Plastic scanner review

18 Sep 2022

source code

The ADS1256 is set to a sample rate of 30kHz, gain=1. Things you can do in firmware to increase the signal/noise ratio:

- choose a lower sample rate (see table 4 in the ADS1256 datasheet, noise at f=100Hz is more than 10x lower than at f=30kHz) NB: pay attention to the settling time, table 13. This is the minimum time you should waiting between two successive LED 'colors'. So in practice this will be a trade-off between how fast you can scan and how accurate you want the results to be.
- set higher PGA gain (see also table 4). You can even implement 'auto ranging' in firmware: first do a measurement at gain=1 and if your measured signal is small enough, repeat the measurement at gain=2/4/etc where possible.
- Optionally, you can take several measurements and add them together or release other analysis for extra resolution (averaging) or robustness (outlier rejection).

- If the above things are not necessary, you can probably opt for a simpler=cheaper ADC in the future. • When making adjustments, be aware that you may need to fine-tune the delay(5) between collecting each sample. The following factors are the most important: ÿ the settling time of the analog Opamp filter (see filter calculation

response)

ÿ settling time of the ADC (table 13 in the datasheet) ÿ

response speed of the photodiode / LEDs (presumably not significant)

Hardware - schematic design

Power filter

As a rule of thumb, each chip (microcontroller, opamp, etc) needs at least 1 capacitor for each power pin (e.g. AVDD / DVDD for the ADC). This is missing from the ADC and opamp. The lack of these so-called 'decoupling capacitors' will cause more noise and can even cause your circuit to not work or be unreliable.

Which values to use can often be found in the datasheet (see eg Figure 25, p28 of the ADS1256 datasheet). If this is not specified you can use 100nF as a rule of thumb.

ADC

Consider the ADS1255, it's the same as the 1256, but without the 8 channels (which you don't use). Can save costs (fog available..).

5V vs 3.3V signals

Is it correct that you are using an Arduino Uno (or similar) with 5V I/O pins?
While the ADC digital inputs are 5V tolerant, the output signal (MISO) is 3.3V logic. It is safer (at least for the MISO line) to use a level shifter. Due to, among other things, tolerances on parts, this can cause problems in series production.

LED driver

5V vs 3.3V signals

Is it correct that you are using an Arduino Uno (or similar) with 5V I/O pins?
Then it may be that the I2C interface does not always work reliably. Consider powering the TLC59208 at 5V, then you don't need a level shifter to solve this.

Accuracy LEDs

LEDs give off an amount of light that depends on the current through the LED.
In this case this is:

$(3.3V - V_f)/R$

So keep in mind that the intensity of the light that your LEDs emit also depends on:

• (tolerance on the resistor is usually negligible compared to the LEDs) •
tolerance deviation of the 3.3V supply (relative to the ADC reference) • tolerance on
the V_f (forward voltage) of the LED. This is often significant and temperature dependent

It is worth considering feeding the LEDs from a separate power supply (separate LDO) and also using this same supply as a reference for your ADC. Suppose the 3.3V is actually 3.2V, then your ADC will scale with it and the error will be less.

A step further would be to feed the LEDs with a so-called constant current driver.
This allows you to eliminate the effect of varying forward voltage.

To test the extent to which this is necessary, you could set up a test in which you repeatedly measure the same material over a longer period of time to see how large the spread is.

Chip select pin

You indicated that the chip select does not work. I cannot directly give a reason for this in the design. If I remember correctly you soldered it directly to GND as a workaround?

A few things to try:

- measure the voltage on both sides of R12, this will indicate whether the Arduino
  whether the ADC is the cause
- Connect the CS to GND in series with the multimeter and measure how much current will flow. This gives an indication whether the line is actively being pulled high (> 1 mA) or whether it is a pullup (eg internal pullup in the ADC or arduino). • check if your Arduino is not broken: can you toggle pin 10 apart from the PCB? • is your Arduino pin in output mode (pinmode(10, OUTPUT))

Transimpedance amplifier

Design is very similar to this application note from TI: https://www.ti.com/lit/an / dlpa072/dlpa072.pdf Explains the design of a DPL-based spectrometer.

Look especially at chapter 3.5. Here you will find the relevant calculations for the transimpedance amplifier.

