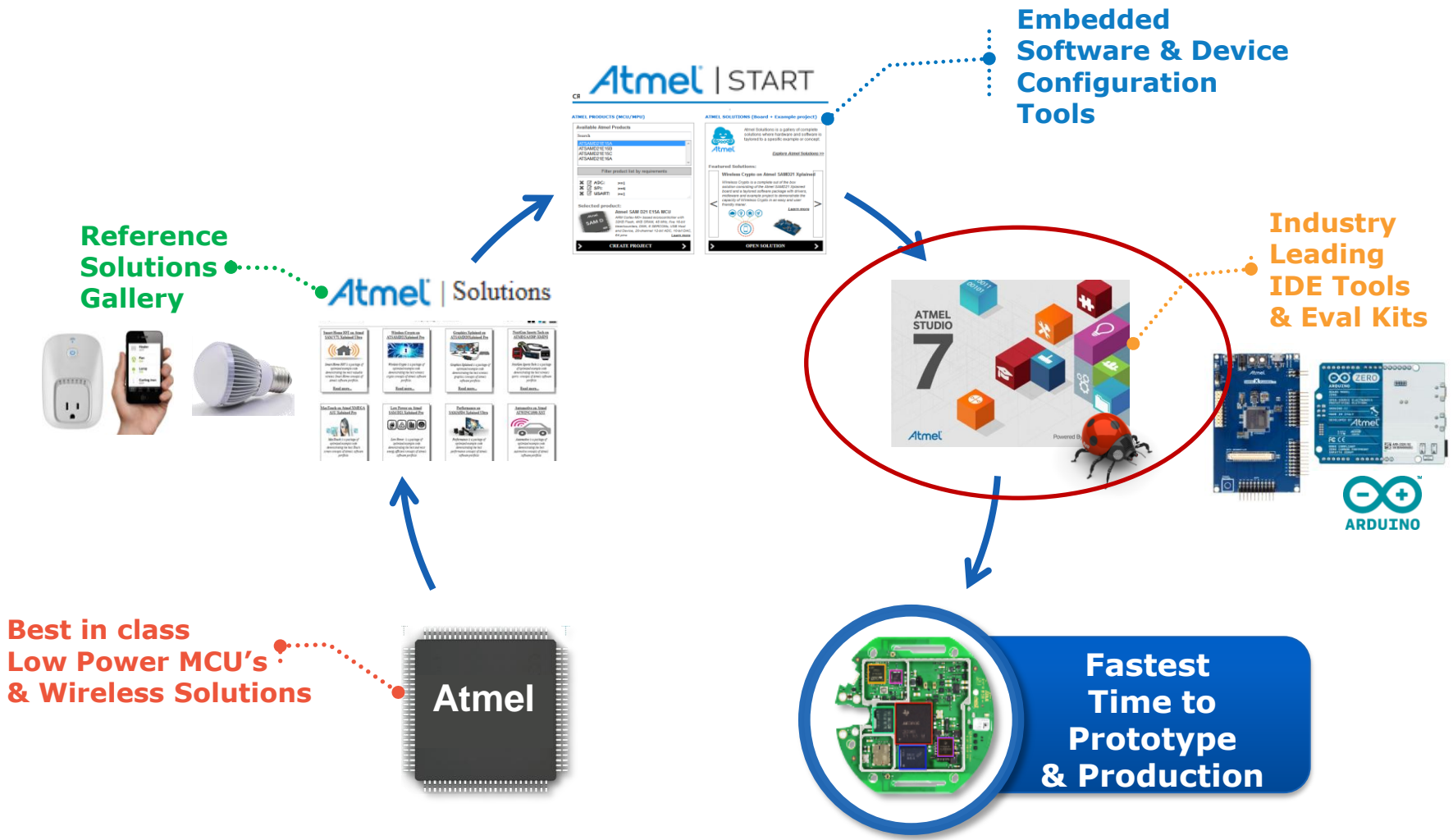
- Studio plug-ins for step-by-step guidance for setting up and testing touch & RF systems.



- **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 unattended installs.

Ease of use Visualization Tools

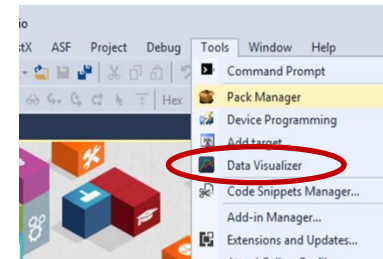
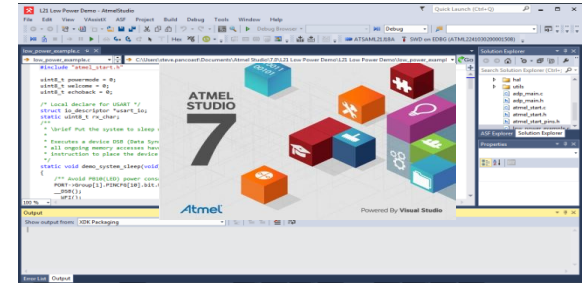
- Support for Power visualization and Data visualization
- Real time graph power - visually illustrating MCU power modes
- Power Debugging

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>



Welcome to Atmel Start BETA

This tool will help you select and configure software components, drivers, middleware and example projects to tailor your embedded application in a usable and optimized manner.



The workflow is quite straight forward: Filter MCUs by requirements before starting a project. Next you add components to your project, configure each component at will, export or copy the component, and paste them into your favourite IDE for further development.

For more information, see our help section or feel free to use the tool tips in the different parts of the tool.

CREATE NEW PROJECT

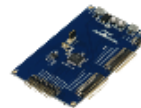
Select a board or a device directly or by filtering the list by hardware or software requirements.

Add Requirements

Filter on board name...	Filter on device name...
Custom board	ATSAMD21J16B
SAM C21 Xplained Pro	ATSAMD21J17A
SAM D10 Xplained Mini	ATSAMD21J17A
SAM D11 Xplained Pro	ATSAMD21J17A
SAM D20 Xplained Pro	ATSAMD21J17A
SAM D21 Xplained Pro	ATSAMD21J18A
SAM L21 Xplained Pro	ATSAMD21J18A

SAM D21 Xplained Pro

The Atmel® | SMART™ SAM D21 Xplained Pro evaluation kit is a hardware platform to evaluate the Atmel ATSAMD21J18A microcontroller. Supported by the Atmel Studio integrated development...



More >

CREATE NEW PROJECT >

BROWSE EXAMPLES

Browse the gallery of complete examples, where hardware and software is tailored to specific example or concept.

ADP Hello Example

This demo uses the ADP usage example function to write data to the Data Visualizer through EDBG DGI.

Browse All Examples >

OPEN EXAMPLE >

LOAD EXISTING PROJECT

LOAD PROJECT FROM FILE >

OR

RESUME AUTOSAVED PROJECT >

Use this option if you want to restore a locally saved project. Browse and select

Your latest project will always be stored in your web browser. Use this option to

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.

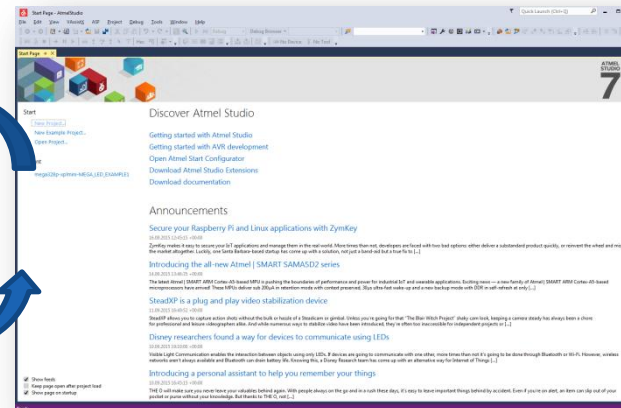
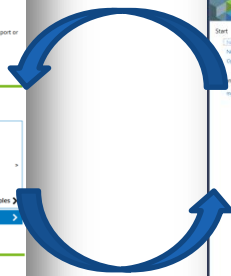
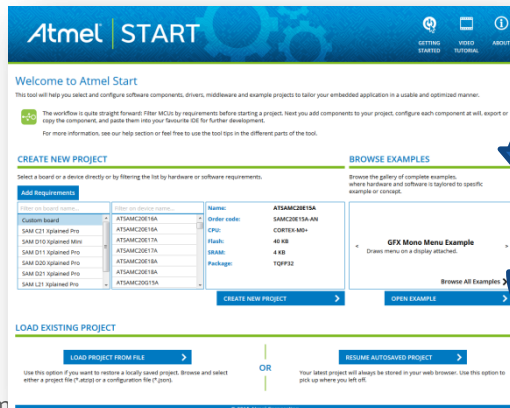
Explore software components, examples and complete reference solutions

