

ISO15693 Sniffing #312

New issue

[Open](#)

cacke-r opened this issue Jan 5, 2022 · 32 comments



cacke-r

commented
Jan 5, 2022

Contributor

Hello all,
I'm wondering if there is any update on the ISO15693 sniffing.
I'm currently working based on the iso15_sniff branch from the @ceres-c fork.
But it doesn't support card->reader sniff yet. I'm willing to add it - and currently digging into the chameleon and Atmel details I'm not yet familiar with.
I'm wondering if there is already some further progress - that I've not yet found ?

Is there any reason, why the actual status is not yet merged to this repo here ?

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

3 participants



Currently I'm fighting with activating the overflow-IRQ on the TCD0 channel.

Whenever I do so - it the device seems to be dead (no further log output - no reaction on the command interface) - is there any trick with the OVF-IRQ which needs to be considered ?

Thanks in advance for any support, cacke-r



fptrs

commented

Collaborator

Jan 6, 2022

Hi **@cacke-r**,
@ceres-c did not submit a PR regarding ISO15693 sniffing and I'm not sure what's the status of his branch. But there is also no further progress that you've missed, so this might be a good starting point.

Does the red LED light up when the device seems dead? If so, then there is a problem with the ISR and the Chameleon does not recognize your ISR as the TCD0 OVF ISR and triggers BADISR_vect. If you can share the code I could take a look.



cacke-r

commented

Contributor

Author

Jan 6, 2022

Hi @fptrs ,
thanks for your quick reply.
OK, then I'll go ahead from the
@ceres-c branch - and do
what I can.

Thanks for the hint about
BADISR_vect - I missed that I
need to add it in SharedISR.S
:-(.... just an Atmel newbi - so
that part works now.

Then I'll go ahead and get the
sniffing for card working
(reader sniffing already works)
- afterwards we may discuss
how to get the whole package
upstream ? Shall I start with
raising PRs against @ceres-c
repo first ? Or rather directly
here ?

(don't hurry with answering - I
would need some time anyhow
:-))

Regards
cacke-r



ceres-c

commented

Contributor

Jan 7, 2022 •

edited ▼

Hi,
I just returned from a call with @MrMoDDoM about this, since we knew we were working on that branch, but due to the pandemic stopped all development and completely forgot about the current state of the project and the approach we were considering to implement VICC->VCD sniffing. Given that we haven't touched the codebase in 2 years now, we are a bit out of the loop, but we believe we might have figured out a general direction to follow.

(From now on I'm taking for granted that you have knowledge about iso15 air interface, which is defined in ISO15693-2.)

Our plan was to implement both high and low data rate with single and double subcarrier, thus we are looking for a generic solution. We don't know yet if this is feasible and we're assuming to reliably sample the pulses in VICC->VCD communication (strong assumption, one problem at a time, though).

~~We were thinking about using 2 interrupts: one counter and one timer.~~

- **counter:** attached to the input port, will count all the falling edges
- **timer:** invoke an ISR every $f_c / \gcd(28, 32) = f_c / 4$ and check the value of the counter. Depending on this value, we can determine if we're receiving 0, 8 or 9 pulses. This gives us the value of one bit.

We then need, every 8 bits, to save somewhere the value of a whole byte. This could either be handled in the aforementioned timer ISR, or via a third timer, but we have no idea yet which is preferable. That will come as we code, I guess.

This approach should cover all our needs, since we know a priori whether we need to demodulate high or low data rate and single or double subcarrier. The only change we might need would be to recalculate the timers when setting up the codec if using low data rate (multiply by 4, from the standard).

Edit: No, the above wouldn't work. Did I tell you we are out of the loop? I can confirm. It took us way too much to spot the issue here: $f_c/4$ is a too fine measurement and it would be impossible to differentiate the two frequencies; there would be only one falling edge per $f_c/4$ period with both $f_c/28$ and $f_c/32$.

We also recalled Xmega offers a frequency measurement feature via its event system, 8045A-AVR-02/08 § 6.3 suggests an algorithm to do so. We could then set up a:

- **counter**: attached to the input port, will count all the falling edges
- **timer**: every $18,58 \mu s$ (not $18,88$), check the frequency of the data we've just received. Every 2 calls, check whether we got high->low or low->high frequencies and generate the corresponding bit. Also, if this is the first "half" of a bit and we received data on a high frequency account for additional $0,3 \mu s$ in current timer.

