QuickJS\_ESP32 関数リファレンス

[関数一覧]

|  |  |
| --- | --- |
| モジュール名 | 関数名 |
| デフォルト | console.log(str) |
| console.assert(assertion, str) |
| console.info(str) |
| console.debug(str) |
| console.warn(str) |
| console.error(str) |
| console.enableLogio(enable) |
| setTimeout(func, msec) |
| clearTimeout(id) |
| setInterval(func, msec) |
| clearInterval(id) |
| millis() |
| delay(msec) |
| esp32 | deepSleep(usec) |
| reboot() |
| restart() |
| reload([url]) |
| setLoop(func) |
| update() |
| getIpAddress() |
| getMacAddress() |
| getDeviceModel() |
| gpio | pinMode(pin, mode) |
| analogRead(pin) |
| digitalRead(pin) |
| digitalWrite(pin, value) |
| http | postJson(url, [body, [headers]]) |
| postText(url, [body, [headers]]) |
| getJson(url, [qs, [headers]]) |
| getText(url, [qs, [headers]]) |
| imu | getAccelData() |
| getGyroData() |
| getTempData() |
| input | isPressed(btn) |
| isReleased(btn) |
| wasPressed(btn) |
| wasReleased(btn) |
| pressedFor(btn, ms) |
| releasedFor(btn, ms) |
| wasReleasedFor(btn, ms) |
| isTouched() |
| lcd | setRotation(rot) |
| setBrightness(brt) |
| setFont(size) |
| setTextColor(fore, [back]) |
| setTextSize(scale, yscale) |
| setTextDatum(datum) |
| drawPixel(x, y, color) |
| drawLine(x0, y0, x1, y1, [color]) |
| setCursor(x, y) |
| getCursor() |
| textWidth(str) |
| print(str) |
| println(str) |
| fillScreen(color) |
| downloadImage(url) |
| drawImage(x, y) |
| drawImageFile(fname, x, y) |
| width() |
| height() |
| getColorDepth() |
| getFontHeight() |
| mqtt | connect(client\_name, [url, [port]]) |
| disconnect() |
| subscribe(topic, func) |
| unsubscribe() |
| publish(topic, payload) |
| getPayload() |
| rtc | setTime(hours, minutes, seconds) |
| setDate(year, month, day, weekday) |
| getTime() |
| getDate() |
| utils | httpPostJson(url, [body, [headers]]) |
| httpPostText(url, [body, [headers]]) |
| httpGetJson(url, [qs, [headers]]) |
| urlencode(str) |
| base64Encode(str) |
| base64Decode(str) |
| sd | exists(fname) |
| mkdir(dname) |
| remove(fname) |
| rmdir(dname) |
| size(file) |
| readBinary(fname, [offset, [size]]) |
| writeBinary(fname, buffer, [offset, [size]]) |
| readText(fname) |
| writeText(fname, str) |
| isDirectory(fname) |
| list(dname) |
| wire | begin(sda, scl) |
| requestFrom(address, count, [stop]) |
| beginTransmission(address, count) |
| endTransmission([stop]) |
| write(value) |
| available() |
| read([count]) |
| wire1 | ※モジュールwireと同じ |

# モジュール：デフォルト

## console.log(str)

[Function]

コンソールに文字列を出力する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| str | String | 〇 | 文字列 |

[Output]

無し

## console.assert(assertion, str)

[Function]

コンソールに文字列を出力する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| assertion | Bool | 〇 | falseの場合に、文字列を出力する。 |
| str | String | 〇 | 文字列 |

[Output]

無し

## console.info(str)

[Function]

コンソールに文字列を出力する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| str | String | 〇 | 文字列 |

[Output]

無し

## console.debug(str)

[Function]

コンソールに文字列を出力する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| str | String | 〇 | 文字列 |

[Output]

無し

## console.warn(str)

[Function]

コンソールに文字列を出力する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| str | String | 〇 | 文字列 |

[Output]

無し

## console.error(str)

[Function]

コンソールに文字列を出力する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| str | String | 〇 | 文字列 |

[Output]

無し

## console.enableLogio(enable)

[Function]

Log.ioサーバに文字列を送信するかを設定する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| enable | Bool | 〇 | trueの場合、Log.ioサーバに文字列を出力する |