- 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

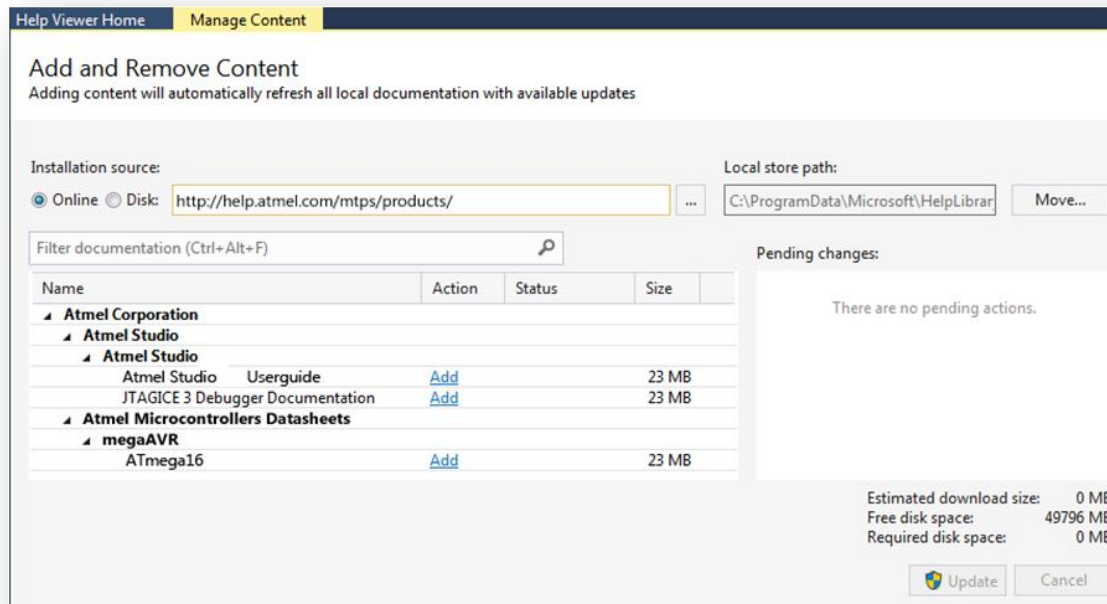
IDE agnostic - supports code/project generation for Studio, IAR and Keil



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

The screenshot displays the Atmel Studio 7 interface. The main window is split into two panes. The left pane is the code editor, showing a C program named `tc_timeout_example2.c`. The code includes a `#define` for `TCCR1A_SFR_IO8(0x2F)` and a `main` function that initializes a timer and enters a loop. The cursor is positioned at the `TCCR1A =` line. The right pane is the IO View, which lists various hardware registers. The `TCCR1A` register is selected and highlighted in blue. Below the list is a bit-field editor for the selected register, showing its address (0x4F), value (0x00), and bit patterns.

Annotations:

- An arrow points to the `ATmega16` dropdown menu in the top right corner of the IDE, with the text "ATmega16 is the context".
- An arrow points to the cursor in the code editor, with the text "Place cursor, press F1".
- An arrow points to the `TCCR1A` register in the IO View, with the text "Select register and press F1".

Name	Address	Value	Bits
EXTERNAL_INTERRUPT			
JTAG			
PORTA			
PORTB			
PORTC			
PORTD			
SPI			
TIMER_COUNTER_0			
TIMER_COUNTER_1			
Prescaler source of Tim... N. 0x00			
TIMER_COUNTER_2			
TWI			
USART			
ICR1	0x46	0x0000	00000000 00000000
OCR1B	0x48	0x0000	00000000 00000000
OCR1A	0x4A	0x0000	00000000 00000000
TCNT1	0x4C	0x0000	00000000 00000000
TCCR1B	0x4E	0x00	00000000 00000000
TCCR1A	0x4F	0x00	00000000 00000000
TIFR	0x58	0x00	00000000 00000000
TIMSK	0x59	0x00	00000000 00000000

Improvement on Atmel Studio 7

Help System Improvement (3)

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

Timer/Counter1 Control Register A – TCCR1A

Bit	7	6	5	4	3	2	1	0	
	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	WGM11	WGM10	TCCR1A
Read/Write	R/W	R/W	R/W	R/W	W	W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7:6 – COM1A1:0: Compare Output Mode for Channel A**
- **Bit 5:4 – COM1B1:0: Compare Output Mode for Channel B**

The COM1A1:0 and COM1B1:0 control the Output Compare pins (OC1A and OC1B respectively) behavior. If one or both of the COM1A1:0 bits are written to one, the OC1A output overrides the normal port functionality of the I/O pin it is connected to. If one or both of the COM1B1:0 bit are written to one, the OC1B output overrides the normal port functionality of the I/O pin it is connected to. However, note that the *Data Direction Register* (DDR) bit corresponding to the OC1A or OC1B pin must be set in order to enable the output driver.

When the OC1A or OC1B is connected to the pin, the function of the COM1x1:0 bits is dependent of the WGM13:0 bits setting. [Table 44](#) shows the COM1x1:0 bit functionality when the WGM13:0 bits are set to a normal or a CTC mode (non-PWM).

Table 44. Compare Output Mode, non-PWM

COM1A1/COM1B1	COM1A0/COM1B0	Description
0	0	Normal port operation, OC1A/OC1B disconnected.
0	1	Toggle OC1A/OC1B on compare match
1	0	Clear OC1A/OC1B on compare match (Set output to low level)
1	1	Set OC1A/OC1B on compare match (Set output to high level)

[Table 45](#) shows the COM1x1:0 bit functionality when the WGM13:0 bits are set to the fast PWM mode.

Table 45. Compare Output Mode, Fast PWM⁽¹⁾

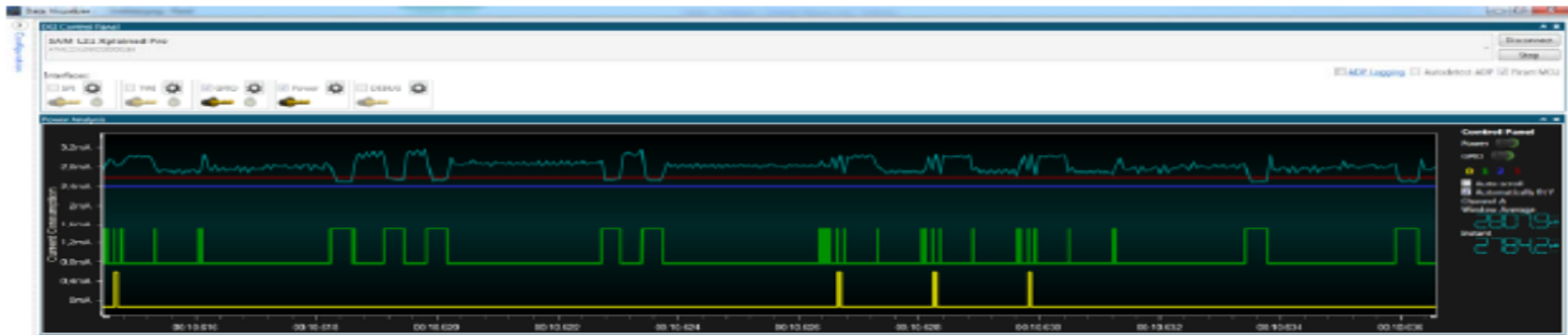
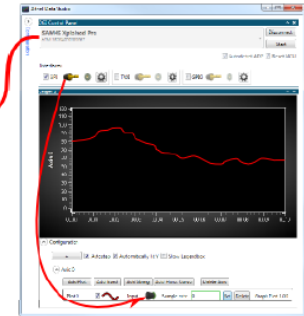
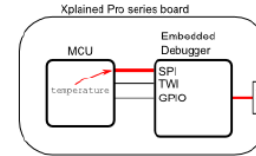
COM1A1/COM1B1	COM1A0/COM1B0	Description
0	0	Normal port operation, OC1A/OC1B disconnected.

