

ADS1299 sample rate / threading

rick

June 2014 edited April 2015 in Hardware

Hi,
can s.o. tell me where and how the sample rate is set in the arduino code? or what is the limiting factor?
i wrote a small c programm to log the raw data without the processing stuff and i measure a fluctuating sample rate of about 243Hz.

I was thinking, the limiting factor is the throughput of the Arduino, but the sample rate did not improve, when i fastened the code in the loop() function.

Comments



chipaudette
June 2014

The sample rate is determined by the clock built into the ADS1299 chip that is on the OpenBCI board. It has its own clock. The user configures the clock (by setting a divider for the clock), but then the chip then goes and triggers its samples on its own.

To set the divider, see the initialize() method of the ADS1299.cpp file in our Arduino library:
<https://github.com/OpenBCI/OpenBCI/blob/master/Arduino/Libraries/ADS1299/ADS1299.cpp>

Howdy, Stranger!

It looks like you're new here. If you want to get involved, click one of these buttons!

Sign In with OpenBCI Register

Sign In Register

New Discussion

Categories

Recent Discussions

Activity

Categories

All Categories	737
Hardware	231
Ganglion	42
Software	242
OpenBCI_GUI	88

After the ADS1299 has taken a sample, the ADS1299 pulls the DRDY pin HIGH (or LOW, I can't remember). The software on the Arduino looks for this pin to switch to that state, at which point the Arduino code asks for the sample values from the ADS1299 and pushes them data out over the serial link to the PC.

This "look for the pin to switch" can either be done via an interrupt routine on the Arduino, or it can be done every time through the Arduino's loop() function. I use the loop() function, as can be see in my "ADSManager.isDataAvailable()" call:

<https://github.com/OpenBCI/OpenBCI/blob/master/Arduino/Sketches/StreamRawData/StreamRawData.ino>

It then gets the data via the line "ADSManager.updateChannelData();". This method is in the ADSManager.cpp library:

<https://github.com/OpenBCI/OpenBCI/blob/master/Arduino/Libraries/ADS1299/ADS1299Manager.cpp>

Note that, even at the slowest sample rate (250 Hz), it's a lot of data for the Arduino serial link...so you really have to make sure that you set your baud rate high enough.

Hope this helps...

Chip

rick

June 2014

Hi Chip,

Headware	43
Research	35
Electrodes	39
Other Platforms	38
Opportunities	40
General Discussion	69

Google Advanced Search

Search



thanks for your answer.

I've set

`#define SCK_MHZ (8)`

in `ADS1299Manager.h` to get a higher SPI speed.

I've also set the sampling rate of the ADS1299 to 500 SPS with this:

```
ADS1299::WREG(CONFIG1,0b10010101); delay(1);
```

in the `initialize()` function.

But I still get only about 243 SPS.

For testing, i changed the Arduino's `loop()` to these 2 lines:

```
sampleCounter++;
```

```
ADSManager.writeChannelDataAsBinary(MAX_N_CHANNELS,sampleCounter);
```

and called

```
startRunning(OUTPUT_BINARY_OPENEEG);
```

in `setup()`

So it sends the same (nulled ?) samples as fast as it can.

But still the same incoming SPS on my computer...

Changing the serial baud rate isn't helping, too (i can decrease the sampling rate to 150 by halving the baud rate).

Did you ever try a higher sampling rate than 250 with the Arduino Uno?

Thanks for any help with this!

rick

June 2014

Got it working!

The fault was at my program, it needed some threading to get all of the data.

ayushgaud

April 2015

Hi

I am facing the same problem with sampling rate. Can you please tell me how you got it working.

Thanks

mrollin USA

July 2015

Echo that; I set sampling rate to 500 sps but am topping out at about 295 sps. [@rick](#), can you describe your solution in any more detail?



wjcroft

July 2015

See these related threads.

<http://www.openbci.com/forum/index.php?p=/discussion/248/constraints-on-data-rates-thru-dongle>

<http://www.openbci.com/forum/index.php?p=/discussion/342/adding-wifi-faster-bt-and-other-connections-for-v4>

<http://www.openbci.com/forum/index.php?p=/discussion/94/using-dpcm-to-save-bandwidth>

<http://www.openbci.com/forum/index.php?p=/discussion/313/actual-sampling-rate-around-250-5-hz>



biomurph Brooklyn, NY

July 2015

@rick I'm also interested in how you got it working.

