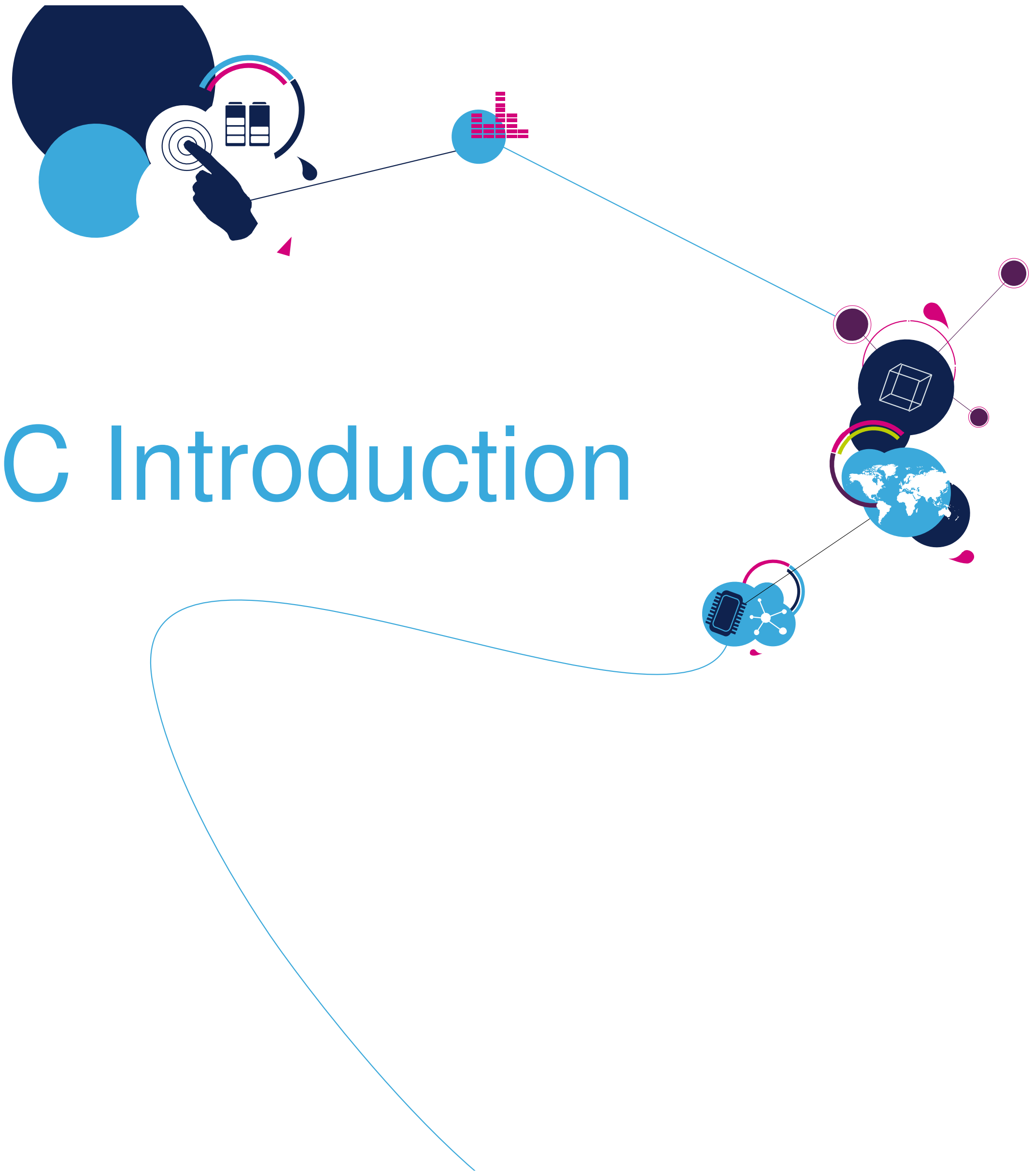
Qualcomm p/f SSC Introduction

Joy YNAG

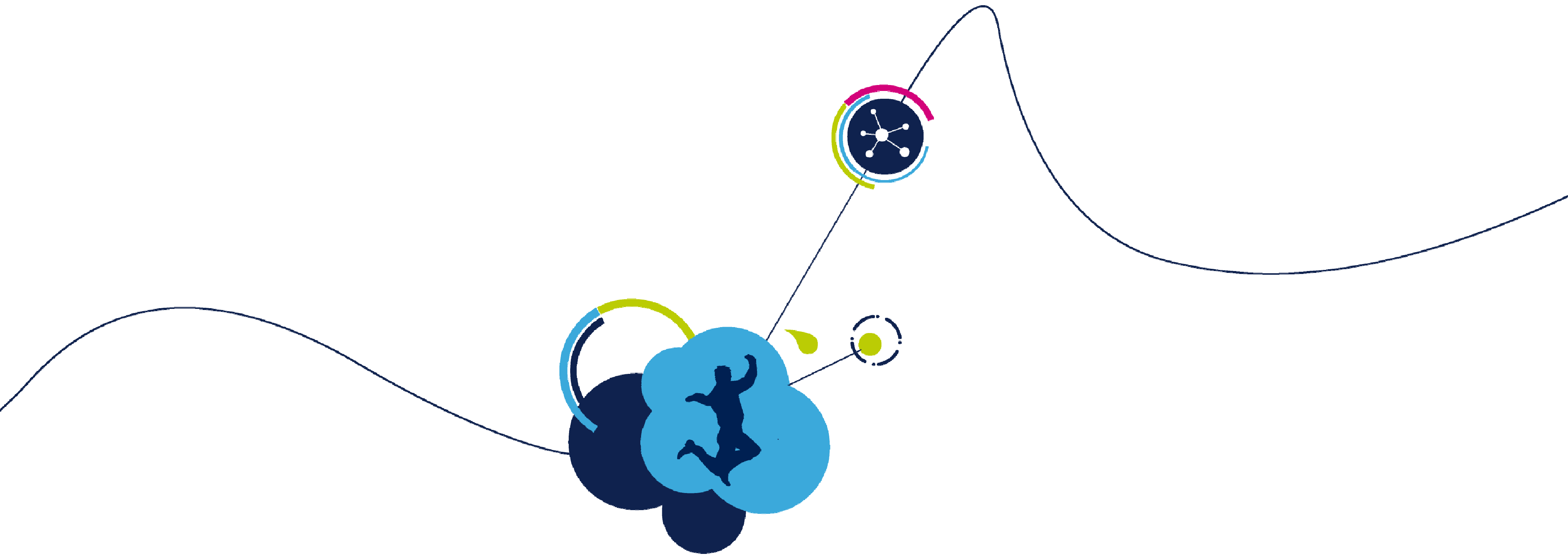
12th/Jun. 2018

life.augmented



History

Version	Author/Date	Description	Note
1.0	Joy YANG/12th/Jun. 2018	Initialize	

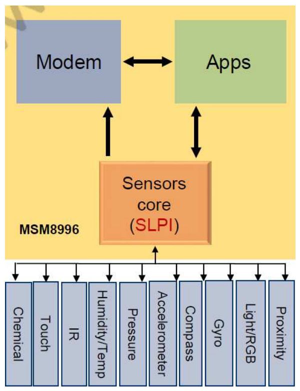


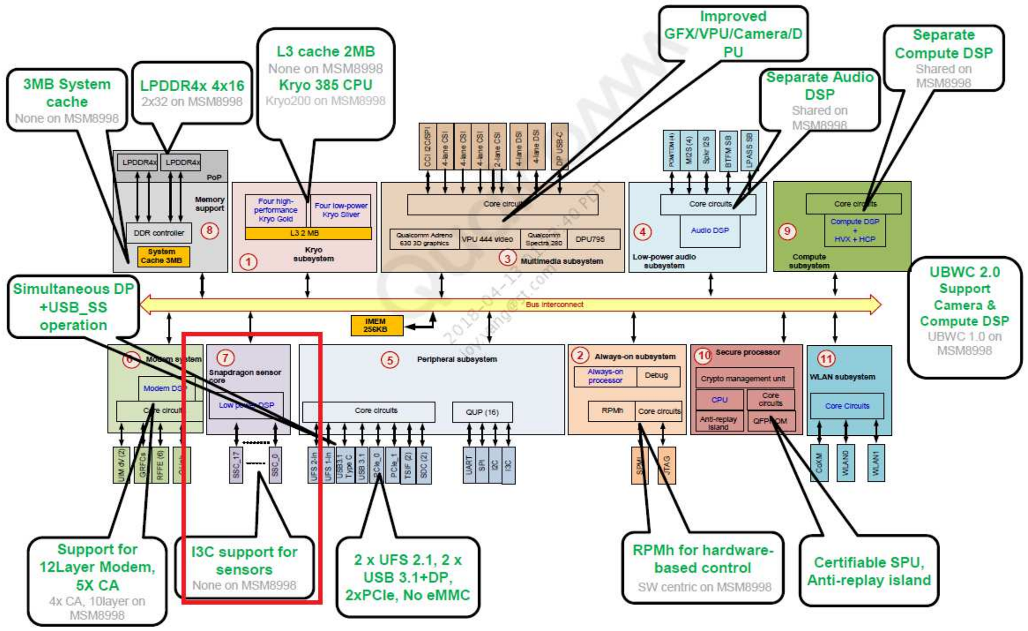
硬件架构介绍

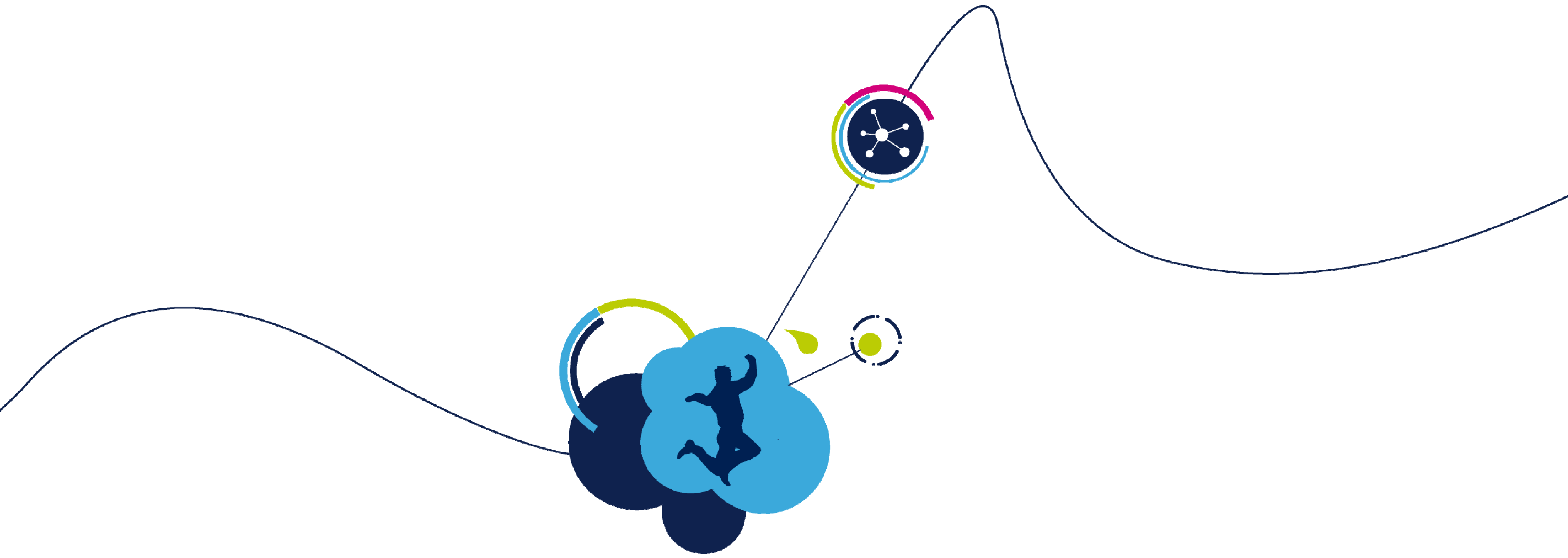
life.augmented

SSC