Amplifier calculations:

```
GBW = 5.5MHz #(based on Opamp specs)
C_s = approx 30e-12 # (c=6+6.5+13+5?=30pF, based on Opamp+photodiode specs)
R_f = 240e3 (240 kOhm) f =
sqrt(5.5Mhz/(2*pi*r_f*C_s)) = approx 350kHz C_min > 2pF
(formula 7 appnote)
```

The opamp used therefore has sufficient gain to operate up to 350kHz. The compensation capacitor (now 51pF) must be at least 2pF for a stable circuit.
A larger value results in lower bandwidth, but also less noise in your signal.

Amplifier gain

The current amplifier has a gain of :

```
gain = 2*R_f = 2*240kOhm = 480 mV / A
```

A small current flows in the photodiode, depending on how much light falls on it, it is converted into a voltage according to the above formula. The optimal situation is that the maximum voltage corresponds to the maximum of your ADC input (so you use the entire 24-bit range).
If not, you can choose a higher gain resistor (or a lower ADC reference voltage).

Filter response: feedback capacitor vs bandwidth

The (simplified) formula below gives an idea of the bandwidth when you use a higher capacitor value:

```
C_ff = 51pF
R_f = 240kOhm

f_co = 1 / (2*pi*r_f*C_ff) = about 13KHz. t_r = r_f *
C_ff = 12ÿs
```

The above formula is an approximation and is only correct as long as the result is significantly lower than the previously calculated amplifier bandwidth (<< 350KHz). The bandwidth is defined as the frequency at which the signal is attenuated by 3dB. As a rule of thumb you can assume that the settling time is about 6 time constants:

t_settle = 6 * t_r = about 0.073 milliseconds

This seems more than fast enough to me, you can safely go for a slightly higher value. So it is not true that this has to be exactly 51pF. You can safely take 47pF (more common value) or a higher value.

Second filter (antialiasing)

After the opamp a differential filter is placed with R=301, C=47nF, the bandwidth of which is:

f_co = 1/(2*pi*(2*301)*47e-9) = 5.6KHz

You can choose other values for R3=R4 and C4 if desired. Please note that:

- R3 and R4 are the same (preferably 1% tolerance), also applies to R1 and R2. • R3 and R4 are not too large (would recommend < 1K, has to do with voltage drop towards your ADC)
- C4 doesn't get so big that it physically becomes a big package (keep it 0805 or smaller).

This filter (especially C4 and certainly C5) should be placed right next to the ADC pins.

Simplification options

The current design is clearly set up with high performance as the goal. I would leave it that way at this stage. If you have collected a lot of data, you always get clearer how accurate the system actually needs to be, or whether you can get away with less fancy hardware.

To consider:

- simpler 'single ended' transimpedance amplifier (uses half of the op amps)
- cheaper ADC (with less extensive features, fewer channels, lower sample rate or maybe 16-bit)
- voltage reference: the absolute accuracy (in volts) is not so important, it is the relative accuracy in relation to how brightly the LEDs light up. So you may be able to use a simple LDO as a power supply for the LEDS and as a reference.

Layout

Below you will find some tips on how the layout could be improved. This will mainly make the product more stable and robust, for example EMC compliance (susceptibility to interference from telephones or vice versa disturbing other devices). In addition, it will have a positive effect on ADC performance.

Ground plane / return currents

The most important improvement you can make in the layout is the addition of a GND plane : try to keep the bottom layer uninterrupted as much as possible and only use short 'jumper' tracks if you have to. This mainly affects the noise around the ADC and the risk of EMC interference. I once wrote a blog about this. If you want to learn more about this I can visit the youtube channel of Robert Feranec recommend.

Decoupling capacitors

This was already mentioned in the schedule review. Place a 100nF or similar capacitor directly next to each power pin of each chip (see datasheets). This is because even PCB traces behave like inductive coils that do not pass high frequencies well. So always place capacitors close to the chip with the shortest possible connection between the VCC/GND pins and the capacitor.

This also applies to filters such as C6/C7/C8 capacitors. If there are several parallel so place the lowest value (C8) close to the pins.

Sensitive signals

Analog input

Analog signals, especially the photodiode, are very sensitive to influence by other signals: because these are such small currents, even a small interference signal is already significant. This already looks pretty good. I myself would route the tracks between D1 and U1 even closer together (as a differential pair) with ground around it (and a continuous GND plane). The filter between the opamp and the ADC can also be a little more compact. Place the C as close to the ADC input as possible.

ADC

You prefer not to have digital signals or other potential sources of interference around the ADC / op amp. Note the routing of the I2C lines and the crystal under the ADC.

Conclusion

In terms of functionality, there are not really things that are necessarily wrong, but there are some points with which you can make the design a lot more robust. In the long run, I would recommend an overhaul for this, especially if you are considering CE marking (EMC compliance).