

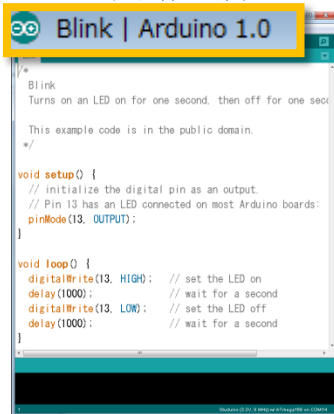
RTno の使い方・スケッチの作り方

(Win7-Visual Studio2013, Win10- Visual Studio2015 共通)

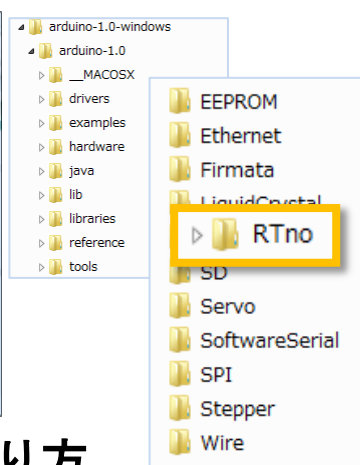
RTno の使い方

1. Stduino (Arduino) の基本的な動作確認する。
2. ArduinoIDE で RTno が使用できるようにする。
3. Stduino (Arduino) に、StarTno 用のスケッチをアップロードする。
4. デバイスマネージャで Stduino (Arduino) が接続されている COM ポート番号を調べる。
5. RTnoProxy.conf をメモ帳で開き、COM の番号に書きかえ、保存する。
6. RTnoProxyComp.exe を起動する。
7. RT システムエディターで動作確認を行う。

1. 基本動作の確認



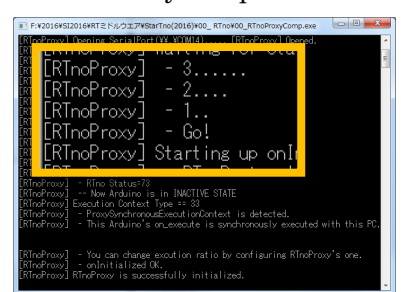
2. RTno のインストール



5. RTnoProxy.conf



6. RTnoProxyComp.exe



スケッチの作り方

Arduino の標準的な void setup(), void loop() という構造ではなく、 onInitialize(), onExecute() など、RTM の表記方法に準拠している。このスケッチを利用して、不要な部分をカットし、増加させたいポートを追加するなどして、スケッチを作成する。

```
#include <UART.h>
#include <RTno.h>
#include <Servo.h>

int LedVal_H;
int LedVal_L;
Servo Servo09;
int ServoVal09;

void rtconf(config_str& conf, exec_ext_str& exec_ext) {
  conf._default.connection_type = ConnectionTypeSerial1;
  conf._default.baudrate = 57600;
  exec_ext.periodic.type = ProxySynchronousExecutionContext;
}

TimedLong in13;
InPort<TimedLong> in13In("LED_13", in13);
TimedLong in09;
InPort<TimedLong> in09In("Servo_09", in09);
TimedLong outA0;
OutPort<TimedLong> outA0Out("Analog_A0", outA0);
```

int RTno::onInitialize() {

```
pinMode(13, OUTPUT);
pinMode(9, OUTPUT);
pinMode(A0, INPUT);
addInPort(in13In);
addInPort(in09In);
addOutPort(outA0Out);

digitalWrite(13, HIGH); // set the LED on
LedVal_H=1;
LedVal_L=19;
in13.data=1;

Servo09.attach(9);
ServoVal09 = 90;
return RTC_OK;
}
```

int RTno::onActivated() {

```
// Write here initialization code.

return RTC_OK;
```

int RTno::onDeactivated() {

```
// Write here finalization code.

return RTC_OK;
}
```

int RTno::onExecute() {

```
digitalWrite(13, HIGH); // set the LED on
delay(LedVal_H); // wait for a second
digitalWrite(13, LOW); // set the LED on
delay(LedVal_L); // wait for a second

if(in13In.isNew()) {
  in13In.read();
  if (in13.data>0) {
    LedVal_H=1000.0/in13.data;
    LedVal_L=1000.0/in13.data;
  }
  else if(in13.data<1) {
    LedVal_H=1;
    LedVal_L=10;
  }
}

if(in09In.isNew()) {
  in09In.read();
  ServoVal09 = in09.data;
  if (ServoVal09<0) ServoVal09=0;
  if (ServoVal09>180) ServoVal09=180;
  ServoVal09 = map(ServoVal09,0,180,0,180);

  Servo09.write(ServoVal09);
  delay(3);
}

outA0.data = analogRead(0);
outA0Out.write();

return RTC_OK;
}
```

int RTno::onError() {

```
return RTC_OK;
```

int RTno::onReset() {

```
return RTC_OK;
```

001 基本動作の確認(Blink, Knob)

Arduino IDE は、バージョンの古い 1.0 を使用する (新しいバージョンでは, RTno 用のスケッチをコンパイルするときにエラーになる). 基本動作を確認するため, LED 点滅させるスケッチ(File→Examples→1.Basics→Blink)をボードに書き込み, ボード上の LED が点滅することを確認する.

