

neurocube プログラミング入門

タイル言語のプログラミングを体験します。

(neurocube は、VB や VC++、Java などでもプログラミングすることができます。)

カーネルブロックの機能の内、ライトとサウンド、カレンダーの機能を使います。

(携帯の機能は、現在のバージョンではまだ使用できません。)

サンプルは neurocube¥sample の中にあります。

1. 信号機を作ってみよう。(サンプルプログラム neuro_shingouki.toy)

使用するブロック

カーネルブロック、バッテリーケース(電池装着)

信号機の仕様

- ・青 30 秒 音楽を鳴らす
- ・青点滅 5 秒
- ・黄色 5 秒
- ・赤 30 秒

前準備

- ・neurocube 制御プログラムを立ち上げます。
- ・カーネルブロックのモードスイッチをP(プログラムモード)にします。
- ・シリアルケーブルをカーネルブロックの RS232C ジャックに差し込み、もう一方をパソコンのシリアルポートに接続します。
- ・バッテリーケースに電池を入れ、ジョイントボタンでカーネルブロックにつなぎます。

プログラミング

ステップ1. カーネルブロックのバインディング



組立てウィンドウを選択し、

組立てウィンドウで



バインドボタンを押します。

4 個のブロックのアイコンが現れます。

(ライト・サウンド・カレンダー・携帯の各機能を自動認識し画面に表示します。)



ここでライト機能の動作確認をしてみましょう。



Light-0

のアイコンをダブルクリック

ライト機能のパネルが現れます。



初期化と制御モードを選び

色をクリック、適当な色を選びます。

ONを押すとカーネルブロックのカラーLEDが点灯し、OFFで消灯します。

モードの選択で種々のパターンで点灯します。

パネル右上の[×]を押してパネルを閉じます。

同様にサウンド機能の動作確認をしてみましょう。



Buzzer-0

のアイコンをダブルクリック

サウンド機能のパネルが現れます。



初期化と制御モードを選び

ボリュームを上げ、

ONします。音が鳴ることを確認して下さい。

モードの選択で種々のパターンで鳴ります。

ここでは、「早い断続音」を選びます。

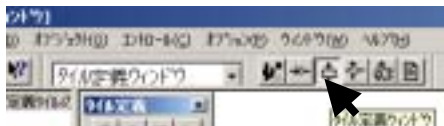
(プログラムの中で青色点灯の時に鳴らします。)

