

# Backup & Restore of IR Remote Controls

Peter Bubestinger-Steindl  
(p.bubestinger @ AV-RD.com)

November 2019

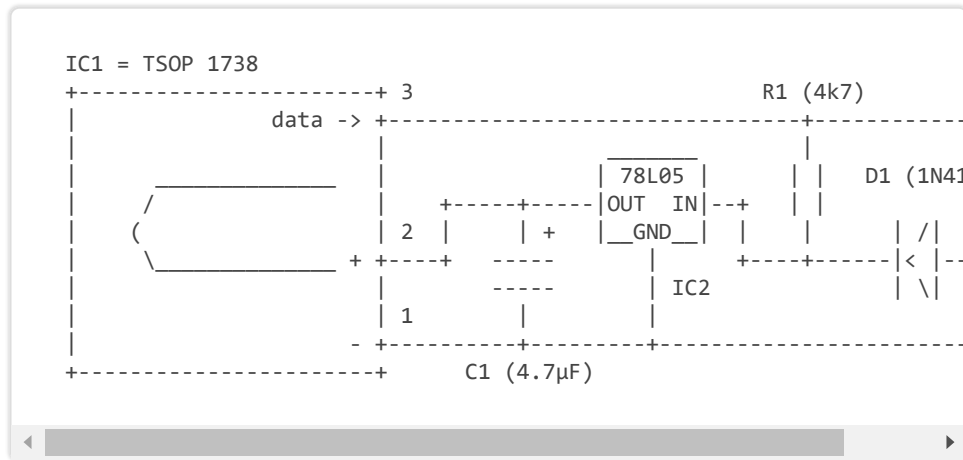
# Initial Motivation

- Used A/V gear: missing remote.
- Some functions require the remote.  
(VHS Rewind?! O.O)

# Requirements

- Must be non-proprietary (HW+SW).
- HW+SW work independent.
- Work beyond end-of-life.
- Any OS.
- Any IR remote.
- Map to (shell) programs.

# Serial Port



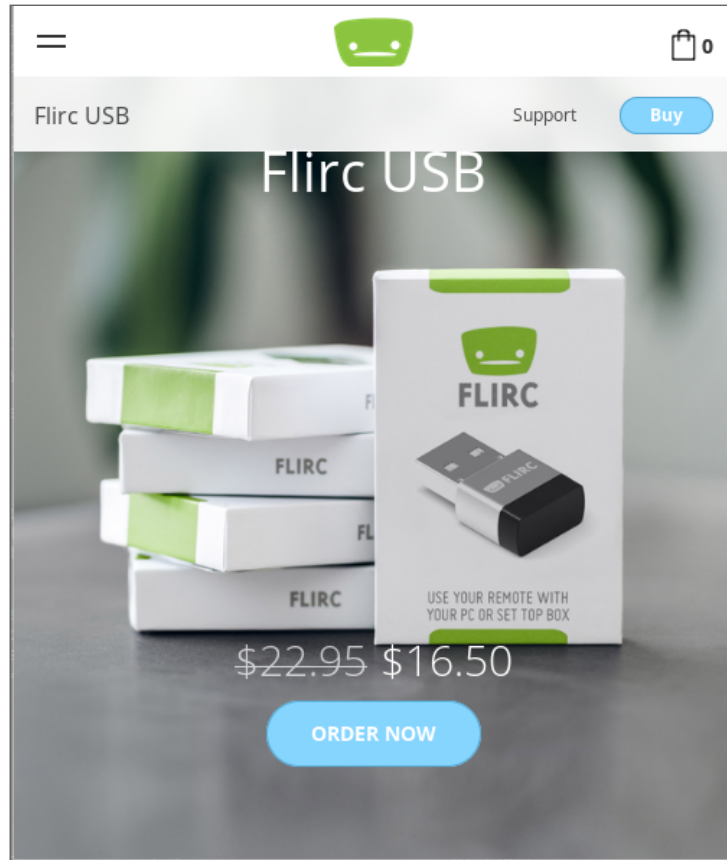
Source: [lirc.org](http://lirc.org)

# Serial Port

*Well, but there is no more serial port...*

What about USB?!