[Output]

無し

## setTimeout(func, msec)

[Function]

指定した待ち時間後に関数を実行する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| func | Function | 〇 | 呼び出される関数 |
| msec | Number | 〇 | 待ち時間 |

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| Number | タイマーID |

## clearTimeout(id)

[Function]

タイマーIDで指定されるsetTiemoutを解除する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| id | Number | 〇 | タイマーID |

[Output]

無し

## setInterval(func, msec)

[Function]

指定した待ち時間ごとに関数を実行する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| func | Function | 〇 | 呼び出される関数 |
| msec | Number | 〇 | 待ち時間 |

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| Number | タイマーID |

## clearInterval(id)

[Function]

タイマーIDで指定されるsetIntervalを解除する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| id | Number | 〇 | タイマーID |

[Output]

無し

## millis()

[Function]

ESP32が起動してからの経過時間を取得する。

[Input]

無し

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| Number | 経過時間(msec) |

## delay(msec)

[Function]

指定時間ディレイする。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| msec | Number | 〇 | 指定時間(msec) |

[Output]

無し

# モジュール：esp32

## deepSleep(usec)

[Function]

指定時間の間、DeepSleepモードに移行する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| usec | Number | 〇 | DeepSleepに移行している時間(usec) |

[Output]

無し

## reboot()

[Function]

ESP32を再起動する。

[Input]

無し

[Output]

無し

## restart()

[Function]

QuickJSを再起動する。

[Input]

無し

[Output]

無し

## reload([url])

[Function]

指定されるURLからモジュール指定ファイル(.json)またはJsファイル(.js)を読み込み、QuickJSを再起動する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| url | String | NULL | 取得するファイルのURL(.jsonまたは.js)  NULLの場合はソース埋め込み値 |

[Output]

無し

## setLoop(func)

[Function]

関数を繰り返し実行する。

ただし、すでにloop()処理で利用済み。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| func | Function | 〇 | 実行する関数 |

[Output]

無し

## update()

[Function]

ESPの更新処理を実行する。M5.update()に相当

[Input]

無し

[Output]

無し

## getIpAddress()

[Function]

IPアドレスを取得する。

[Input]

無し

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| Number | IPアドレス  例：192.168.1.1の場合、0xC0A80101 |

## getMacAddress()

[Function]

MACアドレスを取得する

[Input]

無し

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| [Number] | MACアドレスの配列  例：01:02:03:04:05:06の場合[1, 2, 3, 4, 5, 6] |

## getDeviceModel()

[Function]

ESPデバイスの種類を取得する。

[Input]

無し

[Output]

|  |  |  |
| --- | --- | --- |
| 引数名 | 型 | 内容 |
| model | Number | ESP32デバイスの種類  MODEL\_OTHER 0  MODEL\_M5Core2 1  MODEL\_M5Core 2  MODEL\_M5Fire 3  MODEL\_M5StickCPlus 4  MODEL\_M5CoreInk 5  MODEL\_M5Paper 6  MODEL\_M5Tough 7  MODEL\_M5StickC 8 |

# モジュール：gpio

## pinMode(pin, mode)

[Function]

PINの動作を指定する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| pin | Number | 〇 | PIN番号 |
| mode | Number | 〇 | モード  INPUT 0x01  OUTPUT 0x02  PULLUP 0x04  INPUT\_PULLUP 0x05  PULLDOWN 0x08  INPUT\_PULLDOWN 0x09  OPEN\_DRAIN 0x10  OUTPUT\_OPEN\_DRAIN 0x12 |

[Output]

無し

## analogRead(pin)

[Function]

アナログ出力値を取得する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| pin | Number | 〇 | PIN番号 |

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| Number | アナログ出力値 |

## digitalRead(pin)

[Function]

デジタル出力値を取得する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| pin | Number | 〇 | PIN番号 |

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| Number | デジタル出力値 |

## digitalWrite(pin, value)

[Function]

デジタル出力値を設定する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| pin | Number | 〇 | PIN番号 |
| value | Number | 〇 | デジタル出力値  LOW 0x00  HIGH 0x01 |

