

コマンドラインでのキー操作					
実装	改訂	名前	機能		
0.29		Scroll Lock	カナ入力モードになります。		カナロックキーとして動作します。(押すごとにカナ入力モードと英数入力モードが変わります。)
0.44		カタカナ ひらがな   ローマ字	カナ入力モードになります。		カナロックキーとして動作します。(押すごとにカナ入力モードと英数入力モードが変わります。)
0.17	0.46	↑	一つ前のコマンドまたはプログラム行を呼び出します。		ヒストリー編集モードで、記憶してある一つ前のコマンドを表示します。行編集モードでは一つ前のプログラム行を表示します。初期状態ではヒストリー編集モードで、¥コマンドまたはedlinコマンドで行編集モードになります。
0.17	0.46	↓	次のコマンドまたはプログラム行を呼び出します。		ヒストリー表示モードで、記憶してある次のコマンドを表示します。行編集モードでは次のプログラム行を表示します。初期状態ではヒストリー編集モードで、¥コマンドまたはedlinコマンドで行編集モードになります。
0.17		←	カーソルを左に移動します。		
0.17		→	カーソルを右に移動します。		
0.17		Delete	カーソル上の文字を消去します。		
0.17		ESC	編集中の行をキャンセルします。		
0.01	0.46	Enter	コマンドを実行します。		コマンドを実行します。入力したコマンド10個までヒストリーとして記憶します。1行が40文字を超える行は記憶しません。直前のコマンドと一致する場合も記憶しません。
0.01	0.17	BackSpace	カーソルの前の文字を消去します。		
コマンド一覧					
実装	改訂	名前	機能	書式	説明
0.01		行番号	プログラムを削除します。	<b>行番号</b>	指定した行番号のプログラムを1行削除します。
0.23	0.48	¥	指定したプログラム行を呼び出します。	<b>行番号 ¥ または ¥ 行番号</b>	指定した行をコマンドラインに表示します。行番号を省略すると現在行または先頭行を表示します。
0.17	0.48	edlin	指定したプログラム行を呼び出します。	<b>edlin 行番号</b>	指定した行をコマンドラインに表示します。行番号を省略すると現在行または先頭行を表示します。
0.08		kbset	キーボードの種類を切り替えます。	<b>kbset キーボード種別</b>	0:JP、1:US キーボード種類を省略したときは現在のキーボード種類を表示します。
0.11		autoexec	起動時の自動実行を制御します。	<b>autoexec 自動実行制御</b>	0:自動実行しない、1:ページ番号0のプログラムを自動実行

0.12	0.71	spitft	表示装置を切り替えます。	spitft <b>表示装置</b>	0:ビデオ出力、1:TFT液晶、2:TFT液晶(90度回転)、3:TFT液晶(180度回転)、4:TFT液晶(270度回転) 1,3のときは、40文字30行のテキスト出力、2,4のときは30文字40行のテキスト出力になります。
0.01		delete	プログラムを削除します。	delete <b>開始行番号 - 終了行番号</b>	
0.06		renum	行番号を付け替えます。	renum <b>開始行番号, 増分値</b>	開始行番号省略時は100、増分値は10。参照先の行番号は更新しません。
0.86	0.87	mode	端末のモードを変更します。	mode <b>端末モード</b>	端末モードは0でBASIC(ラインエディター)、1でBASIC(スクリーンエディター)、2でシリアルコンソールとして起動します。端末モードを変更すると再起動がかかります。デフォルトは0です。シリアルコンソール起動時にCTRL+SHIFT+R押下で端末モード0として再起動します。