Have you already laid out a plan to implement VICC-VCD sniffing? Would you like to expose it here so we can exchange ideas? We'd love some criticism on our idea, so we can figure out any pitfall before we start to write code.

Given that we currently have some spare time, once we figure out a solid plan, we could start working together on this project. Would you mind firing me or **@MrMoDDoM** an email so we can maybe use a faster mean of communication?

PS My `iso15_sniff` hasn't been merged with the last commits in `master`, so give me some time to update it.



cacke-r

commented

Contributor

Author

Jan 7, 2022

Hi @ceres-c , @MrMoDDoM ,

great to hear that you are also still willing to continue the paused activity.

To your question, yes I'm aware of the Air interface (ISO15693-2) - and I also already tried to understand your ideas from the comments in the code.

For further discussions I'd raise an e-mail to both of you. My plan as of now was to understand your ideas and get it working :-)

One big question I still have, which I was not able to figure out now:

The ISO15693-2 shows a nice digital signal for the VICC->VCD communication. But in fact

it is just the envelop of the modulated carrier. My question is - what signal do we see on the AVR port ?

(is the carrier already filtered out by the demod HW) - or do we need to analyze the modulated carrier ?



ceres-c

commented

Contributor

Jan 7, 2022 •

Contributor

edited ▼

You just reminded me that (luckily) we laid out our ideas from back then as comments in the sniff codec. Honestly, I did not remember this fact at all: turns out we almost reinvented the wheel today, since the ideas we tried to come up with were almost the same we had 2 years ago, and probably we fell into the same pitfall :)

I'm not sure **@MrMoDDoM** has a public email somewhere so, once you write me I'll add him to the conversation. **@fptrs** I guess you're busy, so we could either keep you posted writing relevant updates here, or add you to the conversation as well.

Yes, the MCU already sees demodulated output, but I'm pretty ignorant on the hardware details. I'll hook up my scope to the port one of these days

(Unrelated: I still have to buy a tonie and fix my branch for that)

--edit-- Just impulse bought a tonie. I guess that will be a good addition to the "stuff I hacked" box



cacke-r

commented

Contributor

Author

Jan 7, 2022

Hi all,
pls find my latest changes to
make the iso15xxx sniffing
work for me (VCD->VICC only)
here:

https://github.com/cacke-r/ChameleonMini/tree/iso15_sniff

I tried to create a PR on
@ceres-c but failed tonight :-)
will try again in the next days.

Regards
cacke-r



fptrs

commented

Collaborator

Jan 27, 2022

Hi guys,
sorry for the late reply.
@ceres-c did you already take
some picture with the scope? I
would like to join your
conversation so that I can
throw in a few ideas.



ceres-c

commented

Contributor

Jan 27, 2022

Yup, I've taken a lot of measurements in the last three weeks, and now have a fully working sniffer in my repo :)

I have tested my code with both my demo board (STM241r-discovery) and different phones with success. Moving the reader or the tag in the field did not yield any issue, as long as the chameleon was between the two. I had to abandon dual subcarrier due to issues with SOF identification: I could not identify a threshold to reliably count pulses, so it was impossible to know when the first 27 pulses ended and the 24 pulses started. Pulses counting proved to be unreliable when different tags were used: the slight differences in the shape of subcarrier pulses generated by different products would result in erroneous counts.

@cacke-r was having some issues with my code on a toniebox, so we're trying to debug it at this point.

BTW one Chameleon died in the process, I hope Peta won't come for me.



fptrs

commented

Jan 27, 2022 •

Collaborator

edited ▼

@ceres-c Looks really good. I just tested it with my phone. If I understand correctly you use the ADC to determine a suitable threshold. It might be also a good idea to take a look at the difference between DEMOD-READER and DEMOD during transmission. I suggest taking scope pictures including both signals as well as their difference, for the dual subcarrier case and also for a response from a tonie. If you can share the scope pics I am happy to take a look.



ceres-c

commented

Contributor

Jan 27, 2022

Yes, I am using the ADC to both find a first suitable threshold and then increase the threshold once the chameleon starts to see pulses.