The bottle neck in data throughput is between the PIC (or ATmega) and the RFduino on the board. For reasons having to do with the way RFduino handles radio interrupts vs serial interrupts, 115200 is the max baud that is stable. That combined with the maximum radio packet size.

The uC wants to send 31 bytes per sample

- 1 byte sample counter
- 8-3 byte ADS data
- 3-2 byte accelerometer data

The serial protocol adds start and stop bits to each byte, for a total of 310 bits that have to be sent.

115200 baud has bit time of 8.68uS per bit, so it takes 2.69mS to send all that data.

If you drop the accelerometer data, you still need 2.17mS to get it all out the door.

How are you getting 500sps throughput?

July 2015

@biomurph Thanks for the super-clear explanation. This makes sense.

Do you know what happens to the sampling behavior when the transfer rate is lower than the specified sample rate? As in, does it skip samples, or wait to sample, or queue samples? The data I'm getting looks a little wonky and I'm wondering if there's some kind of aliasing or something going on.



biomurph Brooklyn, NY

July 2015

Not sure if I understand what you're talking about. Can you give me specifics for your 'transfer rate is lower than the specified sample rate'?

If you are transferring the data at a lower rate than the sample rate, then you will be missing samples. The system should send the latest sample data.

Mike

July 2015 edited July 2015

Hi,

I'm experiencing some really strange behavior with OpenBCI sample rate.

Does it make sense that the sample rate is over 250Hz ? I'm using the default firmware that comes with OpenBCI 8bit board.

Any feedback will be appreciated.

Thanks,
Mike

Mike

July 2015 edited July 2015

One more thing I have just noticed that after half an hour or so I'm starting to get many missing packets - the counter is getting out of order..

I had a suspicion that maybe it has to do with battery power - so I measured the battery but it seems to provide around ~ 5.18V. However, when I measured the OpenBCI 5V pin I get ~ 4.29V - is it normal? Shouldn't I get 5V as specified on the pin? And could there be a situation where low power results in missing packets?

Thanks,
Mike



wjcroft

July 2015 edited July 2015

@Mike , hi. I merged your question into this existing thread.

See the July 7 post above, in particular,

<http://www.openbci.com/forum/index.php?p=/discussion/313/actual-sampling-rate-around-250-5-hz>

Regarding the battery situation, can you try your experiment with fresh batteries; it does sound like you are close to the depletion point. The 5V regulator might require a bit of headroom, Joel @biomurph may comment.

William

Mike

Hello William, I have read the post that you linked. I couldn't see any solution there. So, I was wondering if someone has figured out some way to overcome this issue. It is making much harder to do an accurate signal processing this way.

About the battery, I'm using a adafruit's booster as was once suggested here on the forum with LiPo battery(3.7V, 500mAh).

Thanks,
Mike



wjcroft

July 2015 edited July 2015

Some 'drift' in sample rate is common in A/D conversion,

<https://www.google.com/search?q=eeg+sample+rate+drift>

<http://www.dtic.mil/get-tr-doc/pdf?AD=ADA561715>

Can you say what software you're using? For example, OpenViBE in the Acquisition Server, has the ability to do "drift correction". By dropping or adding samples. I've not looked at the code but this might use some form of interpolation. I don't believe it is doing a complete formal "resampling" (over all samples) by using using spline interpolations, that's slow.

<http://openvibe.inria.fr/acquisition-server/>

The ADS1299 can also be configured with it's own external clock oscillator; not sure how easy that is, would require some hardware modding. Some other boards we know that use ADS1299 have an external clock. That would give you more accuracy.

re: erratic behavior on long sessions. What you're seeing may correspond to some other reports of long session anomalies,

<http://openbci.com/forum/index.php?p=/discussion/324/data-stream-freezes-during-long-sessions>

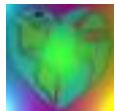
I believe your battery setup is fine there, I use the same Adafruit Powerboost with LiPo cell. It's designed to put out 5.2 V.