# Flirc?



# Flirc: Don't.

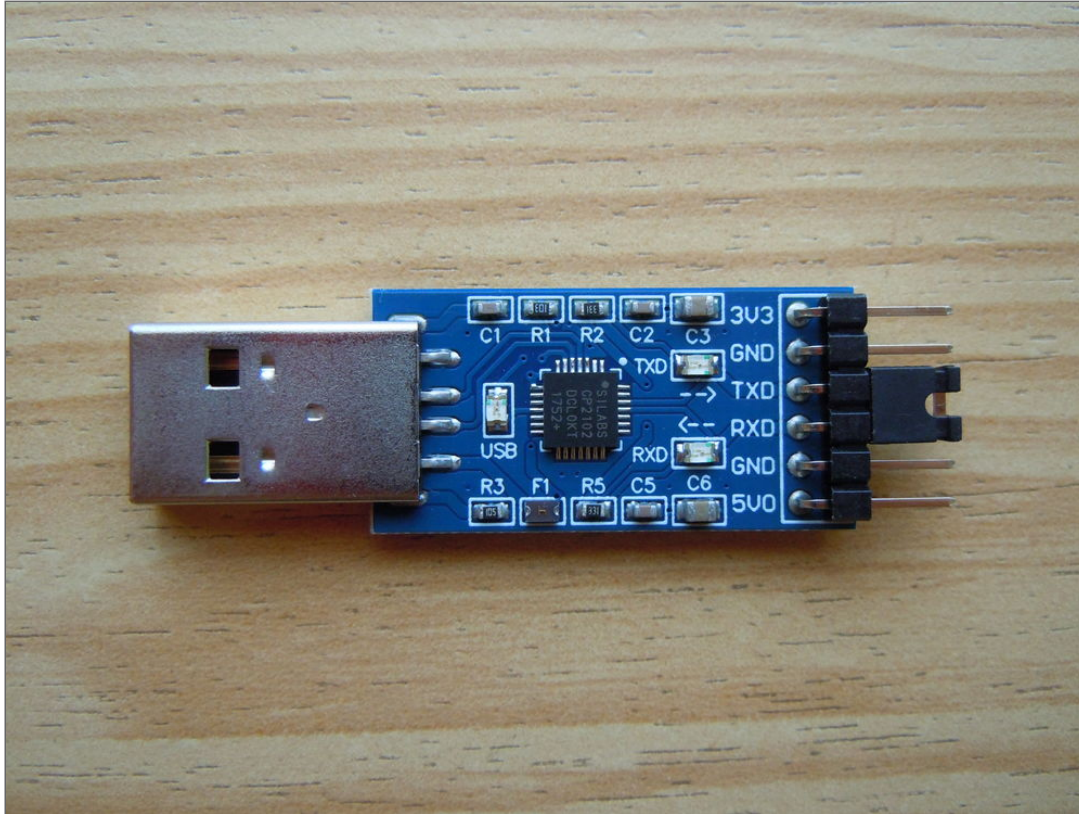
- It's **NOT LIRC!**
- Uses FOSS, but ain't open/FOSS itself.
- **Patented in Feb 2016**

# USB to Serial/UART

- FTDI (**FT232RL**)
- CP21XX (**CP2102**)

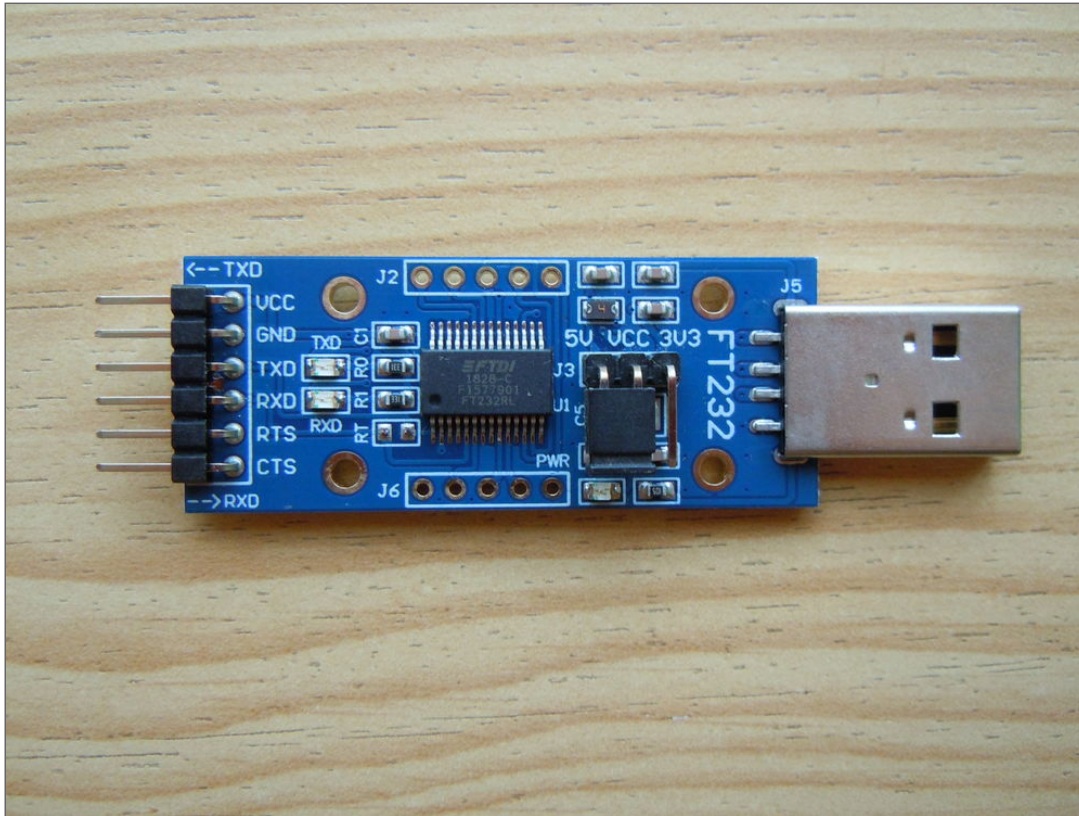


# CP2102



Didn't work.

