Unified API

include/zephyr/sensing/sensing_sensor.h

struct sensing_sensor_api {

sensing_sensor_init_t init;

~~sensing_sensor_reset_t reset;~~

~~sensing_sensor_deinit_t deinit;~~

sensing_sensor_set_interval_t set_interval;
sensing_sensor_set_range_t set_range;
sensing_sensor_set_offset_t set_offset;
sensing_sensor_set_fifo_t set_fifo;
sensing_sensor_set_sensitivity_t set_sensitivity;
sensing_sensor_self_calibration_t self_calibration;

sensing_sensor_get_interval_t get_interval;
sensing_sensor_get_range_t get_range;
sensing_sensor_get_offset_t get_offset;
sensing_sensor_get_fifo_t get_fifo;
sensing_sensor_get_sensitivity_t get_sensitivity;

sensing_sensor_read_sample_t read_sample;

sensing_sensor_sensitivity_test_t sensitivity_test;

sensing_sensor_process_t process;

};

include/zephyr/device.h

DEVICE_DEFINE(dev_id, name, init_fn, ...)

include/zephyr/drivers/sensor.h

__subsystem struct sensor_driver_api {

sensor_attr_set_t attr_set;

sensor_attr_get_t attr_get;

sensor_trigger_set_t trigger_set;

sensor_sample_fetch_t sample_fetch;

sensor_channel_get_t channel_get;

sensor_get_decoder_t get_decoder;

sensor_submit_t submit;

+ sensor_sensitivity_test_t sensitivity_test;

};

include/zephyr/sensing/sensing.h

struct sensing_callback_list {

sensing_data_event_t on_data_event;

};

Struct Relationship

subsys/sensing/sensor/phy_3d_sensor/phy_3d_sensor.c

```
struct sensor_driver_api phy_3d_sensor_api = {
    .attr_set = phy_3d_sensor_attr_set,
    .submit = phy_3d_sensor_submit,
};

DEVICE_DT_DEFINE(node, phy_3d_sensor_init, ...
                 &phy_3d_sensor_api);

SENSING_SENSOR_DT_DEFINE(node, NULL, ...);
```

subsys/sensing/sensor/hinge_angle/hinge_angle.c

```
struct sensor_driver_api hinge_angle_api = {
    .attr_set = hinge_angle_attr_set,
    .submit = hinge_angle_submit,
};

DEVICE_DT_DEFINE(node, hinge_angle_init, ...
                 &hinge_angle_api);

struct sensing_callback_list cb_##node = {
    .on_data_event = hinge_reporter_on_data_event,
};

SENSING_SENSOR_DT_DEFINE(node, &cb_##node, ...);
```

include/zephyr/sensing/sensing_sensor.h

```
SENSING_SENSOR_DT_DEFINE(node, cb_ptr, ...)

        struct sensing_sensor_info __info_##node;

        struct sensing_connection __conns_##node[] = {
                { . callback_list = cb_ptr, .source = xxx },
                { . callback_list = cb_ptr, .source = xxx },
                ...
        };

        struct rtio_iodev __iodev_##node;

        STRUCT_SECTION_ITERABLE(sensing_sensor,
                                __sensing_sensor_##node) = {
                .dev = DEVICE_DT_GET(node);
                .info = &__info_##node;
                .conns = &__conns_##node;
                .iodev = &__iodev_##node;
        };
```

