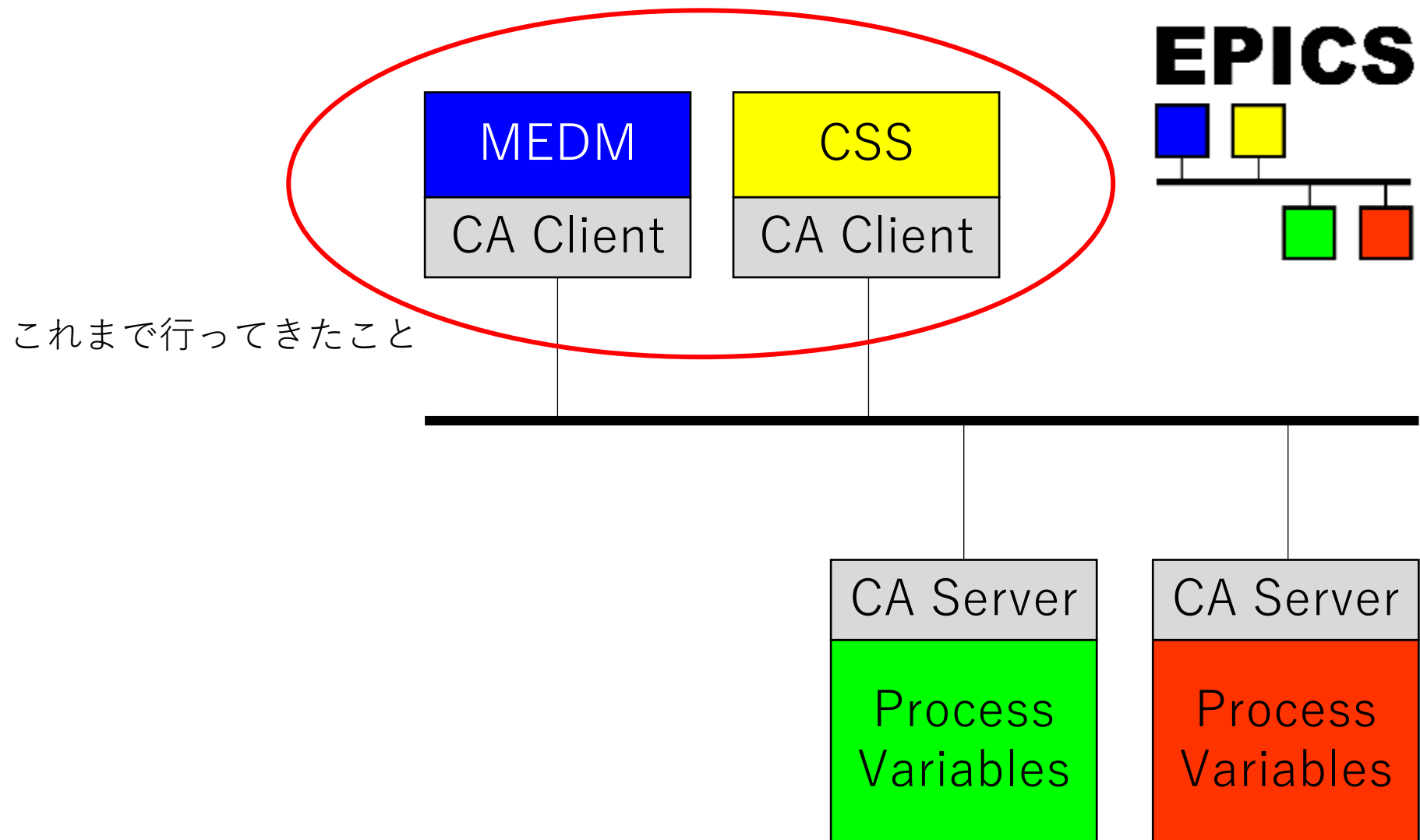
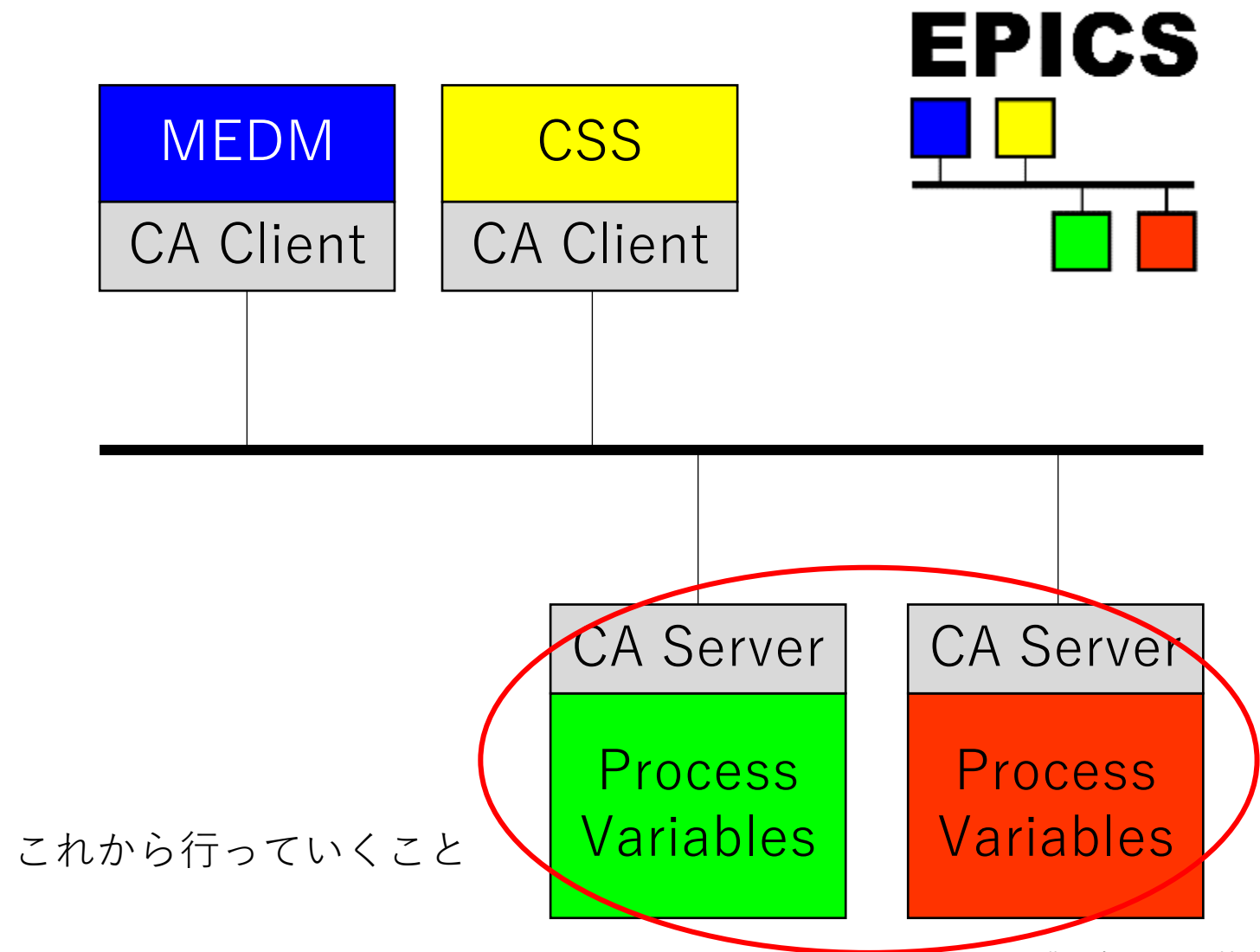


加速器の運転制御を通じて学ぶEPICS入門講習会： DBの基礎 (SoftI/O)

