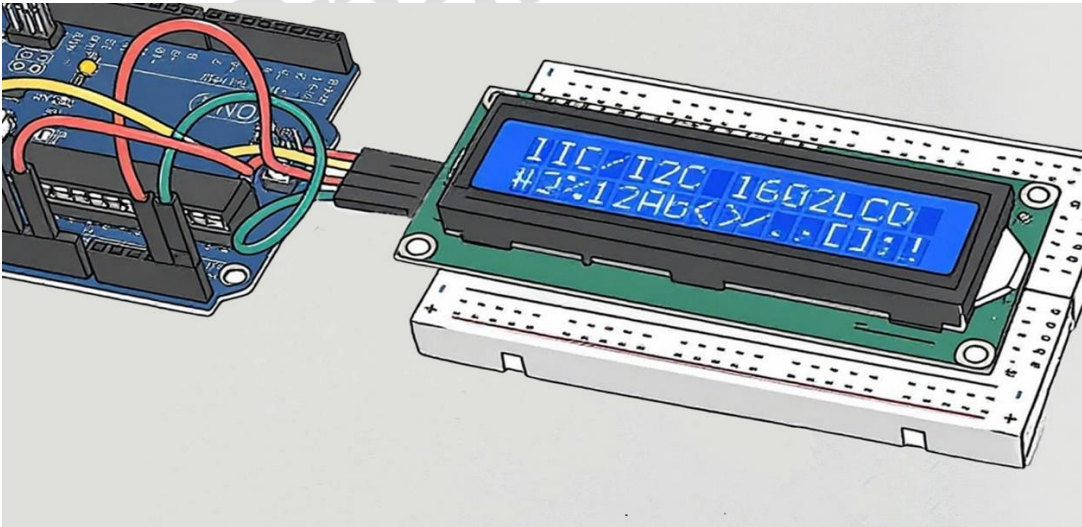


# 1602 LCD モジュール ユーザーマニュアル



Arduino UNO:

配線:

VSS -> GND  
VDD -> 5V  
SDA -> A4  
SCK -> A5

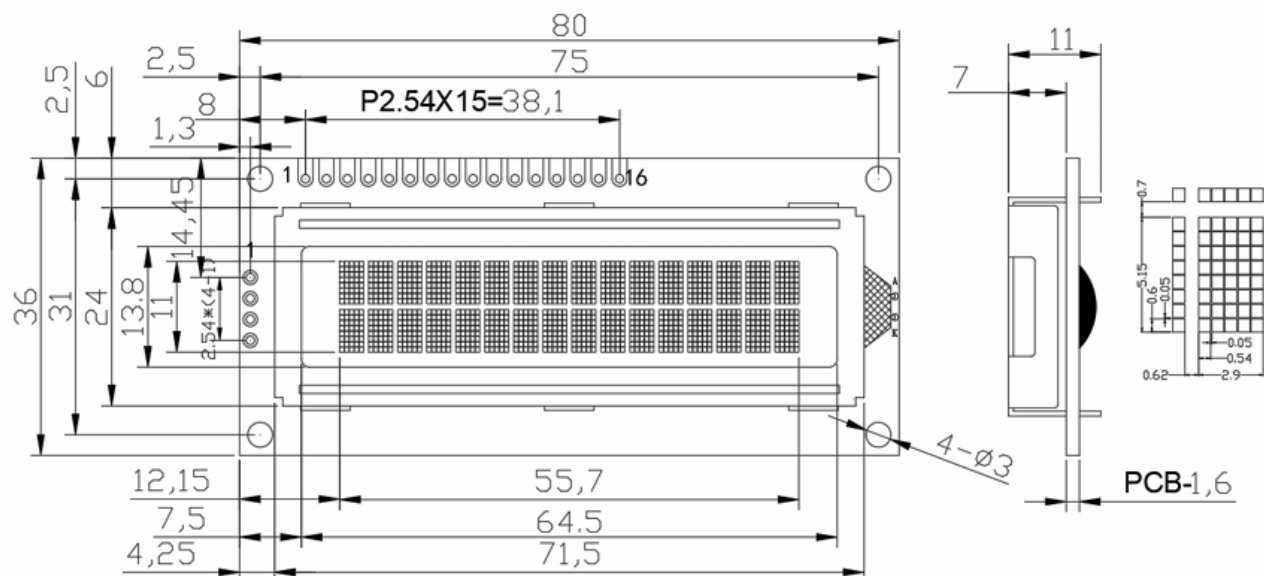
Example:

```
01 #include <Wire.h>
02 #include <LiquidCrystal_I2C.h>
03
04 // LCD オブジェクトを作成 (I2C アドレス 0x27、16 文字×2 行)
05 LiquidCrystal_I2C lcd(0x27, 16, 2);
06
07 void setup() {
08     lcd.init();           // LCD を初期化
09     lcd.backlight();      // バックライトを ON にする
10     lcd.setCursor(0, 0); // カーソルを 1 行目の先頭 (0 列目) に設定
11     lcd.print("Hello, World!"); // 文字列「Hello, World!」を表示
12 }
13
14 void loop() {
15     // ループ内では特に処理しない (表示はそのまま維持される)
16 }
```

