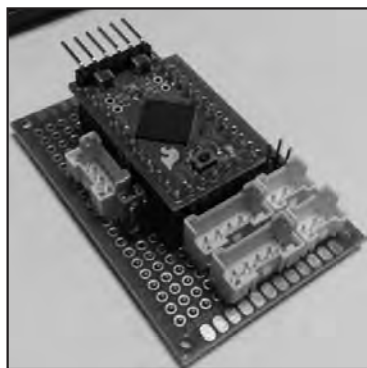


## 短期連載



専用ポート「CAN485」で複数サーボ制御&データ観測

# ArduinoでCAN通信実験

## 〈2〉中編：1対1通信

原野 知則 Tomonori Harano

### ● 本連載の目標…手っ取り早くCANを試す

CAN(Contoller Area Network)は、自動車や工場設備、搬送用ドローンなど身近で使用されているシリアル通信方式です。本連載ではCAN機能を搭載したArduino CAN485(SparkFun Electronics)マイコン3台の複数通信やCANデータ波形観測などを試します。

今回は複数通信の下準備として1対1通信を試みます。

### 目標：最小構成でCAN動作を確かめる

### ● そもそも通信とは？

「通信」をネット検索したところ『通信とは、送り手が情報を送り、また受け手が情報を受けること』と記されていました。電気がなかった頃は煙ですら情報通信の手段でした。この情報伝達を「通信」と称しますが、ここで重要なのは、送り手と受け手があって初めて通信となることです。CAN通信は、この送り手と受け手がないと動作が成立しません。

### ● 送り手に対して受け手(答えてくれる相手)が必要

CANの場合、図1の通信フォーマットの生成と通信可否の判断は通信専用ハードウェアが担います。

### ▶ 送信開始までの流れ

まず送信ユニットが通信線がレセプ(劣性…論理1)であることを確認し、通信線をドミナント(優性…

論理0)状態にすることで通信が開始されます。その後、図1のフォーマット通りの信号を送出し、最後に受信完了のACKを確認します。受信側の専用ハードウェアで正常受信した場合、ACK信号のビットはドミナント状態を返してきます。ドミナントの通知は、マイコンのプログラムに関係なくCANコントローラの設定が正常であればハードウェアが自動的に返します。

### ▶ 送信が完了しない場合

受信相手がいない場合、ACKが返ってこないのでもいつまで経っても送信完了できず、送信リトライを繰り返します。この辺が通常のUARTシリアル通信と異なります。UARTのように通信相手の状態に関係なく信号を垂れ流すことができません。送信テストやデバッグを実施するには、受信完了を答えてくれる相

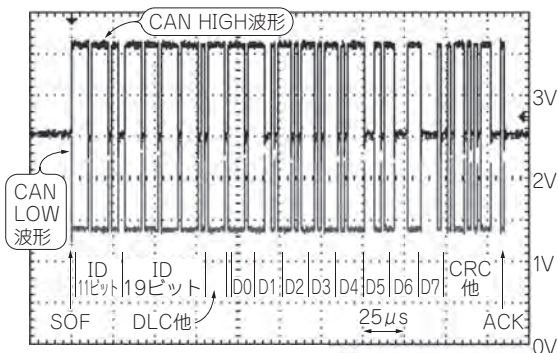


図2 実際のCAN通信波形  
本編で紹介するサンプル・プログラムで実行

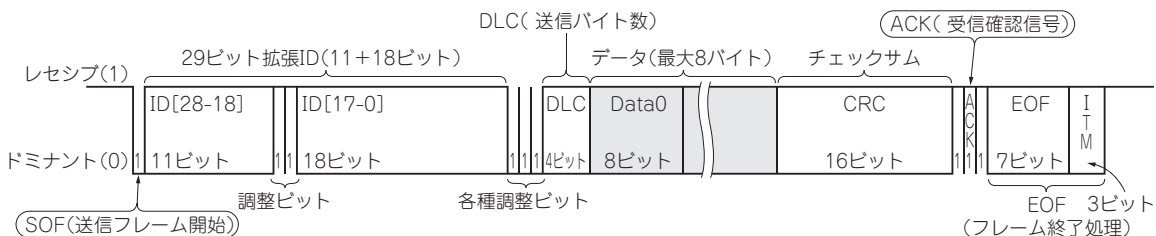


図1 CAN拡張IDフォーマット