- SSC（Snapdragon Sensor Core）本质是集成在骁龙应用处理器中的sensor hub，
  - 中低端平台中其与音频集成在一起，部分硬件资源共享， 称为aDSP（audio DSP），
  - 高端平台中其独占硬件资源， 称为SLPI（Sensor Low Power Island），
  - 通讯接口：I2C， SPI， I3C（SDM845）.
- 优势：
  - 高性能，
  - 低硬件成本，
  - 低功耗





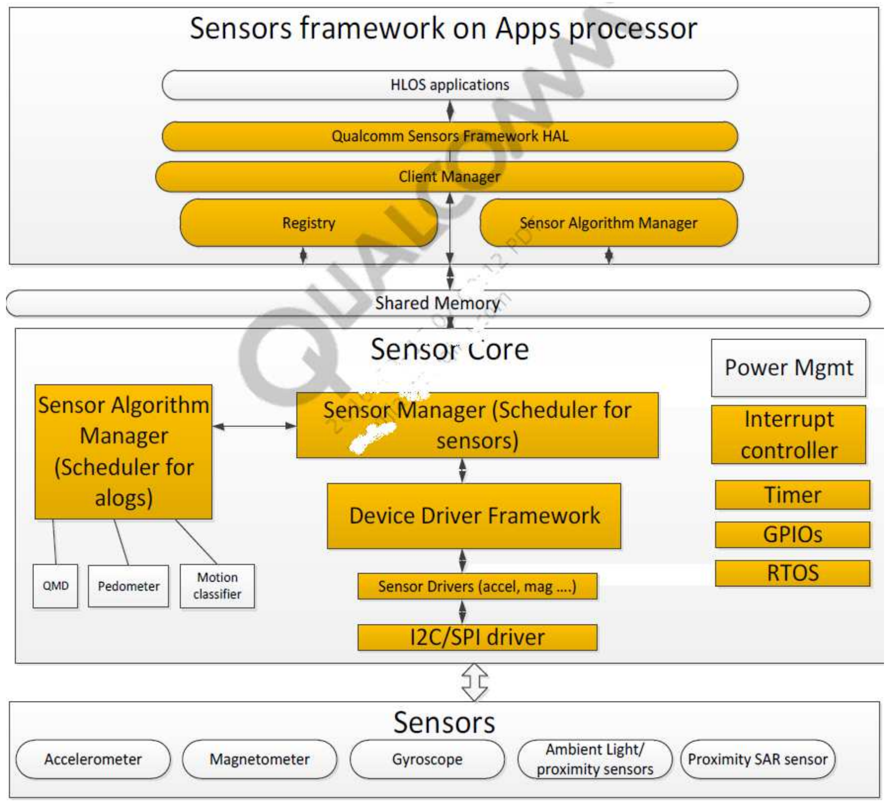


软件架构介绍

life.augmented

软件架构-DDF简介

- DDF（Device Driver Framework）架构适用于2017年9月之前的应用处理器，
- AP侧：
  - ✓HAL，：实现应用层通用接口，
  - ✓守护进程：客户端请求监听
  - ✓SAM，：算法应用管理相关，
- SSC侧：
  - ✓SMGR：电源，配置及数据流处理，