Mike

August 2015 edited August 2015

One more thing - how would you differentiate between a packet that holds accelerometer data and one that doesn't? By default accelerometer data should be received at 25Hz(which also keeps changing...) but if the data rate of the EEG changes repeatedly it is almost impossible to compute it accurately? Right now, I'm checking if X,Y,Z are all zeros, if so I ignore it but this is obviously wrong.

It would be nice to hear what the original designer had in mind.



wjcroft

August 2015

@Mike , let's mention Joel @biomurph , and ask for his tips on how the accelerometer values are interpolated into the stream. Your current heuristic of checking for all zeros sounds just fine to me. Did you see the dtic.mil link above to their drift correction paper? LabStreamingLayer also has a drift mechanism.

William

Mike

August 2015

Well, I'm still having issues with the OpenBCI device. Not very often the device suddenly start to transmit packets with sample counter = 0.

Is this known issue? Has anyone reported on it?



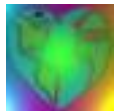
biomurph Brooklyn, NY

August 2015

Hi @Mike,

The accelerometer values are only added to the data stream when they are updated by the accelerometer. When there is no fresh accelerometer values, the XYZ is set to zeros. This is to allow for user auxiliary data to be inserted into the data stream.

You are using the 8bit Board, and our 5V regulator has an LDO, but I'm not surprised if you are seeing that kind of voltage drop if you're using AA batteries in the holder we provide. It is likely that the drop outs you are seeing are due to low battery power.



wjcroft

August 2015

Joel, I think he's using the Adafruit Powerboost 500C, which is designed to put out a constant 5.2V regardless of charge level of LiPo cell. It would be a good test for Mike to compare that with operation using the standard AA cell holder and higher input voltage. I'm using the 500C here with no issues.



biomurph Brooklyn, NY

August 2015

Then I'm not sure what the issue is. I don't have a Powerboost here, perhaps I'll get one and try it.



wjcroft

August 2015

Or just use a variable voltage power supply and slowly come down from 6V to 5V, to see how low the LDO regulator can deal with it. Does seem like 5.2V should be enough, if it is indeed LDO. Lady Ada set that 5.2 output assuming that it would be plenty of margin for 5V regulators.

Mike

August 2015

Hi, Yes, as William said, I'm using Adafruit Powerboost 500C.
Would love to hear any update on this issue.

Thanks,
Mike

JakeStew

August 2015

> Does seem like 5.2V should be enough, if it is indeed LDO.

You never know. Calling something a "LDO" depends on a number of factors other than just the voltage dropout vs. current.

A LDO can have dropout voltages anywhere from below 0.2v to over a volt. Most linear regulators

with dropout voltages under 0.2v call themselves "ultra-low dropout".

I doubt this is the problem though, but it would be the easiest thing to check. Even if it's getting 5.0v and the LDO has a 0.3v dropout... you should still be getting 4.7v, which should be plenty to run the board.

Do we have specs on the minimum board voltage required?



biomurph Brooklyn, NY

August 2015

The only thing running at 5V on the board is the ATmega328P.

Page 316 of the DS below shows safe operating area given clock and voltage. We do appear to be in range.

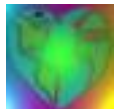
<http://www.atmel.com/Images/doc8161.pdf>

Mike

August 2015

Any of you tried to run OpenBCI 32bit version directly through a LiPO battery? Any issues with that?

Thanks,
Mike



wjcroft

August 2015

@Mike , regarding running direct from LiPo cell, see this previous comment.

Basically a single LiPo cell at 3.7v would not give you much margin over 3.3v. But when the revised board is available with new regulator, then with two series connected cells (up to 8.4v) should be ok.

Mike

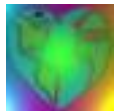
August 2015

Thanks William.

I'm not sure why a single LiPO at nominal 3.7V 500mAh shouldn't be sufficient for running the 32bit board. Did you try to run it? How long did it last?

Your comment about the revised board sounds interesting. Any idea when are the revised board is going to be available? Is there a draft that shows changes in the revised board?

Thanks,
Mike



wjcroft

August 2015

Mike, no I have not tried the single LiPo cell. I'm not sure how much of the 500 mAH capacity you would get. It would depend on the discharge characteristics of the cell (discharge curve). After it falls below 3.3v, you'd be taking your chances.

Here's one such curve,

Leave a Comment

B

I

abc

T

tT

H1

T