加速器研究施設 加速器第四研究系 SuperKEKB制御グループ
佐々木 信哉



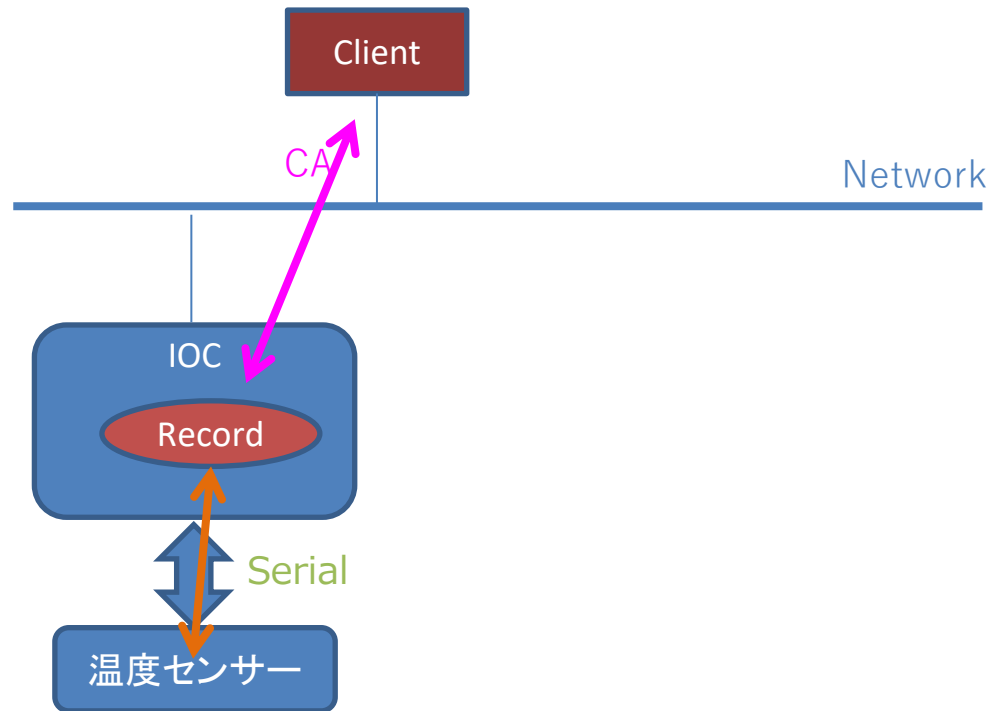


IOCとSoft IOC

- IOC (Input/Output Controller)

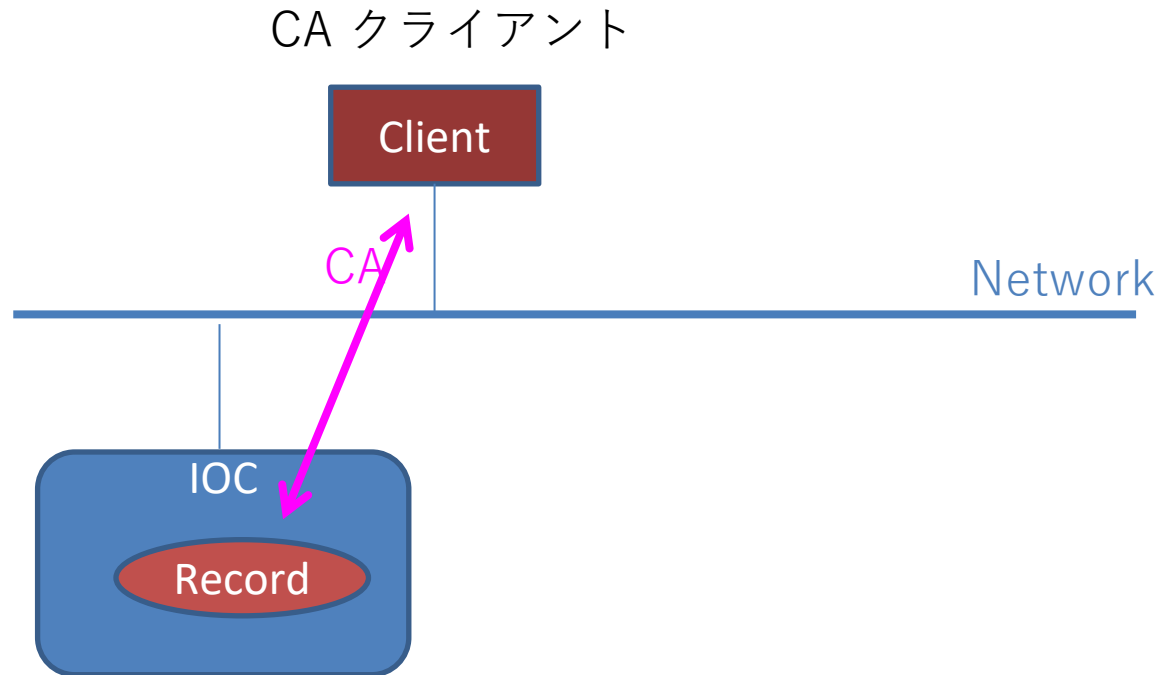
- 現実の機器の値とPVを結びつける役割
- IOCではPVをレコードという単位で管理 (PV≡レコード)

CA クライアント



IOCとSoft IOC

- Soft IOC (Input/Output Controller)
 - 現実の機器との接続がないIOC



レコードとは？

- IOCの中でPVを管理する機構・仕組み
 - 1レコード \doteq 1PV
- 定期的もしくはイベントなどによって何らかの動作（プロセス）を行う
- レコードの種類によって異なる動作が行われる
 - ハードウェアから現在の値を読み込む
 - ハードウェアに値を書き込む
 - 任意の計算を行う
 - …
- レコード同士を結び付ける（リンクする）ことができる
 - 値の受け渡し
 - プロセスの連動（Processing Chains）
- レコードのフィールド（メタデータ）によってレコードの動作を制御・変更できる

EPICS Databaseとは？

- IOC実行時にメモリ上で管理される**レコードの集合**および**管理機構**
- Databaseの定義（レコードの定義）はテキストファイルで行う：
 - **Databaseファイル**や**Record Instance File**と呼ぶ
- Databaseファイル（Record Instance File）の作成方法
 - 「myDatabase.db」のような**テキストファイルを直接編集**
 - VDCTやCapFastといった**ツールで作成**する

Databaseファイルにはどんなことを書く？

- Databaseファイルには以下のようなことを定義する

| どんなレコードが | いつ | どのような順序で | どのような動作 |
|----------|----|----------|---------|
| | | | |

Databaseファイルにはどんなことを書く？

- Databaseファイルには以下のようなことを定義する

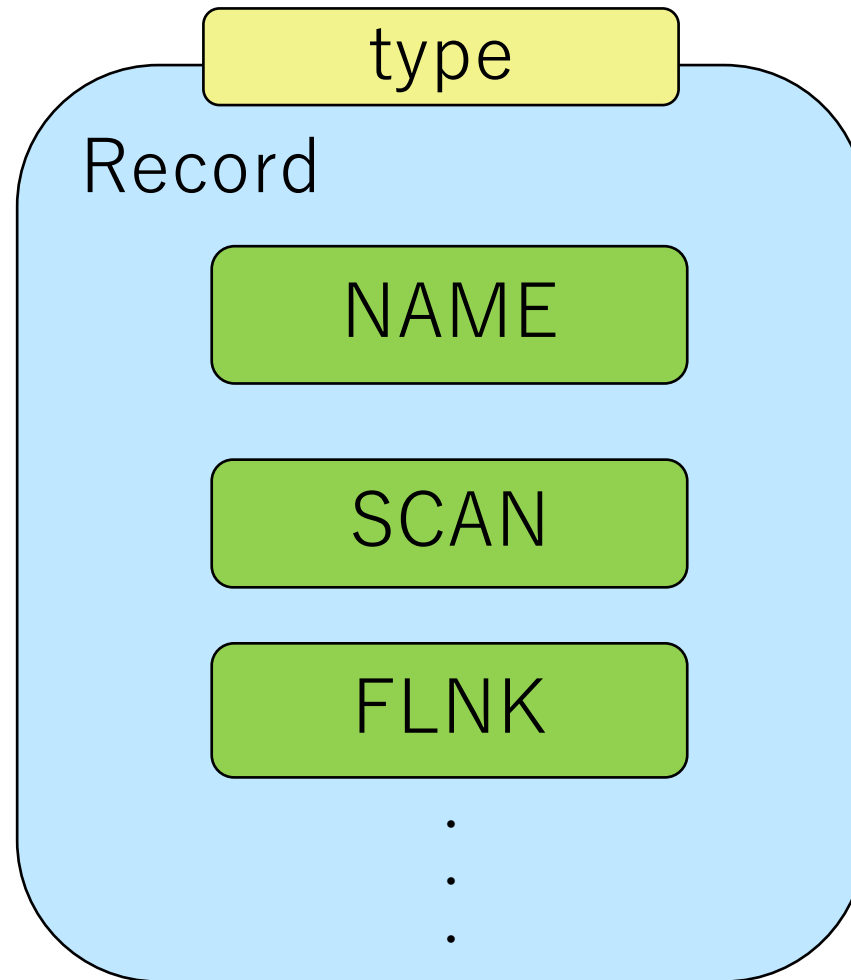
| どんなレコードが | いつ | どのような順序で | どのような動作 |
|----------------------------|------|------------------|--------------------|
| Record type Record name | SCAN | Processing Chain | 各Field: INP, OUTなど |

