MSX0入門



目次

1. はじめに	
2. IoT BASIC	
3. Grove デバイス	
4. IoT 命令	
4.1. IOTPUT	
4.2. IOTGET	
4.3. IOTFIND	
4.4. IOTINIT	
5. デバイスノード	
6. 制御例	
6.1. バッテリー電流/レベルを取得する	
6.2. 音センサー	

1. はじめに

MSX0 が登場しましたが、SNS を眺めてみると「どうやって使ったら良いか分からない」「なにができるかわからない」という声を見かけました。せっかく登場した MSX0、確かに解説書の類いが付属のドキュメントのみ。付属のドキュメントも従来の MSX 用のものがメインで、MSX0 から追加された内容の説明は必要最小限のレベル。技術書などを読み慣れている人が、なんとか実験などを繰り返せば使えるようになる、といった情報量なので、当然ながら技術書を読み慣れていない人など、多くの一般人には「難解な代物」に見えるのかもしれません。本書は、そういった方達に向けて、もう少しだけ分かりやすく解説できればと思い執筆しました。私個人としては、MSX0 向けのアプリケーションが増えて、MSX0 が盛り上がって、そして MSX 全体が盛り上がる手助けになれば幸いです。

本書では、MSX0で追加になっている項目に絞って解説したいと思います。従来のMSXと互換の部分に関しては、MSX0に付属のBASIC解説書(3_MSX-BASIC Version 2.0.pdfや4_MSX-BASIC Version 3.0.pdf)などをお読みいただければと思います。特に、本書ではMSX-BASICをある程度使えることを前提としています。

本書に記載の内容が対象とする MSX0 のバージョンは、MSX0stack クラウドファンディング の初期状態のバージョンとします。今後のファームウェアバージョンアップによって変わる可能性が ありますのでご注意下さい。

2. IoT BASIC

MSX0 には、IoT BASIC と呼ばれる「MSX-BASIC に IoT 関連の機能を拡張する仕組み」が追加されています。起動すると、Fig.1 起動画面のような画面になります。

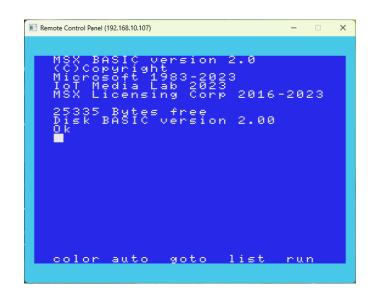


Fig.1 起動画面

従来の MSX を起動したときの表示と若干異なっています。 IoT Media Lab 2023 の表示 がありますね。この表示があると、IoT BASIC が使える状態になっています。「IoT BASIC が使える」とは、Grove デバイスの制御用命令が使えるということです。

IoT BASIC の追加命令はシンプルで、下記のコマンドのみになっています。後ほど一つずつ順番に説明していきます。

```
CALL IOTPUT( "デバイスノード", 値 )

CALL IOTGET( "デバイスノード", 変数名 )

CALL IOTFIND( "デバイスノード", 数値変数名 )

CALL IOTFIND( "デバイスノード", 配列変数名(0), 要素数 )

CALL IOTINIT()
```

命令の数は多くなく、たったの 5 種類です。これらは Grove デバイスに作用するのですが、命令の説明の前にまず Grove デバイスについて説明したいと思います。

3. Grove デバイス

MSX0 は、M5stackをベースにして、ハードウェアはほぼそのままに MSX エミュレーターをインストールしたものです。M5stack は、Grove 端子と呼ばれる接続端子を持っており、ここに接続したデバイスを制御したり、あるいはデバイスから得られる情報を読み取ることによって様々なことを行えるマシンです。M5stackをベースとする MSX0 にも Grove 端子が付いています。Grove 端子は、皆同じ形をしていますが種類があります。Fig.2 赤い Grove 端子や Fig.3 黒・水色の Grove 端子ですね。接続する Grove デバイスは対応する Grove 端子に接続する必要があります。