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

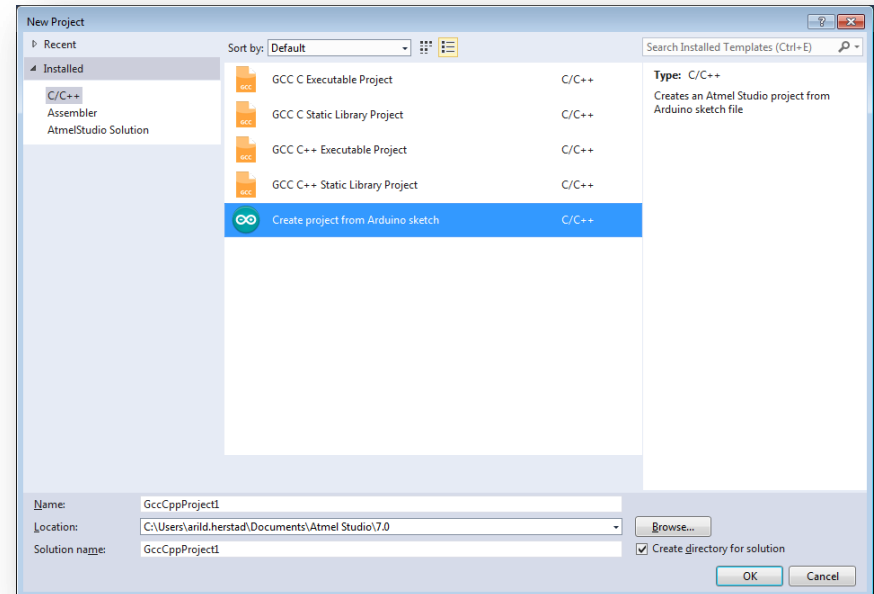


From Maker to Market

Ease path for Arduino users to move to Atmel Studio

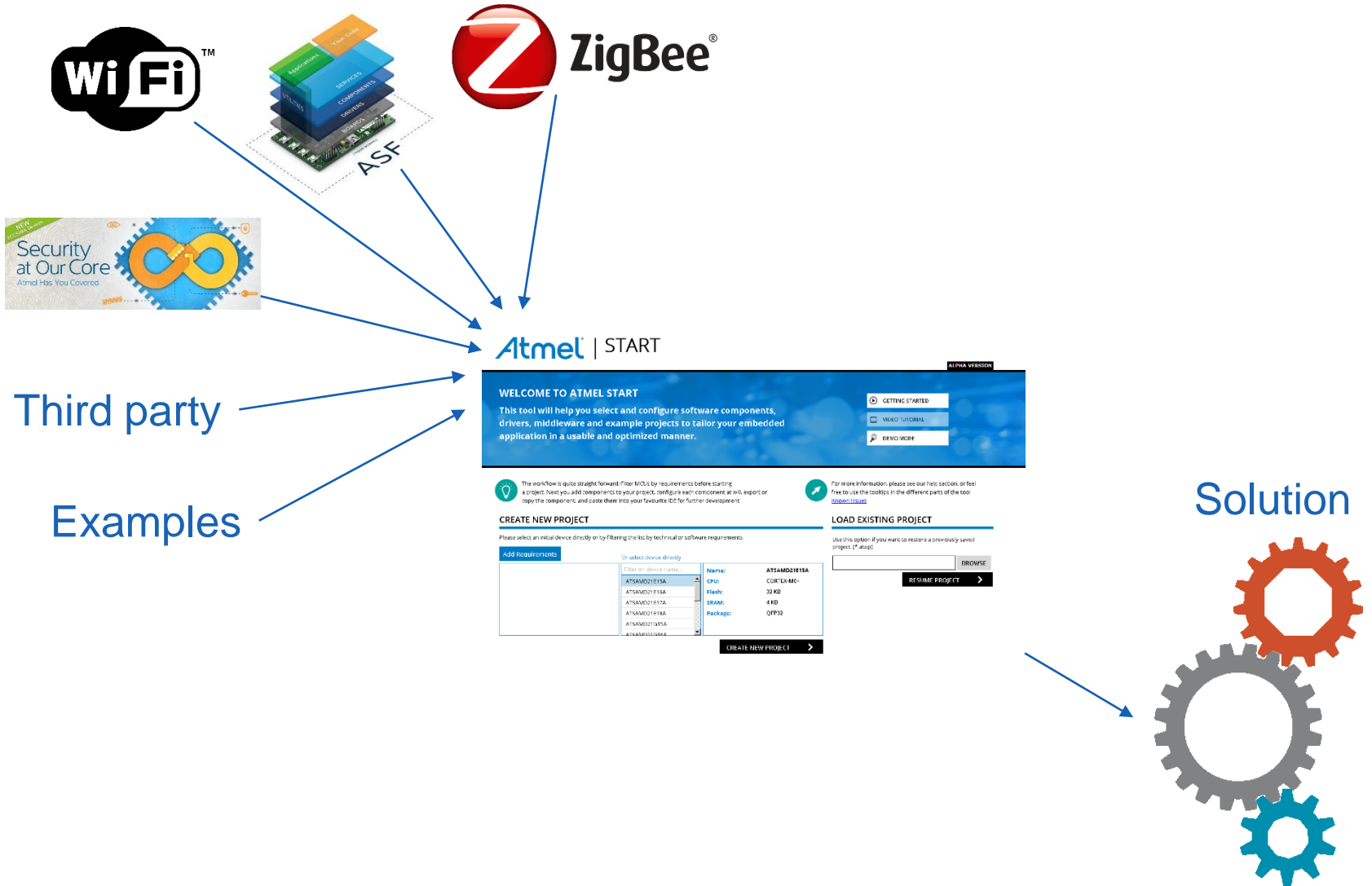


- Import Arduino project into Studio
- Opened as a C++ project with include paths to the Arduino library source code



Atmel Start

A hub for software examples and solutions



Dashboard and software configuration

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

MY SOFTWARE COMPONENTS



EXAMPLE

GFX Mono Menu Example

MIDDLEWARE

Monochrome Text(0) | Monochrome Font(0) | Monochrome Graphics(0) | Monochrome Display(0) | Monochrome Widget(0) | Monochrome Display Con...(0)

DRIVERS

External IRQ(0) | SPI(0)

SPI(0)

Serial Peripheral Interface (SPI) master communication in synchronous/blocking mode