# 技術資料

1. I<sup>2</sup>C アドレス: **0x27**

2. 外形寸法:



シリアル I<sup>2</sup>C インターフェース:

ピン番号	記号	機能	
1	VSS	電源供給	0V
2	VDD		+5V
3	SDA	シリアルデータ	
4	SCK	シリアルクロック	

## 3. 接続および通信仕様

### ■ アドレス設定

I<sup>2</sup>C 通信を利用する本モジュールは、I/O エクスパンダ IC「PCF8574」を搭載しています。スレーブアドレス (Slave Address) は、IC の A0~A2 ピンの設定により決定されます。

### ビット構成 内容

7bit アドレス 0100 A2 A1 A0

書き込み時 最下位ビット = 0

読み込み時 最下位ビット = 1

たとえば、A2=A1=A0=0 の場合、スレーブアドレスは **0x27** または **0x3F** となります (モジュールによって異なります)。

### ■ データビット接続 (D[0:7])

ビット 信号名	説明
D[0] RS	レジスタ選択 (0 = 命令、1 = データ)
D[1] RW	読み書き制御 (0 = 書き込み、1 = 読み取り)
D[2] E	イネーブル信号 (立ち上がりでデータ確定)
D[3] BL	バックライト制御 (1 = 点灯、0 = 消灯)
D[4:7] DB4 ~ DB7	LCD データバス (4bit モード)

LCD は 4 ビット通信モードで動作します。そのため DB4 ~ DB7 のみを使用します。

## ■ 通信データフォーマット

I<sup>2</sup>C 通信では、以下の手順でデータが送信されます。

1. **スタート条件 (Start Condition)**  
SDA が High → Low に変化した時点で通信開始。
2. **スレーブアドレス送信**  
上位 7bit アドレス + R/W ビット (0: 書き込み、1: 読み込み)
3. **スレーブからの ACK (応答信号)**
4. **データ送信 (DATA1、DATA2 …)**  
各データバイト後にもスレーブから ACK 応答。
5. **ストップ条件 (Stop Condition)**  
SCL High の状態で SDA が Low → High に変化して通信終了。