Fig.2 赤いGrove端子



Fig.3 黒・水色の Grove 端子

赤い Grove 端子は、Port.A と呼ばれ、 I^2C (アイ スクエア シー)と呼ばれるインターフェースになっています。

黒い Grove 端子は、Port.B と呼ばれ、GPIO と呼ばれるインターフェースになっています。 水色の Grove 端子は、Port.C と呼ばれ、UART と呼ばれるインターフェースになっています。

「インターフェースってなんぞ?」と思う方もいるかもしれませんが、接続するデバイスとの対話 の仕方みたいなものですね。赤いのは $\mathbf{I}^2\mathbf{C}$ というルールで繋がるぞ!ということです。赤い端子に、水色のデバイスを繋げても、 $\mathbf{I}^2\mathbf{C}$ と \mathbf{UART} では異なるルールなので、ちゃんと通信できない、ということですね。 $\mathbf{M5}$ 用に売られているデバイス類は、接続コネクタに色が付いていますので、同じ色の端子に繋げれば良いです。

他のデバイス用に作られた Grove で同じ形状の端子のデバイスも存在しますが、I2C なのか GPIO なのか UART なのか調べて、ピンアサインが同じで、電圧も同じであれば対応する色の端子に接続して使えますが、ピンアサインとかよく分からないという人は、Fig.4~M5 用の Grove デバイスのような M5 のものを使うことをオススメします。スイッチサイエンス(https://www.switch-science.com/) から購入できます。



Fig.4 M5 用の Grove デバイス

grove 端子のピンアサインは、下記のようになっています。

	ピン番号	意味
	1	SCL
Port.A	2	SDA
	3	+5
	4	GND

	ピン番号	意味
	1	GPIO
Port.B	2	GPIO
	3	+5
	4	GND

	ピン番号	意味
	1	RXD
Port.A	2	TXD
	3	+5
	4	GND

MSX0stack クラウドファンディング開始前に貸し出された「MSX0 先行試作機」には、Grove Beginner Kit For Arduino (https://www.switch-science.com/products/6361) が付属していました。中央の Arduino Uno 互換ボードから制御するための Grove のキットですが、この Grove の一部を利用することが出来ます。



残念ながら、現時点の MSXO では、素直に使えないものも含まれています。

4. IoT 命令

ここからは、各 IoT 命令について説明します。たった 5 命令ですが、全部丸暗記する必要はありません。どんな命令があるか、ざっくり覚えておいて、詳細を知りたくなったらこの章を見直してリファレンスのように使っていただければ充分です。

「命令の意味なんていいから、何か動くプログラムが欲しい!」という方は、6. 制御例をご覧下さい。

CALL は、_ と省略することが出来ます。CALL IOTGET は、_IOTGET と書いても同じです。

4.1. IOTPUT

【大書】

CALL IOTPUT(<デバイスノード>, <値>)

【機能】

デバイスへ値を送信します。

〈デバイスノード〉 を文字列で指定します。制御するデバイスを識別するために決められている文字列です。どのようなデバイスノードを指定できるかは、次章で説明します。

<値> は、指定のデバイスノードが示すデバイスへ送信する値です。数値を送信する場合は数値を、 文字列を送信する場合は文字列を指定します。

デバイスノードによって、数値を送信するタイプと、文字列を送信するタイプとがありますので、 デバイスノード表を参考に適切な方を指定して下さい。

4.2. IOTGET

【書式】

CALL IOTGET(〈デバイスノード〉, 〈変数名〉)

【機能】

デバイスから値を取得します。

〈デバイスノード〉 を文字列で指定します。制御するデバイスを識別するために決められている文字列です。どのようなデバイスノードを指定できるかは、次章で説明します。

〈変数名〉 は、指定のデバイスノードが示すデバイスから受け取った値を格納する変数の名前です。 数値を得たい場合は数値変数名、文字列を得たい場合は文字列変数名を指定します。 デバイスノードによって、数値が得られる場合と、文字列が得られる場合と異なりますので、デバイスノード表を参考に適切な方を指定して下さい。