Databaseファイルにはどんなことを書く？

例えば：

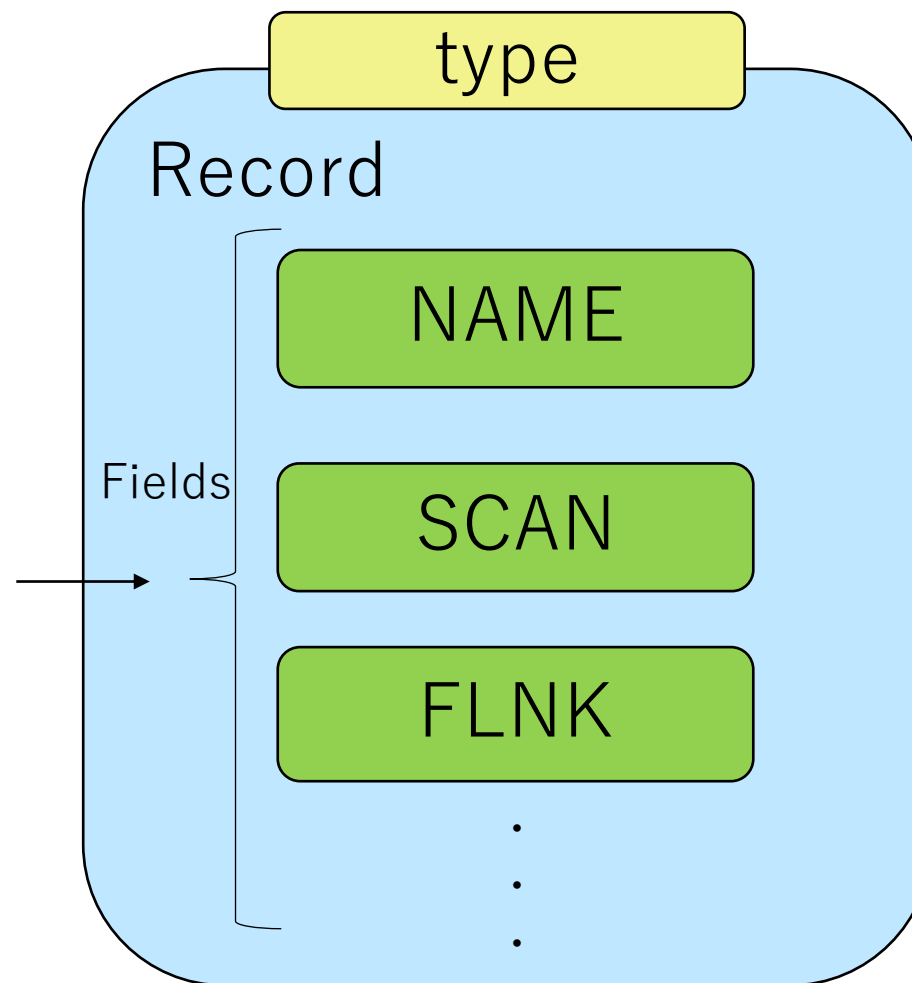
1秒ごとに2つの温度センサの平均を計算して10度より大きい時に機器の電源をONにしたい時

| どんなレコードが | いつ | どのような順序でどのような動作 |
|--|-----|--|
| 実数レコード (AI1、AI2) 計算レコード (CALC1) バイナリ出力レコード (BO1) | 1秒毎 | <ol style="list-style-type: none">1. AI1が温度センサの値を取得2. AI2が温度センサの値を取得3. CALC1がAI1とAI2の値を使い平均の温度を計算し、10度より大きい時1、以下の時0を出力4. BO1がCALC1の計算結果の値を使って機器のON/OFFを制御 |



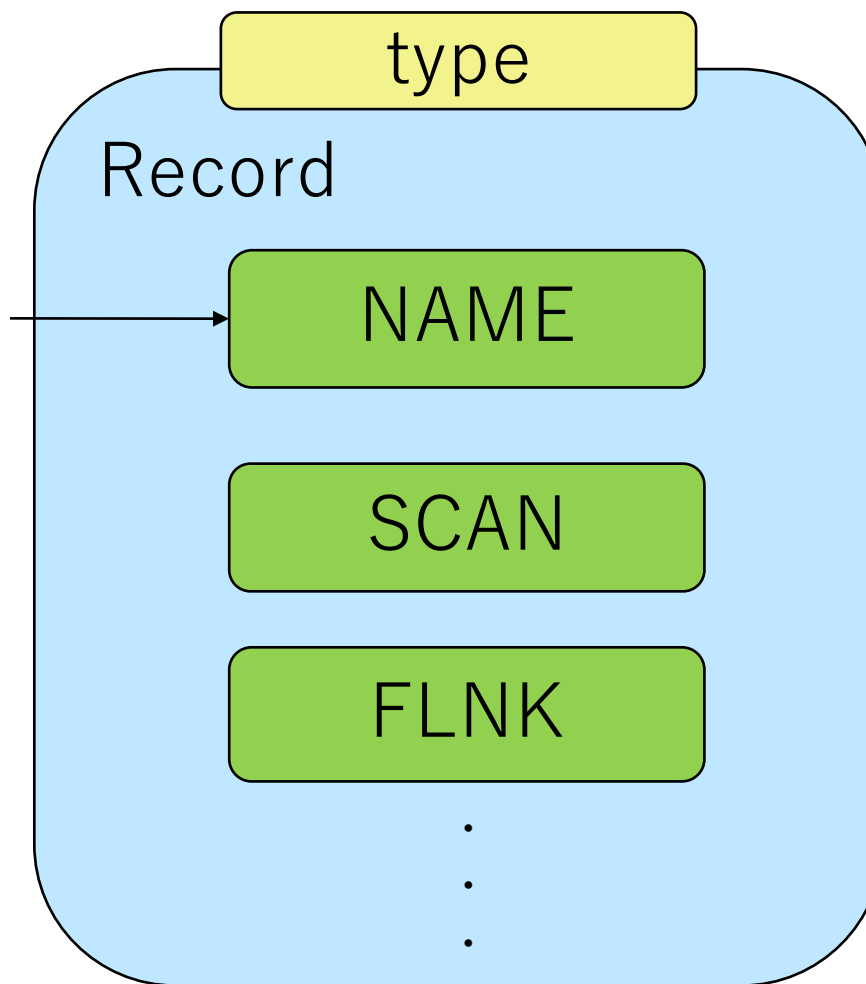
レコード

- フィールドによって各種設定を行う
- typeによってフィールドの種類が変わる



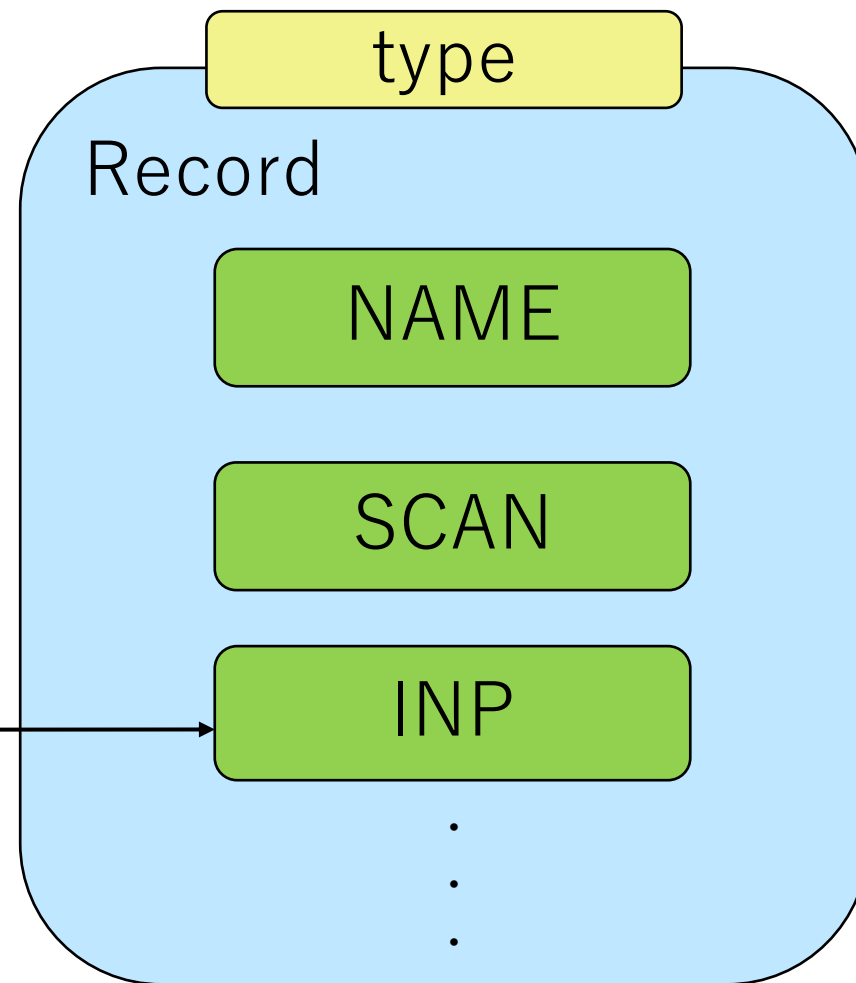
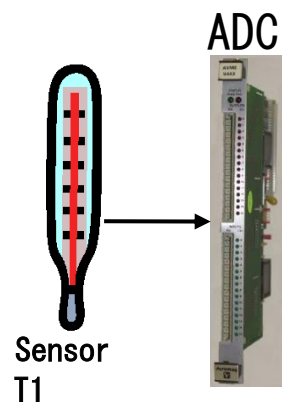
レコード