確認が終れば[×]を押してパネルを閉じます。

ステップ2. コマンドタイルの作成

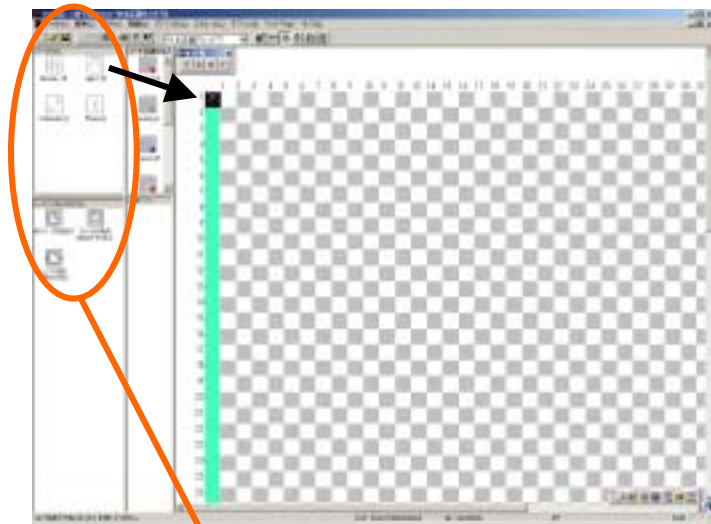
ライト機能・サウンド機能の命令を表すタイル(コマンドタイル)を作ります。

タイル定義ウィンドウで作成します。



画面選択メニューから、タイル定義ウィンドウを選びます。

タイル定義ウィンドウに移ります。



パーツリスト

バインド済み(ネットワーク構築された)ブロックの一覧

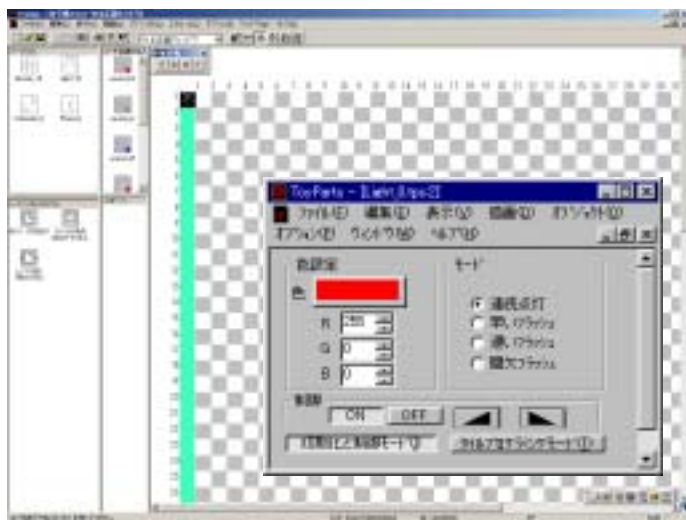
赤色点灯タイルの作成



Light-0

パーツリストから、
ライト機能のアイコンをドラッグ & ドロップで右の緑ライン上に持ってきます。

これをダブルクリックしてライトの
パネルを表示します。



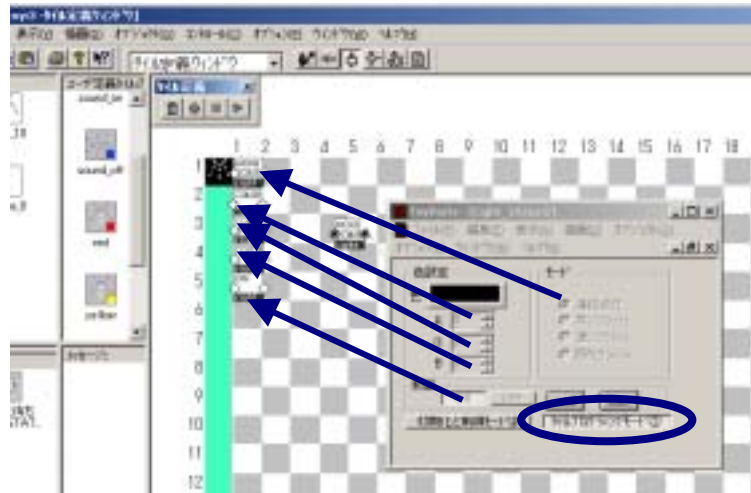
初期化と制御モードにします。

色をクリックすると、カラーパレット
が現れるので赤色を選択します。

選択した色が に表示され、同時
に に RGB で数値表示されます。

(逆に、数値入力しても OK です。)

タイルプログラミングモードに切替
えて、順にコマンドをドラッグ &
ドロップしていきます。



- ・モード(連続点)
- ・色 R
- G
- B
- ・制御(ON)

コマンドの配置が終わればコマンド(命令)タイルの中身は出来上がりです。*
 後は、そのコマンドタイルに名前を付け、内容を表す適当なアイコンを付ければ、コマンドタイルの完成です。

☐ を押してライトのパネルを閉じます。

*「タイルのテスト実行」で、作成中のタイルの動作確認ができます。



(D) のテスト実行ボタンを押して、カーネルブロックが赤色点灯することを確認して下さい。

確認が終わったら (C) の「停止」ボタンを押します。

(テスト実行の停止)

(B) 「登録」ボタンを押します。

コマンドアイコンリストが現れます。

リストの中から適当なアイコンを選び、選択ボタンを押します。

(自分のオリジナルアイコンをデザインすることもできます。)

名前を入力するダイアログが現れます。

適当な名前を付け(ここでは、赤色点灯のタイルですから、red という名前を入力)、OKボタンを押します。

これで赤色点灯させるコマンドタイルは完成です。

(完成したタイルは、ユーザ定義タイルのリストの中に入り、プログラミングに即、使用できます。)

同様にして、青色点灯、青点滅、黄色点灯のタイルを作成して下さい。

サンプルプログラムでは、それぞれの名を “blue”, “blue_blink”, “yellow” としています。

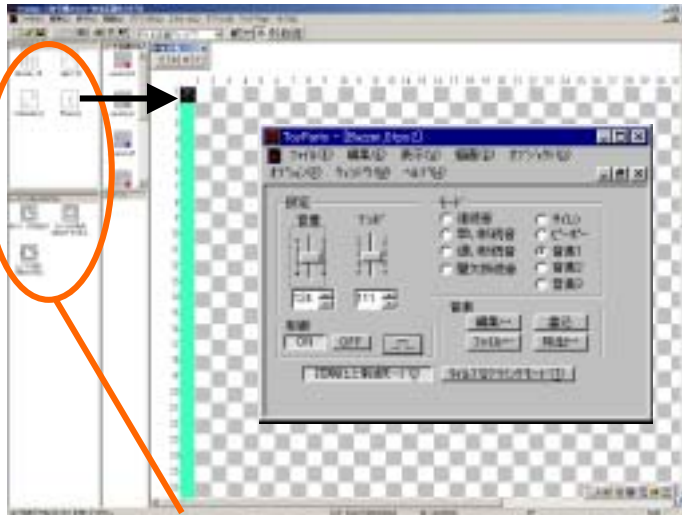
同様に、**サウンド(音)を鳴らす**コマンドタイルを作成します。

ライトのコマンドタイルを作成したのと同じ**タイル定義**ウィンドウで、



Buzzer-0

パーツリストから、サウンド機能のアイコンをドラッグ & ドロップで右の緑ライン上に持てきます。これをダブルクリックしてサウンドのパネルを表示します。



パーツリスト

バインド済み(ネットワーク構築された)ブロックの一覧



タイルプログラミングモードにします。

「早い断続音」

「ボリウム」

「OFF」の順に、コマンドをドラッグ & ドロップします。

サウンドパネルを閉じます。



① ② ③ ④

② 「登録」ボタンを押します。

コマンドアイコンリストが現れます。

リストの中から適当なアイコンを選び、**選択**ボタンを押します。

名前を入力するダイヤログが現れます。

適当な名前を付け(ここでは、sound_init と名前を入力)、

OKボタンを押します。

これでサウンド初期状態のコマンドタイルは完成です。

(完成したタイルは、ユーザ定義タイルのリスト内に入っています。)

