Running NuttX on Spresense

Introduction and Hands-on

Takayoshi.Koizumi@sony.com
Senior Manager
IoT Solution Business Divition
Sony Semiconductor Solutions Co.

Preparation for Hands-on

- Boards
  - Spresense Main Board x 1
  - Spresense Extension Board x 1
  - WiFi Add-on Board x 1
- Other things prepared by yourself
  - micro USB cable x 2
  - microSD Card x 1
  - Earphone x 1
  - Linux PC


MainBoard


ExtensionBoard


WiFi AddOn

- Software on PC for hands-on
  - NuttX normal build environment
  - mkspk tool : https://github.com/sonydevworld/cxd56.git
- Bootloader FW
  - https://developer.sony.com/file/download/download-spresense-firmware-v1-3-000

- Source tree for hands-on
  - NuttX :
    - git clone https://bitbucket.org/nuttx/nuttx.git
  - apps :
    - git clone https://bitbucket.org/nuttx/apps.git
  - Spresense SDK :
    - git clone –recursive https://github.com/sonydevworld/spresense.git

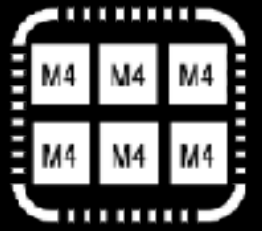
Today's Agenda

- Introductions
  - Spresense and CXD5602
  - Documentations
  - Status of Upstreaming to Original NuttX
- Hands-on with NuttX
  - nsh
    - With explanation of bootup mechanism of CXD5602
  - usbnsh
  - rndis
  - WiFi with gs2200m
  - OpenOCD ICE debugging
    - Just Demo because board needs pin header to connect JTAG ICE.
- Demos with Spresense SDK
  - Audio Player
  - GNSS
  - ASMP : Multicore framework
- Future plan for upstreaming

SPRESENSE™

Provided from Jly/2018

CPU	ARM® Cortex®-M4F x 6
Clock	Up to 156MHz
SRAM	1.5MB
Flash Memory	8MB
Digital I/O	GPIO, SPI, I2C, UART, PWM
Analog Inputs	6ch (3.3V range)
Audio I/O	8ch Digital MICs or 4ch Analog MICs, Stereo Speaker
GNSS	GPS, GLONASS, BeiDou, Galileo
Others	Camera IF, SD CARD, I2S



Low Power Multi Processor

- 28nm FD-SOI*3 technology
- 0.7V core voltage
- ASMP framework*4 for the multi processor

*3 Fully Depleted Silicon-On-Insulator to enable ultra-low-power features
*4 Software Framework to make communication between processors

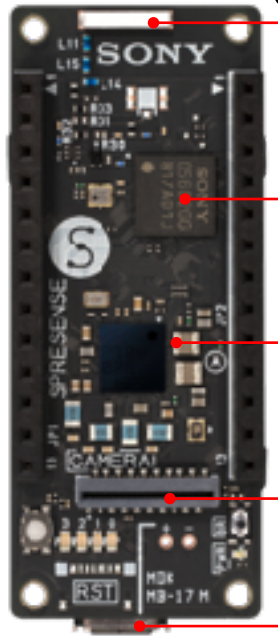
Audio Products for Music Lovers Provide New User Experience

- 192kHz/24bit High-Resolution audio
- 4 analog or 8 digital microphone inputs
- Class-D full digital amplifier

Positioning Features

- Ultra low power consumption
- GPS, GLONASS, QZSS
  Multiple GNSS systems supported

SPRESENSE™ BOARD CONFIGURATION



GPS chip Antenna

CXD5602GG
(Main chip)

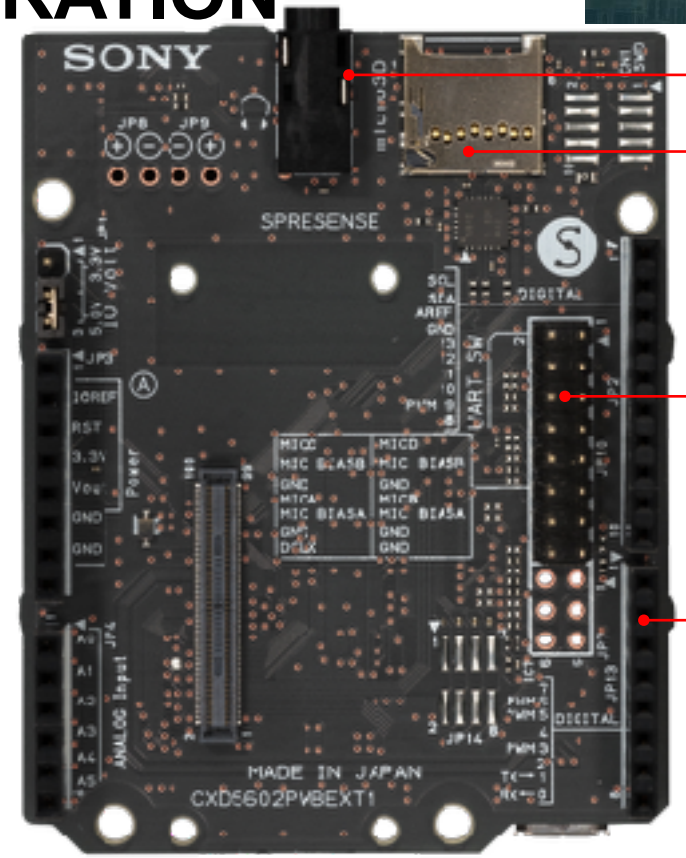
CXD5247GF
(PMIC/Audio)

Camera I/F

USB Debug Interface

SPRESENSE main board

Size	50.0 mm x 20.0 mm
GNSS	GPS, GLONASS
IO(1.8V)	GPIO, UART, I2C, I2S, SPI
Others	4 Application LEDs



Headphone

SD Card Slot

Pin Header
for Microphones

Arduino
Compatible
Connectors

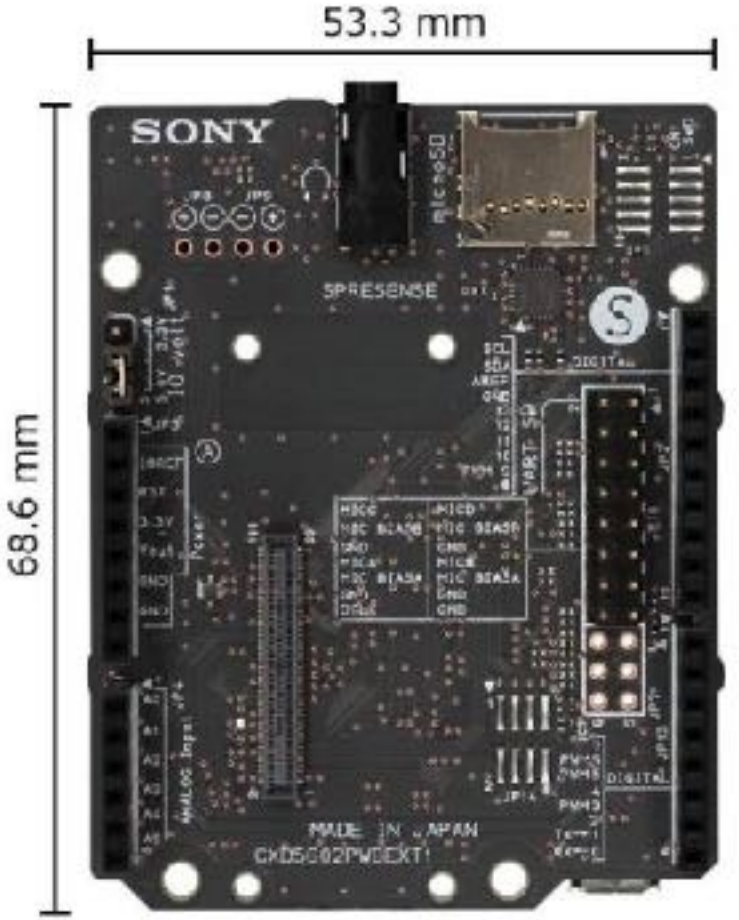
SPRESENSE extension board

Size	68.58 mm x 53.34 mm
Audio	Pin Header for 4 analog or 8 digital mics Headphone Jack
IO(3.3/5V)	Arduino compatible digital pins 5V range analog inputs