ステートメント一覧

実装	改訂	名前	機能	書式	説明
0.01	0.23	goto	指定した行番号に制御を移します。	goto <b>行番号</b>	指定した行番号に制御を移します。行番号には式が使えます。
0.01	0.39	gosub	指定した行番号に制御を移します。	gosub <b>行番号, パラメーター列</b>	現在実行中の次の行を記憶し、指定した行番号に制御を移します。行番号には式が使えます。パラメーター列は省略できますが、指定した場合はパラメーターを順にデータスタックに積みみます。パラメーター列は数値式をコンマで区切って指定します。
0.01	0.39	return	gosubで記憶した行番号の次に制御を移します。	return <b>戻り値</b>	gosubで記憶した行番号に制御を移します。戻り値は省略可能ですが、指定した場合は戻り値をデータスタックに積みみます。
0.39		push	データスタックに値を積みみます。	push <b>数値式</b>	データスタックに数値式で指定した値を積みみます。
0.22		while	条件を満たしている間はwhile~wendをループします。	while <b>式 ~ wend</b>	式で指定した条件を満たしている間はwhile~wendの間の文を繰り返します。
0.01	0.19	for	指定回数分ループします。	for <b>初期化式 to 終了値 step 増分値 ~ next</b>	初期化式で指定した変数の値が終了値になるまでfor~nextの間の文を繰り返します。
0.19	0.22	break	ループから脱出します。	break for または break while	ループから抜けます。ループがネストしている場合は最も内側のループから抜けます。
0.23		continue	ループの先頭に戻ります。	continue for または continue while	ループの先頭に戻ります。
0.01		end	プログラムを終了します。	end	
0.01	0.42	list	プログラムを表示します。	list <b>開始行番号 - 終了行番号</b>	
0.01	0.35	new	プログラムをすべて消去します。	new	プログラム、変数、配列をクリアします。汎用メモリはクリアしません。

0.01	0.35	run	プログラムを実行します。	run 行番号	変数、配列をクリアしてプログラムを指定した行番号から実行します。汎用メモリはクリアしません。行番号が省略された場合は最も小さい番号から実行されます。
0.01		tron	トレースモードにします。	tron	
0.01		troff	トレースモードを解除します。	troff	
0.01	0.34	save	プログラムをEEPROMに書き出します。	save ページ番号, ファイル名	ページ番号は0~7までです。ファイル名は省略可能です。
0.01	0.35	load	EEPROMからプログラムを読み込みます。	load ページ番号 load ファイル名	ページ番号指定 (0~7) またはファイル名指定でプログラムをEEPROMを読み込みます。
0.35	0.43	chain	EEPROMからプログラムを読み込んで実行します。	load ページ番号, 行番号 load ファイル名, 行番号	ページ番号指定 (0~7) またはファイル名指定でプログラムをEEPROMを読み込み、指定した行番号からプログラムを実行します。変数、配列、汎用メモリをクリアしません。行番号を省略すると、先頭行から実行されます。
0.15	0.34	files	EEPROMのプログラム一覧を表示します。	files	ページ番号0~7のファイル一覧を表示します。属性として[P] (プログラム)、[D] (データ)、[X] (不正な形式) のいずれかを表示します。
0.01	0.06	rem	コメントを表します。	rem 任意の文字	0.06から省略形 ' が使用可能。
0.01		if	条件分岐します。	if 式 then 文 else 文	
0.01	0.03	locate	カーソルを移動する。	locate X座標, Y座標	
0.08	0.41	cursor	カーソル表示/消去を切り替えます。	cursor カーソル表示指示	0:消去、1:表示
0.01	0.32	print	テキスト画面に文字列を表示します。	print 式	0.06から省略形 ? が使用可能。
0.03	0.26	scroll	矩形内のテキストを1文字分スクロールします。	scroll スクロール方向, X座標, Y座標, 幅, 高さ	スクロール方向は0:左、1:右、2:上、3:下 矩形指定用のX座標以下は省略可能で、省略時はスクロール対象が画面全体になります。
0.01	0.35	cls	画面を消去します。	cls クリア対象	消去対象は1がテキスト画面、2がグラフィック画面、3が両方。省略時は3とみなします。
0.01	0.32	gprint	グラフィック画面に文字列を表示します。	gprint X座標, Y座標, 文字列, 色コード	
0.01	0.20	pset	グラフィック画面に点を打ちます。	pset X座標, Y座標, 色コード	
0.01	0.12	line	グラフィック画面に線を描きます。	line 始点X座標, 始点Y座標, 終点X座標, 終点Y座標, 色コード	
0.01	0.12	circle	グラフィック画面に円を描きます。	circle 中心X座標, 中心Y座標, 半径, 色コード	
0.05	0.12	ellipse	グラフィック画面に楕円を描きます。	ellipse 中心X座標, 中心Y座標, 半径, 係数 a, 係数 b, 色コード	
0.12	0.36	gput	グラフィック画面に正方形パターンを描きます。	gput X座標, Y座標, 幅, アドレス, 色コード	指定座標が左上の頂点となる指定幅(8の倍数) の正方形パターンを描画します。パターンは汎用メモリの指定アドレスから取得します。