4.3. IOTFIND

【書式1】

CALL IOTFIND(<デバイスノード>, <数値変数名>)

【書式 2】

CALL IOTFIND(<デバイスノード>, <配列変数名>(0), <個数>)

【機能】

デバイスから複数の値をまとめて取得します。

〈デバイスノード〉 を文字列で指定します。制御するデバイスを識別するために決められている文字列です。どのようなデバイスノードを指定できるかは、次章で説明します。

書式1は、書式2で取得できる個数を取得します。

書式 2 は、<個数>で指定の数だけ値を取得して、<配列変数名>で指定した配列変数に格納します。

要注意なのが、この命令はあまりメモリチェックが厳密ではないので、配列変数は必ず<個数>で指定する数と同じか、それ以上の要素数にしておく必要があります。取得する個数よりも配列要素数の方が少ないと、メモリを破壊してバグりますのでご注意下さい。<個数>を省略すると最大数取得します。先ほどの理由から、省略はあまりオススメしません。

4.4. IOTINIT

【書式】

CALL IOTINIT()

【機能】

MSX0 では、MSX-BASIC の出力をターミナルに流し込むために、MSX-BASIC や MSX-DOS で使われる文字表示ルーチンに細工をしています。CALL KANJI 等を使うと、この細工が壊れることがあり、その修復のための再初期化命令です。

5. デバイスノード

IOT 命令は、IOTINIT()以外の命令は全て第1引数に 〈デバイスノード〉 を指定するようになっています。〈デバイスノード〉 は、複数接続したり、いろいろな種類のデバイスを接続するので、どのデバイスを制御するのかを指定するキーワードです。

