

# UECS用ロジック開発ツール 活用マニュアル2

Arsprout 8ch 制御盤を使った活用例



西日本農業研究センター  
2019年3月

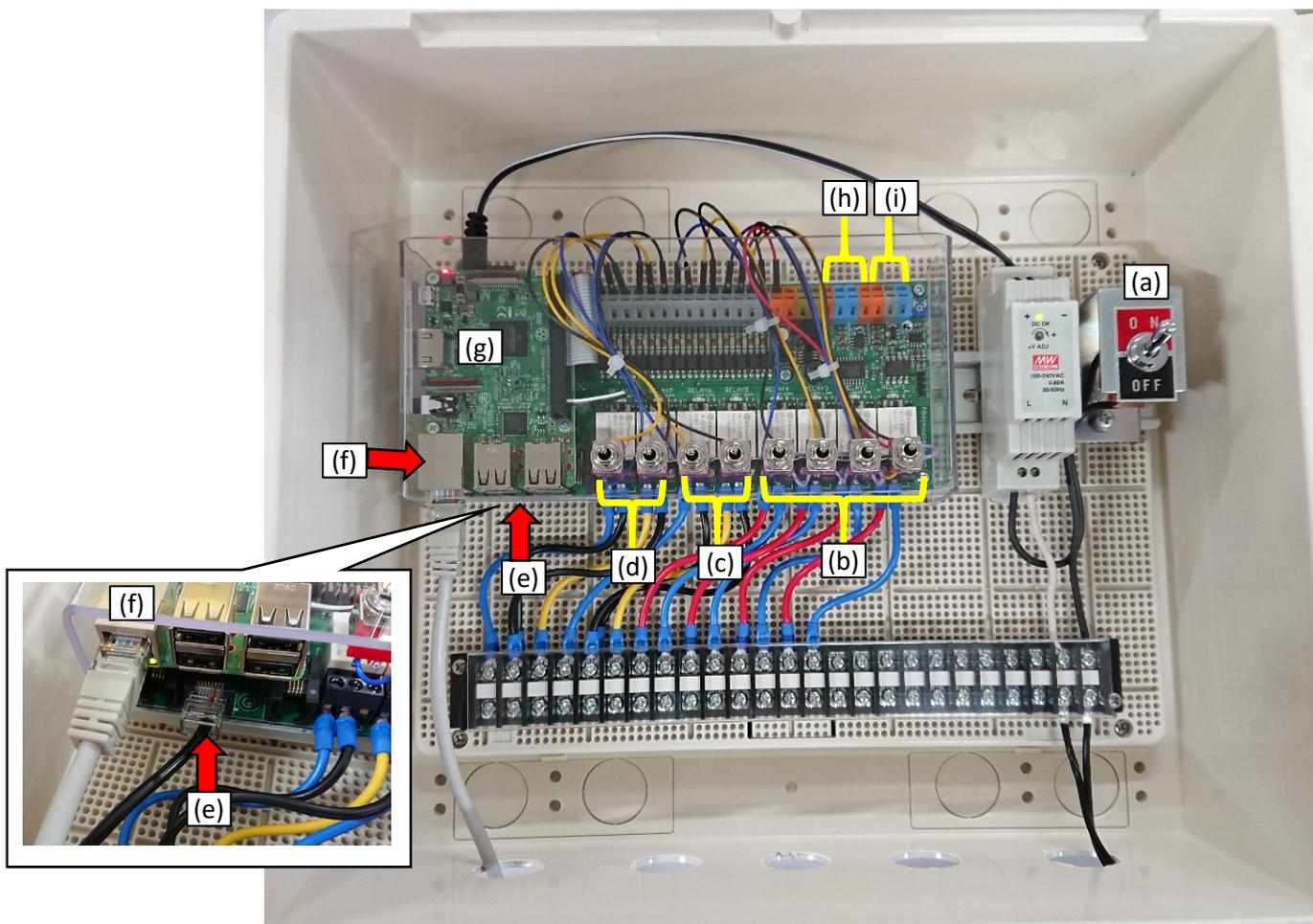
## 目次

第1節	制御盤の設定	1
1.	Arsprout 8ch 制御盤とは	2
2.	起動	3
3.	制御盤の設定確認	4
4.	リレーの設定確認	5
5.	温度センサの設定変更	6
6.	アナログ電圧入力の設定変更	7
7.	制御モードの変更	8
8.	ロジック開発ツールの起動	9
9.	機器の登録	10
10.	機器の登録	11
11.	動作テスト	12
第2節	単純なサーモスタットを作る	13
1.	フローの配置	14
2.	制御内容の作成 (1)	15
3.	制御内容の作成 (2)	16
4.	完成したスクリプト	17
5.	制御の実行	18
第3節	サーモスタットの改良	19
1.	電球を使ったサーモスタットの実験	20
2.	単純なサーモスタットの弱点	21
3.	単純なサーモスタットの弱点	22
4.	実行間隔の調整	23
5.	実行間隔の調整結果 (30分間のデータ)	24
6.	不感帯付きのサーモスタット	25
7.	2つのリレーを使った制御 (1)	26
8.	2つのリレーを使った制御 (2)	27
付録	サンプルプログラム	28

## 第 1 節 制御盤の設定

# 1、Arsprout 8ch 制御盤とは

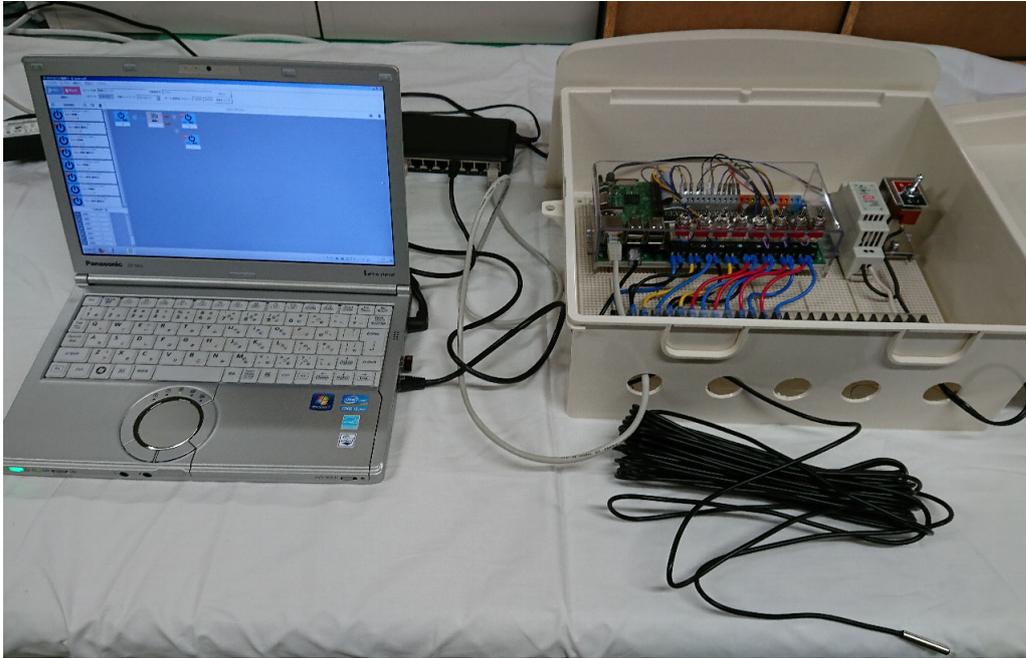
Arsprout 8ch制御盤とは株式会社ワビットが開発し、サカタのタネが販売を行っているUECS対応制御盤です。ここでは、完成品でソフトウェア書き込み済みのSDカードをセットアップ済みのものを使用します。



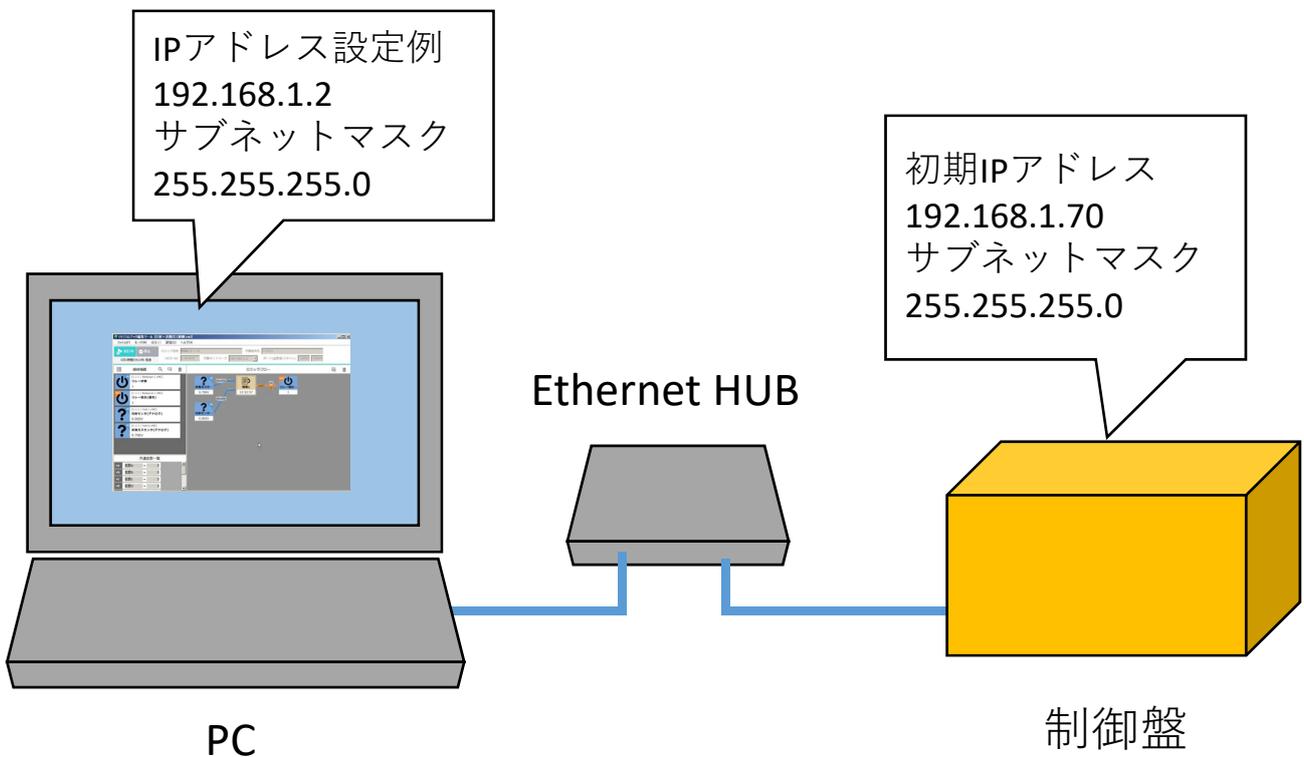
- (a)電源スイッチ
- (b)無電圧接点リレー×4ch (最大5A)
- (c)(d)窓開閉モーター制御リレー(2個でモーター1つを制御)
- (e)1-wire温度センサ接続ポート(マイコンの下の基板)
- (f)LANポート
- (g)制御用マイコンボード(Raspberry Pi3)
- (h)アナログ電圧入力×2ch(0~10V入力可能)
- (i)アナログ電圧出力×1ch