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| Number | デジタル出力値 |

# モジュール：http

## postJson(url, [body, [headers]])

[Function]

中継サーバを介して、JSONをHTTP Post呼び出しし、JSONを取得する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| url | String | 〇 | URL |
| body | Object | {} | HTTP bodyに指定する値 |
| headers | Object | {} | HTTP headerに指定する値  application/jsonは自動付与 |

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| Object | 戻り値 |

## postText(url, [body, [headers]])

[Function]

中継サーバを介して、JSONをHTTP Post呼び出しし、Textを取得する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| url | String | 〇 | URL |
| body | Object | {} | HTTP bodyに指定する値 |
| headers | Object | {} | HTTP headerに指定する値  application/jsonは自動付与 |

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| String | 戻り値 |

## getJson(url, [qs, [headers]])

[Function]

中継サーバを介して、JSONをHTTP Get呼び出しし、JSONを取得する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| url | String | 〇 | URL |
| qs | Object | {} | QueryStringに指定する値 |
| headers | Object | {} | HTTP headerに指定する値 |

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| Object | 戻り値 |

## getText(url, [body, [headers]])

[Function]

中継サーバを介して、JSONをHTTP Get呼び出しし、Textを取得する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| url | String | 〇 | URL |
| qs | Object | {} | QueryStringに指定する値 |
| headers | Object | {} | HTTP headerに指定する値  application/jsonは自動付与 |

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| String | 戻り値 |

# モジュール：imu

## getAccelData()

[Function]

加速度を取得する。

[Input]

無し

[Output]

|  |  |  |
| --- | --- | --- |
| 型 | 項目 | 内容 |
| Object | x | X方向の加速度 |
| y | Y方向の加速度 |
| z | Z方向の加速度 |

## getGyroData()

[Function]

ジャイロを取得する。

[Input]

無し

[Output]

|  |  |  |
| --- | --- | --- |
| 型 | 項目 | 内容 |
| Object | x | X方向のジャイロ |
| y | Y方向のジャイロ |
| z | Z方向のジャイロ |

## getTempData()

[Function]

内部温度を取得する。

[Input]

無し

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| Float | 内部温度 |

# モジュール：input

## isPressed(btn)

[Function]

ボタンの押下状態を取得する

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| btn | Number | 〇 | ボタンの指定  BUTTON\_A 1  BUTTON\_B 2  BUTTON\_C 3 |

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| Bool | ボタンの押下状態 |

## isReleased(btn)

[Function]

ボタンのリリース状態を取得する

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| btn | Number | 〇 | ボタンの指定  BUTTON\_A 1  BUTTON\_B 2  BUTTON\_C 3 |

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| Bool | ボタンのリリース状態 |

## wasPressed(btn)

[Function]

ボタンが押下されたかどうかを取得する

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| btn | Number | 〇 | ボタンの指定  BUTTON\_A 1  BUTTON\_B 2  BUTTON\_C 3 |

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| Bool | ボタンの押下状態 |

## wasReleased(btn)

[Function]

ボタンがリリースされかどうかを取得する

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| btn | Number | 〇 | ボタンの指定  BUTTON\_A 1  BUTTON\_B 2  BUTTON\_C 3 |

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| Bool | ボタンのリリース状態 |

## pressedFor(btn, ms)

[Function]

ボタンが指定期間以上押下されているかどうかを取得する

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| btn | Number | 〇 | ボタンの指定  BUTTON\_A 1  BUTTON\_B 2  BUTTON\_C 3 |
| ms | Number | 〇 | 指定期間(msec) |

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| Bool | ボタンの押下状態 |

## releasedFor(btn, ms)

[Function]

ボタンが指定期間リリースされているかどうかを取得する

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| btn | Number | 〇 | ボタンの指定  BUTTON\_A 1  BUTTON\_B 2  BUTTON\_C 3 |
| ms | Number | 〇 | 指定期間(msec) |

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| Bool | ボタンのリリース状態 |

## wasReleasedFor(btn, ms)

[Function]

ボタンが指定期間リリースされかどうかを取得する

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| btn | Number | 〇 | ボタンの指定  BUTTON\_A 1  BUTTON\_B 2  BUTTON\_C 3 |
| ms | Number | 〇 | 指定期間(msec) |

