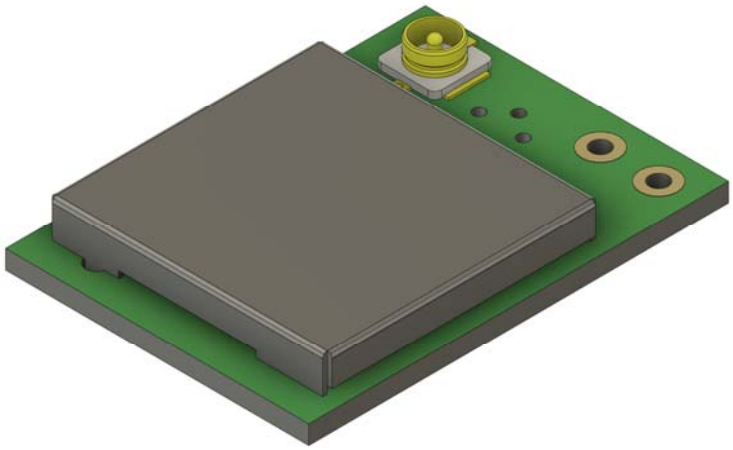
V1.12

特定小電力 LoRa/FSK 無線モジュール

ソフトウェア

リファレンスマニュアル

LRA1



株式会社アイ・ツー

〒279-0001　千葉県浦安市当代島 2-9-30

TEL:047-711-0914　　FAX:047-711-0915

お問合せ先:info@i2-ele.co.jp

1. 改版履歴

Revision	日付	内容
1.0	2019/12/24	初版
1.1	2020/03/30	機能追加による加筆、修正
1.11	2020/04/02	加筆、LCD コントラスト設定追加
1.12	2020/05/13	9.予約変数と定数の章、TICK/CKOCK の説明変更

2. 目次

3. 概要

LRA1 モジュールには BASIC 言語の機能を簡略化(サブセット化)した Tiny BASIC の LRA1-BASIC を搭載しています。

この LRA1-BASIC には LoRa 無線およびいくつかの周辺機器を制御する機能が追加されています。

機能を簡略化しているために一般的な BASIC 言語とは仕様が異なる部分もあります。

本リファレンスマニュアルは ファームウェアの Ver.0.33.b に対応しています。

4. 接続

4.1 PC との接続

LRA1 モジュールと PC をシリアルポート経由で接続し、TeraTerm 等のシリアル通信ソフトからテキストベースで制御します。

デフォルトのシリアルポート通信速度は 115200bps です。

本リファレンスマニュアルでは、TeraTerm を使用した場合で解説します。

4.2 外部マイコンとの接続

外部のマイコンから本モジュールを使用する場合も、PC の TeraTerm からと同様にシリアルポートで制御することになります。

ただし、単に LoRa 送受信だけを行う場合は、起動時に「RECV」や「COMM」コマンドを実行することで、外部マイコンとは送受信データーのみのやり取りにすることも可能です。(詳細は各コマンドの解説を参照)

4.3 外部ペリフェラルとの接続

LRA1-BASIC には評価ボードに搭載されている外部ペリフェラル(BME280、LCD)を制御する機能を実装しています。

他のペリフェラルの使用は想定されていません。

5. ユーザーインタフェース

5.1 文字列行の入力

コマンド等の文字列が入力可能な状態になると、コマンドプロンプト「＞」が表示されます。

シリアルポートから入力された文字列は CR(改行)までを1行とします。

命令文および変数名は大文字/小文字を区別しません。

コメントおよび文字列中の内容は大文字小文字が区別され、入力したとおりとなります。

1行の文字数は 254 文字以内です。254 文字を超えた入力は無視されます。

5.2 インタプリタ

LRA1-BASIC はインタプリタ方式です。

入力された文字列は中間言語のバイトコードに変換し、各コマンドの処理を順次実行します。または、中間言語に変換されたバイトコードをプログラム領域に格納します。

一般的に、入力された命令文字列よりも中間言語への変換後の方がバイト数は少なくなります。

バイトコードを格納するプログラム領域のサイズは 4096 バイトです。プログラム領域は RAM 上にあるので、システムリセットや電源断によって消去されますが、FLASH メモリーへの書き込み/読み出しが可能です。

5.3 動作モード

動作モードには、「インタラクティブモード」と「プログラム実行モード」があります。

・インタラクティブモード:

シリアルポートからの入力文字列を命令文として、1行毎に対話形式でコマンドを実行します。

・プログラム実行モード　:

RAM 上のプログラムエリアにあるプログラムを順次実行します。

LRA1 モジュールの起動直後はダイレクトモードです。

AUTO 変数の設定によって、起動直後に指定のコマンドを自動実行することも可能です。

6. 言語仕様

6.1 変数

一般変数 ： A～Z のアルファベット1文字で表わします。(26 個)

配列変数 ： @(添字)で表される1次元配列が1つ。(添字の範囲は 0～255)

添字には式を使用できます。

特殊変数 ： 特別な用途の変数(詳細は各変数の項目を参照)

文字列変数:ありません。

全ての変数はグローバル変数で 32 ビット符号付整数となります。ローカル変数は存在しません。

6.2 数値定数

使用可能な数値定数は符号付き 32 ビット整数のみ。

数値にマイナス符号は使用できるが符号としてのプラス符号は使用できません。

デフォルトの数値は 10 進数で、数値の前に「＄」をつけると 16 進数となる。

6.3 文字列定数

文字列はダブルクォーテーションで囲みます。

PRINT 文などの一部のコマンドで文字列が使用できます。

変数には文字列を設定できません。

6.4 式と演算

6.4.1 式

数値定数および以下の演算子を使用したものが「式」となります。

演算中にオーバーフローしてもエラーにはなりませんが、0で除算した場合はエラーになります。演算には次の演算子を使用します。演算の順番は一般的な BASIC 言語と同様です。