MSX0 に付属のドキュメント「Fig.5 2_IoT_BASIC ノード情報.pdf」に一覧表があります。

host/batteny/current host/batteny/level host/lesp host/lip host/lip host/name host/name host/power/off host/power/off host/power/reboot host/power/wait	整数	標要	単位	備考
host/battery/level host/heap host/ip host/media/dsk/* host/power/off host/power/reboot host/power/wait		パッテリー電流	mA	ボトムベース装着時は測定不可
host/ip host/media/dsk/* host/name host/power/off host/power/reboot host/power/wait	整数整数	パッテリー残量レベル ヒーブメモリ残量	%	
host/media/dsk/* host/name host/power/off host/power/reboot host/power/wait		ヒーブメモリ残量		
host/name host/power/off host/power/reboot host/power/wait	文字列	IPアドレス	-	
host/power/off host/power/reboot host/power/wait		ファイル名	-	
host/power/reboot host/power/wait	文字列	ホスト名	-	**************************************
host/power/wait	整数	システムシャットダウン	-	1を書き込むとシャットダウン 1を書き込むとシステム再起動
	整数	システム再起動 シャットダウンから起動するまでの待ち時間	-	1を書き込むとンステム再起動
	整数文字列	ンヤットダワンから起動するまでの待ち時間	sec	
host/sw_version	又干列	システムソフトウエアバージョン Wi-Fi アクセスポイントSSID	-	
host/wifi/aplist/*	数数	Wi-Fi アクセスポイントSSID Wi-Fi RSSI	-	
host/wifi/level			-	CONTRACTOR AND
host/wifi/restart	整数	Wi-Fi 設定值反映	\perp	1を書き込むと設定値を反映し、必要 に応じて再接続
device/accel/x	整数整数	加速度センサX 加速度センサY	-	パッテリーボトムなどIMUデバイス接
device/accel/y	整数	加速度センサY	-	
device/accel/z	整数	加速度センサZ アナログ入力(12bit)	-	
device/analog/in		アナログ人力(12bit)	-	
device/analog/out	整数	アナログ出力(12bit)	-	DHT20を利用する場合は記動時から
device/dht/humidity	整数	温度	%	PortAに接続しておいて下さい
device/dht/temperature	整数	温度	°C	
device/i2c_i/*	パイト列	i2c通信パッファ(メッセージボックス)		
msx/me/drive/a	文字列	ファイル名	_	msx/meは動作しているUnit自身
msx/me/id	整数	ユニット番号		
msx/me/if/NET0/conf/addr	文字列 整数	IPアドレス ボート番号		
msx/me/if/NET0/conf/port	整数	ボート書号		
msx/me/if/NET0/connect msx/me/if/NET0/msg	整数	接続状態、制御 メッセージボックス		
msx/me/if/NET0/msg	パイト列	メッセージボックス		
msx/me/kb	文字列	コンソール入力		
msx/me/pm/cpu/load	整数	Z80 CPUエミュレーション処理負荷	%	
msx/me/pm/opu/percent	整数	Z80 CPUL	%	
msx/me/pm/fps	整数	■ 東野フレームレート	fps	
msx/me/pm/reboot	整数 文字列	エミュレータの再起動 ディスクイメージファイル名		1を書き込むとエミュレータの再起動
msx/u0/drive/a		ディスクイメージファイル名	\vdash	
msx/u0/id	整数	ユニット番号	_	
msx/u0/if/NET0/conf/addr	文字列	IPアドレス	-	
msx/u0/if/NET0/conf/port	整数	ポート番号	-	
msx/u0/if/NET0/connect		接続状態、制御 メッセージボックス コンソール入力	-	
msx/u0/if/NET0/msg	パイト列	メッセージボックス	-	
msx/u0/kb	文字列	コンソール入力	-	
msx/u0/pm/opu/load	整数	Z80 CPUエミュレーション処理負荷 Z80 CPUレート		
msx/u0/pm/cpu/percent	整数	Z80 CPUL-F	3	
msx/u0/pm/fps	整数	280 CPOレート 画面更新フレームレート エミュレータの再起動	fps	
msx/u0/pm/reboot	文字列	エミエレーテの丹成園	-	1を書き込むとエミュレータの再起動
msx/u1/drive/a		ディスクイメージファイル名	-	
msx/u1/id msx/u1/if/NET0/conf/addr	整数	ユニット番号 IPアドレス	-	