SPRESENSE board lineup

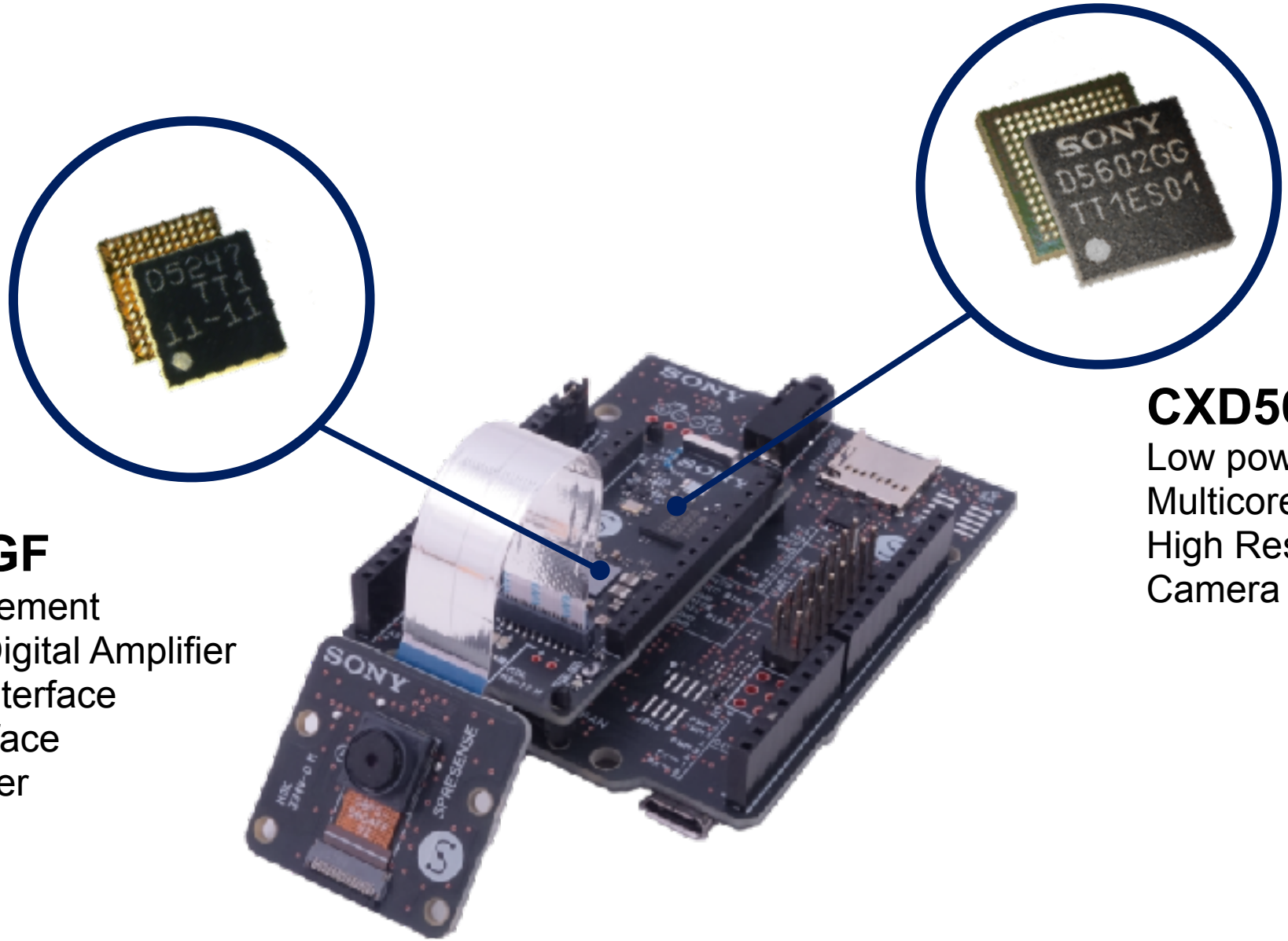
CAMERA BOARD

MAIN BOARD



EXTENSION BOARD

SONY



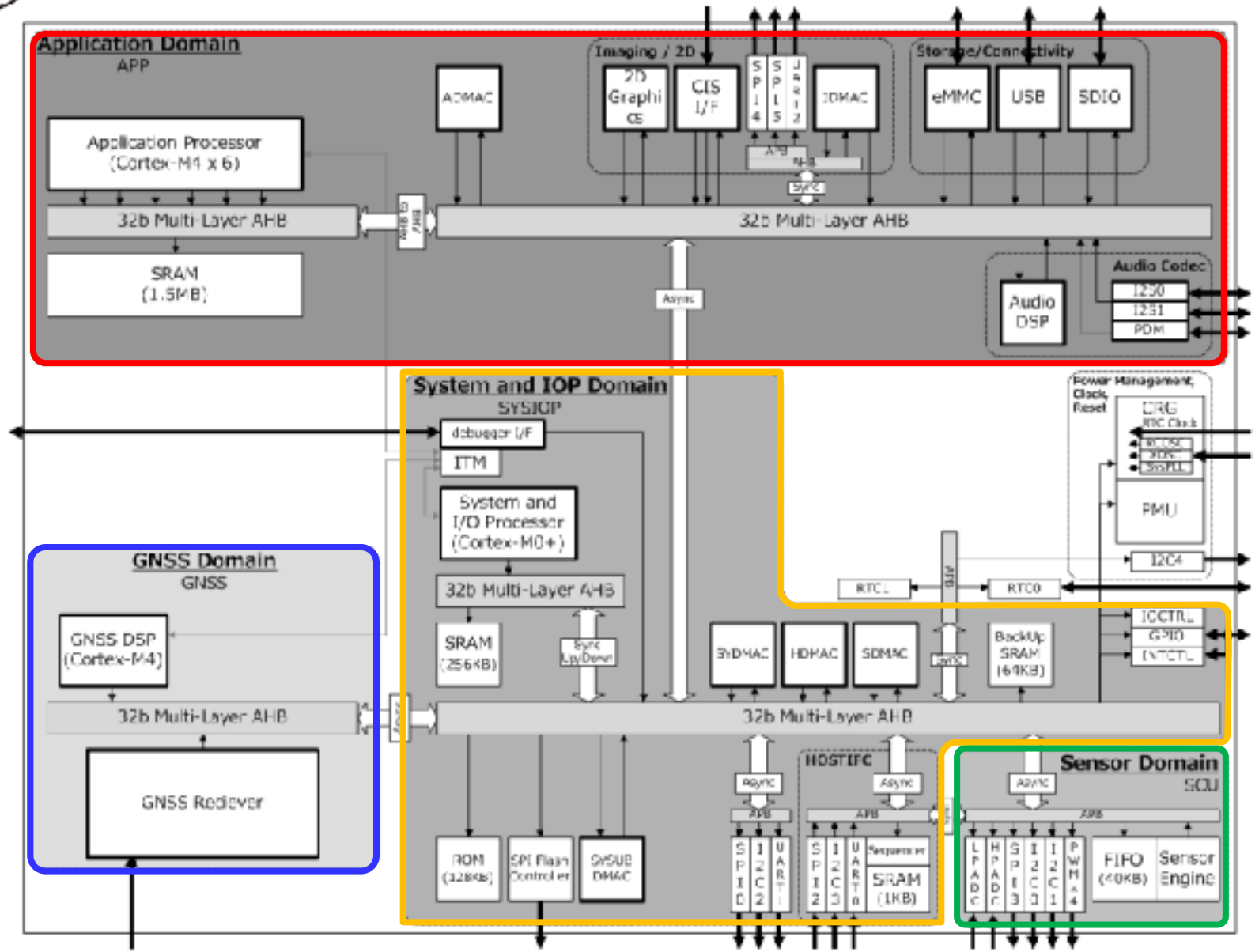
CXD5247GF

Power Management
Class D Full Digital Amplifier
Microphone Interface
Speaker Interface
Battery Charger

CXD5602GG

Low power GNSS functions
Multicore processor
High Resolution Audio Codec
Camera Interface

Inside of CXD5602



Application Domain

Application SW works.