  - ✓DDF：框架接口封装，
  - ✓SAM：算法应用管理相关，



life.augmented

软件架构-DDF驱动移植

- 在<adsp_proc/slpi_proc>\sensors\dd\qcom\ inc\sns_dd.h中添加入口函数定义，

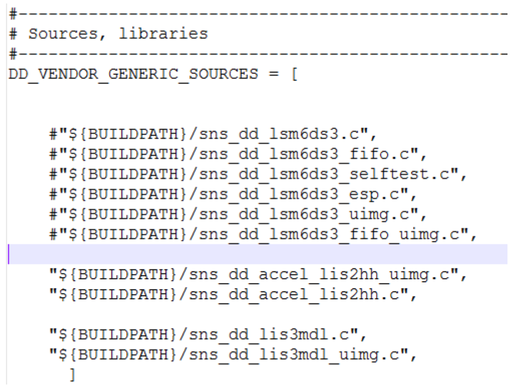
  /* Make the device function lists sharable with SMGR */.

  extern sns_ddf_driver_if_s sns_dd_<new_sensor_model>_if;

- 添加新驱动代码(<adsp_proc/slpi_proc>\sensors\dd\qcom\src)及修改编译脚本（<adsp_proc/slpi_proc>\sensors\dd\qcom\build\dd_qcom.scons），