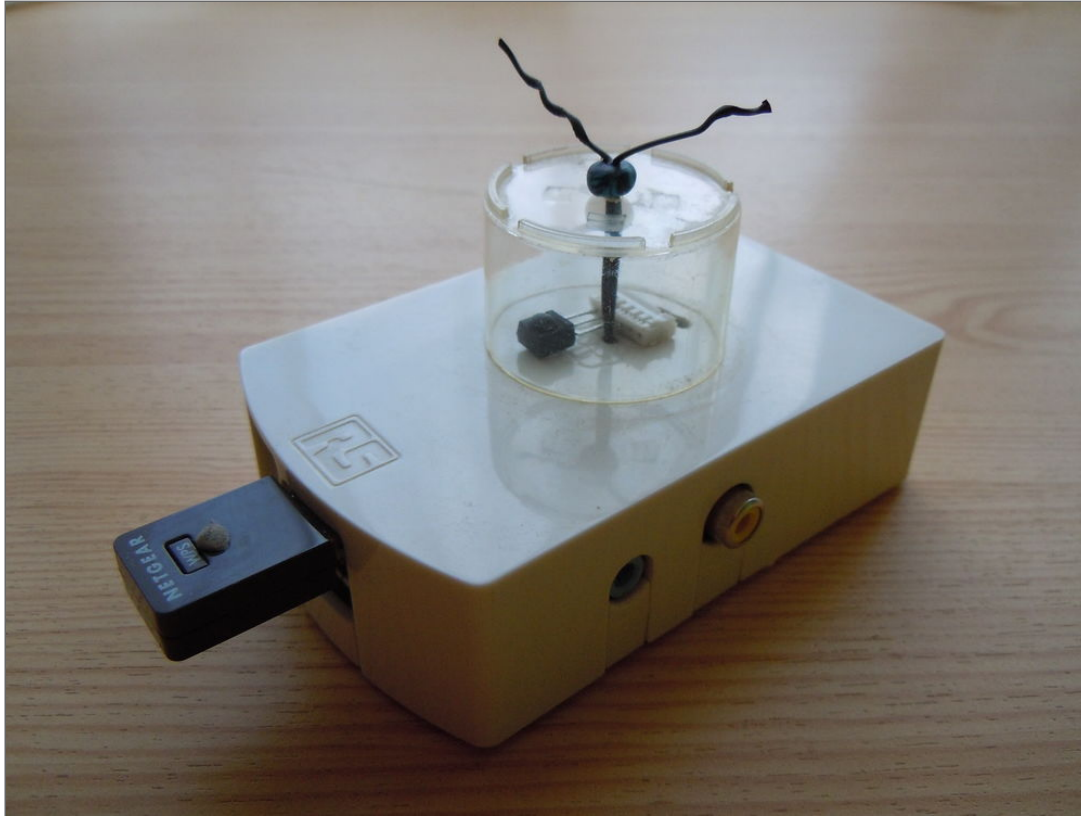
# FT232RL



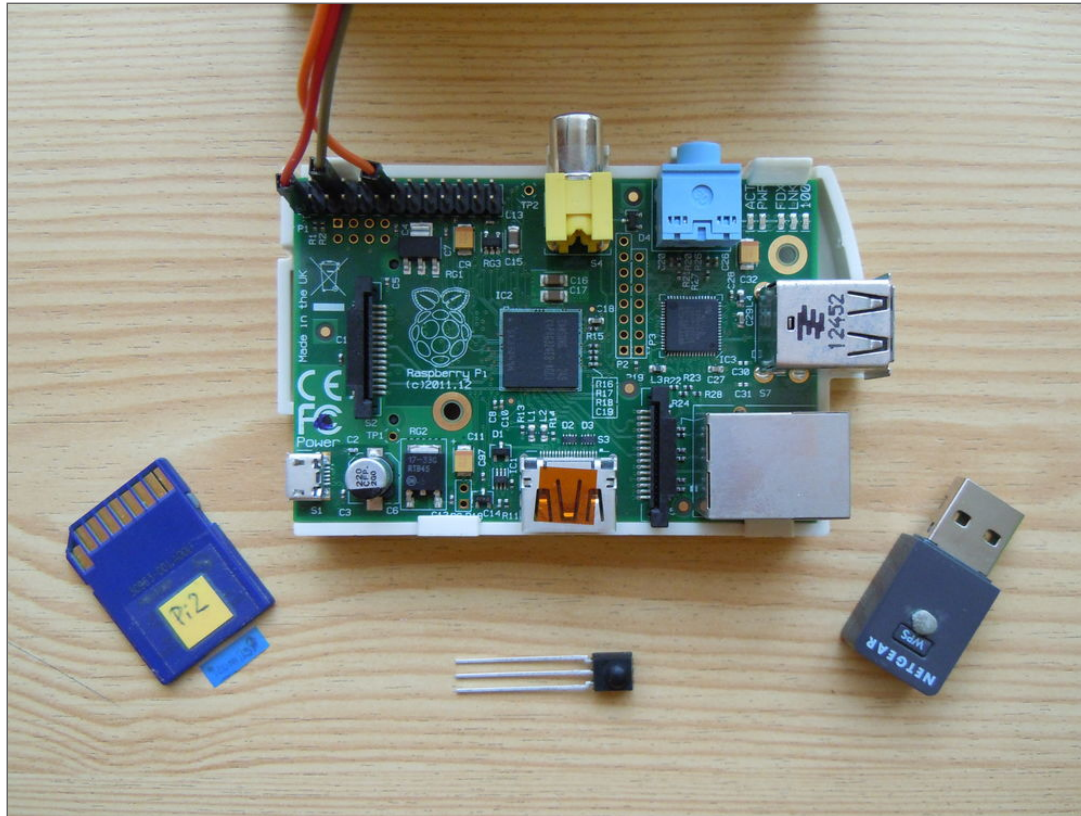
Didn't work reliably...