6.4.2 算術演算子

以下の算術演算子が使用できます。

演算子	演算	使用例
＋	加算	A=B+C
－	減算	A=B-C
*	乗算	A=B*C
／	除算	A=B/C
%	剰余	A=B%C

6.4.3 単項演算子

以下の単項演算子が使用できます。

演算子	演算	使用例
—	符号反転	A=—B
～	ビット反転	A=～B
！	否定(NOT)	A=！B

6.4.4 論理演算子

以下の論理演算子が使用できます。

演算子	演算	使用例
＆＆	論理積 (AND)	If A=B && A=C Then
｜｜	論理和 (OR)	If A=B || A=C Then

6.4.5 ビット演算子

以下のビット演算子が使用できます。

演算子	演算	使用例
&	論理積(AND)	A=B&C
｜	論理和(OR)	A=B|C
^	排他的論理和(XOR)	A=B^C

6.4.6 比較演算子

以下の比較演算子が使用できます。

演算子	演算	使用例
=	一致	If A=B Then
<>	不一致	If A<>B Then
<	未満	If A<B Then
<=	以下	If A<=B Then
>	超過	If A>B Then
>=	以上	If A>=B Then

比較結果が真のときは1で偽のときは0となる。

6.4.7 その他

( ) : 括弧

カッコ内の演算が優先されます。

6.5 代入分

「変数 = 式」のように、変数名に続けて「=」の後に式を記述します。

式の演算結果を変数に代入します。

変数には一般の変数(A～Z)、配列変数(@)、LoRa 変数、予約変数を使用できます。

(一般的な BASIC にあるような LET 命令はありません。)

6.6 マルチステートメント文

命令文は、「:」(コロン)でつなげてマルチステートメントが可能です。

ただし、1行は 254 文字または、中間コードに変換したときに 254byte 以下で、それを超えるとエラーとなります。

6.7 行番号 (Line number)

プログラム各行の先頭につけられた番号です。すべてのプログラム行には行番号が必要です。

行番号は0～65535 の数値です。

プログラムは行番号の順番に実行されます。

プログラムを入力したときは、行番号順に並び変えられます。

6.8 ラベル

プログラムの GOTO や GOSUB などの分岐先には行番号の他にラベル文字列を使うことができます。

本 LRA1-BASIC では「__(アンダーバー)」から始まる文字列をラベルとして扱います。

ラベルは行番号直後のみに設定できる。行の途中にラベルがあってもエラーにはならないがジャンプ先としては無効です。

プログラム中に同一のラベル文字列が存在した場合、プログラムの先頭に近いほうが有効とります。

行番号はすべてのプログラム行に付くことになりますが、ラベルは必要な行のみにつけます。

ラベル文字列の大文字/小文字は区別されます。

6.9 コメント

「'（シングルクォート）」以降の内容が行末までコメントとなります。

コメント内容によってプログラムの動作に影響しません。

6.10 動作の中断

シリアルポートから「Break 信号」または「Break キャラクタ」を受信すると、プログラムおよびコマンドを中断します。

中断したプログラム動作は RESUME 命令で中断箇所から再開することができます。

ただし、エラー等で処理が中断した場合は RESUME 命令での再開はできません。

※「Break 信号」と「Break キャラクタ」

BRKCH 変数が 0 のときは「Break 信号」、0 以外のときは BRKCH 変数に設定されている ASCII コードの「Break キャラクタ」が有効となります。BRKCH 変数のデフォルト値は「0x03(Ctrl-C)」です。

0以外の設定のとき、「Break 信号」は無効です。「Break 信号」と「Break キャラクタ」の両方を有効にすることはできません。

6.11 入れ子

GOSUB～RETURN、FOR～NEXT、DO/WHILE～LOOP などの入れ子のスタックはすべて共通で 16 段階までです。入れ子が 16 段階を超えると、スタックオーバーフローエラーとなります。

7. BASIC コマンド

7.1 分岐命令

7.1.1 GOTO

書式：GOTO [ラベル]

ラベルに指定された行にジャンプする。

ラベルには行番号またはラベル文字列を指定する。

ジャンプ先に行番号を指定する場合は、数値だけでなく変数を使用した式も指定可能。

7.1.2 GOSUB

書式:GOSUB [ラベル]

ラベルに指定された行のサブルーチンにジャンプします。

ラベルの指定は GOTO と同様。

サブルーチンの RETURN 命令で、本命令の次の命令に戻ります。

7.1.3 RETURN

書式:RETURN

サブルーチンから戻ります。

7.2 条件分岐、繰り返し

7.2.1 IF～THEN～ELSEIF～THEN～ELSE～ENDIF

書式:IF [式] THEN ～ ELSEIF [式] THEN ～ ELSE ～ ENDIF

式の演算結果が0以外「真」のときは THEN に続く命令文を実行します。

式の演算結果が0「偽」のときは ELSEIF の条件判定を、ELSEIF が省略された場合は ELSE に続く命令文を実行します。

ELSEIF、ELSE は省略できます。(IF の後には、THEN と ENDIF が必要)

THEN、ELSE 直後に行番号またはラベルを記載すると、GOTO を記述したときと同様の動作となります。

7.2.2 FOR～TO～STEP～NEXT

書式:FOR [変数名=式1] TO [式2] STEP [式3] ～ NEXT

変数名に式1を代入した後、変数が式2になるまで NEXT との間をループします。

プログラムが NEXT に来る毎に、変数に式3の値が加算されます。

STEP 式3は省略可能で、省略された場合は1が指定されたことになります。

7.2.3 DO～LOOP

書式:DO ～ LOOP

DO～LOOP を繰り返します。

7.2.4 DO～LOOP WHILE

書式:DO ～ LOOP WHILE [式]