同様に、サウンド**ON**だけのタイルを作成し、名前を "sound_on" とします。

また、サウンド**OFF**だけのタイルを作成し、これを "sound_off" とします。

ステップ3. タイルを使ったプログラミング

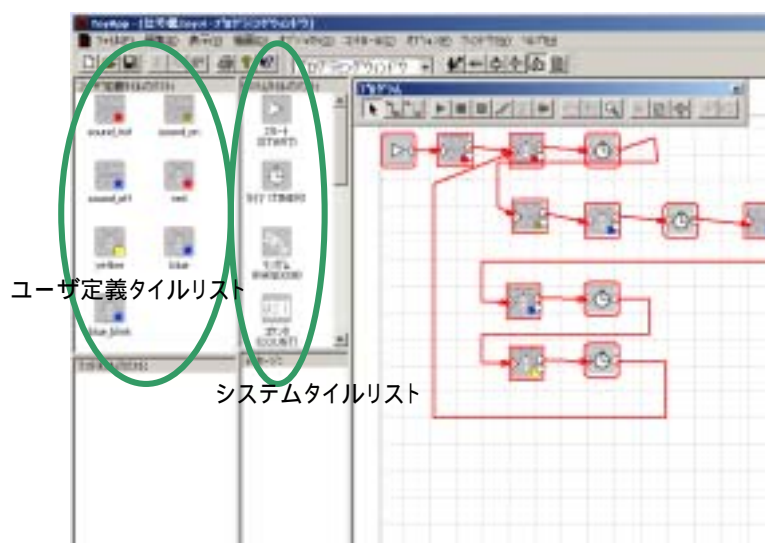
ステップ2で作成したタイルを使って、プログラミングします。

プログラミングウィンドウで作成します。



画面選択メニューから、
プログラミングウィンドウを選びます。

プログラミングウィンドウに移ります。

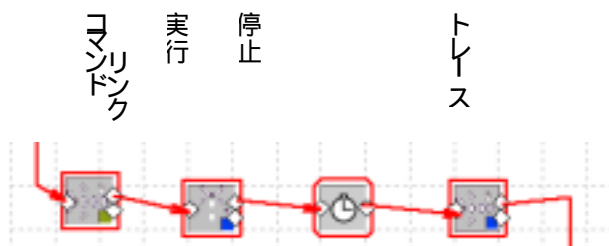


プログラミングについては、第 4 章で
詳しく説明していますが、ここでは
取りあえず、ステップ2で作成したタイル
を使って、信号機のプログラムを作りま
す。


ユーザ定義タイルリストの中から、
“red”, “blue”, “blue_blink”, “yellow”
“sound_init”, “sound_on”, “sound_off”
等のステップ2で作成したコマンドタイルと、
システムタイルのリストの中から
タイマーを数個、
プログラミングウィンドウにドラッグ&ドロップ
して配置します。

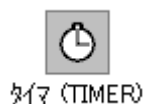


それらのコマンドタイルを矢印(コ
マンドリンク)でつないでいきます。



矢印(コマンドリンク)のつながり方

をクリックしてから、始点、終点をクリックして
下さい。(マウスのカーソルが“”から“+”に
変わった時にクリックして下さい。)



タイマーは、ダブルクリックしてタイマー定数
を入れて下さい。(単位は 1/1000 秒です。)

実行ボタンで出来上がったプログラムを実行します。

トレース機能(このボタンを押すと、実行中のタイルの色が変わり、プログラムの
進行状況が良く理解できます。

プログラム実行を停止します。

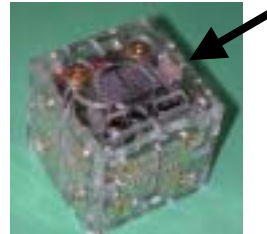
ステップ4. 信号機の完成

ステップ3の、 でプログラム実行ボタンを押すことにより、neurocube にはプログラムが書き込まれています。

PC と neurocube の接続ケーブルを外します。

カーネルブロックのモード切り替えスイッチを R (RUN モード(自律モード)にします。)

電源(バッテリーケース)を一度外し、改めてセットします。



最初に設計した仕様通り、動きます。

・青	30 秒	音楽を鳴らす
・青点滅	5 秒	
・黄色	5 秒	
・赤	30 秒	
これをくり返します。		

2. サンプル2 目覚まし時計(カレンダー機能) (サンプルプログラムneuro_alarm_no_power.toy)

カーネルブロックの カレンダー機能を使います。

カレンダー機能については、2.3 カレンダー機能、2.4 LED 表示器の章で詳しく述べています。

使用するブロック

カーネルブロック、表示器、AC アダプタ or バッテリーケース(電池装着)