GENERAL

USER GUIDE | RENAME COMPONENT | REMOVE COMPONENT

COMPONENT SETTINGS

Driver: HAL:Driver:SPI Master Sync | Instance: SERCOM5

CLOCKS

Slow: Generic clock generator 0 | Core: Generic clock generator 0

SIGNALS

MISO: PB16 | MOSI: PB22 | SCK: PB23

ATMEL:SAML21_DRIVERS:0.0.1::HAL:DRIVER:SPI_MASTER_SYNC CONFIGURATION

BASIC CONFIGURATION

Receive buffer enable: | Character Size: 8 bits

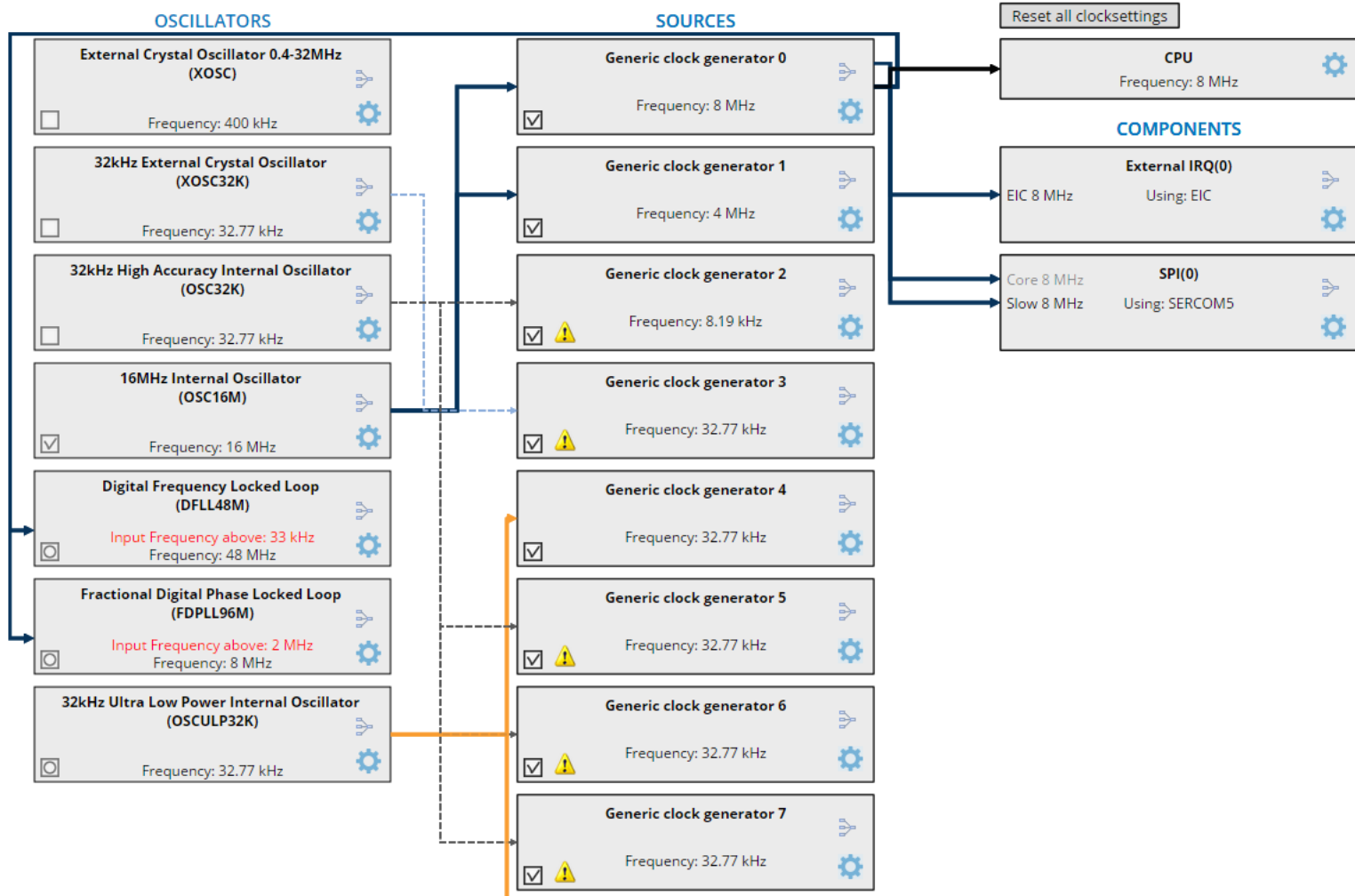
ADVANCED CONFIGURATION

Enable:


Configure the clocks


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


CLOCK CONFIGURATOR





Configure the pinmux


RETURN TO DASHBOARD


VIEW CODE

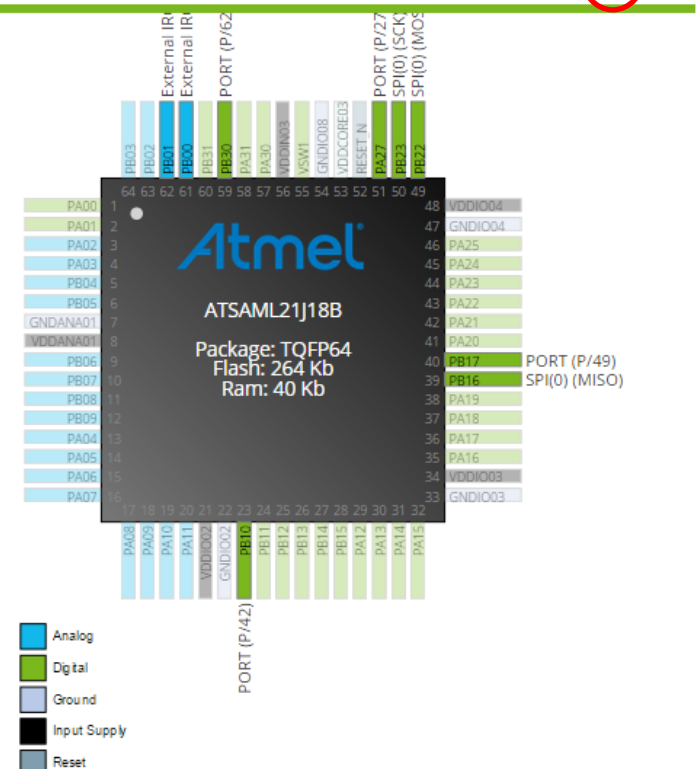

SAVE CONFIGURATION


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:

Pin mode:

Initial level:

Project preview and exporting

Project preview and exporting

 RETURN TO FRONT PAGE	 VIEW CODE	 SAVE CONFIGURATION	 EXPORT SOFTWARE COMPONENTS	 HELP AND SUPPORT
--	---	--	--	--

MY SOFTWARE COMPONENTS



```
PREVIEW - ATMEL_START.C
hal
  hpl
    gcl
      gpio_set_pin_mux(GPIO(GPIO_PORTA, 6), GPIO_MUX_D);
    }
  ser
  void SERCOM0_CLOCK_init(void)
  {
    _pm_enable_bus_clock(PM_BUS_APBC, SERCOM0);
    _gclk_enable_channel(SERCOM0_GCLK_ID_CORE, CONF_GCLK_SERCOM0_CORE_SRC);
  }
  void SERCOM0_init(void)
  {
    SERCOM0_CLOCK_init();
    usart_sync_init(&SERCOM0_usart, SERCOM0);
    SERCOM0_PORT_init();
  }
/**
 * Example of using SERCOM0 to write "Hello World" using the IO abstraction.
 */
void SERCOM0_example(void)
{
  struct io_descriptor *io;
  usart_sync_get_io_descriptor(&SERCOM0_usart, &io);

  io->write(io, "Hello World!", 12);
}
```

Project preview and exporting

Project preview and exporting

RETURN TO FRONT PAGE	VIEW CODE	SAVE CONFIGURATION	EXPORT SOFTWARE COMPONENTS	HELP AND SUPPORT
----------------------	-----------	--------------------	----------------------------	------------------

MY SOFTWARE COMPONENTS



```
PREVIEW - ATMEL_START.C
```

```
hal
  hpl
    gcl
      gpio_set_pin_mux(GPIO(GPIO_PORTA, 6), GPIO_MUX_D);
    }
  ser
  sys
  atmel_sta
  atmel_sta
  other files

// <y> Initial level
// <false"> Low
// <true"> High
false);

void SERCOM0_CLOCK_init(void)
{
    _pm_enable_bus_clock(PM_BUS_APBC, SERCOM0);
    _gclk_enable_channel(SERCOM0_GCLK_ID_CORE, CONF_GCLK_SERCOM0_CORE_SRC);
}

void SERCOM0_init(void)
{
    SERCOM0_CLOCK_init();
    usart_sync_init(&SERCOM0_usart, SERCOM0);
    SERCOM0_PORT_init();
}

/**
 * Example of using SERCOM0 to write "Hello World" using the IO abstract:
 */
void SERCOM0_example(void)
{
    struct io_descriptor *io;
    usart_sync_get_io_descriptor(&SERCOM0_usart, &io);

    io->write(io, "Hello World!", 12);
}
```

Implementation examples included, in **ATMEL_START.c**

These examples & comments become your quickstart guide.

Project preview and exporting

Project preview and exporting

 RETURN TO FRONT PAGE	 VIEW CODE	 SAVE CONFIGURATION	 EXPORT SOFTWARE COMPONENTS	 HELP AND SUPPORT
---	--	---	---	---

MY SOFTWARE COMPONENTS



EXPORT SOFTWARE COMPONENTS

DOWNLOAD YOUR CONFIGURED COMPONENTS

Download a generated pack containing all your configured software components.

Use the Makefile-option if you want the generated pack to also include a standalone Makefile.

DOWNLOAD PACK 

Include Makefile:

Project preview and exporting

Project preview and exporting

 RETURN TO FRONT PAGE	 VIEW CODE	 SAVE CONFIGURATION	 EXPORT SOFTWARE COMPONENTS	 HELP AND SUPPORT
--	---	--	--	--

MY SOFTWARE COMPONENTS



EXPORT SOFTWARE COMPONENTS

DOWNLOAD YOUR CONFIGURED COMPONENTS

Download a generated pack containing all your configured software components.

Use the Makefile-option if you want the generated pack to also include a standalone Makefile.