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| Bool | ボタンのリリース状態 |

## isTouched()

[Supported]

* M5Core2

[Function]

ボタンが指定期間リリースされかどうかを取得する

[Input]

無し

[Output]

|  |  |  |
| --- | --- | --- |
| 型 | 項目 | 内容 |
| Object | x | X座標 |
| y | Y座標 |

※タッチされなかった場合は、null

# モジュール：lcd

## setRotation(rot)

[Function]

LCDの表示の向きを指定する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| rot | Number | 〇 | 表示の向き(0～3) |

[Output]

無し

## setBrightness(brt)

[Function]

LCDの輝度を指定する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| rot | Number | 〇 | 輝度(0～255) |

[Output]

無し

## setFont(size)

[Function]

フォントを指定する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| size | Number | 〇 | フォントの種類  指定可能な数値は、コンパイル時に決定 |

[Output]

無し

## setTextColor(fore, [back])

[Function]

フォント色を指定する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| fore | Number | 〇 | フォントの色(RRGGBBh等) |
| back | Number | 透過色 | フォントの背景色 |

[Output]

無し

## setTextSize(scale, [yscale])

[Function]

フォント色を指定する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| scale | Float | 〇 | フォントの拡大率 |
| yscale | Float | scaleと同値 | フォントの縦方向の拡大率 |

[Output]

無し

## setTextDatum(datum)

[Function]

フォントの表示位置を設定する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| datum | Number | 〇 | フォントの表示位置  top\_left 0  top\_center 1  top\_right 2  middle\_left 4  middle\_center 5  middle\_right 6  bottom\_left 8  bottom\_center 9  bottom\_right 10  baseline\_left 16  baseline\_center 17  baseline\_right 18 |

[Output]

無し

## drawPixel(x, y, color)

[Function]

点を描画する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| x | Number | 〇 | X座標 |
| y | Number | 〇 | Y座標 |
| color | Number | 〇 | 色(RRGGBBhなど) |

[Output]

無し

## drawLine(x0, y0, y1, y2, [color])

[Function]

線を描画する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| x0 | Number | 〇 | 開始X座標 |
| y0 | Number | 〇 | 開始Y座標 |
| x1 | Number | 〇 | 終了X座標 |
| y1 | Number | 〇 | 終了Y座標 |
| color | Number | 文字色 | 色(RRGGBBhなど) |

[Output]

無し

## setCursor(x, y)

[Function]

カーソル位置を設定する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| x | Number | 〇 | X座標 |
| y | Number | 〇 | Y座標 |

[Output]

無し

## getCursor()

[Function]

カーソル位置を取得する。

[Input]

無し

[Output]

|  |  |  |
| --- | --- | --- |
| 型 | 項目 | 内容 |
| Object | x | X座標 |
| y | Y座標 |

## textWidth(str)

[Function]

文字列の表示幅を取得する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| str | String | 〇 | 文字列 |

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| Number | 表示幅 |

## print(str)

[Function]

LCDに文字列を表示する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| str | String | 〇 | 文字列 |

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| Number | 表示文字数 |

## println(str)

[Function]

LCDに文字列を改行付きで表示する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| str | String | 〇 | 文字列 |

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| Number | 表示文字数 |

## fillScreen(color)

[Function]

LCDを色で塗りつぶす。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| color | Number | 〇 | 塗りつぶす色(RRGGBBh等) |

[Output]

無し

## downloadImage(url)

[Function]

画像ファイルをダウンロードする。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| url | String | 〇 | URL |

[Output]

無し

## drawImage(x, y)

[Function]

ダウンロードした画像ファイルを表示する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| x | Number | 〇 | 表示するX座標 |
| y | Number | 〇 | 表示するY座標 |

[Output]

無し

## drawImageFile(fname, x, y)

[Function]

SDにある画像ファイルを表示する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| fname | String | 〇 | SDにあるファイル名 |
| x | Number | 〇 | 表示するX座標 |
| y | Number | 〇 | 表示するY座標 |

[Output]

無し

## width()

[Function]

LCDの幅を取得する。

[Input]

無し

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| Number | LCDの幅 |

## height()

[Function]