# Raspberry Pi

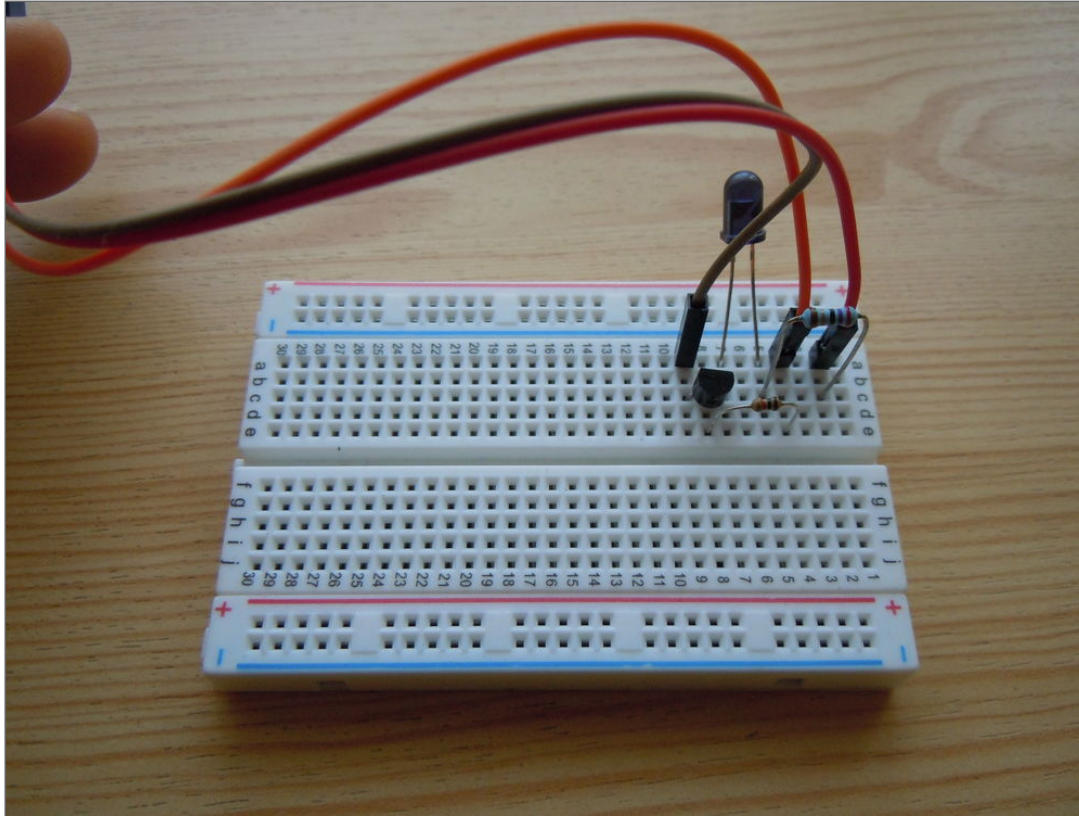
# Receiver (RPI 1)