※図では写真撮影のため複数の部品を取り外してあります。  
※リレー8chを搭載していますが、モーターにはリレーを2つ使うので、このモデルは通常リレー4chとモーター2chの制御盤になっています。

## 2、起動



※付属の1-wire温度センサも接続します。



最初にPCと制御盤の間をHUBを介してLANケーブルで接続し、全ての機器の電源を入れます。PCのIPアドレスと制御盤のIPアドレスが重ならないように設定します。制御盤の初期IPは192.168.1.70です。制御盤は起動に30秒～1分かかります。

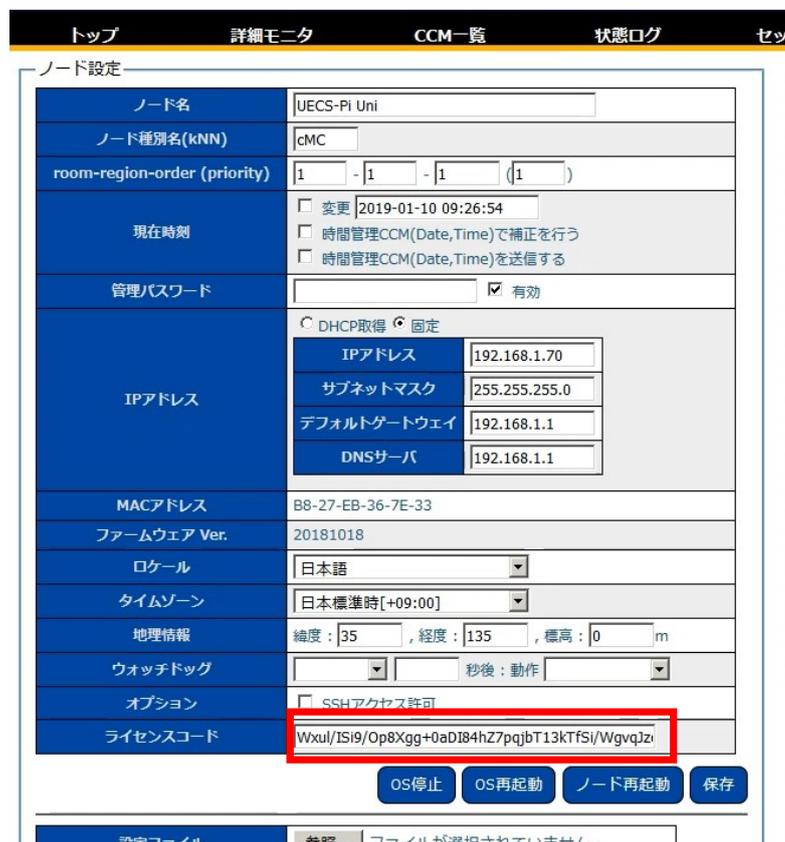
### 3、制御盤の設定確認



(1)PCからブラウザのURL指定欄に"192.168.1.70"と入力してアクセスすると、ログイン画面が出てくるので、初期パスワード"admin"を入力します。



(2)次に出てきた画面からセットアップ→ノード設定を選びます。



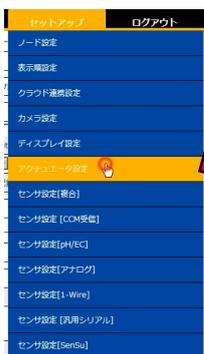
(3)一番下に有効なライセンスコードが入力されていることを確認します。

IPアドレス、ログインパスワード、時計などを適宜設定します。

設定を変更したらOSを再起動します。

## 4、リレーの設定確認

(1)再びブラウザで制御盤にアクセスし、ログインします。



(2)セットアップ→アクチュエータ設定に入る。

(3)上の方に設定選択という欄があるので、ここからスイッチ1、スイッチ2、スイッチ3、スイッチ4を選んで図の初期値になっていることを確認します。

出荷時の設定では(a)欄は楕円の箇所がスイッチごとに少し違いますが、(b)欄は全て同じ空欄になっています。

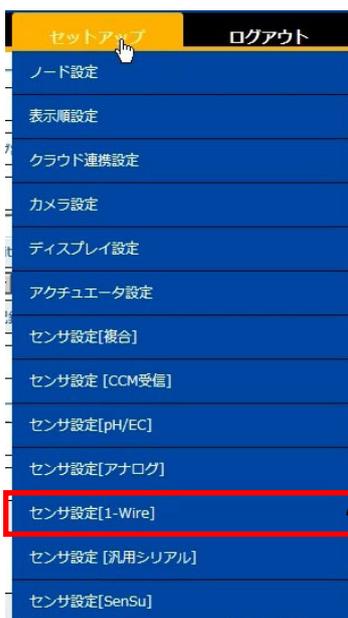
設定を書き換えた場合は、保存ボタンを押します。  
最後にノード再起動ボタンを押すことで設定したとおりに動作するようになります。

※この設定が異なると意図した動作をしないことがあります。

## 5、温度センサの設定変更



(1) 付属の1-wire温度センサの利用設定を行います。



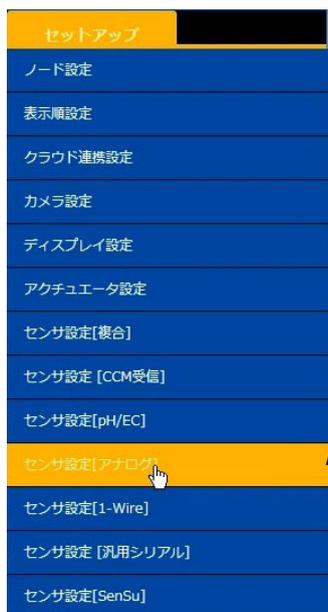
(2) PCからブラウザで制御盤にログインし、セットアップ→センサ設定(1-wire)に入ります。

トップ	詳細モニタ	CCM一覧	状態ログ	セットアップ	ログアウト
センサ設定[1-Wire]					
設定選択	Maixm DS18B20				
センサ機種	Maixm DS18B20				
表示名	MAXIM_DS18B20				
1-Wireアドレス	28C9D38309000D8				
計測間隔	1 秒				
計測値選択	温度 <input type="radio"/> 無効 <input checked="" type="radio"/> 有効				
表示名	Temperature				
CCM設定	項目名: InAirTemp ノード種別: cMC 送信レベル: A-1S-0 単位: C 精度: 1 [room:1] -[region:1] -[order:1] (priority:1)				
検出設定	検出方式: 瞬時値 検出時間: 秒				
補正式(y=ax <sup>2</sup> +bx+c)	x <sup>2</sup> + x +				
制限範囲	下限: ~ 上限: 範囲外: <選択>				
記録間隔	300 秒 <input type="checkbox"/> クラウド連携				
<input type="button" value="ノード再起動"/> <input type="button" value="削除"/> <input type="button" value="保存"/>					

(3) 図の赤枠の部分の設定を変更します。(1-Wireアドレスは違う文字が表示される場合があります)

設定を書き換えた場合は、保存ボタン、ノード再起動ボタンの順に押し設定を反映します。

## 6、アナログ電圧入力の設定変更



(1)PCからブラウザで制御盤にログインし、セットアップ→センサー設定(アナログ)に入ります。

センサー設定[アナログ]

有効CH	<input checked="" type="checkbox"/> AI1 <input checked="" type="checkbox"/> AI2
CCM表示名(AI1)	ADC-CH1
CCM設定(AI1)	項目名(AI1): Volt.1    ノード種別(AI1): cMC 送信レベル(AI1): A-1S-0    単位(AI1): V    精度(AI1): 3 <small>[room:1] -[region:1] -[order:1] (priority:1)</small>
A/D変換設定(AI1)	値タイプ: 電圧値(V)    分解能: 18 bit    ゲイン: x1
計測設定(AI1)	計測間隔: 1 秒    検出方式: 瞬時値    検出時間: 秒
変換式(AI1)	<input checked="" type="radio"/> なし <input type="radio"/> 二次式 <input type="radio"/> サーミスタ式
制限範囲(AI1)	下限(AI1):    ~ 上限(AI1):    範囲外(AI1): <選択>
記録間隔(AI1)	300 秒 <input type="checkbox"/> クラウド連携
CCM表示名(AI2)	ADC-CH2
CCM設定(AI2)	項目名(AI2): Volt.2    ノード種別(AI2): cMC 送信レベル(AI2): A-1S-0    単位(AI2): V    精度(AI2): 3 <small>[room:1] -[region:1] -[order:1] (priority:1)</small>
A/D変換設定(AI2)	値タイプ: 電圧値(V)    分解能: 18 bit    ゲイン: x1
計測設定(AI2)	計測間隔: 1 秒    検出方式: 瞬時値    検出時間: 秒
変換式(AI2)	<input checked="" type="radio"/> なし <input type="radio"/> 二次式 <input type="radio"/> サーミスタ式
制限範囲(AI2)	下限(AI2):    ~ 上限(AI2):    範囲外(AI2): <選択>
記録間隔(AI2)	300 秒 <input type="checkbox"/> クラウド連携

