# Introduction

In response to shortages during the Covid−19 pandemic, TU Delft is developing the POxiM transmissive pulse oximeter. A pulse oximeter is a medical instrument which doctors use to monitor the heart rate and blood oxygen saturation of patients during critical stages of and revalidation from Covid−19. POxiM is intended to be fully qualified for use leading up to and after intensive care, thereby freeing existing more stringently qualified pulse oximeters to be used in intensive care. The main design goal is supplyability: 1000s of units should be locally manufacturable even during severe lockdowns. Simplicity and supply chain diversity are important secondary goals in service of this main goal.

POxiM consists of a disposable finger probe and a reusable wrist computer, containining nearly all of the circuitry. This is it technically only the schematic of the wrist computer, it shows all of the circuitry and can be regarded as the main electrical schematic of POxiM. It is also intended to give an overview of the operation of POxiM, motivate detailed design decisions, and to document the derivation of electronic component parameters. This schematic does not aim to justify the system−level and electronic high−level design decisions. Instead, those are motivated in the POxiM report and accompanying system engineering notes respectively.

Apart from this introduction, the top−level sheet briefly explains the working principle of POxiM and gives an block diagram of the physical system. Each subsheet explains the implemantion enclosed therein.

# Bibliography

This design was informed by NXP's AN4327 appnote, TI's SLAA655, SLAA274, and SLAA458 appnotes, and TI's TIDA−00311 reference design.

# Changelog

Rev A.0: Completed first iteration
Rev A.1: Incorporated small learnings from first full prototype

# How it works

POxiM is based on the tranmissive pulse oximetry principle. This means that it determines the patient's heart rate and blood oxygen concentration (SPO2) by measuring the portion of red (660nm) and infrared (940nm) light that passes through their finger.

This is enabled by two effects. First, the optical properties of hemoglobin change when it bonds with oxygen, so that blood with a high oxygen saturation has a different absorption ratio between red and infrared light than blood with low oxygen saturation. Second, the pressure variations caused by heartbeats cause blood vessels to expand and contract, thus changing the amount of absorption at both of these wavelengths.

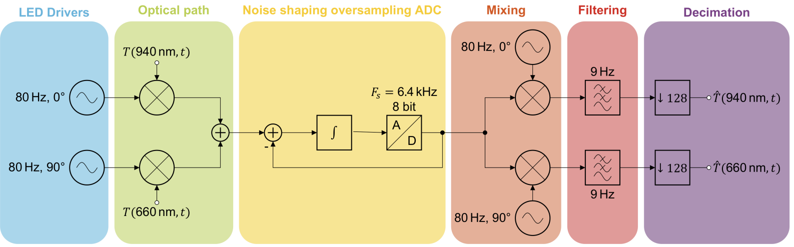
A transmissive pulse oximeter has three main tasks:

- Estimating the transmissivity of 660nm and 940nm light through a finger.
- Using this data to estimate the pulse rate and SPO2 of the patient.
- Presenting this data to the patient and/or caretaker.

The latter two are relatively easy from a hardware standpoint. POxiM uses a microcontroller to calculate the heart rate and SPO2 from the transmissivity data using industry standard formulas and a custom low−weight peak extraction algorithm. This microcontroller also drives the display.

Estimating the transmissivity is significantly more difficult. The information is carried not only in the average DC component of the intensity, but also in the changing AC component of the signal, which has an amplitude of only 0.1% of the DC signal. Hence, measuring the signal requires a large dynamic range.

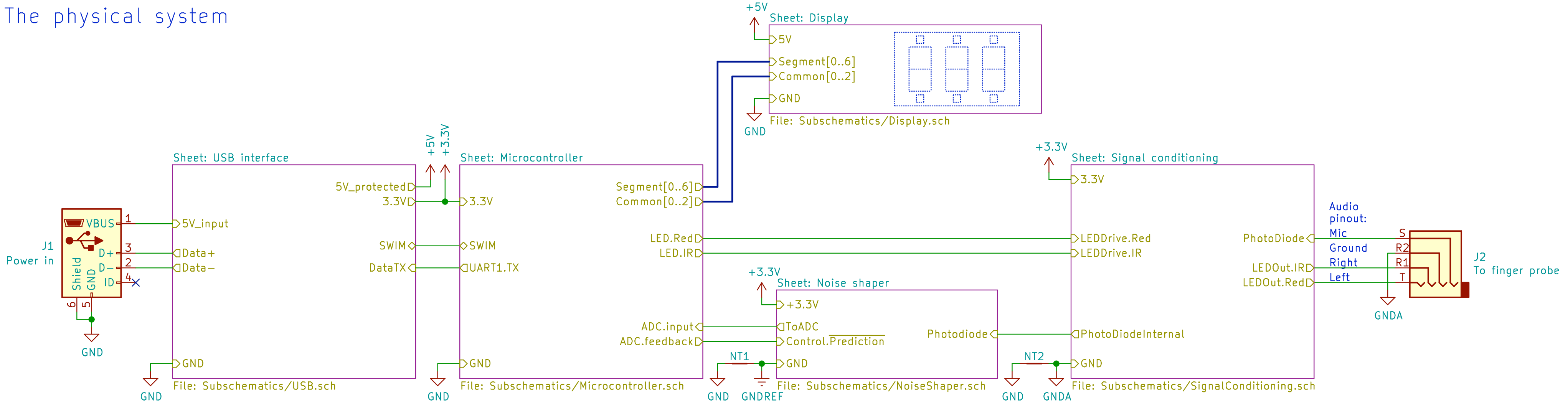
This dynamic range is achieved through oversampling combined with noise shaping, so that the design can be adapted to most microcontrollers. Oversampling spreads the ADC noise over a larger bandwidth frequencies, and noise shaping uses feedback to form a filter which suppresses this noise in the bandwidth of interest, at the expense of higher noise beyond the bandwidth of interest. Digital filters are then used to filter away the noise beyond the bandwidth of interest.