## ■ データ送信例 (書き込み動作)

```

Start
↓
スレーブアドレス (0100A2A1A00)
↓
コマンドデータ (DATA1)
↓
LCD データ (DATA2)
↓
Stop

```

## ■ 機能概要

本モジュールは I<sup>2</sup>C 通信により、わずか 2 本の信号線 (SCL・SDA) で LCD1602 を制御できます。

RS / E / DB4 ~ DB7 / バックライトなどの各信号は、PCF8574 I/O 拡張 IC を介して操作されます。

## 4. 命令の説明

本モジュールは、**4ビット / 8ビット MCU 並列通信モード**を備えています。

4ビットまたは8ビットバスは、**命令レジスタのDLビット**によって選択されます。

読み書き操作では、2つの8ビットレジスタが使用されます。

1つは**データレジスタ (DR)**、もう1つは**\*\*命令レジスタ (IR) \*\***です。

データレジスタ (DR) は、**DDRAM / CGRAM データの読み書き用一時格納領域**として使用されます。

対象のRAMは、**RAM アドレス設定命令**によって選択されます。

RAMの読み書き操作はすべて内部的に自動で行われます。

MCU が DR からデータを読み出すと、**DDRAM / CGRAM のデータが自動的に DR へ転送**されます。  
同様に、MCU が DR へデータを書き込むと、**DR 内のデータが自動的に DDRAM / CGRAM へ転送**されます。  
命令レジスタ (IR) は、**MCU から送られる命令コードを格納**します。  
MCU は命令データを読み出すことはできません。  
レジスタの選択は **RS ピンの設定**によって切り替えられます。

#### RS / RW ビットの組み合わせによる操作

RS	RW	操作内容
L	L	命令書き込み操作 (MCU → IR へ命令コード書き込み)
L	H	ビジーフラグ (DB7) とアドレスカウンタ (DB0 ~ DB6) の読み出し
H	L	データ書き込み操作 (MCU → DR へデータ書き込み)
H	H	データ読み出し操作 (MCU ← DR からデータ読み出し)

#### 4.1 ビジーフラグ (BF)

**BF = “1”** の場合、内部処理中を意味します。  
この間、新しい命令は実行できません。  
RS = “0”、R/W = “1” (命令読み出し操作) の時、  
**BF の値は DB7 ピンから読み取ることができます。**  
次の命令を実行する前に、**BF ≠ “1”**であることを確認してください。

#### 4.2 アドレスカウンタ (AC)

命令レジスタから送られた **DDRAM / CGRAM アドレス**は、アドレスカウンタに格納されます。  
データを DDRAM / CGRAM に書き込みまたは読み出すたびに、  
**アドレスカウンタの値は自動的にインクリメントまたはデクリメント**されます。  
RS = “0”、R/W = “1” の時、アドレスカウンタの内容を **DB0 ~ DB6** から読み出すことができます。

#### 4.3 表示データ RAM (DDRAM)

本モジュールの DDRAM アドレス範囲は以下の通りです：  
**00H ~ 27H および 40H ~ 67H**

#### 4.4 キャラクタジェネレーション ROM (CGROM)

CGROM には以下の文字フォントデータが格納されています：

- 5×8 ドットマトリクス文字：192 文字
- 5×11 ドットマトリクス文字：64 文字

#### 4.5 キャラクタジェネレーション RAM (CGRAM)

CGRAM は **5×8 ドットマトリクス文字を 8 文字分格納**できます。  
各文字は **\*\*8 行×5 ビット (8 バイト) \*\***で構成されています。  
ユーザーは、CGRAM に独自の文字データを書き込むことで、  
**8 個のユーザー定義文字**を作成できます。  
作成した文字は、**DDRAM に 00H ~ 07H** を書き込むことで呼び出すことができます。

#### DDRAM / CGRAM アドレス対応図



Character Code (DDRAM data)								CGRAM address						CGRAM Data								Pattern number
D7	D6	D5	D4	D3	D2	D1	D0	A5	A4	A3	A2	A1	A0	P7	P6	P5	P4	P3	P2	P1	P0	
0	0	0	0	×	0	0	0	0	0	0	0	0	0	×	×	×	0	1	1	1	0	pattern1
											0	0	1				0	0				
											0	1	0				0	0				
											0	1	1				0	0				
											1	0	0				1	0				
											1	0	1				0	0				
											1	1	0				1	0				
											1	1	1				1	0				
											0	0	0				0	0				
0	0	0	0	×	0	0	1	0	0	1	0	0	0	×	×	×	0	1	1	1	0	pattern2
											0	0	1				1	0	1			
											0	1	0				1	0	0			
											0	1	1				1	0	0			
											1	0	0				1	0	0			
											1	0	0				0	0	1			
											1	1	0				1	1	0			
											1	1	1				1	1	0			
											0	0	0				0	0	0			
* * * *								* * * *						* * * *								
0	0	0	0	×	1	1	1	1	1	1	0	0	0	×	×	×	1	1	1	1	1	pattern8
											0	0	1				1	0	0			
											0	1	0				1	0	0			
											0	1	1				1	1	0			
											1	0	0				1	0	0			
											1	0	1				1	0	1			
											1	1	0				1	1	0			
											1	1	1				1	1	0			
											0	0	0				0	0	0			