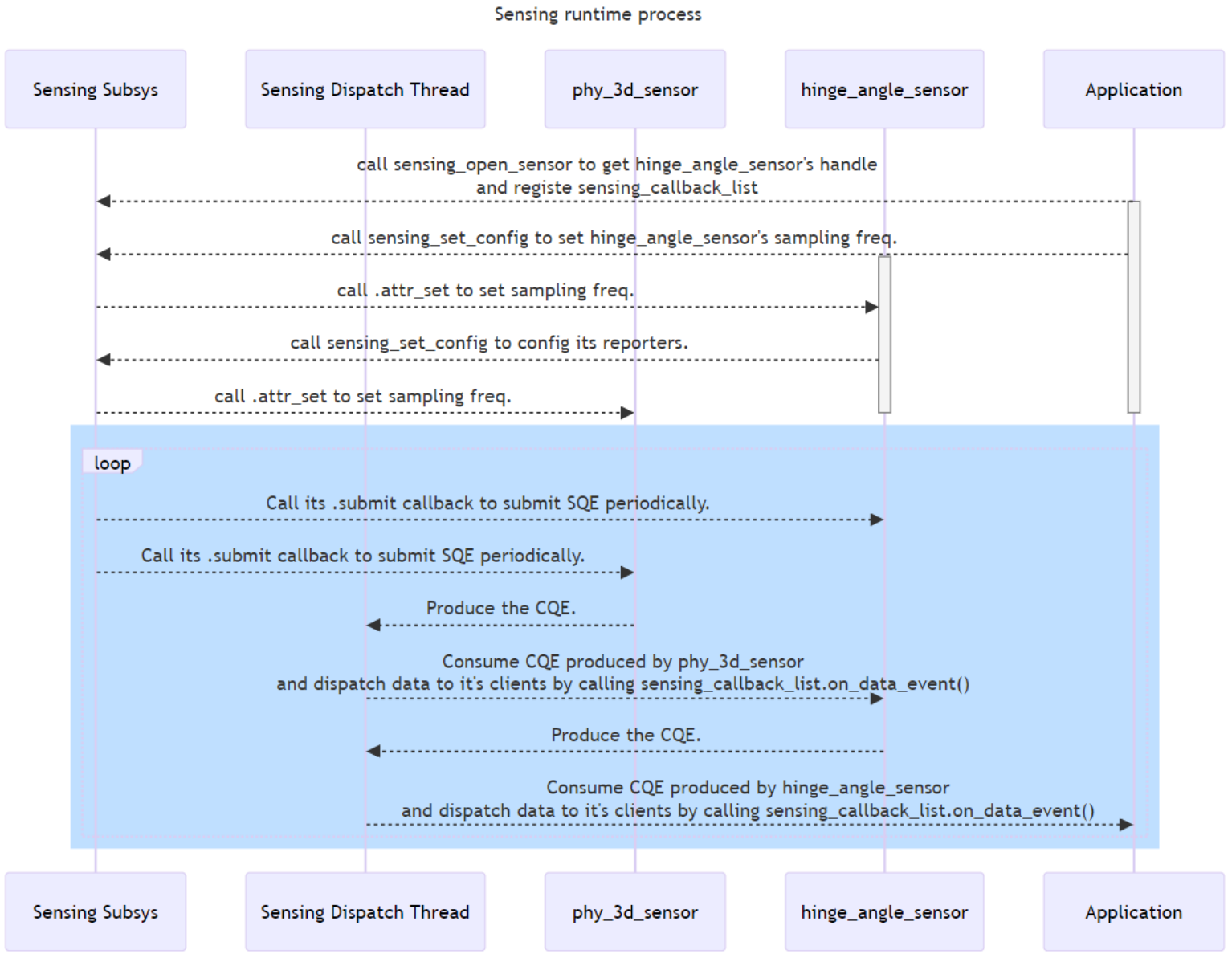
```
STRUCT_SECTION_START(sensing_sensor);
    sensing_sensor_phy_3d_sensor_node;
    sensing_sensor_hinge_angle_node;
  ...
STRUCT_SECTION_END(sensing_sensor);
```

Build Sensor Tree

1. The ***conns*** field of *sensing_sensor* and the ***source*** field of *sensing_connection* were assigned basing on the DTS configuration at compile time.
   The DTS parser also sorted the sensors by device's dependence relationship.

2. In sensing sensor's init function, the sensor which has the reporter sensors can get the handles for its reporters for future use by calling sensing_sensor_get_reporters().

Sensing runtime process



Demo code

Link:

https://github.com/zephyrproject-rtos/zephyr/pull/64478

Build and run on native_posix with BMI160 emulator

CMD: west build -b native_posix samples/subsys/sensing/simple/ -t run