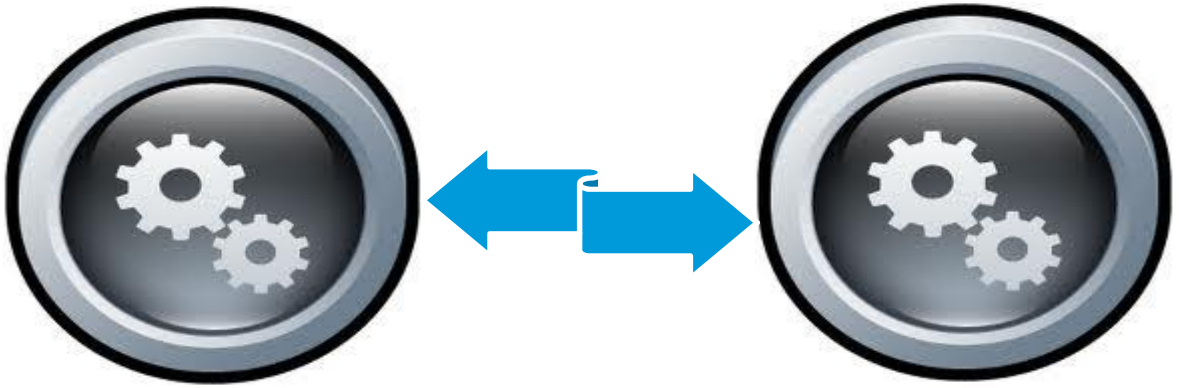
Product Management

who,where, date



nRF51 Technical Training

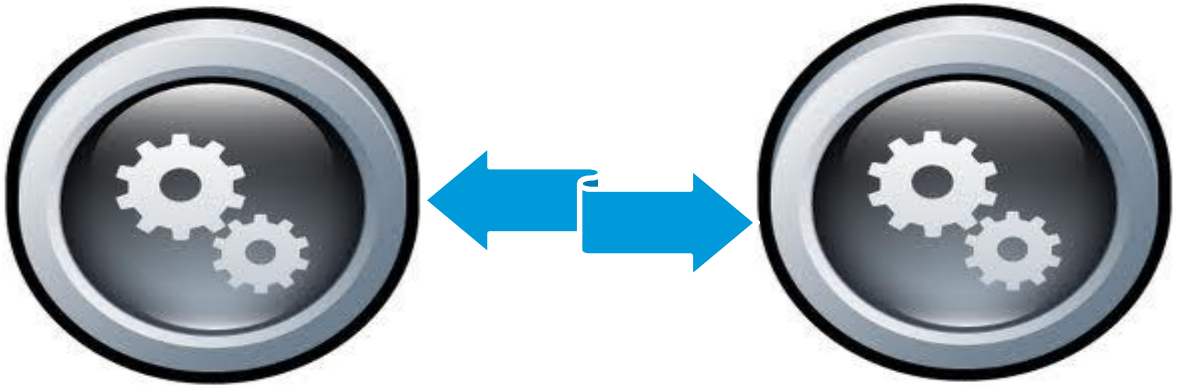
NORDIC SEMICONDUCTOR

Objectives

- HW architecture
- 2.4 GHz radio
- SoC framework
- Tool chain

NORDIC SEMICONDUCTOR

Product Management

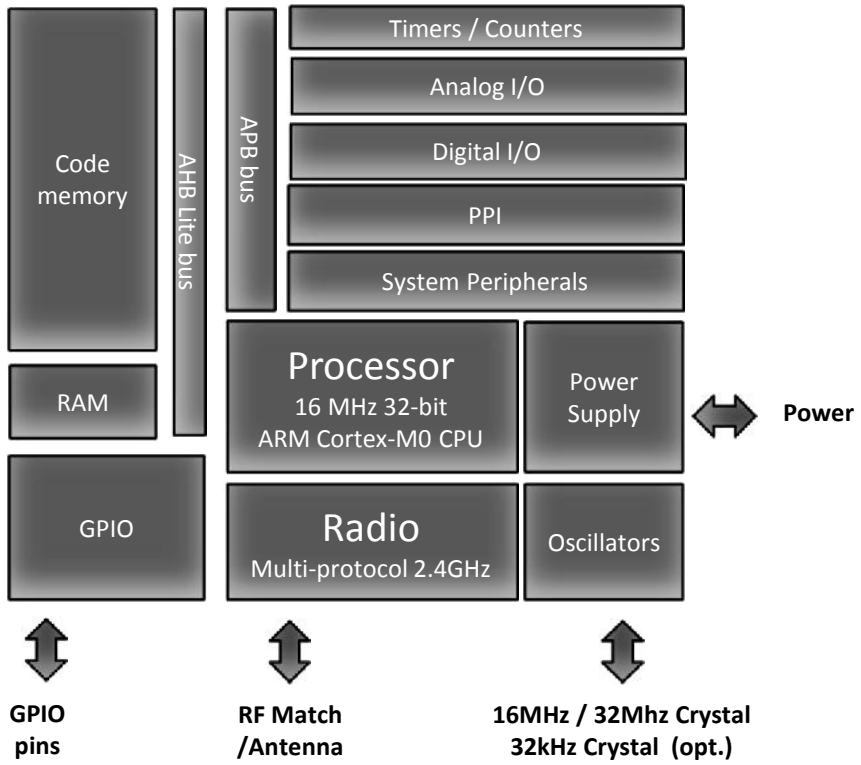


nRF51 HW architecture

NORDIC SEMICONDUCTOR

Main nRF51 HW features

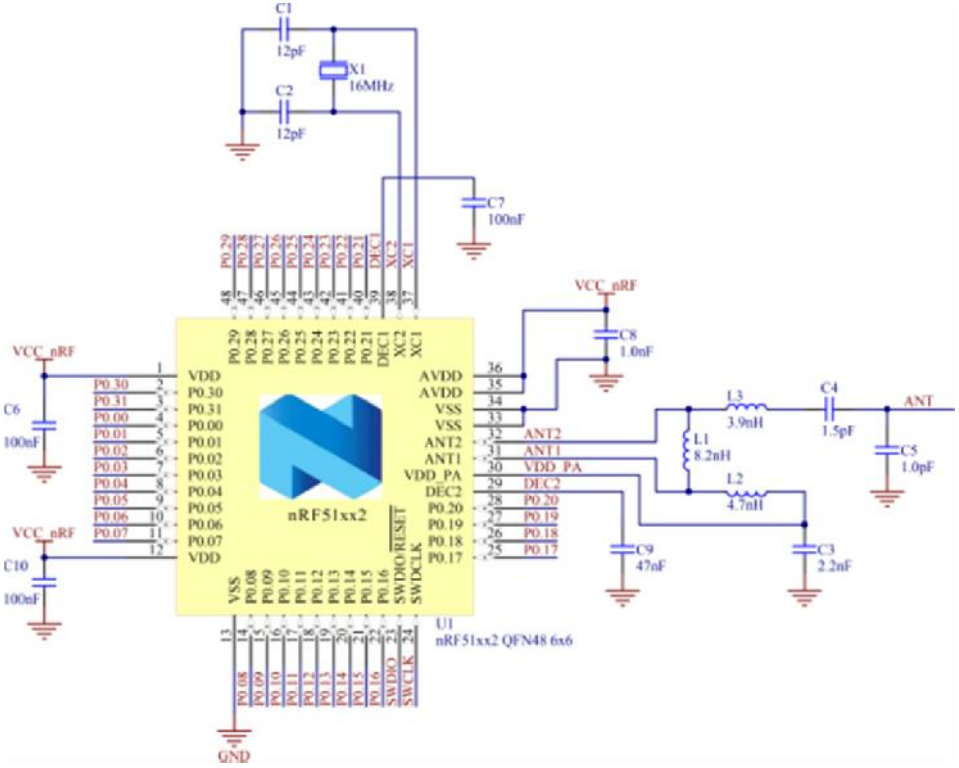
- Reference schematic
- 3 Power supply options
- Flexible I/O mapping
- PPI system
  - Peripheral interface
  - Peripheral task/event system
- Orthogonal power management
  - System ON/OFF
  - Individual peripheral control
- EasyDMA memory access
  - Peripheral specific DMA access



NORDIC SEMICONDUCTOR

Simple layout

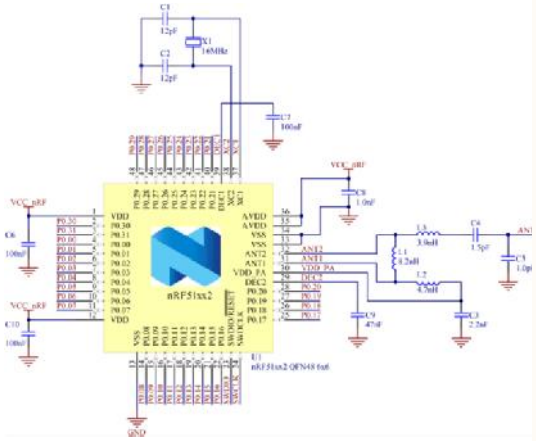
- Same as nRF24L:
  - Antenna match architcture
  - 16MHz crystal
  - 2 cap for internal decoupling
- Taken away:
  - IREF resistor
  - Resistor on 16 MHz crystal
- New:
  - 2 Pin program and debug IF
    - Not overlapping GPIO!
  - All system components in 1 area
    - Simplify re-use of layout
  - De-coupling at corners
    - Simplify layout



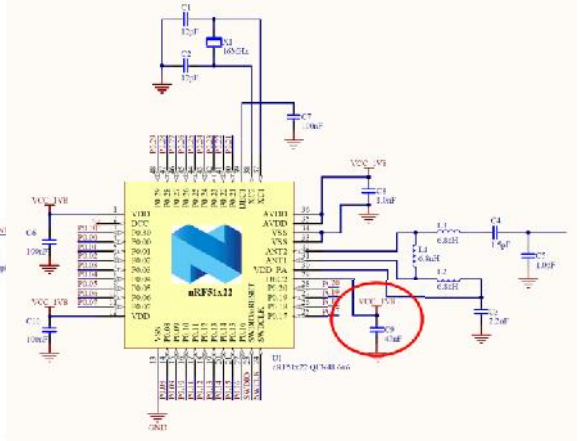
NORDIC SEMICONDUCTOR

Power supply options

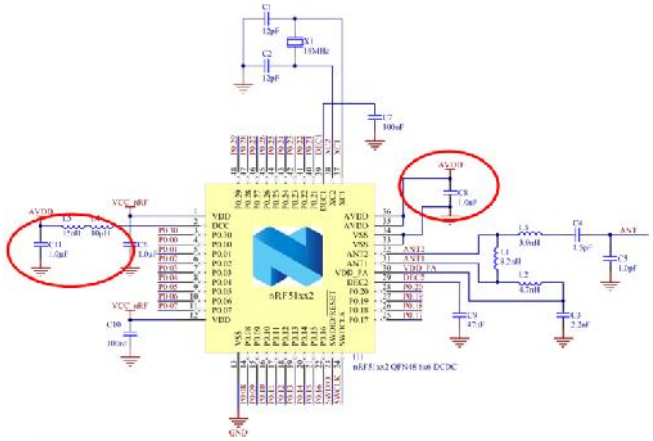
On-chip LDO	▪ **1.8 to 3.6V** supply range
1.8V direct supply (By passing on-chip LDO)	▪ **1.75 to 1.95V** supply range ▪ For applications with externally regulated 1.8V power supply
On-chip drop-down DC/DC	▪ **2.1 to 3.6V** supply range ▪ DC/DC can be switched on/off in FW. ▪ LDO mode can be used with DC/DC layout.