以下は、あなたが提供した\*\*「7.6 指令描述」および「7.7 显示位置与 RAM 地址对应表」\*\*を日本語に翻訳した  
 正式な技術説明書文です。  
 (LCD1602 モジュールの日本語版マニュアルとしてそのまま使えます)

#### 4.6 命令の詳細説明

##### 命令一覧表

命令	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	実行時間	説明
表示クリア	0	0	0	0	0	0	0	0	0	1	1.53ms	DDRAM に 20H (空白コード) を書き込み、アドレスカウンタを 00H に設定
リターン	0	0	0	0	0	0	0	0	1	-	1.53ms	アドレスカウンタを 00H に設定し、カーソルを初期位置に戻す (DDRAM 内容は保持)

入力モード設定	0	0	0	0	0	0	0	1	I/D	SH	39μs	カーソル移動方向と表示全体の移動を設定
表示 ON/OFF	0	0	0	0	0	0	1	D	C	B	39μs	表示・カーソル・点滅の制御設定
シフト	0	0	0	0	0	1	S/C	R/L	-	-	39μs	カーソルまたは表示全体の移動方向を設定（DDRAM データは保持）
機能設定	0	0	0	0	1	DL	N	F	-	-	39μs	データバス幅、行数、フォントサイズを設定
CGRAM アドレス設定	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	39μs	CGRAM のアドレスを設定
DDRAM アドレス設定	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	39μs	DDRAM のアドレスを設定
忙信号 & アドレス読取	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0	0μs	BF（ビジーフラグ）とアドレスカウンタを読み取る
データ書き込み	1	0	D7	D6	D5	D4	D3	D2	D1	D0	43μs	DDRAM/CGRAM ヘデータを書き込む
データ読み出し	1	1	D7	D6	D5	D4	D3	D2	D1	D0	43μs	DDRAM/CGRAM からデータを読み出す

※「-」は無視されるビットを示す。

#### 4.6.1 表示クリア

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	0	1

20H（空白コード）を全 DDRAM アドレスに書き込み、アドレスカウンタを 00H に設定することで表示をクリア。  
カーソルを初期位置に戻し、入力モードを増加（I/D = 1）に設定。

#### 4.6.2 リターン

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	1	-

カーソルを初期位置に戻し、アドレスカウンタを 00H に設定。  
DDRAM の内容は保持されます。

#### 4.6.3 入力モード設定

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	1	I/D	SH

I/D: カーソル移動方向

- I/D = 1 → カーソル右移動（アドレス増加）
- I/D = 0 → カーソル左移動（アドレス減少）

## SH: 表示シフト

- SH=0 → 表示は移動しない
- SH=1 → 書き込み時に表示全体が I/D 方向にシフト

### 4.6.4 表示 ON/OFF 制御

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	1	D	C	B

- D: 表示制御**  
D=1 表示 ON / D=0 表示 OFF (データは保持)
- C: カーソル表示**  
C=1 カーソル表示 / C=0 カーソル非表示
- B: カーソル点滅**  
B=1 点滅 ON / B=0 点滅 OFF

### 4.6.5 カーソル / 表示シフト

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	1	S/C	R/L	-	-

カーソル位置または表示全体を左 / 右に移動します。

DDRAM データは保持されます。

S/C	R/L	動作内容
0	0	カーソル左移動、アドレス-1
0	1	カーソル右移動、アドレス+1
1	0	表示全体左シフト (カーソル追従)
1	1	表示全体右シフト (カーソル追従)

### 4.6.6 機能設定

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	DL	N	F	-	-

- DL: データバス幅**
  - 1=8ビットモード
  - 0=4ビットモード (2回転送)
- N: 行数設定**
  - 0=1行表示
  - 1=2行表示
- F: フォント設定**
  - 0=5×8ドット
  - 1=5×11ドット