目覚まし時計の仕様

- ・朝、決まった時間に 音楽を鳴らす
- ・時刻を表示

前準備

- ・neurocube 制御プログラムを立ち上げます。
- ・カーネルブロックのモードスイッチをP(プログラムモード)にします。
- ・シリアルケーブルをカーネルブロックの RS232C ジャックに差し込み、もう一方をパソコンのシリアルポートに接続します。
- ・バッテリーケースに電池を入れ、ジョイントボタンでカーネルブロックにつなぎます。
- ・同様に表示器をカーネルブロックにジョイントボタンでつなぎます。

プログラミング

ステップ1. バインディング



組立てウィンドウを選択し、

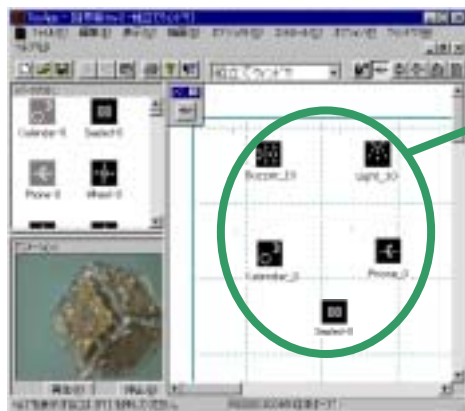
組立てウィンドウで



バインドボタンを押します。

5 個のブロックのアイコンが現れます。

(カーネルのライト・サウンド・カレンダー・携帯の各機能と、LED 表示器が自動認識され画面に表示されます。)

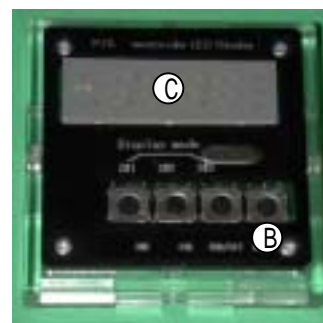
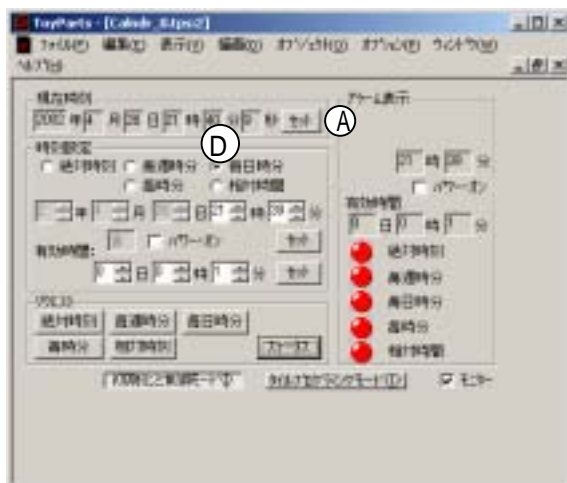


ステップ2. アラーム時刻の設定



自動認識され画面に現れたアイコンの内、
カレンダーアイコンをダブルクリックします。

カレンダーパネルが表示されます。



まず、現在の時刻をセットします。

- ① ① 現在の時刻の「セット」ボタンを押します。 カーネルブロックに現在時刻が書き込まれます。
次の方法で、正しく時刻がセットされたか確認します。
- ② ② 表示器の右端のモードボタンを“nou”の表示が出るまで数回押します。
・“nou”が出たらボタンを押すのを止め、表示器の LED 表示③ の変化を見て下さい。
・年、月日、時分の順に表示されます。

次にアラームの時刻を設定します。

- ・目覚まし時計ですから、時刻の設定は ④ 「毎日時分」を選びます。

まず、動作チェックをしてみましょう。

モニターにチェックを入れます。

現在時分より2分先にアラームの時刻を設定し「セット」ボタンを押します。

有効時間^{*}は1分にし、「セット」ボタンを押します。

リクエストの「毎日時分」を押します。右側のアラーム表示ランプがきちんと
設定通りに反応しているか確認します。(設定時刻になると、毎日時分の
ランプが ON (緑) になれば OK です。)

モニターチェックを外します。

- ・目覚ましの鳴る時刻を設定し 「セット」ボタンを押します。
- ・有効時間^{*}を1分にし 「セット」ボタンを押します。

(^{*} 有効時間とは、アラーム ON になっている時間。ここでは、設定した時刻から
1分間、アラーム ON の信号が返されることを意味します。)