On-chip LDO



1.8V Direct

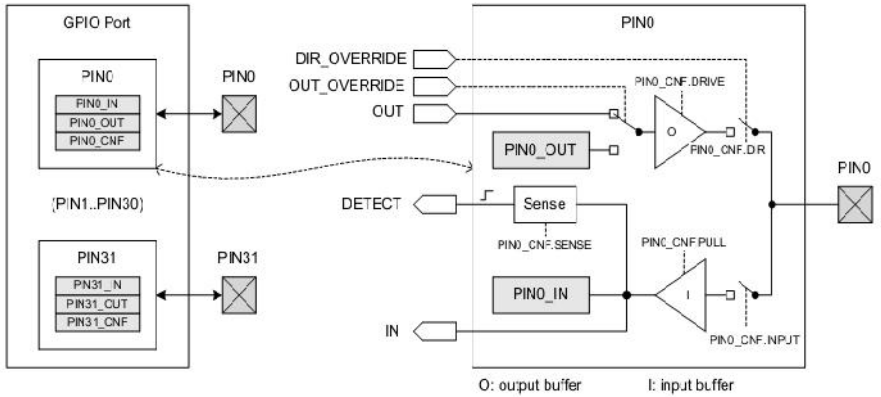


On-chip DCDC

NORDIC SEMICONDUCTOR

GPIO

- Up to 32 I/O in one port
- Wake from system off on any GPIO
- Pull up and pull down available in every pin.
- Output configurable in several modes with different strengths.
  - CMOS (normal/high drive)
  - Open source
  - Open drain
- Input buffer disconnect available to reduce power.



Flexible I/O mapping

- Serial I/F and special function I/O pins can be freely mapped to any GPIO
  - SPI, 2Wire, UART
  - Quad demodulator, GPIO TE
  - Enables nRF51 to adapt to external needs
  - Simpler/Better PCB routing
    - Better GND!
- Analog inputs & 32kHZ XO
  - Still locked to specific pins
- Put I/O's where you need them
- Registers in each peripheral allow you to choose which GPIO to be used for each signal.
  - Ex: SPI:
    - PSELMSCK: P0.04
    - PSELMISO: P0.05
    - PSELMOSI: P0.06

Peripheral interface architecture

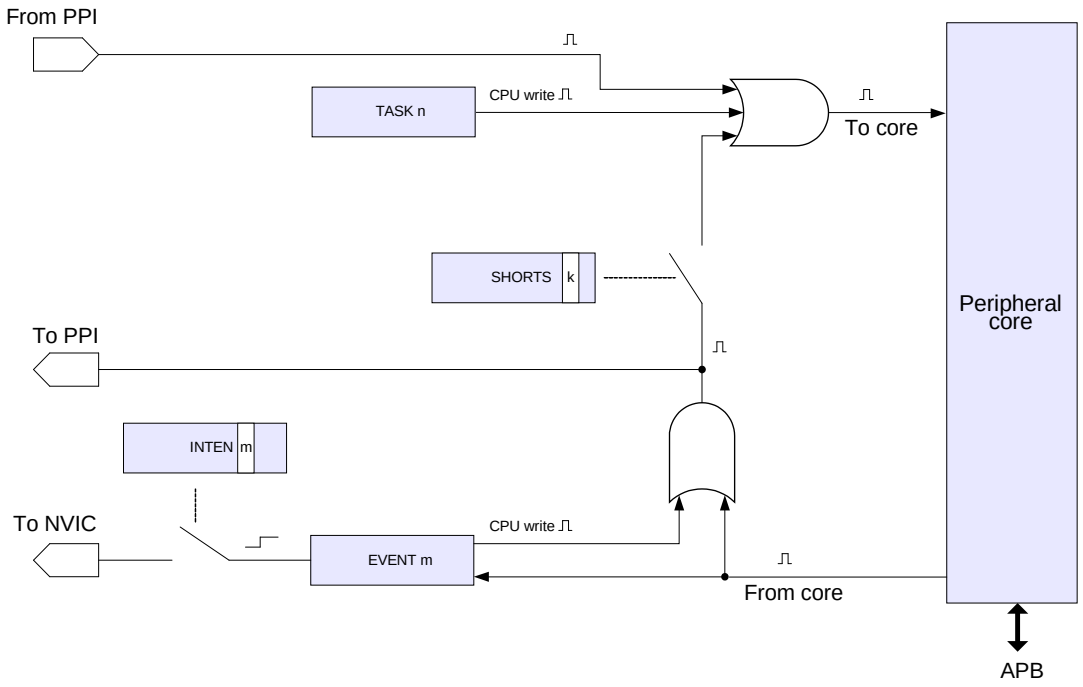
- Each peripheral can be accessed through the ARM Advanced Peripheral Bus (APB)

- In addition they also support:
  - Tasks: Registers used to trigger actions in a peripheral.
  - Events: Registers indicating an event has occurred.
  - Shortcuts: Registers enabling direct hardware connection between events and tasks inside a peripheral.

- Interrupt enable registers
  - Used for configuring which events will be routed to the core to produce an interrupt.
  - Standard interrupt system

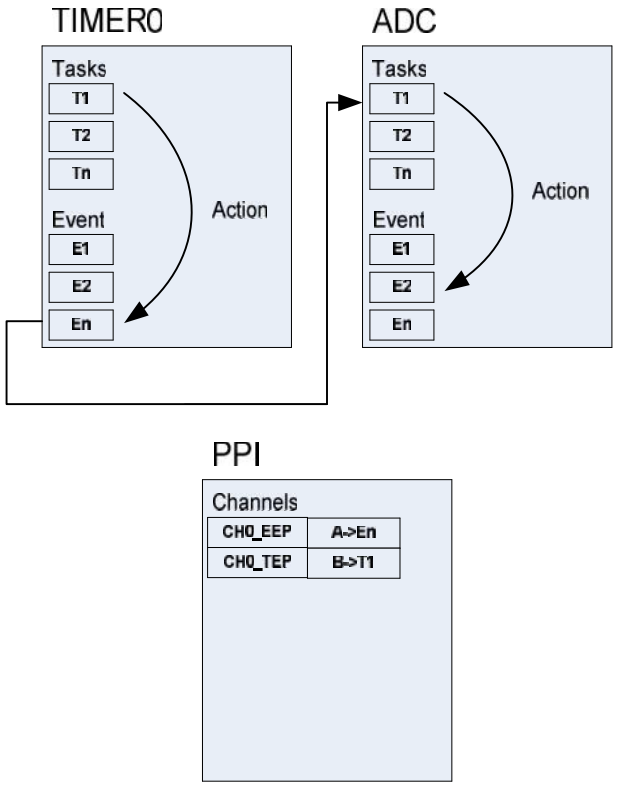
PPI peripheral

- CPU can write tasks to trigger actions.
- Shortcut
  - Direct connection between an event and a task within the same peripheral
- PPI is equivalent to Shortcuts, but fully configurable between an event and a task in two different peripherals



PPI Channel setup

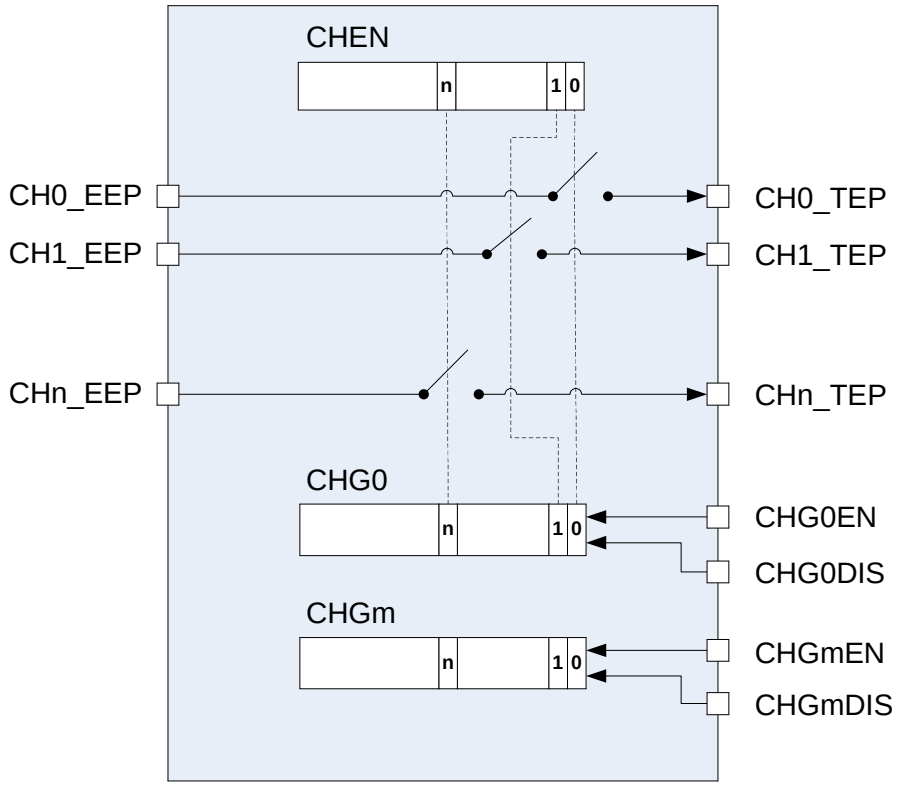
- PPI Channel
  - Task End-Point Register (TEP)
  - Event End-Point Register(EEP)
  - Channel enable.
- Usage
  - Connect A to B
  - NRF_PPI->CH0_EEP =
    (uint32_t) &NRF_TIMER0->EVENT_COMPARE0;
  - NRF_PPI->CH0_TEP =
    (uint32_t) &NRF_TIMER1->TASKS_START;



