



IW74IR-AR

Arduino 赤外線リモコン・家電操作シールド(IR シールド)

東京デバイスズ

Rev. 1.2.1

Arduino 赤外線リモコン・家電操作シールド(通称、IR シールド)は、家電などによく使われている赤外線リモコンの機能を、Arduino に簡単に追加できる拡張シールドです。わずかなプログラムを書くだけで、Arduino から家電を操作したり、逆に家電のリモコンを使って Arduino を操作したりできます。また、複数台の Arduino の赤外線通信機能としても活用できます。



注意事項： 本製品は、取扱いに必要な専門的知識を持つ技術者の研究開発・実験・試作等を利用目的として設計されています。機器への組込や長時間運用の信頼性は未検証です。必要がある場合には十分な試験・検証を行ってください。人命や財産に重大な損害が予想される用途には使用できません。本製品の仕様および本文書の内容は予告なく変更される場合があります。

1. Arduino への接続方法

IR シールドのピン配列は、Arduino の標準的なボード(Arduino Uno, Duemillanove 等)に対応しています。その他の Arduino をお使いの場合も、+5V ピン、GND ピン、デジタル 0 番ピン、デジタル 1 番ピンを接続すれば使用できます。

IR シールドを搭載した Arduino にスケッチを書き込む場合には、IR シールド上にある「SW1」と印刷されたディップスイッチをすべて OFF にしてください。DIP スイッチ上に「1 2 3 4」と刻印されている側が OFF です。OFF にしないでスケッチを書き込もうとすると書き込みエラーとなります。

2. 使用方法

IR シールドは、赤外線信号が照射されると「IR パケット」と呼ばれるバイナリデータを Arduino に向けてシリアル送信します。逆に、Arduino から IR パケットを受信すると、赤外線信号を発信します。

2.1. IR シールドから Arduino への通信 (Arduino で赤外線信号を受信する)

Arduino のプログラムからシリアルバッファを常に監視してください。IR パケットが到達すると、まず Sync 信号と呼ばれる 0xFF というデータを受信されます。Sync を受信したら、IR パケットの仕様(後述)に従って、続けて送信されてくるシリアルデータをすべて受信して下さい。格納した IR パケットはプログラム上で自由に扱ってください。例えばストレージに保存したり、ネットワーク経由で配信したり、などです。

参考までに、IR シールドは赤外線ヘッダーの検知後、トレーラーの OFF 時間が約 6.6msec を超えたら受光を終了し直ちに IR パケットをシリアル送信する仕様となっています。

2.2. Arduino から IR シールドへの通信 (Arduino から赤外線信号を出す)

IR シールドは常に Arduino からのシリアル信号を監視しています。IR シールドは Arduino から送られてくる Sync 信号を監視し、Sync 信号を検知すると IR パケットを受信します。チェックサムが正しい事を確認すると直ちに赤外線を発光し、赤外線の発光が終了するタイミングで ACK 信号(正常に赤外線を発光したという合図)を Arduino に返します。ACK 信号は 1 バイトで、Arduino から送信した IR パケットのチェックサムと同じ値を返します。

なお、IR シールドは赤外線の発光中は新たな IR パケットを受信することができません。連続的に IR パケットを送信して赤外線を発光させたい場合は、一度 ACK を受信してから、次のデータを IR シールドに送信して下さい。

3. 信号の受信の調節(スレッシュホールドレベルの設定)

リモコンの信号規格は各メーカーによって異なり、標準化されていません。そのため IR シールドを使うには、リモコンのデータを受信するために「閾値設定」と呼ばれる設定が必要になります。ここで言う閾値とは、赤外線信号のパルス幅から“0”と“1”と判定するための時間です。適切な閾値はリモコンごとによって変わります。

IR シールド上には「SW1」と印刷されたディップスイッチがあります。閾値は、スイッチに記載されている「1 2 3 4」の 4 ビットで指定します。値は 0001~1111(十進数で 1~15)の範囲で指定できます。このディップスイッチの値と閾値の時間(μ 秒)への換算式を式 1 に示します。

$$\text{閾値の時間} = \text{ピンの値}(1\sim 15) \times 4 \times 26.3 \mu \text{秒} \cdots(\text{式 1})$$