(2)図の赤枠の部分を書き換えます。(項目名(AI1)の部分を書き換える)に項目名(AI2)の部分は"Volt.2"と入力する)

設定を書き換えた場合は、保存ボタン、ノード再起動ボタンの順に押し設定を反映します。

## 7、制御モードの変更

(1) 「トップ」に戻ります。

UECS-Pi Uni

トップ 詳細モニタ CCM一覧 状態ログ セットアップ ログアウト

トップ

ノード状態

IPアドレス	現在時刻	日の出	南中	日の入り	状態
192.168.1.70	2019-01-10 09:23:37 [+09:00]	07:09	12:08	17:06	正常

センサ

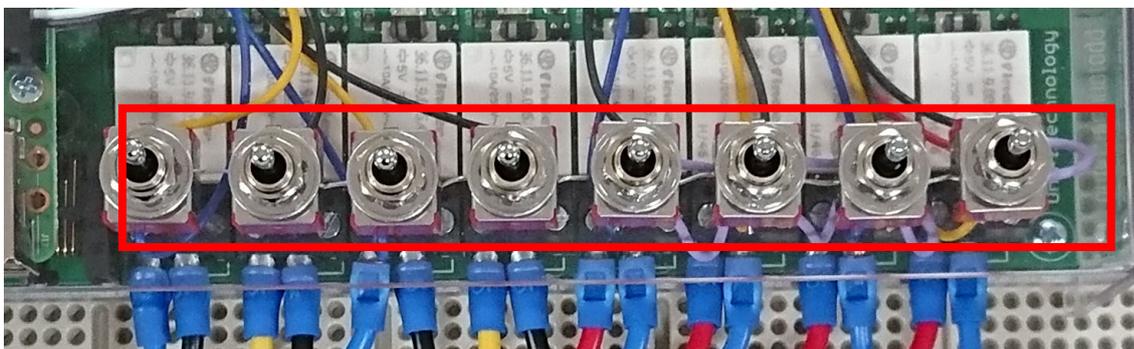
No.	名称	現在値	更新日時	設定
1	<a href="#">ADC-CH1</a>	0.000 [V]	2019-01-10 09:23:30	⚙️
2	<a href="#">ADC-CH2</a>	0.001 [V]	2019-01-10 09:23:30	⚙️

アクチュエータ

No.	名称	モード	操作	現在値	更新日時	設定
1	<a href="#">スイッチ1</a>	自律	AUTO	OFF	2019-01-10 09:24:07	⚙️
2	<a href="#">スイッチ2</a>	自律	AUTO	OFF	2019-01-10 09:24:08	⚙️
3	<a href="#">スイッチ3</a>	自律	AUTO	OFF	2019-01-10 09:24:10	⚙️
4	<a href="#">スイッチ4</a>	自律	AUTO	OFF	2019-01-10 09:24:11	⚙️
5	<a href="#">ポジション1</a>	WEB強制操作	0[%]	0[%]	2019-01-10 09:22:30	⚙️
6	<a href="#">ポジション2</a>	WEB強制操作	0[%]	0[%]	2019-01-10 09:22:31	⚙️

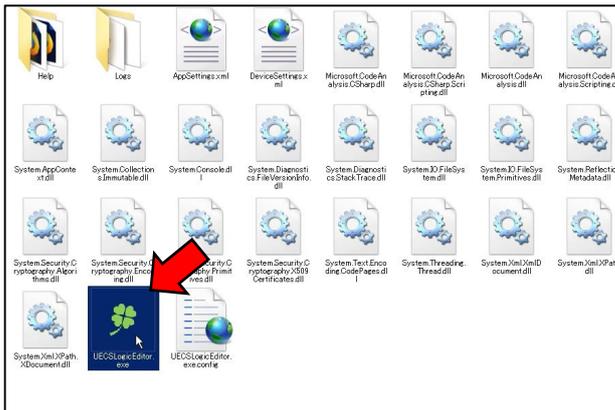
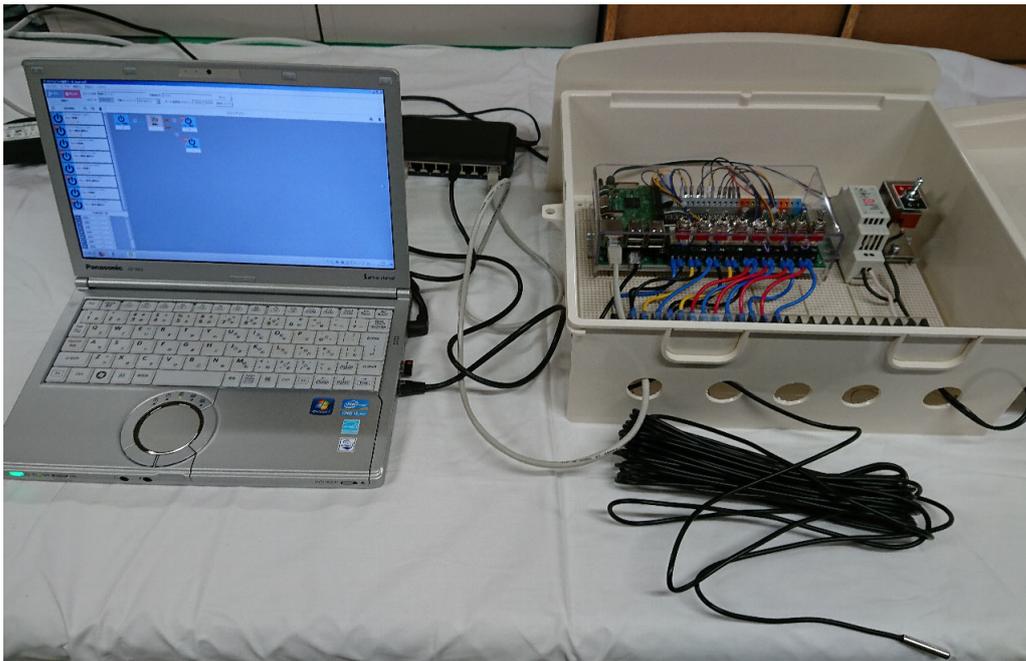
更新

(2) 図の「操作」のところでスイッチ1～4を全て“AUTO”に変更して更新ボタンを押します。その後、必ずブラウザを閉じます。

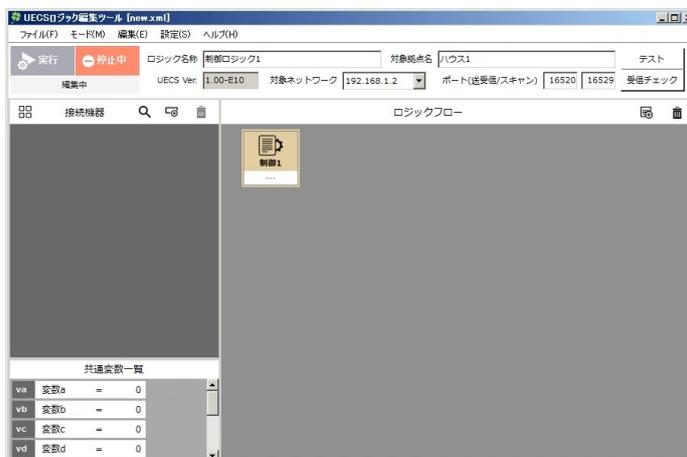


(3) 制御盤の3点切り替えスイッチを全て中央に入れます。

## 8、ロジック開発ツールの起動



(1)PCのロジック開発ツールがインストールされたフォルダの中の四葉マークをダブルクリックします。



(2)最初に左のような画面が表示されます。

## 9、機器の登録



(1)接続機器の自動探索ボタンを押します。

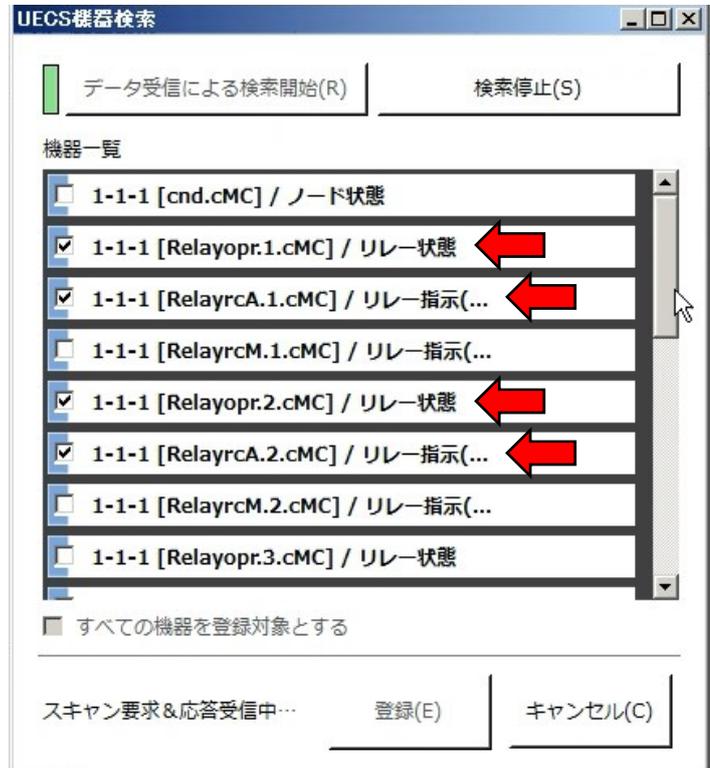
(2)検索画面が出るのでスキャンによる検索開始(S)を押します。

(3)右のように機器が沢山検出されるので、スクロールしつつ

Relayopr.1.cMC	リレー状態
RelayrcA.1.cMC	リレー指示
Relayopr.2.cMC	リレー状態
RelayrcA.2.cMC	リレー指示
Volt.1.cMC	電圧
Volt.2.cMC	電圧
InAirTamp.cMC	温室内気温

