

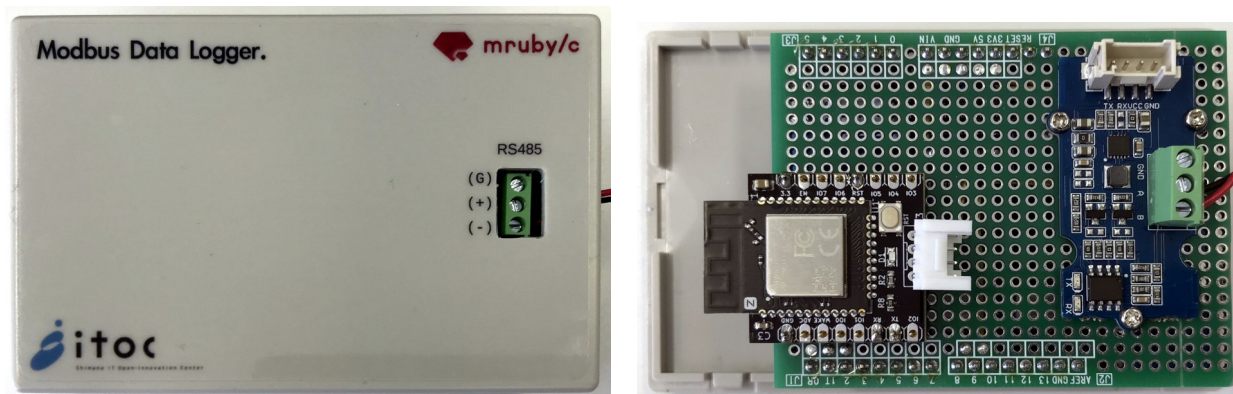
# Modbus ベース温度ロガー開発について

2022/1/19

ITOC 専門研究員 東裕人

(株)島根情報処理センター製「Rboard」に RS485 コンバータを接続し、Modbus 経由でオムロンの温調器の温度ロガーの開発を行ったので報告する。

## 外観



## ハードウェア

Modbus は Programmable Logic Controller (PLC) 用のフィールドネットワークであり、物理層は RS485 が使われることも多い。今回テスト対象としたオムロン製の温調器 (E5CC-QX2ASM-004) でも RS485 が採用されているため、RS485 専用機として設計した。

RS485 とは UART 経由で送受信するものとし、今回は Grove 規格で接続が可能なモジュール SeedStudio 社製の 103020193 (<https://www.seeedstudio.com/Grove-RS485-p-2924.html>) を採用した。

WiFi モジュールは、Grove 規格で接続が可能な製品の中から、入手性がよくスイッチサイエンスにて購入可能な CRESCENT-014 を選定した。(<https://www.switch-science.com/catalog/5795/>) ただし、後述するとおり、開発時はケーブル一本で接続して非常に簡易に作業ができたが、基板の固定ができないため、最終成果物では半田付けとしている。

## ソフトウェア

本品は、UART を使い Modbus (RS485) 通信を行う。そのため、mruby/c 標準の UART クラスを使ってプログラミングが可能である。また、Modbus プロトコルも mruby/c を使って実装する事とし、ユーザープログラムとの分離を行わないことで、内部動作を明確にする。

UART に関して、Rboard は標準的には UART が 1 系統しかユーザープログラムに開放されていない。しかし、今回のシステムでは、UART を要求するインターフェースが、RS485 と WiFi モジュールの 2 系統必要となる。

一方、grobe Digital 端子は今回のシステムでは使わないため、UART 端子と Digital 端子をプログラムで切り替えながら、一つの UART ハードウェアを共用することにした。

### 【データ取得部抜粋】

```
def modbus_read_var( sio, slave_address, read_address, count = 1)
# Modbus プロトコルに従い、リクエスト文字列 (+CRC) を作る
s = sprintf("%c\x03%c%c%c%c".b, slave_address, read_address >> 8, read_address & 0xff,
count >> 8, count & 0xff );

crc = crc16(s)
s << (crc & 0xff) << (crc >> 8)

# リクエストを送信
sio.write(s)

# 結果を受信
r1 = read_with_timeout(sio, 5)

(後略)
```

サーバーへのデータ送信は、http post プロトコルで REST-API で受け付けるサーバーに対してデータ送信を行う事を想定した。ここで、t\_current は現在の温度、t\_preset は目標温度である。

### 【送信例】

```
URL
http://api.example.jp/datalogger.rb?ctrl=omron_tc&action=post"

POST DATA
{"type":"OMRON TC", "mac_address":"4C:55:CC:18:43:BD", "t_current":68.1, "t_preset":68}
```

