# EEVblog Electronics Community Forum

A Free & Open Forum For Electronics Enthusiasts & Professionals

**Hello horo**

Show unread posts since last visit.
Show new replies to your posts.
March 05, 2025, 08:21:05 am

Entire forum Search

**News:**
Be excellent to each other.

Home Help Search Profile About us My Messages Calendar Links
Members Logout

EEVblog Electronics Community Forum » Profile of hoenicke » Show Posts » Messages

Profile Info Actions

## Show Posts

This section allows you to view all posts made by this member. Note that you can only see posts made in areas you currently have access to.

Messages Topics Attachments Thanked Posts Posts Thanked By User

## Messages - hoenicke

Pages: [**1**]

### 1 Test Equipment / Re: Hantek 6022BE 20MHz USB DSO

« **on:** May 03, 2015, 01:08:39 pm »

**Quote from: momus on May 02, 2015, 08:00:54 pm**

> I think I found something quite valuable today :
> https://github.com/rpm2003rpm/HT6022_Driver
>
> It looks like that guy did more or less the same kind of reverse engineering as rpcope and hoenicke. Didn't mod the firmware though (Is it even possible that they didn't notice this project ?)

Thanks for the link, I didn't find this project when I searched for Linux support a month ago.

**Quote**

> It works straight out of the box on linux ($ make && ./a.out), and it's written in C. (Which is great for me because I was getting headaches learning python's dynamic typing )
> It uses the stock firmware, but it should be as simple as a copy/paste to make it upload the modded one instead.

Accessing the scope is really not that difficult, once you understand libusb1. Python makes it easier to prototype something, but I see no reason why one shouldn't be able to port this to C.

The procedure to setup the scope is as follows:

1. Find the USB device (04b4:6022 or 04b5:6022 when firmware present)
2. If firmware is not present, send it via 0xa0 control messages. Then restart at 1.
3. Set number of channels (new feature of my firmware), sampling frequency, scale for ch1 and ch2 via 0xe0-0xe5 control messages.

4. Start sampling (0xe4 control message with data 1)
5. Read samples via bulk or isochronous (new feature) transfer.
6. Optional for new firmware: stop sampling (0xe4 message with data 0).

If you want to continuously sample over a long time, you should use the asynchronous libusb1 API.   This is probably the most difficult technical part.

**Quote**

> It's lacking some comments, and a few variables names are a bit mysterious, but it works, (without being root).
> If we implement a rolling buffer, and have the gui in another thread, would we still need isochronious usb transfer ?

Isochronous usb gives gapless sampling for up to 24 MB/s  (24 MHz one channel or 12 MHz two channels).  With bulk transfers there is no guarantee.  You get a mostly gapless sample if you sample at up to 30 MHz into a single block, but every extra message sent over the USB port or every delay in the usb driver of more than 40 µs may corrupt your samples.

Features like triggering, calibration, etc. have to be done in software at a higher level.  The driver will just read the raw samples continuously from the device.

Reply Quote Notify

## 2 Test Equipment / Re: Hantek 6022BE 20MHz USB DSO

« **on:** May 02, 2015, 03:07:22 pm »

**Quote from: 2010kira2010 on April 28, 2015, 01:13:37 pm**

> **rpcope1**
>
> I damaged EEPROM.
> In the EEPROM stored vid/pid device.
> 8byte
>
> Presumably: C0 B4 04 22 60 00 00 00

Yes, exactly:

**Code:** [Select]

```
>python example_linux_readeeprom.py
c0b4042260000000...
```

Reply Quote Notify

Pages: [**1**]

EEVblog Main Site EEVblog on Youtube EEVblog on Twitter EEVblog on Facebook EEVblog on Odysee