msx/u1/ff/NET0/conf/addr		サート 英島	+	
msx/u1/if/NET0/conf/port msx/u1/if/NET0/connect	整数	ボート番号 接続状態、制御 メッセージボックス コンソール入力	-	
msx/u1/if/NET0/msg	17.71.70	March 25 March 7	-	
msx/u1/kb	バイト列 文字列	アッセーンホックス	-	
msx/u1/pm/opu/load	数数	Z80 CPUエミュレーション処理負荷 Z80 CPUレート	-	
msx/u1/pm/opu/percent	整数	780 CPUIL-b	4	
msx/u1/pm/fps	100 Sh	国面更新フレームレート	fps	
msx/u1/pm/reboot	整数整数	国団更新フレームレート エミュレータの再起動	1,00	1を書き込むとエミュレータの再起動
	整数	バックグラウンドUnitのZ80実行速度の割合		0:0%, 1:1%, 2:2%, 3:5%, 4:10%
conf/bg_cpu_perf	200		-	5:25%, 6:50%, 7:100%
conf/bg_snd_mix	整数	パックグラウンドUnitの音声出力ののON/OFF	-	
sonf/bk_color	整数	周围の周辺の色を強制的に黒	-	
conf/brightness	整数整数	画面の周辺の色を強制的に黒 LCDパックライトの明るさ UnitOのカートリッジ番号	-	0~10
conf/cartridge_id	整数	Unituのカートリッン番号	-	0:Zanac, 1:Paipanic 0:OFF, 1:M5LCD
	整数	メイン画面の出力先	-	U:OFF, 1:MSLCD
	文字列	Unit0のDiskイメージ Unit1のDiskイメージ	-	-
sonf/dsk0		Unt IOUSK1メーン	-	0.4 38 4 + + + - +
conf/dsk0 conf/dsk1	雅敦	液晶モード RAMのサイズ	+	0:ノーマル、1:左右反転 0:64K、1:128K、2:256K、3:512K、4:1M
conf/dsk0 conf/dsk1 conf/lcd_mode	BN 201			
conf/dsk0 conf/dsk1 conf/lod_mode conf/ram_size_id	整数	U王		C.O-H.C. T. T.E.OFC, E.E.O.OFC, G.D.T.E.I.C. 4.1181
conf/dsk0 conf/dsk1 conf/lod_mode conf/rem_size_id conf/remote/enable	整数 整数 整数	リモート機能のON/OFF 設定値保存		
conf/dsk0 conf/dsk1 conf/lod_mode conf/ram_size_id conf/remote/enable conf/save	整数整数	リモート機能のON/OFF 設定値保存		1を書き込むと設定値を保存
conf/dskl conf/dskl conf/odskl conf/nd mode conf/ram_size_id conf/remote/enable conf/save conf/setup	整数 整数 整数 整数	リモート機能のON/OFF 設定値保存 SetupUtility記載 音楽出力フィルタのON/OFF		
conf/dsk0 conf/dsk1 conf/lod_mode conf/ram_size_id conf/ram_size_id conf/ram_size_id conf/save conf/save conf/save	整数 整数 整数	リモート機能のON/OFF 設定値保存 SetupUtility記載 音楽出力フィルタのON/OFF		1を書き込むと設定値を保存
conf./dak0 conf./dak1 conf./da mode conf./ram.size id conf./ram.size id conf./ram.size id conf./save conf./save conf./save conf./save conf./save conf./save	整数整数	リモート機能のON/OFF 設定値保存 SetupUtility起動 資声出力フィルタののON/OFF 普量 ステータスモニタの表示モード		1を書き込むと設定値を保存 1を書き込むとSetupUtility記動 0~10 0:CFF, 1:SIMPLE, 2:SIMPLE(S),
conf./dak0 conf./dak1 conf./od.mode conf./am.size.jd conf./rams.eize.jd conf./ramsele conf./save conf./save conf./sound/colume conf./sound/rolume conf./sound/rolume	整数 整数 整数 整数 整数	リモー・機能のON/OFF 設定値保存 SetupUtility起動 音声出力フィルタののON/OFF 音量 ステータスモニタの表示モード MSSY2-72 - 1 タムア		1を書き込むと設定値を保存 1を書き込むとSetupUtility起動 0~10 0へOFF、ISIMPLE、2-SIMPLE(S), 0MSX、1-MSX2、2-MSX2+