# Receiver



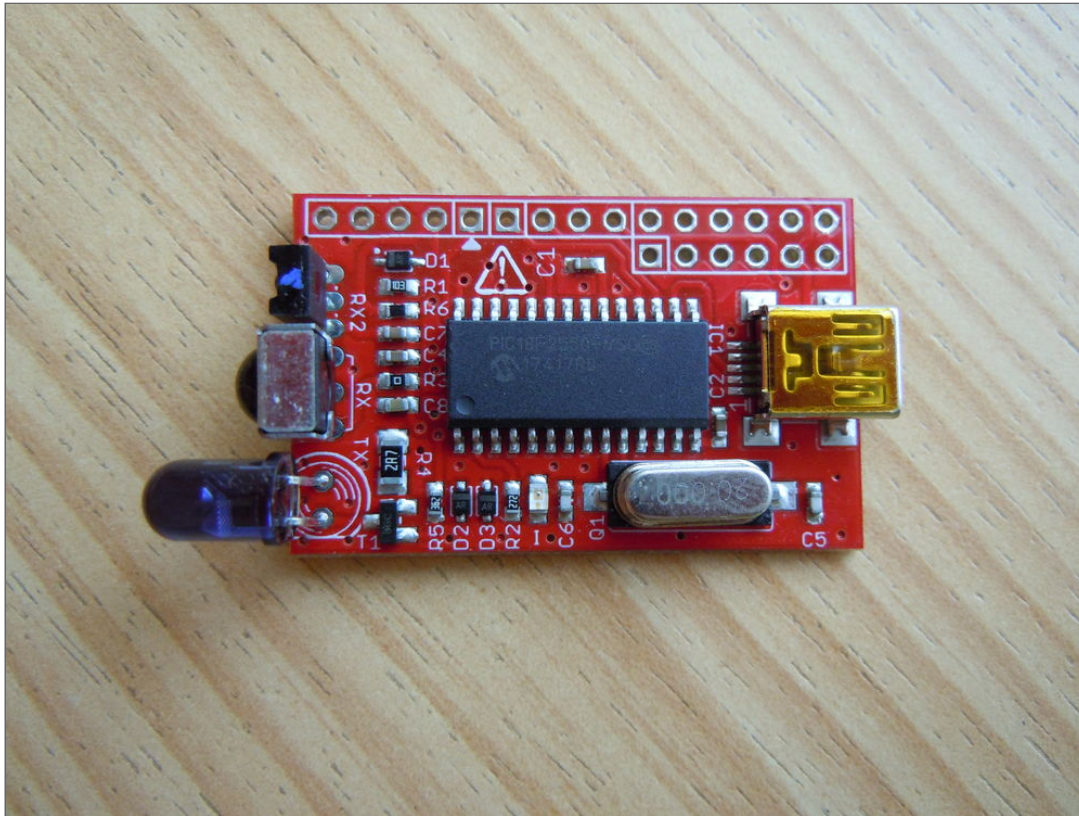
# Sender



Yes, it's a prototype.

# IRToy

What about non-RPi?



# IRToy

- **DangerousPrototypes**
- ca. 20\$ at **Seedstudio**
- Timing on-board
- **Schematics**
- **FOSS (Github)**
- “Just works”
- As sender + receiver



# LIRC

*Configuration?*

# lircd

```
vim
Please take the time to finish this file as described in
# https://sourceforge.net/p/lirc-remotes/wiki/Checklist/
# and make it available to others by sending it to
# <lirc@bartelmus.de>
#
# This config file was automatically generated
# using lirc-0.9.4c(default) on Thu Jan 31 15:17:38 2019
# Command line used: -u Cyberhome_DVD1.lircd.conf
# Kernel version (uname -r): 4.9.59+

begin remote