DO～LOOP 間を実行した後に式の判定をして、式の結果が 0 以外「真」のとき、DO～LOOP WHILE を繰り返します。

7.2.5 WHILE～LOOP

書式:WHILE [式] ～ LOOP

式の結果が 0 以外「真」のとき、WHILE～LOOP を繰り返します。

7.2.6 EXIT

書式:EXIT

FOR～NEXT、DO/WHILE～LOOP の繰り返し処理から抜け出ます。

NEXT または LOOP(LOOP WHILE)の直後の命令に処理を移します。

入れ子の場合は一番近いループにのみ適用されます。

7.2.7 CONTINUE

書式:CONTINUE

FOR～NEXT、DO/WHILE～LOOP のループ内での処理中に、本命令以降の処理をスキップして再び繰り返し処理を実行します。

入れ子の場合は一番近いループにのみ適用されます。

7.3 入出力コマンド

7.3.1 PRINT、?

書式:PRINT [書式] または ?[書式]

数値, 文字列を出力します。"PRINT"のかわりに "?" を代用できます。

書式には1つまたは複数の式または文字列を指定できます。

数値や式は文字列に変換されます。

複数の式、文字列はコンマやセミコロンで区切って指定します。

コンマで区切るとタブ出力になり、セミコロンで区切ると続けて出力します。

最後をセミコロンで終了すると改行しません。

PRINT 文中では、文字列、CHR 関数、FORM 関数などが利用できます。

7.3.2 INPUT

書式:INPUT [変数]

シリアルポートからの入力を待ち、入力された数値を変数に格納する。

入力する数値の先頭が＄の場合は、16進数として扱います。

変数名の後ろに＄をつけると、入力された1文字目の ASCII コードを変数に格納する。

7.3.3 OUTP

書式:OUTP [式1], [式2]

GPIO のポートに出力します。

式 1 にはポート番号を指定します。(下表にあるポート番号のみ指定可能です。)

式2がOの時はポートに Low を出力、O以外の時は High を出力します。

ポート	名前	ピン
0	PA00	Pin.32
1	PA01	Pin.31
6	PA06	Pin.7
7	PA07	Pin.8
8	PA08	Pin.9
9	PA09	Pin.14
13	PA13	Pin.15
14	PA14	Pin.24

ポート	名前	ピン
15	PA15	Pin.23
16	PA16	Pin.16
17	PA17	Pin.17
18	PA18	Pin.18
19	PA19	Pin.25
22	PA22	Pin.26
23	PA23	Pin.20
24	PA24	Pin.28

ポート	名前	ピン
25	PA25	Pin.27
27	PA27	Pin.5
28	PA28	Pin.19
34	PB02	Pin.1
35	PB03	Pin.4
54	PB22	Pin.6
55	PB23	Pin.21

本関数を実行すると指定されたポートは出力ポートに設定されます。

PA06、PA14 などシステムで使用するポートについてはできるだけ使用しないようにしてください。

7.4 実行制御コマンド

7.4.1 END

書式:END

プログラムの実行を終了して、ダイレクトモードになります。

RESUME での再開はできません。

7.4.2 STOP

書式:STOP

プログラムの実行を中断します。

「Break 信号」まはは「Break キャラクタ」でのプログラムの中断と同様です。

RESUME 命令で、この BREAK の次の命令からの再開が可能です。

7.4.3 RESUME

書式:RESUME

STOP 命令または、シリアルポートからの「Break 信号」まはは「Break キャラクタ」で中断した処理を再開します。

エラー発生によって中断した場合は RESUME 命令で再開することはできません。

7.4.4 PAUSE

書式:PAUSE

シリアルポートから1文字が入力されるまで待ちます。

シリアルポートからの Break によって中断が可能です。(プログラムも中断します。)

7.4.5 DELAY

書式:DELAY [式]

式で指定した時間(1msec 単位)だけ待ちます。

シリアルポートからの「Break 信号」まはは「Break キャラクタ」によって中断が可能です。この場合、プログラムも中断します。

7.4.6 SLEEP

書式:SLEEP [式]

省電力モードで待機します。

[式]で指定された値によって以下のように動作します。

式の値	Sleep 解除時間	PA06 ピンの変化
＞ 0	[式]で指定された値の絶対値の時	無効
＜ 0	間(秒単位)が経過	Low→High の変化で Sleep 解除
＝ 0	無効	

「Break 信号」や「Ctrl-C」では解除されません。

Sleep コマンドで PA06 ピンが有効に指定されると、INP 関数や OUTP コマンドにかかわらず、強制的に入力ポートに再設定されます。Sleep コマンドで PA06 を使用する場合は、他の目的で PA06 を使用しないようにしてください。

7.4.7 DEEP

書式:DEEP [式]

SLEEP よりもさらに消費電流を落とした、超省電力モード(Deep Sleep)で待機します。

ただし、SLEEP コマンドと違い Deep Sleep が解除されるとシステムリセットとなります。

[式]で指定された値によって以下のように動作します。

式の値	Sleep 解除時間	PA06 ピンの変化
＞ 0	[式]で指定された値の絶対値の時	無効
＜ 0	間(秒単位)が経過	Low→High の変化で Deep Sleep 解除
＝ 0	無効	

「Break 信号」や「Ctrl-C」では解除されません。

Sleep コマンドで PA06 ピンが有効に指定されると、INP 関数や OUTP コマンドにかかわらず、強制的に入力ポートに再設定されます。Sleep コマンドで PA06 を使用する場合は、他の目的で PA06 を使用しないようにしてください。

7.4.8 RESET

書式:RESET

システムリセットを実行します。

7.5 データー関連コマンド

7.5.1 DATA

書式:DATA [式1], [式2]・・・

READ コマンドで読み込む数値を記載します。(数値だけでなく、演算式も記述可能です。)

1つ以上の式をカンマで区切って記述します。

READ 命令の位置とは関係なく、プログラム中のどこでも配置できます。

7.5.2 READ

書式:READ [変数]

DATA コマンドで記載された式の数値を、変数に読み込みます。

本コマンドを実行するたびに、DATA 文に設定された数値をプログラムの先頭に近い順から読み込まれます。

読み込む DATA が無くなってから READ するとエラーとなります。

7.5.3 RESTORE

書式:RESTORE

READ 命令で読み込む DATA 命令の順番を、プログラムの先頭に戻します。

7.6 その他のコマンド

7.6.1 DEVID

書式:DEVID

LRA1 モジュール毎に固有のデバイス ID を表示します。

7.6.2 RANDOMIZE

書式:RANDOMIZE [式]