I've seen the two signals on a scope and I am indeed using both of them on purpose: first threshold is calculated from DEMOD, further on DEMOD-READER is used because, albeit different, it's easier to sample since it's not returning to zero and will then more likely yield useful data.

I don't have a toniebox to test, so I'm probably calling it a day right now. I'm going to play the "works on my machine" card and leave the fine tuning of the threshold to someone else. Getting to this point took already too many hours.



fptrs

commented

Collaborator

Jan 28, 2022

My idea was to use the DEMOD-READER and DEMOD signal both as input to the AC, most of the time the signals are quite equal but since the DEMOD-READER signal passes an additional low pass filter it is slower and we might not need the DAC for a threshold. Anyway nice work @ceres-c and hopefully we get the tonies to work again.



ceres-c

commented

Contributor

Jan 28, 2022

I'll be damned, this is smart:
I'm going to try it right now.
Wait to merge the PR



ceres-c

commented

Contributor

Jan 28, 2022 •

edited ▼

Ok, it is indeed much more reliable when the signal is in a sensible range, but I'm not sure it'd help with smaller signals since it's still going to miss the first few pulses. Also, with a specific card (EM4233), the count is completely out of place: with high antenna coupling, the reading could vary from 21 to 40 pulses on first pause. With these results, it would be impossible to implement dual subcarrier anyway

I could indeed do without the DAC threshold, though, and retain current "heuristic" to decode single subcarrier, albeit simply changing the ADC source results in wrong decoding, so I might have to fix something else as well

--edit--

I've been playing around with this and due to the shape of signals, after a pause, when a signal is faint, there would be too few crossings of the two signals, thus it's even less reliable. At least with my setup and an EM4233 tag. It might be an unlucky specimen, but still with previous code it did work (without physically moving anything in the setup)



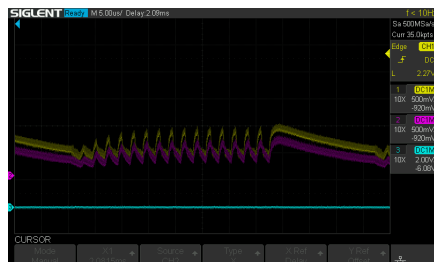
ceres-c

commented

Contributor

Jan 28, 2022

This is the last modulated bit-half of the SOF, as you can see `READER-DEMOD` is indeed not crossing `DEMOD`, so... I don't know.



ceres-c

commented

Contributor

Jan 28, 2022

@cacke-r do you mind testing [this branch](#) with the toniebox? If it fixes your issues, we can work with @fptrs idea



cacke-r

commented

Contributor

Author

Jan 28, 2022

@ceres-c , @fptrs

If tested the following versions:

1. your branch as is
2. the relevant changes (hope I caught them all) on my version (incl my

debug info)

3. Version 2) with an additional - to set ACA INTMODE to FALLING in `isr_SNIFF_ISO15693_ACA_ACO_VECT` as well (you changed it only in `CardSniffInit`)

This is not working for. In non of the cases, even the first IRQ `isr_SNIFF_ISO15693_CODECTIMER_TIMESTAMPS_CCA_VECT` doesn't be called.

(Basically the same symptom as with the DACB based way)

For me - the approach using the DACB is somehow better - because I can fix it by setting a constant value.

Means sniffing works very well with the Tonie box - the only drawback we currently have is that the automatic threshold doesn't work.

So, I'd vote to go ahead merging the DACB based solution. And if we don't find another smart way for the Tonie box - I would still see the following as good path to go.

- Having the automatic threshold detection in the Codec as 'default'
- offering an API to the Application layer to deactivate auto-threshold

- and implement 'autocalib' command (comparable to the ISO14443)

.. this would help for the TonieBox ... and for any other corner cases we are not aware of now.

What do you think about this ?
If you like it - I could go ahead implementing that approach.



ceres-c

commented

Contributor

Jan 30, 2022

This sounds good to me. I'll follow this approach with an additional configurable fixed threshold. Maybe we can make the codec default to automatic threshold detection and add a forced threshold if configured in application



cacke-r

commented

Contributor

Author

Jan 30, 2022

Okay, that's sounds good - then I'll take this approach forward once your PR is merged.



fptrs

commented

Collaborator

Jan 31, 2022

For me - the approach using the DACB is somehow better - because I can fix it by setting a constant value. Means sniffing works very well with the Tonie box - the only drawback we currently have is that the automatic threshold doesn't work.