各「セット」ボタンを押すことで カーネルブロックのメモリーに各項目が書き込まれます。

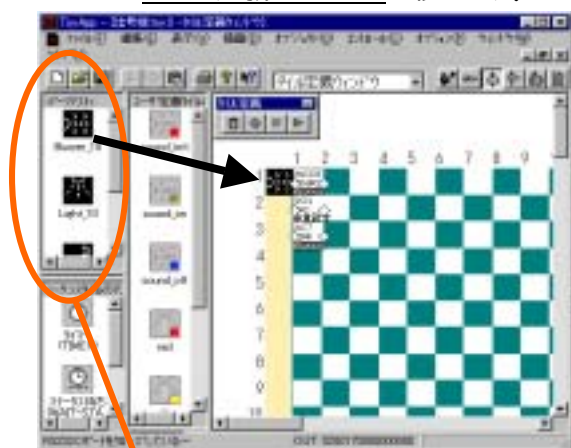


ステップ3. コマンドタイルの作成

目覚まし音のタイル(コマンドタイル)を作ります。

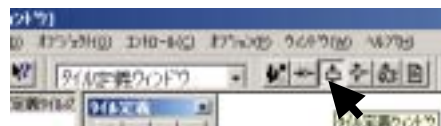
タイル定義ウィンドウで作成します。

タイル定義ウィンドウに移ります。



パーツリスト
バインド済み(ネットワーク構築された)ブロック

画面選択メニューから、タイル定義ウィンドウを選びます。



目覚まし音の作成



Buzzer-0

パーツリストから、サウンド機能のアイコンをドラッグ&ドロップで右の縦ライン上に持ってきます。

これをダブルクリックしてサウンドの
パネルを表示します。



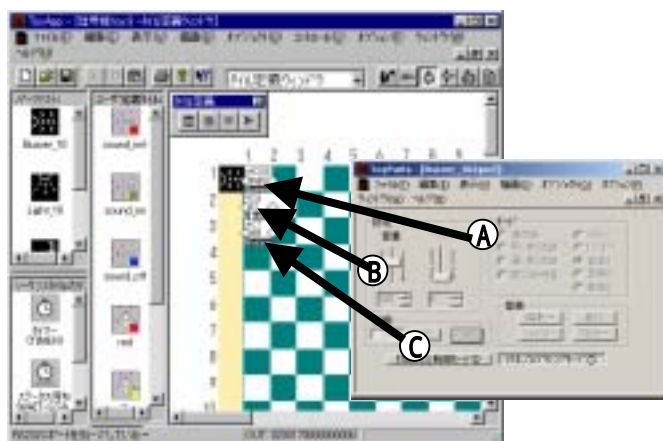
このウィンドウでも、各機能の
モニタリングができます。

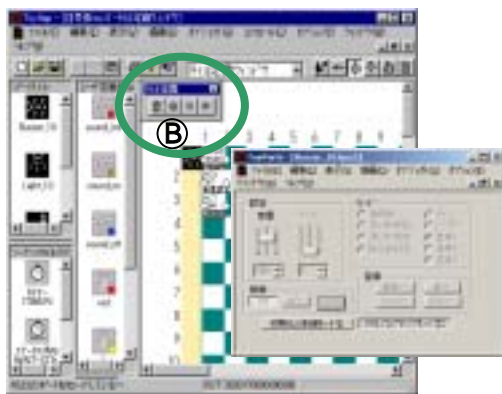
初期化と制御モードにします。

連続音を選び、音量を上げて
ON します。(音が鳴れば OK)
音量や、音の種類を変えてみて
下さい。

タイルプログラミングモードに切替
えます。

① 目覚ましとして鳴らしたい音、
又は音楽、② 音量、③ OFF の
順に制御要素をドラッグ&ドロップ
していきます。





サウンドパネルを閉じます。



「登録」ボタン(B)を押します。

コマンドアイコンリストが現れます。(中身の出来上ったタイルに、アイコンを付けます。)

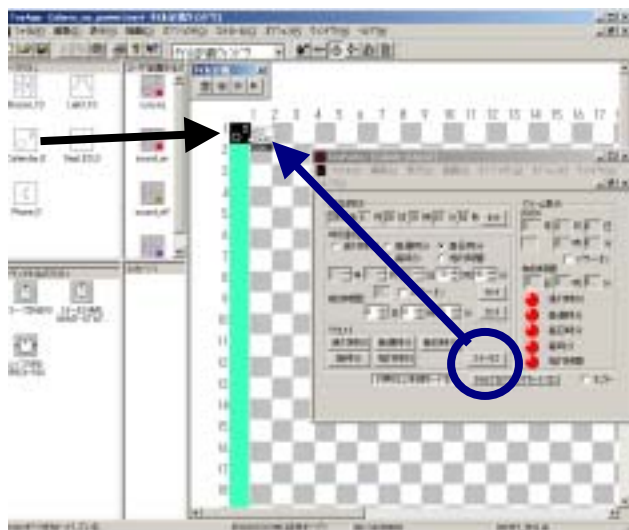
リストの中から適当なアイコンを選び、**選択**ボタンを押します。
名前を入力するダイアログが現れます。

当な名前を付け(ここでは、sound_init と名前を入力)、
OKボタンを押します。

(サウンド初期化のタイルの完成です。)

同様に、サウンド **ON** だけのタイルを作成し、名前を "sound_on" とします。
また、サウンド **OFF** だけのタイルを作成し、これを "sound_off" とします。
これでサウンドのコマンドタイルが3個できました。

もう一つ、カーネルブロックに現在のアラーム状態を返す要求する命令タイルを作ります。
同じくタイル定義ウィンドウで



パーツリストから
カレンダーのアイコンをウィンドウに
ドラッグ&ドロップし、ダブルクリック。

カレンダーのパネルを**タイルプロ
グラミングモード**にして
制御要素**ステータス**ボタンをタイル
作成ウィンドウにドラッグ&ドロップ。
カレンダーパネルを閉じ、
適当なアイコンを選び、
名前を付けます。(cal_req)

