

リモート I/O CONPROSYS nano の制御用 STARS クライアントの 開発と活用

○石井晴乃[#]、仁谷浩明、小菅隆
高エネルギー加速器研究機構 物質構造科学研究所

概要

近年、高エネルギー加速器研究機構 (KEK) 放射光実験施設 Photon Factory (PF) ではより安定した実験装置運用の為、実験室内の環境データ (空調、装置の冷却水、圧搾空気の圧力、ビームライン真空値等) の監視及び記録に関する需要が高まっている。

これらの環境データ取得には、点在する多種多様なセンサのアナログ出力をデジタル化し収集する必要があり、アナログ入力やリモート制御の機能を比較的安価に実現できる、コンテック社製のリモート I/O CONPROSYS nano を採用した。

今回は、Simple Transmission and Retrieval System (STARS) [1]を介して CONPROSYS nano を制御する汎用的なクライアントの開発とその活用について紹介する。

1. はじめに

PF では、放射光源加速器で発生させた放射光から X 線や軟 X 線、真空紫外線などの波長の光を取り出し利用することで様々な物質・生命科学の研究を行っている。

近年の放射光利用実験では、ナノスケール、マイクロスケールの微小なビームを制御することが求められており、これまで以上に環境変動の影響を考慮しなければならない。そのため実験ホールの気温、湿度、冷却水流量や水温などの環境データの収集が重要となっている。PF 実験ホールでは以前よりリモート I/O を使用した分散型の計装システムを構築し、環境測定を行っていたが、老朽化によるリモート I/O の故障や当時設置したセンサでは現在の実験ホールを網羅することができなくなったため再整備を行うこととなった。環境モニターシステムの再整備では、設置が容易であり汎用性の高い I/O を有していることや安価であることに着目し、コンテック社製のリモート I/O CONPROSYS nano と制御プロトコルとして STARS を採用した。

主に CONPROSYS nano の導入と開発したソフトウェアの詳細を報告する。

2. CONPROSYS nano

CONPROSYS nano は RS-232C、LAN インターフェイスに加え 4 つの拡張スロットを備えた CPU ユニットと CPU ユニットに差し込む各種 I/O モジュールで構成され、I/O モジュールの組み合わせ次第で様々な機能を実装することができる。CPU は Modbus TCP Slave や Windows 環境において PC 本体の COM ポートと同様に使用できる仮想 COM ドライバに対応している。今回 Windows 環境以外での実装が予想されたため Modbus TCP Slave で通信する制御プログラムの開発を行った。

2.1 CPU ユニット

CONPROSYS nano の CPU ユニットは CPSN-MCB271-1-041 と CPSN-MCB271-S1-041 がある。ハードウェア的な違いとして CPSN-MCB271-1-041 は LAN ポートが 2 つ用意されておりデジチェーン接続のようにネットワークを構成することが可能である。ソフトウェア面では両者に違いはなく、設置環境に合わせて選択すればよい。図 1 は CONPROSYS nano の CPU ユニットと 1 スロット目に熱電対入力モジュールを取り付けた写真で、CPU ユニットと I/O モジュールには、CPU ユニットの電源コネクタより 12V-24 VDC で供給を行う。消費電力は使用する I/O モジュールの種類と数によって異なるため、各モジュールに合わせて電源を選定する必要がある。またアナログ入出力を扱う際はノイズを抑制するために CPU ユニットへの安定した電源の供給が必要となる。

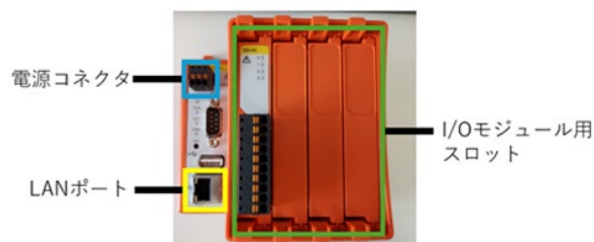


図 1. CONPROSYS nano

2.2 I/O モジュール

上記で紹介した CPU ユニットにアナログ入出力やデジタル入出力など様々な機能を実装するためには I/O モジュールが必須となる。1 つのユニットに実装できる I/O モジュール数は最大 4 つで、1 モジュールにつきおおよそ 4 から 8 チャンネルの入出力が用意されている。I/O モジュールのうち Modbus TCP プロトコルで制御可能な I/O モジュールを表 1 にまとめる。これらのハードウェア接続図を図 2 に示す。

表 1. Modbus TCP プロトコル対応 I/O モジュール

モジュール名	型式
アナログ入力 モジュール	CPSN-AI-1208LI CPSN-AI-2408LI
アナログ出力 モジュール	CPSN-AO-1602LC
熱電対入力 モジュール	CPSN-SSI-4C
デジタル入力 モジュール	CPSN-DI-08L CPSN-DI-08BL CPSN-DI-16BCL
デジタル出力 モジュール	CPSN-DO-08L CPSN-DO-08BL CPSN-DO-08RL CPSN-DO-08BRL
デジタル入出力 モジュール	CPSN-DIO-08SL
リレー モジュール	CPSN-RRY-4PCA
カウンタ モジュール	CPSN-CNT-3201I
シリアル通信 モジュール	CPSN-COM-1PD

