

# subscription Input signal grouping

SFC는 제어기가 전달하는 메세지 단위에 속해있는 CAN signal 각각을 추상화 된 signal로 수신하고, yaml에 정의된 subscription에서 Input signal group 단위(data chunk)로 일괄 수신하여 처리

```
- tireInfoChanged:
  inputType: VALUE_CHANGED
  duration:
  values:
    - Vehicle.CV.TPMS_CV.Input_TPMSWheelLocation: mInput_TPMSWheelLocation
    - Vehicle.CV.TPMS_CV.Input_LowTirePressureStatus: mInput_LowTirePressureStatus
    - Vehicle.CV.TPMS_CV.Input_HighTireTemperatureStatus: mInput_HighTireTemperatureStatus
    - Vehicle.CV.TPMS_CV.Input_LowTPMSSensorBatteryStatus: mInput_LowTPMSSensorBatteryStatus
    - Vehicle.CV.TPMS_CV.Input_TirePressureValue: mInput_TirePressureValue
    - Vehicle.CV.TPMS_CV.Input_TireTemperatureValue: mInput_TireTemperatureValue
    - Vehicle.CV.TPMS_CV.Input_TPMSSensorBatterySOCValue: mInput_TPMSSensorBatterySOCValue
```

## 다수의 시그널을 하나의 callback으로 Grouping General Case

- case1)
  - 사양에서 다수의 시그널 입력으로 하나의 Output을 출력하는 경우
- case2)
  - 서로 다른 동작 사양에서 동일한 CAN Message Group의 연관된 신호들이, 입력 순서에 따라 문제가 발생할 경우

Example)

상위 tireInfoChanged 콜백에서 수신받는 signal들의 경우 타이어 location의 모든 정보(공기압, 온도 등등)는 CAN Msg(frame)단위로 특정 주기마다 전달

SFC는 전달 받은 타이어에 대한 정보를 CAN Msg에 속해있는 각각의 CAN Signal에 매칭된 추상화 시그널로 변환하여 추상화 시그널 그룹(data chunk)으로 수신

*tireInfoChanged 함수에서 subscription하고 있는 7개의 차량 시그널은 하나의 CAN Msg로 구성*

## <문제 발생 가능 case>

case) **Input\_TPMSWheelLocation** 과 **Input\_TirePressureValue** 의 callback이 분리된 경우

```
- tireInfoChanged:
  inputType: VALUE_CHANGED
  duration:
  values:
    - Vehicle.CV.TPMS_CV.Input_LowTirePressureStatus: mInput_LowTirePressureStatus
    - Vehicle.CV.TPMS_CV.Input_HighTireTemperatureStatus: mInput_HighTireTemperatureStatus
    - Vehicle.CV.TPMS_CV.Input_LowTPMSSensorBatteryStatus: mInput_LowTPMSSensorBatteryStatus
    - Vehicle.CV.TPMS_CV.Input_TirePressureValue: mInput_TirePressureValue
    - Vehicle.CV.TPMS_CV.Input_TireTemperatureValue: mInput_TireTemperatureValue
    - Vehicle.CV.TPMS_CV.Input_TPMSSensorBatterySOCValue: mInput_TPMSSensorBatterySOCValue
- wheelLocationChanged:
  inputType: VALUE_CHANGED
  duration:
  values:
    - Vehicle.CV.TPMS_CV.Input_TPMSWheelLocation: mInput_TPMSWheelLocation
```

