

The Tinusaur Project

IRRELib

i

Rate This

IRRELib is a library for working with IR (infrared) receivers and processing remote control commands. The library handles the NEC infrared remote control protocol used in wide variety of devices today. The code is written specifically for ATtiny85 and is part of the Tinusaur project. As a receiver in the examples is used the TSOP4838 by Vishay.

Sending and Receiving

The transmission of the IR commands requires only a microcontroller and an infrared LED, available from a wide variety of sources. The easiest way to send commands is to use a general purpose (universal) remote control.

The reception of the modulated commands for the NEC protocols is easily accomplished with specialized IR receivers, most readily available from Sharp Corporation and Vishay Intertechnology. These receivers include a photo-diode, an automatic gain control (AGC) circuit, and a demodulator. The demodulated signal is then decoded with a microcontroller.

TSOP4838 Receiver

(https://tinusaur.files.wordpress.com/2016/06/tsop4838-infrared-receiver_c700x500.jpg) As part of the development of this library, an infrared receiver is used – more specifically the Vishay TSOP4838 that works at 38 kHz.

The TSOP4838 are miniaturized IR receiver modules for infrared remote control systems. A PIN diode and a preamplifier are assembled on lead frame, the epoxy package contains an IR filter. The demodulated output signal can be directly connected to a microprocessor for decoding. They are optimized to suppress almost all spurious pulses from Wi-Fi and CFL sources. They may suppress some data signals if continuously transmitted.

More technical information about the TSOP4838 receiver as PDF is available at:

<http://www.vishay.com/docs/82459/tsop48.pdf>
 (<http://www.vishay.com/docs/82459/tsop48.pdf>)



The Protocol

There are many protocols for sending and receiving IR commands.

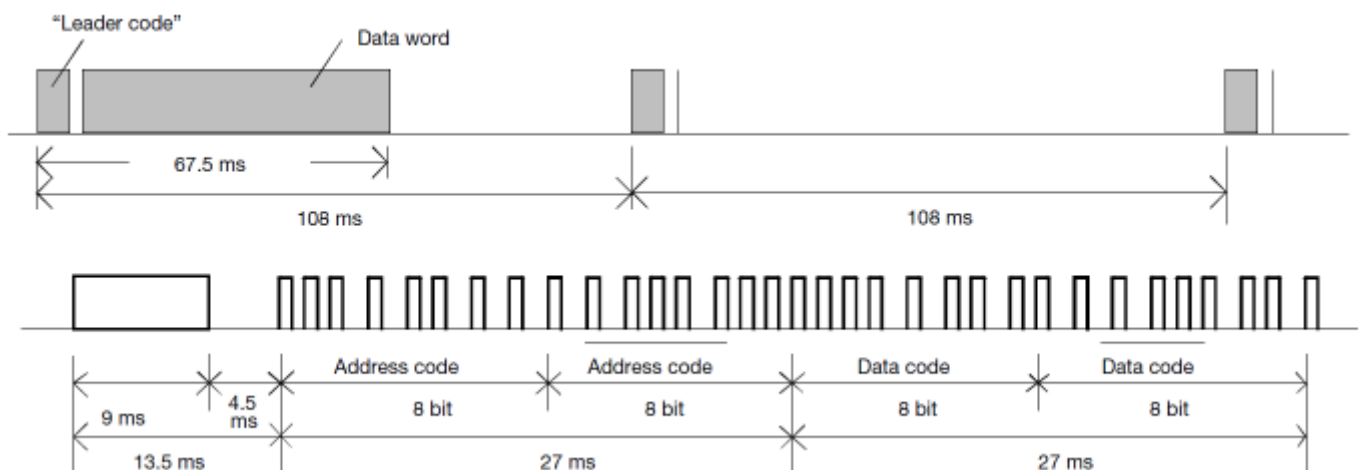
The most commonly used protocol is the NEC protocol, which specifies a carrier frequency of 38 kHz. The NEC protocol is used by the vast majority of Japanese-manufactured consumer electronics. In the NEC protocol, each manufacturer is assigned a unique code that is contained in the transmitted command, avoiding the possibility of false triggering by other remote handsets.

This library uses the NEC protocol.

The received data consists of 32 bits or 4 bytes where

- 1st byte is the device id
- 2nd byte is the same id but inverted
- 3rd byte is the command code
- 4th byte is the command but inverted

It is important to note that not all transmitters (remote controls) follow that format – sometimes the supposed inverted bytes contain other data instead.