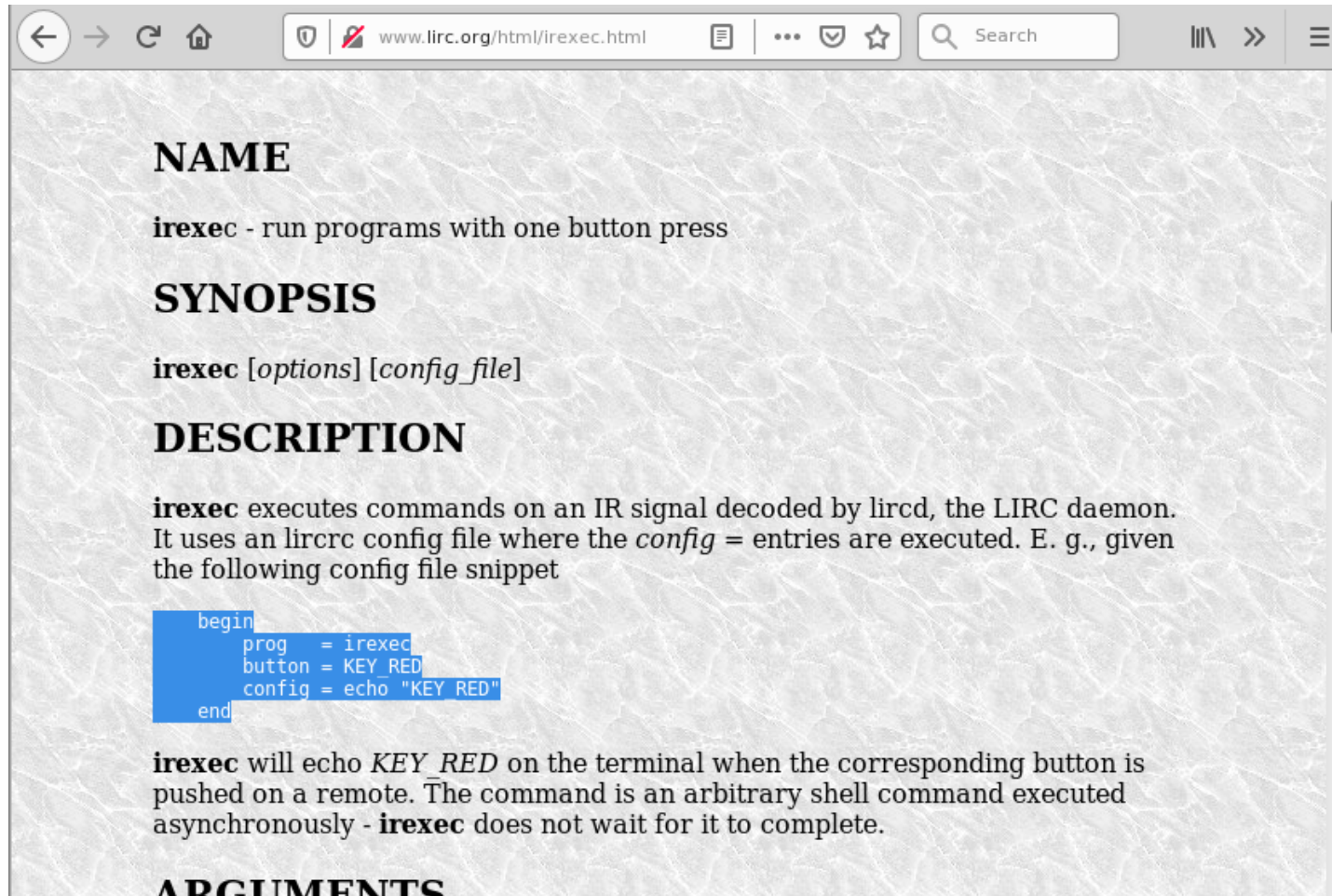
  name Cyberhome_DVD1
  bits 32
  flags SPACE_ENC | CONST_LENGTH
  eps 30
  aeps 100

  header 9051 4460
  one 608 1641
  zero 608 516
  ptrail 602
  repeat 9054 2212
  suppress_repeat 3
  gap 108061
  toggle_bit_mask 0x0
  frequency 38000

  begin codes
    KEY_PLAY 0x4EB318E7
    KEY_STOP 0x4EB39867
    KEY_REWIND 0x4EB3609F
    KEY_FASTFORWARD 0x4EB37887
    KEY_CLEAR 0x4EB38A75
    KEY_SELECT 0x4EB3B24D
    KEY_UP 0x4EB322DD
    KEY_DOWN 0x4EB3B847
    KEY_LEFT 0x4EB338C7
    KEY_RIGHT 0x4EB312ED
    KEY_ENTER 0x4EB33AC5
    KEY_TITLE 0x4EB302FD

1,1 Top
```