大よその目安として、まずは1,000usec付近を設定してから微調整しましょう。例えば2ビット目と4ビット目をONにすると、二進数で0101となり、これは十進数で10ですから、式1より閾値は1,052 μ 秒となります。

IR シールドがリモコンの値を正しく受光しているかを知る目安として、赤色の LED があります。この LED はシリアル信号の受信エラーが発生した時に光るほか、赤外線の閾値が正しく設定されていない場合にも光ります。

閾値のピンを適切に設定するとエラーは消えます。稀に、閾値をどのように設定してもエラーが消えないリモコンがあります。その様なリモコンは IR パケットで表現できない仕様のもので IR シールドで取り扱う事はできません。

4. 赤外線リモコン信号の解説

赤外線リモコンは、人間の目には見えない波長の光(=赤外線)を使って情報を伝えます。信号は赤外線 LED が38kHz(1秒間に38,000回、26.3usec間隔)で高速に点滅することで表現されます。一般的には、LED が38kHzで点滅している時がON、LED が完全に消灯している時がOFFとなります。

ここで、実際のリモコン信号のONとOFFがどの様に組み合わせられているか見てみましょう。Figure 1は、赤外線受光器によりリモコン信号を受信して、オシロスコープでキャプチャした物です。

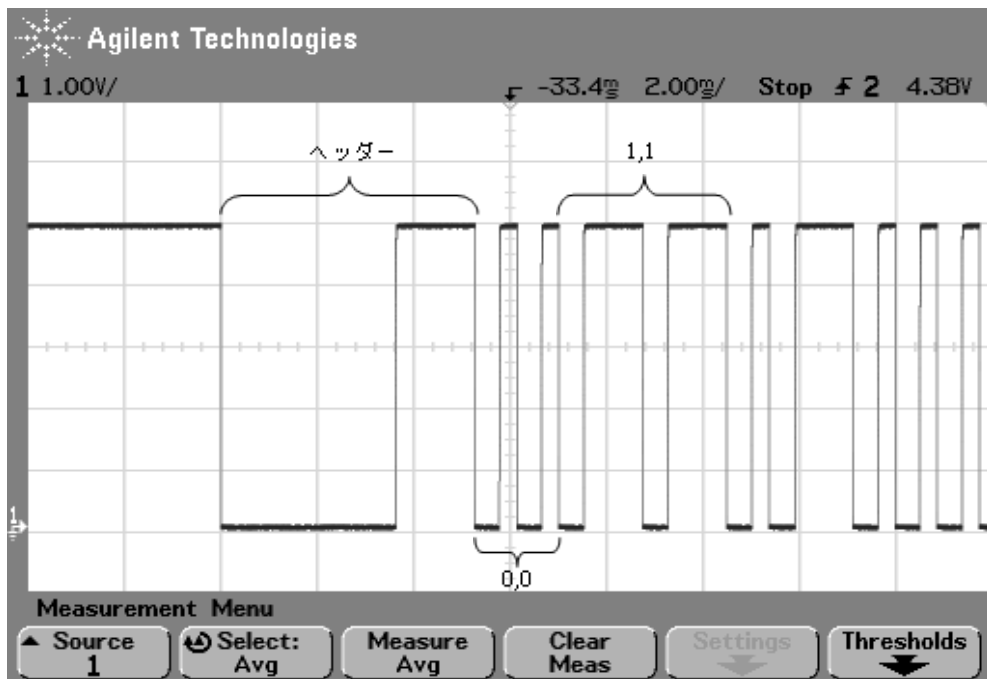


Figure 1 リモコン信号の開始部

使用した赤外線受光器は負論理ですから、Figure 1はONが0Vで、OFFが5Vとして表現されています。なお、赤外線受光器は38kHzの点滅(バースト信号)を滑らかにした信号を出力しますので、Figure 1では38kHzのパルスは見えません。

Figure 1は、あるリモコン信号の開始部を拡大しています。多くのリモコンの特徴として、最初に比較的長いONとOFF(パルス)を送信します。これはヘッダー (またはリーダー)と呼ばれ「これからデータを送りますよ」と言う合図に