subscription을 위와 같이 정의한 경우,

동일 CAN Msg 그룹의 signals들이 전송되면, 하기의 SFCPluginImple과 같이 한 transaction에서 일괄 수신 됨 (하기 이미지의 빨간 박스 참조)

```

void onSubscription(const ClientContextRef& clientContext,
                  const SubscriptionRef& subscription,
                  const SubscriptionCacheMap& cachedData) override {
    switch (subscription->getSubscriptionId()) {
        case 0: {
            SFCArgument arg0 = valueFromHashMap(cachedData, "SFC.Private.IGNElapsed.ElapsedOn0ms").value_or(std::make_tuple(nullptr, std::make_shared<SignalValue>())); // NOLINT
            ccoss::TPMS_CV::ArgumentsIgnElapseOn0msChanged arg(subscription, arg0);
            mImpl.onIgnElapseOn0msChanged(arg);
            break;
        }
        case 1: {
            SFCArgument arg0 = valueFromHashMap(cachedData, "SFC.Private.IGNElapsed.ElapsedOn500ms").value_or(std::make_tuple(nullptr, std::make_shared<SignalValue>())); // NOLINT
            ccoss::TPMS_CV::ArgumentsIgnElapseOn500msChanged arg(subscription, arg0);
            mImpl.onIgnElapseOn500msChanged(arg);
            break;
        }
        case 7: {
            SFCArgument arg0 = valueFromHashMap(cachedData, "Vehicle.CV.TPMS.CV.Input_TPMSWarningStatus").value_or(std::make_tuple(nullptr, std::make_shared<SignalValue>())); // NOLINT
            SFCArgument arg1 = valueFromHashMap(cachedData, "Vehicle.CV.TPMS.CV.Input_TPMSLowPressWarnStatus").value_or(std::make_tuple(nullptr, std::make_shared<SignalValue>())); // NOLINT
            SFCArgument arg2 = valueFromHashMap(cachedData, "Vehicle.CV.TPMS.CV.Input_TPMSHighTempWarnStatus").value_or(std::make_tuple(nullptr, std::make_shared<SignalValue>())); // NOLINT
            ccoss::TPMS_CV::ArgumentsTellTaleStatusChanged arg(subscription, arg0, arg1, arg2);
            mImpl.onTellTaleStatusChanged(arg);
            break;
        }
        case 8: {
            SFCArgument arg0 = valueFromHashMap(cachedData, "Vehicle.CV.TPMS.CV.Input_LowTirePressureStatus").value_or(std::make_tuple(nullptr, std::make_shared<SignalValue>())); // NOLINT
            SFCArgument arg1 = valueFromHashMap(cachedData, "Vehicle.CV.TPMS.CV.Input_HighTireTemperatureStatus").value_or(std::make_tuple(nullptr, std::make_shared<SignalValue>())); // NOLINT
            SFCArgument arg2 = valueFromHashMap(cachedData, "Vehicle.CV.TPMS.CV.Input_LowTPMSSensorBatteryStatus").value_or(std::make_tuple(nullptr, std::make_shared<SignalValue>())); // NOLINT
            SFCArgument arg3 = valueFromHashMap(cachedData, "Vehicle.CV.TPMS.CV.Input_TirePressureValue").value_or(std::make_tuple(nullptr, std::make_shared<SignalValue>())); // NOLINT
            SFCArgument arg4 = valueFromHashMap(cachedData, "Vehicle.CV.TPMS.CV.Input_TireTemperatureValue").value_or(std::make_tuple(nullptr, std::make_shared<SignalValue>())); // NOLINT
            SFCArgument arg5 = valueFromHashMap(cachedData, "Vehicle.CV.TPMS.CV.Input_TPMSSensorBatterySOCValue").value_or(std::make_tuple(nullptr, std::make_shared<SignalValue>())); // NOLINT
            ccoss::TPMS_CV::ArgumentsTireInfoChanged arg(subscription, arg0, arg1, arg2, arg3, arg4, arg5);
            mImpl.onTireInfoChanged(arg);
            break;
        }
        case 9: {
            SFCArgument arg0 = valueFromHashMap(cachedData, "Vehicle.CV.TPMS.CV.Input_TPMSWheelLocation").value_or(std::make_tuple(nullptr, std::make_shared<SignalValue>())); // NOLINT
            ccoss::TPMS_CV::ArgumentsWheelLocationChanged arg(subscription, arg0);
            mImpl.onWheelLocationChanged(arg);
            break;
        }
        default:
            break;
    }
    mImpl.enqueue();
}

```

```

void onTireInfoChanged(const ArgumentsTireInfoChanged& args) {
    mLowTirePressureStatus = args.mInput_LowTirePressureStatus;
    mHighTireTemperatureStatus = args.mInput_HighTireTemperatureStatus;
    mLowTPMSSensorBatteryStatus = args.mInput_LowTPMSSensorBatteryStatus;
    mTirePressureValue = args.mInput_TirePressureValue;
    mTireTemperatureValue = args.mInput_TireTemperatureValue;
    mTPMSSensorBatterySOCValue = args.mInput_TPMSSensorBatterySOCValue;
    updateConstantTireInfo();
}

void onWheelLocationChanged(const ArgumentsWheelLocationChanged& args) {
    mTPMSWheelLocation = args.mInput_TPMSWheelLocation;
    updateConstantWheelLocationInfo();
}

```

## <정상 동작 case>

(MCU로 부터 CAN Data 수신 순서가 다음과 같을 때 : Input\_TirePressureValue → Input\_TPMSWheelLocation)

- Input\_TirePressureValue 에 해당하는 callback이 먼저 호출, Input\_TPMSWheelLocation callback이 다음에 호출

즉, MCU가 전달하는 Vehicle Signal 순서인 `tireInfoChanged` > `wheelLocationChanged` 순으로 callback 호출 및 로직 수행이 이루어 지게 됩니다.