Studuino を使用する場合, Studuino 用の環境設定が必要である. Artec のホームページを参照して設定する. StarTno_00 では, Arduino の基本機能のみを使う. Studuino を持っていない場合, Arduino で代用できる.

LED の点滅により, PC とボードの接続が確認できた.

arduino-1.0-windows

arduino-1.0

Arduino1.0 をインストールし, Arduino.exe を起動する

Arduino-1.0 の起動画面

arduino.exe

Arduino-1.6 は, ×.

File

Examples

1.Basics

Blink

blinkを開く

書き込む前に, ターゲットボードの設定, COM ポートの指定が必要

ボードに書き込む

```

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000);            // wait for a second
  digitalWrite(13, LOW);  // set the LED off
  delay(1000);            // wait for a second
}

```

次に, サーボモータとボリュームの動作確認を行う. 電源供給したままの配線はトラブル原因になるため, PC と接続されている USB ケーブルを抜き, 配線する. サーボモータの信号線(白色)は 9 番ピンに, グランド線(黒色)は GND に, 電源供給線(赤色)は Vin に接続する. ボリュームの信号線(白色)は A0 に, グランド線(黒色)は GND に, 電源供給線(赤色)は 5V に接続する. スケッチは, Knob(File→Examples→Servo→Knob)を使用する. PC と Studuino(Arduino)を USB ケーブルで接続し, スケッチを書き込む.

配線する

電源

3.3V

5V

GND

GND

VIN

A0

信号線

AREF

GND はグラウンド電位の基準(0V)

LED の長い脚を 13 に

A0(スタディーノ)

GND 5V A0

Knob

```

// Controlling a servo position using
// by Michal Rinott <http://people.ir

#include <Servo.h>

Servo myservo; // create servo object

int potpin = 0; // analog pin used to
int val; // variable to read the value from the pot

void setup()
{
  myservo.attach(9); // attaches the servo on
}

void loop()
{
  val = analogRead(potpin);
  val = map(val, 0, 1023, 0, 179);
  myservo.write(val);
  delay(15);
}

```

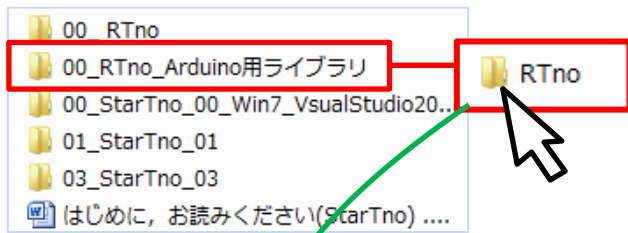
スタディーノの場合, ボリューム Sensor 用の 3 ピンコネクター A0 を使用できる. ただし, 3.3V である.

スタディーノを使用し, Servo 用の 3 ピンコネクター D9 に接続して使用する場合は, 電池が必要.

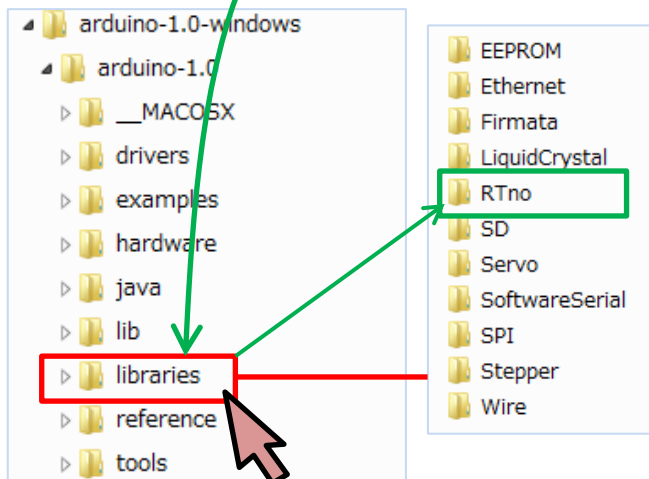
Knob 開き, ボードに書き込む. ボリューム回し, サーボモータ回転することを確認する.