LCDの高さを取得する。

[Input]

無し

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| Number | LCDの高さ |

## getColorDepth()

[Function]

LCDの表示色の深度を取得する。

[Input]

無し

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| Number | LCDの深度 |

## getFontHeight()

[Function]

フォントの高さを取得する。

[Input]

無し

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| Number | フォントの高さ |

# モジュール：mqtt

## connect(client\_name, [url, [port]])

[Function]

MQTTブローカに接続する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| client\_name | String | 〇 | MQTTクライアント名 |
| url | String | ソース埋め込み値 | MQTTブローカのURL |
| port | Number | ソース埋め込み値 | MQTTブローカのポート番号 |

[Output]

無し

## disconnect()

[Function]

MQTTブローカと切断する。

[Input]

無し

[Output]

無し

## subscribe(topic, func)

[Function]

MQTTを購読する。

購読するトピックは1つのみであり、すでに購読中のトピックは解除される。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| topic | String | 〇 | トピック名 |
| func | Function | 〇 | 受信時にコールされる関数 |

[Output]

無し

## unsubscribe()

[Function]

MQTTを購読する。

[Input]

無し

[Output]

無し

## publish(topic, payload)

[Function]

MQTTにメッセージを配信する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| topic | String | 〇 | トピック名 |
| payload | String | 〇 | 送信する内容 |

[Output]

無し

## getPayload()

[Function]

受信したMQTTメッセージを取得する。

[Input]

無し

[Output]

|  |  |  |
| --- | --- | --- |
| 型 | 項目 | 内容 |
| Object | topic | 受信したトピック名 |
| payload | 受信内容 |

# モジュール：rtc

## setTime(hours, minutes, seconds)

[Function]

RTCに時刻を設定する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| hours | Number | 〇 | 時 |
| minutes | Number | 〇 | 分 |
| seconds | Number | 〇 | 秒 |

[Output]

無し

## setDate(year, month, day, weekday)

[Function]

RTCに日付を設定する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| year | Number | 〇 | 時 |
| month | Number | 〇 | 分 |
| day | Number | 〇 | 秒 |
| weekday | Number | 〇 | 曜日 |

[Output]

無し

## getTime()

[Function]

RTCから時刻を取得する。

初期値は、ESP32起動時にNTPから取得された値。

[Input]

無し

[Output]

|  |  |  |
| --- | --- | --- |
| 型 | 項目 | 内容 |
| Object | Hours | 時 |
| Minutes | 分 |
| Seconds | 秒 |

## getDate()

[Function]

RTCから日付を取得する。

初期値は、ESP32起動時にNTPから取得された値。

[Input]

無し

[Output]

|  |  |  |
| --- | --- | --- |
| 型 | 項目 | 内容 |
| Object | Year | 年 |
| Month | 月 |
| Date | 日 |
| WeekDay | 曜日 |

# モジュール：utils

## httpPostJson(url, [body, [headers]])

[Function]

JSONをHTTP Post呼び出しし、JSONを取得する。

HTTPSは未対応

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| url | String | 〇 | URL |
| body | Object | {} | HTTP bodyに指定する値 |
| headers | Object | {} | HTTP headerに指定する値  application/jsonは自動付与 |

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| Object | 戻り値 |

## httpPostText(url, [body, [headers]])

[Function]

JSONをHTTP Post呼び出しし、Textを取得する。

HTTPSは未対応

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| url | String | 〇 | URL |
| body | Object | {} | HTTP bodyに指定する値 |
| headers | Object | {} | HTTP headerに指定する値  application/jsonは自動付与 |

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| String | 戻り値 |

## httpGetText(url, [body, [headers]])

[Function]

JSONをHTTP Get呼び出しし、Textを取得する。

HTTPSは未対応

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| url | String | 〇 | URL |
| qs | Object | {} | QueryStringに指定する値 |
| headers | Object | {} | HTTP headerに指定する値  application/jsonは自動付与 |

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| String | 戻り値 |

## urlencode(str)

[Function]

文字列をURLエンコードする。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| str | String | 〇 | 文字列 |

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| String | URLエンコードされた文字列 |

## base64Encode(str)

[Function]

文字列をBase64エンコードする。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| str | String | 〇 | 文字列 |