乱数の種を設定します。

式で与えられた値で、RND()関数が生成する乱数の種を設定します。

式が省略された場合は0が指定されたものとします。

この乱数の種は、デバイス ID から計算した値を使用するので、式の値が同じでも LRA1 モジュールの個体毎に異なる値となります。

システム起動時は RANDOMIZE 0 が実行された状態です。

7.6.3 VER

書式:VER

ソフトウェアのバージョンを表示します。

7.7 プログラム操作コマンド

7.7.1 EDIT

書式:EDIT [式]

プログラム入力モードの禁止/許可を設定します。

式が0のときまたは省略されたとき「プログラム入力禁止モード」になります。

式が0以外の時は「プログラム入力許可モード」になります。

システム起動時は「プログラム入力禁止モード」です。

「プログラム入力禁止モード」:

このモードはダイレクトモードでのコマンドプロンプトが「>」になります。

ダイレクトモードで、数値から始まる行が入力されてもプログラム行の書き込みはエラーになります。またプログラム操作関連のいくつかのコマンドがエラーになります。

不用意な数値入力によるプログラムの誤消去や誤登録を防止することができます。

「プログラム入力許可モード」:

このモードはダイレクトモードでのコマンドプロンプトが「>>」になります。

ダイレクトモードで、数値から始まる行が入力されると、プログラム行の書き込みとなります。

インデントや不要なスペース等は無視され、コメントと文字列以外は大文字小文字の区別はありません。プログラム行が書き込まれると、次行にコマンドプロンプトが表示されます。「OK」は表示されません。

入力1行毎に中間コードへの変換が行われるので、連続して行を入力する場合は、コマンドプロンプトが表示されてから次行を入力してください。

また、中間コードへの変換時に文法エラー等があった場合は、その行はプログラムエリアに書き込まれずにエラーメッセージを表示します。

入力されたプログラム行は、行番号の順番にプログラムエリアに書き込まれます。

入力された行番号と同じ行番号のプログラム行が既に書き込まれている場合は、新たに入力されたプログラム行に置き換えられます。行番号に続くコマンド等が無く行番号だけの入力のときは、その行番号の行は削除されます。

7.7.2 NEW

書式:NEW

プログラムと変数をクリアします。

プログラムエリアのプログラムを全て削除します。

各変数も初期化されますが、特殊変数は初期化されません。

7.7.3 RENUM

書式:RENUM [式]

プログラムの行番号を付け替えます。

プログラムエリアのプログラム行の行番号を、式で指定された番号からはじまって 10 ずつ増加するように付け替えます。

式は省略可能で、省略時は 10 と仮定されます。

このコマンドでは行番号のみを付け替えるので、GOTO や GOSUB の飛び先はそのままとなります。

7.7.4 DELETE

書式:DELETE [式1], [式2]

プログラムを削除します。

式 1 から式 2 までの行番号のプログラム行が削除されます。

式 2 が省略された場合は、式 1 からプログラムの最後までが削除されます。

7.7.5 PLOAD

書式:PLOAD

FLASH メモリーに保存されたプログラムをプログラムエリアに読み込みます。

RAM 上にあるプログラムは上書きされます。

7.7.6 PSAVE

書式:PSAVE

プログラムエリアのプログラムを FLASH メモリーに保存します。

FLASH メモリーに保存することができるプログラムは1つのみです。

FLASH メモリーに保存したプログラムは、電源断やシステムリセットでも消去されません。

7.7.7 RUN

書式:RUN [ラベル]

プログラムエリアのプログラムを実行します。

[ラベル]を省略した場合は、プログラムを先頭から実行します。

[ラベル]を指定した場合は、指定した行番号またはラベルからプログラムを実行します。

実行に先立って、各変数は0にクリアされます。(LoRa 設定値等を除く)

7.7.8 LIST

書式:LIST [ラベル]

プログラムエリアのプログラムを表示します。

[ラベル]を省略した場合は、プログラムの先頭から出力します。

[ラベル]を指定した場合は、指定した行番号またはラベル以降のプログラムを出力します。

8. BASIC 関数

8.1 一般関数

8.1.1 RND

書式：RND(式)

0以上の式を超えない値の乱数が戻ります。

8.1.2 ABS

書式：ABS(式)

式の絶対値が戻ります。

8.1.3 INP

書式：INP(式)

GPIO のポートの状態を取得します。

式にはポート番号を指定します。(下表にあるポート番号のみ指定可能です。)

戻り値が0の時はポートが Low、1の時は High が入力されています。

ポート	名前	ピン
0	PA00	Pin.32
1	PA01	Pin.31
6	PA06	Pin.7
7	PA07	Pin.8
8	PA08	Pin.9
9	PA09	Pin.14
13	PA13	Pin.15
14	PA14	Pin.24

ポート	名前	ピン
15	PA15	Pin.23
16	PA16	Pin.16
17	PA17	Pin.17
18	PA18	Pin.18
19	PA19	Pin.25
22	PA22	Pin.26
23	PA23	Pin.20
24	PA24	Pin.28

ポート	名前	ピン
25	PA25	Pin.27
27	PA27	Pin.5
28	PA28	Pin.19
34	PB02	Pin.1
35	PB03	Pin.4
54	PB22	Pin.6
55	PB23	Pin.21

本関数を実行すると指定されたポートは入力ポートに設定されます。

PA06、PA14 などシステムで使用するポートについてはできるだけ使用しないようにしてください。

8.1.4 ADC

書式：ADC(式)

ADC ポートから AD 変換された値を取得します。

式には下表に記載のポート番号のみ指定可能です。(それ以外の指定はエラーとなります)

ポート	入力ソース	戻り値	リファレンス
6	PA06/pin-7	0～4095 ADC 変換の値がそのまま戻ります。	VDD
7	PA07/pin-8		
10	PB02/pin-1		