0.63		gget	グラフィック画面上の正方形パターンを取得します。	gget <b>X座標, Y座標, 幅, アドレス</b>	指定座標が左上の頂点となる指定幅(8の倍数) の正方形パターンを、汎用メモリの指定アドレスから始まる位置に取得します。TFT液晶では使用できません。
0.59		gmode	グラフィック画面での描画モードを変更します。	gmode <b>描画モード</b>	描画モードを変更します。(0;OR, 1:XOR) 変更された描画モードは以降のグラフィック画面への描画命令で有効になります。描画モードの変更はTFT液晶では使用できません。
0.36		video	ビデオ出力を制御します。	video <b>ビデオ出力指示</b>	0:停止、1:開始
0.35		clv	変数をクリアします。	clv	変数、配列をクリアします。汎用メモリはクリアしません。
0.01		clt	タイマーをクリアします。	clt	
0.25		cltms	ミリ秒タイマーをクリアします。	cltms	
0.01	0.96	beep	ブザーを鳴らします。	beep <b>音階, 時間</b>	音階は21でA音440Hzで、1増やすと半音上がります。時間の単位はmsです。
0.01		pause	指定時間分待ちます。	pause <b>時間</b>	時間の単位はmsです。
0.01	0.06	input	キー入力した数値を変数に代入します。	input <b>文字列, 変数</b>	変数に文字列変数も使用可能。
0.14	0.95	ioctl	I/Oポートの入出力モードを設定します。	ioctl <b>ポート番号, 入出力モード</b>	ポート番号はピンマップ表を参照。入出力モードは0:通常出力、1:オープンドレイン出力、2:デジタル入力、100:アナログ入力(ポート1またはポート5のみ設定可能)、200:PWM制御
0.13	0.95	out	I/Oポートに出力します。	out <b>ポート番号, 出力指示</b>	ポート番号はピンマップ表を参照。出力指示は1でHigh、0でLow。
0.25	0.95	pwm	PWMを制御行います。	pwm <b>ポート番号, 周波数, デューティ比</b>	ポート番号が2~9, 100,101が有効です。使用前にioctlで設定する必要があります。周波数の単位はHzで20~2000000に対応。デューティ比の単位は%です。周波数に0を指定するとPWM出力を停止します。
0.33		mpoke	汎用メモリに値を書き込みます。	mpoke <b>アドレス, 値</b>	汎用メモリの指定したアドレスに値を書き込みます。アドレスの範囲は0~8007です。値の範囲は0~255です。
0.34	0.41	mwrite	汎用メモリの内容をすべてEEPROMに書き込みます。	mwrite <b>ページ番号, ファイル名</b>	汎用メモリの内容をEEPROMの指定したページに書き込みます。ファイル名は省略可能です。
0.34	0.35	mread	EEPROMから汎用メモリに読み込みます。	mread <b>ページ番号 mread ファイル名</b>	ページ番号指定 (0~7) またはファイル名指定でデータを汎用メモリに読み込みます。
0.35		mset	汎用メモリにデータを書き込みます。	mset <b>アドレス, 値, サイズ</b>	汎用メモリに指定したアドレス (0~8007) からサイズ(1~8008) 分の指定した値 (0~255) を書き込みます。