### 4.6.7 CGRAM アドレス設定

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0

CGRAM アドレスをアドレスカウンタに設定し、  
MCU からの CGRAM データを有効にします。

### 4.6.8 DDRAM アドレス設定

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0

DDRAM アドレスを設定し、MCU からのデータを有効にします。

- 1 行表示モード：00H～4FH
- 2 行表示モード：第 1 行 00H～27H、第 2 行 40H～67H

#### 4.6.9 忙信号とアドレス読み出し

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0

BF=1 のとき、内部動作中で命令は実行できません。

BF=0 になるまで待機してください。

この命令で同時にアドレスカウンタの値も読み出せます。

#### 4.6.10 RAM へのデータ書き込み

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	0	D7	D6	D5	D4	D3	D2	D1	D0

8 ビットデータを DDRAM または CGRAM に書き込みます。

書き込み後、アドレスカウンタは I/D 設定に従って増減します。

#### 4.6.11 RAM からのデータ読み出し

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	1	D7	D6	D5	D4	D3	D2	D1	D0

DDRAM または CGRAM から 8 ビットデータを読み出します。

初回の読み出し結果は無効で、2 回目以降が有効です。

読み出し後、アドレスカウンタは I/D 設定に従い増減します。

#### 4.6.12 初期化

電源投入時、自動的に以下の初期化処理が行われます：

1. 表示クリア命令：DDRAM 全体に 20H を書き込み
2. 機能設定：DL=1 (8 ビットモード)、N=0 (1 行)、F=0 (5×8)
3. 表示制御：D=0 (表示 OFF)、C=0 (カーソル OFF)、B=0 (点滅 OFF)
4. エントリーモード設定：I/D=1 (増加)、SH=0 (シフトなし)

初期化中、ビジーフラグは“1”を維持します。

#### 4.7 表示位置と RAM アドレス対応表

表示位置	DDRAM アドレス	表示位置	DDRAM アドレス
1-1～1-10	00～09	2-1～2-10	40～49
1-11～1-20	0A～13	2-11～2-20	4A～53
1-21～1-30	14～1D	2-21～2-30	54～5D
1-31～1-40	1E～27	2-31～2-40	5E～67

※「1-1」は第 1 行の最初の文字位置を表します。

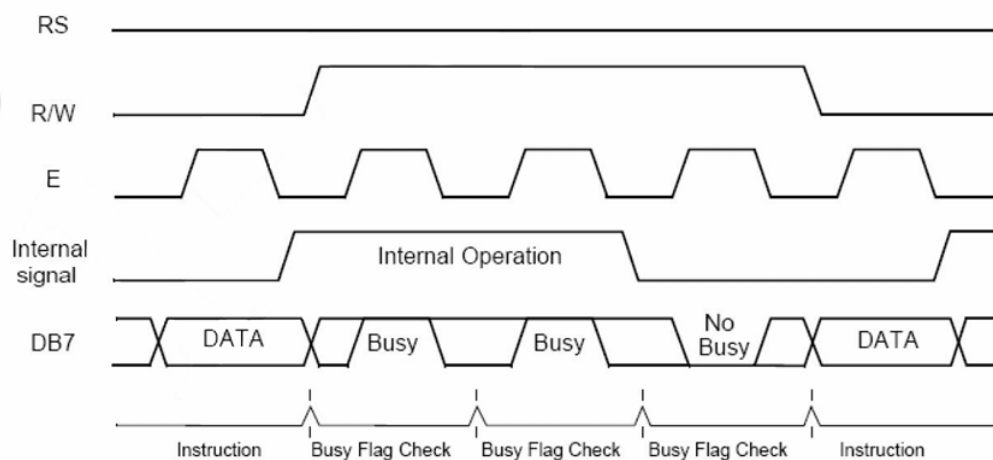
### 5. 動作タイミング (操作シーケンス)