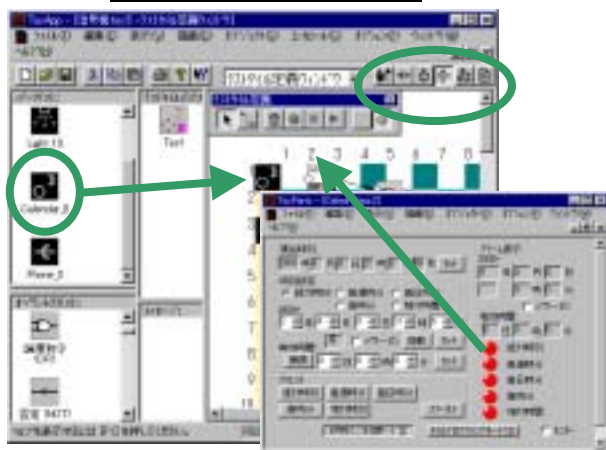
(このタイルは、必ずしも必要ではありません。プログラムを実行した時がたまたまアラーム
の有効時間内であったという時のみ、意味を持ちます。)

ユーザ定義タイルのリスト内に以上 4 個のタイルが入っているか確認してください。

ステップ4. アラーム時刻タイルの作成
テストスタイル定義ウィンドウで作成します。



テストスタイル定義ウィンドウに移ります。



② 登録ボタンを押し、
作成したタイルに適切なアイコンを付け、
名前を付けます。(ここでは every_day_alarm とします。)

面選択メニューから、テストスタイル
定義ウィンドウを選びます。

(このウィンドウでは、条件判定
文のタイルを作成します。)

パーツリストからカレンダーのアイコ
ンを右のライン上にドラッグ&ドロ
ップします。

タイルプログラミングモードを押し、
毎日時分のランプをドラッグ&ドロ
ップします。

カレンダーのパネルを閉じます。



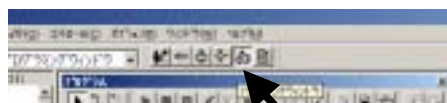
① ② ③ ④

これで、毎日、設定した時間になったら、という条件判定文のタイルが完成です。

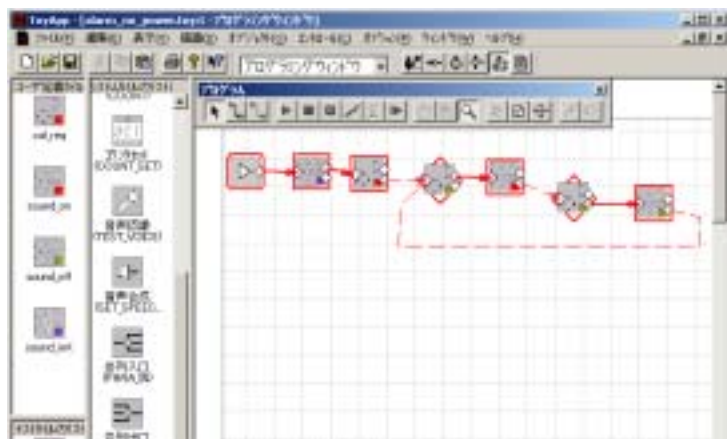
ステップ5. 作成したタイルを使ってプログラミング

ステップ3, 4で作成したタイルを使って、プログラミングします。

プログラミングウィンドウで作成します。



プログラミングウィンドウに移ります。



画面選択メニューから、
プログラミングウィンドウを選びます。

ユーザ定義コマンドタイル、ユーザ定義
テストタイル、システムタイルのリス
トの中からタイルプログラミングウィ
ンドウにドラッグ&ドロップしてして配
置します。

それらのコマンドタイルを矢印(コ
マンドリンク・割り込みリンク)でつ
ないでいきます。



出来上がったプログラムは、実行ボタンを押せばプログラム実行されます。

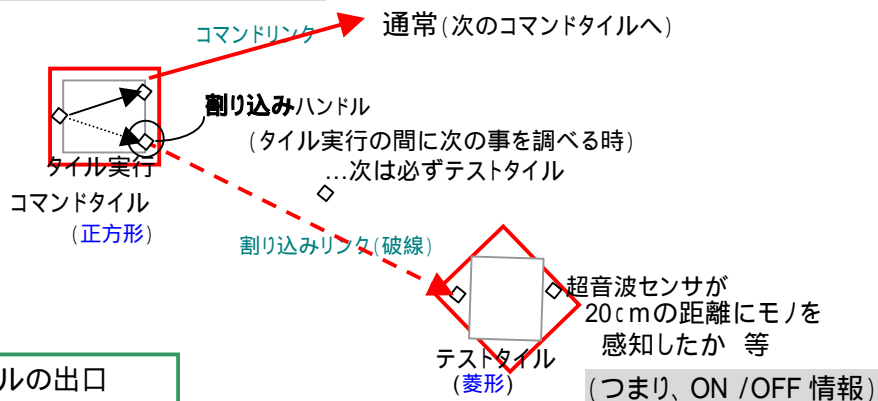
取りあえず、2～3分後にアラームを設定して、きちんと動作するのを確認して下さい。それから正規の時間に設定し直して下さい。