- 在<adsp_proc/slpi_proc>\sensors\build\Sensors.scons定义新驱动配置宏，用于将驱动源码和UUID绑定，

  env.Append(CPPDEFINES = ["CONFIG_SUPPORT_<NEW_SENSOR_MODEL>"])



```
#--------------------------------------------------
# Sources, libraries
#--------------------------------------------------
DD_VENDOR_GENERIC_SOURCES = [


    #"${BUILDPATH}/sns_dd_lsm6ds3.c",
    #"${BUILDPATH}/sns_dd_lsm6ds3_fifo.c",
    #"${BUILDPATH}/sns_dd_lsm6ds3_selftest.c",
    #"${BUILDPATH}/sns_dd_lsm6ds3_esp.c",
    #"${BUILDPATH}/sns_dd_lsm6ds3_uimg.c",
    #"${BUILDPATH}/sns_dd_lsm6ds3_fifo_uimg.c",

    "${BUILDPATH}/sns_dd_accel_lis2hh_uimg.c",
    "${BUILDPATH}/sns_dd_accel_lis2hh.c",

    "${BUILDPATH}/sns_dd_lis3mdl.c",
    "${BUILDPATH}/sns_dd_lis3mdl_uimg.c",
      ]
```

life.augmented

软件架构-DDF驱动移植

- 申请新的UUID， 并在AP侧和SSC侧进行定义，

  <adsp_proc/slpi_proc>\Sensors\common\inc\sns_reg_common.h

  android\vendor\qcom\proprietary\sensors\dsps\sensordaemon\common\inc\ sns_reg_common.h

  ```
  #define SNS_REG_UUID_<NEW_SENSOR_MODEL> \
  {0x11,0x22,0x33,0x44,0x55,0x66,0x77,0x88,0x99,0xAA,0xBB,0xCC,0xDD,0xEE,0xFF,0x11}
  ```

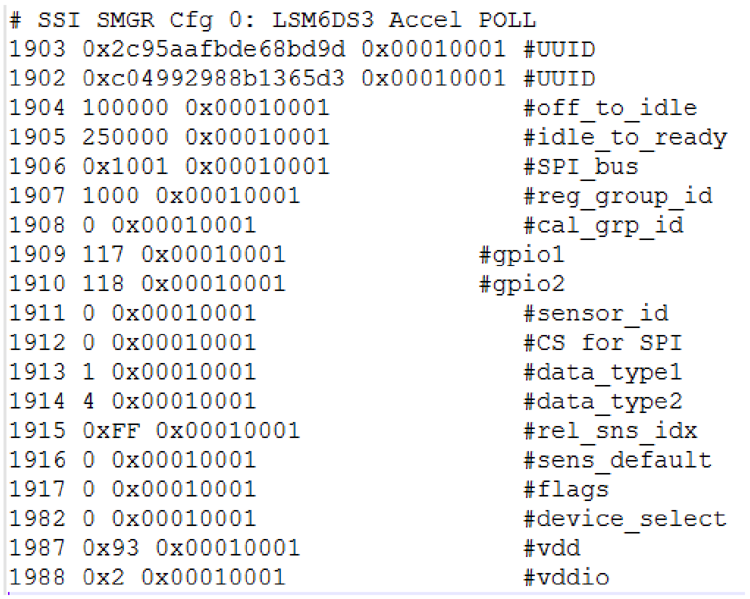
- 通过之前定义的新驱动配置宏将驱动和UUID绑定，

  <adsp_proc\slpi_proc>/ Sensors/smgr/src/sns_smgr_reg.c, update smgr_sensor_fn_ptr_map[]:

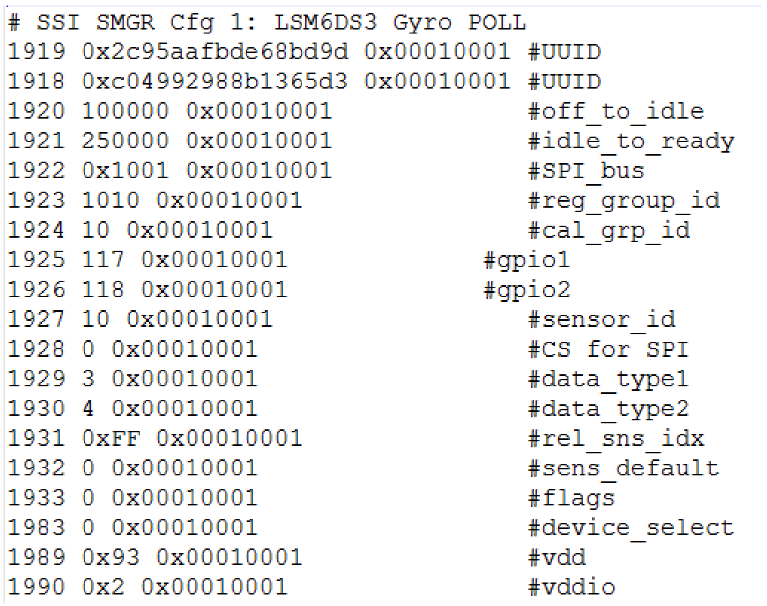
  ```
  #ifdef CONFIG_SUPPORT_<NEW_SENSOR_MODEL>
  { SNS_REG_UUID_XXXX, & sns_dd_<new_sensor_model>_if},
  #endif
  ```

软件架构-DDF驱动移植

- 配置文件
  - 用于选择所需要加载的驱动（通过UUID），
  - 特定的硬件资源配置，
  - 文件所在路径：/etc/sensors/sensor_def_intrinsyc.conf



```
# SSI SMGR Cfg 0: LSM6DS3 Accel POLL
1903 0x2c95aafbde68bd9d 0x00010001 #UUID
1902 0xc04992988b1365d3 0x00010001 #UUID
1904 100000 0x00010001             #off_to_idle
1905 250000 0x00010001             #idle_to_ready
1906 0x1001 0x00010001             #SPI_bus
1907 1000 0x00010001               #reg_group_id
1908 0 0x00010001                  #cal_grp_id
1909 117 0x00010001             #gpio1
1910 118 0x00010001             #gpio2
1911 0 0x00010001                  #sensor_id
1912 0 0x00010001                  #CS for SPI
1913 1 0x00010001                  #data_type1
1914 4 0x00010001                  #data_type2
1915 0xFF 0x00010001               #rel_sns_idx
1916 0 0x00010001                  #sens_default
1917 0 0x00010001                  #flags
1982 0 0x00010001                  #device_select
1987 0x93 0x00010001               #vdd
1988 0x2 0x00010001                #vddio
```



```
# SSI SMGR Cfg 1: LSM6DS3 Gyro POLL
1919 0x2c95aafbde68bd9d 0x00010001 #UUID
1918 0xc04992988b1365d3 0x00010001 #UUID
1920 100000 0x00010001             #off_to_idle
1921 250000 0x00010001             #idle_to_ready
1922 0x1001 0x00010001             #SPI_bus
1923 1010 0x00010001               #reg_group_id
1924 10 0x00010001                 #cal_grp_id
1925 117 0x00010001             #gpio1
1926 118 0x00010001             #gpio2
1927 10 0x00010001                 #sensor_id
1928 0 0x00010001                  #CS for SPI
1929 3 0x00010001                  #data_type1
1930 4 0x00010001                  #data_type2
1931 0xFF 0x00010001               #rel_sns_idx
1932 0 0x00010001                  #sens_default
1933 0 0x00010001                  #flags
1983 0 0x00010001                  #device_select
1989 0x93 0x00010001               #vdd
1990 0x2 0x00010001                #vddio
```