0.35		mmove	汎用メモリー内のデータ移動を行います。	mmove <b>転送先アドレス, 転送元アドレス, サイズ</b>	汎用メモリー内のデータを移動します。転送先アドレスと転送元アドレスが重なっても正しくコピーします。
0.39		mdata	汎用メモリーにデータ列を設定します。	mdata <b>データ列</b>	汎用メモリーへのポインターが指すアドレスからデータ列で指定する値を書き込みます。汎用メモリーへのポインターはmptr文で設定します。データ列はデータ（1バイト）をコマで区切って指定します。データ列のサイズ分、汎用メモリーへのポインターが増加します。
0.39		mptr	汎用メモリーへポインターを設定します。	mptr <b>アドレス</b>	汎用メモリーへのポインターに指定したアドレスを設定します。初期値は0です。汎用メモリーへのポインターはdata文が参照、更新します。
0.35		vmove	汎用メモリーとテキストVRAM間のデータ移動を行います。	vmove <b>アドレス, 方向</b>	汎用メモリの指定アドレスにテキストVRAMの内容をコピーします（方向指定が0のとき）、また、汎用メモリの指定アドレスの内容をテキストVRAMにコピーします。テキストVRAMサイズは1000バイトありますので、アドレスの指定に注意してください。
0.33	0.65	cpeek	キャラクターパターンを汎用メモリーに読み出します。	cpeek <b>文字コード</b> cpeek -1, <b>パターン指定</b>	文字コードに対応するキャラクターパターンを汎用メモリーのアドレス8000~8007に読み込みます。文字コードはシフトJISコードで指定します。 文字コードに-1、パターン指定に数値を指定すると、システムキャラクターを汎用メモリーに読み込みます。パターン指定が0~127の場合は8×8ピクセルのパターンを汎用メモリーの8000~8007に読み込みます。パターン指定が1000~1133の場合は16×16ピクセルのパターンを汎用メモリーの8000~8031に読み込みます。
0.10	0.33	cpoke	キャラクターコードにキャラクターパターンを設定します。	cpoke <b>キャラクターコード, パターン指定</b>	指定したキャラクターコード（0~255）のパターンを設定します。パターンは、数値（0~128）で指定します。数値に対応するパターンはシステムキャラクター表を参照してください。ただし、数値に-1を指定した場合は、汎用メモリーのアドレス8000~8007にあるパターンが設定されます。パターン指定を省略すると、デフォルトのパターンにリセットされます。
0.84	0.88	i2c	I2Cトレース情報を表示します。	i2c <b>チャンネル, トレーススイッチ</b>	指定したI2Cチャンネルの簡易トレースを制御します。トレーススイッチが0でOFF、1でONです。表示形式は以下の通りです。 [S]スタートコンディション [P]ストップコンディション [R]リピーテッドコンディション [A]ACK [N]NACK [xx]データ（16進表示）  パラメーターをすべて省略した場合は、I2Cの有効アドレスリストを I2Cn[xx]形式で表示します。nはチャンネル番号、xxは7ビットの16進数で表現したアドレスです。

0.89		i2ci	I2Cバスを使用できるようにします。	i2ci <b>チャンネル, クロックレート</b>	<p>指定したI2Cチャンネルを使用できるようにします。ORANG pico type A等のPS/2キーボードが使用できる機種ではチャンネル1は使用できません。</p> <p>クロックレートの単位はkHzです。</p> <p>SDA、SCLをそれぞれCPUの6番ピン、7番ピンに接続するとI2Cのチャンネル2が使用できます。チャンネル2はプルアップ済みでEEPROMが接続されています。(異なるアドレスのデバイスを接続できます。)</p> <p>SDA、SCLをそれぞれCPUの18番ピン、17番ピンに接続するとI2Cのチャンネル1が使用できます。18番ピン、17番ピンはオープンですので必要に応じてプルアップ抵抗を接続してください。</p>
0.55	0.77	i2c1	I2C1バスを使用できるようにします。	i2c1 <b>クロックレート</b>	<p>ORANGE pico type S等のキーボードが使用できる機種ではI2C1は使用できません。</p> <p>I2C1バスを使用する前にi2c1文を一度実行する必要があります。クロックレートの単位はkHzです。クロックレートは省略でき、デフォルト値は400です。SDA、SCLをそれぞれCPUの18番ピン、17番ピンに接続するとI2Cが使用できます。18番ピン、17番ピンはオープンですので必要に応じてプルアップ抵抗を接続してください。</p>
0.55		i2c1start	I2C1バスにスタートコンディションを出します。	i2c1start	
0.55		i2c1stop	I2C1バスにストップコンディションを出します。	i2c1stop	
0.55		i2c1restart	I2C1バスにリピーテッドスタートコンディションを出します。	i2c1restart	
0.55	0.83	i2c1ack	I2C1バスにACK/NACKを出します。	i2c1ack <b>モード</b>	モードが1でACK、0でNACKを送出します。モードを省略したときはACKを出します。
0.55	0.77	i2c2	I2C2バスを使用できるようにします。	i2c2 <b>クロックレート</b>	<p>I2C2バスには、すでにEEPROMが接続されていますので、I2c2文は内部的に実行済みです。I2C2文を実行しなくてもかまいません。クロックレートの単位はkHzです。クロックレートは省略でき、デフォルト値は400です。SDA、SCLをそれぞれCPUの6番ピン、7番ピンに接続するとI2Cが使用できます。6番ピン、7番ピンはプルアップしてあります。</p>
0.54		i2c2start	I2C2バスにスタートコンディションを出します。	i2c2start	
0.54		i2c2stop	I2C2バスにストップコンディションを出します。	i2c2stop	
0.54		i2c2restart	I2C2バスにリピーテッドスタートコンディションを出します。	i2c2restart	
0.54	0.83	i2c2ack	I2C2バスにACK/NACKを出します。	i2c2ack <b>モード</b>	モードが1でACK、0でNACKを送出します。モードを省略したときはACKを出します。