プログラム停止します

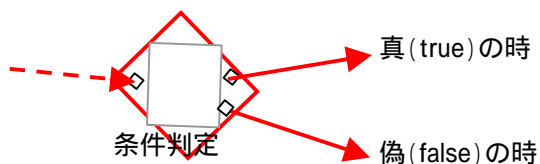
(注) プログラミングにおける約束事

コマンドリンク (実線)・割り込みリンク (破線)
コマンドタイルの出口と、リンクの種類 について

デジタル情報の条件判定文の場合



テストタイルの出口



コマンドリンク (実線)
割り込みリンク (破線)
それぞれのボタンを押してから
始点と終点をクリックします。

【注】「アナログ情報取り込み」と、「アナログ情報の演算」に関するテストタイルについては P49、50 を参照

ステップ6. 目覚まし時計の完成

ステップ5の、 でプログラム実行ボタンを押すことにより、neurocube にはプログラムが書き込まれています。

PC と neurocube の接続ケーブルを外します。

カーネルブロックのモード切り替えスイッチを **R(RUN モード)** にします。

AC アダプタ or 電源(バッテリーケース)を一度外し、改めてセットします。

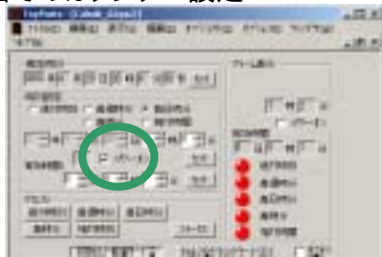

翌日の朝には設定通り、貴方を起こしてくれるでしょう。


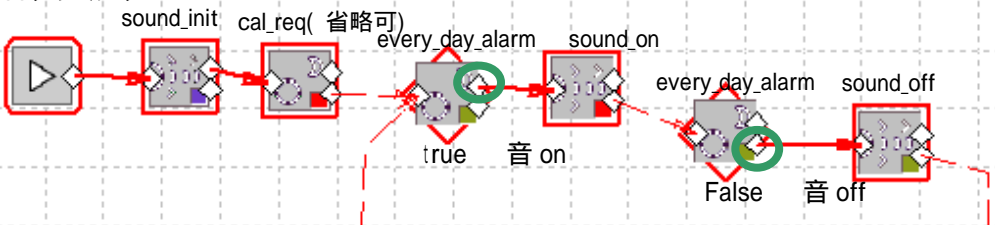
neurocube には、MIDI から MA1 に変換する neurocube ミュージックツールも標準装備していますので、目覚まし音楽も自由に編集・設定することができます。

次は、省電力機能を使い、サンプル2と全く同じ働きをさせるプログラムです。

サンプル2 と比較してみましょう。

3 サンプル3 目覚まし時計(省電力機能) (サンプルプログラム ¥neuro_alarm_power.toy)

サンプル3 (省電力機能) (サンプルプログラム ¥alarm_power.toy)	
組立て画面でのカレンダー設定	 <ul style="list-style-type: none"> ・現在時刻 セット ・毎日時分(ラジオボタン)選ぶ ・アラーム時刻を入力し セット ・有効時間入力し、 セット ・アラーム時刻を入力し セット ・有効時間入力し、 セット
プログラミング	 <p>設定された時間に毎日電源が入り、有効時間の間アラーム信号が返される</p>
カーネルブロックのモードスイッチ	A (アラームモード)

サンプル2 (省電力機能使用せず) (サンプルプログラム ¥alarm_no_power.toy)	
組立て画面でのカレンダー設定	 <ul style="list-style-type: none"> ・現在時刻 セット ・毎日時分(ラジオボタン)選ぶ ・アラーム時刻を入力し セット ・有効時間入力し、 セット
プログラミング	
カーネルブロックのモードスイッチ	R (RUN モード)

注意！ 組立て画面でのカレンダー設定は、サンプルプログラムを使う場合でも、その都度設定することが必要です。(カーネルブロックのメモリーへの書き込み)

逆に、現在カーネルブロックに書き込んであるアラームの値を見たい時は、組立て画面のカレンダーパネルでモニターにチェックを入れ、各アラームのリクエストボタンを押すとパネル右側に表示されます。
(. ブロック・機能の単体制御と、モニタリング 2.3 (カレンダー機能).3 リクエストの項)

次は、アナログ演算機能を使用するプログラムです。

neurocube はデジタル情報だけでなく、数値を扱うことができます。

レジスタ機能により値を取り込み、その値を次のステップの設定値として使用することができます。

4 . サンプル4 超音波距離計 (サンプルプログラム ¥neuro_ult_distnce_meter.toy)

目的

超音波センサで距離を測り、LED表示器に表示させます。

使用するブロック

カーネルブロック、超音波ブロック、表示器、バッテリーケース(電池装着)