NORDIC SEMICONDUCTOR

PPI Functional description

- Enable or disable
  - Individual PPI channels (CHEN)
  - Groups of PPI channels (CHG)
- PPI configuration
  - n: number of channels
  - m: number of groups

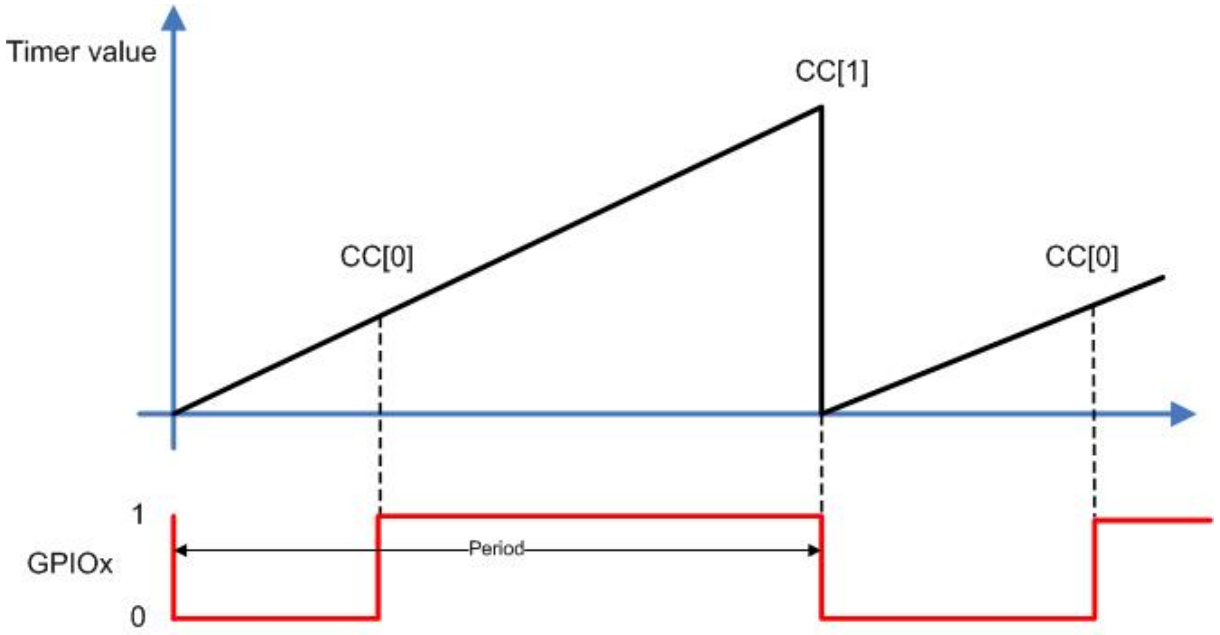


GPIO tasks and events

- GPIOTE - Module enabling peripherals to interact with GPIO using PPI
  - Can be configured to generate an event when a GPIO change logic level
    - Level or edge triggered
  - React to a tasks to produce change on a GPIO used for output:
    - Low to high, High to low, toggle
- nRF51822: 4 GPIOTE channels

- Benefits
  - Enable peripherals to respond directly to an input from a pin
    - Real time capture of data without CPU processing
    - Ex: GPIO generate timer capture or start ADC sample
  - Enable a timer to directly control outputs
    - Generate accurate (periodic) output without CPU processing.
    - Ex: a timer generate a PWM output directly

PWM - PPI example



Match	Event	PPI	Task	Comment
CC[0]	COMPARE[0]	0	GPIOTE set	
CC[1]	COMPARE[1]	1	GPIOTE clear	Shortcut: clear timer
CC[0]	COMPARE[0]	0	GPIOTE set	

Power management

- System OFF mode
  - Deepest power saving mode
  - Most functions and peripherals are powered down and non-responsive
  - Wake-up from pin and wake-up or reset only
  - Registers are not retained
  - Part or full RAM can be retained
- System ON mode
  - Fully operational mode, CPU and peripherals are responsive at all times
  - RAM and registers are always retained
  - CPU can be set into sleep via WFI and WFE instructions
  - CPU and peripherals controls its clock autonomously, depending on activity level, to optimize for energy use.

Power management

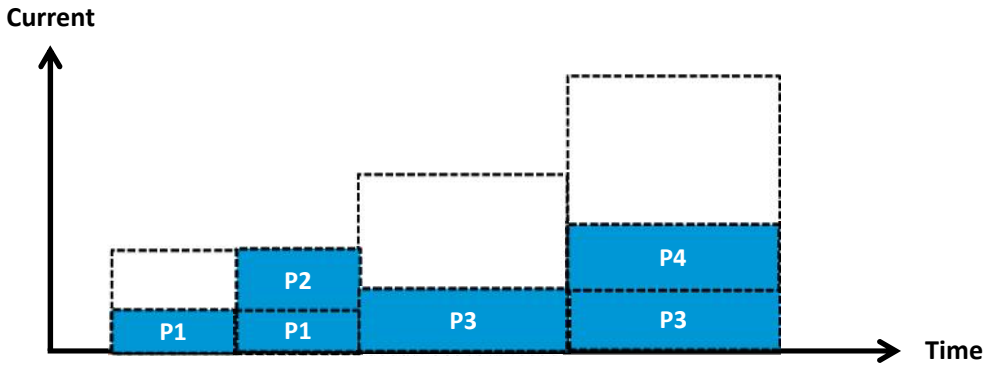
- Sub-power modes in System ON mode
  - Idle power can be traded for responsiveness
  - LOWPWR for minimum power
    - Lowest power clock source (oscillator) only started when a module need them
  - CONSLAT for constant latency wakeup (more power)
    - Clock sources kept active to minimize response time.
- Power supply supervisor.
  - Provides an early warning of impending power failure.
  - Can be configured at different levels.
  - Generates an EVENT when warning level reached.

New power management scheme

Minimizes current 'waste'

No predefined power modes

Only blocks doing something consume power



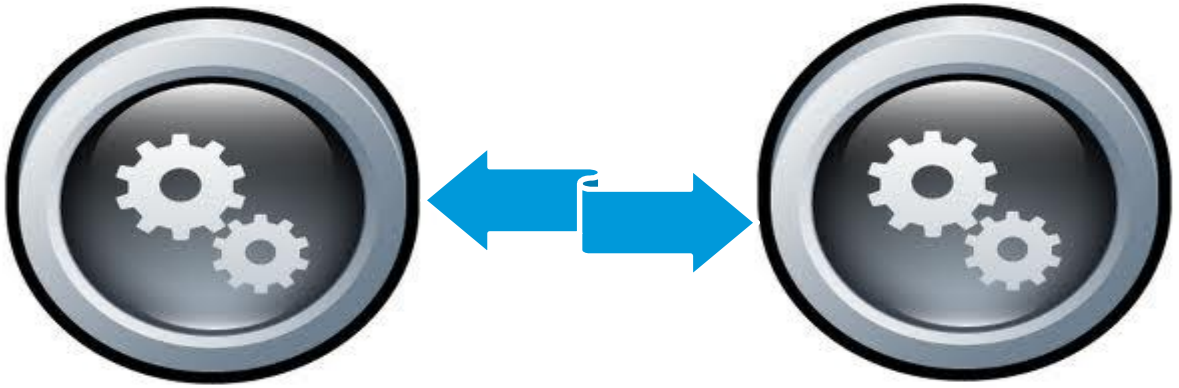
NORDIC SEMICONDUCTOR

EasyDMA

- A mechanism to enable selected peripherals to access the RAM directly.
  - More power efficient compared to register buffering
  - Faster compared to manually moving data using the CPU
  - No generic DMA controller to set up and maintain.

- Easy to use and robust
  - Only one parameter to configure, the RAM pointer.
  - One dedicated EasyDMA function per DMA enabled peripheral

Product Management



nRF51 2.4 GHz radio

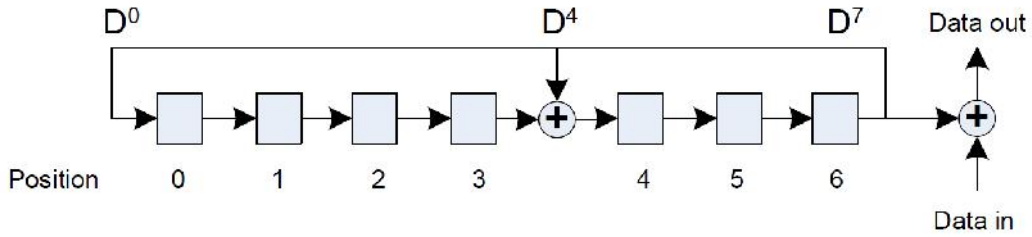
NORDIC SEMICONDUCTOR

Radio Interface

- The radio uses a simple DMA to access RAM (cannot access code space)
  - Transmit/received data stored directly to a RAM address
  - An address pointer of the data to be sent are given to the radio
- Register mapped configuration (SFR)
  - CPU has direct access to the radio registers.
- The radio is very simple, but relies on the PPI system and SW.
  - Makes it more flexible than the nRF24L !
- Fully tested ESB and Gazell libraries in SDK
  - Core functionality precompiled rather than source code for easier support
- Custom protocols can be developed by the customer

Data whitening

- The RADIO is able to do packet whitening and de-whitening.
  - When enabled, whitening and de-whitening will be handled by the RADIO automatically as packets are sent and received.
- The whitening word is generated using a linear feedback shift register, which then is XOR'ed with the data packet that is to be whitened, or de-whitened.
- Data whitening is done in order to randomize the data from highly redundant patterns and to minimize DC bias in the packet
- Most efficient if the data contains long parts of zeroes or ones