软件架构-ST DDF驱动

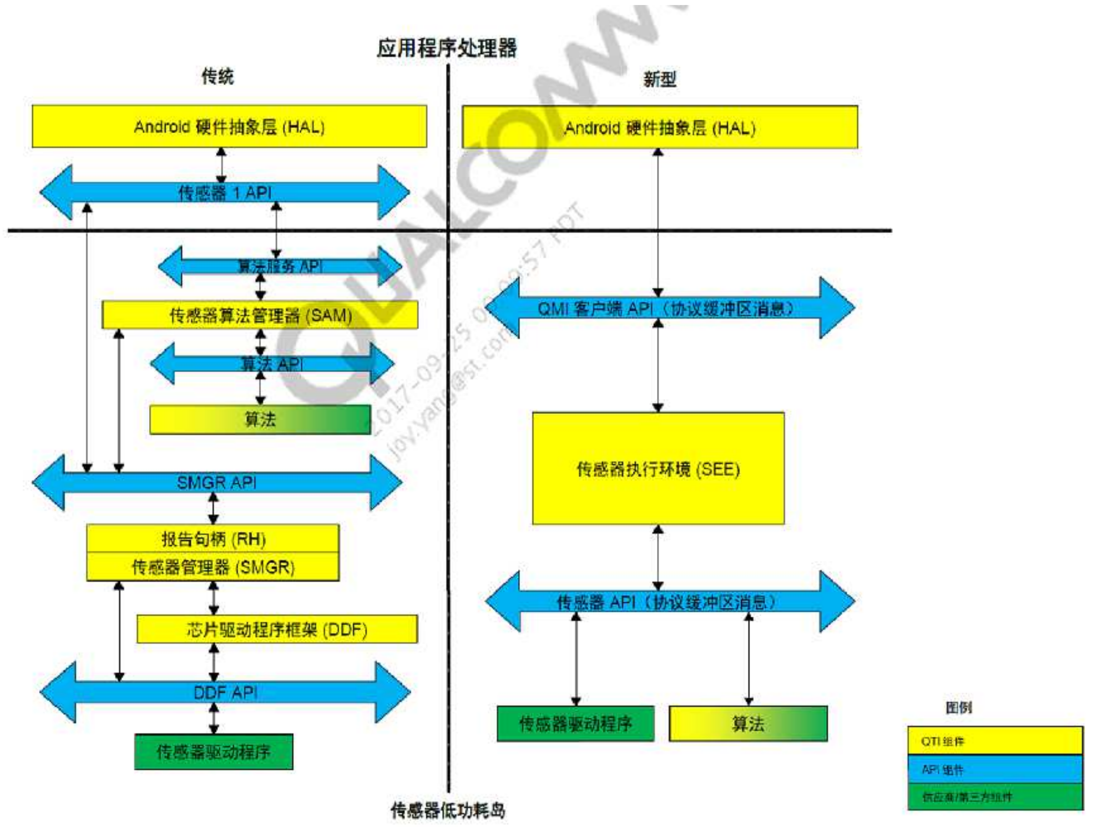
- ST目前支持的DDF驱动及版本信息列表如下：

	p/n	latest version
DDF	LIS2HH12	V3.8.0
	LIS2DS12	V1.2.0
	LIS3DH	V3.6.0
	LSM6DS3	V6.7.0
	LSM6DS3TR-C	V5.22.19
	LSM6DSL	V5.22.19
	LSM6DSM	V5.22.19
	LIS2MDL	V1.2.0
	LIS3MDL	V1.1.0
	LPS25H	V1.4.0
	LPS22HB	V1.6.0
	HTS221	V1.2.0