をチェックします。

(この後の実験に使う項目だけ選んでいます)



(4)検索停止ボタンを押したあと登録ボタンを押します。



UECSでは一つのリレー（あるいは制御対象）に対して指令コマンドが2つ出てきます。普段は"rcA"という文字が入っている方を使います。

UECSの命名規則では一つの機械の中に同じ機能が複数ある場合は、"?????.1.xxx"、"?????.2.xxx"、"?????.3.xxx"のように番号で表します。例えばRelayopr.1.cMCはリレー1の状態、Relayopr.2.cMCはリレー2の状態を示します。

# 10、機器の登録

接続機器リストが登録されましたが、同じ名前が複数のアイコンに付いてしまっています。

このままだと分かりにくいので名前をつけ直します。

以下は表示名の変更例です。

CCM識別子	表示名
Relayopr.1.cMC	リレー状態1
RelayrcA.1.cMC	リレー指示1
Relayopr.2.cMC	リレー状態2
RelayrcA.2.cMC	リレー指示2
Volt.1.cMC	電圧1
Volt.2.cMC	電圧2

ここまで作ったら最後にファイルに名前をつけて保存してください。

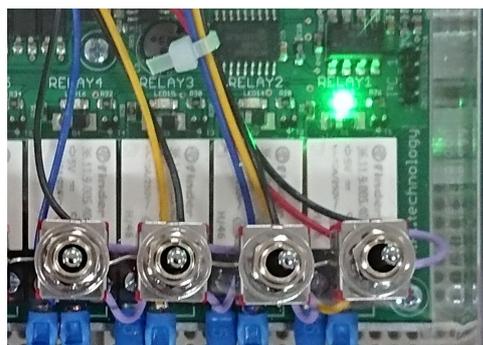
# 1 1、動作テスト



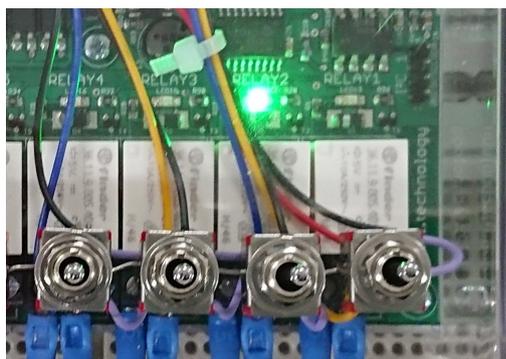
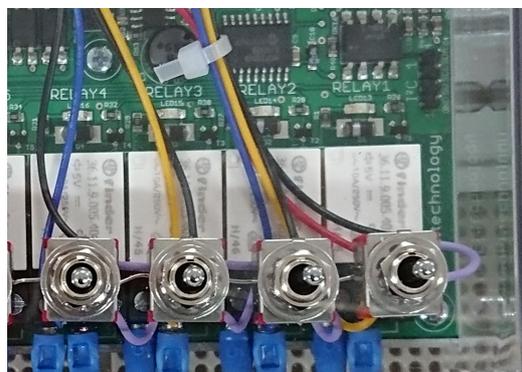
(1)接続機器リストの中のリレー指示1または2をダブルクリックして設定画面を出します。



(2)リレー指示1のテスト送信データに1を入力してテスト送信を押すと1番目のリレーがONになります。



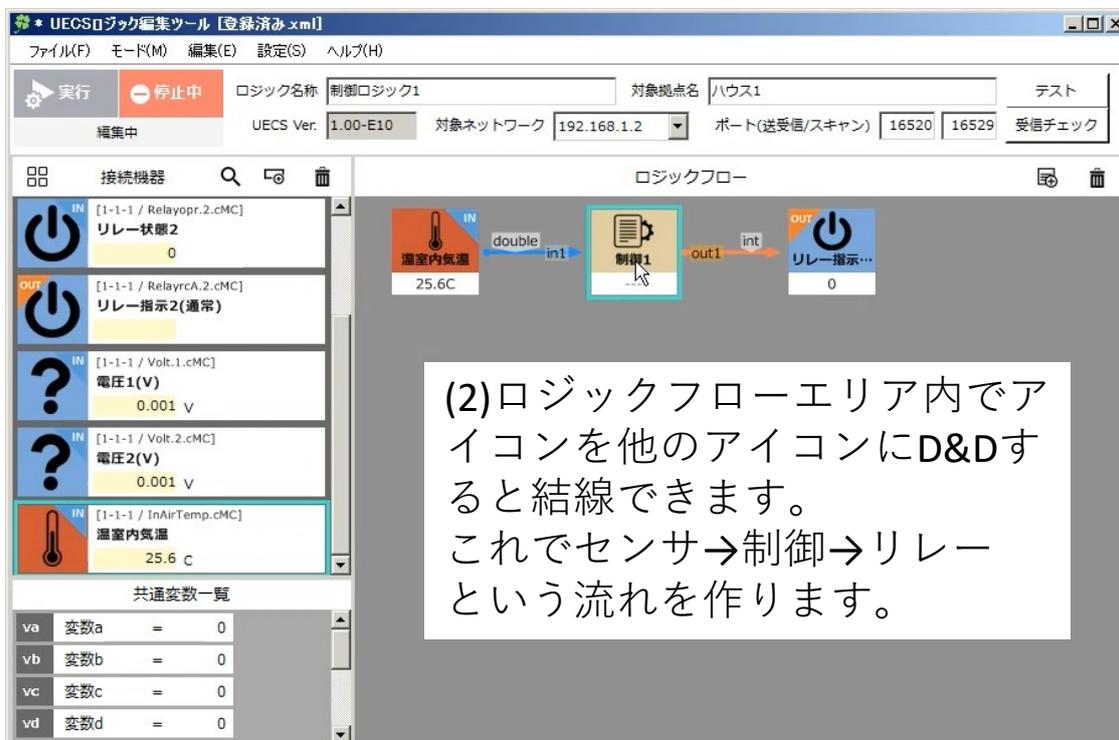
(3)テスト送信データに0を入力してテスト送信を押すとリレーがOFFになります。



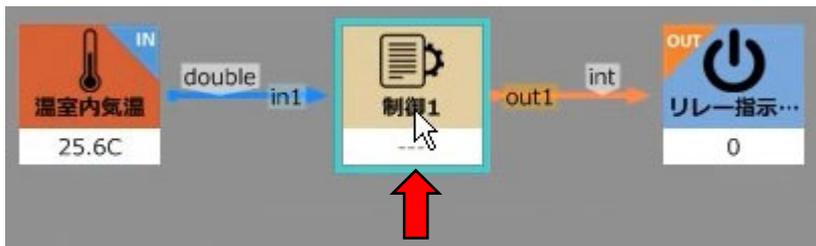
(4)リレー指示2の方からは2番目のリレーを制御できます。

## 第2節 単純なサーモスタットを作る

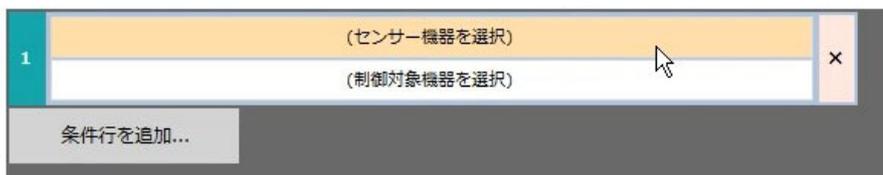
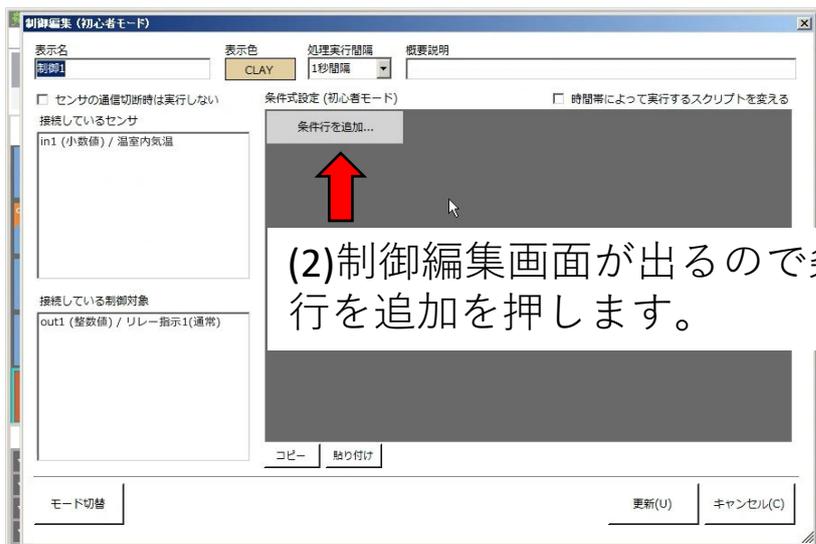
# 1、フローの配置



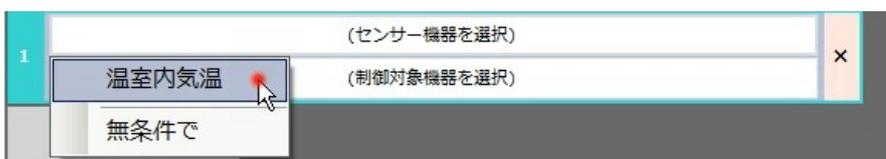
## 2、制御内容の作成（1）



(1)黄色の制御アイコンをダブルクリックします。



(3)「センサー機器を選択」



(4)「室内気温」



(5)欄に25を入力  
「より小さい」とき

### 3、制御内容の作成（2）

1	温室内気温	が	固定値	25	より小さい	とき	x
(制御対象機器を選択)							