なっています。Figure 1 の例では、ヘッダーに続いて比較的短い ON と OFF(パルス)が 2 回続き、次に OFF が長めの ON, OFF(パルス)が 2 回続いています。

このとき、注目するところはパルスの長さです。一般的には、短いパルスの組を“0”とし、ON か OFF どちらか長い物が含まれるパルスを“1”と定義します。その様に信号を読んでいくと、ヘッダーのあと 0,0,1,1,0,1,0,0,0…とデータ本体を読む事ができます。

次に、Figure 2 に信号の全体像を示します。このリモコンの信号は全体で約 50msec の発光時間です。発光時間はデータの長さによって異なり、短い物で数 msec、長い物では数百 msec の物まで色々あります。

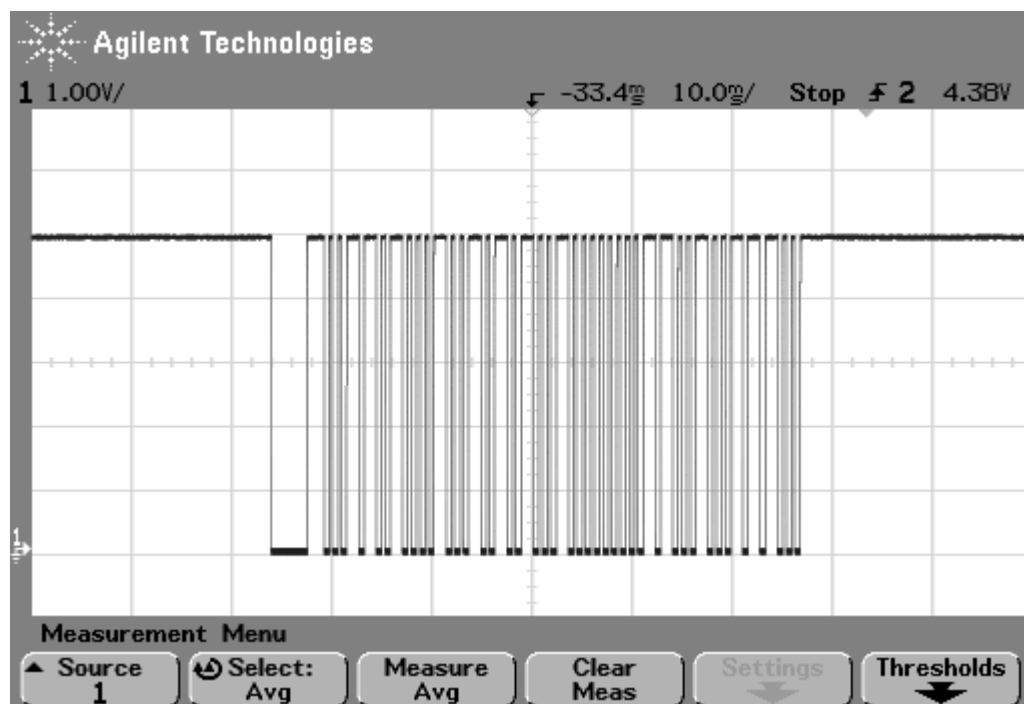


Figure 2 リモコン信号の全体像

Figure 3 に信号の終了部を示します。“0”“1”の羅列の最後には、ある一定時間の ON があり、次に長い OFF が続きます。これをトレーラーと呼びます。長い OFF がどこまで続くのかは次の信号のヘッダーが到達しないと分かりません。次の信号が送信されなければ OFF の時間は不明です。IR シールドでは、トレーラーがある一定の長さまで続くと 1 パケットの受信が終了したとみなします。