System and IOP Domain

System Management
and Power domain control

Sensor Domain

Intelligent Sensor Hub

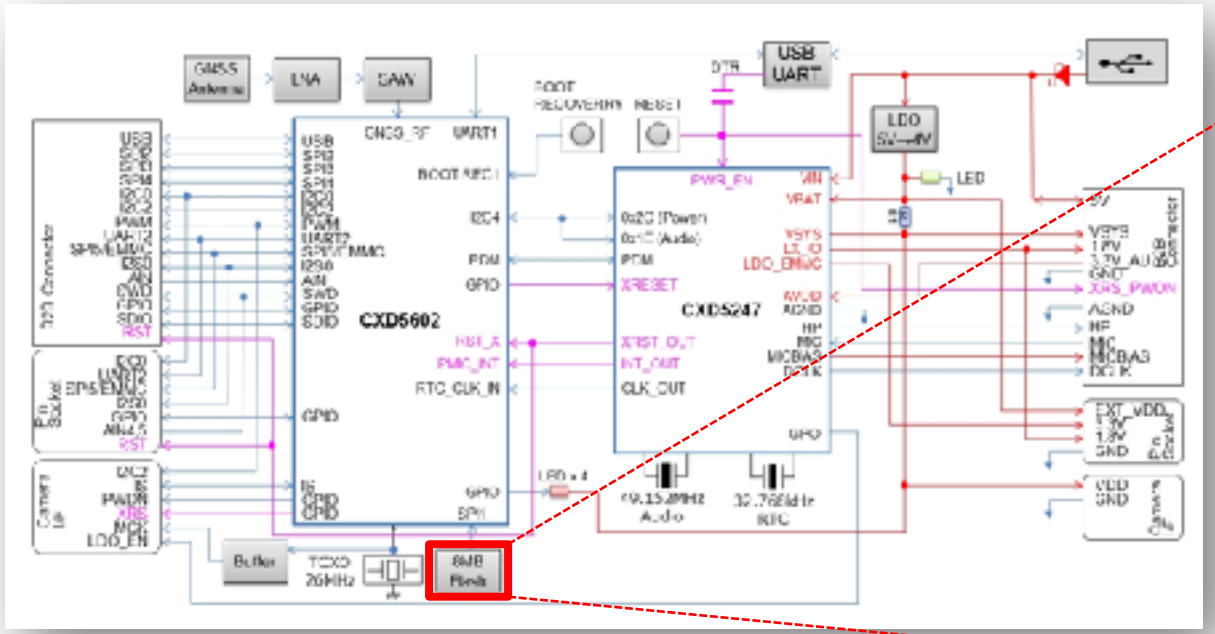
GNSS Domain

Independent GNSS positioning

SONY

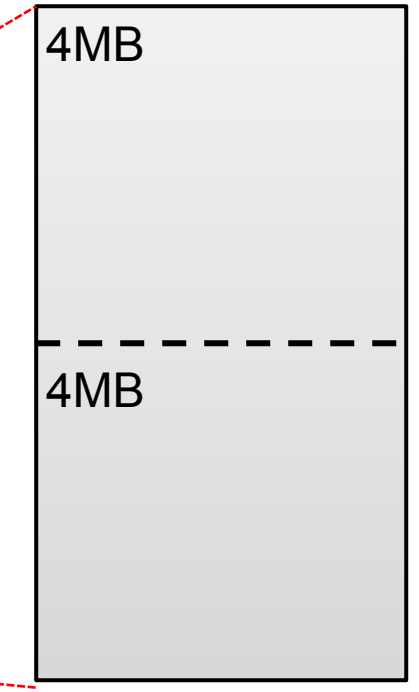
SPI-Flash Filesystem

Spresense Main Board Block Diagram



No Internal Flash in CXD5602.
Non-volatile storage is 8MB SPI-Flash.

8MB SPI-Flash



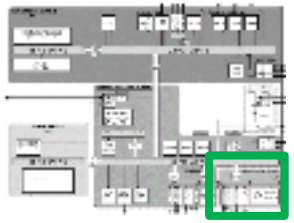
Sony original filesystem to store spk(/espk) files

Free area for user.
SMART FS of NuttX by default.

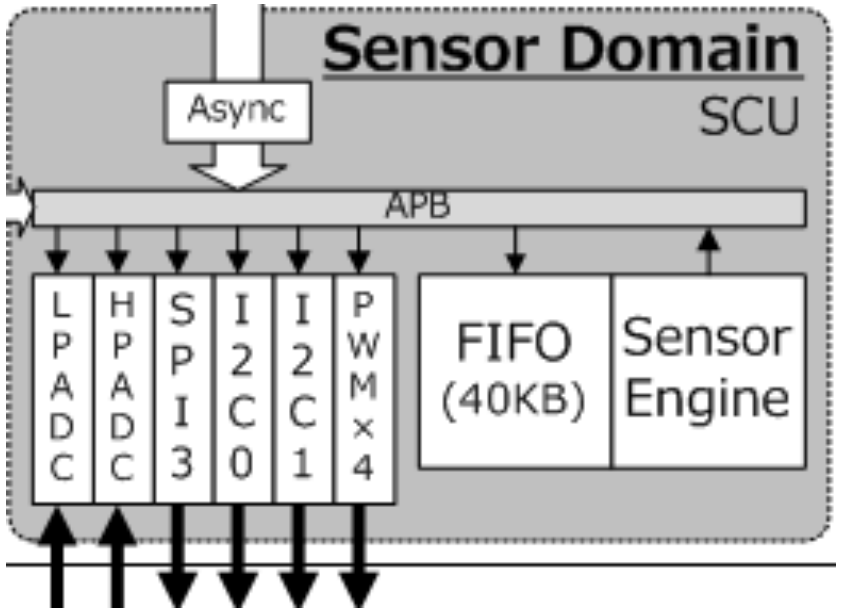
In SPI-Flash, there are 2 partitions.

SONY

Sensor Domain (Sensor Control Unit)

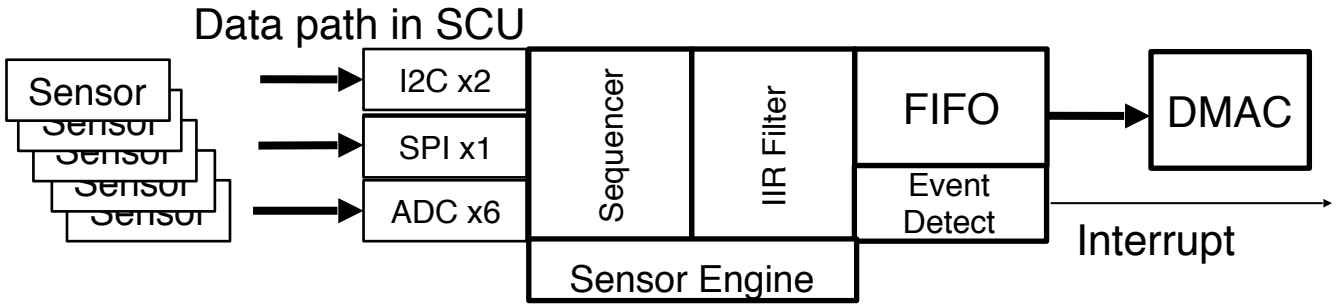


Intelligent internal sensor hub with three functions

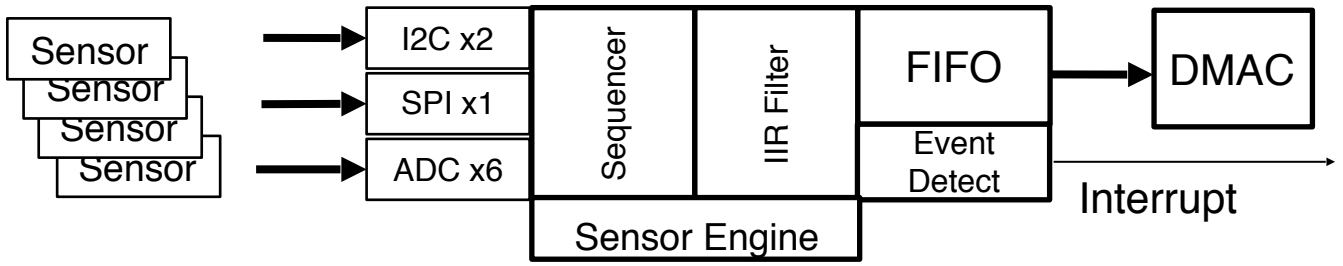


- Automatic polling of sensor data using sequencer circuitry
- 2-stage IIR Filter
- "Event Detector" which detects the peak etc. of the sensor signal