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| String | Base64エンコードされた文字列 |

## base64Decode(str)

[Function]

文字列をBase64デコードする。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| str | String | 〇 | 文字列 |

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| String | Base64デコードされた文字列 |

# モジュール：sd

## exists(fname)

[Function]

SDにファイルの存在を確認する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| fname | String | 〇 | ファイル名 |

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| Bool | trueの場合、ファイルが存在する |

## mkdir(dname)

[Function]

SDにディレクトリを作成する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| dname | String | 〇 | ディレクトリ名 |

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| Bool | trueの場合、成功 |

## remove(fname)

[Function]

SDからファイルを削除する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| fname | String | 〇 | ファイル名 |

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| Bool | trueの場合、成功 |

## rmdir(dname)

[Function]

SDからディレクトリを削除する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| dname | String | 〇 | ディレクトリ名 |

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| Bool | trueの場合、成功 |

## size(fname)

[Function]

SDのファイルのサイズを取得する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| fname | String | 〇 | ファイル名 |

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| Number | ファイルサイズ |

## readBinary(fname, [offset, [size]])

[Function]

SDのファイルをバイナリとして読み出す。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| fname | String | 〇 | ファイル名 |
| offset | Number | 0 | 読み出す先頭位置 |
| size | Number | -1 | 読み出すサイズ。-1の場合は、ファイルの最後まで |

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| ArrayBuffer | バイナリ |

## writeBinary(fname, buffer, [offset, [size]])

[Function]

SDのファイルをバイナリとして書き出す。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| fname | String | 〇 | ファイル名 |
| buffer | ArrayBuffer | 〇 | バイナリ |
| offset | Number | -1 | 書き出す先頭位置。-1の場合は、ファイルの末尾 |
| size | Number | -1 | 書き出すサイズ。-1の場合は、バイナリのサイズ |

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| Number | 書き出したサイズ |

## isDirectory(fname)

[Function]

SDのディレクトリかを確認する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| fname | String | 〇 | ファイル名 |

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| Bool | trueの場合、ディレクトリ |

## list(dname)

[Function]

SDのディレクトリにあるファイル一覧を取得する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| dname | String | 〇 | ディレクトリ名 |

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| [String] | ファイル名の配列 |

# モジュール：wire

## begin(sda, scl)

[Function]

I2Cバスを初期化する。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| sda | Number | 〇 | SDAのポート番号 |
| scl | Number | 〇 | SCLのポート番号 |

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| Bool | 成功 or 失敗 |

## requestFrom(address, count, [stop])

[Function]

I2Cデバイスからの受信を要求します。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| address | Number | 〇 | I2Cデバイスのアドレス(7ビット) |
| count | Number | 〇 | 要求バイト数 |
| stop | Bool | true | STOPメッセージを送った後にI2Cを開放するかどうか |

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| Number | 受信したバイト数 |

## beginTransmission(address, count)

[Function]

I2Cバスの送信処理を開始します。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| address | Number | 〇 | I2Cデバイスのアドレス(7ビット) |

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| Number | 受信したバイト数 |

## endTransmission([stop])

[Function]

I2Cバスの送信処理を終了します。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| stop | Bool | false | trueの場合、STOPメッセージを送信してから終了 |

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| Number | 終了結果  成功 0  それ以外 0以外 |

## write(value)

[Function]

I2Cデバイスに送信します。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| value | Number or  [Number] | 〇 | Numberの場合、1バイト送信  [Number]の場合、複数バイト送信 |

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| Number | 送信バイト数 |

## available()

[Function]

I2Cデバイスから受信可能バイト数を取得します。

[Input]

無し

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| Number | 受信可能なバイト数 |

## read([count])

[Function]

I2Cデバイスから受信します。

[Input]

|  |  |  |  |
| --- | --- | --- | --- |
| 引数名 | 型 | 必須/デフォルト値 | 内容 |
| count | Number | 1 | 受信するバイト数 |

[Output]

|  |  |
| --- | --- |
| 型 | 内容 |
| Number or  [Number] | count指定がない場合、受信データ1バイト  count指定がある場合、受信データの配列 |

# モジュール：wire1

※モジュールwireと同じ