	First CAN Msg	Second CAN Msg
Input_TirePressureValue	50	40
Input_TPMSWheelLocation	1	2
output	1번 Wheel Location / 공기압 50	빨간 박스 : 이전 1번 Wheel Location / 공기압 40 파란박스 : 이번 2번 Wheel Location / 공기압 40

Jira: EXNCP-32310

AltonService 1.85.0 이후 버전에서는 1번의 output Transaction에서 같은 노드의 output은 최신 값으로 갱신

1. First CAN Msg에 의해 정상적인 (1번 Wheel Location + 공기압 50)이 output으로 송신
2. Second CAN Msg에 의해, 빨간 박스 callback에 의해 (1번 Wheel Location + 공기압 40) 연산 및 setNode
3. Second CAN Msg에 의해, 파란 박스 callback에 의해 (2번 Wheel Location + 공기압 40) 이 최종 output으로 송신

```
Signal Received ----- (2023-09-26 19:26:36.854)
Received : SFC.TPMS_CV.Constant.TPMSWheelLocation.Stat = 1 (U)
Received : SFC.TPMS_CV.Constant.TirePressure4_OR.Value = 50 (D)
```

```
Signal Received ----- (2023-09-26 19:26:57.725)
Received : SFC.TPMS_CV.Constant.TPMSWheelLocation.Stat = 2 (U)
Received : SFC.TPMS_CV.Constant.TirePressure4_OR.Value = 40 (D)
```

2번, 3번은 한번의 transaction으로 `onTireInfoChanged` 와 `onWheelLocationChanged` callback 동작 후 같은 노드의 경우 최신 값으로 update되어 output으로 전달되기 때문에, 나뉜 callback에 의해 불필요한 동작은 하나 이슈가 발생되지는 않음

### <이슈 발생 case>

```
void onTireInfoChanged(const ArgumentsTireInfoChanged& args) {
    mLowTirePressureStatus = args.mInput_LowTirePressureStatus;
    mHighTireTemperatureStatus = args.mInput_HighTireTemperatureStatus;
    mLowTPMSSensorBatteryStatus = args.mInput_LowTPMSSensorBatteryStatus;
    mTirePressureValue = args.mInput_TirePressureValue;
    mTireTemperatureValue = args.mInput_TireTemperatureValue;
    mTPMSSensorBatterySOCValue = args.mInput_TPMSsensorBatterySOCValue;
    updateConstantTireInfo();
    updateEvent();
}

void onWheelLocationChanged(const ArgumentsWheelLocationChanged& args) {
    mTPMSWheelLocation = args.mInput_TPMSWheelLocation;
    updateConstantWheelLocationInfo();
}
```

상위의 동작 루틴은 동일하고, `updateConstantTireInfo()` 이후 `updateEvent()` 를 하는 경우,

<정책>

Jira: EXNCP-12108

한번의 output transaction에서 Event Node가 있는 경우 하기 SFC Base code와 같이 `preFlushSignals()` function에 의해 이전 enqueue된 Constant의 노드를 먼저 전달

(Constant1/Constant2/Event1 → Constant1+Constant2 전달 → Event1 전달)

```

void enqueue() {
    if (isBlocked() == true) { // Blocked do not make any signals.
        return;
    }

    if (mAddressValueList.empty()) {
        clearCache();
        return;
    }
    std::set<SignalName> forcedUpdateSignalNameSet{};
    const std::set<std::string>& groupNameSet = mForceUpdateGroup ? mTargetGroupSet : mUpdatedGroupSet;
    mForceUpdateGroup = false;
    for (const auto& groupName : groupNameSet) {
        const auto& signalNameList = mGroupToSignalNameMap[groupName];
        for (const auto& signalName : signalNameList) {
            forcedUpdateSignalNameSet.emplace(signalName);
        }
    }

    preFlushSignals(mAddressValueList, forcedUpdateSignalNameSet);

    if (!mAddressValueList.empty()) {
        mSFCPlugIn->enqueueSignals(mAddressValueList, forcedUpdateSignalNameSet);
    }
    clearCache();
}

```

1. First CAN Msg에 의해 정상적인 (1번 Wheel Location + 공기압 50)이 output으로 송신
2. Second CAN Msg에 의해, 빨간 박스 callback에 의해 (1번 Wheel Location + 공기압 40) 연산 및 setNode
3. (녹색박스) updateEvent() 수행 시 Event Node enqueue 시 기존의 constant node preflush
4. 2번의 (1번 Wheel Location + 공기압 40)이 output으로 송신
5. Second CAN Msg에 의해, 파란 박스 callback에 의해 (2번 Wheel Location + 공기압 40) 이 최종 output으로 송신

1번	<p>Signal Received ----- (2023-09-27 11:36:50.903)</p> <p>Received : SFC.TPMS_CV.Constant.TirePressure4_OR.Value = 50 (D)</p> <p>Received : SFC.TPMS_CV.Constant.TPMSWheelLocation.Stat = 1 (U)</p>
4번	<p>Signal Received ----- (2023-09-27 11:36:56.111)</p> <p>Received : SFC.TPMS_CV.Constant.TPMSWheelLocation.Stat = 1 (U)</p> <p>Received : SFC.TPMS_CV.Constant.TirePressure4_OR.Value = 40 (D)</p>
5번	<p>Signal Received ----- (2023-09-27 11:36:56.111)</p> <p>Received : SFC.TPMS_CV.Constant.TirePressure4_OR.Value = 40 (D)</p> <p>Received : SFC.TPMS_CV.Constant.TPMSWheelLocation.Stat = 2 (U)</p>

4번의 output 출력으로 EXNCP-37862 이슈 발생한 이력이 있어, 관련 Input signal을 동일한 callback으로 group화 하여 수정