11	PB03/pin-4		
16	PA08/pin-9		
17	PA09/pin-14		
24	CPU 内部の温度センサの ADC 値		
25	Bandgap 電圧(1.0V)の ADC 値		
26	CPU コア電圧(Typ:0.9～1.2V)	1mV 単位の計測値	内部 Bandgap (1.0V)
27	VDD 電圧	(誤差があります)	

入力ソースが 6～17 の外部ピンの場合は、OUTP コマンドや INP 関数の設定にかかわらず、この関数実行時にアナログ入力に設定されます。

8.2 文字列関数

文字列関数は数値式には使用できません。文字列用の変数もありません。

ただし、PRINT、LPINT、TXD、SEND コマンドの引数などの文字列の設定や出力するところでは文字列が使用可能です。

8.2.1 CHR

書式:CHR(式)

式の下位 8 ビットをそのまま文字列中に出力します。

PRINT コマンドなどで特定のキャラクタコードを利用したい場合などに使用します。

8.2.2 WCHR

書式:WCHR(式)

式の下位16ビット(Little endian)をそのまま文字列中に出力します。

PRINT コマンドなどで Word サイズのバイナリデーターを利用したい場合などに使用します。

8.2.3 FORM

書式:FORM(書式文字列、式)

書式文字列の指定に従って、式の数値を文字列に出力します。

書式文字列で使用する書式指定文字は以下のとおり。

D :10 進数(デフォルト)

X :16 進数(0～9、A～F)

x :16 進数(0～9、a～f)

0 :リーディングゼロ(無指定の場合は空白文字)

1～9:出力桁数指定(無指定の場合は可変長)

- :符号用に1文字分を左端に追加します。

+ :正数の場合に＋符号を付加します。

無指定の場合、デフォルトの 10 進数・可変長となります。

例:次のように使用する

```
Print Form("",-1234)      -> “-1234” を表示
Print Form("X04", 1234)   -> “04D2“ を表示
Lprint Form("-05", 1234)  -> “01234” を LCD 表示
Send Form("-5",-1234)     -> “- 1234” を LoRa で送信
Txd=Form("-+5",1234)      -> “+ 1234” を LoRa 送信バッファに設定
```

9. 予約変数と定数

本 BASIC には、システムで予約された変数と定数があります。

以下の変数は参照のみ可能です。

変数名	内容	説明
TICK	システムカウント値	システム起動から 1/1000 秒毎にカウントアップされる値です。 リセット、電源 ON、DEEP からの復帰時にクリアされます。
CLOCK	RTC クロック値	システム起動から 1 秒毎にカウントアップされる値です。 リセットと電源 ON でクリアされます。 DEEP からの復帰時はクリアされずに継続します。
INKEY	入力文字	シリアルポートからの入力された文字の ASCII コードです。 入力がない場合は0となります。

以下の変数は参照と設定が可能で、SSAVE で Flash メモリーに保存し、SLOAD で読み出すことができます。

変数名	内容	説明
ECHO	エコーバック	シリアルポートに入力された文字のエコーバックを指定します。 0:エコーバックなし、0以外:エコーバックあり。SSAVE/SLOAD コマンドでの保存/復元ができます。 デフォルト値は1です。
AUTO	自動実行文字列	システム起動時に自動実行する BASIC コマンドを文字列で設定します。マルチステートメントでの記述が可能です。設定できる文字列の最大文字数は63文字です。 SSAVE/SLOAD コマンドで保存/復元ができます。 デフォルト値は空白です。
BAUD	ボーレート	シリアルポートの転送速度を設定、参照します。 この変数を設定すると、即時にシリアルポートの設定に反映されます。設定可能な値は次の通りで、それ以外はエラーです。 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 38400, 76800, 115200 SSAVE/SLOAD コマンドで保存/復元ができます。 デフォルト値は 115200 です。
BRKCH	Break キャラクタ	プログラムの動作を中断するために使用する「Break キャラクタ」の ASCII コードを設定します。 0のときは「Break 信号」となります。 0以外に設定されている場合は、Break 信号は無効です デフォルト値は 「0x03 (Ctrl-C)」です。

以下の定数は固定値で、参照のみ可能です。

定数名	内容	説明
FALSE	偽	0と同等
TRUE	真	1と同等

10. エラーメッセージ

エラーが発生したときは、以下のエラーメッセージを表示して処理を中止します。

プログラムモードのときは、エラーが発生した行の行番号も表示します。

エラーメッセージ	内容
Syntax error	文法エラーです。認識できない命令文や記述ミスがあります。
Division by 0 error	0で除算しました。
Array index over error	配列の添え字の範囲が負数または 255 を超えています。 バッファの指定範囲を超えたときもこのエラーとなります。
Parameter error	パラメーターが必要な命令にパラメーターが指定されていません。 または、パラメーターの範囲を超えた値が指定されました。
Stack overflow error	スタックがオーバーしました GOSUB,FOR,DO,WHILE の入れ子が8段を超えました。
Can't resume error	プログラムを RESUME 命令で再開できません。 処理がエラーで中止されたときは再開ができません。
Label not found error	GOTO,GOSUB の指定先のラベル番号が見つかりませんでした。
Unless from pg-mode error	プログラムモードでは実行できない命令文です。 LOAD,NEW,LIST,RUN,RESUME 命令は、プログラムモードで実行できません。
Program area overflow error	プログラム用メモリーの容量がオーバーしました。
Loop nothing error	DO/WHILE 命令で、対応する LOOP 命令がありません。
Endif not found error	IF 命令に対応する ENDIF 命令がありません。
Device access error	デバイス(LCD,BME280 など)のアクセスエラーです。
Edit mode error	プログラム編集モードではありません。
Unexpected Next error	NEXT 命令に対応する FOR 命令がありません。
Unexpected Return error	RETURN 命令に対応する GOSUB 命令がありません。
Unexpected Loop error	LOOP 命令に対応する DO/WHILE 命令がありません。
Unexpected Exit error	EXIT 命令に対応する FOR または DO/WHILE 命令がありません。
Unexpected Continue error	CONTINUE 命令に対応する FOR または DO/WHILE 命令がありません。