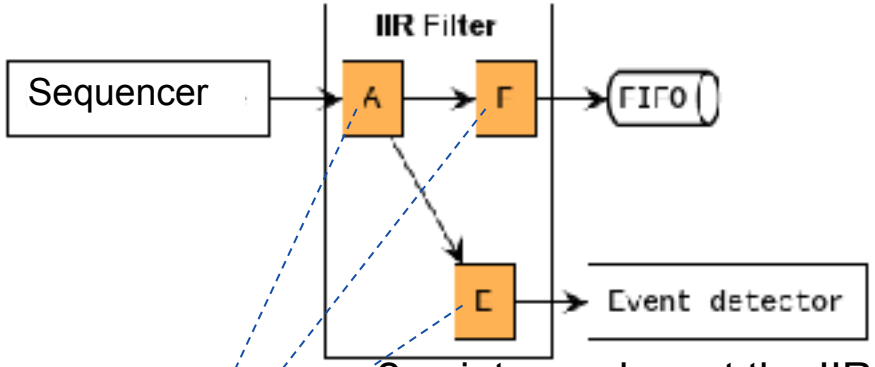
- I/F for sensors (I2C/SPI/ADC)
- 40KB FIFO RAM for storing sensor data



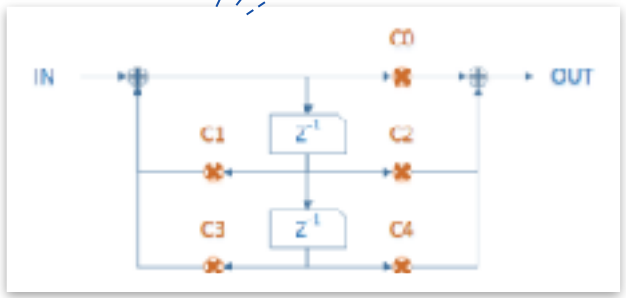
Sensor Domain (Sensor Control Unit)



IIR Filter

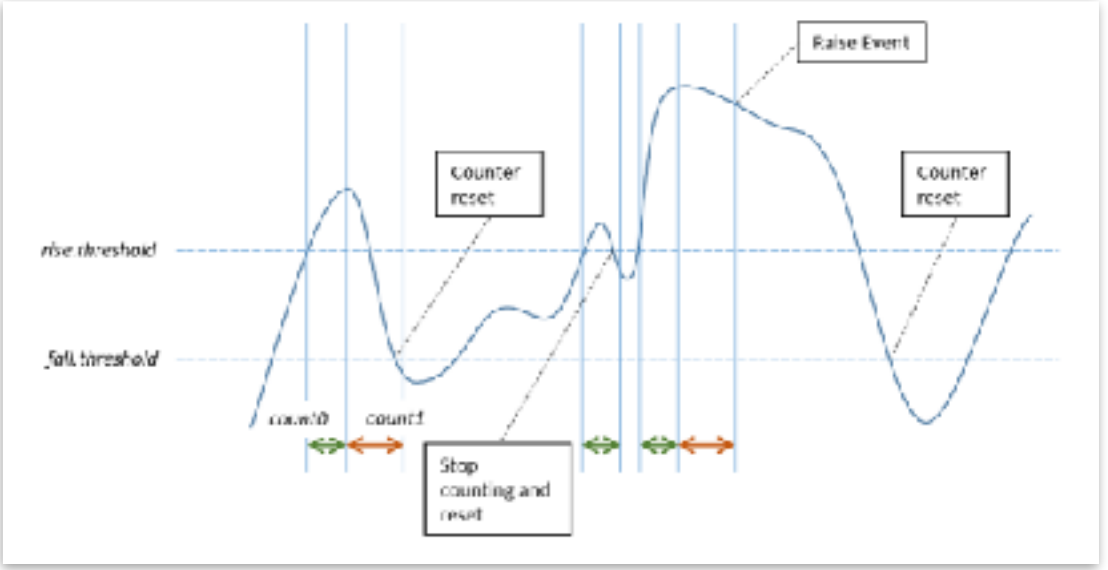


2 points can be set the IIR filter in 3 place.

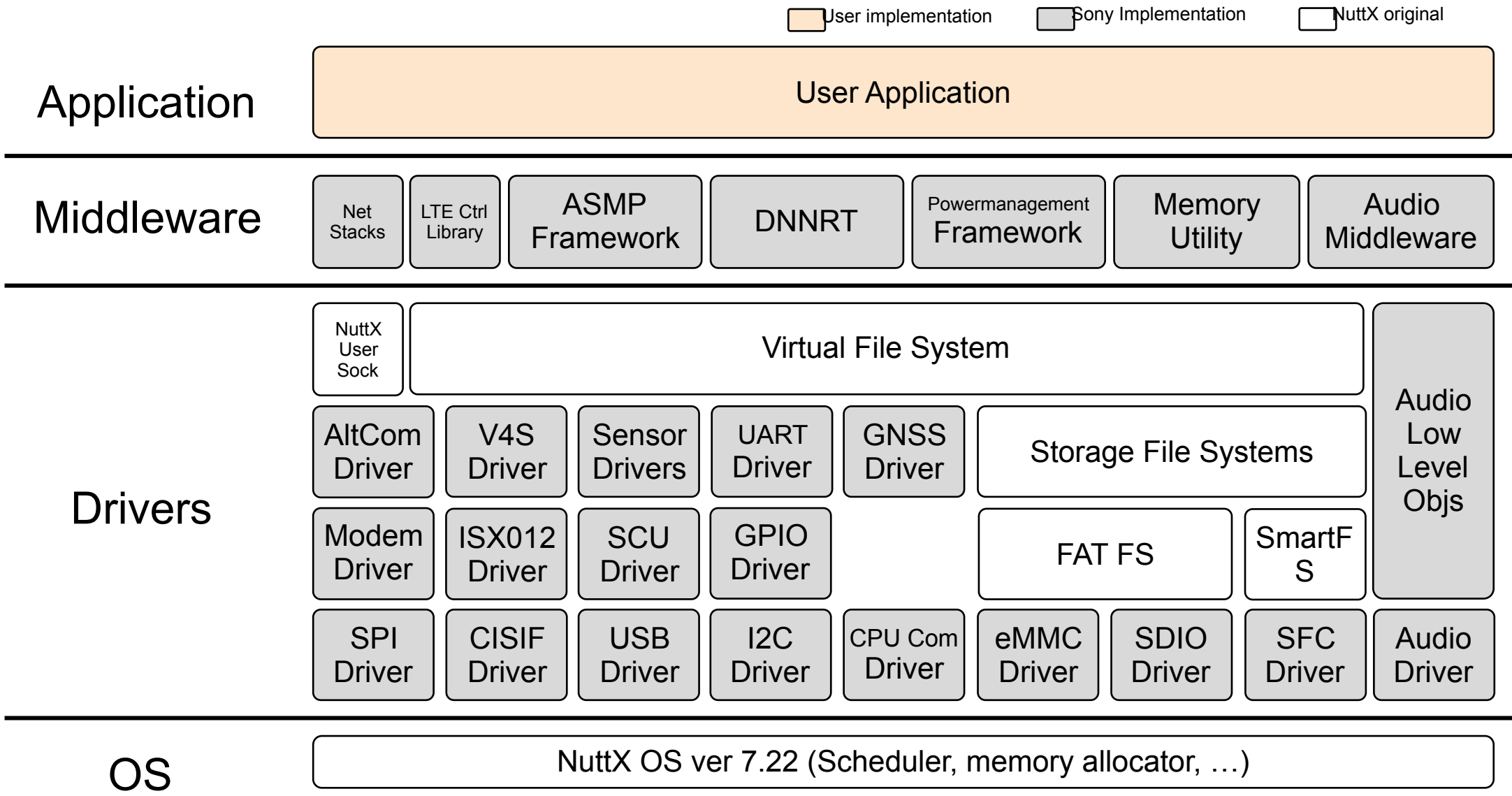


Event Detector

Has rising counters and falling counter with threshold settings.
It detects the signal waveform to some extent and wakes up the CPU by using those settings.