# irexec



**NAME**

**irexec** - run programs with one button press

**SYNOPSIS**

**irexec** [*options*] [*config\_file*]

**DESCRIPTION**

**irexec** executes commands on an IR signal decoded by lircd, the LIRC daemon. It uses an lircrc config file where the *config* = entries are executed. E. g., given the following config file snippet

```
begin
prog = irexec
button = KEY_RED
config = echo "KEY RED"
end
```

**irexec** will echo *KEY\_RED* on the terminal when the corresponding button is pushed on a remote. The command is an arbitrary shell command executed asynchronously - **irexec** does not wait for it to complete.

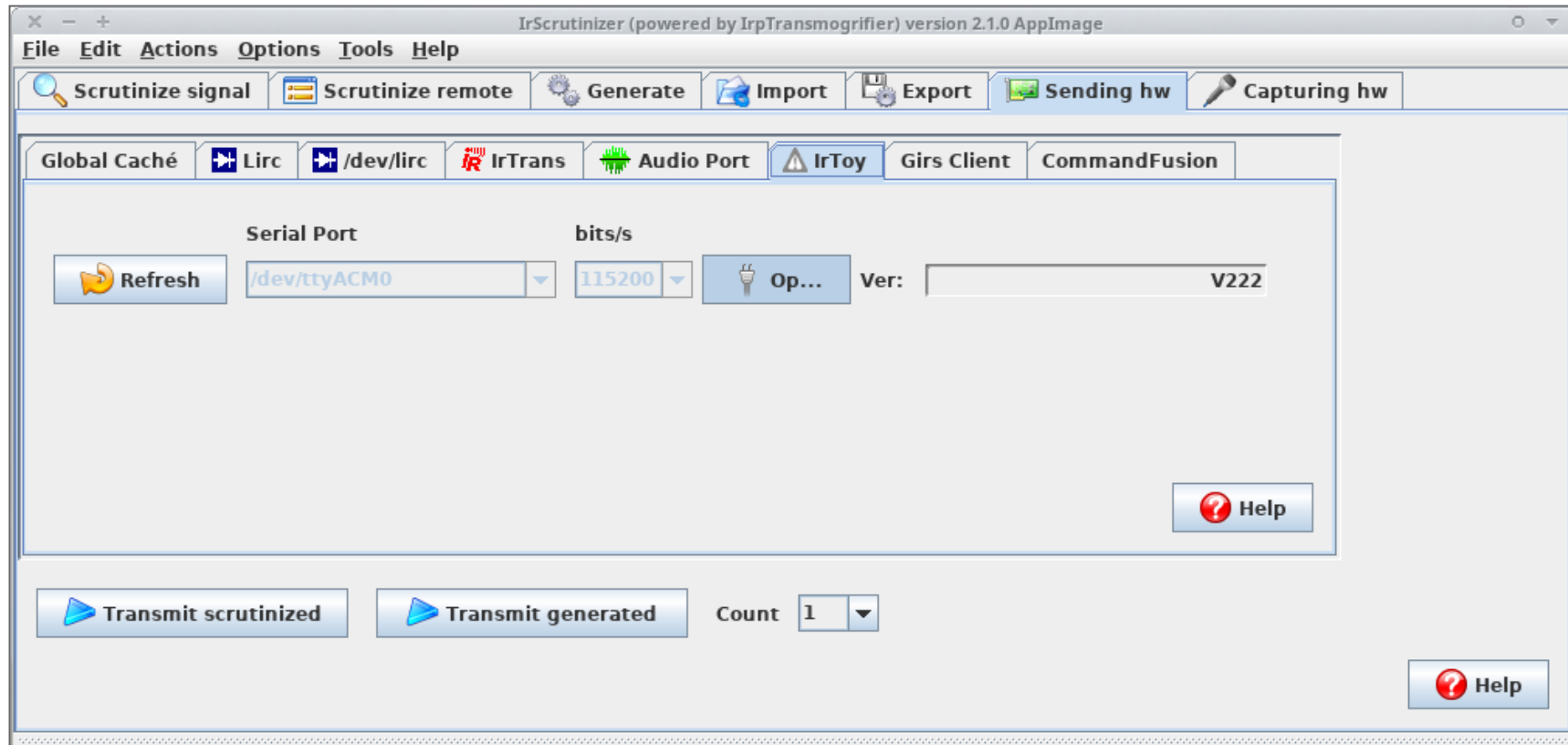
**ARGUMENTS**

# HARCToolbox

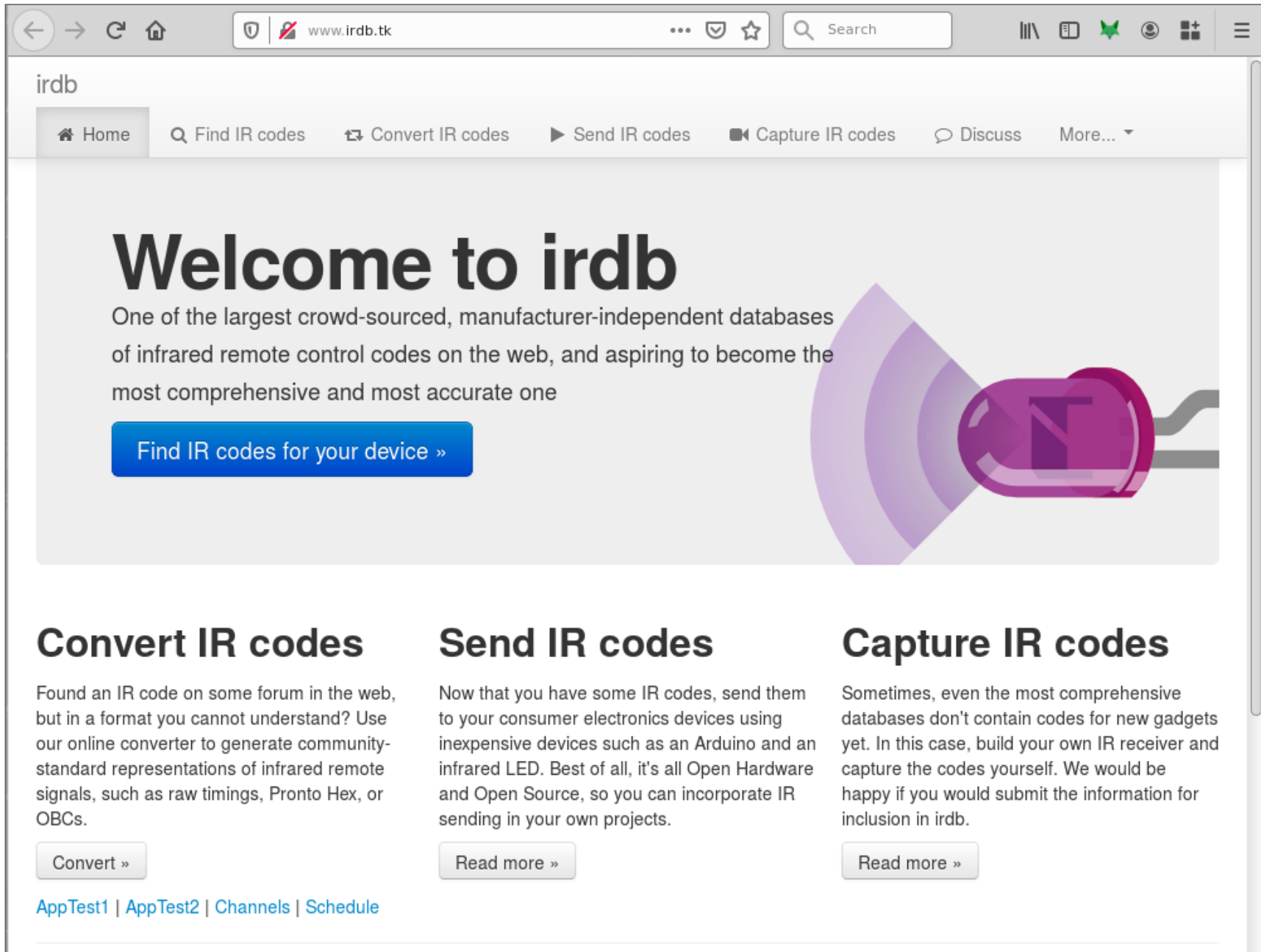
- Website: **harctoolbox.org**
- **Girr**: Universal format for IR commands/remotes
- **IrScrutinizer**
- **IR signal resources**
- ...