ボリュームを回すと, サーボモータが回転する.

002 ArduinoIDE で RTno が使用できるようにする。



ダウンロードしたファイルの中の
RTno フォルダを
Arduino-1.0-windows の
libraries にコピー&ペーストする



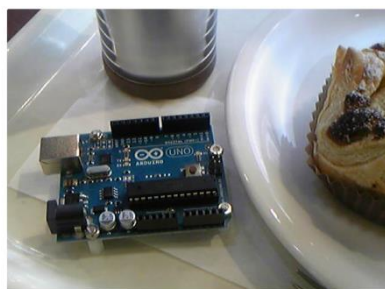
RTno の詳細については、菅氏の Web ページを参照して下さい

<http://ysuga.net/?p=124>

ysuga.net

Robot, inline-Skate, and more.

トップ このサイトについて ロボットスキルウェア ロボット ブログ インラインスケート DART



RTno (RT-middleware + arduino)

組み込みマイコン用RTC開発キット(RTno(アールディーノ))を開発しました。

ROBOMECH2011での発表資料:

発表(PDF)

発表(PDF)

2011での発表資料:

スライド(SlideShare)

ニュース:

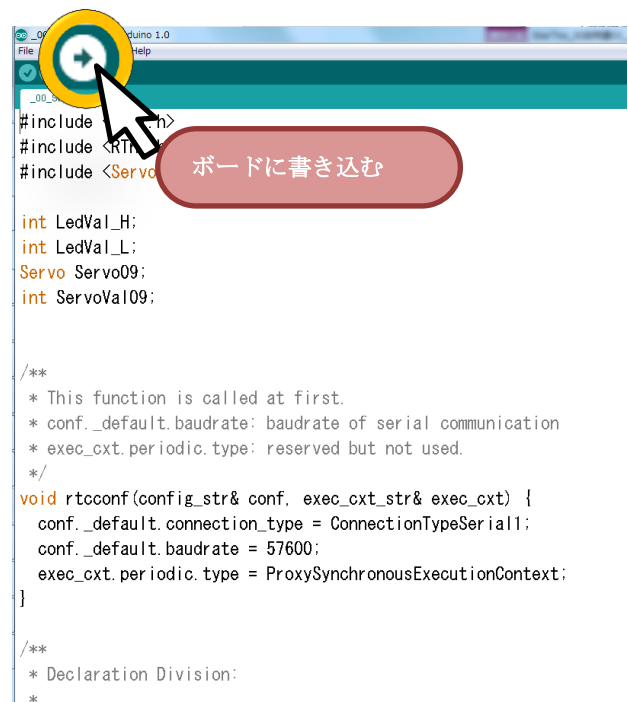
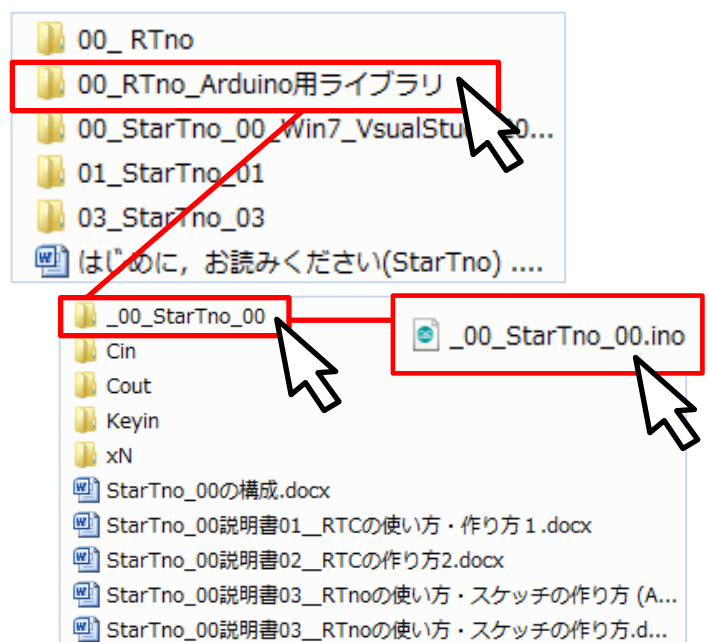
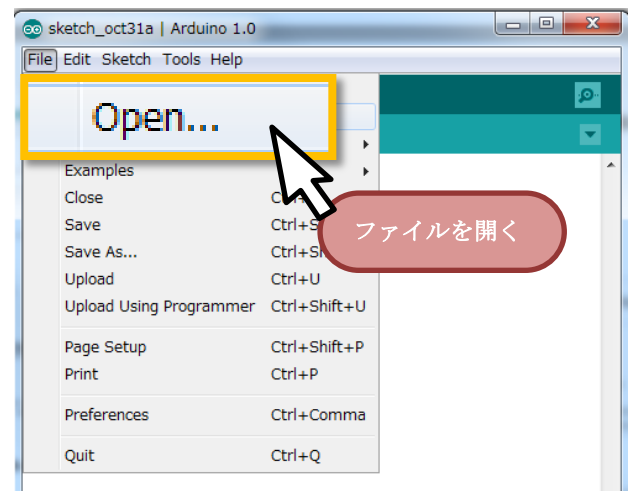
RTno Release 4.0をリリースしました。Ethernetへの対応が滞っていますが、プロトコル仕様を見直し、プロキシRTCを改良して通信の安定性が向上しました。mbedではRTnoV4で開発してください。