(https://tinusaur.files.wordpress.com/2016/06/nec_transmission_code_c1400x500.png)

NEC Transmission Code

More technical information about the communication protocols as PDF is available at:

<http://www.vishay.com/docs/80071/dataform.pdf> (<http://www.vishay.com/docs/80071/dataform.pdf>)

The Library

The library is receiving the commands the calls a user defined function every time there is a command providing a data structure (rather a pointer to it) that contains the data – the 4 bytes.

Here is a source code fragment showing the user defined receiving function

```
1  void irrelib_received_userfunc(irrelib_data_p received_data) {
2      switch ((*received_data).status) {
3          case IRRELIB_STATUS_OK:
4              // Every time a command is received ...
5              PORTB ^= (1 << LED5_PORT); // Flip the bit
6              break;
7          case IRRELIB_STATUS_REPEAT:
8              // Every time a command is repeated ...
9              break;
10         case IRRELIB_STATUS_ERROR:
11             // Every time there is an error ...
12             break;
13     }
14 }
```

The command is accessible through a **(*received_data).com_id** expression.

Since this library (the projects) is work in progress the best ways to see how it works and how to use it is to look at the source code – it has a lot of comments. It also has a sample testing program that lights up some LED's upon receiving commands.

External Resources

The source code of the IRRELib library is available at:

<https://bitbucket.org/tinusaur/irrelib> (<https://bitbucket.org/tinusaur/irrelib>)

Credits

As it is mentioned in the source code part of this work was inspired by and based on the “ATtiny85 IR Remote Control Receiver” by David Johnson-Davies

Technoblogy <http://www.technoblogy.com> (<http://www.technoblogy.com>)

IR Remote Control Receiver: <http://www.technoblogy.com/show?V6F> (<http://www.technoblogy.com/show?V6F>)

IR Remote Control Tool (NEC): <http://www.technoblogy.com/show?UVE> (<http://www.technoblogy.com/show?UVE>).

Other projects and useful links

Consumer IR

https://en.wikipedia.org/wiki/Consumer_IR (https://en.wikipedia.org/wiki/Consumer_IR)

NEC Protocol IR (Infrared) Remote Control With a Microcontroller

<http://www.circuitvalley.com/2013/09/nec-protocol-ir-infrared-remote-control.html>
(<http://www.circuitvalley.com/2013/09/nec-protocol-ir-infrared-remote-control.html>)

NEC protocol compatible infrared remote decoder lib for Atmel AVR

<https://github.com/MalteP/nec-decoder> (<https://github.com/MalteP/nec-decoder>)

TinyPCRemote – An ATtiny85 Based Infrared PC remote control

Projects

How to use the IR library with an attiny

<http://www.instructables.com/id/Attiny-IR-library/> (<http://www.instructables.com/id/Attiny-IR-library/>)

ATtiny & IR remote control Wiebelbot

<http://www.instructables.com/id/Wiebelbot/> (<http://www.instructables.com/id/Wiebelbot/>)

Infrared Dedicated Decoder

<https://www.hackster.io/rayburne/infrared-dedicated-decoder-c10052>
(<https://www.hackster.io/rayburne/infrared-dedicated-decoder-c10052>)

Scavenger Hunt Beacon

<http://www.openhardwarehub.com/projects/42-Scavenger-Hunt-Beacon>
(<http://www.openhardwarehub.com/projects/42-Scavenger-Hunt-Beacon>)

Rickrolling remote control prank

[Rickrolling remote control prank](http://hackaday.com/2012/09/10/rickrolling-remote-control-prank/) (<http://hackaday.com/2012/09/10/rickrolling-remote-control-prank/>)

TinyPCRemote – An ATtiny85 Based Infrared PC remote control

(<https://nathan.chantrell.net/20121014/tinypcremote-an-attiny85-based-infrared-pc-remote-control/>)

TinyPCRemote – An ATtiny85 Based Infrared PC remote control

Here's a cheap way to build your own fully customisable infrared PC remote control. If you already have a suitable infrared remote control going spare you can build one of these for under £4, it will allow you to use most infrared remote controls to issue keyboard commands (single characters or a string) on your ...
Continue reading



nathan.chantrell.net

67

GitHub: [**https://github.com/nathanchantrell/TinyPCRemote**](https://github.com/nathanchantrell/TinyPCRemote)
([**https://github.com/nathanchantrell/TinyPCRemote**](https://github.com/nathanchantrell/TinyPCRemote))

An ATtiny85 Home Automation IR Device

[**An ATtiny85 Home Automation IR Device \(http://falldeaf.com/2013/11/an-attiny85-home-automation-ir-device/\)**](http://falldeaf.com/2013/11/an-attiny85-home-automation-ir-device/)