他と被らない任意の名前を付ける

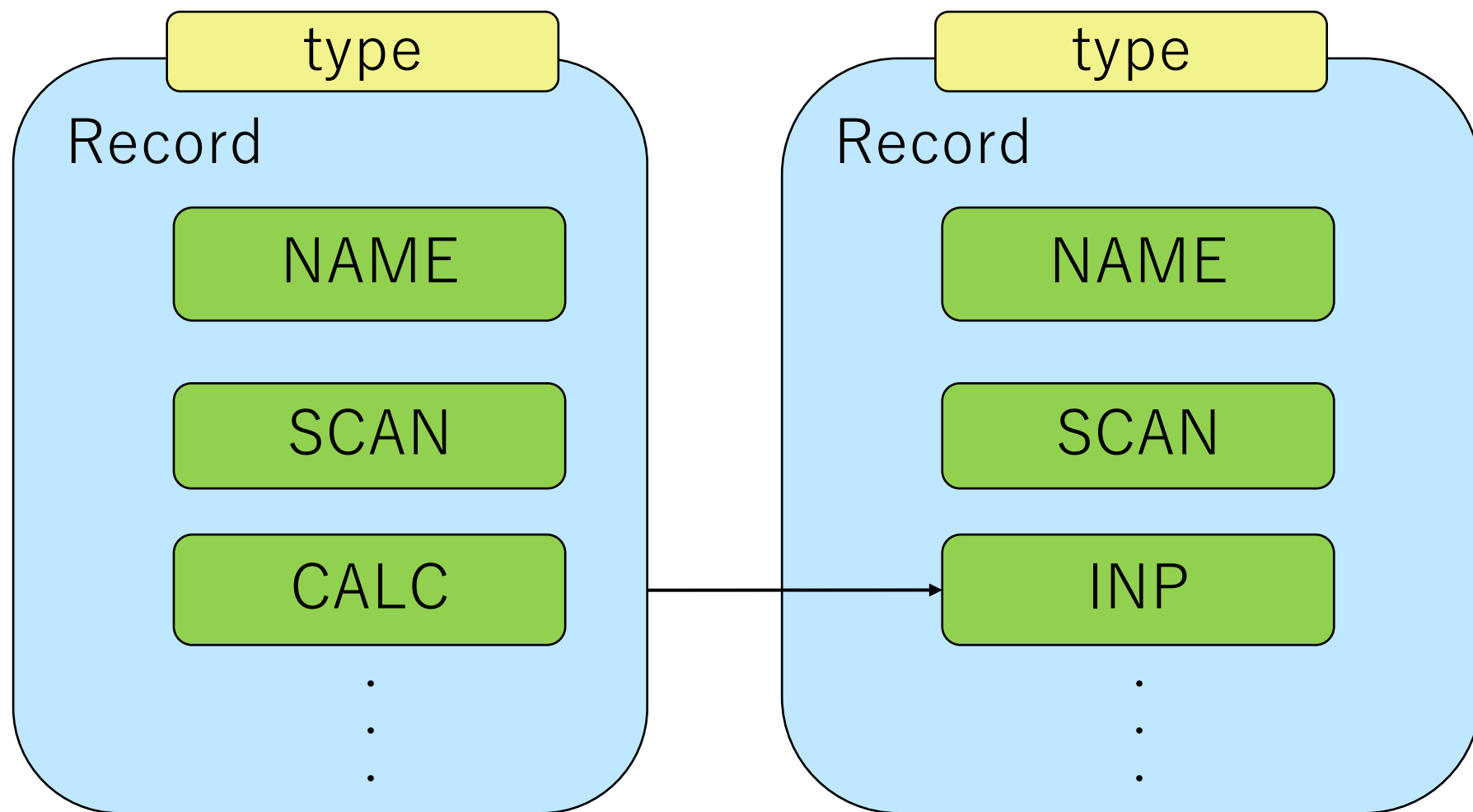


レコード

ハードウェアと接続したり



レコード



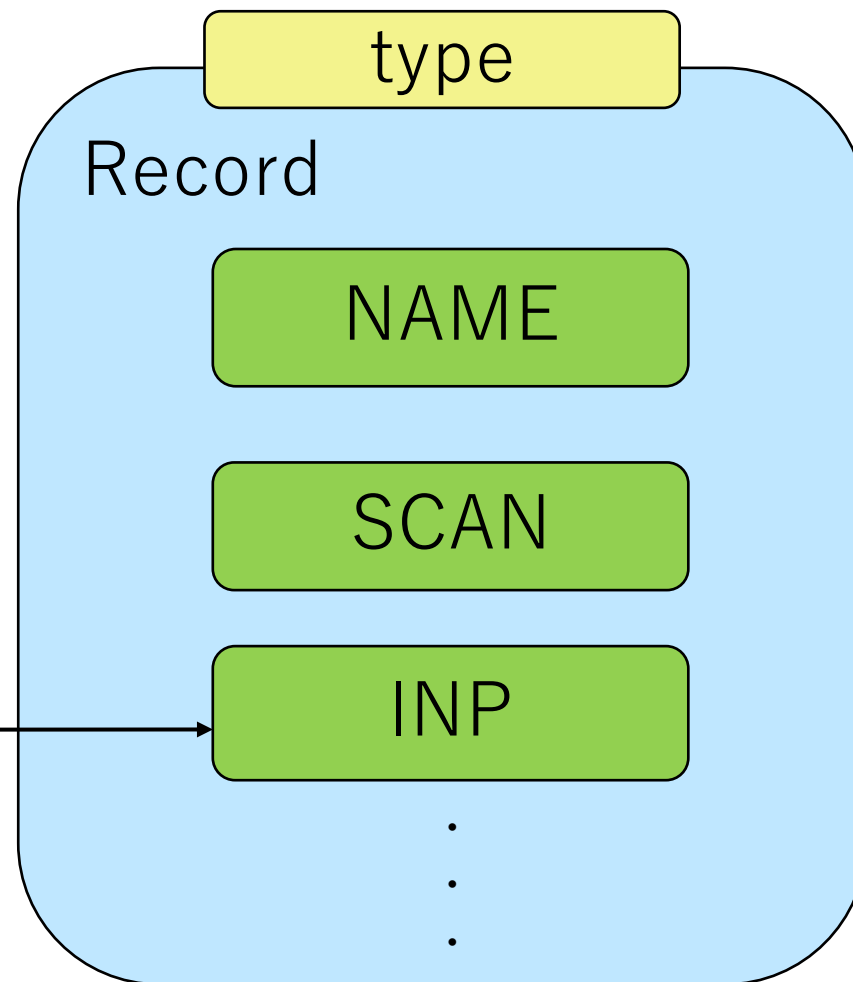
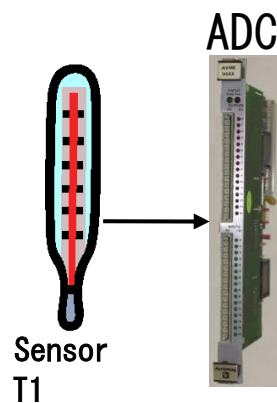
レコード同士を繋いだりする

レコードの動作 (Record Activity)

- レコードは何らかの動作（プロセス）を行う。
 - ハードウェアや他レコードからデータの取得
 - ハードウェアや他レコードへデータの書き込み
 - 演算
 - などなど
- レコードの動作はレコードタイプとフィールドの設定で決まる
- レコードはプロセスされない限り何も動作しない

レコード

レコードがプロセス（動作）すると
センサーのデータをADCで取り込む



レコードタイプ

- ai/ao float型/double型データの入出力（実数）
 - longin/longout 整数型データの入出力
 - bi/bo 0あるいは1の値をとるデータの入出力
 - calc 演算処理
-
- この他にも様々なレコードタイプがあります