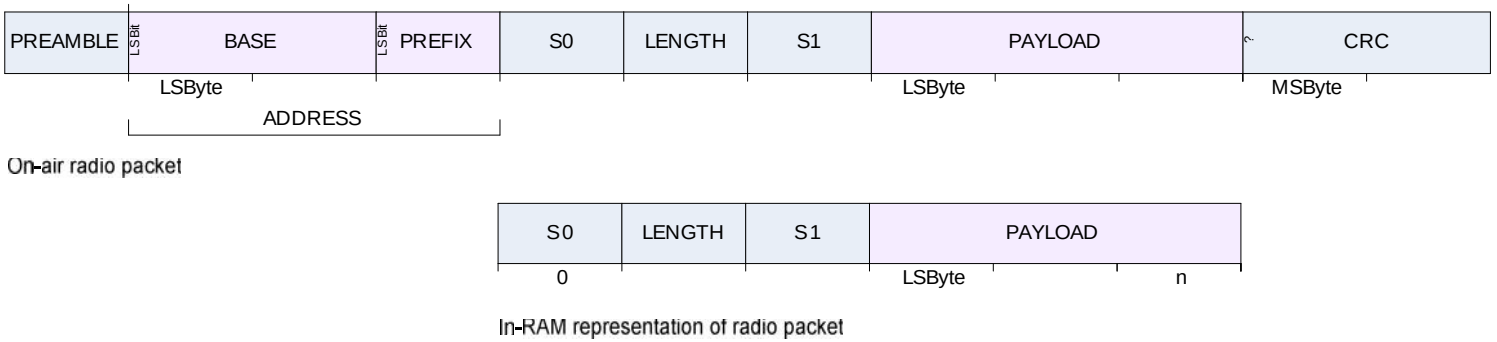


NORDIC SEMICONDUCTOR

RSSI

- Measurement of the received signal strength is started by using the RSSISTART task and the sample can be read from the RSSISAMPLE register.
- The sample period of the RSSI is 25 us.
  - The RSSI measurement will be the average received signal strength during this sample period.
- For the RSSI sample to be valid the radio has to be enabled in receive mode and the reception has to be started (RXEN task followed by RSSISTART task)

Radio packet

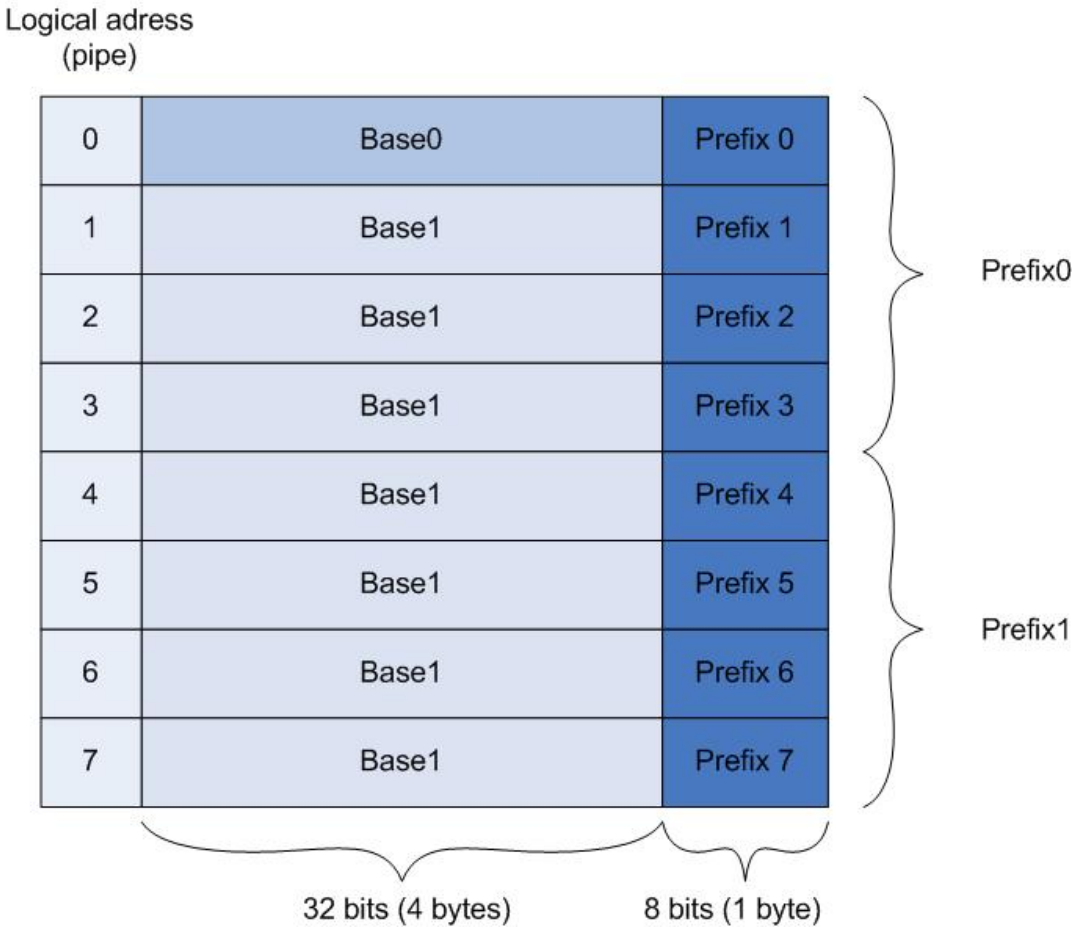


- Packet format:
  - **Preamble -** 0xAA or 0x55, as before
  - **Adress –** 4 byte Base and 1 byte prefix
  - **S0 -** Configurable content, 0 to 8 bits length
  - **Lenght** - Length of the payload
  - **S1 -** Configurable content, 0 to 8 bits length
  - **Payload -** 1 to 255 bytes (includes S0, Length and S1)
  - **CRC -** 8,16 or 24 bit
- S0, Lenght and S1 can be removed for backward compability
- For ESB: S0 is removed, Lenght is 6 bit and S1 is 3 bit to mimic the PID field. Content set in the coresponding registers updated in SW.

NORDIC SEMICONDUCTOR

Adressing

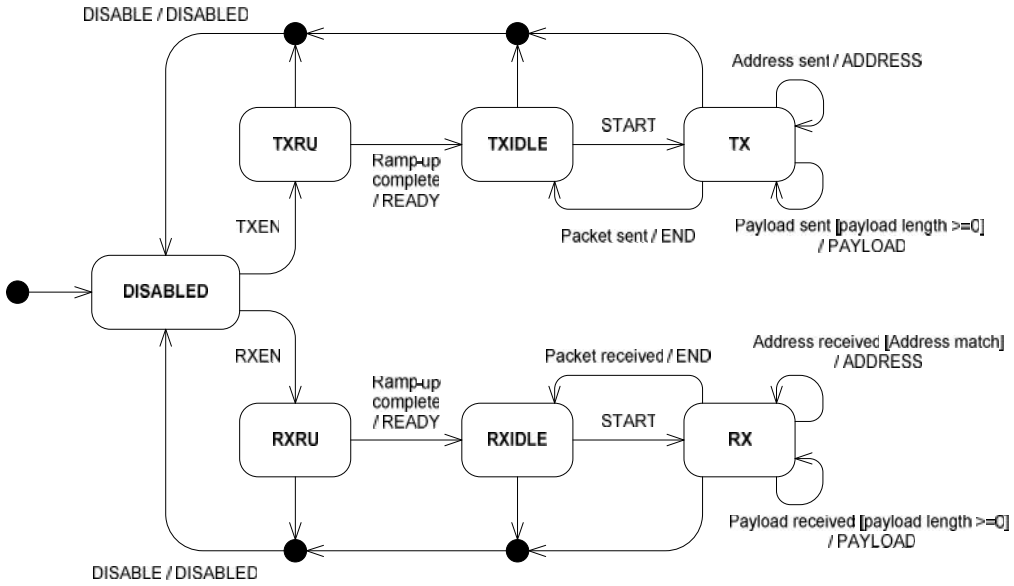
- On air address field split in Base and prefix fields.
- Base0, Base1, Prefix0 and Prefix1 are physical registers
- Can support 8 links
- On air address set once
  - Logical adressing used when the receiver is used by CPU



NORDIC SEMICONDUCTOR

Radio states

State	Description
DISABLED	No operations are going on inside the radio and the power consumption is at a minimum.
RXRU	The radio is ramping up and preparing for reception. 130us
RXIDLE	The radio is ready for reception to start.
RX	Reception has been started and the addresses enabled in the RXADDRESSES register are being monitored.
TXRU	The radio is ready for transmission to start. 130us
TXIDLE	The radio is ready for transmission to start.
TX	The radio is transmitting a packet



All functionality previously done by the internal state machine must now be handled by shortcuts/PPI/MCU!

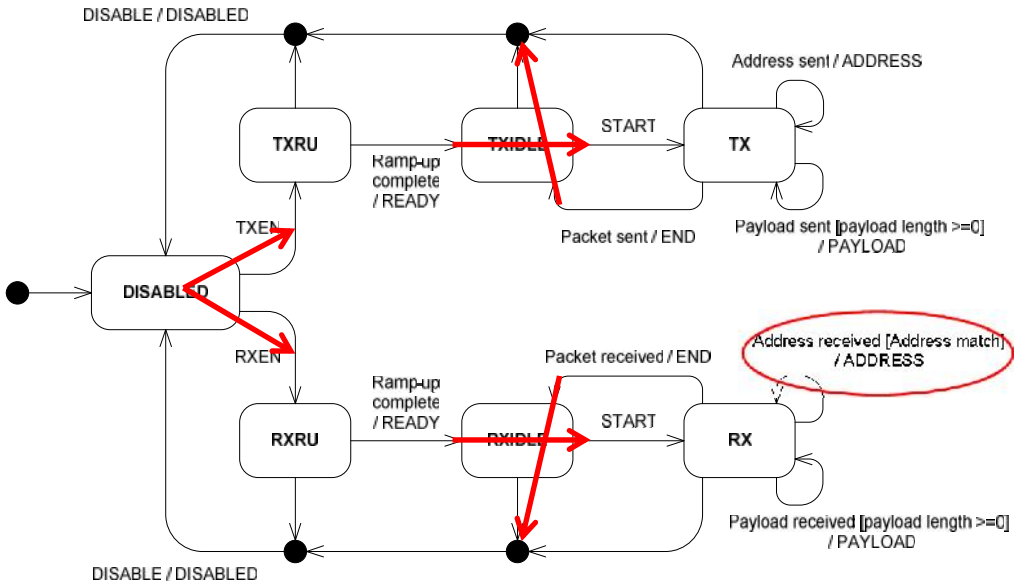
NORDIC SEMICONDUCTOR

Shortcuts

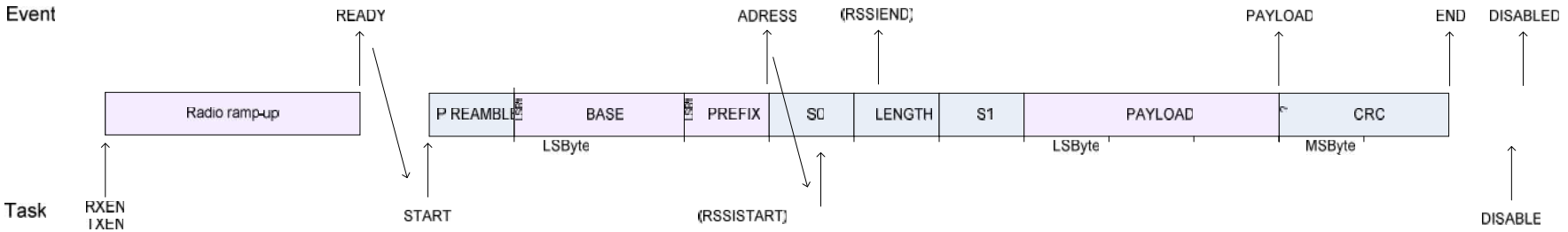
Used to speed up the radio handling.