前準備・プログラミング

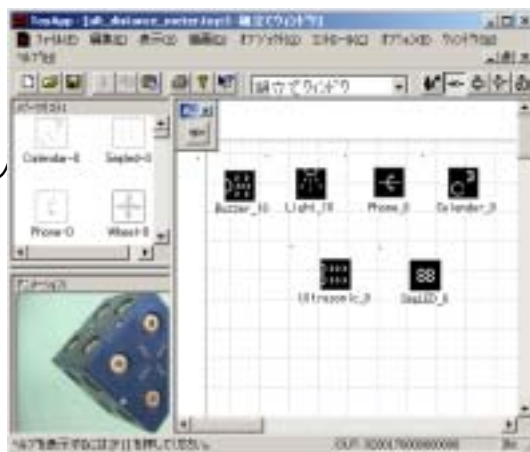
バインディングまでは同じです。

また、各タイルの登録なども同じ手順ですので、サンプル1、2を参照下さい。

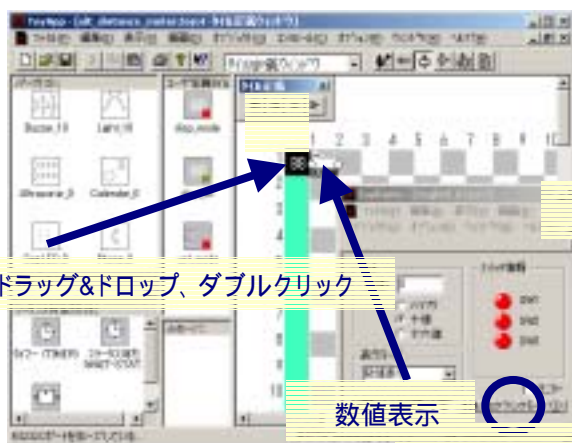
サンプル4以下は主なタイルの中身と、プログラミング画面の説明を主に説明します。

ステップ1. 組立てウィンドウ

バインディングするとカーネルブロックの3機能(+ 携帯のアイコン:現在のバージョンでは使用できません)と超音波、そしてLED表示器のアイコンが現れます。



ステップ2. タイル定義ウィンドウ



LED表示器のモード設定タイル
・数値表示モード

5. サンプル5 距離によるボリュームコントロール (サンプル≠neuro_volume_distance.toy)

パラレル処理機能、アナログ演算機能、レジスタ機能を使用するプログラムです。

目的

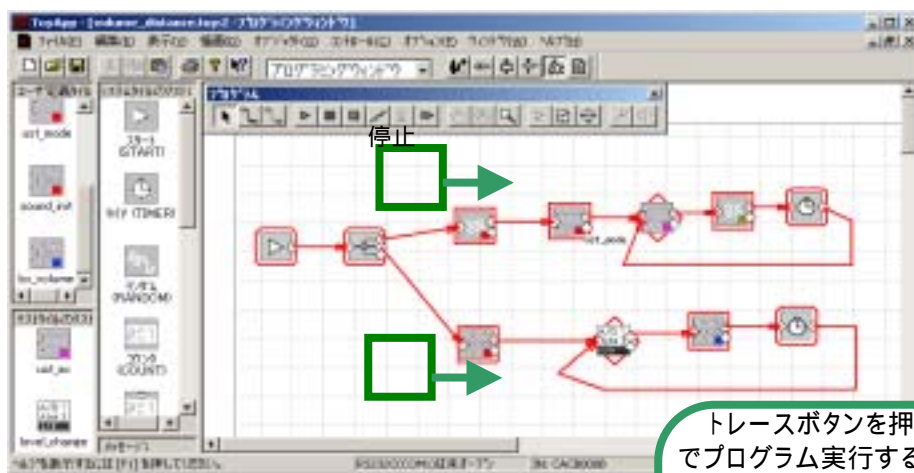
超音波センサに手をかざし、その距離の変化により、インタラクティブに音楽の音量を変化させます。(近くなれば小さく、遠くなれば大きく)

使用するブロック

カーネルブロック、超音波ブロック、表示器、バッテリーケース(電池装着)

ここでは、4の超音波距離計と同時並行して動くようにプログラムします。

超音波距離計で取り込んだ数値によって音量を設定します。



このプログラムが並行して動きます。



超音波からの距離情報がレジスタ AX に取り込まれ、その値を読み込んで 10 進で LED 表示器に表示します。



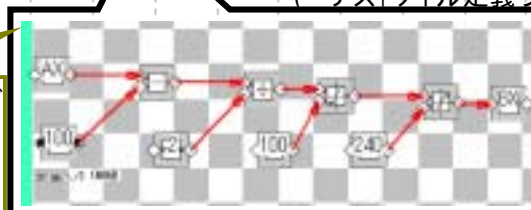
sound 初期化

音楽モード 2
音量設定
ON



レジスタ BX の内容を読み込む。
その値を音量設定値にする。

(テストタイル定義ウィンドウで作成)



でレジスタ AX に取り込んだアナログ情報値(AX とします)を使い、
 $(AX - 100) \div 2 = a$ とすると
 $100 < a < 240$ の時
 a がレジスタ BX に記憶される
 $a < 100$ の時
 100 がレジスタ BX に記憶される
 $a > 240$ の時
 240 がレジスタ BX に記憶される

のプロセス 超音波からの距離情報が例えば 400mm の時
 $(400 - 100) \div 2 = 150$ の音量で音楽演奏する、
 距離が変わればそれに応じて音量が変化する

6 . サンプル6 距離によるテンポコントロール (サンプル¥neuro_tempo_distance.toy)

超音波センサに手をかざし、その距離の変化により、インタラクティブに音楽のテンポを変化させます。(近くなれば遅く、遠くなれば速く)

サンプル5を参考に作成してみてください。