(1) 「制御対象機器を選択」

1	温室内気温	が	固定値	25	より小さい	とき	x
(制御対象機器を選択)							
リレー指示1(通常)							
かつ							
または							

(2) 「リレー指示1」

1	温室内気温	が	固定値	25	より小さい	とき	x
(制御対象機器を選択)							
リレー指示1(通常)							
に				固定値	1	を送信	

(3) 欄に1を入力  
「固定値1を送信」

1	温室内気温	が	固定値	25	より小さい	とき	x
(制御対象機器を選択)							
リレー指示1(通常)							
に				固定値	1	を送信	

(4) 「条件行を追加」

1	温室内気温	が	固定値	25	より小さい	とき	x	
(制御対象機器を選択)								
リレー指示1(通常)								
に				固定値	1	を送信		
2	(センサー機器を選択)							x
(制御対象機器を選択)								

(5) 「センサー機器を選択」

1	温室内気温	が	固定値	25	より小さい	とき	x	
(制御対象機器を選択)								
リレー指示1(通常)								
に				固定値	1	を送信		
2	(センサー機器を選択)							x
(制御対象機器を選択)								
無条件で								

(6) 「無条件で」

1	温室内気温	が	固定値	25	より小さい	とき	x	
(制御対象機器を選択)								
リレー指示1(通常)								
に				固定値	1	を送信		
2	無条件で							x
(制御対象機器を選択)								

(7) 「制御対象を選択」

1	温室内気温	が	固定値	25	より小さい	とき	x	
(制御対象機器を選択)								
リレー指示1(通常)								
に				固定値	1	を送信		
2	無条件で							x
(制御対象機器を選択)								
リレー指示1(通常)								
かつ								
または								

(8) 「リレー指示1」

1	温室内気温	が	固定値	25	より小さい	とき	x	
(制御対象機器を選択)								
リレー指示1(通常)								
に				固定値	0	を送信		
2	無条件で							x
(制御対象機器を選択)								

(9) 欄に0を入力  
「固定値0を送信」

## 4、完成したスクリプト



最後に更新ボタンを押します。

この操作で「25°C未満でリレーON」「それ以外の場合はリレーOFF」という制御が完成しました。

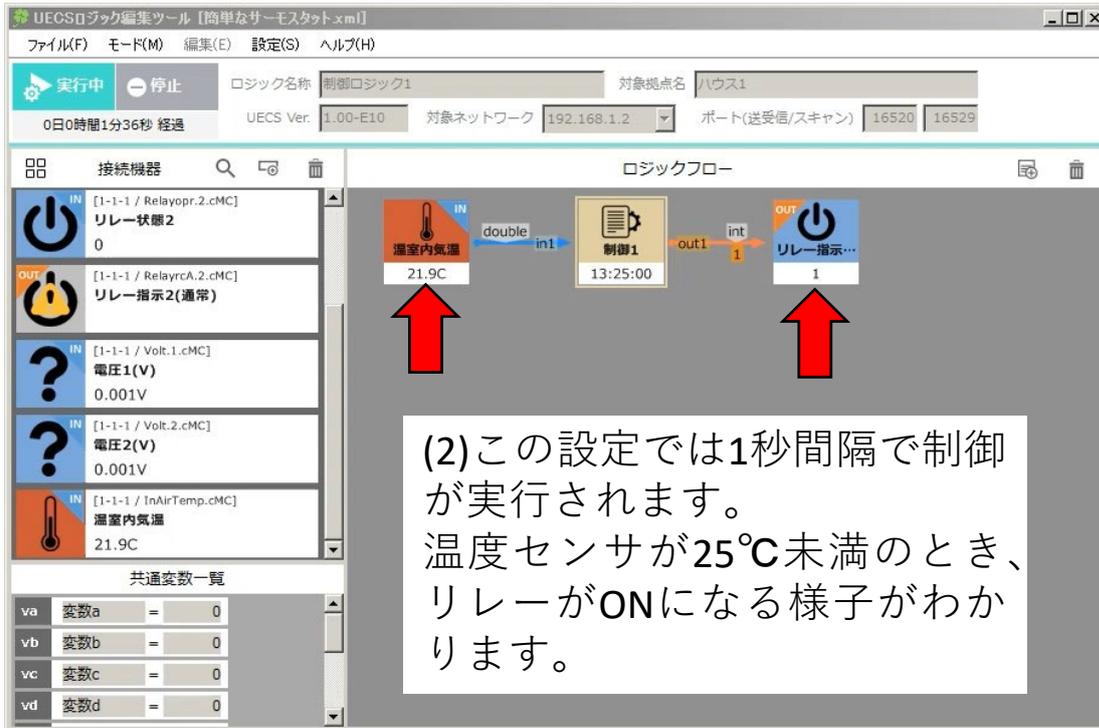
リレーに温度を上げる装置を接続しておけば、温度を25°Cに保つサーモスタットとして機能します。