Can also be combined with the PPI

- READY_START
  Starts TX or RX after ramp up (130us)
- END_DISABLE
  Disables the radio after TX or RX period ended
- DISABLED_TXEN
  Starts the TX after the disabled event has been issued
- DISABLED_RXEN
  Starts the RX after the disabled event has been issued
- ADDRESS_RSSISTART
  Starts the RSSI measurent after address match



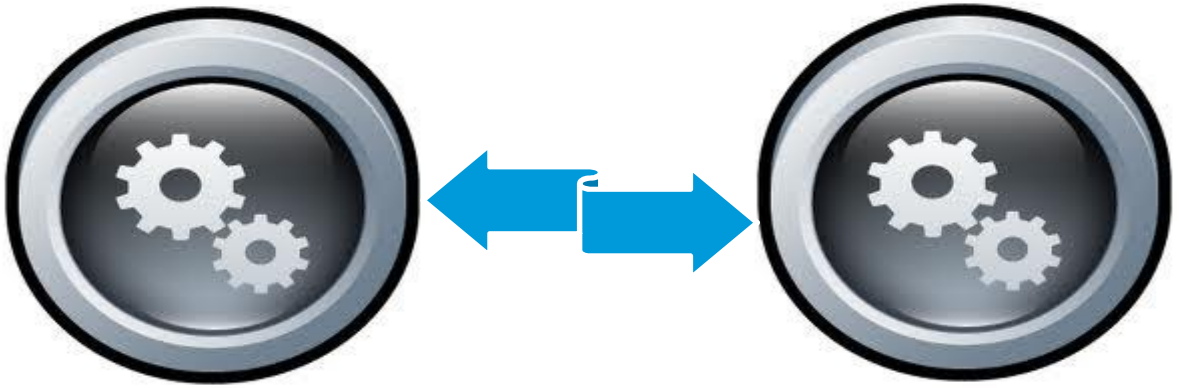
Radio events and tasks



- Tasks starts the “action”.
- The radio will issue events when a task or parts of it is finished
- Events valid and equal in TX and RX mode. An event can be ignored or linked to an INT
- The MCU can respond to events if needed
- Shortcuts can be set to start a task based on an event

NORDIC SEMICONDUCTOR

Product Management

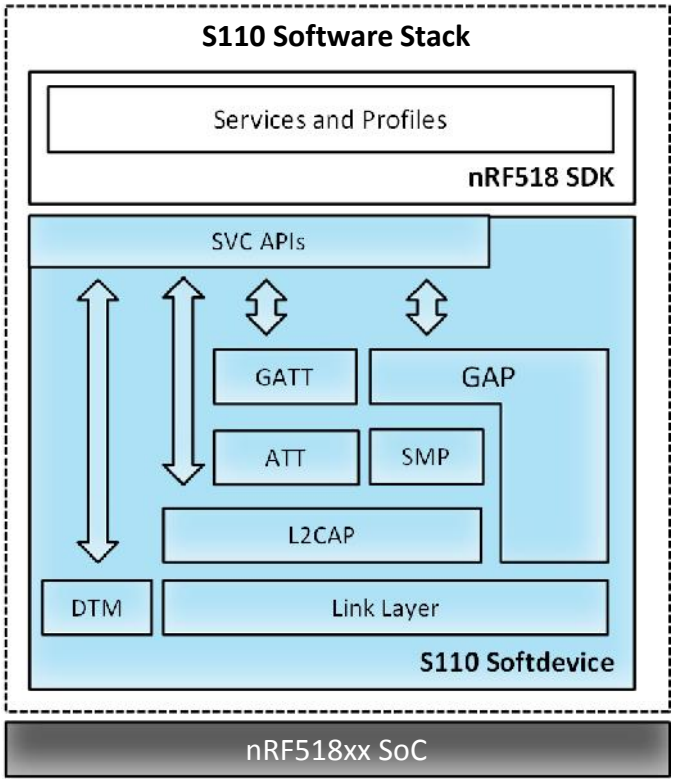


nRF51 Soft Device

NORDIC SEMICONDUCTOR

SoftDevices

- What is a SoftDevice
  - Stand alone pre compiled FW block
  - Programmed separately from application
  - No link time dependency on application
- What does it contain?
  - ANT or BLE protocol stacks
  - Support modules
  - Keeps stack and application separate
- Located in a reserved memory space
  - Ensures run time protection
- Softdevice + device profiles in SDK
  - Complete **Software Stack**

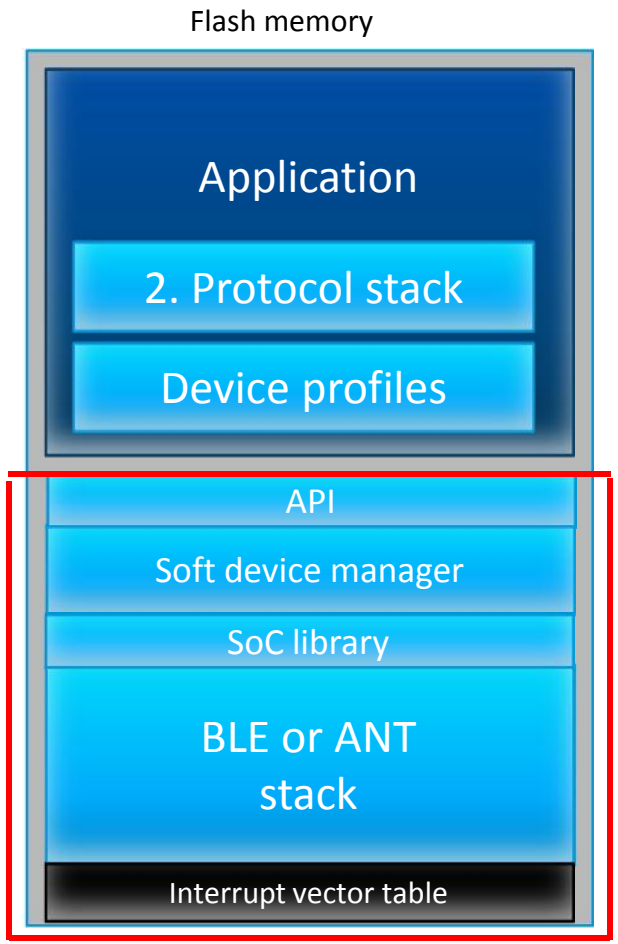


Example: S110 *Bluetooth* low energy Software stack:

NORDIC SEMICONDUCTOR

Memory map

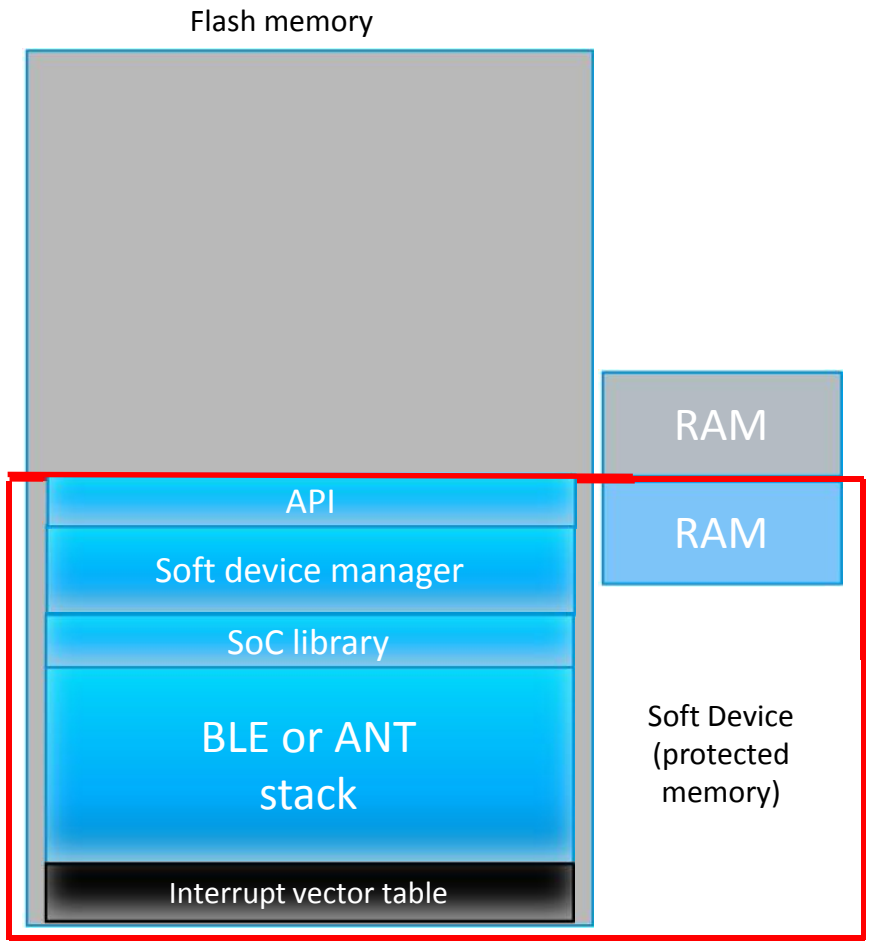
- nRF51 has one continuous, physical flash memory space
- Can be split in 2 regions
  - Region 0: Soft Device (if used)
  - Region 1: Application/ opt. 2nd protocol
- Application/2. protocol stack can never directly access reserved Soft Device flash memory
- Attempts to do so will give Hard Fault interrupt to the CPU.



