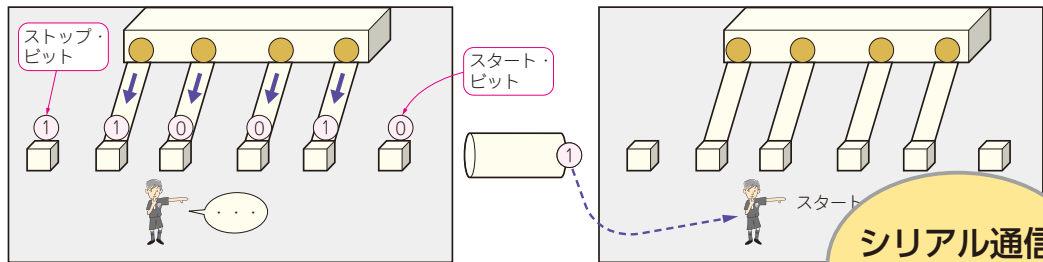


# イントロダクション

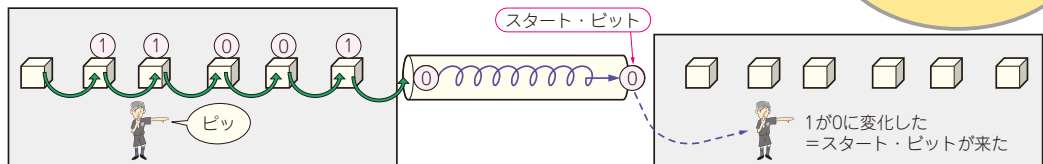
## 3ステップで作りながら学ぶ! 定番シリアル通信

編集部

### ステップ1…シリアル通信を基礎から理解



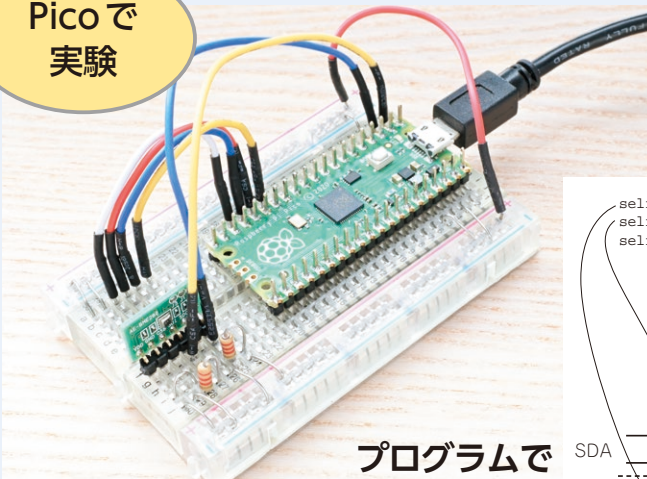
(c) ステップ2: 送信側シフト・レジスタにデータがセットされる。送信側シフト・レジスタではスタート・ビットを用意する



(d) ステップ3: 送信側で笛を吹いてスタート・ビットから送信開始。受信側では、受信した値が1から0に変化したので、スタート・ビットが届いたことが分かる

### ステップ2…ビット・バンギングでゼロから作る

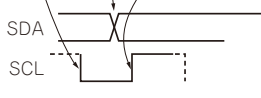
Picoで  
実験



プログラムを  
作りながら  
UART/I<sup>2</sup>C/SPI  
を学ぶ

```
self.scl.init ( Pin.OUT )  
self.sda.init ( Pin.IN )  
self.scl.init ( Pin.IN )
```

SCLを“L”にして  
SDAにvの値を出力  
SDAが確定したらSCLを“H”に  
(SCLが“H”になったことによって  
ターゲットがデータを取り込む)  
複数ビットを出力する際にはこ  
れを繰り返す



プログラムで  
波形を作る

# ステップ3…プロトコル・アナライザを作る

第1部  
良い理由  
基本技術  
定番の4つ  
図解リアル通信

第2部  
UART  
I2C  
SPI  
I3C  
ゼロから作る

第3部  
アナライザ  
プロトコル

第4部  
対策集  
トラブル

特設  
徹底解説  
CAN

# 困ったときの駆け込み寺…シリアル通信トラブル・シューティング集

## 電気回路一般編

### 1 活線挿抜時の通信異常とラッチア

- **トラブルに至る前提条件**  
シリアル通信は、異なるシステム間を接続し、相互

図7 I2Cのクロストークによる波形  
互いの信号線の電流の影響(クロストーク)によって波形が乱れる

# 信頼性、耐ノイズ性、システム拡張性に優れる! CAN徹底解説

## 基礎から解説

データ・フレーム	アービトレーション・フィールド				エラー・フレーム
CAN ID	RTR	IDE	r0	エラー・フラグ	
11ビット	1ビット	1ビット	1ビット	6ビット	
受信ノード RxID (0x1A: (00110101111))	1ビット	1ビット	1ビット	6ビット	
受信ノード RxID (0x2AF: (01010101111))	1ビット	1ビット	1ビット	6ビット	
接続ノード RxID	1ビット	1ビット	1ビット	6ビット	
レギュラリ伝送路 (FSマストID)	1ビット	1ビット	1ビット	3ビット	6ビット
				ノイズ混入	8ビット

ラズパイでソケット通信