## 5、制御の実行

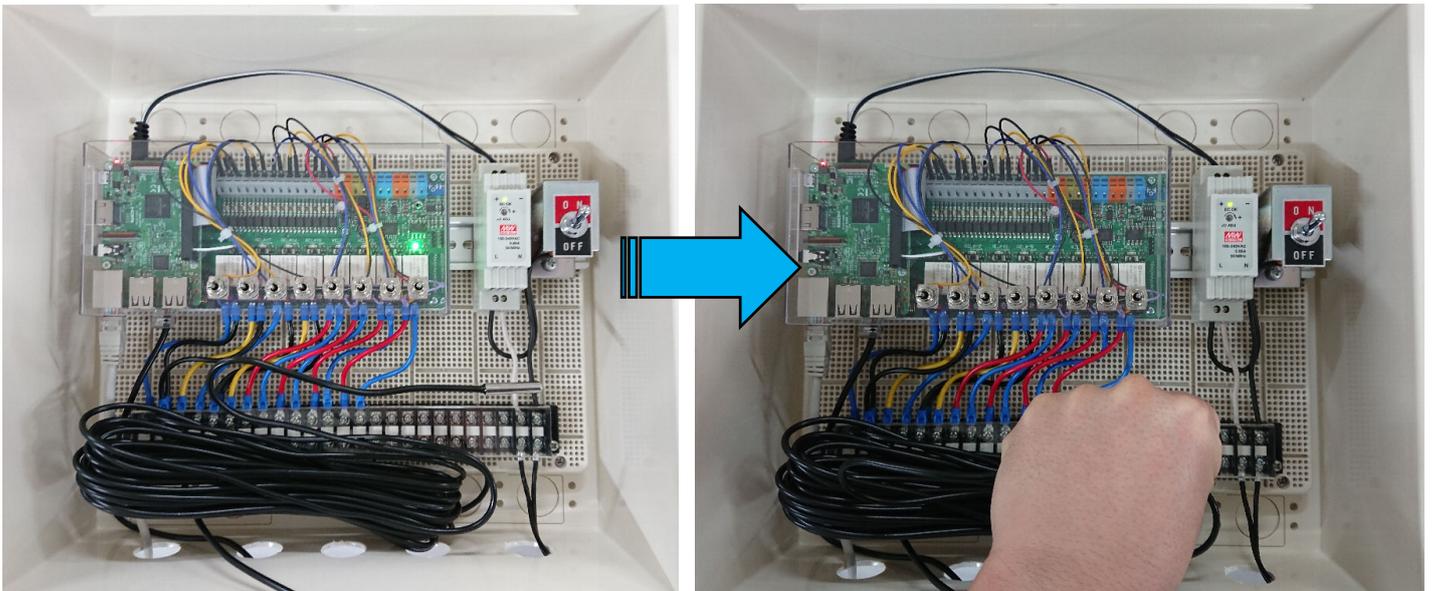


(1)実行ボタンを押して制御を実行してみます。

※実行前に作成したデータに名前をつけて保存してください。



(2)この設定では1秒間隔で制御が実行されます。  
温度センサが25°C未満のとき、リレーがONになる様子わかります。

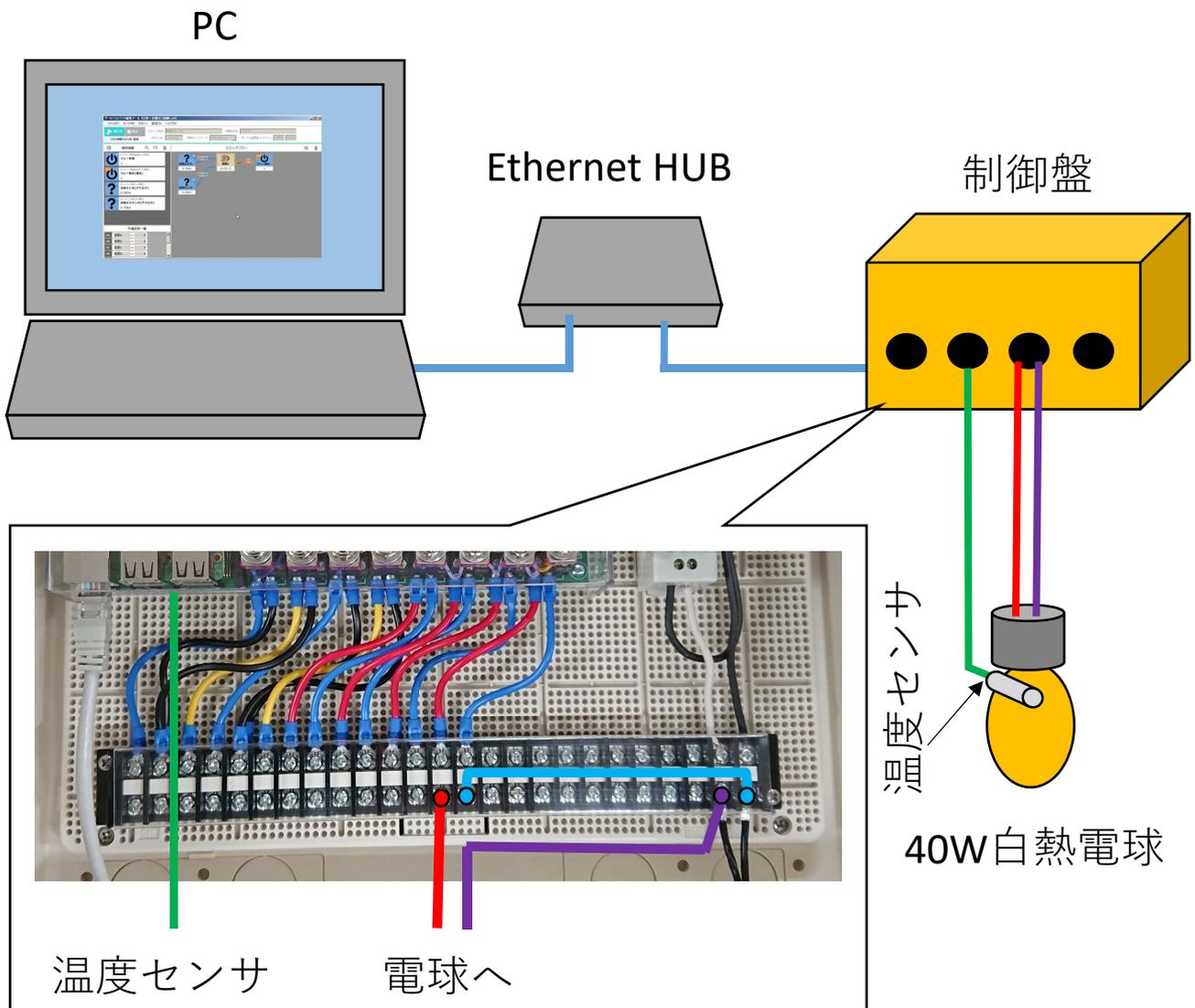


(3)試しに温度センサを握って温度を上げると25°Cを超えたところでリレーがOFFになるのが分かります。

### 第3節 サーモスタットの改良

# 1、電球を使ったサーモスタットの実験

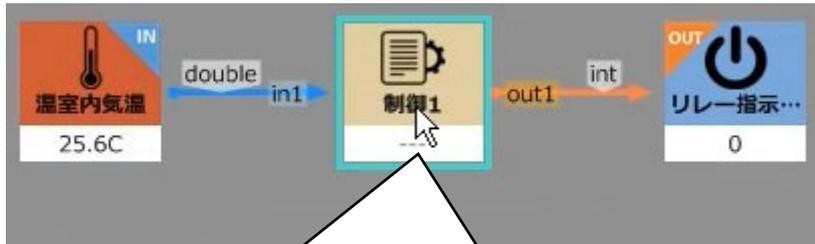
ここでは白熱電球を熱源としてサーモスタットの試験を行います。



白熱電球にはフィラメント付近に1-Wire温度センサ先端のケーブルを3周巻き付けた後、針金で縛って密着させます。

安全のため40W以下の電球以外使わないでください。  
電球は浮かせてください。  
口金に金属を接触させないでください。

## 2、単純なサーモスタットの弱点



1	温室気温	が	固定値	<input type="text" value="40"/>	より小さい	とき	×
	リレー指示1(通常)	に	固定値	<input type="text" value="1"/>	を送信		
2	無条件で						×
	リレー指示1(通常)	に	固定値	<input type="text" value="0"/>	を送信		

条件行を追加...

第2節で作った制御の設定温度を書き換えて40°Cにしました。



40°C未満



40°C以上

この制御を実行すると、白熱電球が点いたり消えたりしながら、温度を40°C付近に保つように制御されます。

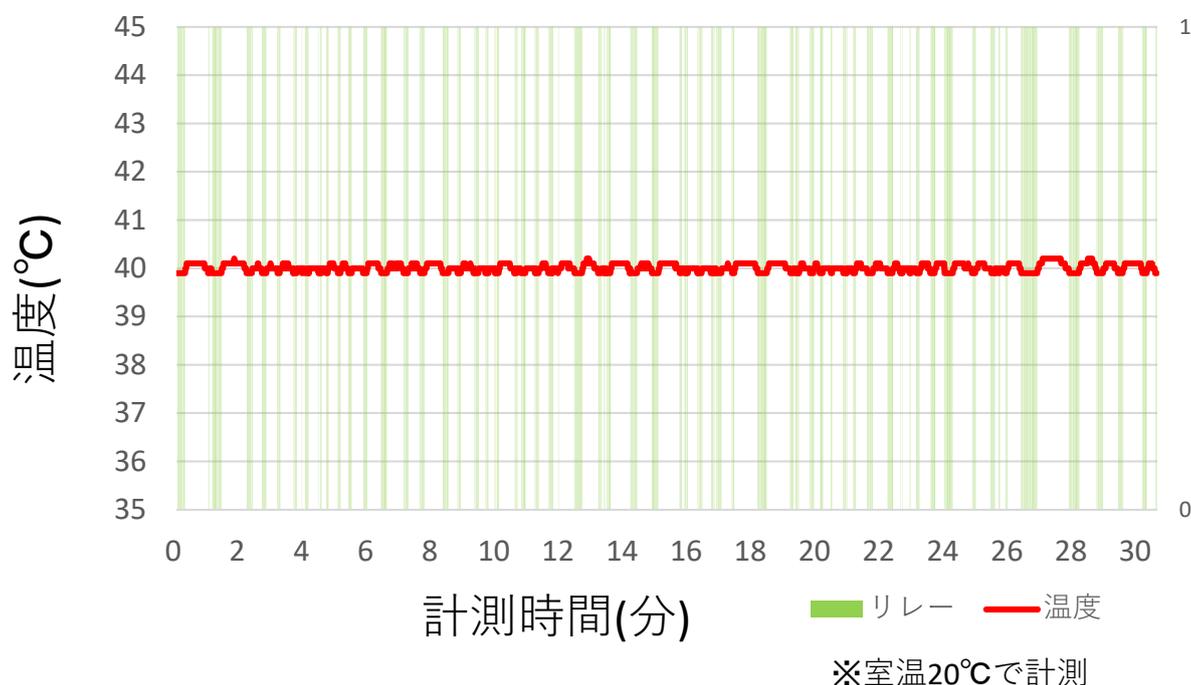
しかし、この方法にはON/OFFの回数が非常に多くなりやすいという弱点があります。

### 3、単純なサーモスタットの弱点

秒単位でリレーの挙動と温度変化を記録したものが以下のグラフになります。

グラフが潰れてしまって全てを表示できないが、

... 30分間に86回も点滅していた！



何度もON/OFFすると制御の精度は上がりますが、機器の寿命が縮むおそれがあります。

そのため、過剰なON/OFFを防ぎ、機器の消耗と制御の正確さのバランスを取る必要があります。

## 4、実行間隔の調整

AY 処理実行間隔 1秒間隔

概要説明

条件式設定 (初心者モード)  時間帯によって実行するスクリプトを変える

1	温室内気温	が	固定値	40	より小さい	とき	X
	リレー指示1(通常)	に	固定値	1	を送信		
無条件で							
2	リレー指示1(通常)	に	固定値	0	を送信		X

条件行を追加...

AY 処理実行間隔 10秒間隔

概要説明

条件式設定 (初心者モード)  時間帯によって実行するスクリプトを変える

1	温室内気温	が	固定値	40	より小さい	とき	X
	リレー指示1(通常)	に	固定値	1	を送信		
無条件で							
2	リレー指示1(通常)	に	固定値	0	を送信		X

条件行を追加...

AY 処理実行間隔 60秒間隔

概要説明

条件式設定 (初心者モード)  時間帯によって実行するスクリプトを変える

1	温室内気温	が	固定値	40	より小さい	とき	X
	リレー指示1(通常)	に	固定値	1	を送信		
無条件で							
2	リレー指示1(通常)	に	固定値	0	を送信		X

条件行を追加...

過剰なON/OFFを抑制する手段には制御の実行間隔を変更する方法があります。ただし、間隔が出力先の有効期限を超えてはいけません。(図では60秒間隔に設定してOK)

機器タイプ

センサ

制御対象

指示タイプ

60秒間隔(180秒有効)