life.augmented

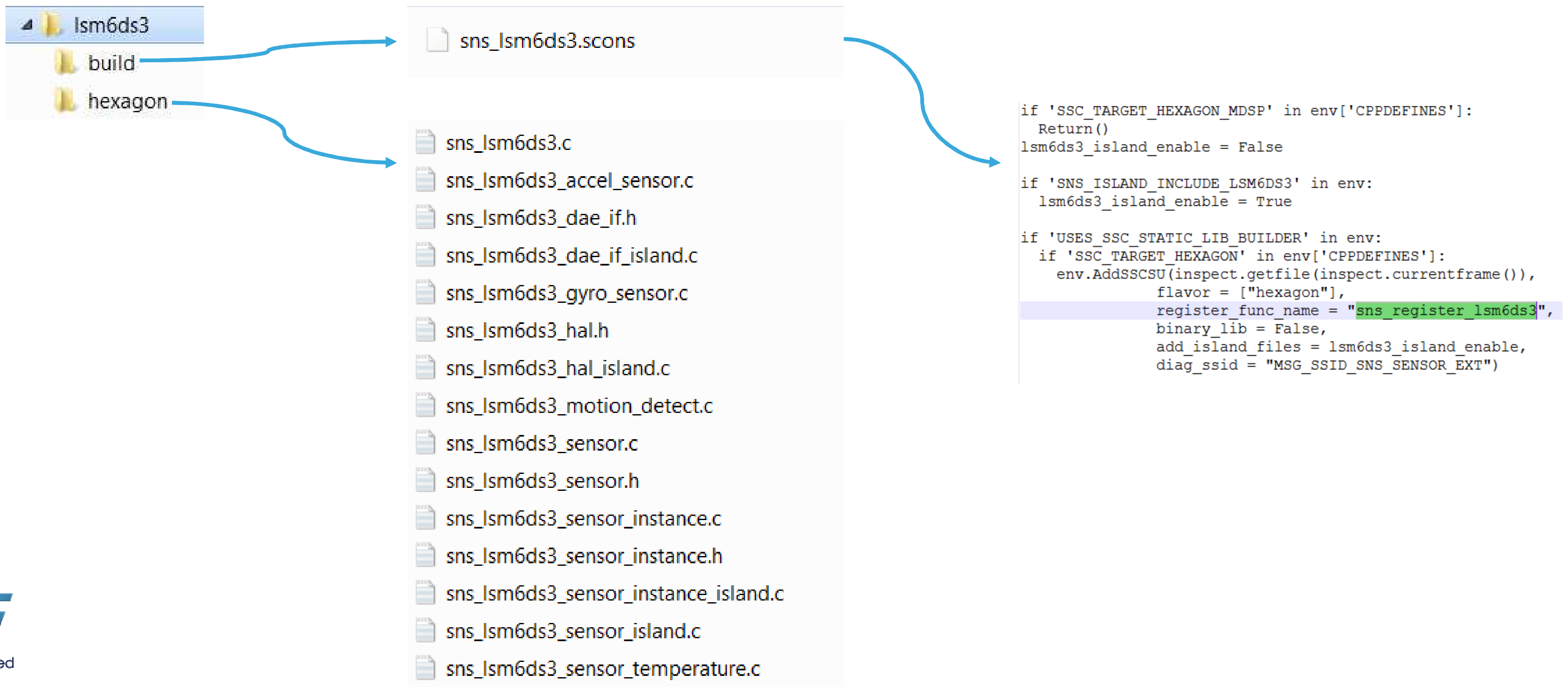
软件架构-SEE简介

- SEE（Sensors Execution Environment）是新一代的传感器管理软件架构，适用于2017年9月起发布的芯片组，



软件架构-SEE驱动移植

- 在<adsp_proc/slpi_proc>\slpi_proc\ssc\sensors中新建驱动文件夹



sns_lsm6ds3.scons

sns_lsm6ds3.c
sns_lsm6ds3_accel_sensor.c
sns_lsm6ds3_dae_if.h
sns_lsm6ds3_dae_if_island.c
sns_lsm6ds3_gyro_sensor.c
sns_lsm6ds3_hal.h
sns_lsm6ds3_hal_island.c
sns_lsm6ds3_motion_detect.c
sns_lsm6ds3_sensor.c
sns_lsm6ds3_sensor.h
sns_lsm6ds3_sensor_instance.c
sns_lsm6ds3_sensor_instance.h
sns_lsm6ds3_sensor_instance_island.c
sns_lsm6ds3_sensor_island.c
sns_lsm6ds3_sensor_temperature.c

```
if 'SSC_TARGET_HEXAGON_MDSP' in env['CPPDEFINES']:
  Return()
lsm6ds3_island_enable = False

if 'SNS_ISLAND_INCLUDE_LSM6DS3' in env:
  lsm6ds3_island_enable = True

if 'USES_SSC_STATIC_LIB_BUILDER' in env:
  if 'SSC_TARGET_HEXAGON' in env['CPPDEFINES']:
    env.AddSSCSU(inspect.getfile(inspect.currentframe()),
               flavor = ["hexagon"],
               register_func_name = "sns_register_lsm6ds3",
               binary_lib = False,
               add_island_files = lsm6ds3_island_enable,
               diag_ssid = "MSG_SSID_SNS_SENSOR_EXT")
```

软件架构-SEE驱动移植

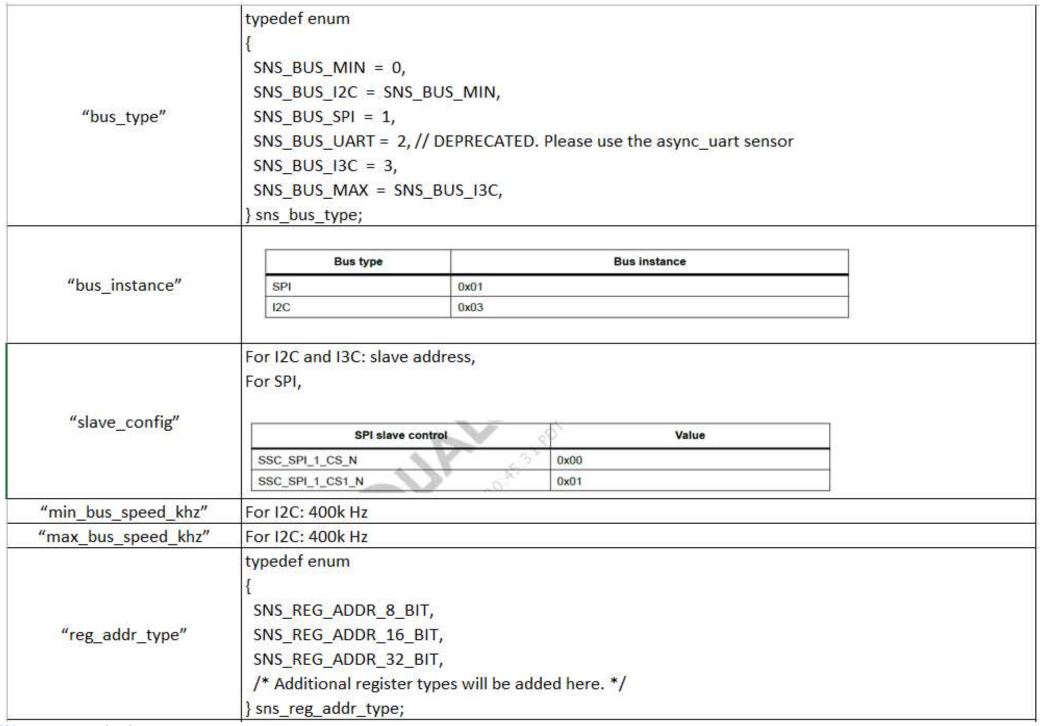
- 在SEE框架中注册新驱动

```
<adsp_proc/slpi_proc>\slpi_proc\ssc\framework\src\sns_static_sensors.c
const sns_register_entry sns_register_sensor_list[] =
{
   …
   { sns_register_lsm6ds3, 1},
   …
}
```