@cacke-r can you share a picture of your setup with the tonie box and scope pics as well. I was not able to get a good scope pic for a tonie, only thing I catch with the scope is the box itself, the figurines response however does not show up, for me there's only noise on DEMOD and DEMOD-READER pin

@ceres-c after the final changes and resolving the conflicts with the master branch I'm going to merge your PR. I think we can keep this issue open for the improvements of the threshold detection

cacke-r



commented

Contributor

Author

Jan 31, 2022

@fptrs please find here a photo of my Setup. But it is quite unspectacular 😊 it is all boxed.

Getting scope pics is rather complicated- I already told @ceres-c . I need to borrow one from work, and see how I can fix it to measure. It is still on my todo list - but I need some bigger calm timeslot- which is quite rare currently



Will share it once i have it.

But according to the adc values I recorded - I can imagine well that the signal is very noisy.

Still the capture works very reliably - once I set a proper dac value.

I'll now go ahead with the Autocalibration, for my use case.



cacke-r

commented Contributor Author

Feb 3, 2022

@ceres-c , @fptrs

I'm now done with the autocalibration command. Since it is my first contribution here, I'd be great if you could have a first look at my branch before I raise a PR - just to check if I'm missing something fundamental for this repo (e.g. coding styles, degree of documentation, etc).

What is still on my todo before finally raising a PR is:

- check for a branchfree

saturation for setting
DACB.CH0DATA

- change default behaviour
to 'auto-threshold'

@ceres-c It would be great, if
you could test with your cards,
whether the auto-calibration
works for you as well
(even if you don't need it :-))

ah, and here is my branch
[https://github.com/cacker-
r/ChameleonMini/commits/iso1
5_sniff](https://github.com/cacker/ChameleonMini/commits/iso15_sniff)
(relevant are the last 6
commits)

Thanks for your feedback.



ceres-c

commented
Feb 4, 2022

Contributor

Will check the code and test
tomorrow :)



cacke-r

commented

Contributor

Author

Feb 4, 2022

Cool, thanks.

Few things I forgot to mention:

- command autocalibration assumes a continuous communication b/w reader and card
- I needed to increase the timeout for commands to 150 (15s). Since the tonie box is sending cyclic request too slowly.
- I've left in debug traces. In case it fails for you. Would be great to get those.



cacke-r

commented

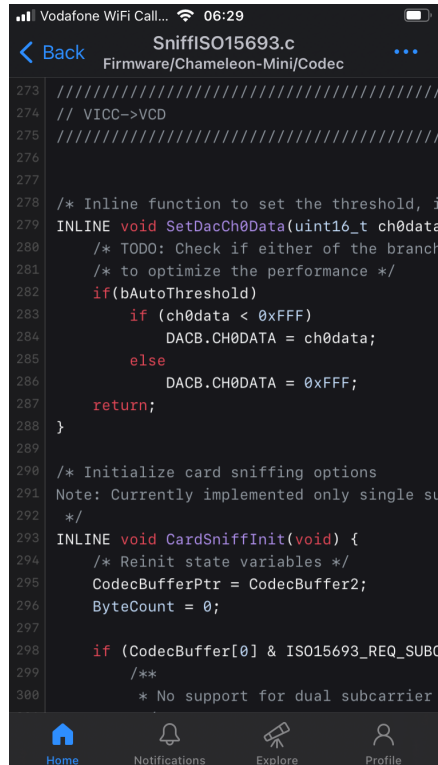
Contributor

Author

Feb 4, 2022

@ceres-c

I just realized that there is a bug in my latest push. Missing {} in setdacdata (see picture).



```
273 //////////////////////////////////////////////////
274 // VICC->VCD
275 //////////////////////////////////////////////////
276
277
278 /* Inline function to set the threshold, i
279 INLINE void SetDacCh0Data(uint16_t ch0data
280 /* TODO: Check if either of the branch
281 /* to optimize the performance */
282 if(bAutoThreshold)
283     if (ch0data < 0xFFF)
284         DACB.CH0DATA = ch0data;
285     else
286         DACB.CH0DATA = 0xFFF;
287     return;
288 }
289
290 /* Initialize card sniffing options
291 Note: Currently implemented only single su
292 */
293 INLINE void CardSniffInit(void) {
294     /* Reinit state variables */
295     CodecBufferPtr = CodecBuffer2;
296     ByteCount = 0;
297
298     if (CodecBuffer[0] & ISO15693_REQ_SUBC
299     /**
300     * No support for dual subcarrier
```



ceres-c

commented

Contributor

Feb 5, 2022

I have been testing it tonight and the threshold calculation seems to be working. I have not taken scope pictures to compare performance of the two systems with a given signal amplitude and I'll do it tomorrow in the morning, but this looks good to me.