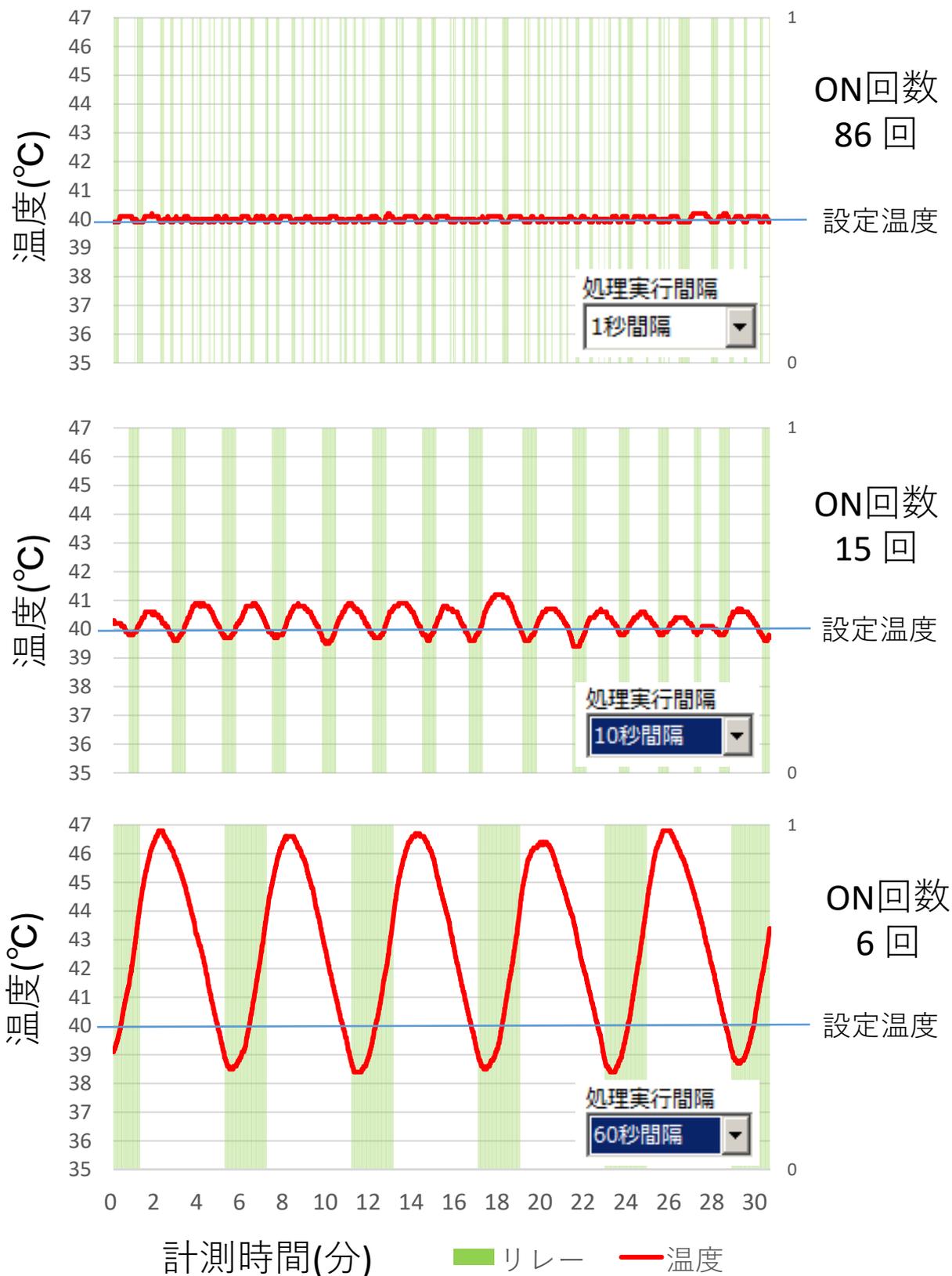
テスト送信アーダ

1

テスト送信



## 5、実行間隔の調整結果（30分間のデータ）



スクリプトの実行間隔を10秒、60秒にするとリレーのON回数は大きく減らすことができました。しかし、実行間隔を伸ばすと、温度を一定に保つ精度が落ちます。60秒間隔では設定温度を大きく超えてしまい、実用には問題があることが分かります。

## 6、不感帯付きのサーモスタット

処理実行間隔 概要説明  
 AY 1秒間隔

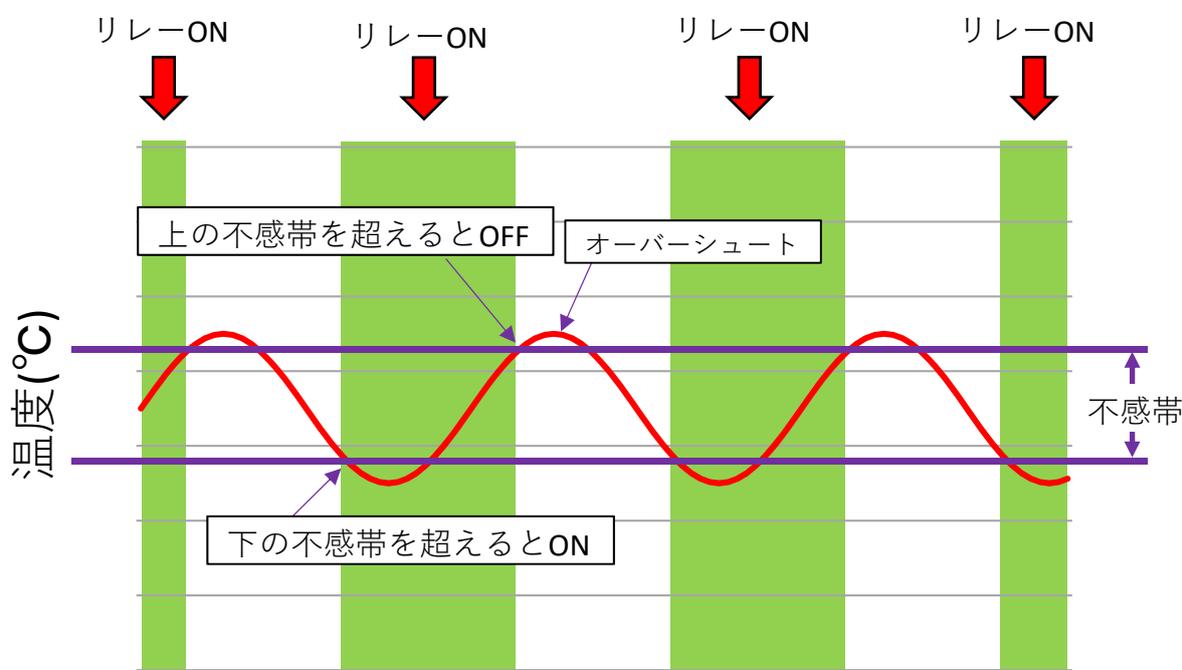
条件式設定 (初心者モード)  時間帯によって実行するスクリプトを変える

1	温室内気温	が	固定値	39.5	より小さい	とき	×
	リレー指示1(通常)	に	固定値	1	を送信		
2	温室内気温	が	固定値	40.5	より大きい	とき	×
	リレー指示1(通常)	に	固定値	0	を送信		
3	無条件で						×
	リレー指示1(通常)	に	過去値	リレー指示1(通常)	の前回送信値を送信		

条件行を追加...

サーモスタットの動作を改良しました。「不感帯」を設けることで精度とON/OFF回数のバランスを取ります。図では39.5～40.5°Cの間を温度が往復するような制御を行います。この時、39.5～40.5°Cの範囲は制御を行わない遊びの領域、不感帯になります。

これを例えば39～41°Cの範囲に変更すれば温度制御の精度は落ちますがON/OFFの回数は減らせます。誤差の許容範囲、許されるON/OFF頻度など、装置の性質や制御の目的に合わせて不感帯の幅を変えることで対応することができます。

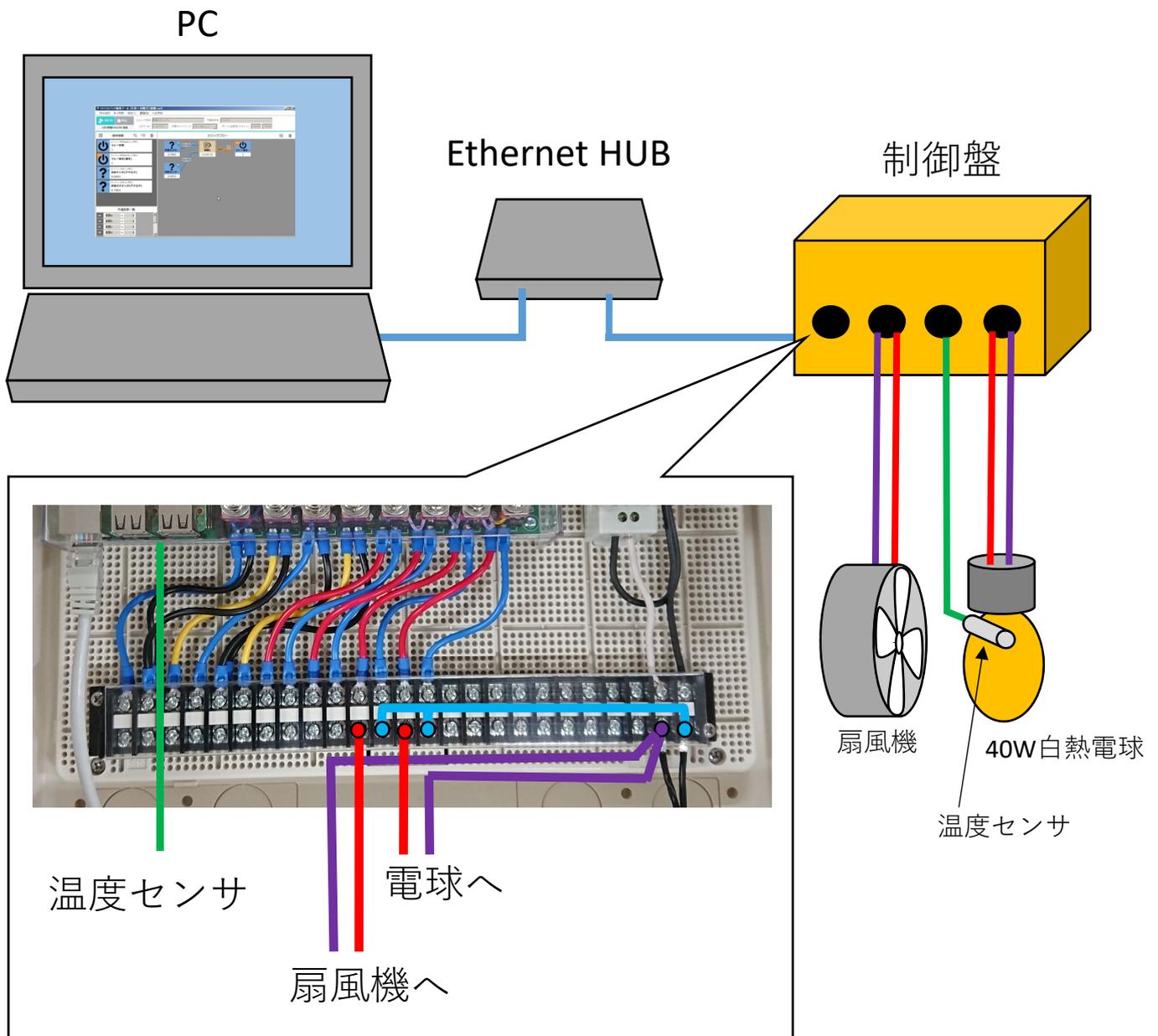


### 不感帯のイメージ

- ※センサの反応には遅れがあるため、不感帯の範囲に収まらずオーバーシュートするのが一般的です。
- ※「不感帯」を「中立帯」「ヒステリシス」と呼ぶこともあります。