conf./ dak0 conf./ dak1 conf./ lod, mode conf./ remote/ nable conf./ remote/ nable conf./ remote/ nable conf./ save conf./ save conf./ sound/ filter conf./ sound/ rotume conf./ sound/ rotume conf./ sound/ rotume	整数 整数 整数 整数 整数 整数	リモー・機能のON/OFF 設定値保存 SetupUtility起動 音声出力フィルタののON/OFF 音量 ステータスモニタの表示モード MSSY2-72 - 1 タムア		1を書き込むと設定値を保存 1を書き込むよSetupUtility記動 0~10 0.OFF, 1:SIMPLE, 2:SIMPLE(S), 0.MSX, 1:MSX2, 2:MSX2+ 0.OFF, 1:REYBOARD, 2:JOY1, 3:JOY
confriding id confridiskl confridiskl confridiskl confridiskl confrient id confrient	整数数数数数数数数数数数数数数数数数数数数数数数数数数数数数数数数数数数数	リモー・機能のON/OFF 設定値保存 SetupUtility起動 音声出力フィルタののON/OFF 音量 ステータスモニタの表示モード MSSY2-72 - 1 タムア		1支養を込むと設定値を保存 1支援を込むとSetupUtility記載 0∼10 0·OFF, 1:SIMPLE, 2:SIMPLE(S), 0·MSX, 1:MSX2, 2:MSX2+ 0·OFF, 1:KEYBOARD, 2:JOY1, 3:JOY 0·OFF, 1·○3
conff dek0 conff dek1 conff flod mode conff dek1 conff flod mode conff ram size jd conff ram size jd conff ramete/ enable conff save conff save conff save conff save conff sound/volume conff sound/volume conff system type conff size monitor conff system type conff size flog monitor conff size flog conff size same conff size flog conff size same conff size flog conff size same conff size flog confidence	整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整	リモー・機能のON/OFF 設定値保存 Setuplin/収録 管声出力フルタののON/OFF 自責 ステータスモニタの表示モード MSSシステムタイプ MSデームバッド接触設定 汎用ポートの連載の速さ 2月ポートの連載の速さ		1を書き込むと設定値を保存 1を書き込むとSetupUtility記載 0~10 COFF, 1:SIMPLE, 2:SIMPLE(S), CMSX, 1-MSX2, 2-MSX2+ COFF, 1:4でBOARD, 2:JOY1, 3:JOY 0:OFF, 1:~3
confl (dsN) confl	整数数数数数数数数数数数数数数数数数数数数数数数数数数数数数数数数数数数数	リモー・機能のON/OFF 設定値保存 Setuplin/収録 管声出力フルタののON/OFF 自責 ステータスモニタの表示モード MSSシステムタイプ MSデームバッド接触設定 汎用ポートの連載の速さ 2月ポートの連載の速さ		1を書き込むと設定値を保存 1を書き込むとSetupUtility記載 0~10 COFF, 1:SIMPLE, 2:SIMPLE(S), CMSX, 1-MSX2, 2-MSX2+ COFF, 1:4でBOARD, 2:JOY1, 3:JOY 0:OFF, 1:~3
confl dak0 confl dak1 confl flok mode conflok mode	整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整	リモール機能のON/OFF 記を開発性 を関係性 ままはカフィルタののON/OFF 日 ステータスモニタの表示モード MSVンステムターク MSV-ストルグト等を設定 ファールグト等を表示 ファールグトを表示 ファールグトをた		1を書き込むと設定値を保存 1を書き込むとSetupUtility記載 0~10 COFF, 1:SIMPLE, 2:SIMPLE(S), CMSX, 1-MSX2, 2-MSX2+ COFF, 1:4でBOARD, 2:JOY1, 3:JOY 0:OFF, 1:~3
confi (dell) confi	整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整	リモート機能のON/OFF 設定機能所 Schubling機 算用出力スルタののON/OFF スチータスモニタの表示モード MSシステムタンタイプ MSシステムメンド機能変 以用ボートAの運動の速き ファチック・ファナシットの上級を ファチック・ファナシットの上級を ののfiguration ドージョン		1を書き込むと設定値を保存 1を書き込むとSetupUtility記載 0~10 COFF, 1:SIMPLE, 2:SIMPLE(S), CMSX, 1-MSX2, 2-MSX2+ COFF, 1:4でBOARD, 2:JOY1, 3:JOY 0:OFF, 1:~3