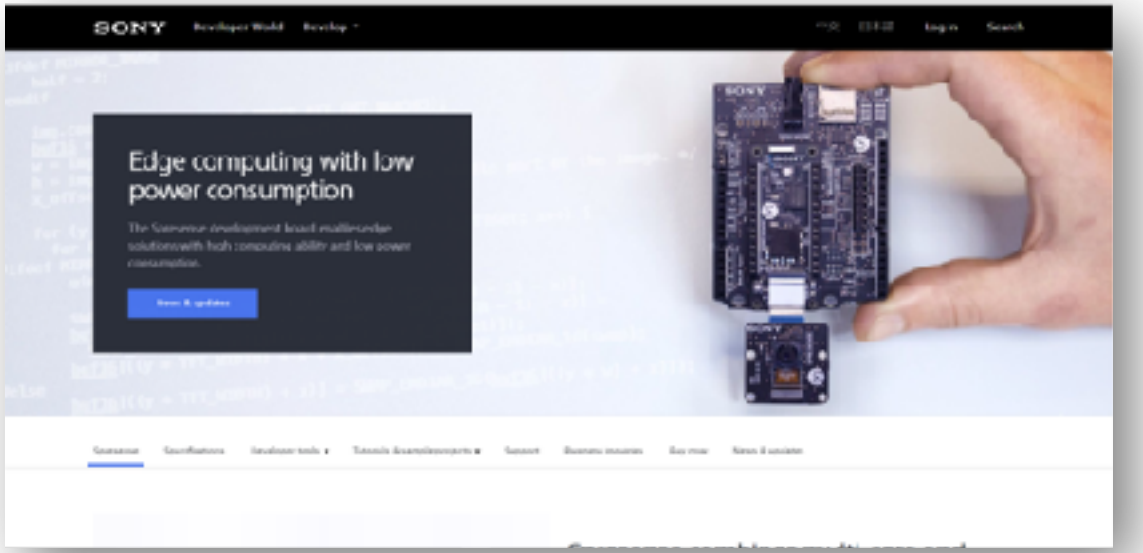


Spresense SW Stack



SONY

Documentations



- Software Documents
  - https://developer.sony.com/develop/spresense/
- HW Documents
  - Board Descriptions
    - https://developer.sony.com/develop/spresense/developer-tools/introduction
  - Board Schematics
    - https://github.com/sonydevworld/spresense-hw-design-files
  - CXD5602 Chip Specifications
    - https://www.sony-semicon.co.jp/products_en/spresense/index.html

Current Status of upstreaming to Original NuttX

Done

CORE
adc
allocateheap
clock
composite
cpufifo
delay
dmac
farapi
flash
gpio
gpioint
i2c
i2cdev
gpioif
icc
idle
leds
pinconfig
pmic
Powermgr

pwm
rtc
scu
sdcard
sdhci
serial
sfc
spi
spisd
sysctl
timer
timerisr
uart
udmac
uid
usbdev
usbmsc
userleds
Wdt

WIFI
gs2200m

SENSORS (I2C)
ak09912
bmi160
Bmp280

DISPLAY
ili9340
lpm013m091a

CAMERA
video
cisif
isx012

Extra HW
buttons
charger
emmc
gauge

GRAPHIC
ge2d

Many functions have been already upstreamed by Alin-san <Alin.Jerpelea@sony.com> and Ishikawa-san <Masayuki.Ishikawa@sony.com>.

Many thanks to them.

Not Yet

AUDIO
audio_amp
audio_baseband

MODEM
altmdm
altmdm_spi

GNSS
cpu1signal
geofence
gnss

BT

SENSORS (SCU)
ak09912
apds9930
apds9960
bcm20706
bh1721fvc
bh1745nuc
bm1383glv
bm1422gmv
bmi160
bmp280
kx022
lt1pa01
Rpr0521rs

EXTRA
backuplog
crashdump

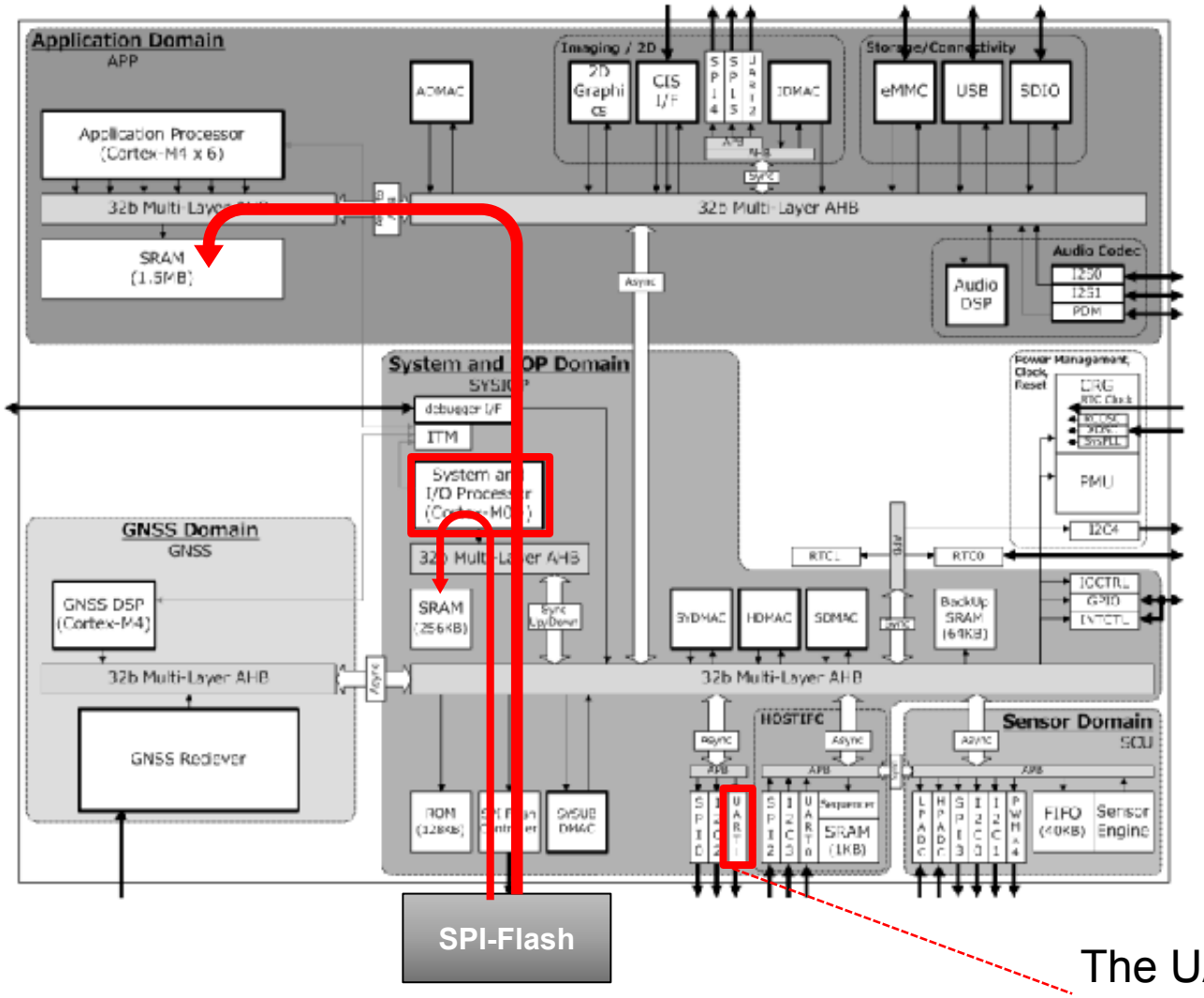
ASMP
sph

HELPER
libm_errno

Hands-on with NuttX

- nsh
  - With explanation of bootup mechanism of CXD5602
- usbnsh
- rndis
- WiFi with gs2200m
- OpenOCD ICE debugging
  - Just Demo because board needs pin header to connect JTAG ICE.