## 稼働試験

以下の構成、条件のもとで、稼働試験を行った。

- 本機と温調器とは、長いケーブルの準備が間に合わなかったため、簡易的に1メートルのツイストペアケーブルで接続する。
- 温調器側には、120Ωの終端抵抗を端子台に共締めする。
- 温調器にソリッドステートリレー経由で負荷を接続する。今回は家庭にあるもので再現可能という制約から、負荷として電熱器、加熱媒体として水500cc、卵2個を選定した。
- 温度センサー(Pt100)を、負荷と十分な熱伝導性を確保した状態で接続する。今回は媒体が水なので、センサー感温部全体が確実に浸かるよう固定する。



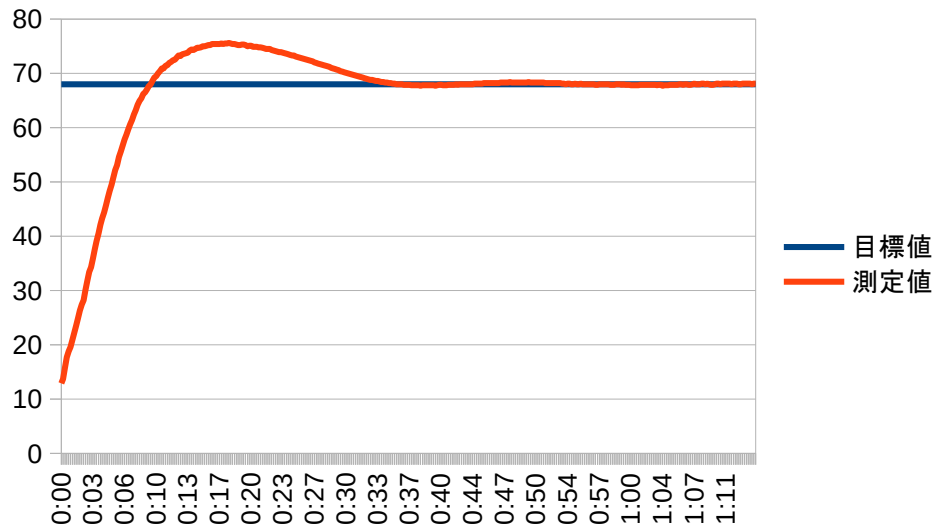
試験装置全体



センサー感温部 拡大

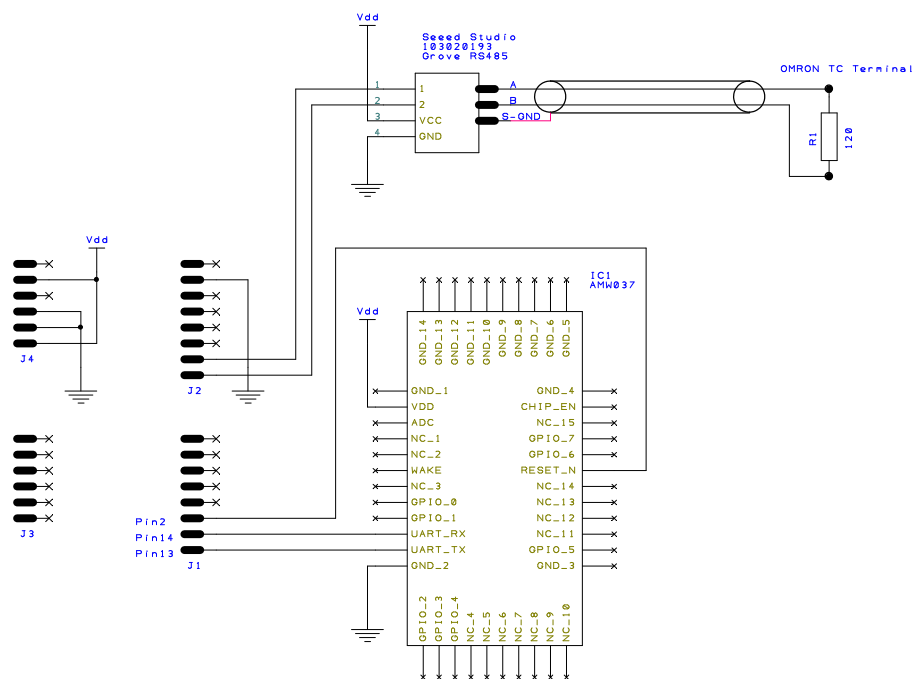
## 試験結果

測定値が目標値に対して一旦オーバーシュートし、30分程度で目標値に安定する様子が記録できた。約1時間稼働させた時のグラフを以下に示す。



## 基板回路図

WiFiモジュールの取付のため、Arduino用ユニバーサル基板を使って、WiFiモジュールおよびRS485モジュールの取付を行っている。以下に回路図を示す。



## パーツリスト

別紙