confi (dell) confi	整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整	リモート機能のON/OFF 記で機能所 Sendoninを整 は		1を書き込むと設定値を保存 1を書き込むとSetupUtility記載 0~10 COFF, 1:SIMPLE, 2:SIMPLE(S), CMSX, 1-MSX2, 2-MSX2+ COFF, 1:4でBOARD, 2:JOY1, 3:JOY 0:OFF, 1:~3
conf (sigh) conf (整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整	リモート機能のON/OFF 最高地域の大力を使う。 直角地力イルタののN/OFF を ステークスモータの表示モード MS/S/ステム人が主義を受 月用ボートAの運動の連立 月用ボートAの運動の連立 ファチュッド(小洋繊維度 ファチュッド(小洋繊維度 つがたるのでは、ページョン (スワード)		1支援を込むと設定値を保存 1支援を込むと8年助比Hisが設置 0~F1 (0~F1 ISMPLE 2-SMPLES) 0MSX IMSX2 2MSX22 0MSX IMSX2 2MSX22 0MSX IMSX2 2MSX22 0MSF I I I STOCK (0 ISMPLES) 0.0FF I I I STOCK (0 ISMPLES) 0.0FF I I I I I I I I I I I I I I I I I I
conf (sist) conf (整整	リモート集他のON/OFF 記で選修符 Setudiniの複整 当のフェイルがののON/OFF ステータスモニタの表示モード MSジースナルタークラット MSジーストルタード MSデーストルタード MSデーストルタード MSデーストルタード MSデーストルタード ファナルタード ファナルタード アンナルタード アンナルタード アンナルタード アンナルタード アンナルタード アンナルタード アンナルタード アンナルタード アンナルタード アンナルタード アンナルタード アンチルタート アンチルタート アン		1支援を込むと設定値を保存 1支援を込むと8年助比Hisが設置 0~F1 (0~F1 ISMPLE 2-SMPLES) 0MSX IMSX2 2MSX22 0MSX IMSX2 2MSX22 0MSX IMSX2 2MSX22 0MSF I I I STOCK (0 ISMPLES) 0.0FF I I I STOCK (0 ISMPLES) 0.0FF I I I I I I I I I I I I I I I I I I
conf (sist) conf (整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整	リモート機能のON(OFF 起き機能が 起きを発生が まま出 フィルタののONOFF 日 ステータスモニタの表示モード MSVニステムタープ MSVニストムマード MSVニストムマード MSVニストムマード MSVニストムマード MSVニストムマード MSVニストムマード MSVニストムマード MSVニストムマード ファチルマード ファチルマード アンディンド・ ドルロード アンディンド・ アンディン・		1支援を込むと設定値を保存 1支援を込むと8年助比Hisが設置 0~F1 (0~F1 ISMPLE 2-SMPLES) 0MSX IMSX2 2MSX22 0MSX IMSX2 2MSX22 0MSX IMSX2 2MSX22 0MSF I I I STOCK (0 ISMPLES) 0.0FF I I I STOCK (0 ISMPLES) 0.0FF I I I I I I I I I I I I I I I I I I
ond (side) on off office of the control of the cont	整数数数数数数数数数数数数数数数数数数数数数数数数数数数数数数数数数数数数	リモート機能のON/OFF 単原出力でルグののON/OFF 単原出力フィルグののON/OFF 2		1支援を込むと設定値を保存 1支援を込むと8年助比Hisが設置 0~F1 (0~F1 ISMPLE 2-SMPLES) 0MSX IMSX2 2MSX22 0MSX IMSX2 2MSX22 0MSX IMSX2 2MSX22 0MSF I I I STOCK (0 ISMPLES) 0.0FF I I I STOCK (0 ISMPLES) 0.0FF I I I I I I I I I I I I I I I I I I
ond (side) of confidence of co	整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整整	リモート機能のON(OFF 起き機能が 起きを発生が まま出 フィルタののONOFF 日 ステータスモニタの表示モード MSVニステムタープ MSVニストムマード MSVニストムマード MSVニストムマード MSVニストムマード MSVニストムマード MSVニストムマード MSVニストムマード MSVニストムマード ファチルマード ファチルマード アンディンド・ ドルロード アンディンド・ アンディン・		1支援を込むと設定信を保存 1支援を込むとSetupUtilit/投数 0-0F 1 SSMPLE 2-SMPLE(S) 0MSX 1MSX2 2MSX2- 0MSX 1MSX2- 0MSX 1MSX2- 0MSX2- 0MSX 1MSX2- 0MSX 1MSX2- 0M