0.56	0.66	turtle	タートルを初期化します。	turtle <b>タートル番号, タートル位置X座標, タートル位置Y座標, タートル表示有無, 遅延時間</b>	タートルの状態を初期化します。パラメーターをすべて省略した場合は、タートル番号1のタートルが有効になり、画面左上の位置(0,0)で進行方向0度、タートル表示あり、遅延時間0ms、ペンを下げた状態になります。タートル番号は1~8が使用できます。タートル表示有無は1でタートルを表示します。遅延時間の単位はms(ミリ秒)で、move、turnの後に指定時間だけ動作を遅延させます。タートル表示はTFT液晶では使用できません。
0.56		turn	タートルの進行方向を変えます。	turn <b>角度</b>	タートルの進行方向が角度(degree)で指定した分、右方向に増えます。角度にマイナスの値を指定すれば左方向に向きます。
0.56		move	タートルを移動します。	move <b>移動距離</b>	タートルを指定した移動距離(ピクセル)分、前進(x,y座標とも増加)します。移動距離にマイナスの値を指定すれば後退します。
0.56		pen	タートルグラフィックスのペンを制御します。	pen <b>描画指示</b>	描画指示を0にすると、ペンを上げます。(以降のmoveで移動のみになります。)描画指示を1にすると、ペンを下げます。(以降のmoveで描画します。)
0.57		sel	対象タートルを指定します。	sel <b>タートル番号</b>	move、turnの対象となるタートルを指定します。move、turnの対象は直前に指定したタートルになります。
0.64	0.65	sprite	スプライトを移動します。	sprite <b>スプライト番号, X座標, Y座標, パターン番号, モード</b>	スプライト番号は1~16で重なって表示したときに、大きい番号が上に表示されます。X座標、Y座標で表示位置を指定します。パターン番号は1000~1133を指定します。パターン番号に負の値を設定すると、汎用メモリーのアドレスから始まるパターンを読み込みます。(-8000を指定すると、8000番地から始まるユーザー設定パターンになります。) モードは0で指定したパターンを指定した座標に移動します。モードに-1を指定するとスプライトを無効化します。モードに1を指定すると、パターンを水平反転、2を指定すると垂直反転します。
0.72		uart	UARTを設定します。	uart <b>チャンネル, モード, ボーレート</b>	指定したチャンネルのモード、ボーレートを設定します。現バージョンでは、チャンネルは1以外は無効です。モードのbit0(LSB)が1のときはコンソール出力されたものを同時にUARTに送信します。モードのbit1が1のときは、UARTからの入力もキー入力のみとみなします。モードは起動時に3となっています。ボーレートの単位はbpsで、300~115200の値で指定します。
0.72		uartput	UARTに送信します。	uartput <b>チャンネル, 文字列</b>	指定したチャンネルに文字列を送信します。現バージョンでは、チャンネルは1以外は無効です。

0.80	0.85	lpt	プリンターの設定をします。	lpt <b>プリンター種別, ボーレート</b>	<p>プリンタ種別は以下の通りです。</p> <p>0: プリンター無効</p> <p>1: ナダ電子製プリンターAS-289R2(ORANGE picoのUART1に接続)</p> <p>2: ナダ電子製プリンターAS-289R2(ORANGE picoのUART2に接続)</p> <p>101: 汎用プリンター(ORANGE picoのUART1に接続)</p> <p>102: 汎用プリンター(ORANGE picoのUART2に接続)</p> <p>ORANGE picoのUARTのTXD (UART1はCPUの11番ピン、UART2は21番ピン) とプリンター側のRXDを接続します。また、それぞれのGND同志を接続します。AS-289R2ジャンパー設定 はJP5の2と3をショートします。</p> <p>ボーレートは省略できます。省略した場合は、9600bpsになります。ナダ電子製プリンターAS-289R2では38400を指定することができます。(プリンター側のJP6をショートすることが前提)</p>
0.80		lprint	プリンターに印字します。	lprint <b>式</b>	print文と同じ書式でプリンターに印字します。プリンターが無効のときは、コンソールに出力します。
0.80		llist	プリンターにプログラムリストを印字します。	list <b>開始行番号 - 終了行番号</b>	list文と同じ書式でプリンターに印字します。プリンターが無効のときは、コンソールに出力します。
0.81			画面のハードコピーをとります。	copy <b>コピー種別</b>	コピー種別は1でテキスト画面、2でグラフィック画面のハードコピーを取ります。TFT液晶や汎用プリンターでは1のみ有効です。
0.96		play	MML文字列を演奏します。	play <b>文字列式</b>	<p>MMLで記述した文字列を演奏します。文字列の終端に達するまでplay文は終了しないで演奏を続けます。</p> <p>文字列式を省略した場合は、汎用メモリーの先頭にあるデータから演奏します。この場合、play文は即終了し、バックグラウンドで演奏します。汎用メモリー内のMML文字列が終端に達したときに演奏が終了します。また、プログラムが終了したときは、途中でも演奏を終了します。</p> <p>※MMLの書式は別紙参照</p>
0.96		mstr	汎用メモリーに文字列を設定します。	mstr <b>文字列式</b>	汎用メモリーへのポインターが指すアドレスから文字列を書き込みます。汎用メモリーへのポインターはmptr文で設定します。文字列のサイズ分、汎用メモリーへのポインターが増加します。