- 配置文件
  - 路径：/persist/sensors/registry/config
  - 格式：JSON
  - 每个物理传感器包含对应的平台特定和驱动程序特定两个配置文件，

软件架构-SEE驱动移植

- 平台配置文件：/persist/sensors/registry/config/lsm6ds3/msm8996_lsm6ds3_0.json



“bus_type”	typedef enum
	{
	SNS_BUS_MIN = 0,
	SNS_BUS_I2C = SNS_BUS_MIN,
	SNS_BUS_SPI = 1,
	SNS_BUS_UART = 2, // DEPRECATED. Please use the async_uart sensor
	SNS_BUS_I3C = 3,
	SNS_BUS_MAX = SNS_BUS_I3C,
	} sns_bus_type;

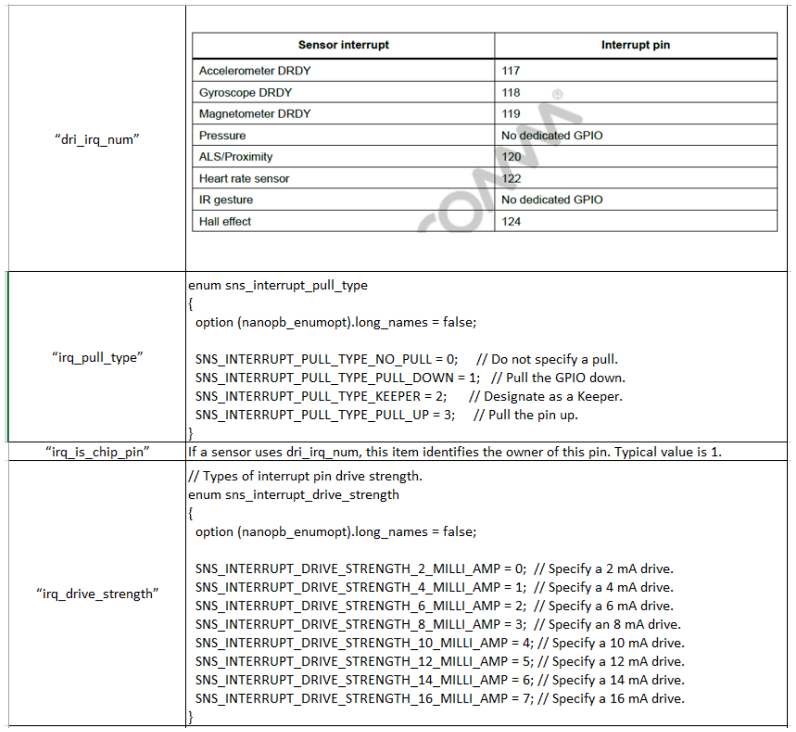
“bus_instance”	

Bus type	Bus instance
SPI	0x01
I2C	0x03

“slave_config”	For I2C and I3C: slave address, For SPI,

SPI slave control	Value
SSC_SPI_1_CS_N	0x00
SSC_SPI_1_CS1_N	0x01

“min_bus_speed_khz”	For I2C: 400k Hz
“max_bus_speed_khz”	For I2C: 400k Hz
“reg_addr_type”	typedef enum { SNS_REG_ADDR_8_BIT, SNS_REG_ADDR_16_BIT, SNS_REG_ADDR_32_BIT, /* Additional register types will be added here. */ } sns_reg_addr_type;



“dri_irq_num”	

Sensor interrupt	Interrupt pin
Accelerometer DRDY	117
Gyroscope DRDY	118
Magnetometer DRDY	119
Pressure	No dedicated GPIO
ALS/Proximity	120
Heart rate sensor	122
IR gesture	No dedicated GPIO
Hall effect	124

“irq_pull_type”	enum sns_interrupt_pull_type
	{
	option (nanopb_enumopt).long_names = false;
	
	SNS_INTERRUPT_PULL_TYPE_NO_PULL = 0; // Do not specify a pull.
	SNS_INTERRUPT_PULL_TYPE_PULL_DOWN = 1; // Pull the GPIO down.
	SNS_INTERRUPT_PULL_TYPE_KEEPER = 2; // Designate as a Keeper.
	SNS_INTERRUPT_PULL_TYPE_PULL_UP = 3; // Pull the pin up.
	}

“irq_is_chip_pin”	If a sensor uses dri_irq_num, this item identifies the owner of this pin. Typical value is 1.