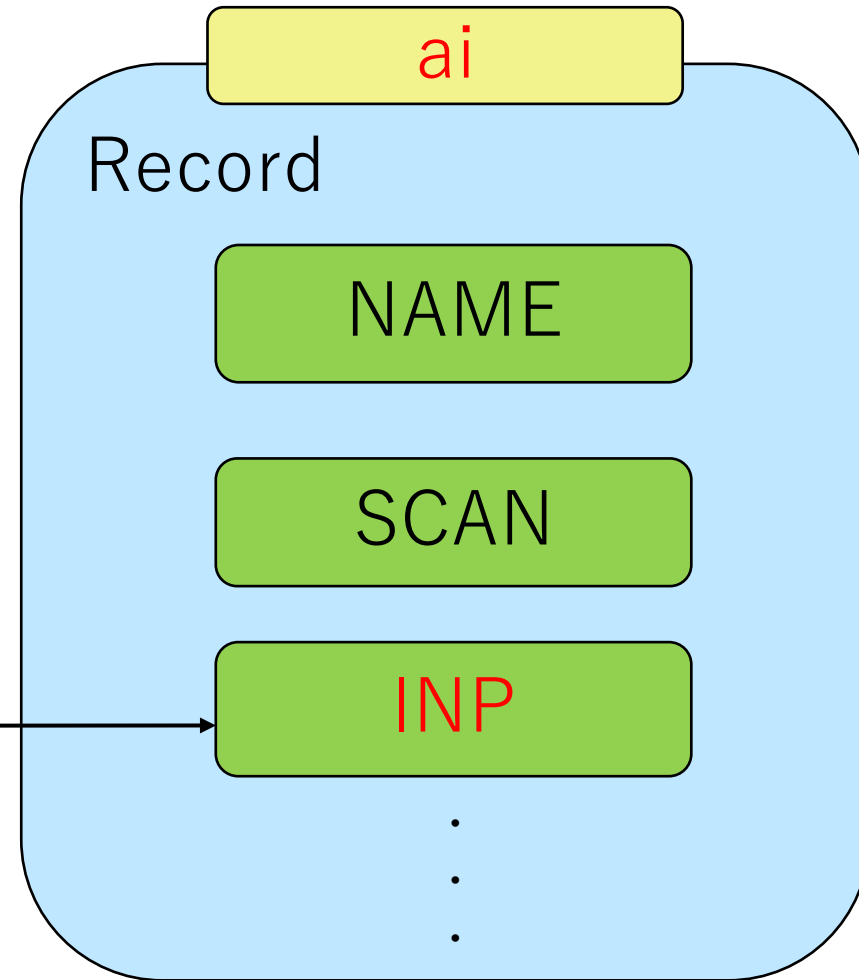
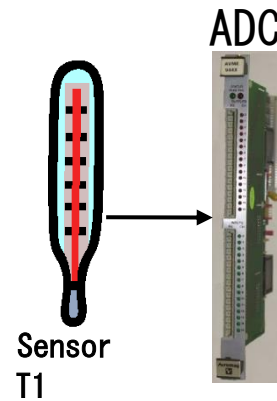
レコードタイプ

例)

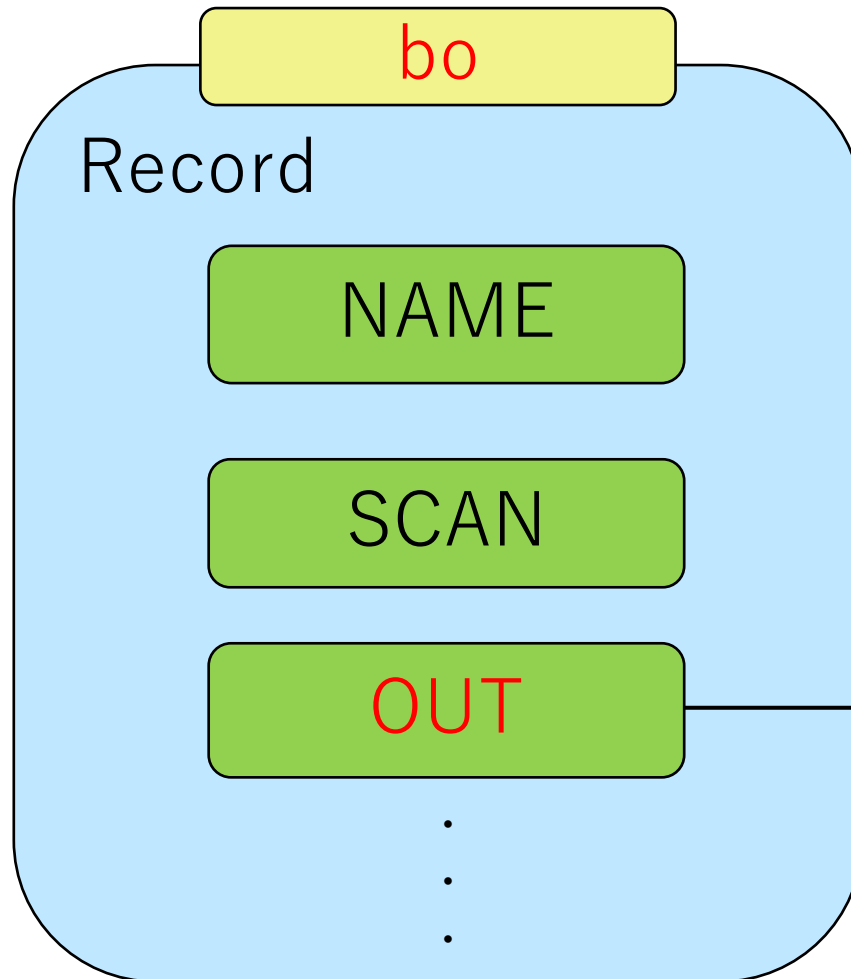
- ai/ao 1.12
 - longin/longout 50
 - bi/bo 0 or 1
 - calc 演算処理
-
- この他にも様々なレコードタイプがあります