言語:
日本語
English

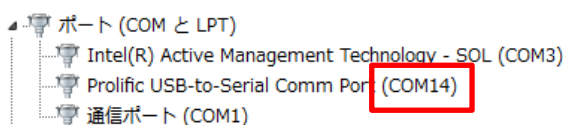
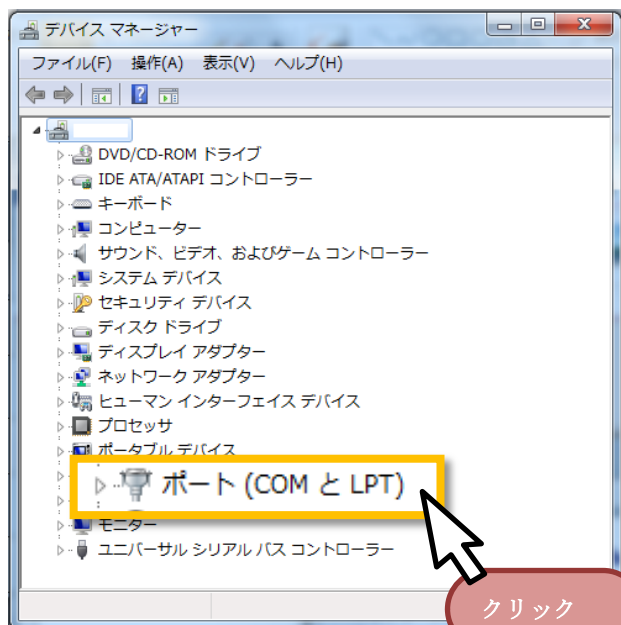
カテゴリ
ブログ
ロボット
ロボットスキルウェア
ROS
RTミドルウェア
RTM4.0
RTM4.0
インストール
ツール開発
開発環境
ロボット開発環境

最近の投稿
非同期系学習のための非同期系開発環境
RTM4.0マーケティング2016の開催
RTM4.0のインストール方法
RTM4.0のインストール方法
ロボットスキルウェアを使ってロボット開発

003 Stduino (Arduino) に, StarTno 用のスケッチをアップロード

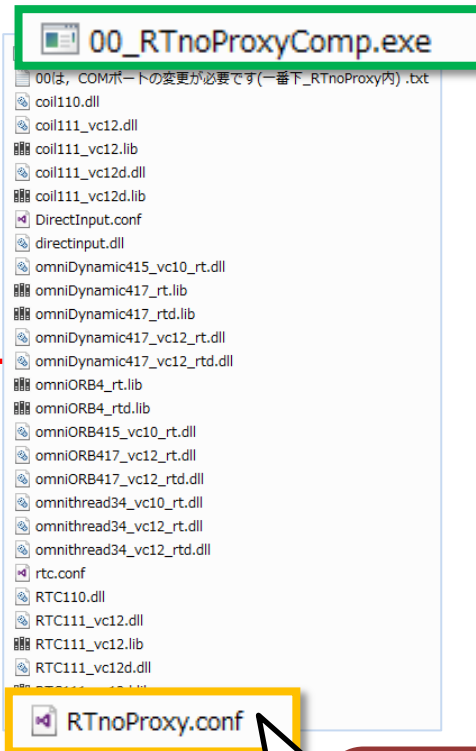
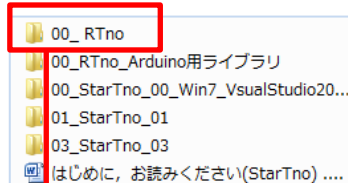