I have a couple of remarks,
code wise.

We might need a way to
enable/disable the manual
threshold calculation system.

A new command would be fit,
but I fear it'd be annoying for
you to implement.

[https://github.com/cacker-
r/ChameleonMini/blob/4bb664
08009f11bc83c3289a53fad5c
717da1f96/Firmware/Chameleo
n-
Mini/Application/Sniff15693.c#
L136](https://github.com/cacker/ChameleonMini/blob/4bb66408009f11bc83c3289a53fad5c717da1f96/Firmware/ChameleonMini/Application/Sniff15693.c#L136)

I'd go for a

`CodecThresholdSet` with a
new value here, instead of a
for loop repeating 3
increments

<https://github.com/cacke-r/ChameleonMini/blob/4bb66408009f11bc83c3289a53fad5c717da1f96/Firmware/ChameleonMini/Application/Sniff15693.c#L179>

Are we sure `min_succ_th` can be considered 0 when running autocalibration twice? I guess it is reset on AppReset, but if the chameleon is not removed from the field then this value will not be reset. It'd be better to reset `min / max` and `autocalib_state` once a correct threshold is found (i.e. in `SniffISO15693FinishAutocalib`)

Also, I see you still have the `DemodByteCount` variable, so maybe you're missing last commit from my branch

See you tomorrow with some more empirical evidence :)



cacke-r

commented Contributor Author
Feb 5, 2022

Great, thanks for the feedback.
I'll work on that.



ceres-c

commented Contributor

Feb 5, 2022 •

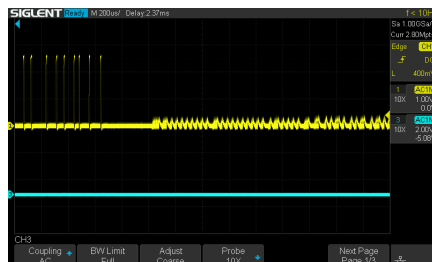
edited ▼

I made some tests: all these numbers are related to 20 inventories per run.

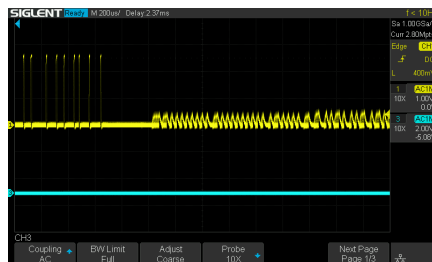
Test procedure: turn off chameleon, do a couple of 20 inventories run without issuing "autocalibrate" command.

Issue autocalibrate (while asking an indefinite number of inventories), ask for 20 inventories, autocalibrate again, 20 inventories, autocalibrate, 20 inventories. Can you confirm this procedure is correct?

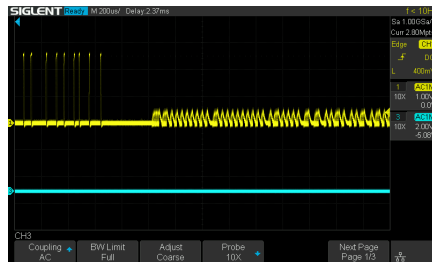
With this signal amplitude I got 0 frames with both the new autocalibrate and with old code



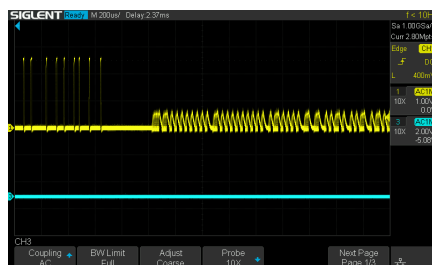
- new autocalibrate: 0 card frames
- old code: 0 card frames