#### 5.1 8 ビット MCU インターフェース



インターフェースのデータ幅を **8ビットモード** に設定した場合、データは\*\*8ビットポート（DB0～DB7）\*\*を通じて同時に読み書きされます。動作タイミングは下図のようになります。

（\*8ビットデータの読み書きシーケンス図を参照）



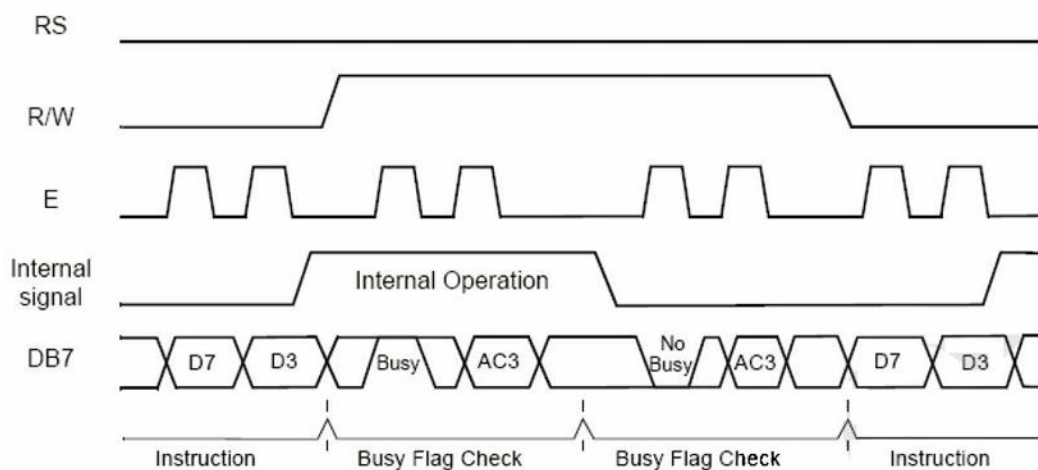
## 5.2 4ビット MCU インターフェース

インターフェースのデータ幅を **4ビットモード** に設定した場合、データは\*\*8ビットポートの上位4ビット（DB4～DB7）\*\*のみを使用して読み書きが行われます。この場合、データは次の順序で転送されます：

1. 上位4ビット（高4ビット）を先に送信
2. 続いて下位4ビット（低4ビット）を送信

動作タイミングは下図のようになります。

（\*4ビットデータの転送シーケンス図を参照）



## 6 字庫

b7-b4 b3-b0	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	CG RAM (1)			0	1	P	\	P				-	9	E	o	p
0001	(2)		!	1	A	Q	a	q			.	7	7	4	a	q
0010	(3)		"	2	B	R	b	r			"	イ	ツ	×	β	θ
0011	(4)		#	3	C	S	c	s			」	ウ	テ	E	ε	ω
0100	(5)		\$	4	D	T	d	t			、	エ	ト	ト	μ	σ
0101	(6)		%	5	E	U	e	u			.	オ	大	1	ε	0
0110	(7)		&	6	F	V	f	v			ヲ	カ	ニ	ヨ	ρ	Σ
0111	(8)		'	7	G	W	g	w			フ	キ	ヌ	ラ	g	π
1000	(1)		(	8	H	X	h	x			イ	ウ	ホ	リ	フ	又
1001	(2)		)	9	I	Y	i	y			ウ	ク	ル	ル	リ	ユ
1010	(3)		*	:	J	Z	j	z			エ	コ	ロ	レ	j	チ
1011	(4)		+	:	K	L	k	l			★	サ	ヒ	ロ	*	万
1100	(5)		,	<	L	※	1	1			ホ	シ	フ	フ	φ	円
1101	(6)		-	=	M	I	m	>			ユ	ズ	ハ	フ	ト	÷
1110	(7)		.	>	N	^	n	*			ヨ	セ	ホ	ハ	ん	
1111	(8)		/	?	O	L	o	←			ッ	ソ	マ	"	る	■

詳しくは GitHub へ: [/SparkKit\\_1602A\\_I2C\\_LCD](https://github.com/SparkKit/1602A_I2C_LCD)