レコードの例：ai

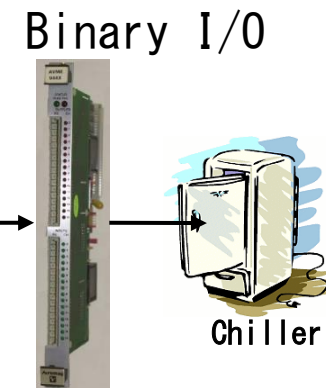
プロセス（動作）するとセンサーのデータをADCでレコードに取り込む



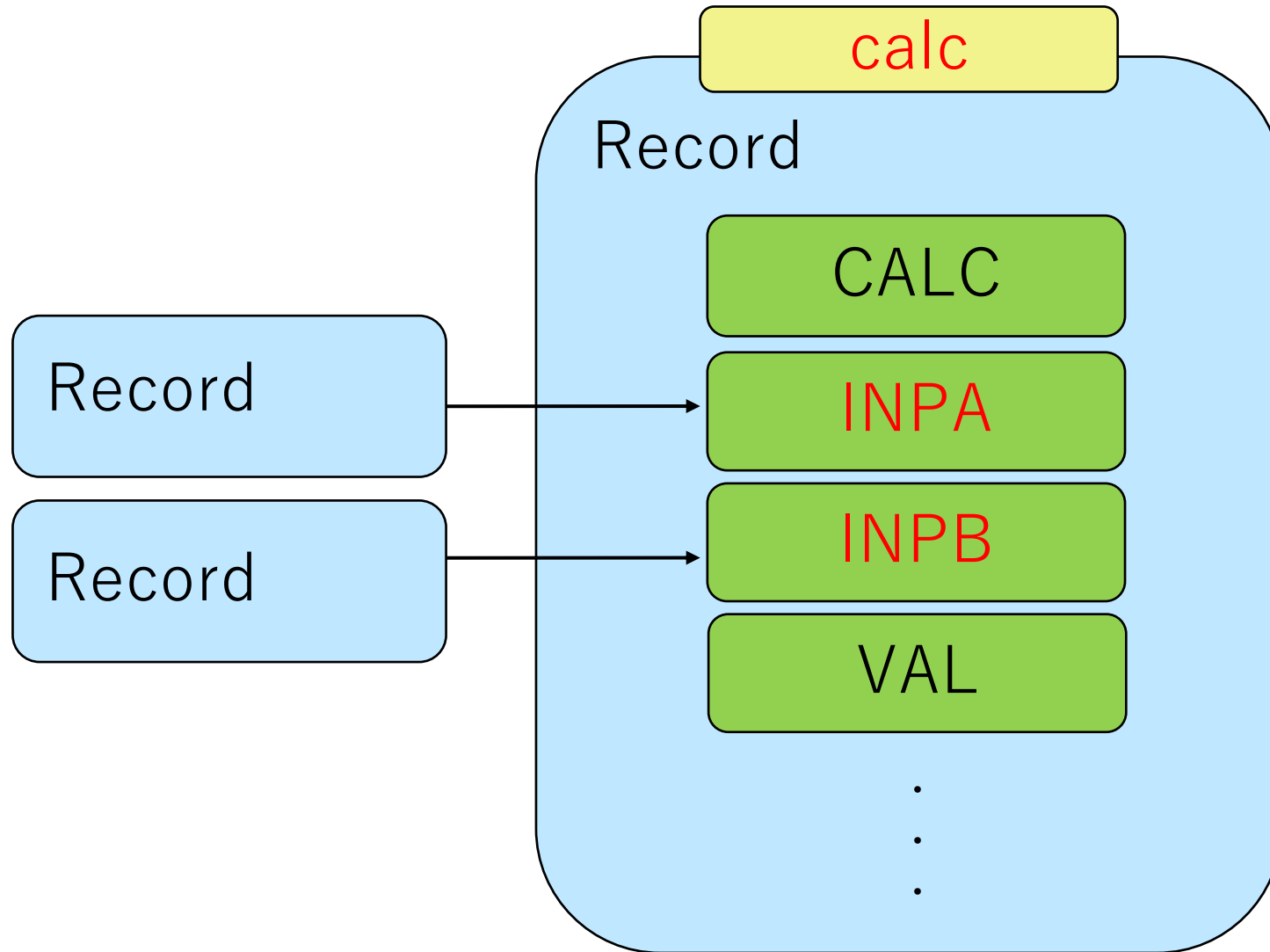
レコードの例：bo



プロセス（動作）するとレコードのデータを Binary I/O (Hi/Lo, 1/0) で出力する



レコードの例：calc



calcのレコードがプロセスすると

1. INPA、INPBの値を取り込み
2. CALCで書かれた演算を行い
3. 演算結果をVALに格納

Analog output Recordの作成例 (Source)

| | | | |
|-------|-----------------------------------|-------|---------------------------|
| | type | レコード名 | |
| | record(ao, "DemandTemp") { | | |
| | field(DESC, "Temperature Demand") | | レコードの説明 |
| | field(SCAN, "1 second") | | Scan Algorithm |
| | field(PINI, "NO") | | Process at Initialization |
| | field(HOPR, "80") | | High operation range |
| フィールド | field(LOPR, "20") | | Low operation range |
| | field(DTYP, "Soft Channel") | | デバイスタイプ |
| | field(DRVH, "100") | | Drive High |
| | field(DRVL, "0") | | Drive Low |
| | field(OUT, "#C0 S0") | | Output link |
| | } | | |

レコードのフィールド

- 全てのレコードにおける共通フィールド（一部）

- Design Timeフィールド

- ✓ NAME レコード名（最大60文字）
 - ✓ DESC レコードの説明（最大40文字）
 - ✓ ASG アクセスセキュリティグループ
 - ✓ PINI IOC起動時にプロセスするかどうか
 - ✓ SCAN スキャンモード
 - ✓ FLNK フォワードリンク

- Runtimeフィールド

- ✓ PROC プロセス実行
 - ✓ PACT プロセスが実行されているか否か
 - ✓ STAT Alarm status
 - ✓ SEVR Alarm severity
 - ✓ TIME タイムスタンプ（最終プロセス時）

レコードのフィールド

□ Input レコードのフィールドの例

- ✓ INP Input link
- ✓ DTYP Device type
- ✓ RVAL Raw data value
- ✓ VAL Value
- ✓ LOPR Low operator range
- ✓ HOPR High operator range

□ Output レコードのフィールドの例

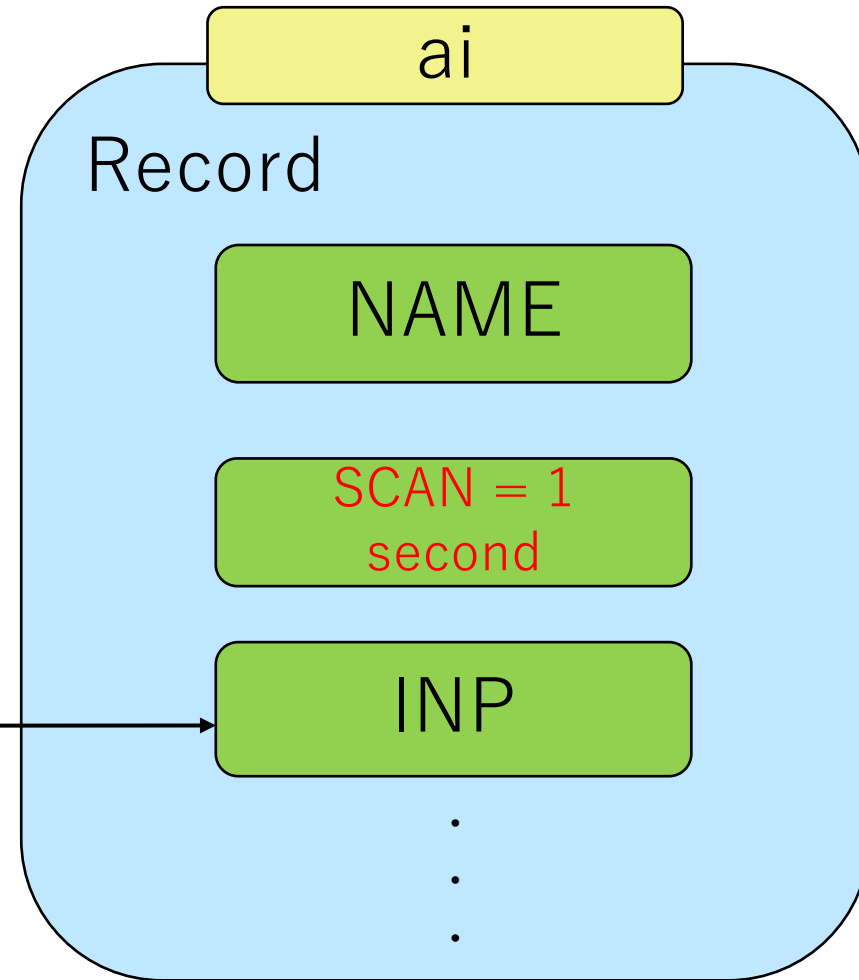
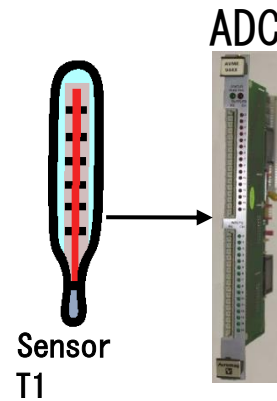
- ✓ OUT Output link
- ✓ OMSL Output Mode Select
- ✓ DRVH Drive High
- ✓ DRVL Drive LOW
- ✓ VAL Value
- ✓ OVAL Output Value

レコードのプロセス方法の指定：SCANフィールド

- レコードのプロセスのタイミングはSCANフィールドで指定する
 - ✓ Periodic 0.1, 0.2, 0.5, 1, 2, 5, 10 (秒)
 - ✓ Passive 外部からの要求でプロセス (デフォルトはこれ)
 - ✓ I/O interrupt 割込みイベント (デバイスがサポートしている場合のみ)
 - ✓ Event イベント動作