004 デバイスマネージャで Studuino (Arduino) が接続されている COM ポート番号を調べる



COM ポートの番号が複数あり、ボードに接続されている番号かわからない場合、PC とボードを接続している USB ケーブルを一度外し、再度接続したときに現れた番号が COM ポートの番号である。

005 RTnoProxy.conf をメモ帳で開き、COM の番号に書きかえ、保存する



```
#####
# connectionType
# --- serial : serial port connection
# --- tcp    : TCP connection
#####
# conf.default.connectionType:tcp
conf.default.connectionType:serial

#####
# ipAddress
#
#####
conf.default.ipAddress:192.168.42.100

#####
# portNumber
#
#####
conf.default.portNumber:23

#####
# comport
#####
conf.default.comport:### #COM14
#conf.default.comport:/dev/ttyACM0

#####
# baudrate
#####
conf.default.baudrate:57600
```

接続する COM ポートの番号に変更する

006 RTnoProxyComp.exe を起動する。

00_RTnoProxyComp.exe

00_RTno フォルダ内の最上段
00_RTnoProxyComp.exe をダブルクリック

```
[RTnoProxy] Configuration Values
[RTnoProxy] - conf.default.connectionType:serial
[RTnoProxy] - Serial Port Connection
[RTnoProxy] - conf.default.comport:YY.YCOM14
[RTnoProxy] - conf.default.baudrate:57600
[RTnoProxy] Opening SerialPort(YY.YCOM14)..... [RTnoProxy] Opened.
[RTnoProxy] Waiting for Startup the arduino...
[RTnoProxy] - 3.....
[RTnoProxy] - 2.....
[RTnoProxy] - 1..
[RTnoProxy] - Go!
[RTnoProxy] Starting up onInitialize sequence.
[RTnoProxy] - RTnoProtocol::setRTnoProfile() called.
[RTnoProxy] - Transfer Profile Request to Arduino.
[RTnoProxy] - Parsing RTnoProfile.
[RTnoProxy] -- Adding InPort (name=LED_13, typeCode=1)
[RTnoProxy] -- Adding InPort (name=Servo_09, typeCode=1)
[RTnoProxy] -- Adding OutPort (name=Analog_A0, typeCode=1)
[RTnoProxy] - Success.
[RTnoProxy] - RTno Status=73
[RTnoProxy] -- Now Arduino is in INACTIVE STATE
[RTnoProxy] Execution Context Type == 33
[RTnoProxy] - ProxySynchronousExecutionContext is detected.
[RTnoProxy] - This Arduino's on_execute is synchronously executed with this PC.

[RTnoProxy] - You can change execution ratio by configuring RTnoProxy's one.
[RTnoProxy] - on_initialized OK.
[RTnoProxy] RTnoProxy is successfully initialized.
```

```
[RTnoProxy] Opening SerialPort(YY.YCOM14)..... [RTnoProxy] Opened.
[RTnoProxy] Waiting for Startup the arduino...
[RTnoProxy] - 3.....
[RTnoProxy] - 2.....
[RTnoProxy] - 1..
[RTnoProxy] - Go!
[RTnoProxy] Starting up onInitialize sequence.
[RTnoProxy] - RTnoProtocol::setRTnoProfile() called.
[RTnoProxy] - Transfer Profile Request to Arduino.
[RTnoProxy] - Parsing RTnoProfile.
[RTnoProxy] -- Adding InPort (name=LED_13, typeCode=1)
[RTnoProxy] -- Adding InPort (name=Servo_09, typeCode=1)
[RTnoProxy] -- Adding OutPort (name=Analog_A0, typeCode=1)
[RTnoProxy] - Success.
[RTnoProxy] - RTno Status=73
[RTnoProxy] -- Now Arduino is in INACTIVE STATE
[RTnoProxy] Execution Context Type == 33
[RTnoProxy] - ProxySynchronousExecutionContext is detected.
[RTnoProxy] - This Arduino's on_execute is synchronously executed with this PC.

[RTnoProxy] - You can change execution ratio by configuring RTnoProxy's one.
[RTnoProxy] - on_initialized OK.
[RTnoProxy] RTnoProxy is successfully initialized.
```

接続の失敗例 1

Studuino が接続されていない。(物理的に接続していない)
デバイスドライバにない COM ポートの番号
→接続に失敗、動作停止。

```
[RTnoProxy] Configuration Values
[RTnoProxy] - conf.default.connectionType:serial
[RTnoProxy] - Serial Port Connection
[RTnoProxy] - conf.default.comport:YY.YCOM14
[RTnoProxy] - conf.default.baudrate:57600
[RTnoProxy] Opening SerialPort(YY.YCOM14)..... [RTnoProxy] Failed (COM Open Error)
[RTnoProxy] Component create failed.
```