0.97		xfiles	USBメモリー上のファイル一覧を表示します。	xfiles <b>ファイルマスク</b>	指定したファイルマスク(文字列)にマッチするファイルの一覧を表示します。ファイルマスクには、*と?が使用できます。ファイルマスクは省略すると、すべてのファイルが対象になります。
0.97		xload	USBメモリーからテキスト形式で保存されたプログラムを読み込みます。	xload <b>ファイル名</b>	指定したファイル名のプログラムをUSBメモリーから読み込みます。
0.97		xsave	プログラムをテキスト形式で保存します。	xsave <b>ファイル名</b>	指定したファイル名でプログラムをUSBメモリーに保存します。

#### 関数一覧

実装	改訂	名前	機能	書式	説明
0.96		sound	演奏中かどうかを取得します。	sound()	演奏中のときは1、演奏中でないときは0を返します。
0.90	0.93	button	ゲームコントローラーのボタン押下情報を取得します。	button( <b>機能ID</b> )	機能IDは-1でゲームコントローラーを初期化します。実行時に一度呼び出す必要があります。機能IDは0でボタン情報を取得します。ボタンの押下状態はビットで返します。押されている場合は該当ビットが1になります。該当ビットはLSBから十字ボタン上、十字ボタン下、十字ボタン左、十字ボタン右、Aボタン、Bボタン、Xボタン(未使用)、Yボタン(未使用)、Lボタン(未使用)、Rボタン(未使用)、STARTボタン、SELECTボタンです。
0.73		uartget	UARTから受信します。	uartget( <b>チャンネル</b> )	指定したチャンネルの受信バッファの先頭のデータを取得します。受信バッファが空のときは-1を返します。現バージョンでは、チャンネルは1以外は無効です。
0.63		point	指定した点の色コードを読み取ります。	point( <b>X座標, Y座標</b> )	黒のときは0、白のときは1が返ります。TFT液晶では使用できません。
0.89		i2cw	I2Cバスに書き込みます。	i2cw( <b>チャンネル, スレーブアドレス, 書き込みデータ, ストップ指示</b> )	指定したI2Cチャンネルにデータを書き込みます。スレーブアドレスは7ビットで指定します。(R/Wフラグは0が付加されます。)書き込みデータは文字列で指定します。0を送りたいときは%x00を文字列内に記述します。ストップ指示は1でスタートコンディションを送出し、0でリピーテッドスタートコンディションを送出します。  成功すると0が返ります。
0.89		i2cr	I2Cバスから読み込みます。	i2cr( <b>チャンネル, スレーブアドレス, 読み込みバイト数, ストップ指示</b> )	指定したI2Cチャンネルからデータを読み込みます。スレーブアドレスは7ビットで指定します。(R/Wフラグは1が付加されます。)ストップ指示は1でスタートコンディションを送出し、0でリピーテッドスタートコンディションを送出します。  読み込んだデータは汎用メモリーの0番地から順次格納されます。汎用メモリーはtmpeek関数で参照できます。  成功すると0が返ります。