レコードのSCAN

1秒ごとにプロセス（動作）してセンサーのデータをADCでレコードに取り込む



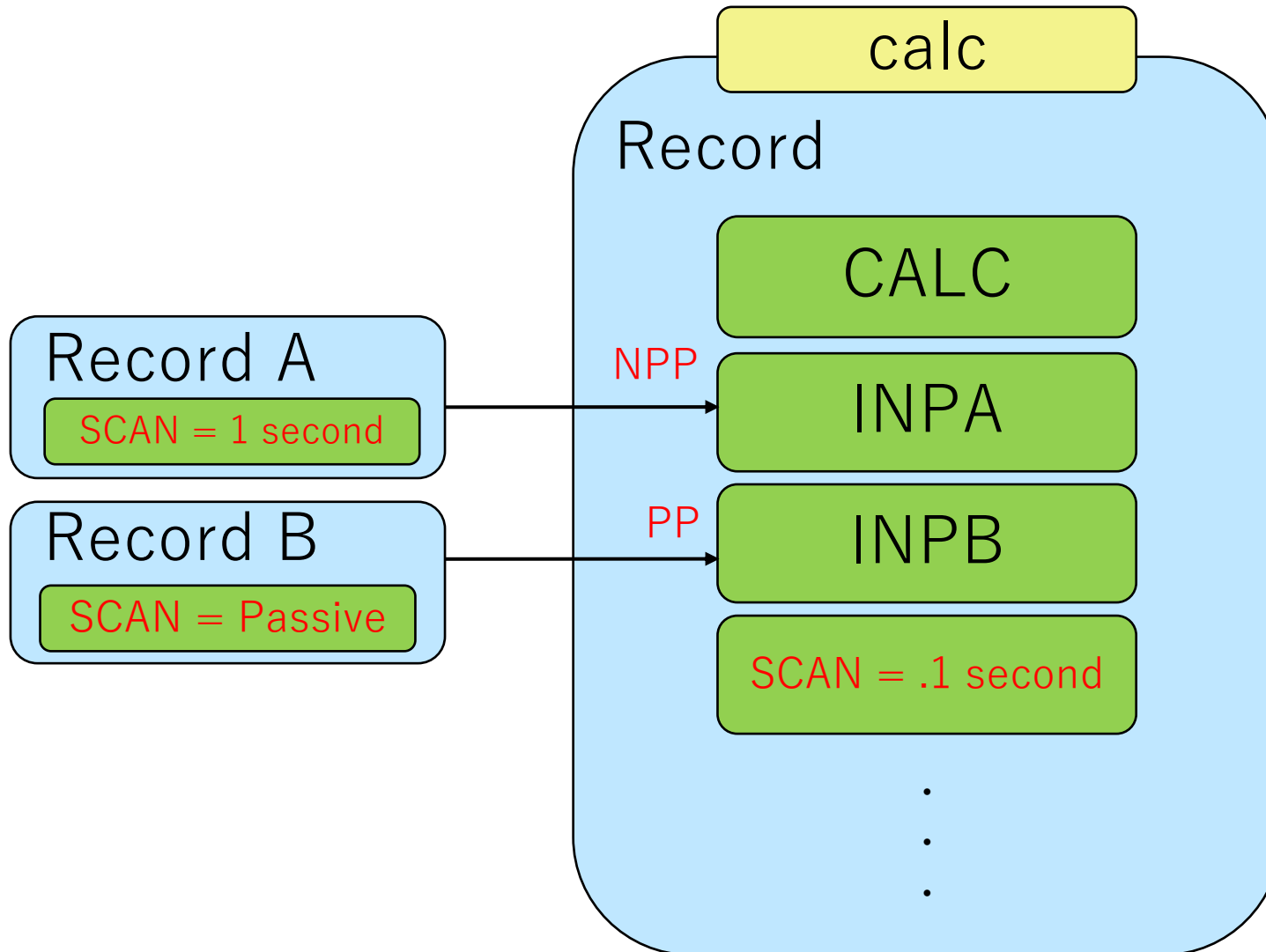
レコードとレコードの接続：リンクフィールド

- リンクは他のレコードとのつながりを表すフィールド
- **INP (Input Link)**
 - データの入力元
- **OUT (Output Link)**
 - データの出力先
- **FLNK (Forward Link)**
 - そのレコードのプロセスが終了したあとにプロセスするレコードの指定

レコード間のリンク

- リンクがレコードを指す場合（Database Linkの場合）はPPかNPPを指定
- **PP : Process Passive**
 - もしもリンク先のレコードのSCANフィールドがPassiveだった場合、**読み込む前**または**書き込んだ後**にそのレコードをプロセスさせる
(If the target record has SCAN=Passive, process it before reading or after writing the value)
- **NPP : Non-Process Passive**（デフォルト）