Bootup mechanism of CXD5602



1. System and IoP processor which is Coretex-M0 starts load own binary named loader.spk from SPK partition in SPI-Flash by Internal ROM Code.
2. loader.spk runs on the Coretex-M0.
3. loader.spk loads nuttx.spk from SPK partition in SPI-Flash to Application RAM.
4. And then, the nuttx.spk starts to run on one Application processor which is Coretex-M4F.

The UART1 is used for flashing executable binaries and default console

Flash bootloader and other binaries.

- Download binaries
  - https://developer.sony.com/file/download/download-spresense-firmware-v1-3-000
  - and expand downloaded zip file.
    - loader.espk, gnss.espk, dnnrt-mp.espk, AESM.espk

use this microUSB port.for flashing.

- Flash them onto Spresense Main Board

In your nuttx.git cloned directory,

$ tools/flash_writer.py -c <port name like /dev/ttyUSB0> -b 1152000 <path to espk file>

e.x. : & tools/flash_writer.py -c /dev/ttyUSB0 -b 1152000 path/to/espk/loader.espk

- Tips option –b is baud rate for the UART. default is 115200. 1152000 is better (10 times higher) but it depends on your PC.
  If you got error when you use 1152000, remove –b option to use default baud rate.

mkspk tool need to be downloaded for making spk file

- To load nuttx binary on Spresense, the file format to have to be converted to spk format which is Son original format.
- To convert, mkspk tool must be downloaded.

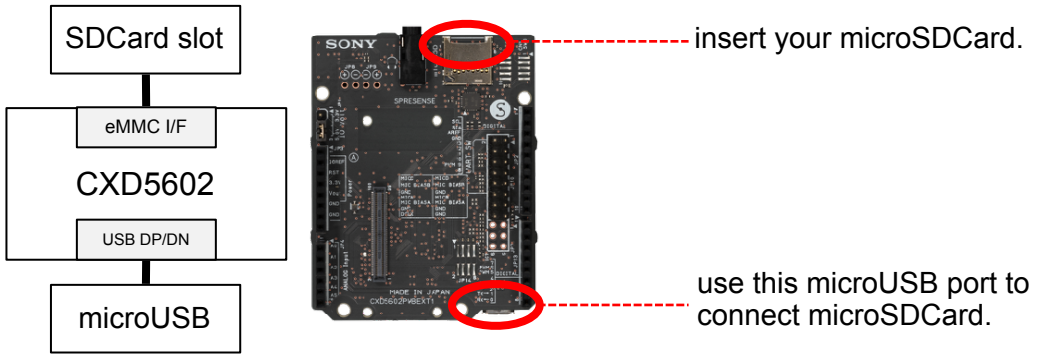
- Instructions
  - Go to nuttx/tools directory and clone the tool repository.
    $ cd path/to/your_tools_directory_in_nuttx_clone
    $ git clone https://github.com/sonydevworld/cxd56.git

nsh

- Instructions
  - Go to nuttx/tools and configure spresense/nsh
    $ cd path/to/your_nuttx_tools
    $ ./configure.sh spresense/nsh
  - Go up a directory and make
    $ cd ..
    $ make
  - Flash built image
    $ tools/flash_writer.py –c <your ttyUSB path> -b 1152000 nuttx.spk
- How to execute
  - Access to NuttShell by using like minicom.
  - You can see NuttShell prompt on it.

usbnsh

- Spresense USB has microSDCard slot. USBMSC can access to the microSDCard via USB on Spresense. This application uses CXD5602 USB and eMMC I/F.

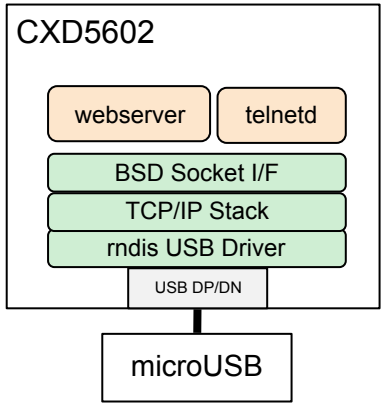


- Instructions
  - Go to nuttx directory and clean by using “distclean”
    $ cd path/to/your_nuttx
    $ make distclean
  - Go to nuttx/tools and configure spresense/usbnsh
    $ cd path/to/your_nuttx_tools
    $ ./configure.sh spresense/usbnsh
  - Go up a directory and make
    $ cd ..
    $ make
  - Flash built image
    $ tools/flash_writer.py –c <your ttyUSB path> -b 1152000 nuttx.spk
- How to execute
  - On your nsh prompt, execute mscom command
    > mscom

rndis

- By using this, Spresense and PC can communicate each other as network connection.

- Instructions
  - Go to nuttx directory and clean by using “distclean”
    $ cd path/to/your_nuttx
    $ make distclean
  - Go to nuttx/tools and configure spresense/rndis
    $ cd path/to/your_nuttx_tools
    $ ./configure.sh spresense/rndis
  - Go up a directory and make
    $ cd ..
    $ make
  - Flash built image
    $ tools/flash_writer.py –c <your ttyUSB path> -b 1152000 nuttx.spk



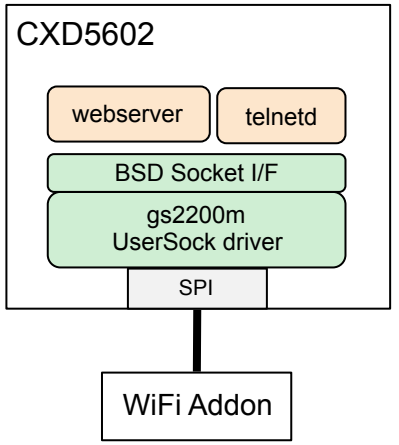
rndis

- How to execute
  - On Spresense side
    - Execute webserver on NuttShell
      > webserver &
    - Confirm ip address
      > ifconfig
  - On your PC side
    - Confirm network interface name of the rndis device.
      $ ifconfig
    - Setup ip address on the network interface.
      $ sudo ifconfig <network interface name> <ip address> netmask 255.255.255.0
  - Then PC can access to Spresense.
    - See IP address of Spresense via browser
    - Connect telnet to Spresense.

WiFi Add-on

- NuttX has UserSocket mechanism which is to support network devices that has protocol stack inside. gs2200m has tcp/ip stack in the device. At this hans-on, Spresense behaves as AccessPoint and will make own network.
  And web-server and telnet daemon run, so PC connected to the Spresense network can access to Spresense via telnet andWebBrowser.