11. LCD 制御

以下のコマンドと変数を使用して評価ボード上の LCD に文字を表示することができます。

I2C で LCD が接続されている必要があります。

11.1 LCD コマンド

11.1.1 LCLR

書式 : Lclr

LCD の表示をクリアします。

11.1.2 LPRINT

書式 : LPrint [引数]

LCD に文字列を表示します。

引数は PRINT コマンドに準拠しますが、改行やタブは無視されます。

表示が1行の桁数を超えても改行しません。

11.2 LCD 変数

11.2.1 LPOS

範囲 : 0～7、64～71

評価ボード上の LCD に文字列を表示するときの表示位置を指定します。

表示位置と設定値は以下のようになります。

1 行目	0	1	2	3	4	5	6	7
2 行目	64	65	66	67	68	69	70	71

LPOS 変数は設定のみ可能です。参照はできません。

11.2.2 LCONT

範囲 : 1～63

評価ボード上の LCD のコントラストを設定します。

1 : 薄い ←→ 63 : 濃い

デフォルト値は25です。

LCONT 変数は設定のみ可能です。参照はできません。

12. BME280 制御

以下のコマンドを使用して評価ボード上の BME280 から気温・湿度・気圧を取得することができます。

I2C で BME280 が接続されている必要があります。

12.1 BME280 コマンド

12.1.1 BME

書式:BME [引数1[,引数2[,引数3]]]

評価ボード上の BME280 で「気温、湿度、気圧」を計測します。

引数が省略された場合は、「気温、湿度、気圧」の順に表示します。

引数を指定したときは、計測値を各変数に格納します。

引数1には気温、引数2には湿度、引数3には気圧が入ります。

表示または変数に格納される各計測値は以下のようになっています。

気温 : 0.1(℃)単位

湿度 : 0.1(%)単位

気圧 : 0.1(hPa)単位

例:

```
>BME
226 391 10223
OK
>BME A,B,C
OK
>PRINT A,B,C
226     391     10223
OK
```

13. LoRa 制御

13.1 LoRa 制御について

LRA1-BASIC では、LoRa 制御用のコマンド、変数、関数を使用して LoRa パケットの送受信をすることができます。

13.2 LoRa 変数

下記の各 LoRa 変数は参照および設定が可能です。

通常の変数と同様に扱えます。(ただし、参照のみで設定不可の変数もあります)

ただし、各 LoRa 変数には最大値と最小値がありその範囲を超えて設定することはできません。各変数の意味については別途資料を参照してください。

変数名	内容	範囲	デフォルト
MODEM	モデム設定	0 : FSK , 1 : LoRa	1
PWR	送信出力	−4～13 (dBm)	13
SF	拡散率(Spread Factor)	7～12	10
BW	帯域幅(Bandwidth)	6 : 62.5kHz , 7 : 125kHz , 8 : 250kHz , 9 : 500kHz	7
CR	符号化率(Cording Rate)	1: 4/5, 2: 4/6, 3: 4/7, 4: 4/8	1
CH	チャンネル	24～61	36
FRQ	送受信周波数	920600～928000(kHz) [参照のみ]	923000
GID	グループ ID	0～65535	0
OWN	自局 ID	0～65535	1
DST	宛先 ID	0～65535	0
RSSI	RSSI	−137～0 [参照のみ]	0
STAT	LoRa 状態	詳細は STAT 変数の項目を参照してください	0
CTRL	コントロール	各ビットの設定により動作を制御します。 詳細は後記	$0000

いくつかの変数は SSAVE コマンドで Flash メモリーに保存し、SLOAD コマンドで復元することができます。

また、システム起動時には自動的に Flash メモリーから復元されて起動します。

Default コマンドでデフォルト値に再設定できます。

13.2.1 MODEM

変調方式を指定します。

範囲:0:FSK / 1:LoRa

FSK 変調では、Sf, Cr, Bw の指定は無効です。

13.2.2 PWR

送信出力を設定します。

範囲:-4～13

設定した出力で送信します。単位は(dBm)です。

電源電圧が標準電圧(3.3V)より低い場合は、実際の送信出力が低くなる場合があります。

13.2.3 SF

拡散率(Spread Factor)を設定します。

範囲:7～12

LoRa 方式での SF7～SF12 に該当します。

BW=9(500kHz)のときは SF10 以上のみ設定可能です。

13.2.4 BW

帯域幅(Bandwidth)を設定します。

範囲:6～9

6 : 62.5kHz , 7 : 125kHz , 8 : 250kHz , 9 : 500kHz

FSK 方式では BW の設定にかかわらず 500kHz となります。

BW=9(500kHz)は SF10 以上のときのみ設定可能です。

13.2.5 CR

符号化率(Cording Rate)を設定します。

範囲:1～4

1 : 4/5 , 2=4/6 , 3: 4/7 , 4: 4/8

FSK 方式では CR の設定は無効です。

13.2.6 CH

送受信チャンネルを設定します。

範囲 : 24～61

ここで設定されたチャンネルで送受信を行います。

CH 変数を設定すると FRQ 変数の値にも反映されます。

13.2.7 FRQ

送受信周波数を参照します。

範囲:920600～928000

CH 変数で設定したチャンネルに対応する周波数を参照します。単位は(kHz)です。

この変数は参照のみ可能です。設定はできません。

13.2.8 GID

グループ ID を設定します。

範囲:0～65535

同じグループ ID(GID)に設定されたモジュール間での送受信が可能です。

別のグループ ID から送信されたフレームは受信しません。

SEND コマンドおよび TXD 変数に文字列を設定したときは、この変数の値が送信バッファの Gid に自動設定されます。

13.2.9 OWN

自局 ID を設定します。

範囲:0～65534

受信側で設定した自局 ID(OWN)と、送信側が送信したフレームの宛先 ID(DST)が一致したフレームを受信することができます。ただし、宛先 ID が 65535 で送信されたフレームは自局 ID にかかわらず受信します。