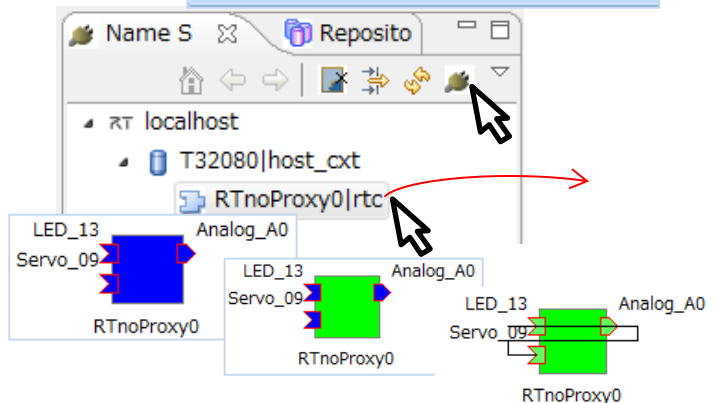
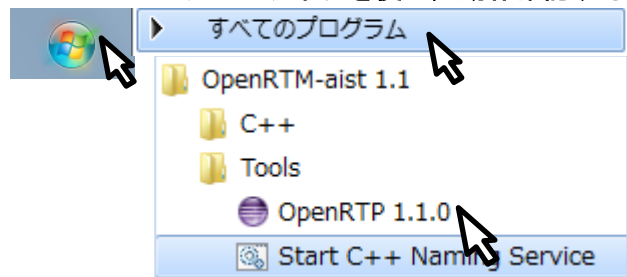
```
[RTnoProxy] Configuration Values
[RTnoProxy] - conf.default.connectionType:serial
[RTnoProxy] - Serial Port Connection
[RTnoProxy] - conf.default.comport:YY.YCOM14
[RTnoProxy] - conf.default.baudrate:57600
[RTnoProxy] Opening SerialPort(YY.YCOM14)..... [RTnoProxy] Failed (COM Open Error)
[RTnoProxy] Component create failed.
```

接続の失敗例 2

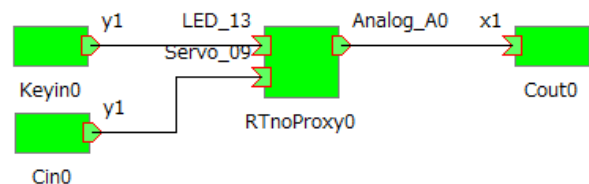
COM ポートの番号がデバイスドライバに存在するが、Studuino(Arduino)の COM ポートの番号ではない場合。
→3, 2, 1, Go!と進んで、停止。

```
[RTnoProxy] Configuration Values
[RTnoProxy] - conf.default.connectionType:serial
[RTnoProxy] - Serial Port Connection
[RTnoProxy] - conf.default.comport:YY.YCOM1
[RTnoProxy] - conf.default.baudrate:57600
[RTnoProxy] Opening SerialPort(YY.YCOM1)..... [RTnoProxy] Opened.
[RTnoProxy] Waiting for Startup the arduino...
[RTnoProxy] - 3.....
[RTnoProxy] - 2.....
[RTnoProxy] - 1..
[RTnoProxy] - Go!
[RTnoProxy] Starting up onInitialize sequence.
[RTnoProxy] - RTnoProtocol::setRTnoProfile() called.
[RTnoProxy] - Transfer Profile Request to Arduino.
```

007 RT システムエディタを使い、動作確認する

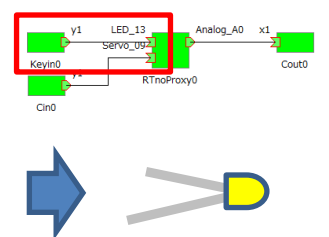


単体テスト : RTnoProxy をアクティベート
Analog_A0 ポートと LED_13 ポート接続→ボリュームを回すと点滅周期が変化する。
Analog_A0 ポートと Servo_09 ポートを接続→ボリュームを回すとサーボモータが回転する。

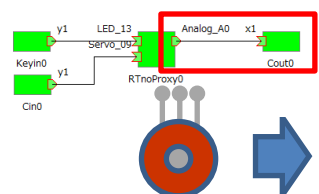
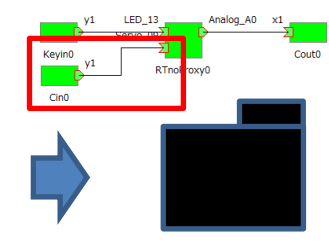