## 7、2つのリレーを使った制御（1）

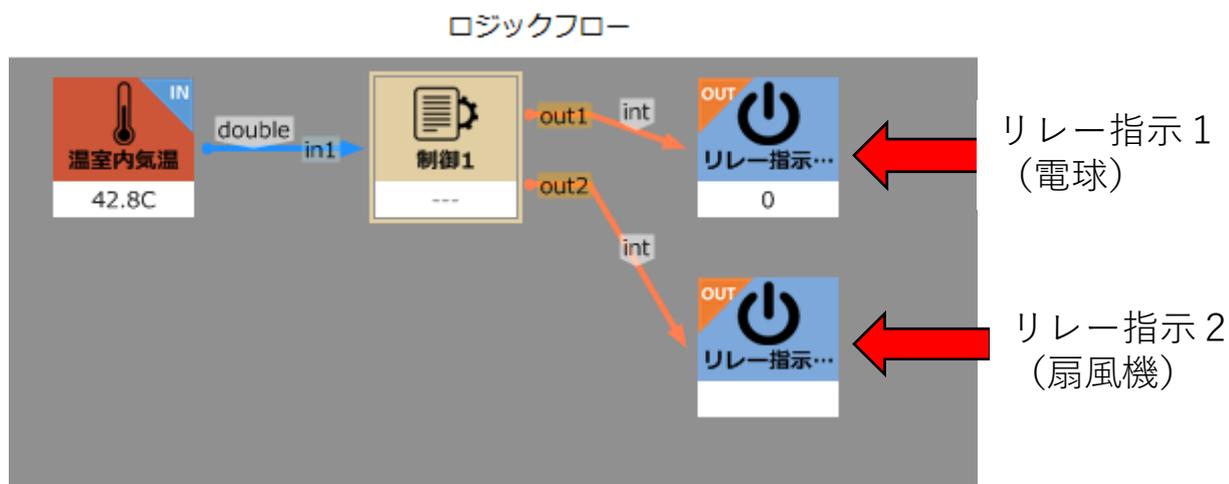
ここで、2つ目のリレーに扇風機を追加してみます。



温度が低いときは電球を点灯し、温度が上がり過ぎたときは扇風機を稼働させてみます。(扇風機は機械式のスイッチを持ち、通電すればすぐに起動するものを使います)

温度を上げる装置を制御して温度を一定に保つものをサーモスタットと呼びますが、ここでの扇風機のように温度を下げる装置を制御するものを逆サーモと呼びます。

## 8、2つのリレーを使った制御（2）



図のように制御1にリレー1とリレー2の2つの出力を接続します。

制御編集 (初心者モード)

表示名: 制御1    表示色: CLAY    処理実行間隔: 1秒間隔    概要説明:

センサの通信切断時は実行しない     時間帯によって実行するスクリプトを変える

接続しているセンサ: in1 (小数値) / 温室内外気温

接続している制御対象: out1 (整数値) / リレー指示1(通常)    out2 (整数値) / リレー指示2(通常)

条件式設定 (初心者モード)

1	温室内外気温	が	固定値	39.5	より小さい	とき	(a)
	リレー指示1(通常)	に	固定値	1	を送信		
2	温室内外気温	が	固定値	40.5	より大きい	とき	x
	リレー指示1(通常)	に	固定値	0	を送信		
3	無条件で						x
	リレー指示1(通常)	に	過去値	リレー指示1(通常)	の前回送信値を送信		
4	温室内外気温	が	固定値	41.5	より大きい	とき	(b)
	リレー指示2(通常)	に	固定値	1	を送信		
5	温室内外気温	が	固定値	40.5	より小さい	とき	x
	リレー指示2(通常)	に	固定値	0	を送信		
6	無条件で						x
	リレー指示2(通常)	に	過去値	リレー指示2(通常)	の前回送信値を送信		

条件行を追加...

コピー    貼り付け

モード切替    更新(U)    キャンセル(C)

制御1の中身です。(a)が電球で(b)が扇風機の制御を行うロジックです。リレーのON条件、OFF条件の大小関係が違うことに注意してください。このように複数のリレーを制御することも可能です。

## 付録 サンプルプログラム

## 1、電球を点滅させる



処理実行間隔 概要説明

AY 1秒間隔

制御

スクリプト編集 (通常モード)  時間帯によって実行するスクリプト

```
if((count % 2)==0)
{
out1=1;
}
else
{
out1=0;
}
```

特殊変数countと剰余演算子の併用で実行回数に応じて動作を変えられます。制御アイコンは出力だけつないでも作動します。

## 2、電球を3分間だけONにする



処理実行間隔 概要説明

AY 1秒間隔

制御

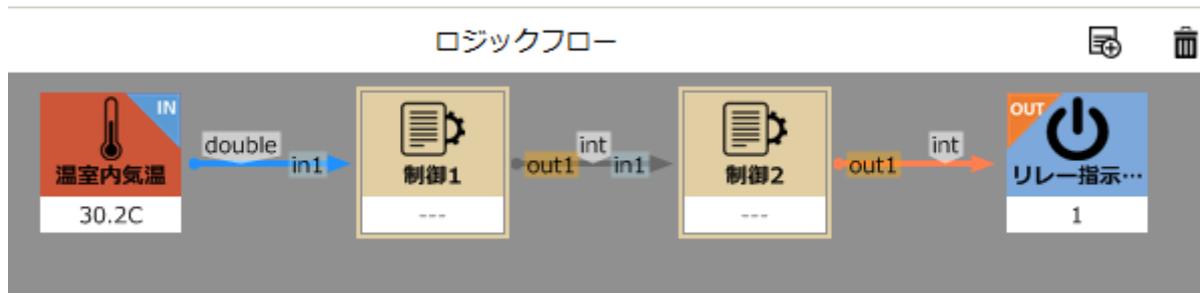
スクリプト編集 (通常モード)  時間帯によって実行するスクリプト

```
if(count==0)//起動時に1回だけ実行される
{va=0;}//カウンタを0にする

if(va<180)
{
out1=1; //ON
va++; //カウンタ加算
}
else
{out1=0;}//OFF
```

countが0になるのは起動直後なのでこれをトリガーにしてカウンタを初期化します。共通変数(va~vj)を使うとスクリプト終了後も値が残ります。

### 3、サーモスタットの改良—ON時間を3分以上確保する



処理実行間隔 概要説明

AY 1秒間隔

### 制御 1

条件式設定 (初心者モード)  時間帯によって実行するスクリプトを変える

1	温室気温	が	固定値	39.5	より小さい	とき	×
	リレー指示1(通常)	に	固定値	1	を送信		
2	温室気温	が	固定値	40.5	より大きい	とき	×
	リレー指示1(通常)	に	固定値	0	を送信		
3	無条件で						×
	リレー指示1(通常)	に	過去値	リレー指示1(通常)	の前回送信値を送信		

条件行を追加...

処理実行間隔 概要説明

AY 1秒間隔

### 制御 2

スクリプト編集 (通常モード)  時間帯によって実行するスクリプトを変える

```

//vaは消えない特殊な変数です
//一度vaが1になるとリレーをON、1秒間隔でvaをカウントアップし、
//180秒経過するまでONを継続します

int ONTIME=180; //必ずONする時間(3分=180秒)
if(count==0)//起動時に1回だけカウンタリセット
    {va=0;}

if(in1==1 || va>0)//リレーがONまたはvaがカウント中の場合、vaを加算する
    {va++;}
if(va>ONTIME) //カウンタvaが指定時間を超えた場合、それ以上カウントしない
    {va=ONTIME;}

if((va>0 && va<ONTIME) || in1==1)
    {out1=1;} //カウンタvaがカウント中かつ指定時間経過していない場合
else
    {out1=0;va=0;} //カウンタvaがカウント中かつ指定時間を超えた場合
  
```

制御 1 には普通のサーモスタットの動作を記述し、制御 2 に上級者モード (通常モード) で時間をカウントする処理を入れます。一度リレーがONになると、その後の気温に関わらず3分以上ONが継続します。

#### 4、サーモスタットの改良—OFF時間を3分以上確保する



処理実行間隔		概要説明	
AY	1秒間隔	制御 1	
条件式設定 (初心者モード) <input type="checkbox"/> 時間帯によって実行するスクリプトを変える			
1	温室気温	が	固定値 39.5 より小さい とき ×
	リレー指示1(通常)	に	固定値 1 を送信
2	温室気温	が	固定値 40.5 より大きい とき ×
	リレー指示1(通常)	に	固定値 0 を送信
3	無条件で		×
	リレー指示1(通常)	に	過去値 リレー指示1(通常) の前回送信値を送信
条件行を追加...			

処理実行間隔		概要説明	
AY	1秒間隔	制御 2	
スクリプト編集 (通常モード) <input type="checkbox"/> 時間帯によって実行するスクリプトを変える			
<pre> //vbは消えない特殊な変数です //一度vbが1になるとリレーをOFF、1秒間隔でvbをカウントアップし、 //180秒経過するまでOFFを継続します  int OFFTIME=180; //必ずOFFする時間(3分=180秒) if(count==0)//起動時に1回だけカウンタリセット     {vb=0;}  if(in1==0    vb&gt;0)//リレーがOFFまたはvbがカウント中の場合、vbを加算する     {vb++;} if(vb&gt;OFFTIME) //カウンタvbが指定時間を超えた場合、それ以上カウントしない     {vb=OFFTIME;}  if((vb&gt;0 &amp;&amp; vb&lt;OFFTIME)    in1==0)     {out1=0;} //カウンタvbがカウント中かつ指定時間経過していない場合 else     {out1=1;vb=0;} //カウンタvbがカウント中かつ指定時間を超えた場合           </pre>			

制御 1 には普通のサーモスタットの動作を記述し、制御 2 に上級者モード (通常モード) で時間をカウントする処理を入れます。一度リレーがOFFになると、その後の気温に関わらず3分以上OFFが継続します。

UECS用ロジック開発ツール活用マニュアル2  
Arspout 8ch 制御盤を使った活用例  
2019年3月

編集・発行

国立研究開発法人農業・食品産業技術総合研究機構  
西日本農業研究センター企画部産学連携室  
〒721- 8514 広島県福山市西深津町6-12-1  
Tel. 084-923-5385 Fax. 084-923-5215

農研機構ホームページ

<http://www.naro.affrc.go.jp/>

本マニュアルに掲載した内容の一部は、  
生研支援センター  
「革新的技術開発・緊急展開事業」（うち地域戦略プロジェクト）  
「UECSプラットフォームで日本型施設園芸が生きるスマート農業の実現」の  
支援を受けて行っています。