送信フレームにはここで設定した自局 ID(Own)も含むので、受信側では送信元を確認することができます。

SEND コマンドおよび TXD 変数に文字列を設定したときは、この変数の値が送信バッファの Own に自動設定されます。

13.2.10 DST

宛先 ID を設定します。

範囲:0～65535

フレーム送信時の宛先 ID(DST)を設定し、受信側で設定した自局 ID(OWN)と一致した場合に受信されます。

宛先 ID が 65535 はブロードキャストとなり、送信されたフレームは受信側の自局 ID にかかわらず受信されます。

SEND コマンドおよび TXD 変数に文字列を設定したときは、この変数の値が送信バッファの DST に自動設定されます。

13.2.11 RSSI

受信パケットの RSSI 値の参照

範囲:-137～0

最後に受信した受信フレームの RSSI を参照します。単位は(dBm)です。

この変数は参照のみ可能です。設定はできません。

13.2.12 STAT

送受信状態および LoRa エラーの参照

範囲 : 0～

この変数で送受信の結果を確認することができます。

基本的には参照のみとなりますが、0のみ設定可能です。

Stat の値	意味
0	IDLE（なし）
4	キャリアセンスエラー
5	送信データー長エラー
8	受信タイムアウト
9	受信 CRC エラー
10	パケット受信

RECV コマンドでタイムアウトを指定した場合などは、コマンドから戻った原因がフレーム受信なのかタイムアウトなのかをこの Stat 変数で確認する必要があります。

13.2.13 CTRL

各種コントロールを設定します。

範囲 ： 0～

本システムの動作や送受信コマンドの振る舞いを指定することができます。

下記の表にある各ビットの論理和で設定します。

値	内容
$0001	RECV コマンドで、RSSI 値の表示をしない
$0002	RECV コマンドで、送信元 ID の表示をしない
$0004	RECV コマンドで、行頭に @ の表示をしない
$0008	RECV コマンドで、CRC エラーを表示する
$0020	LED の制御をしない
$0100	COMM コマンドで受信をしない。送信動作のみ

16 進数で記載していますが、一般の変数と同様に 10 進数での設定も可能です。

13.3 LoRa 送受信バッファ

LoRa の送受信では送受信バッファを使用します。

送受信バッファは以下のようになっています。

Offset	名前	内容
0	Gid	グループ ID（Little endian）
1		
2	Own	Txd： 自局 ID
3		Rxd： 送信元 ID
4	Dst	Txd：宛先 ID
5		Rxd：自局 ID（ブロードキャストの場合は 65535）
6	Rssi	Rxd：RSSI（Txd では未使用）

7	Len	データー長(0～240)
8 ～ 247	Data	送信データー Len で指定したデーター長のデーターを格納する

送信バッファと受信バッファはお互いに独立しています。

送信バッファは TXD/TXDW 変数、受信バッファは RXD/RXDW 変数を使用して操作します。

TXD/TXDW 変数は参照と設定が、RXD/RXDW 変数は参照が可能です。

受信バッファは最後に受信したパケットで上書きされます。

送信バッファは送信が完了してもクリアされることはありません。

13.3.1 TXD, TXDW

送信バッファの設定と参照

書式:Txd(式) または Txd

式には送信バッファの Offset を指定します。

送信バッファの Offset 位置から Byte 単位で参照または設定をします。

値を設定するときに、値が 1byte を超えるときは、下位 8bit が有効です。

例:

```
Txd(7)=10   : Length に 10 を設定します。
A=Txd(10)   : 送信 Data の 2byte 目を参照します。
```

書式:Txdw(式)

式で指定された送信バッファの Offset 位置から、Word(16bit)単位で参照と設定をします。

エンディアンは、Little endian です。Offset が奇数でも構いません。

例:

```
Txdw(4)=$1234   : 宛先 ID(Dst)に$1234 を設定します。
```

式が指定されない場合は、送信バッファの Data を文字列として参照、設定します。

文字列を設定すると、Data に文字列が格納されるとともに Gid, Own, Dst, Len も自動的に設定されます。

例:

```
Txd="1234" : Txd の Data に"1234"の 4byte を設定します。
```

PRINT、LPRINT コマンド中に記述した場合は、送信バッファの Data が文字列として表示されます。

文字列長は受信バッファ中の Len に従います。

例:

```
Print Txd : 送信バッファの Data を文字列表示する
```

送信バッファは SEND コマンドによって送信します。

SENDコマンドの引数に文字列を指定した場合は、Txdに文字列を指定した場合と同様に、送信バッファにその内容が設定されます。

13.3.2 RXD, RXDW

受信バッファの参照

書式:Rxd(式) または Rxd

式には受信バッファの Offset を指定します。

送信バッファの Offset 位置から Byte 単位で参照をします。(Txd と異なり、設定はできません)

各 Offset の場所は 1byte です。

例:

```
A=Rxd(10)  :受信 Data の 2byte 目を参照します。
```

書式:Rxdw(式)

式で指定された送信バッファの Offset 位置から Word(16bit)単位で参照します。

エンディアンは、Little endian です。Offset が奇数でも構いません。

例:

```
A=Rxdw(2)  : A に送信元 ID(Own)を取得します。
```

式が指定されない場合は、受信バッファの Data を文字列として参照します。

PRINT、LPRINT コマンド中に記述できます。

文字列長は受信バッファ中の Len に従います。

例:

```
Print Rxd : 受信バッファの Data を文字列表示
```

13.4 LoRa コマンド

13.4.1 #?

書式:#?