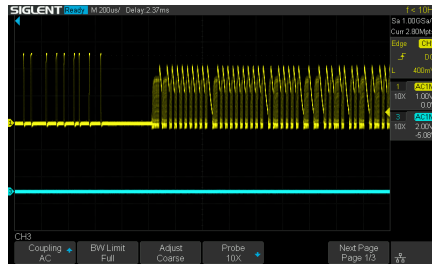
- new autocalibrate:
sometime 1 card frame,
sometime 1 broken frame,
mostly not working
- old code: sometime 1 card
frame, sometime 1 broken
frame, mostly not working



- new autocalibrate:
between 6 and 15 card
frames, possibly
depending on the
calibration result
- old code: between 2 and 5
card frames



- new autocalibrate: ~18
card frames, sometime
broken frames
- old code: ~16 card frames,
sometime broken frames,
sometimes nothing



Working with both setups

I'd say the autocalibration is relatively better than the original code, but still limited



cacke-r

commented

Contributor

Author

Feb 5, 2022

@ceres-c thanks a lot for these intensive tests. So, I'd conclude out of the results, that using a static calibration is (at least) not worse. (looks slightly better - but still in the range of stochastic noise). I'd then go ahead with fixing your findings. But this may take some days - since I'm quite occupied at the weekend.



cacke-r

commented Contributor Author

Feb 5, 2022

BTW - the test procedure looks good to me. The only point, that I needed to check after a power cycle of the device is - if the new threshold was really stored. I had some tries where this was not the case -
so i always double checked it after boot (chemtool -th).



cacke-r

commented Contributor Author

Feb 7, 2022

@ceres-c
Starting to work on your remarks now. Please find some comments on these here.

I have a couple of remarks, code wise. We might need a way to enable/disable the manual threshold calculation system. A new command would be fit, but I fear it'd be annoying for you to implement.

No, thats not annoying - i like this command implementation :-) But as a first step - to keep the PR smaller - i though I'd go with a pre-processor directive ? (ala #ifdef AUTO_THRESHOLD_OFF and set it in the Makefile (commented by default)). I need to change the active configuration anyhow for my local build. What do you think about this as a first step ?

<https://github.com/cacke-r/ChameleonMini/blob/4bb66408009f11bc83c3289a53fad5c717da1f96/Firmware/Chameleon-Mini/Application/Sniff15693.c#L136> I'd go for a `CodecThresholdSet` with a new value here, instead of a for loop repeating 3 increments

Agree - will fix this.

<https://github.com/cacker/ChameleonMini/blob/4bb66408009f11bc83c3289a53fad5c717da1f96/Firmware/ChameleonMini/Application/Sniff15693.c#L179> Are we sure

`min_succ_th` can be considered 0 when running autocalibration twice? I guess it is reset on AppReset, but if the chameleon is not removed from the field then this value will not be reset. It'd be better to reset `min / max` and `autocalib_state` once a correct threshold is found (i.e. in `SniffISO15693FinishAutocalib`)

Actually it is not reset in AppReset - but anyhow AppReset is called in the Command function for the Autocalib command

https://github.com/cacker/ChameleonMini/blob/iso15_sniff/Firmware/Chameleon-Mini/Terminal/Commands.c#L662

But min_succ_th is reset in the SniffISO15693InitAutocalib

https://github.com/cacker/ChameleonMini/blob/iso15_sniff/Firmware/Chameleon-Mini/Application/Sniff15693.c#L104

which is called during the start of AutoCalib process.

So - this should be fine. Or am I missing something ?

Also, I see you still have the `DemodByteCount` variable, so maybe you're missing last commit from my branch

Yes - I'll rebase against emsec/master before raising the PR.



cacke-r

commented

Contributor

Author

Feb 7, 2022

@ceres-c @fptrs

I'm now done with rebasing and cleanup. Pushed it on the master branch of my fork <https://github.com/cacke-r/ChameleonMini/commits/master>

Will do the testing tomorrow and then raise the PR.
Thanks for all your support so far :-)

Is there any best-practice here on "Whom shall I add as reviewer?"



fptrs

commented

Collaborator

Feb 8, 2022

You can add me and **@ceres-c**. I'm going to test and merge it once you've raised the PR. Nice work so far
👍