接続テスト : Keyin,Cin,Cout を起動・接続しアクティベート
Keyin を使い、LED の点滅周期を変化させる。
Cin で数値を入力し、サーボモータを回転させる。
ボリュームを回すと Cout から出力される値が変化する。

```
Activated [1,2,a,↑↓←→]
key 1 → 1 を出力
key 1 → 1 を出力
key 1 → 1 を出力
key 1 → 1 を出力
key up → 元に10足して出力
key up → 元に10足して出力
key up → 元に10足して出力
```



```
onInitialize
onActivated
onExecute
整数を入力してください。
180
onExecute
整数を入力してください。
```



可変抵抗器を距離センサなど
各種センサに、取り換え可能

```
*****209
*****222
*****234
*****244
*****252
*****260
*****270
*****280
*****292
*****303
*****214
```

008 スケッチの作り方

Arduino の標準的な void setup(), void loop() という構造ではなく, onInitialize(), onExecute() など, RTM の表記方法に準拠している. 必要なヘッダファイルインクルードし, 変数を定義する. ポートの設定で入出力ポートを設定する. OnInitialize() で初期化処理を行い, onExecute() にメインとなるスケッチを書く. OnInitialize(), onExecute() は, Arduino の標準的な void setup(), void loop() に対応する.

```
#include <UART.h>
#include <RTno.h>
#include <Servo.h>
```

インクルード

```
int LedVal_H;
int LedVal_L;
Servo Servo09;
int ServoVal09;
```

変数の定義

```
void rtcconf(config_str& conf, exec_cxt_str& exec_cxt) {
    conf._default.connection_type = ConnectionTypeSerial1;
    conf._default.baudrate = 57600;
    exec_cxt.periodic.type = ProxySynchronousExecutionContext;
}
```

ポートの設定

```
TimedLong in13;
InPort<TimedLong> in13In("LED_13", in13);
TimedLong in09;
InPort<TimedLong> in09In("Servo_09", in09);
TimedLong outA0;
OutPort<TimedLong> outA0Out("Analog_A0", outA0);
```

```
int RTno::onInitialize() {
    pinMode(13, OUTPUT);
    pinMode(9, OUTPUT);
    pinMode(A0, INPUT);
    addInPort(in13In);
    addInPort(in09In);
    addOutPort(outA0Out);

    digitalWrite(13, HIGH); // set the LED on
    LedVal_H=1;
    LedVal_L=19;
    in13.data=1;

    Servo09.attach(9);
    ServoVal09 = 90;
    return RTC_OK;
}
```

初期化処理

```
int RTno::onActivated() {

    // Write here initialization code.

    return RTC_OK;
}
```

```
int RTno::onDeactivated()
{
    // Write here finalization code.

    return RTC_OK;
}
```

メイン

```
int RTno::onExecute() {
    digitalWrite(13, HIGH);
    delay(LedVal_H);
    digitalWrite(13, LOW);
    delay(LedVal_L);
    // wait for a second
```

LED_13 点滅

```
if(in13In.isNew()) {
    in13In.read();
    if (in13.data>0) {
        LedVal_H=1000.0/in13.data;
        LedVal_L=1000.0/in13.data;
    }
    else if(in13.data<1) {
        LedVal_H=1;
        LedVal_L=10;
    }
}
```

点滅周期の変更

```
if(in09In.isNew()) {
    in09In.read();
    ServoVal09 = in09.data;
    if (ServoVal09<0 )ServoVal09=0;
    if (ServoVal09>180 )ServoVal09=180;
    ServoVal09 = map(ServoVal09,0,180,0,180);

    Servo09.write(ServoVal09);
    delay(3);
}
```

Servo_09 回転

```
outA0.data = analogRead(0);
outA0Out.write();

return RTC_OK;
}
```

アナログ入力→出力

```
int RTno::onError()
{
    return RTC_OK;
}

int RTno::onReset()
{
    return RTC_OK;
}
```

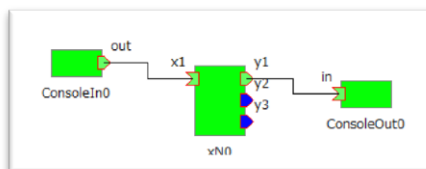
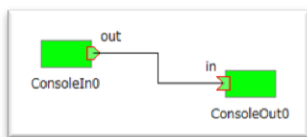
このスケッチを利用して, 不要な部分をカットし, 増加させたいポートを追加するなどして, スケッチを作成する.