NORDIC SEMICONDUCTOR

SoftDevice framework

- SoftDevice manager
  - Your FW can turn SoftDevice ON/OFF
- When ON:
  - SoftDevice mngr. prevents direct access to:
    - RAM used by stack
    - HW used by stack
      - (RADIO, RNG, TIMER, RAM space)
  - **SoC library** gives controlled access to these resources!
- When OFF:
  - Application has access to ALL HW on device
  - Except flash memory used by SoftDevice

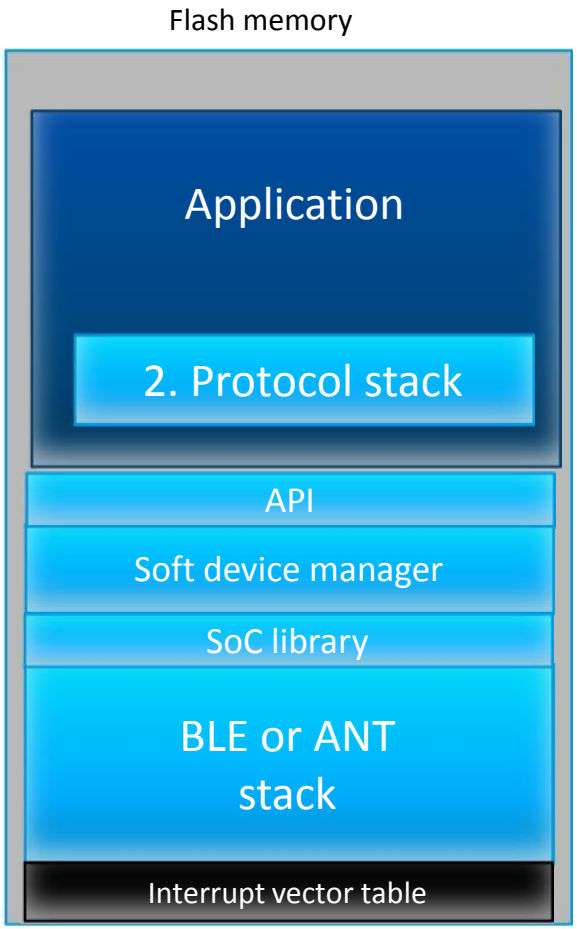


NORDIC SEMICONDUCTOR

Interfacing the SoftDevice

- How do you interface the SoftDevice when no link is set up with your application at compile time?

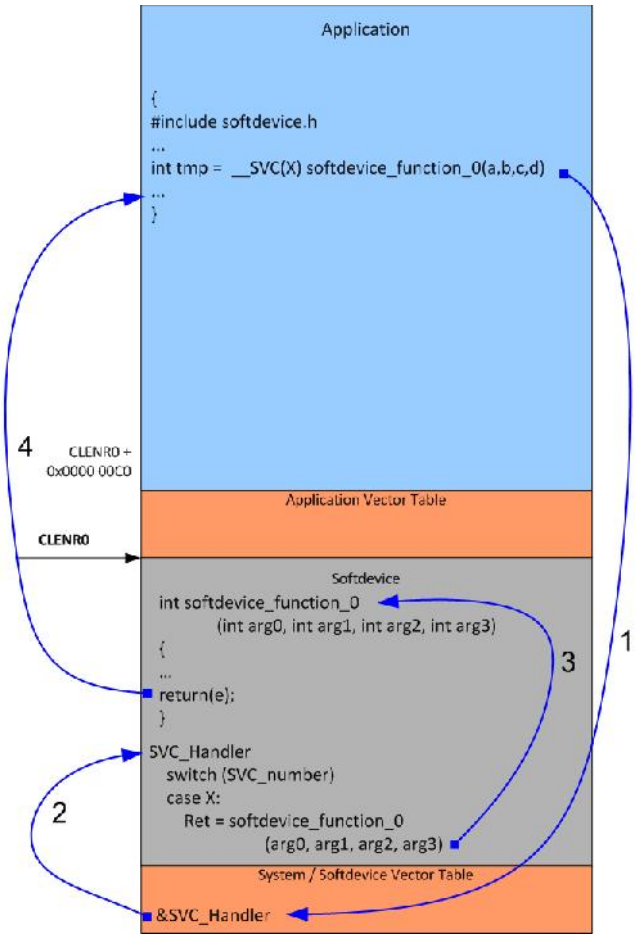
- Special feature in ARM Cortex-M
  - Supervisor calls
  - FW functions that trigger a HW interrupt
  - All API functions use supervisor calls
    - All interaction with SoftDevice interrupt driven
    - No need for RTOS or other framework
    - Threadsafe execution of App and Stack



NORDIC SEMICONDUCTOR

SoftDevice API Call

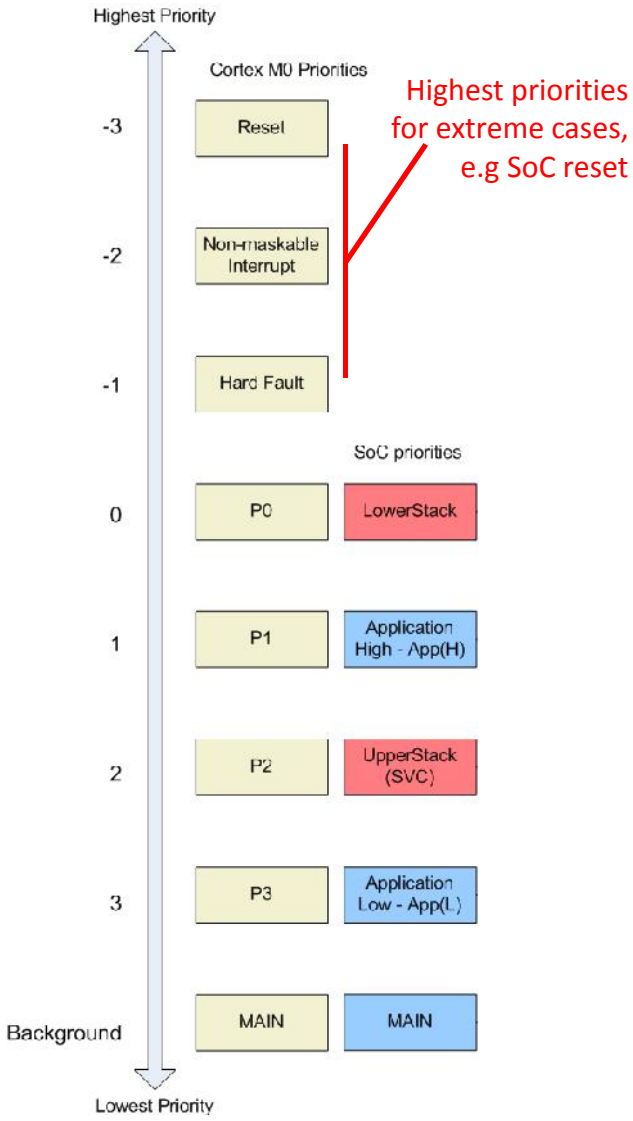
1. The application calls a function with a prototype imported from the SoftDevice API
   - A supervisor call exception is triggered.
2. The SoftDevice SVC handler address is vectored to by the HW interrupt system
3. The SVC handler determines which SVC number matches the call and calls the correct ISR
   - The function executes
4. The Function returns and program execution returns to the application



NORDIC SEMICONDUCTOR

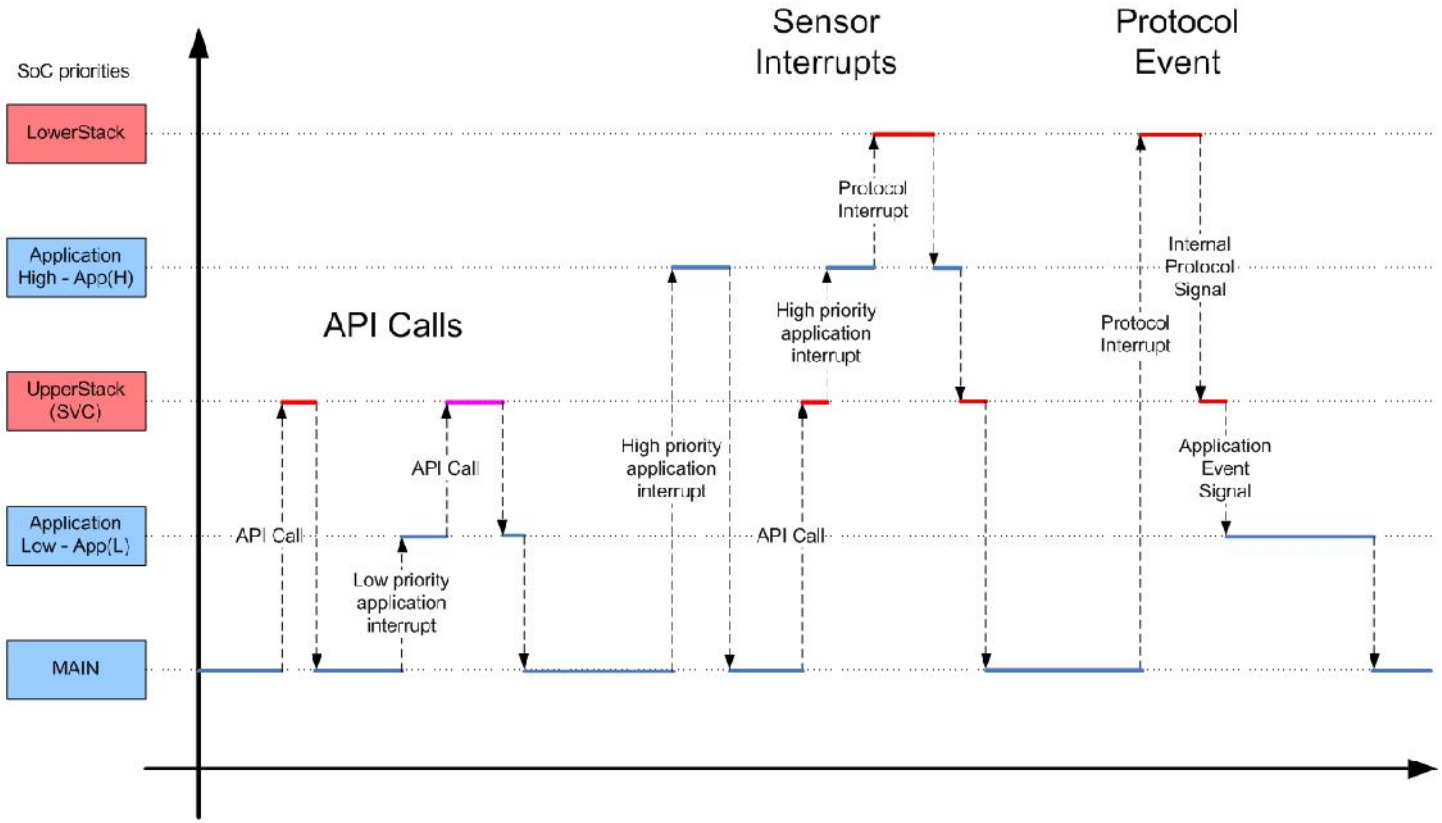
SoftDevice interrupts

- SoftDevice processing are interrupt driven
  - Access to CPU must be shared between Application and SoftDevice interrupts
- M0 has 4 level of interrupt priority and main context:
  - 2 levels of interrupt priority used by SoftDevice:
    - P0: lower stack = Link-controller
    - P2: Upper stack = Host
  - 2 levels + Main context left for Application
    - P1/P3: High/low application priority
    - Application can interrupt stack processing!
- Hard Fault interrup triggered if application try to access areas protected for SoftDevice