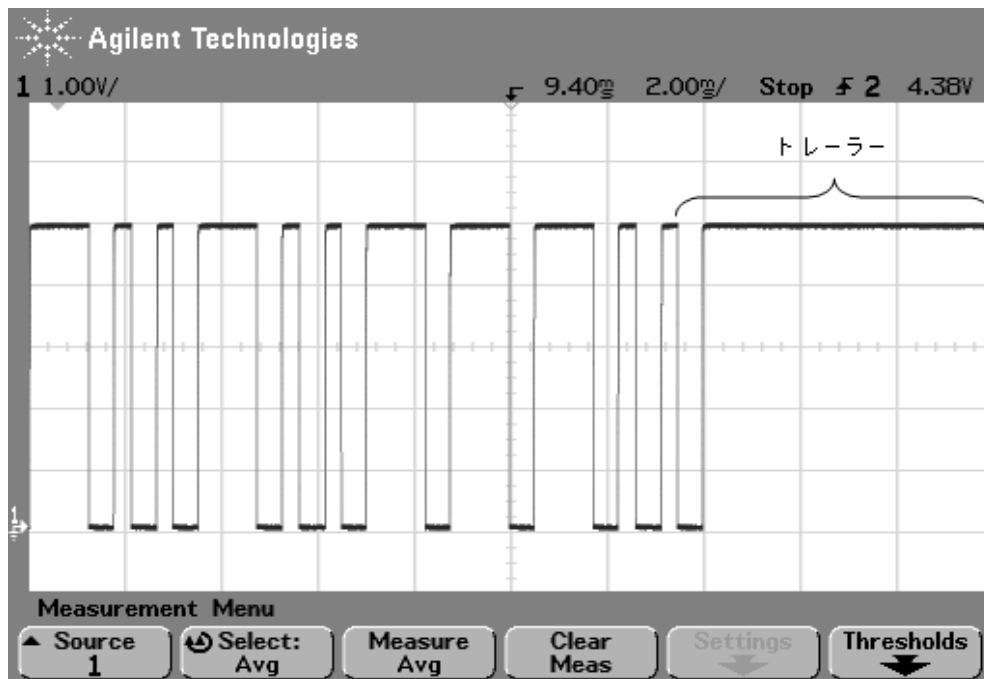


Figure 3 リモコン信号の終了部分

この様に、多くの赤外線リモコンの信号は、ヘッダー部、データ部、トレーラー部から構成されています。データ部の“0”“1”はパルスの長さで決まる物が一般的です。IR シールドでは、このような一般的なリモコンのデータ構造に対応した「IR パケット」を定義して、Arduino との通信に使用します。

なお、リモコン信号は標準化されていません。上でご説明した信号の形式は比較的多く存在する物ですが、中にはパルスの長さの組み合わせがもっと複雑な物も存在するようです。(※本製品はそのような複雑な信号には対応していません。)

5. IR パケット仕様

IR シールドは赤外線を受光し IR パケットと呼ばれる標準的なデータ形式に変換します。逆に IR パケットを受け取ると赤外線信号を発光します。IR パケットは、前述した比較的多数のリモコン信号を表現するためのデータ構造で、IR シールドに対してシリアル通信(19,200bps、歩調同期、8ビット、パリティビット無し、ストップビット1)をするために定義されました。IR パケットの仕様を Table 1 に示します。

Table 1 IR パケット仕様

バイト数	データの意味	備考
1	Sync 信号	0xFF 固定値
1	ヘッダーONの長さ (上位8ビット)	unsigned short 26.3usec(38kHzの周期)を単位とする数値 最大 1520 まで(約 20msec)
1	ヘッダーONの長さ (下位8ビット)	
1	ヘッダーOFFの長さ (上位8ビット)	
1	ヘッダーOFFの長さ (下位8ビット)	
1	データ1のONの長さ	unsigned char 26.3usec(38kHzの周期)を単位とする数値 最大 255 まで(約 6msec)
1	データ1のOFFの長さ	
1	データ0のONの長さ	
1	データ0のOFFの長さ	
1	トレーラーONの長さ	
1	トレーラーOFFの長さ	unsigned short 最大 400 ビットまで
1	データのビット数 (上位8ビット)	
1	データのビット数 (下位8ビット)	unsigned char 最大 50 バイトまで
データのビット数/8 の切り上げ	データ部	
1	チェックサム	unsigned char 上記全てのデータを加算したときの低位8ビット

6. 製品カスタムサービス

東京デバイスズはお客様のニーズに応じて基板外形や機能・性能をカスタムいたします。詳しくは東京デバイスズ Web サイトの「製品カスタム」メニューからサービス内容をご確認ください。

東京デバイスズ株式会社
Copyright © 2024 Tokyo Devices, Inc. All rights reserved.
tokyodevices.jp