DOWNLOAD PACK 

Include Makefile:

Re-configure project

LOAD EXISTING PROJECT

Your latest project will always be stored in your web browser.
Use this option to pick up where left you left off.

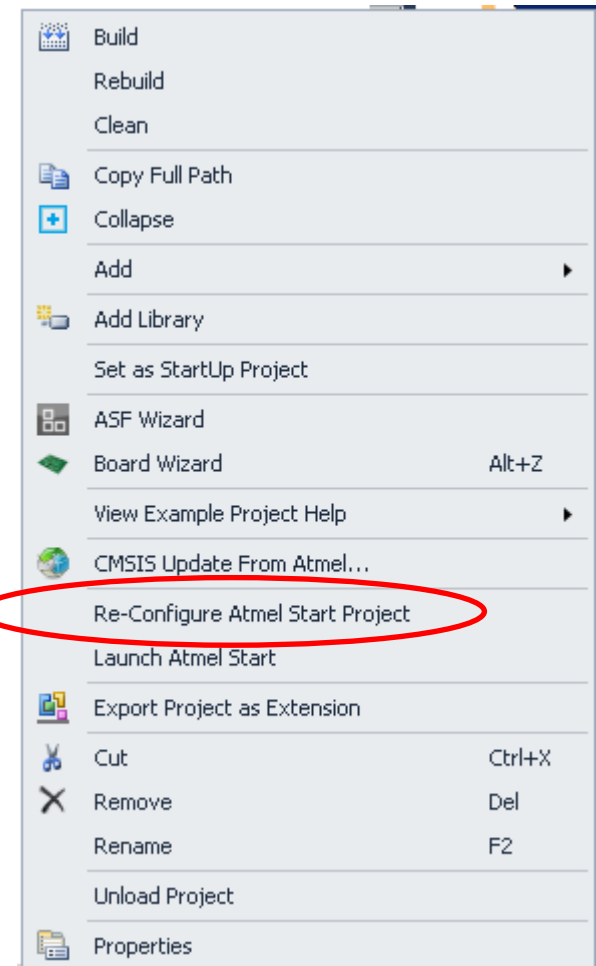
RESUME AUTOSAVED PROJECT >

Use this option if you want to restore a previously saved project.
(* .atzip)

BROWSE

RESUME PROJECT >

Update and export
new configuration



Re-configure project

LOAD EXISTING PROJECT

Your latest project will always be stored in your web browser. Use this option to pick up where left you left off.

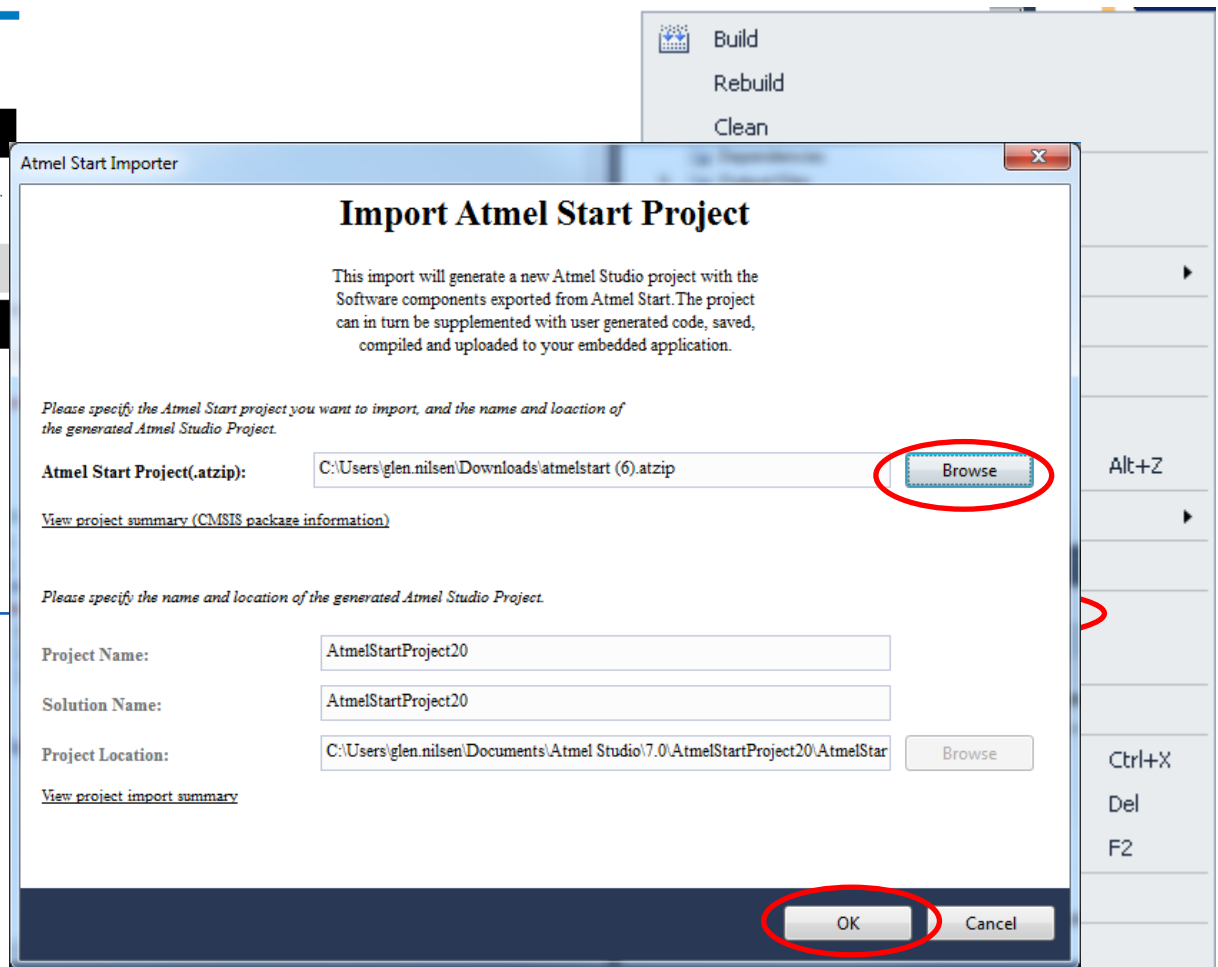
RESUME AUTOSAVED PROJECT >

Use this option if you want to restore a previously saved project. (*.atzip)

BROWSE

RESUME PROJECT >

Update and export new configuration



Re-configure project

LOAD EXISTING PROJECT

Your latest project will always be stored in your web browser. Use this option to pick up where left you left off.