Big thanks to **Bengt Mårtensson!**

# IrScrutinizer







The screenshot shows a web browser window with the URL [www.irdb.tk](http://www.irdb.tk). The page title is "irdb". The navigation menu includes: Home, Find IR codes, Convert IR codes, Send IR codes, Capture IR codes, Discuss, and More... The main content area features a large heading "Welcome to irdb" and a subheading: "One of the largest crowd-sourced, manufacturer-independent databases of infrared remote control codes on the web, and aspiring to become the most comprehensive and most accurate one". A prominent blue button says "Find IR codes for your device »". To the right is an illustration of a purple IR remote control with concentric purple circles representing its signal range. Below this are three columns of featured services:

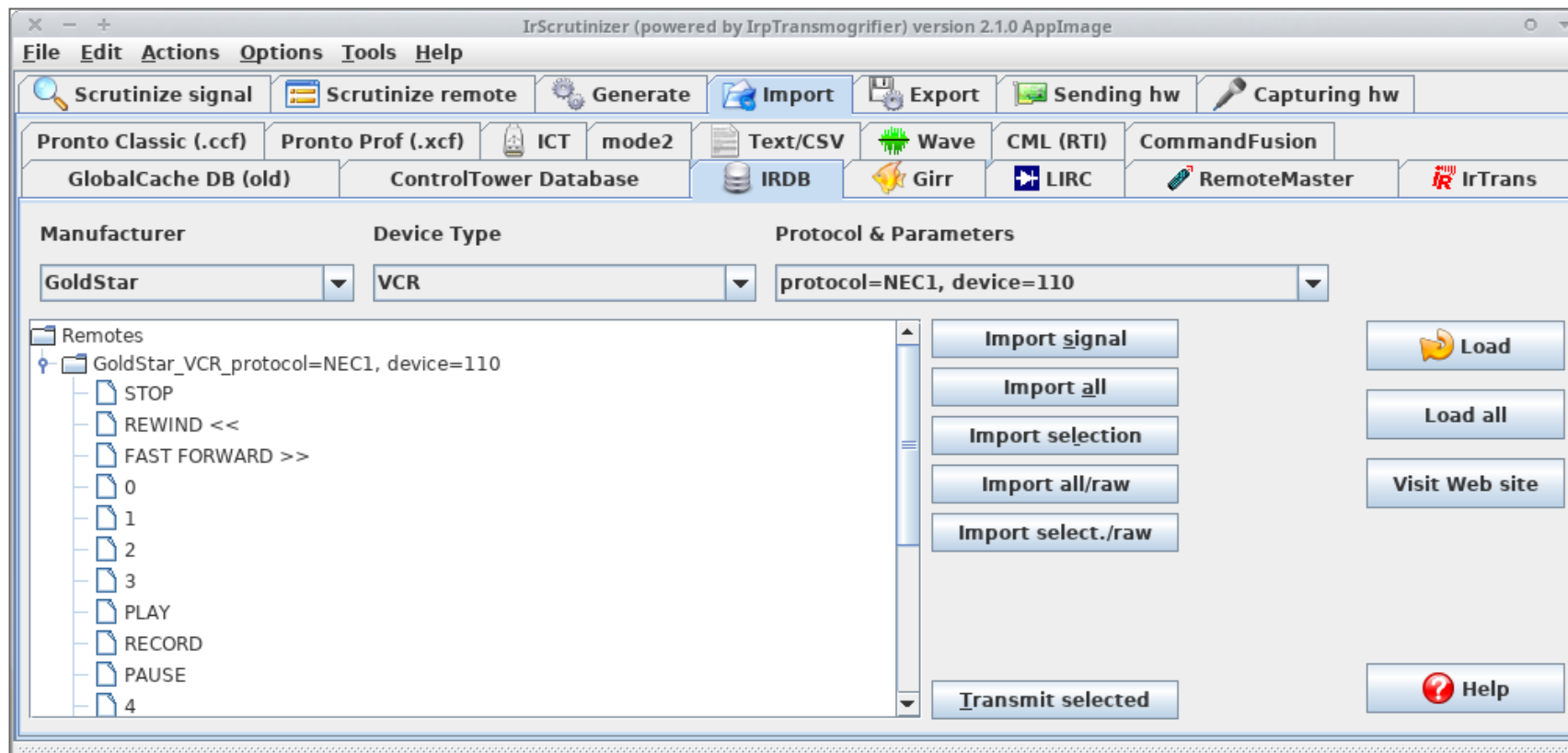
- Convert IR codes**: Found an IR code on some forum in the web, but in a format you cannot understand? Use our online converter to generate community-standard representations of infrared remote signals, such as raw timings, Pronto Hex, or OBCs. Button: "Convert »".
- Send IR codes**: Now that you have some IR codes, send them to your consumer electronics devices using inexpensive devices such as an Arduino and an infrared LED. Best of all, it's all Open Hardware and Open Source, so you can incorporate IR sending in your own projects. Button: "Read more »".
- Capture IR codes**: Sometimes, even the most comprehensive databases don't contain codes for new gadgets yet. In this case, build your own IR receiver and capture the codes yourself. We would be happy if you would submit the information for inclusion in irdb. Button: "Read more »".

At the bottom left, there are links: [AppTest1](#) | [AppTest2](#) | [Channels](#) | [Schedule](#)





# IrDB



# Summary

- Used A/V gear: missing remote.
- Some functions require the remote.
- **Backup & exchange IR-remote backups.**
- Actually: Remote control stuff!
- ...over the network, et al.

# Questions?

# Comments?

IRremoteESP8266