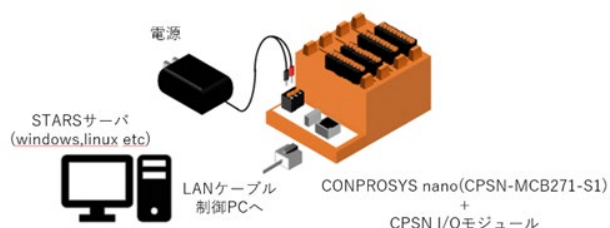


図 2. ハードウェア接続図

3. ソフトウェア構成と開発プログラム

3.1 STARS

今回開発した CONPROSYS nano 制御クライアントの基本ソフトウェアは、PF でビームライン標準制御ソフトウェアとして利用されている STARS を採用している。STARS は STARS Server に対し TCP/IP Socket を利用して複数のクライアントプログラムが接続し、相互にテキストベースメッセージの送受を行うシステムである。最もシンプルなソフトウェア構成図を図 3 に示す。

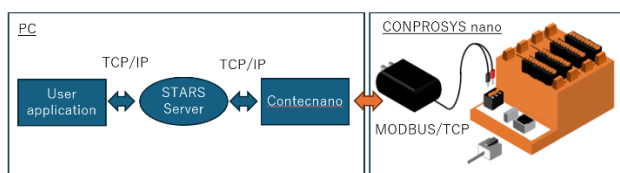


図 3. ソフトウェア構成

User application は STARS を介して CONPROSYS nano 制御クライアントである Contecnano にコマンドを送ることで、コマンドで指定した I/O モジュールの特定チャンネルからデータを取得することができる。例として、1 スロット目に I/O モジュール CPSN-AI-2408LI を取り付けた場合のアナログ入力データを読む送受信コマンドを挙げる。

[User application 送信コマンド]

Contecnano.ai.ch3 GetValue

[User application 受信コマンド]

Contecnano.ai.ch3>Userapp @GetValue 4.21

上記のようにすべての I/O モジュールに対しコマンドは CONPROSYS nano の I/O モジュール名の一部 (ai や di 等) とチャンネル番号 (ch) の組み合わせでできている。

3.2 MODBUS TCP

CONPROSYS nano 制御クライアントは MODBUS TCP Slave と呼ばれるプロトコルで通信する必要がある。MODBUS 通信ではコイル、入力ステータス、入力レジスタ、保持レジスタの 4 つのデータ領域に対し決められたデータフォーマットのコマンドで読み書きを行う。表 2 に MODBUS TCP のデータフォーマットと入力レジスタの先頭アドレスから 2 Byte のデータを読み出すときの値を例として示す。クエリデータは読み書きするレジスタのアドレスや書き込むデータからできており、そのフォーマットはファンクションコードによって異なる。

表 2. MODBUS TCP データフレームの構成

フィールド名	例 (入力レジスタ読み出し)
スレーブアドレス (1byte)	0x00
ファンクションコード (1 byte)	0x04
クエリデータ (N Byte)	0x00,0x00,0x00,0x02

3.3 CONPROSYS nano 制御クライアント

クライアントプログラムの開発は Microsoft Visual studio 2022 で行った。Linux OS では Ubuntu 22.04 での動作確認を行っている。環境構築が容易なことからプログラミング言語として Python 3 を採用した。本クライアントは Python 3.8.6 以降での動作確認を行っている。

CONPROSYS nano はスロットに接続された I/O モジュールの組み合わせに応じて、それらの入出力データを格納するレジスタアドレスが変化する。また I/O モジュールによって 1 チャンネルのデータサイズが異なるだけでなく、シングルエンド入力や差動入力など複数の入力タイプを持つものや入力レンジを変更できるものもある。そのためチャンネル数や、バイトデータを変換する際の換算式をプログラムで適

宜変更する必要がある。

これらのことを踏まえて CONPROSYS nano クライアントが User application からの問い合わせに対し適切なデータを返すために、まず図4のようなフローで初期設定を行う。