各種設定値を一括表示します。

13.4.2 DEFAULT

書式:DEFAULT

各種設定をデフォルト値に戻します。

このコマンドでは設定値を FLASH に保存されません。

13.4.3 SSAVE

書式:SSAVE

設定値を Flash メモリーに保存します。

ここで保存した設定はシステム起動時に自動的に読み込まれます。

また SLOAD コマンドで読み込むことができます。

13.4.4 SLOAD

書式:SLOAD

設定値を Flash メモリーから読み出します。

SSAVE で Flash メモリーに保存した設定内容が読み出されます。

13.4.5 SEND

書式:SEND [文字列]

LoRa パケットを送信します。

文字列が指定された場合は、文字列を送信バッファの Data に格納し、そのパケットを送信します。

このとき、バッファの Data に文字列が格納されるとともに Gid, Own, Dst, Len も自動的に設定されます。

文字列には BASE64 またはパーセントエンコーディングの指定が可能です。

(詳細は文字列のエンコーディングを参照してください)

文字列が指定されない場合は、送信バッファの内容がパケットとして送信されます。

RECV コマンドで受信動作中のときは、本コマンドの実行によって受信が終了します。

送信エラーについて

SF,CR,BW,CH の組み合わせによっては、指定された送信データー長(Len)のデータ(Data)を送信する時間が電波法の規定時間を超える可能性があり、その場合はパケットが送信されずに invalid_data_length エラーとなります。また、送信しようとした CH が他の送信機(LoRa 方式以外や他社製も含む)によって使用中のために CH に空きがないときは、パケットは送信されずに no_free_ch エラーとなります。

13.4.6 RECV

書式:RECV [引数]

LoRa パケットの受信を開始します。

以下の受信条件全てに合致したパケットを受信すると、そのパケットを受信バッファに格納します。

受信パケットの SF,CR,BW が SF 変数、CR 変数、BW 変数と同じ設定 受信パケットの Gid＝GID 変数 受信パケットの Dst=OWN 変数、または受信パケットの Dst=65535

引数の値によって以下の動作となります。

引数	動 作
＞0	引数にはタイムアウトの時間(msec)が指定されます。 パケット受信またはタイムアウト時間経過でコマンド戻ります。受信終了の STOP コマンドは不要です。終了原因を Stat 変数で確認する必要があります。
0	受信動作を開始してすぐにコマンドから戻ります。 STOP コマンドで受信を終了するまでは受信動作を継続します。 受信状態は Stat 変数で確認してください。Stat 変数が 0 のときのみパケットを受信します。受信パケットの処理(Timeout の処理を含む)が完了したら Stat 変数をクリアして、次のパケットが受信できるようにする必要があります。
＜0 なし	「Break 信号」または「Break キャラクタ」が入力されるまでは受信動作を継続し、受信したパケットを表示します。 表示内容は「@, RSSI, 送信元 ID, 受信 Data」となります。 Ctrl 変数の設定内容によって、表示内容を変更できます。
&	受信パケットを BASE64 でエンコードして表示します。 それ以外は「引数なし」と同じ動作です。 （詳細は文字列のエンコーディングを参照してください）
%	受信パケットをパーセントエンコードして表示します。 それ以外は「引数なし」と同じ動作です。 （詳細は文字列のエンコーディングを参照してください）

13.4.7 RXSTOP

書式：RXSTOP

受信動作を終了します。

RECV コマンドで引数が0で受信を開始したとき、受信を終了するために使用します。

13.4.8 COMM

書式：COMM [引数]

簡易的な双方向通信モードを開始します。

受信パケットがあるとパケットを表示し、入力文字列が改行されると、その文字列のパケットを送信します。

「Break 信号」または「Break キャラクタ」が入力されるまでは動作を継続します。

PA06 が High→Low になったら Sleep 動作に入り、Low→High で Sleep 解除して動作を継続します。

引数	動 作
なし	受信パケットのデーターをそのまま表示します。 入力文字列をそのまま送信します。(改行コードは含みません)
&	受信パケットを BASE64 でエンコードして表示します。 入力文字列は BASE64 でデコードして送信されます。
%	受信パケットをパーセントエンコードして表示します。 入力文字列はパーセントデコードして送信されます。

現状では本コマンドは暫定的な動作です。

13.5 LoRa エラー

LoRa コマンドでエラーが発生すると、以下のエラーメッセージが表示されます。

LoRa エラーが発生しても BASIC プログラムの動作は中断しません。

STAT 変数にはエラー番号が格納されます。

エラーメッセージ	内 容	STAT 変数
no_free_ch	キャリアセンスエラー	4
invalid_data_length	送信データー長エラー	5
timeout	受信タイムアウト	8
CRC_Error	受信 CRC エラー	9

通常の BASIC コマンドと異なり、BASIC プログラムの動作は中断されません。

13.6 文字列のエンコーディング

文字列は BASE64 または%でエンコード/デコードすることができます。

PRINT コマンドの前に「＆」をつけると、BASE64でエンコードした文字列を表示します。

PRINT コマンドの前に「%」をつけると、パーセントでエンコードした文字列を表示します。

```
%Print <------- Percent encode
&Print <------- Base64 encode
```

↑改行コードもエンコードされることに注意してください。(改行表示されません。)

改行コードを含めない場合は文字列の最後をセミコロンで終えてください。

例 : `&Print "12345";`

文字列を以下のように指定すると、エンコードされた文字列として扱います。

```
&"xxxxxxxxxxxx" <---- Base64 decode
%"xxxxxxxxxxxx" <---- Percent decode
```

例

```
>TXD=&"MTIzNDU="
OK
>PRINT TXD
12345
OK
>&PRINT TXD
MTIzNDU=OK  <---- 改行されないので、コマンド終了の"OK"が続いて表示されます。
>SEND
OK
```

LoRa コマンドにおけるエンコーディング

RECV コマンド の引数を「%」または「＆」にすると、受信データーをエンコードして表示します。

Recv & <----- Base64 encode

Recv % <----- Percent encode

COMM コマンドの引数を「%」または「＆」にすると、入力された文字列はデコードされて送信します。また受信データーはエンコードして表示します。

Comm & <----- Base64 encode/decode

Comm % <----- Percent encode/decode