Connection between WiFi Addon and Spresense MainBoard



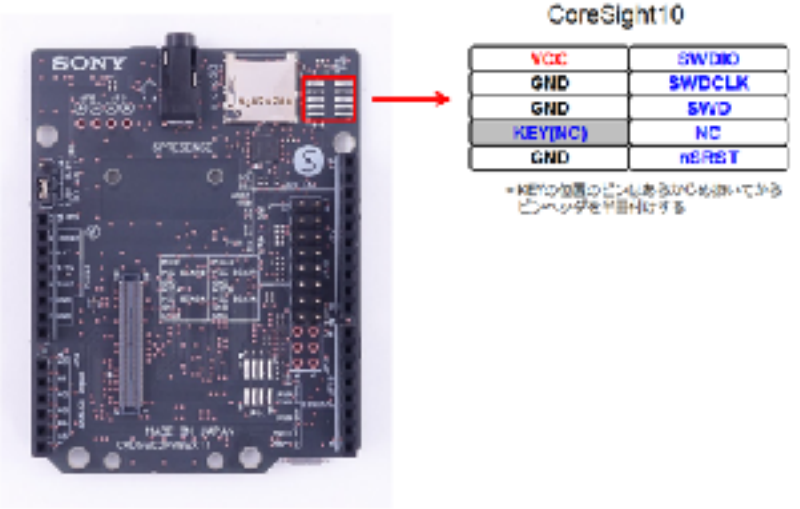
- Instructions
  - Go to nuttx directory and clean by using “distclean”
    $ cd path/to/your_nuttx
    $ make distclean
  - Go to nuttx/tools and configure spresense/wifi
    $ cd path/to/your_nuttx_tools
    $ make distclean
    $ ./configure.sh spresense/wifi
  - Go up a directory and make
    $ cd ..
    $ make
  - Flash built image
    $ tools/flash_writer.py –c <your ttyUSB path> -b 1152000 nuttx.spk

WiFi Add-on

- How to execute
  - On Spresense side
    - Execute gs2200m user sock daemon as AP mode on NuttShell
      > gs2200m –a <channel number (1 to 11)> <unique ssid name> <10 digits password for WEP> &
      ex. > gs2200m –a 1 taka_net 1234567890 &
    - Execute webserver and telnetd as a daemon on NuttShell
      > webserver &
      > telnetd &
    - Confirm IP address of Spresense on NuttShell
      > ifconfig
  - On PC side
    - Connect the SSID you set, and access to the IP address of Spresense from a browser.

OpenOCD debugging

- CXD5602 has JTAG ICE I/F, but Spresense Extension board has no pin header for it.
  So if you want to use it, you need soldering on that.



- CXD5602 JTAG ICE is supporting CMSIS-DAP.
  Our recommendation of ICE adaptor is
  - Keil ULINK2
  - NXP LPC-Link2
  - But other ICE supporting CMSIS-DAP can be used.
    ex. ULINK-ME can be used.

- CXD5602 config for OpenOCD is already marged in https://github.com/gnu-mcu-eclipse/openocd

OpenOCD debugging

- How to use
  - Start OCD process
    $ openocd –f interface/cmsis-dap.cfg –f target/cxd5602.cfg
  - Start GDB
    $ arm-none-eabi-gdb
  - In GDB,
    - Connect to OpenOCD
      (gdb) target remote localhost:3333
    - Load nuttx elf file
      (gdb) monitor reset halt
      (gdb) file nuttx
      (gdb) load
    - Set break points like
      (gdb) b hello_main
    - Start nuttx
      (gdb) continue
  - On NuttShell side, execute hello, then break is occured
    > hello

Demos with Spresense SDK

- Audio Player
- GNSS
- ASMP : Multicore framework

Spresense SDK

Spresense SDK is built based on NuttX v7.22. In the repository, there are 2 big part.
One is nuttx, the other is sdk.

```
spresense.git tree
 +----- nuttx/
 |          Spresense kernel based on NuttX v7.22
 |
 +----- sdk/
 |          Spresense specific implementations
 |
 +----- examples/
 |          Spresense application examples
 |
 +----- externals/
            External libraries SDK is using.
```

Build system is based on NuttX but some differences exist.
Especially, nuttx and sdk build is split.

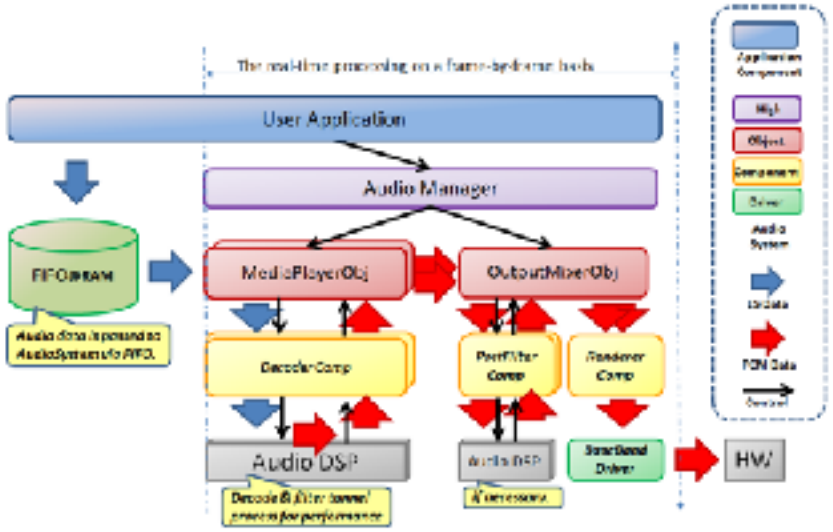
In this workshop, I will show you some Spresense SDK demos from source code build.

Audio Player

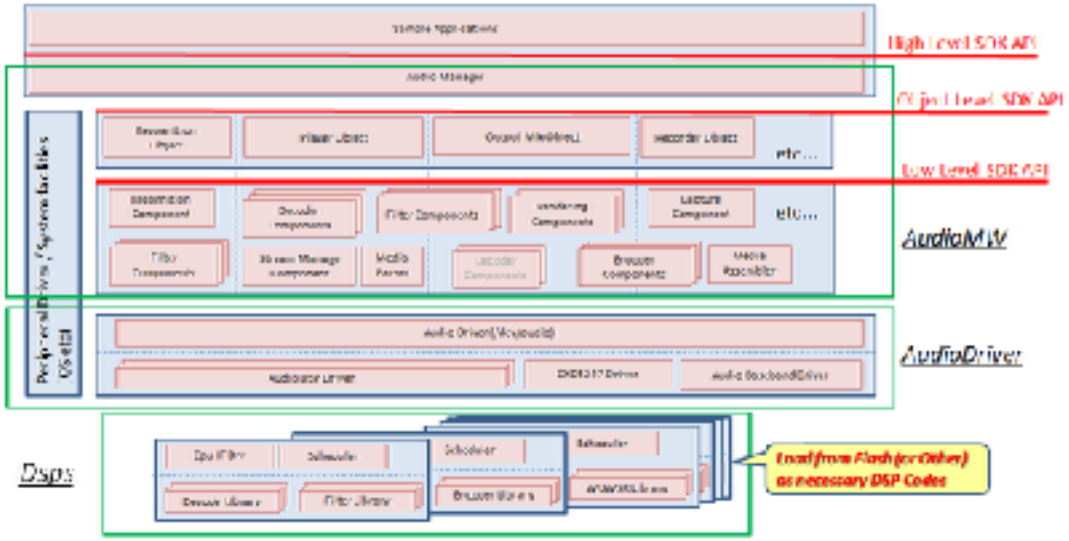
- Spresense has Audio in/out functionality.
- The software stack is like OpemMAX.

- Current supporting codecs are
  - MP3 and WAV

- In this demo, it Spresense play MP3 file.



Audio Software Stack of Spresense SDK



Gather and compose some components for behave as Audio Player.

Audio Player

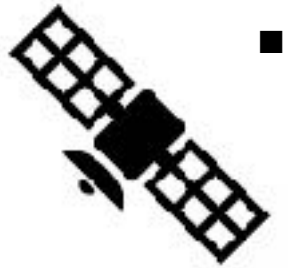
- Build NuttX kernel
  - Instructions
    - Go to spresense/sdk/ directory
      $ cd path/to/sdk_directory_in_cloned_spresense_repository
    - Kernel default configure, you can choose “release” or “debug”
      $ tools/configure.py –k release
    - Build kernel
      $ make buildkernel
    - Then nuttx side is built.
- Build sdk side with audio_player examples
  - The example code is in spresense/examples/audio_player
  - Instructions
    - In the same directory “sdk”, configure audio_player example configuration with sdcard.
      $ tools/configure.py examples/audio_player device/sdcard
    - Enable USB MSC command by using menuconfig
      $ make menuconfig
      [Syste tools] -> [USB Mass Storage Deice Commands]
    - Build it
      $ make
    - Then nuttx.spk is created on the directory.