NORDIC SEMICONDUCTOR

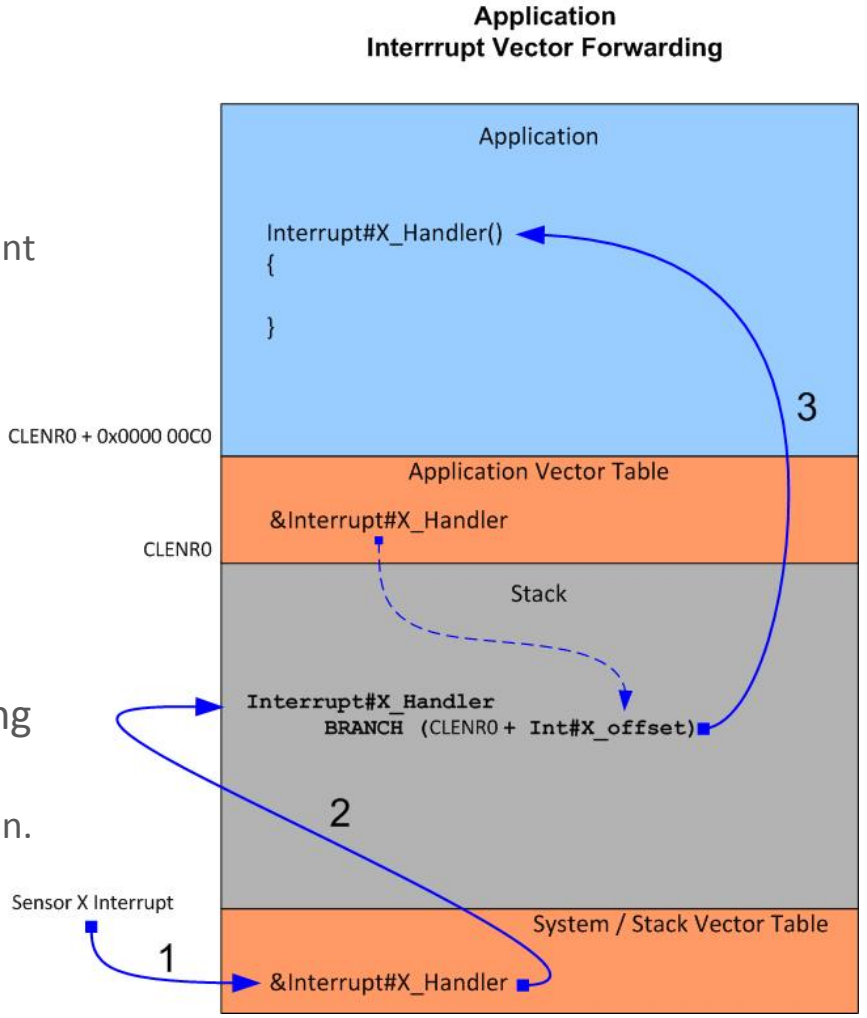
Example of nested interrupt handling



NORDIC SEMICONDUCTOR

Handling of ‘Application’ interrupts

- SoftDevice controls the actual M0 vector table
  - Application defines its own vector table!
  - i.e. Application same as with no SoftDevice present
  - Soft device handles vector forwarding
- SoftDevice enabled:
  - Vectors not used by the SoftDevice forwarded
- SoftDevice disabled
  - All vectors forwarded
- Due to this vector forwarding interrupt handling will be delayed by a few clock cycles
  - How many is given in each soft device specification.



Software Architecture Benefits

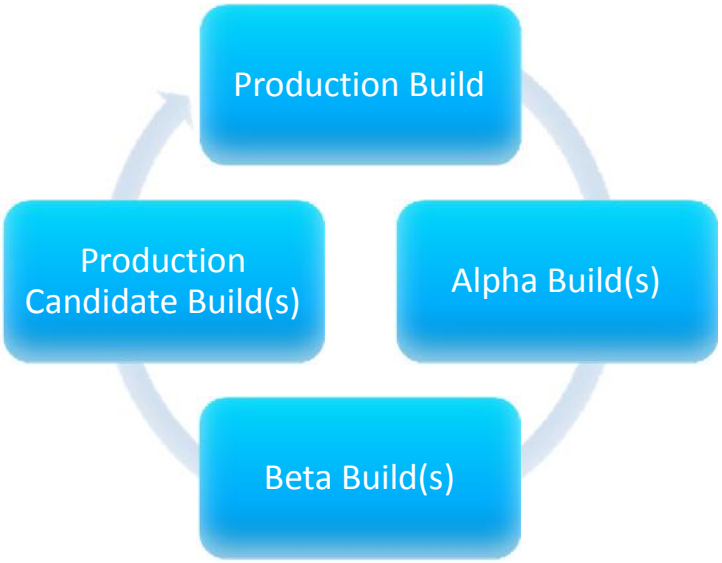
Flexibility	▪ Softdevice can be Enabled and Disabled at run-time to give application full access to HW resources including Radio and RAM ▪ No RTOS dependencies ▪ The application and Stack can be updated (programmed) separately
Ease of Development	▪ Simple application programmer model with no link time dependencies ▪ No proprietary application development model or RTOS ▪ Minor SoftDevice / stack updates do not require application to be re-compiled
Code safety	▪ Stack is not re-linked constantly during application development ▪ QA and qualification on binary going into end-user product ▪ Stack is run-time protected ▪ Reduced risk of application bugs affecting the Stack

Soft Device distribution

- nRF51 BLE Soft Devices will be programed by the customer
- Advantages:
  - Quicker releases of updates from Nordic
    - Bug fixes / new features
  - Nordic can continue releasing new Soft device version without forcing PCN.
  - Softdevices can be qualified and re-listed
- nRF51 series featuring ANT will have preloaded ANT SoftDevice

NORDIC SEMICONDUCTOR

SoftDevice Release Process

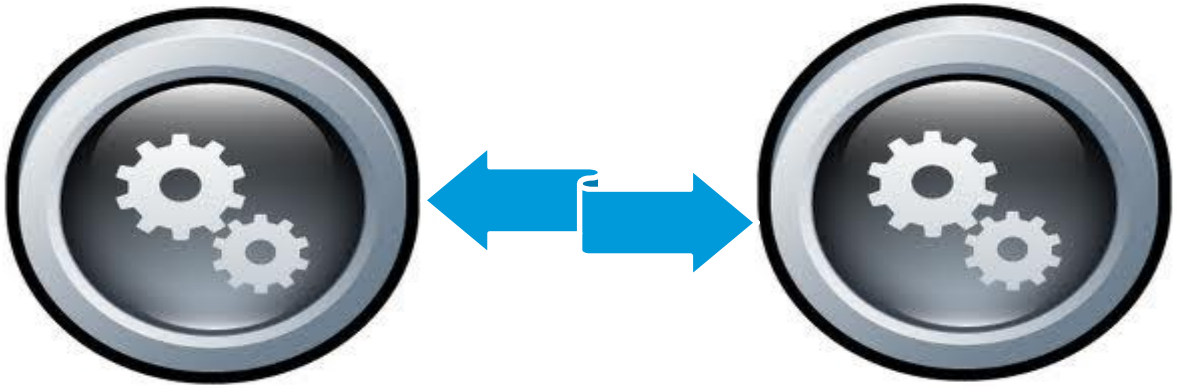


Alpha build	▪ Implemented feature verification ▪ Suitable for preview purposes
Beta Build	▪ Functional testing finalized ▪ HW integration testing on Silicon ▪ Suitable for development purposes
Production Candidate Build	▪ Feature-complete Engineering Build ▪ Mass production QA not finalized ▪ Suitable for design qualification
Production Build	▪ Mass production QA finalized ▪ Suitable for mass production

NORDIC SEMICONDUCTOR

Product Management

Part 1

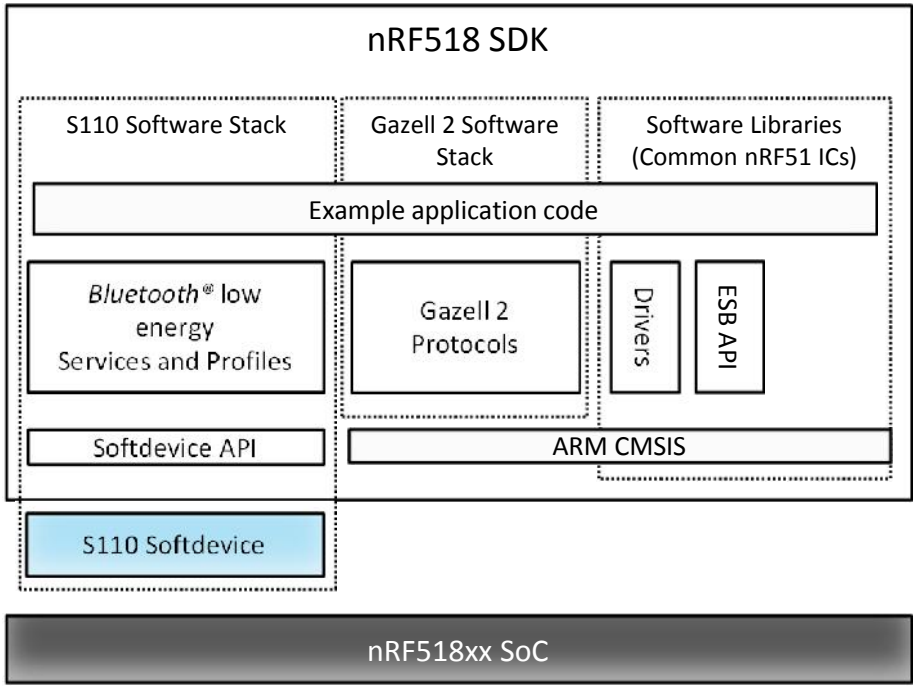


nRF51 Development

NORDIC SEMICONDUCTOR

nRF518 SDK

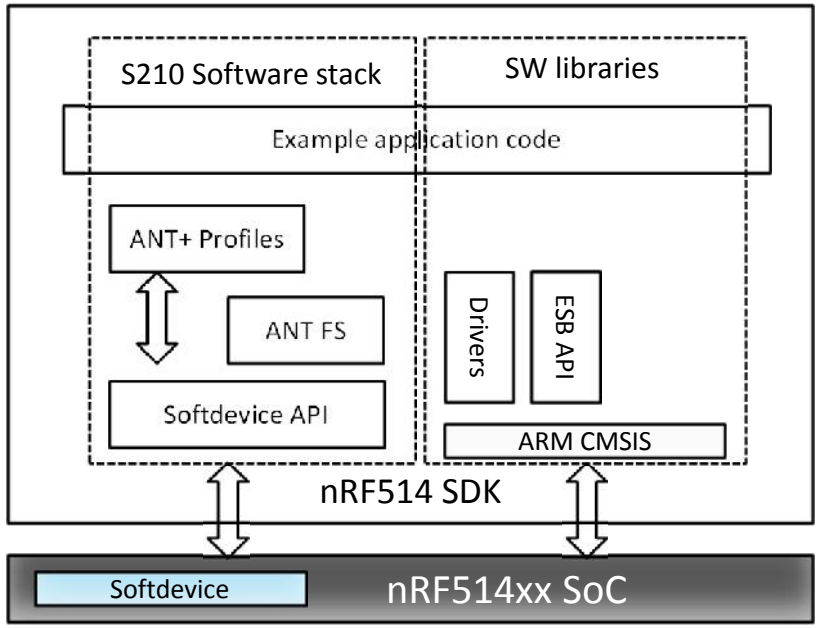
- Common SDK for nRF518xx ICs
- Software stacks
  - S110 *Bluetooth*® low energy
  - Gazell™ 2 Proprietary 2.4GHz RF
- Software Libraries
  - Common for all nRF51 ICs
  - ARM CMSIS (-CORE, -SVD, -DSP)
  - Enhanced ShockBurst™ API
  - Drivers for device peripherals (SPI, 2 wire, UART, timers etc.)