0.75		i2c1put	I2C1バスに文字列を出力します。	i2c1put(文字列式)	指定した文字列を出力し、ACK応答があれば0を返します。
0.54		i2c1write	I2C1バスに1バイトのデータを出力します。	i2c1write(数値式)	指定したデータを出力し、ACK応答があれば0を返します。
0.54		i2c1read	I2C1バスから1バイトのデータを入力します。	i2c1read()	読み取ったデータを返します。
0.75		i2c2put	I2C2バスに文字列データを出力します。	i2c2put(文字列式)	指定した文字列を出力し、ACK応答があれば0を返します。
0.54		i2c2write	I2C2バスに1バイトのデータを出力します。	i2c2write(数値式)	指定したデータを出力し、ACK応答があれば0を返します。
0.54		i2c2read	I2C2バスから1バイトのデータを入力します。	i2c2read()	読み取ったデータを返します。
0.39		pop	データスタックから取り出した値を返します。	pop()	データスタックから値を取り出し、取り出した値を返します。
0.14	0.95	in	IOポートの状態を返します。	in(ポート番号)	ポート番号はピンマップ表を参照。指定したポートがHighのときに1、Lowのときに0を返します。使用前に一度ioctl文で入力モードにしておく必要があります。
0.21		adc	アナログポートの値を返します。	adc(ポート番号)	指定したポートの電圧0~3.3Vに応じて0~1023の値を返します。使用前に一度ioctl文でアナログモードにしておく必要があります。
0.32		rgb	16ビットカラーの値を返します。	rgb(RED値, GREEN値, BLUE値)	RED値(0~255)、GREEN値(0~255)、BLUE値(0~255)から16ビットの色を作ります。TFT液晶の色指定に使用します。
0.33		mpeek	指定した汎用メモリのアドレスの値を返します。	mpeek(アドレス)	アドレスの範囲は0~8007です。
0.78	0.79	format\$	指定した出力書式に変換した文字列を返します。	format\$(出力書式, ...)	出力書式はC言語のprintfの書式に準じます。 %[フラグ][フィールド幅][精度][長さ修飾子]変換指定子 の形で使用できます。使用できる変換指定子はc、d、i、u、o、x、X、e、fです。出力書式に続くパラメーターの数は可変長で、数と型が出力書式内で指定した変換指定子と合っている必要があります。
0.01		left\$	文字列の左側から指定された数の文字列を取り出します。	left\$(文字列式, 数値式)	
0.01		right\$	文字列の右側から指定された数の文字列を取り出します。	right\$(文字列式, 数値式)	
0.01		mid\$	文字列の指定場所から指定された数の文字列を取り出します。	mid\$(文字列式, 開始位置, 文字数)	
0.01		chr\$	アスキーコードに対応する文字を返します。	chr\$(数値式)	
0.01	0.50	str\$	数値を10進数文字列にします。	str\$(数値式)	Ver 0.50~より実数にも対応します。

0.03	0.32	asc	文字式のアスキーコードを返します。	asc(文字式)	
0.03		hex\$	数式を16進数の文字列に変換します。	hex\$(数値式)	
0.05		instr	文字列を検索します。	instr(文字列式, 文字列式)	
0.01		len	文字列の長さを返します。	len(文字列式)	
0.07	0.50	val	文字列を数値に変換します。	val(文字列式)	Ver 0.50~より実数にも対応します。
0.03	0.09	vpeek	テキスト画面上の指定値の文字コードを返します。	vpeek(X座標, Y座標)	
0.01		tick	タイマーの値を返します。	tick()	単位は1/60秒です。タイマーをクリアするにはcltを使用します。
0.25		tickms	ミリ秒タイマーの値を返します。	tickms()	単位はms(ミリ秒)です。タイマーをクリアするにはcltmsを使用します。
0.01		free	プログラムの残り領域サイズを返します。	free()	
0.01		ver	バージョン番号を返します。	ver()	
0.26	0.42	kbstatus	キーの押下状態を返します。	kbstatus()	キーの押下状態をビットで返します。押されている場合は該当ビットが1になります。該当ビットはLSBから←、→、↑、↓、スペースキー、左SHIFT、右SHIFT、左CTRL、右CTRLです。
0.02		inkey	押されたキーコードを返します。	inkey()	
0.01		rnd	ランダムな値を返します。	rnd(数値式)	
0.40		sin	数値式の正弦を返します。	sin(数値式)	
0.40		cos	数値式の余弦を返します。	cos(数値式)	
0.40		tan	数値式の正接を返します。	tan(数値式)	
0.52		asin	数値式の逆正弦を返します。	asin(数値式)	
0.52		acos	数値式の逆余弦を返します。	acos(数値式)	
0.40		atan	数値式の逆正接を返します。	atan(数値式)	
0.40		exp	自然対数eの数値式乗を返します。	exp(数値式)	
0.40		log	自然対数eを底とする対数を返します。	log(数値式)	
0.52	0.93	log10	常用対数10を底とする対数を返します。	log10(数値式)	
0.40		sqrt	数値式の平方根を返します。	sqrt(数値式)	
0.49		ceil	数値式の小数点以下を切り上げた値を返します。	ceil(数値式)	
0.49		floor	数値式の小数点以下を切り下げた値を返します。	floor(数値式)	
0.52		sinh	数値式の実双曲線正弦を返します。	sinh(数値式)	
0.52		cosh	数値式の実双曲線余弦を返します。	cosh(数値式)	
0.52		tanh	数値式の実双曲線正接を返します。	tanh(数値式)	
0.52		pow	べき乗を計算します。	pow(x, y)	xのy乗を返します。