“irq_drive_strength”	// Types of interrupt pin drive strength.
	enum sns_interrupt_drive_strength
	{
	option (nanopb_enumopt).long_names = false;
	
	SNS_INTERRUPT_DRIVE_STRENGTH_2_MILLI_AMP = 0; // Specify a 2 mA drive.
	SNS_INTERRUPT_DRIVE_STRENGTH_4_MILLI_AMP = 1; // Specify a 4 mA drive.
	SNS_INTERRUPT_DRIVE_STRENGTH_6_MILLI_AMP = 2; // Specify a 6 mA drive.
	SNS_INTERRUPT_DRIVE_STRENGTH_8_MILLI_AMP = 3; // Specify an 8 mA drive.
	SNS_INTERRUPT_DRIVE_STRENGTH_10_MILLI_AMP = 4; // Specify a 10 mA drive.
	SNS_INTERRUPT_DRIVE_STRENGTH_12_MILLI_AMP = 5; // Specify a 12 mA drive.
	SNS_INTERRUPT_DRIVE_STRENGTH_14_MILLI_AMP = 6; // Specify a 14 mA drive.
	SNS_INTERRUPT_DRIVE_STRENGTH_16_MILLI_AMP = 7; // Specify a 16 mA drive.
	}

软件架构-SEE驱动移植

- 驱动配置文件：/persist/sensors/registry/config/lsm6ds3/lsm6ds3_0.json

“is_dri”	This item identifies whether the sensor stream is interrupt based (data ready, watermark, motion, etc. interrupt) or polling. •DRI sensors use value 1. •Polling sensors use value 0.
“hw_id”	This item is the unique identifier for the sensor hardware. It is typically used to differentiate between multiple sensors of the same hardware.
“res_idx”	Physical sensors typically have multiple supported resolutions (and corresponding ranges). Sensors publish an array of supported resolutions. This item identifies the default resolution used by the sensor and is the index into the array of supported resolutions.
“sync_stream”	This item identifies whether the sensor supports any synchronous streaming mode such as S4S, I3C, etc.

```
{
  "config":
  {
    "hw_platform": ["MTP", "Dragon", "Surf", "HDK"],
    "soc_id": ["291", "246", "305", "341"]
  },
  "lsm6ds3_0":{
    "owner": "sns_lsm6ds3",
    ".accel":{
      "owner": "sns_lsm6ds3",
      ".config":{
        "owner": "sns_lsm6ds3",
        "is_dri":{ "type": "int", "ver": "0",
          "data": "1"
        },
        "hw_id":{ "type": "int", "ver": "0",
          "data": "0"
        },
        "res_idx":{ "type": "int", "ver": "0",
          "data": "2"
        },
        "sync_stream":{ "type": "int", "ver": "0",
          "data": "0"
        }
      }
    },
```

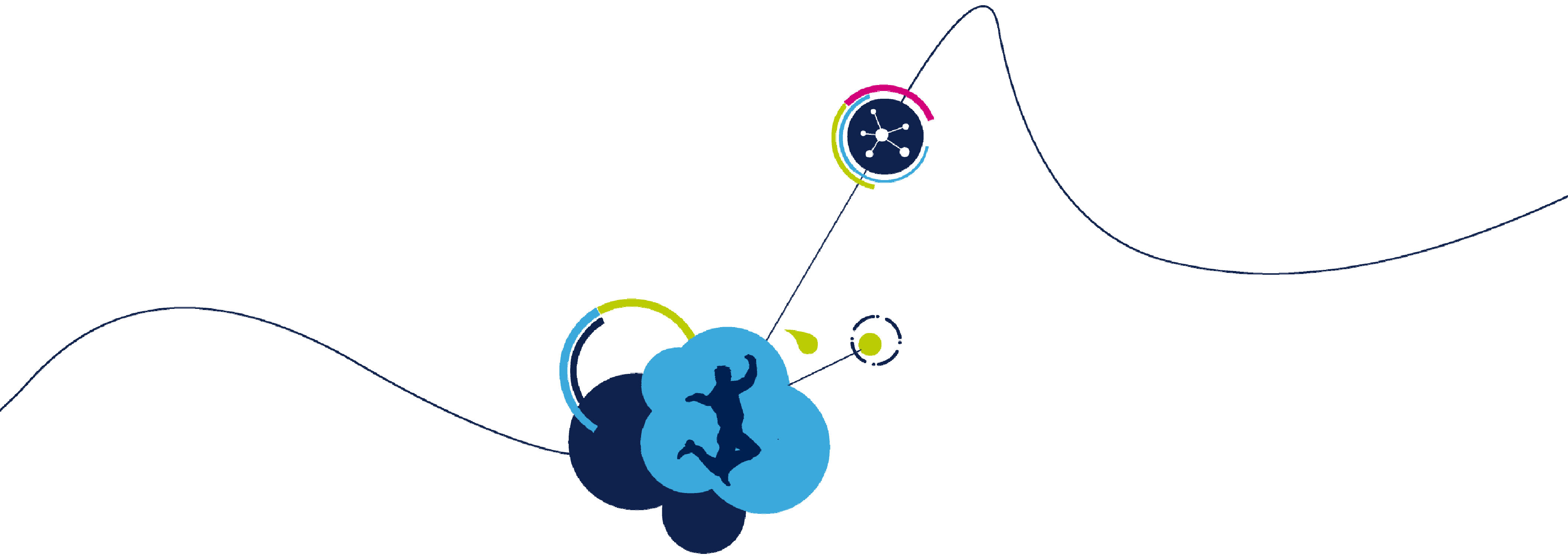
life.augmented

软件架构-ST SEE驱动

- ST目前支持的SEE驱动及版本信息列表如下：

	p/n	latest version
SEE	LSM6DS3TR-C	V1.58.01
	LSM6DSM	V1.59.0
	LSM6DSO	V1.13.0
	L3GD20H	V1.0.6
	LIS2MDL	V1.4.0
	LPS22HH	V1.0
	LPS22HB	V1.12.0
	LPS22HX	V2.1.0

life.augmented



谢谢！

life.augmented