Fig.5 2_IoT_BASICノード情報.pdf

〈デバイスノード〉 によって、IOTPUT のみ対応・IOTGET のみ対応・両方対応などの差があるのですが、残念ながら現状その記載はありません。概要に記載されている内容から推測したり、あるいは実際に試してみて確認することになります。

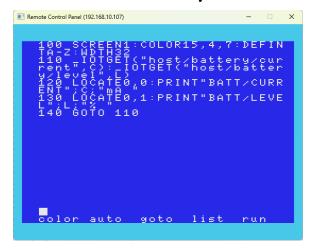
例えば、ノード名「host/battery/level」ですが、バッテリー残量を読み取るのみですので、CALL IOTGET()のみ対応です。

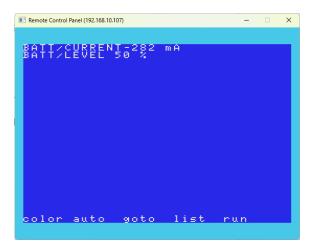
接続した Grove デバイス(センサーや表示装置など)だけでなく、MSX0 本体に内蔵されているセンサーの情報を取得したり、MSX0 の MSX エミュレーターの情報取得・設定なども全てこのデバイスノードを使って _IOTGET(), _IOTPUT(), _IOTFIND() で制御していきます。使う命令の種類は少ないですが、デバイスノードが多いので間違えないようにご注意下さい。

6. 制御例

本章は、簡単なサンプルプログラムと、それによってどんな情報が得られるのか、あるいはどのように作用するのか、を説明する章です。興味を持った作例は、実際にサンプルプログラムを入力して、動作を確認してみて下さい。「こんなことが出来る」というサンプル集にもなっています。

6.1. バッテリー電流/レベルを取得する



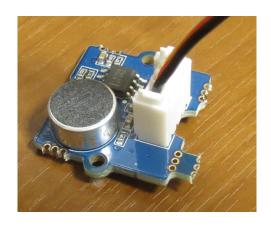


```
100 SCREEN1:COLOR15,4,7:DEFINTA-Z:WIDTH32
110
_IOTGET("host/battery/current",C):_IOTGET("host/battery/leve
l",L)
120 LOCATE0,0:PRINT"BATT/CURRENT";C;"mA "
130 LOCATE0,1:PRINT"BATT/LEVEL";L;"% "
140 GOTO 110
```

行 110 で デバイスパス "host/battery/current" から値を取得して変数 C へ、デバイスパス "host/battery/level" から値を取得して変数 L へ格納。 行 $120\sim130$ でそれらの値を表示。 行 140 で行 110 へ戻る。

IOTGET で簡単にバッテリーレベルを得られるので、バッテリーレベルに応じて何か表示を変えたりすることも容易に実現できます。

6.2. 音センサー



Grove Beggner Kit For Arduino の中から、MIC Grove を選択します。

コネクタの付け根に SIG, NC, VCC, GND と書かれています。SIG は MIC の制御信号。 NC は No Connect で未接続の意味。VCC は電源 +5V です。これに該当する M5 の Grove は、黒(Port.B)です。MSX0 の黒い Grove 端子に接続して下さい。

_IOTGET("device/analog/in",L) を実行することで、変数 L にマイクの入力レベルが $0\sim4095$ で得られます。音を録音するほど高速ではないので、手を叩いたか検知する、等の「大きな音をトリガーにする」ような使い方が良いかと思います。

デバイスノード "device/analog/in" は、 $Grove \ o \ 1$ 番ピン (MIC $Grove \ o \ SIG \ と書かれたピン) に入力されてくる信号を <math>A/D$ 変換し、 $0\sim4095$ 値としてゲットできます。

ちなみに、 $_{10TPUT}$ ("device/analog/out", D) による値 D の出力ですが、こちらは Grove の $_{2}$ 番ピンへ出力されます。 $_{1}$ 番ピンに接続されている出力デバイスには使えないのでご注意下さい。

100 SCREEN1:DEFINTA-Z:WIDTH32

110 _IOTGET("device/analog/in",L)

120 PRINT L:GOTO 110



動いている最中に手を叩くと、表示されている数値が突如 4095 に跳ね上がる。そしてすぐに元に戻る。変数に得られた値が所定の数値以上か、未満かで処理を分けることで、現実世界の音とリンクした動作が出来るわけですね。