解説

全体のまとめ

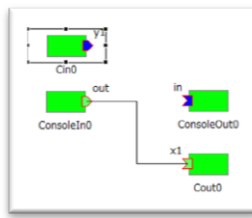
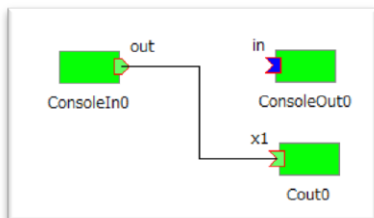
□StaraTno_00 説明書 01(Win7-VS13) : RTC の使い方・作り方 1

RT システムエディタで、コンソールイン、コンソールアウトを接続・アクティベートし動作確認を行った。RT ビルダーで RTC の基本設計を行い、CMake でファイルを作り、VsualC++で RTC を作成した。xN は、入力値を 2, 10, 100 倍にして出力。



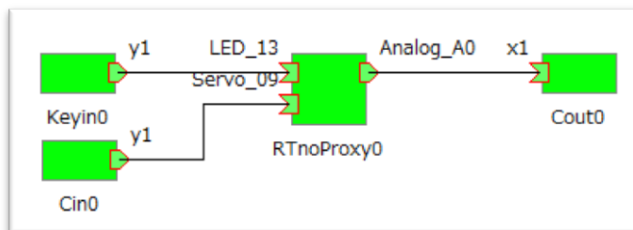
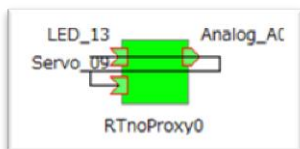
□StaraTno_00 説明書 02(Win7-VS13) : RTC の作り方 2

Cout, Cin, Keyin という 3 つの RTC を作成した。RT ビルダー、CMake, VsualC++で RTC の作成し、RT システムエディタで動作確認を繰り返すことで、RTC 作成の基本手順が身につくようにした。



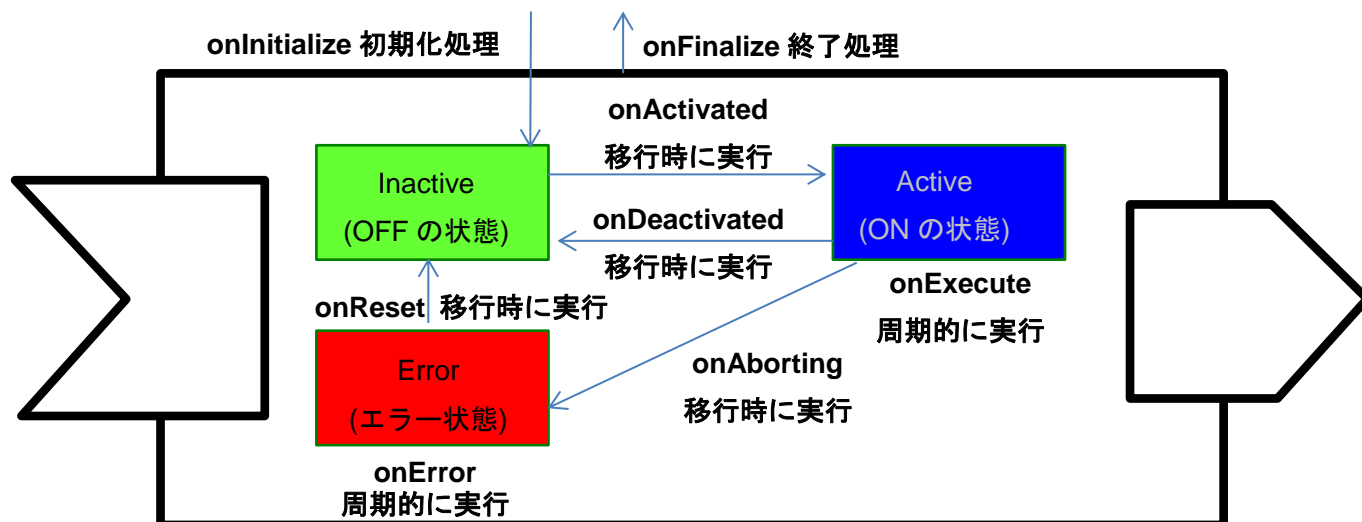
□StaraTno_00 説明書 03 : RTno の使い方・スケッチの作り方

RTno を使用する環境を整備し、RTno の使い方を確認した。Arduino 用スケッチの構造を説明した。



内部状態

RTC を作成には、StaraTno_00 説明書 01 で説明した 3 つの内部状態を理解する必要がある。



ファイル

主要なフォルダ及びファイルを示します。緑の四角い枠で囲まれたファイルが RTC.

