

IR Signale auswerten

Ziel

Die Signale einer Infrarot (IR) Fernbedienung sollen ausgelesen und interpretiert werden. Dazu wird ein IR Empfänger mit dem Arduino verbunden. Das Signal aus dem Empfänger wird gesampelt und aus den Abständen zwischen den Pegelwechseln der gesendete RC5 Code bestimmt.

Vorgehensweise

Das IR Signal aus der Fernbedienung besteht aus einer etwa 38kHz schnellen Schwingung. Diese Schwingung wird mit einer niedrigeren Frequenz (etwa 1kHz) an- und abgeschaltet. Der Empfänger liefert als Ausgangssignal ein „LOW“ solange das 38 kHz IR Signal besteht und ein „HIGH“ wenn kein IR Signal empfangen wird (siehe Abbildung 1). Verbinden Sie den IR Empfänger mit dem Arduino und überprüfen Sie, dass der IR Empfänger ein auswertbares Signal liefert.

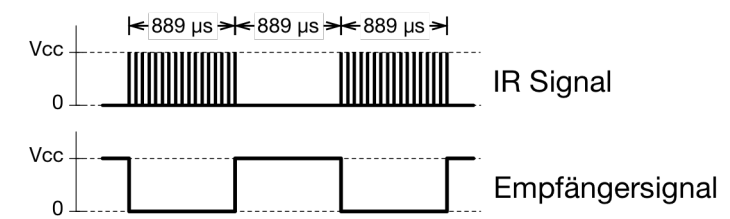


Abbildung 1) Ausgangssignal des IR Empfängers

Die Nachrichten, welche die IR Fernbedienung übermittelt entsprechen dem RC5 Standard. Sie sind insgesamt 14 Bit lang und „Manchester kodiert“. Beim Manchester Encoding ist die Information in den Flanken des Signals kodiert. Eine steigende Flanke entspricht dabei der logischen „0“, eine fallende Flanke der logischen „1“ (siehe Abbildung 2). Schreiben Sie nun ein Programm, welches das Empfängersignal sampelt und aus den Abständen der Flanken den übertragenen RC5 Code rekonstruiert.

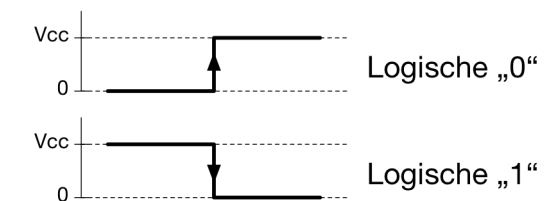


Abbildung 2) Manchester Encoding

Ein Beispiel für die Zusammensetzung eines RC5 Codes als Signal am Empfänger ist in Abbildung 3 gezeigt. Die Nachricht beginnt mit 2 Startbits (S1, S2) – diese sind immer auf logisch „1“ gesetzt. Anschließend folgt

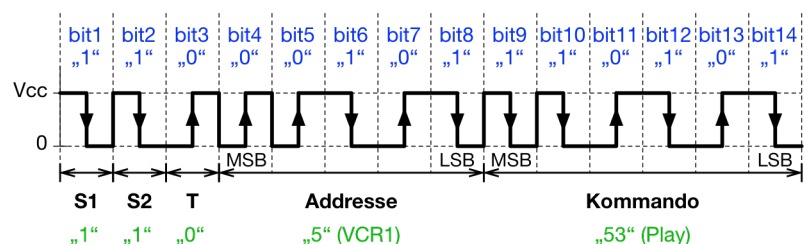


Abbildung 3) Struktur einer RC5 Nachricht

das sogenannte „Toggle Bit“. Diese Bit ändert immer dann seinen Wert, wenn eine Taste auf der Fernbedienung ein weiteres Mal gedrückt wird. Dadurch lässt sich mehrfaches Drücken einer Taste von einer Unterbrechung des Signals unterscheiden (z.B. durch Hindernisse im Signalweg). Die nächsten 5 Bit kodieren die Adresse und die letzten sechs Bit das Kommando (LSB = Least Significant Bit, MSB = Most Significant Bit).

Schreiben Sie ein Programm, das zuverlässig das Toggle Bit, die Adresse und das Kommando von jeder neuen Nachricht ausgibt.

Sinnvoll ist eine Aufteilung in drei separate Funktionen:

- Abtasten der Flanken und Speicherung der 28 Bits für eine Nachricht
- Eine Funktion für das Manchester Decoding, die aus den 28 Bits 14 macht
- Eine Funktion, zum Rausziehen der Werte Für Toggle Bit, Adresse und Kommando

Vorbereitung

Beschäftigen Sie sich mit Manchester Encoding und dem Aufbau des Signals. Entwerfen und implementieren Sie zu den drei beschriebenen Funktionen entsprechende Algorithmen zur Dekodierung des Signals aus den Samples. Beschäftigen Sie sich mit der Speicherausrichtung von Strukturen in C (Stichwort: *bit field alignment*).

Achtung

Der Empfänger empfängt auch Fehlsignale. Stellen Sie durch geeignete Maßnahmen sicher, dass nur korrekte und vollständige Nachrichten ausgewertet werden. **Es dürfen keine Bibliotheken und keine Interrupt-Routinen verwendet werden.**

Notengebung

4,0 (Anwesend); 3,0 (Das kodierte Signal wird erfasst); 2,3 (+ Das Signal wird dekodiert aber nicht zerlegt); 2,0 (+ Korrekt zerlegte Komponenten des Signals); 1,7 (+ Korrekte Interpretation des Toggle Bits zur Ausgabe neuer Nachrichten); 1,3 (+ Code sauber geschrieben und dokumentiert); 1,0 (+Verwendung von *bit field alignment*, möglichst speichersparende Implementierung)

Wichtige Funktionen

- `micros()`
- `bitClear(...)`
- `bitSet(...)`

Sie brauchen

- Arduino Board, USB Kabel, IR Empfänger, Steckbrücken, Fernbedienung