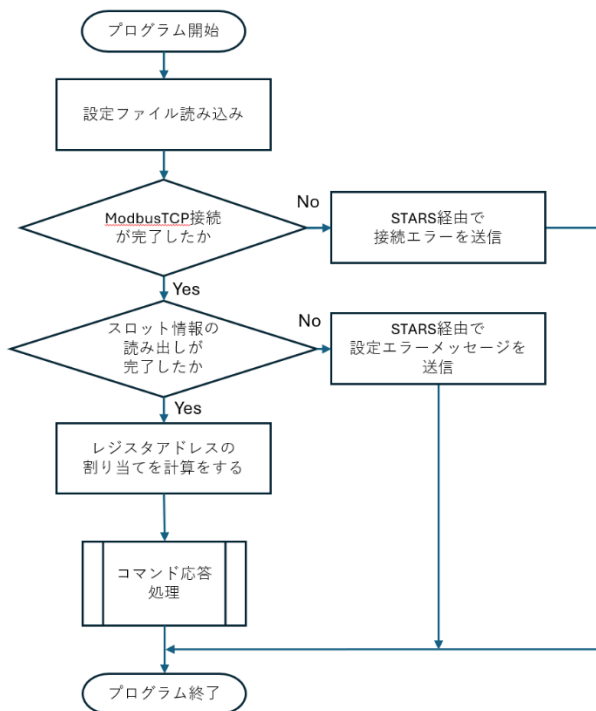


図4. 初期設定フローチャート

まず通信に必要な IP アドレスやポートなどの情報を設定ファイルより読み込み CONPROSYS nano に接続する。次に、スロットに接続されている I/O モジュールの番号を MODBUS 通信により取得する。レジスタアドレスの割り当てでは、各スロットの I/O モジュールに割り当てられるレジスタアドレスとチャンネル番号を計算する。これをもとに図5のような STARS コマンドから MODBUS TCP のデータフレームに変換する。次に図6のようなフローで問い合わせに回答するようにプログラムを作成した。



I/Oモジュール	STARSコマンド	Modbus TCPデータフレーム
Slot1:AI-1208LI	Contecnano.ai.ch0 GetValue	[¥0x00,¥0x04,¥0x00,¥0x00,¥0x00,¥0x01]
Slot1:AI-1208LI	Contecnano.ai.ch1 GetValue	[¥0x00,¥0x04,¥00,¥0x00,¥0x01,¥0x01]
Slot1:AI-1208LI	Contecnano.ai.ch2 GetValue	[¥0x00,¥0x04,¥00,¥0x00,¥0x02,¥0x01]
⋮	⋮	⋮
Slot2:AI-2408LI	Contecnano.ai.ch8 GetValue	[¥0x00,¥0x04,¥00,¥0x00,¥0x08,¥0x02]

図5. STARS コマンドと MODBUS TCP データフレーム

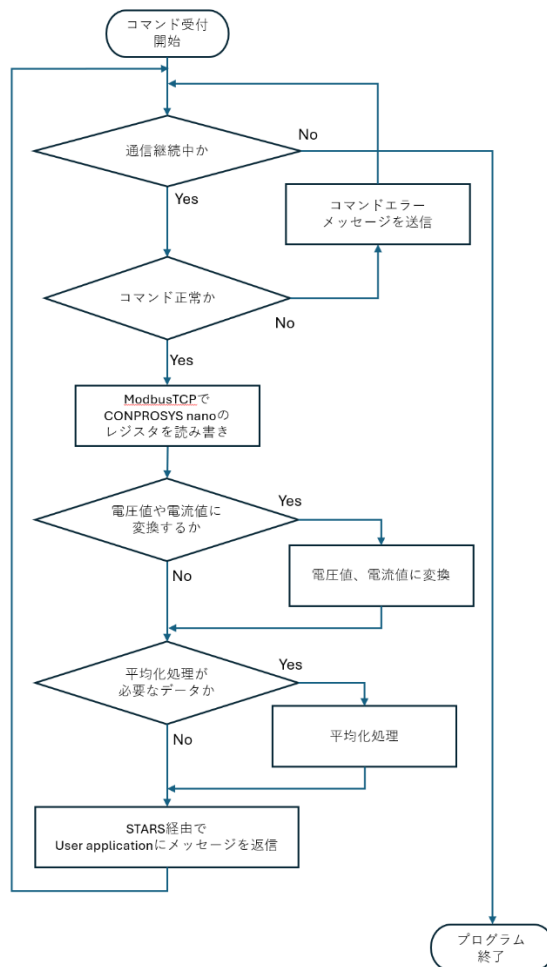


図6. コマンド応答処理のフローチャート

MODBUS 通信で取得できるデータは MODBUS データ形式のため、電圧値や電流値などを扱う場合は計算を行う。また、それらのデータを平均処理したい場合には平均処理を行う。これら 2 つの処理を行うためのパラメータはプログラムが起動時に読み込む設定ファイルに書き込む必要があるが、設定しない場合でも動作するプログラムとなっている。

このように本クライアントプログラムでは、設定が必要な情報のみ設定ファイルに記載し、できる限り汎用的に CONPROSYS nano を制御できるような工夫をした。

4. 活用例

PF 実験ホールでは 30 台以上の CONPROSYS nano が稼働しており、アナログ電圧、電流出力のセンサからデータを収集し、それらのデータを気温や温度、流量や圧力などに変換しデータベースに蓄積している [2]。



図 7. PF 実験ホール環境データ

5. まとめと今後について

今回開発した CONPROSYS nano 制御用 STARS クライアントは複雑な設定を用いず CONPROSYS nano の様々な I/O モジュールを制御することができる。PF ではアナログ出力センサからのデータ収集や、デジタル出力 I/O モジュールを用いた機器制御など幅広い場面で活躍している。今後は開発したプログラムをブラッシュアップするほか、マニュアル整備を行う予定である。

参考文献

- [1] <https://stars.kek.jp>
- [2] 石井 晴乃ほか、9PS10「実験ホール環境モニタリングシステムの構築」、第 35 回日本放射光学学会年会・放射光学科学合同シンポジウム、2022 年 1 月 7 日～9 日。