NORDIC SEMICONDUCTOR

nRF514 SDK

- SDK for application development on ANT™ nRF514xx ICs
- ANT-FS and ANT+ Device profiles
- Software Libraries
  - Common for all nRF51 ICs
  - ARM CMSIS (-CORE, -SVD, -DSP)
  - Enhanced ShockBurst™ API
  - Drivers for device peripherals (SPI, 2 wire, UART, timers etc.)



NORDIC SEMICONDUCTOR

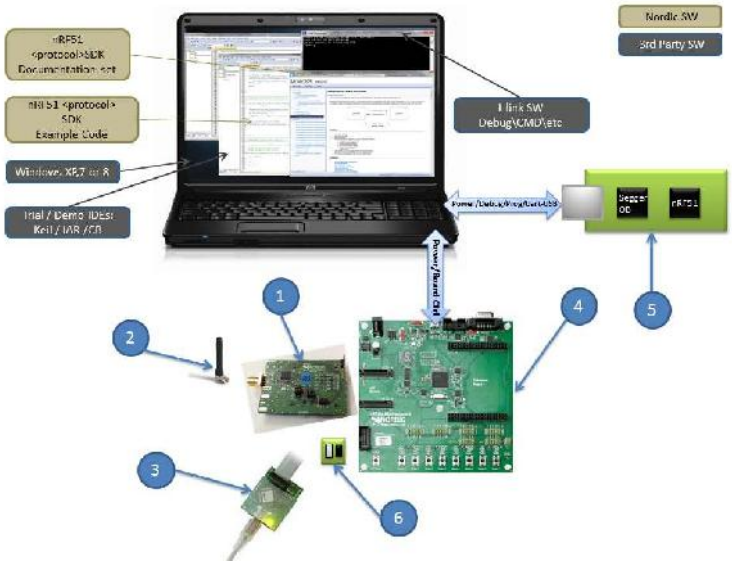
Tool chain support

- nRF51x SDK will come in 3 versions compatible with different tool chains:
  - Keil ARM-MDK
  - IAR
  - GCC
- Each IDE will be fully supported for compile, debug, simulation and programming
- Keil ARM-MDK available now
- IAR and GCC available Q4.

Single Wire Debug (SWD)

- Standard ARM component, part of ARM CoreSight technology.
- 2-pin ARM Serial Wire debug interface:
  - no GPIO's used
  - one pin shared with nRESET.
  - 4 breakpoints
  - 2 watchpoints
  - MicroTrace buffer for instruction tracing
- Large ecosystem of third party debugger tools and software available
  - Keil uLink support
  - SEGGER J-Link support (SEGGER J-Link will be integrated in our kits).

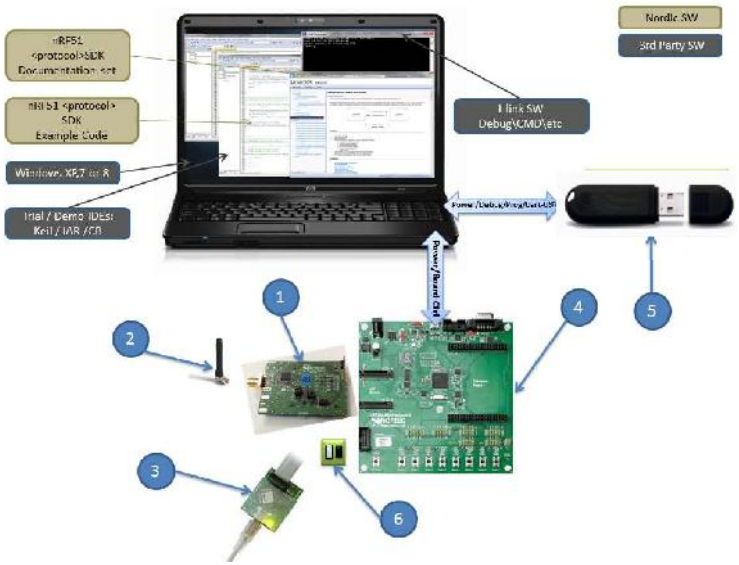
nRF51822-DK
nRF51822 Development Kit



- Evaluation, prototyping and development
  - Bluetooth® low energy and 2.4GHz RF
  - Hardware and Software
- 1,4: Works with nRFgo Starter Kit (nRF6700)
  - Pin-header Access to all GPIO's
- 3: Segger hardware debug solution
  - Wide range of ARM debug, programming and IDE support
- 5: USB module with on board debug HW
  - convenient software development
- Minimum requirements
  - Windows XP, Vista or 7
  - Two USB ports
  - nRFgo Starter Kit

NORDIC SEMICONDUCTOR

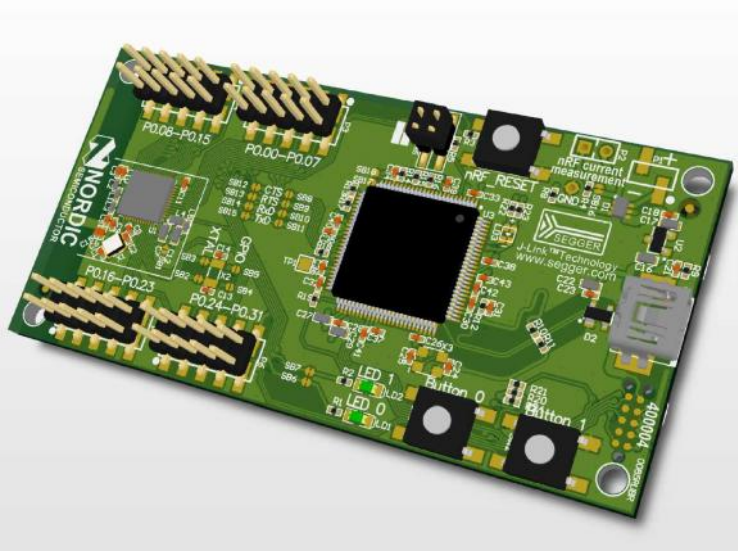
nRF51422-DK
nRF51422 Development Kit



- Evaluation, prototyping and development
  - ANT, ANT+ and 2.4GHz RF
  - Hardware and Software
- 1&4: Works with nRFgo Starter Kit (nRF6700)
  - Pin-header Access to all GPIO's
- 3: Segger hardware debug solution
  - Wide range of ARM debug, programming and IDE support
- 5: nRF24AP2-USB stick
  - Connection to ANTware and PC applications
- Minimum requirements
  - Windows XP, Vista or 7
  - Two USB ports
  - nRFgo Starter Kit

NORDIC SEMICONDUCTOR

nRF51x22-EK
nRF51x22 Evaluation Kit



- Evaluation & early prototyping
  - nRF51822-EK: Bluetooth® low energy and 2.4GHz RF
  - nRF52422-EK: ANT™
  - Hardware and Software
- 1 Stand alone evaluation board with on board debug solution.
- 1 USB module with on board debug HW
  - convenient software development
- LIMITED SW support:
  - Simple examples enabling evaluation & development
  - NOT as extensive as nRFgo and nRF51x SDK!!
- Minimum requirements
  - Windows XP, Vista or 7
  - Two USB ports

NORDIC SEMICONDUCTOR

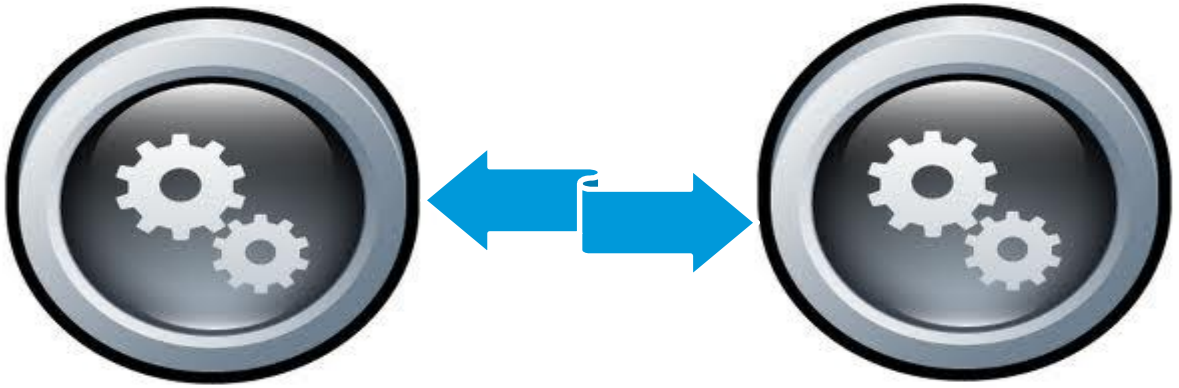
Further training

- In depth works shops will be held when nRF51 development kits are available
- nRF51 works shop
  - 2 days
  - Introduction to nRF51 SDK
  - Hands on work on nRF51822 DK and nRF518 SDK
  - Gazell and dual protocol
- Bluetooth low energy workshop
  - 2 days
  - Bluetooth low energy theory
  - Working with nRF51 SoftDevice
  - GATT, GAP and security
  - Server and clients
  - Hands on work with nRF518 SDK

NORDIC SEMICONDUCTOR

Product Management

Part 1



nRF51 Technical Training

NORDIC SEMICONDUCTOR