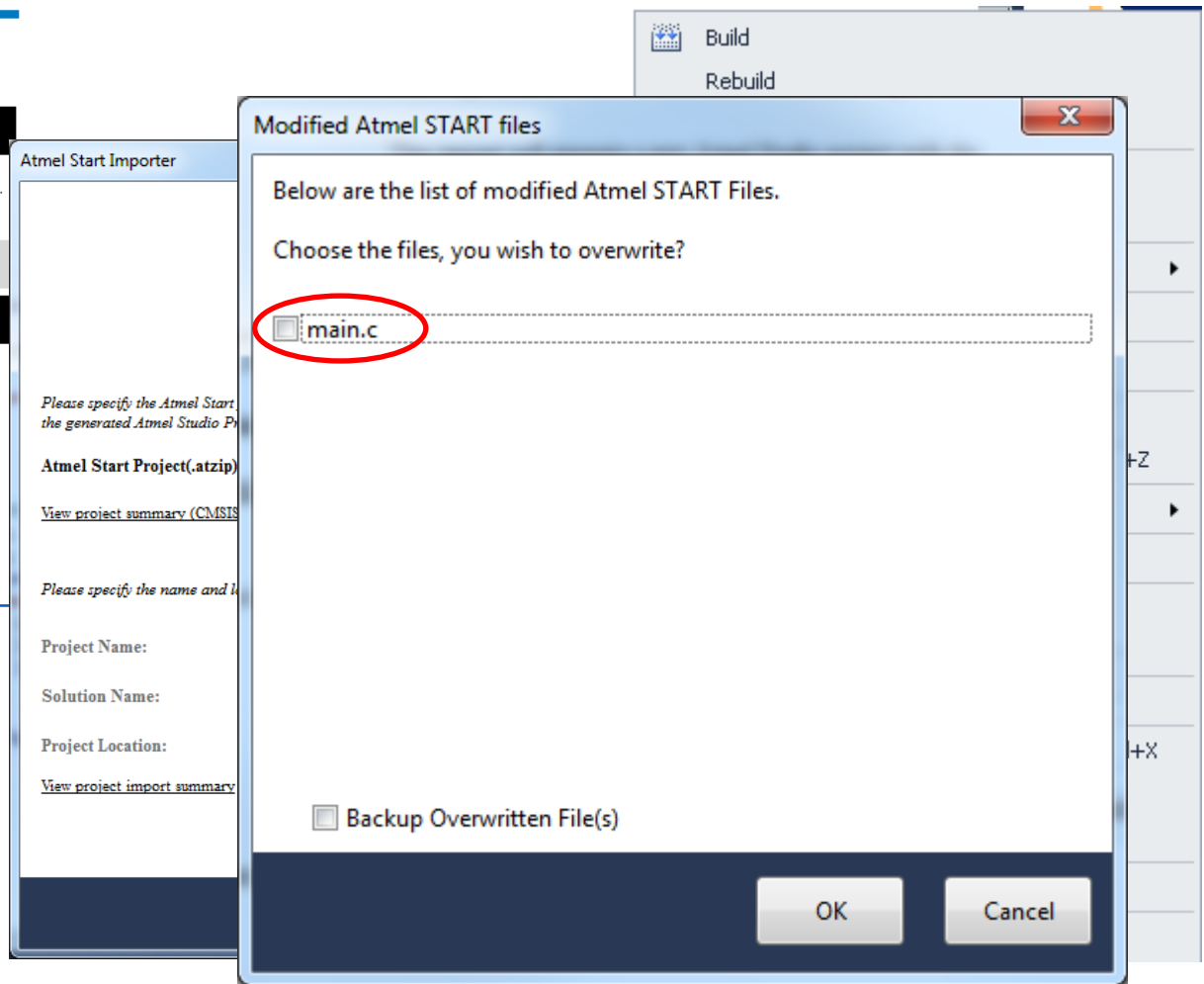
RESUME AUTOSAVED PROJECT >

Use this option if you want to restore a previously saved project. (*.atzip)

 BROWSE

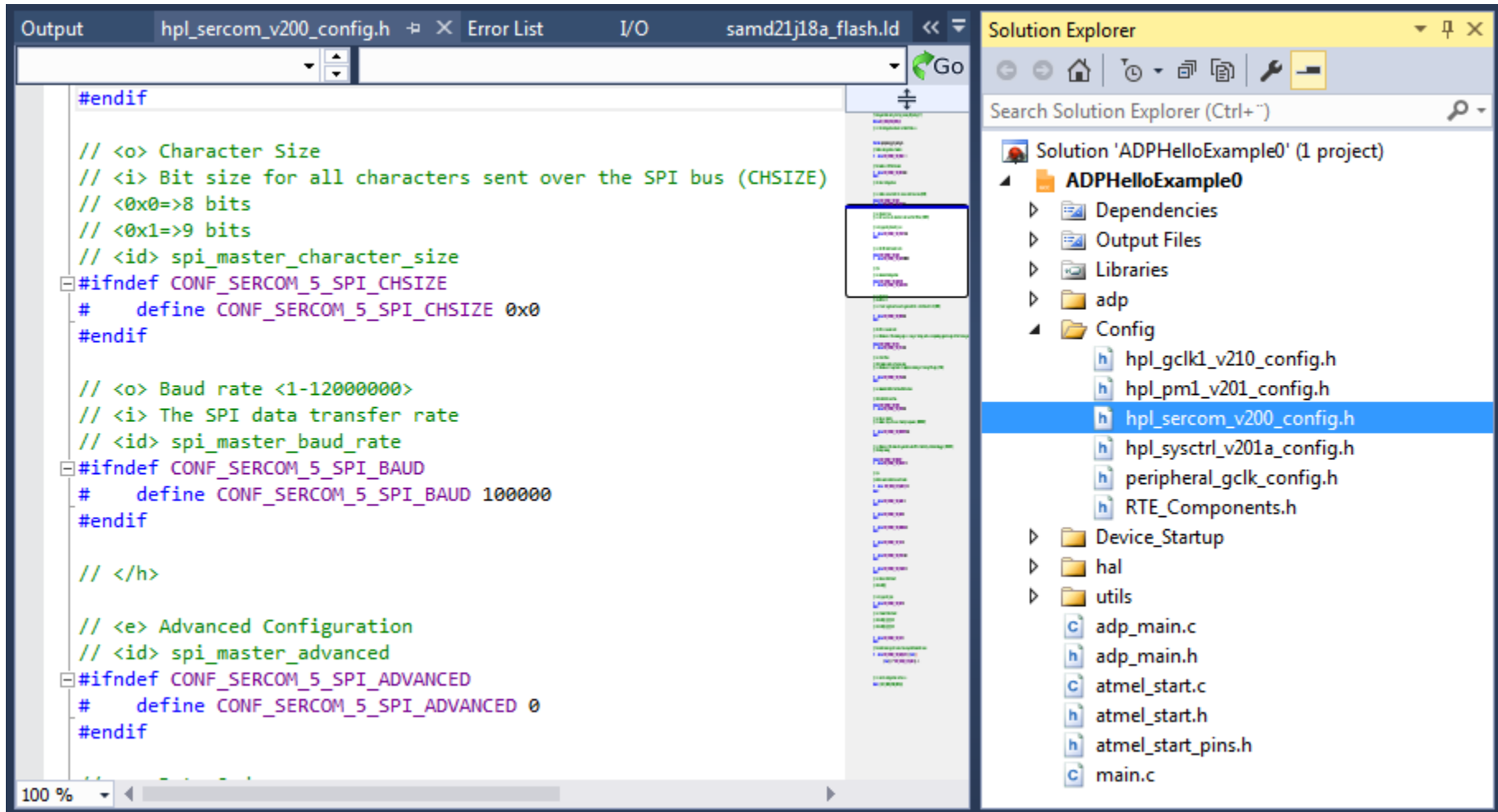
RESUME PROJECT >

Update and export new configuration



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

The screenshot shows the configuration interface for the 'ATMEL:SAMD21_DRIVERS:0.0.1:HAL-DRIVER:SPI_ASYNC CONFIGURATION' component. It is divided into several sections:

- GENERAL:** Includes 'USER GUIDE', 'RENAME COMPONENT', and 'REMOVE COMPONENT' buttons.
- COMPONENT SETTINGS:** Shows 'Driver' set to 'Atmel:SAMD21_Drivers:0.0.1:HAL-Driver:SPI_...', 'Instance' as 'SERCOM0', and 'CLOCKS' with 'Slow' and 'Core' both set to 'Generic clock generator 0 - (8 MHz)'. There are also 'SIGNALS' dropdowns for 'DI' (PA04), 'DO' (PA06), and 'SCK' (PA07).
- ATMEL:SAMD21_DRIVERS:0.0.1:HAL-DRIVER:SPI_ASYNC CONFIGURATION:** This section is further divided into:
 - BASIC CONFIGURATION:** Includes 'Receive buffer enable' (unchecked), 'Character Size (CHSIZE)' set to '8 bits', and 'Baud rate' set to '1000000'.
 - ADVANCED CONFIGURATION:** Includes 'Data Order (DORD)' set to 'MSB first', 'Clock Polarity (CPOL)' set to 'SCK is low when idle', 'Clock Phase (CPHA)' set to 'Sample input on trailing edge', 'Immediate Buffer Overflow Notification (IBON)' set to 'In data stream', 'Hardware controlled SS pin (MSEN)' (unchecked), 'Run in stand-by' (unchecked), and 'Debug Stop Mode' set to 'Keep running'.

```
#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**

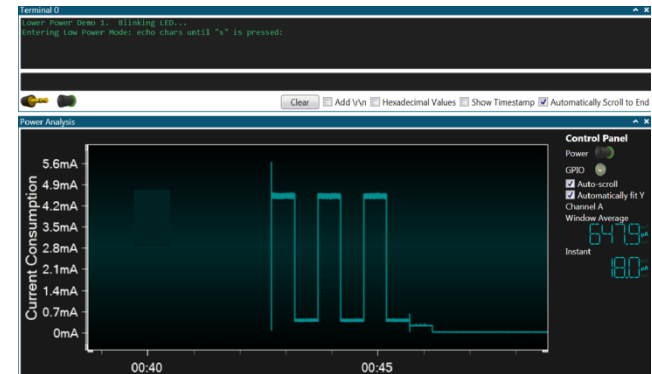
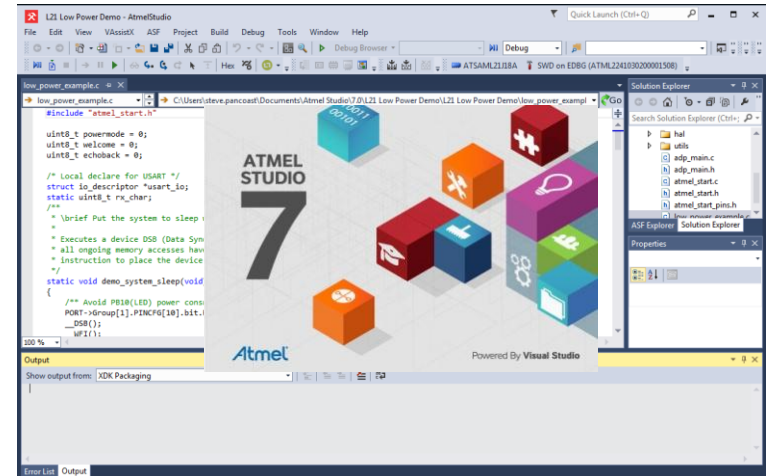
- Support for Power visualization and Data visualization
- Real time graph power - visually illustrating MCU power modes
- Power Debugging

- **Smaller Download and Faster**

- Modular download reduces download, web based updates and features
- Latest MS Shell 2015 improves performance and startup times

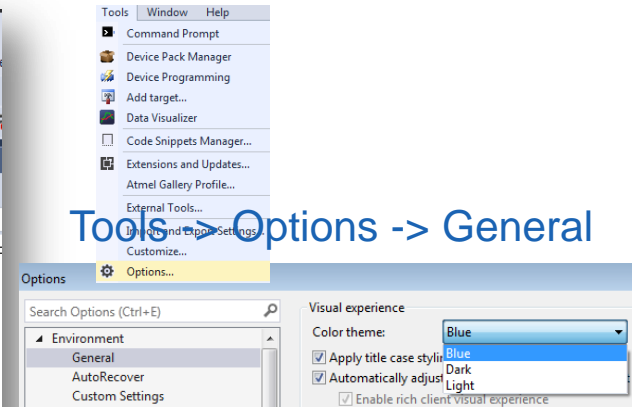
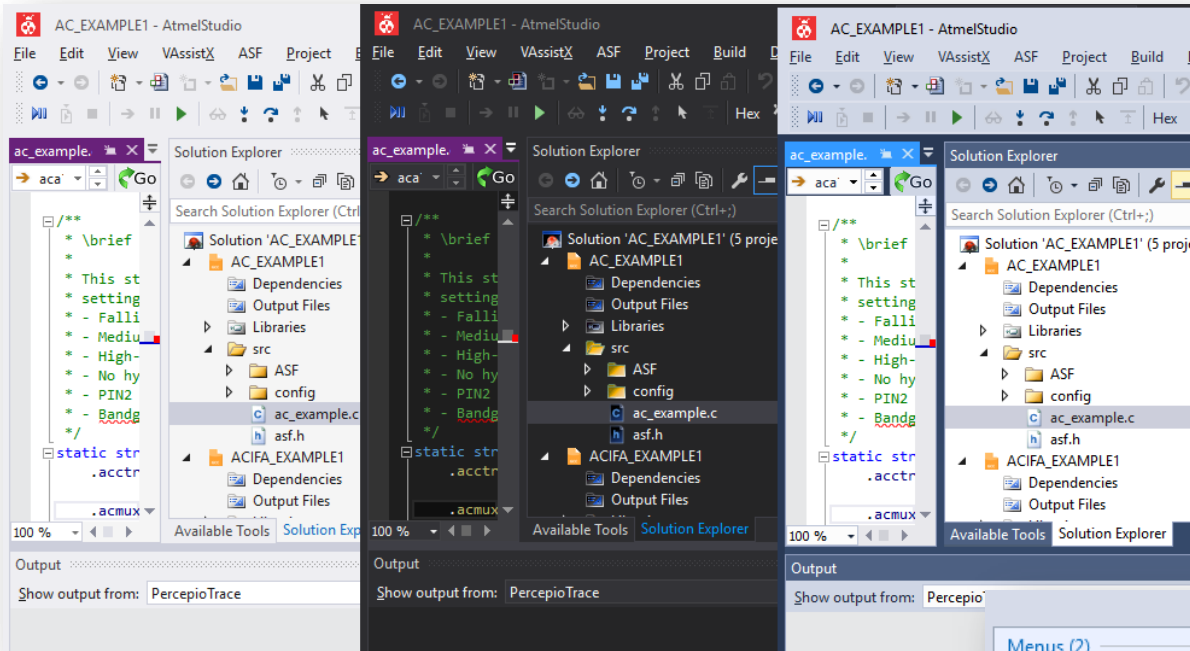
- **Full Arduino Support "Maker to Market"**