レコード



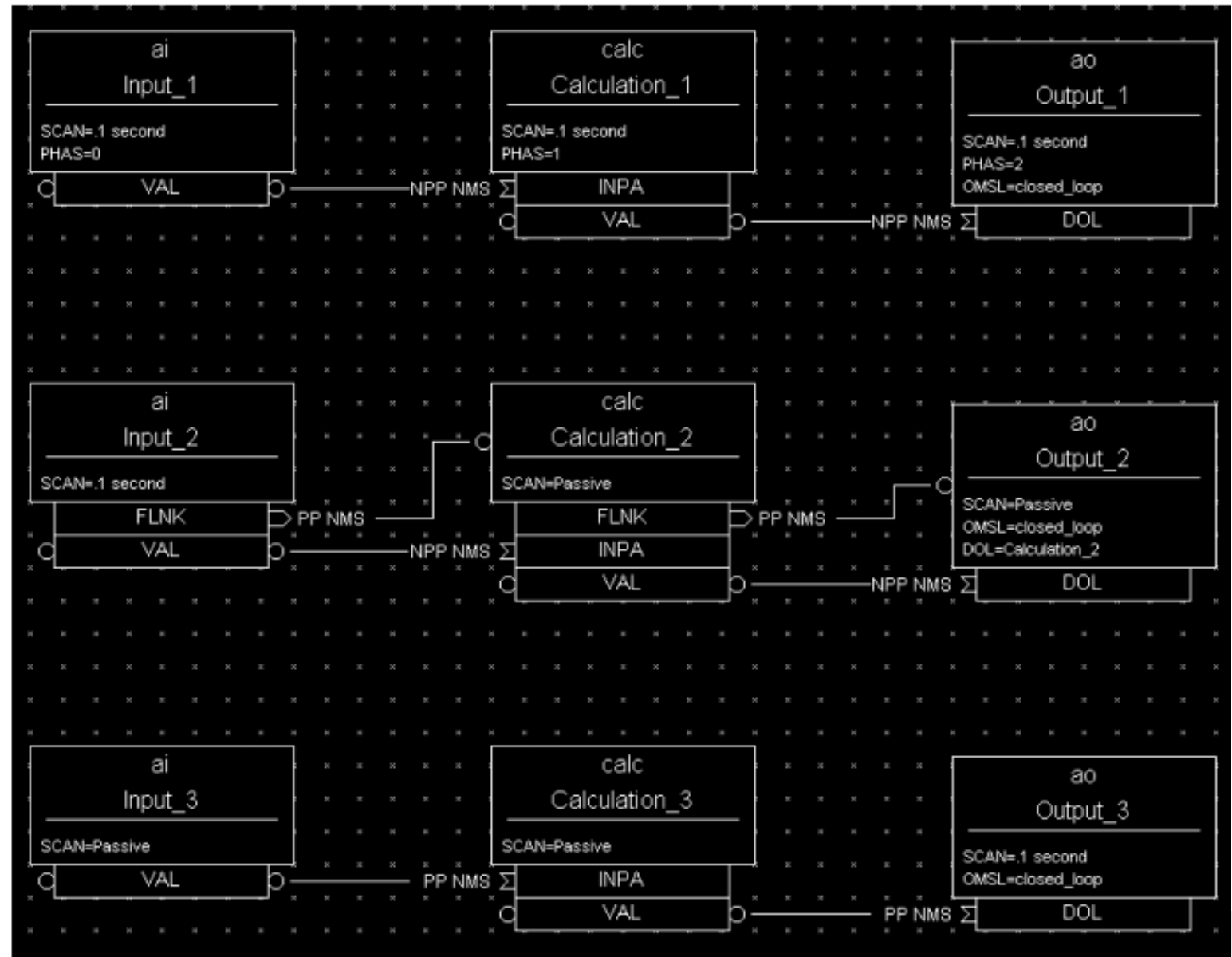
calcのレコードが0.1秒ごとにプロセスして

1. INPAの値を取り込む
2. Record Bをプロセスしたあと値を取り込む
3. CALCで書かれた演算を行う
4. 演算結果をVALに格納

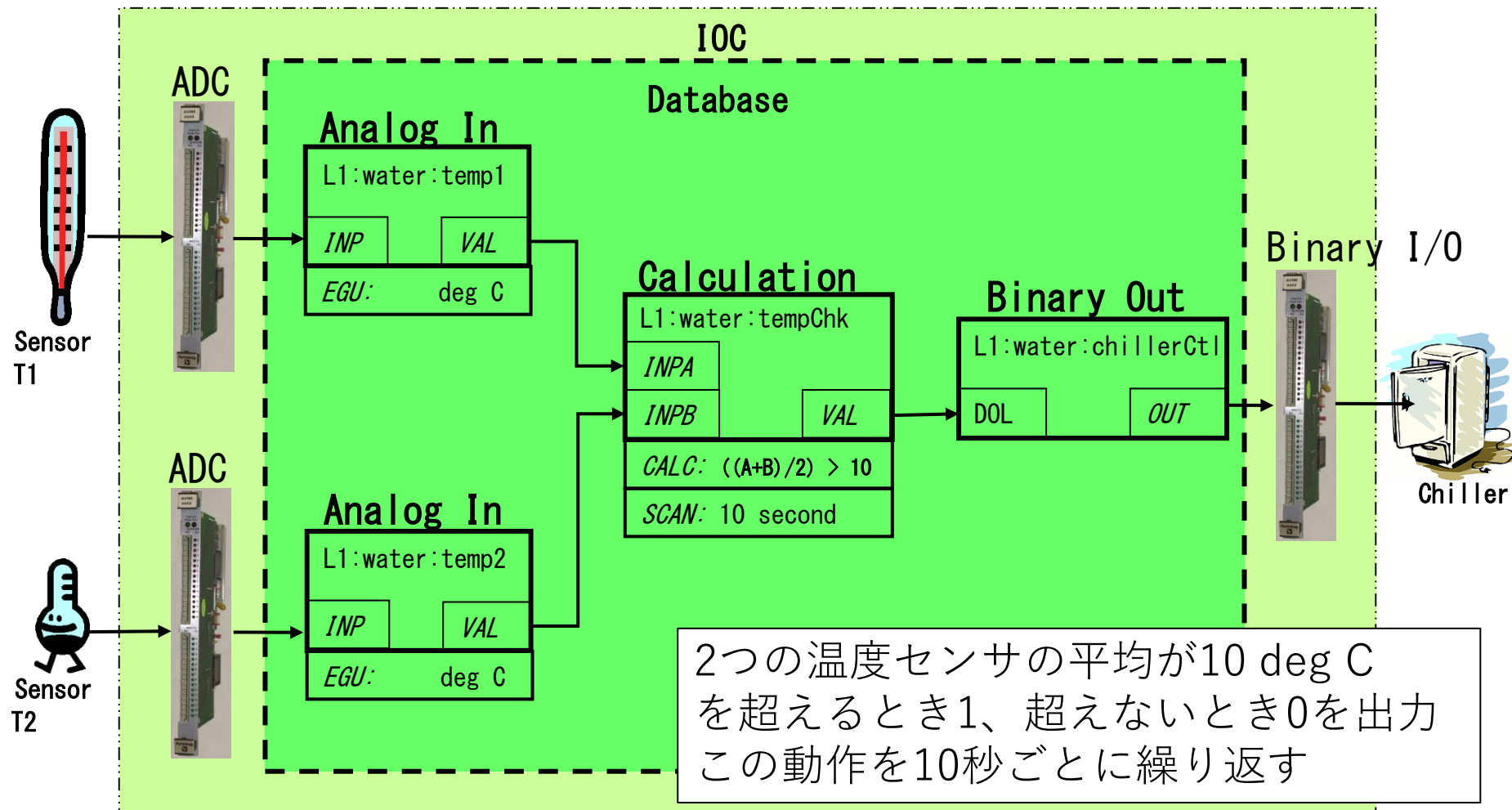
Processing Chains

全て同じ動作をするが、
異なる書き方をしている。

左のレコードから右の
レコードへプロセスしていく。

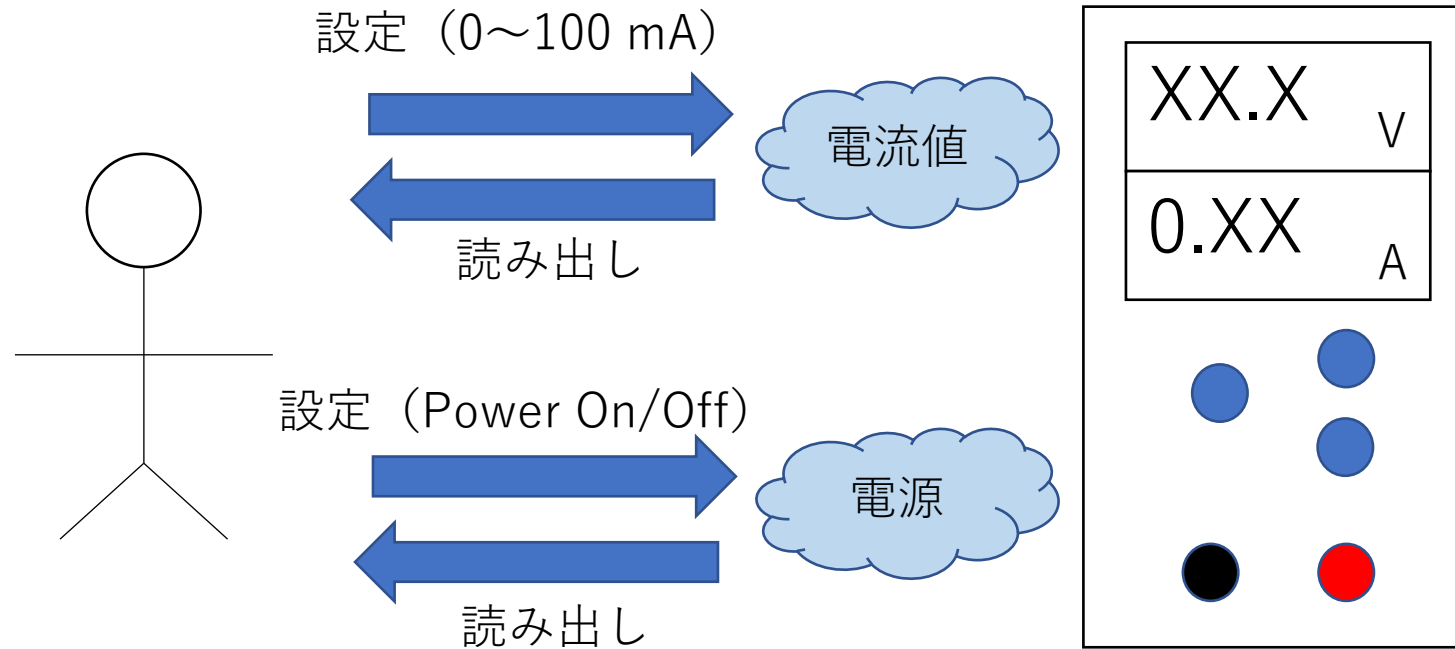


EPICS データベースの例



EPICS データベースで仮想電源作成する

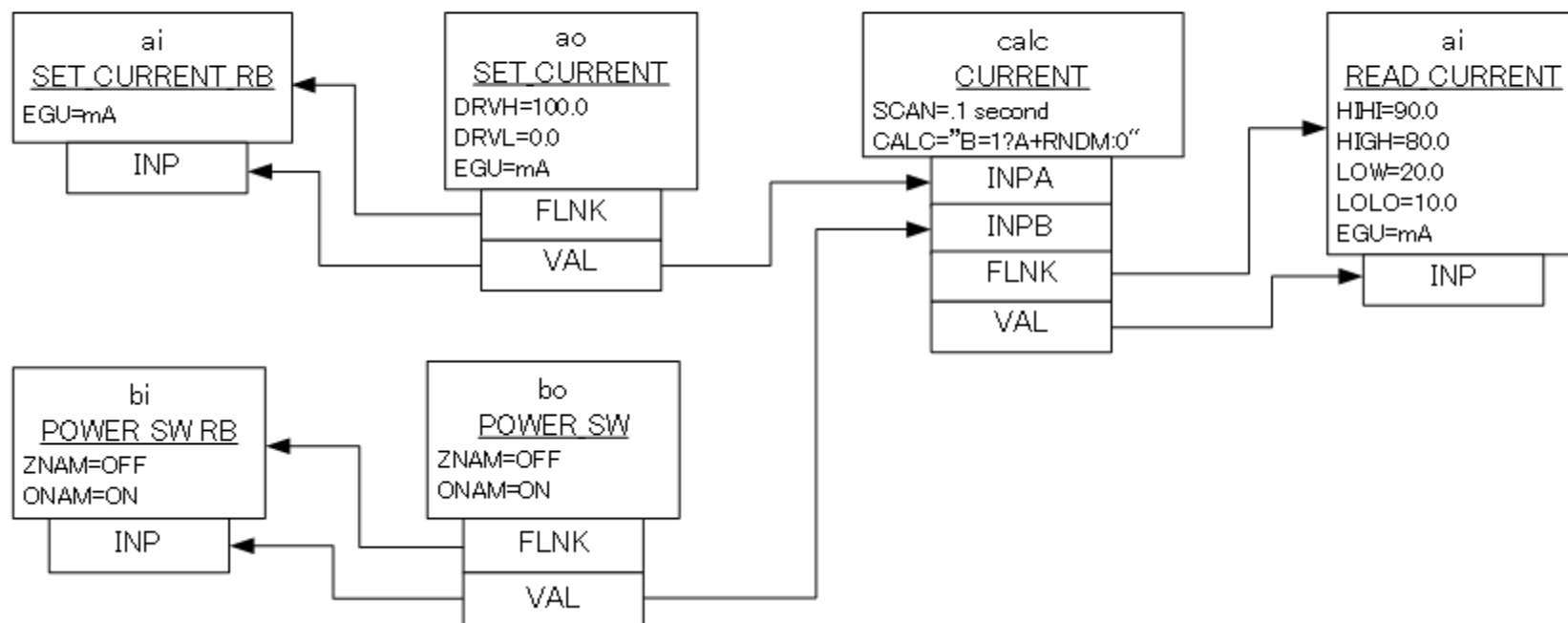
- ここでは仮想的な電源作成を通して制御システム構築を体験する



EPICS Databaseを作成してsoftlocを起動しよう

- 作成するデータベース

Power Supply Simulate EPICS Database



まとめ

- IOCではDatabaseでレコードを定義して管理する
- Databaseでは以下のようなことを定義する
 - どんなレコードが (Record name)
 - いつ (SCAN)
 - どのような順序で (Processing Chains)
 - どのような動作/プロセス (Record type, Fields)