演算子一覧 (優先度が低い順)

実装	改訂	名前	分類	書式	説明
0.01		or	二項演算子		左辺と右辺の論理和を返します。
0.29			二項演算子		左辺と右辺の論理和を返します。
0.01		and	二項演算子		左辺と右辺の論理積を返します。
0.29		&&	二項演算子		左辺と右辺の論理積を返します。
0.07			二項演算子		左辺と右辺のビットごとの論理和を返します。
0.07		^	二項演算子		左辺と右辺のビットごとの排他的論理和を返します。
0.07		&	二項演算子		左辺と右辺のビットごとの論理積を返します。
0.01		=	二項演算子		左辺と右辺を比較します。等しい場合に1を返します。
0.23		==	二項演算子		左辺と右辺を比較します。等しい場合に1を返します。
0.03		<>	二項演算子		左辺と右辺を比較します。等しくない場合に1を返します。
0.23		!=	二項演算子		左辺と右辺を比較します。等しくない場合に1を返します。
0.01		<	二項演算子		左辺と右辺を比較します。左辺と右辺が文字列の場合は文字列の大小（コード順）を比較します。
0.01		<=	二項演算子		左辺と右辺を比較します。左辺と右辺が文字列の場合は文字列の大小（コード順）を比較します。
0.01		>	二項演算子		左辺と右辺を比較します。左辺と右辺が文字列の場合は文字列の大小（コード順）を比較します。
0.01		>=	二項演算子		左辺と右辺を比較します。左辺と右辺が文字列の場合は文字列の大小（コード順）を比較します。
0.07		<<	二項演算子		論理左シフト
0.07		>>	二項演算子		論理右シフト

0.01		+	二項演算子		左辺と右辺を加算します。左辺と右辺が文字列の場合は文字列を連結します。
0.01		-	二項演算子		
0.01		*	二項演算子		
0.01		/	二項演算子		
0.01		mod	二項演算子		左辺を右辺で割った余りを返します。
0.19		%	二項演算子		左辺を右辺で割った余りを返します。
0.19		not	単項演算子		論理反転します。
0.19		!	単項演算子		論理反転します。
0.19		~	単項演算子		ビット反転

エラーメッセージ一覧

実装	改訂	メッセージ	説明
0.01		Syntax error	
0.01		Out of memory	
0.01		Out of string space	
0.01		String too long	
0.01		Type mismatch	
0.01		Illegal function call	
0.01		Undefined line number	
0.01		Illegal line number	
0.01		RETURN without GOSUB	
0.01		NEXT without FOR	
0.01		WEND without WHILE	
0.01		Subscript out of range	
0.01		Redimensioned array	
0.01		Division by zero	
0.22		BREAK without FOR	
0.22		BREAK without WHILE	
0.23		CONTINUE without FOR	
0.23		CONTINUE without WHILE	
0.24		Formula too complex	
0.29		Undefined label	
0.29		Duplicate label	

0.34			Bad File		mread、loadでファイル属性が合っていないものを読もうとした。互換性のない古いファイルを読もうとした。
0.35			File not found		mread、loadで指定ファイルが見つからない。
0.39			Stack overflow		データスタックが一杯でpush文を実行できない。
0.39			Stack underflow		データスタックにデータがないのにpop文を実行しようとした。
0.97			Drive not ready		USBメモリーが挿入されていない。
0.97			Disk I/O error		USBメモリーアクセス中に予期せぬエラーが発生した。
0.97			Timeout		サブシステムと通信中にタイムアウトした。