Audio Player

- How to execute
  - Needed files should be store in SDCard.
    - Copy spresense/examples/audio_player/AUDIO and spresense/examples/audio_player/PLAYLIST into top directory of SDCard.
    - Make “BIN” directory on top of SDCard.
    - Copy spresense/sdk/modules/audio/dsp/MP3DEC into the “BIN” directory.
  - Then execute player on NuttShell
    > player

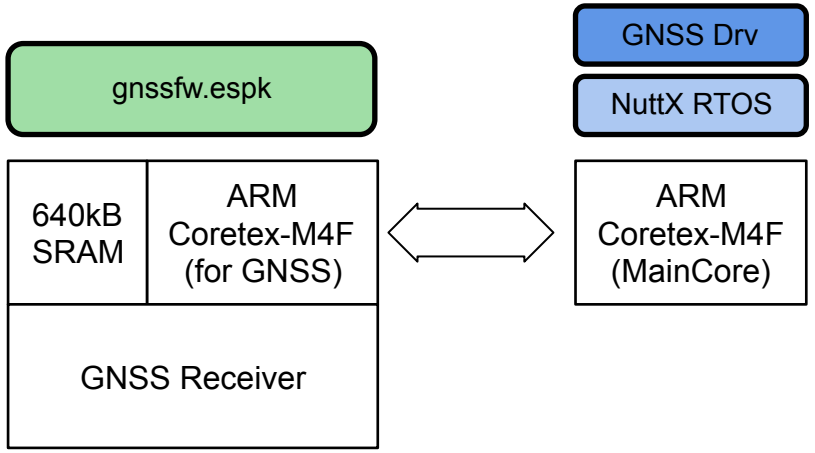
GNSS

Spresense GNSS is completely independent from NuttX. The driver on NuttX is just stub I/F to GNSS CPU (Coretex-M4F).

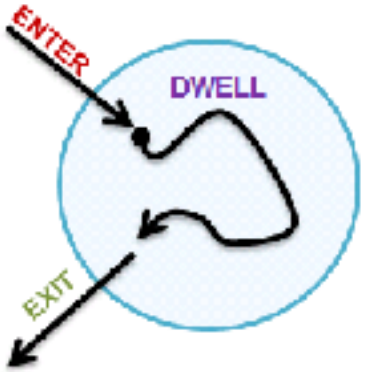


■ Supported satelite :
GPS / QZSS / GLONASS
(/ Beidu / Galileo)

■ GNSS core will communicate with Application CPU NuttX runs.



■ Geofence is also support in the GNSS core.



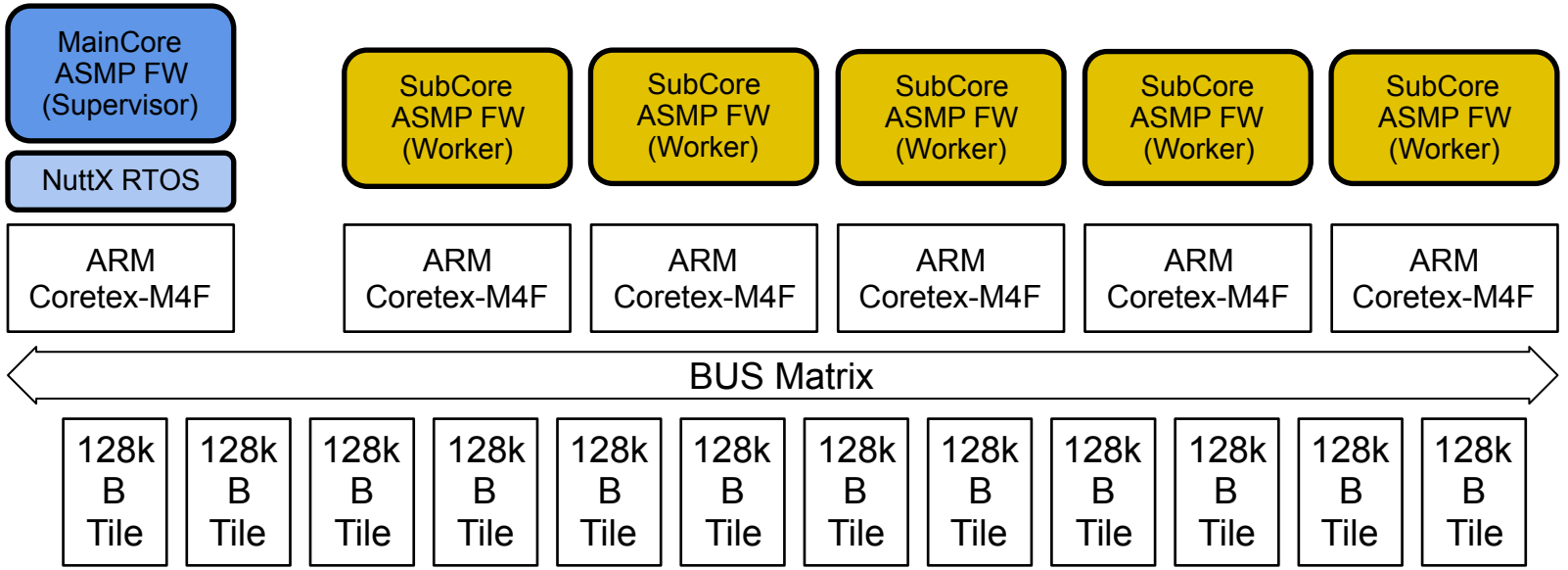
GNSS

- Build sdk side with gnss examples
  - The example code is in spresense/examples/gnss
  - Instructions
    - In the same directory “sdk”, configure audio_player example configuration.
      $ tools/configure.py examples/gnss
    - Build it
      $ make
    - Then nuttx.spk is created on the directory.
- How to execute
  - Execute gnss command on NuttShell
    > gnss
  - To detect your position if satellite signal is detected.

ASMP

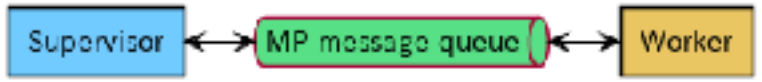
Managing Sub-CPUs and Memory resources.

- CPU Resource
- Manage Memory Tiles
- Provide simple library for synchronizing CPUs



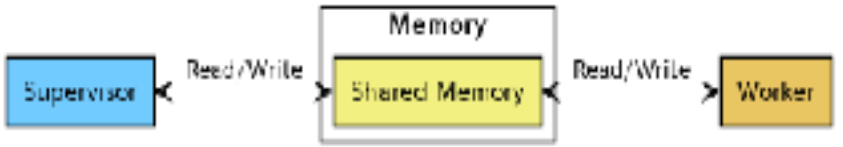
- Three synchronize mechanism

— Message queues between CPUs—



— Mutex between CPUs —

— Shared memory —



ASMP

- Build sdk side with asmp examples
  - The example code is in spresense/examples/asmp
  - This example make one binary inclutind sub-core elf file in ramfs.
  - So first, mount the ramfs and then execute sub-core elf file on another Coretex-M4F.

  - Instructions
    - In the same directory “sdk”, configure asmp example configuration.
      $ tools/configure.py examples/asmp
    - Build it
      $ make
    - Then nuttx.spk is created on the directory.
- How to execute
  - Execute gnss command on NuttShell
    > gnss

Future plan

Future plan

Aug	Sep	Oct	Nov	Dec	'20/Jan	Feb	Mar

Upstreaming

Update NuttX version of Spresense SDK

Not Yet

AUDIO	SENSORS	EXTRA
audio_amp	(SCU)	backuplog
audio_baseb	ak09912	crashdump
and	apds9930	
	apds9960	ASMP
MODEM	bcm20706	sph
altmdm	bh1721fvc	
altmdm_spi	bh1745nuc	HELPER
	bm1383glv	libm_errno
GNSS	bm1422gmv	
cpu1signal	bmi160	
geofence	bmp280	
gnss	kx022	
	lt1pa01	
BT	Rpr0521rs	

These function is under consideration for upstreaming because those are very specific to CXD5602.

SPRESENSE™

I hope you enjoy it!

Thank you.

SONY

Appendix

“Xperia Ear Duo” is a commercial product from Sony Mobile provided March/2018.
And it is using CXD5602 and Spresense SDK including NuttX v7.22.