- Import Arduino sketches as C++ projects
- Supports Arduino HW boards with EDBG
- (Arduino IDE will also start supporting Atmel Explained Kits)



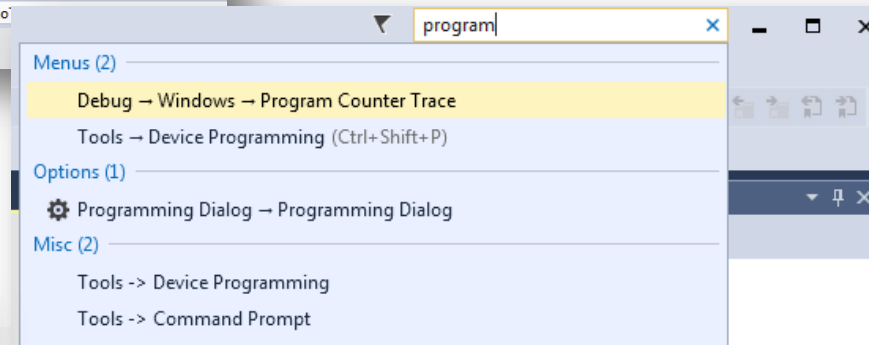
Platform update

Chose theme: Dark, blue or light



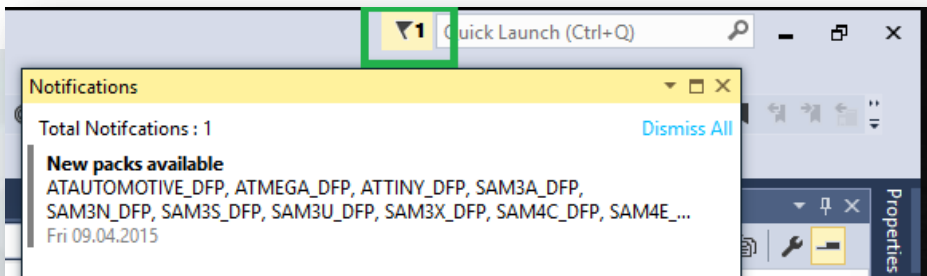
Tools -> Options -> General

Quick launch



Get update notifications

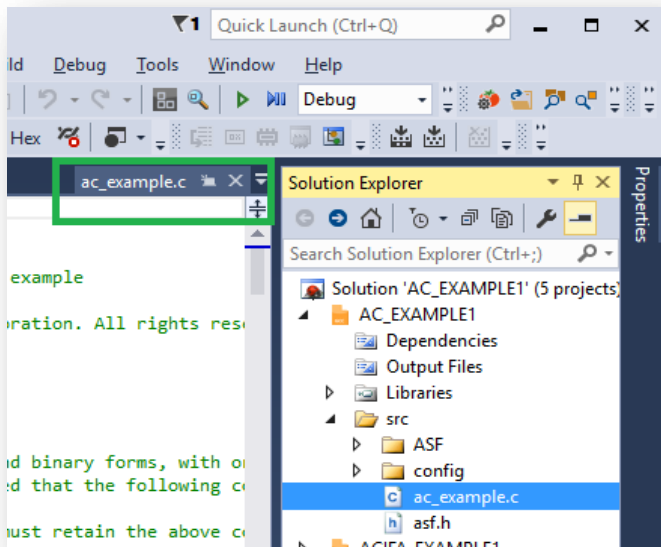
- New part packs available



Platform update

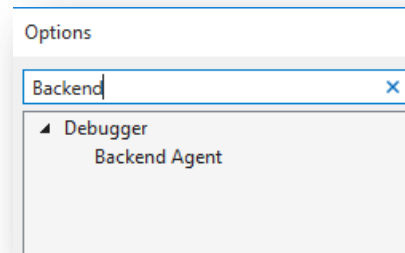
Preview selected item

- single click file item to preview in solution explorer

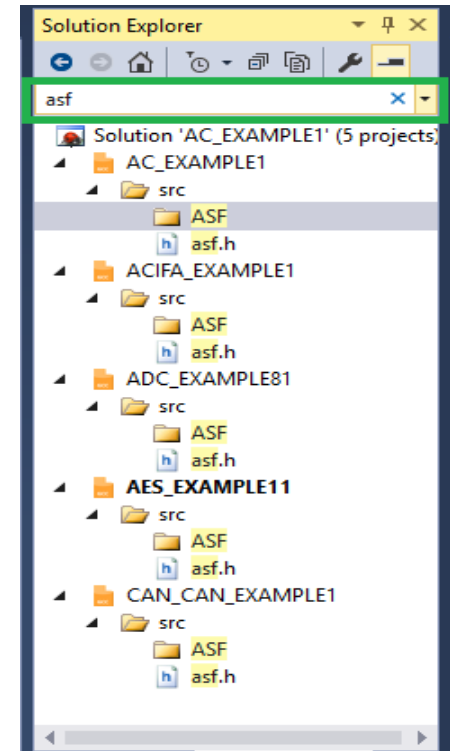


Search in:

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

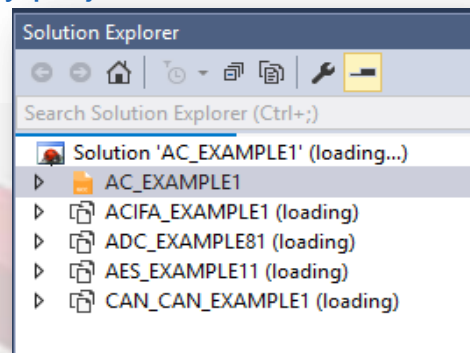


- Solution explorer



Asynchronous project load

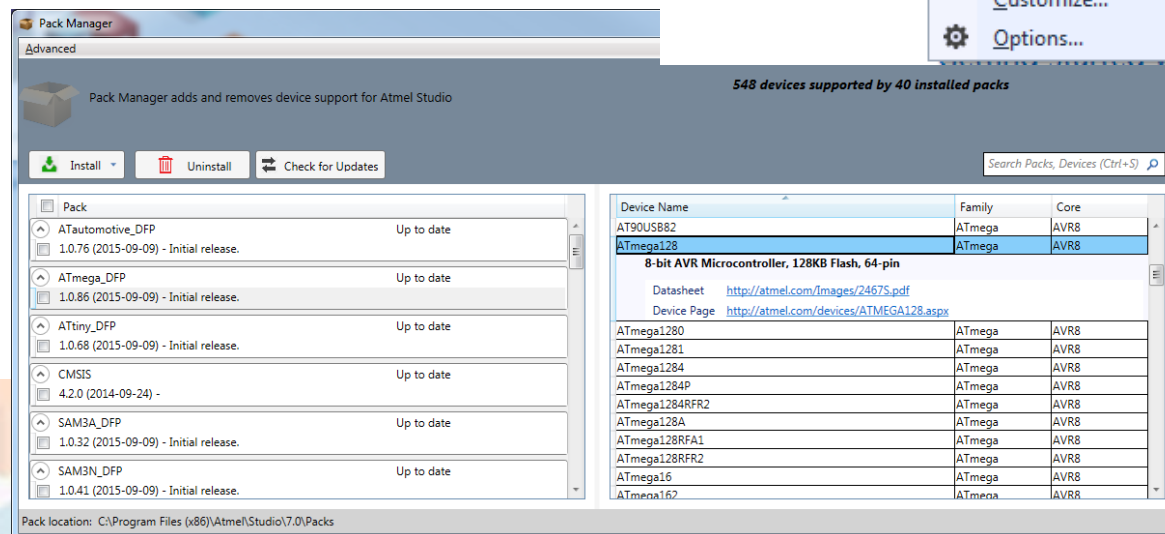
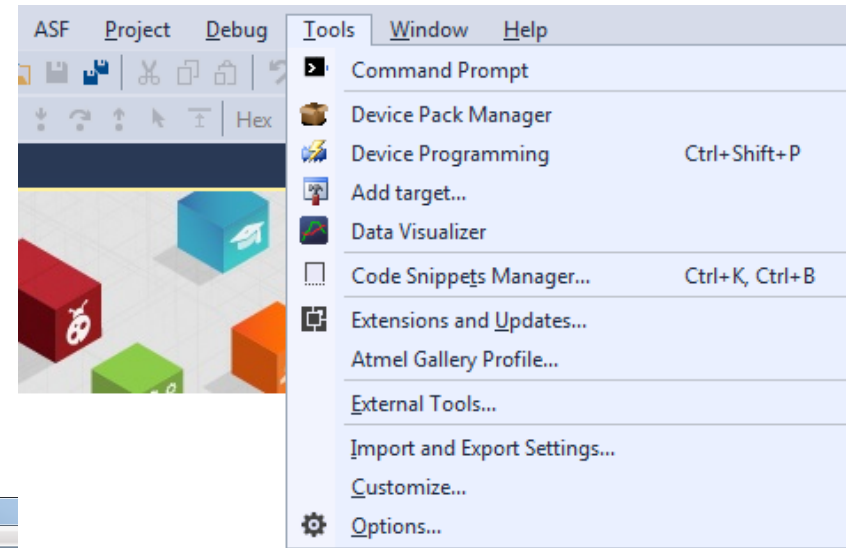
- When loading many projects part of a solution



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

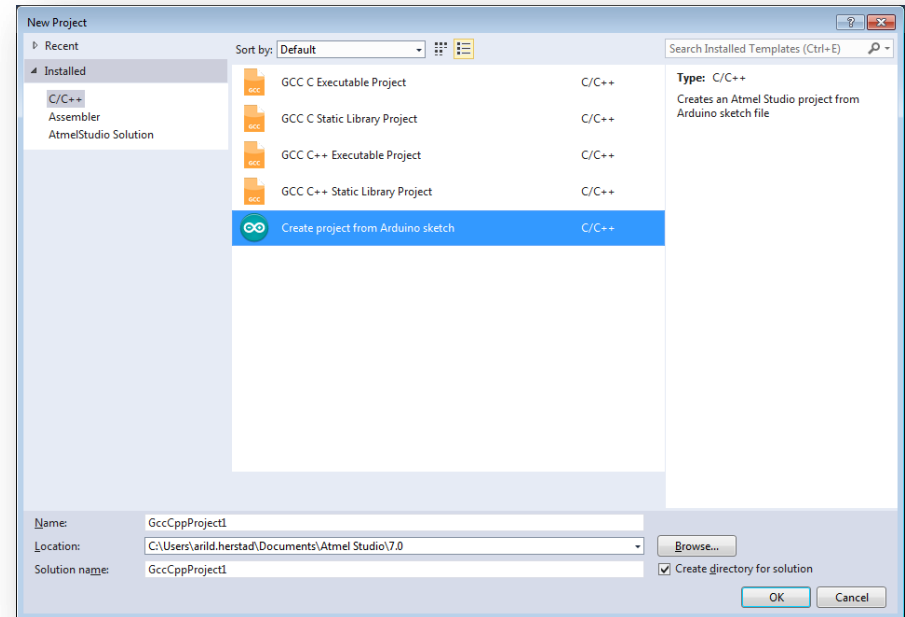


From Maker to Market

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





Enabling Unlimited Possibilities®



© 2015 Atmel Corporation.

Atmel®, Atmel logo and combinations thereof, Enabling Unlimited Possibilities®, and others are registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.

Disclaimer: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. EXCEPT AS SET FORTH IN THE ATMEL TERMS AND CONDITIONS OF SALES LOCATED ON THE ATMEL WEBSITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS AND PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and products descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.