There is only a single optical channel to avoid the cost of optical filters and multiple channels. This introduces a problem, since the channel must be shared between the two wavelengths. Since the noise shaping circuitry is not compatible with sharing the channel in time, the channel is shared in the frequency domain. This is achieved by Quadrature Amplitude Modulation (QAM): the wavelengths are both modulated at 80Hz, but with unique and orthogonal phases. They are recovered by quadrature demodulation in software. This is the same principle used in software defined radio.

Finally, different skin types require a system dynamic range beyond the dynamic range required to measure the transmissivity. This is achieved by varying the amplitude of the modulating signal to achieve brightness control.

# The physical system




# Mechanical

H1 MountingHole
H2 MountingHole
H3 MountingHole
H4 MountingHole
H5 AlignmentHole
H6 AlignmentHole

# Production

FID1 Fiducial
FID2 Fiducial
FID3 Fiducial
FID4 Fiducial

# Power budget

There is 100mA available from a single power unit of USB. It is allocated as follows:

- Microcontroller: 5mA
- Readout circuit amplifier: 1mA
- Readout LEDs: 2 · 10mA = 20mA
- Display: 74mA
  - Digits: 7 · 10mA = 70mA
  - Transistor biasing: 4mA

TU Delft

Pulse/Oximeter measurement unit
By Arthur Admiraal & Daan de Groot
POxIM

POXiM

Sheet: /
File: POxiM−wristcomputer.sch

**Title: POxIM wrist computer**

| | |
|---|---|
| Size: A4 Date: 2020−07−16 | **Rev: A.1** |
| KiCad E.D.A. kicad (5.1.0−0) | Id: 1/6 |

# Working principle

This is an integrating differential pulse-code modulation quantiser inspired by a delta-sigma ADC and based directly on the lecture 'Sigma-Delta Modulation' of the TU Delft EE2S31 course on Signal Processing. See also: https://cas.tudelft.nl/Education/courses/ee2s31/

Together with the firmware, the goal of this circuit is to redistribute the ADC noise (both quantisation and otherwise) away from the low frequencies of interest, and towards higher frequencies. This enables a digital low-pass filter to nearly eliminate ADC noise, so that a much better effective resolution can be achieved (from 10 bits to 20 bits).

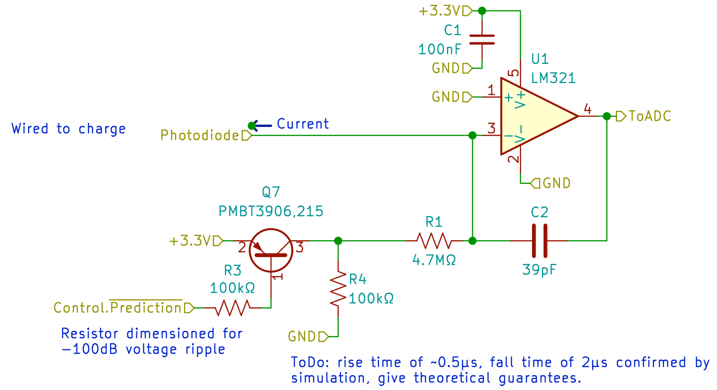
# Dimensioning

As a rule of thumb, noise shaping of order p and filtering to 1/(2^n) of the nyquist bandwidth gives:
$\Delta SQNR = n \cdot (3 + 6p)$ dB
Whereas additional bits of effective resolution give:
$\Delta SQNR = B \cdot 6.02$ dB
Assuming a common 10-bit ADC with a conservative 8 bits of effective resolution, using the first order noise-shaping implemented here and equating for 10 additional bits gives:
$12 \cdot 6.02 \text{ dB} < n \cdot (3 + 6 \cdot 1) \text{ dB}$
$n = 10 \cdot 6.02 / 9 \approx 6.69$
Since sample numbers which are a power of two are easier to work with, we round to n=7. Then the sample rate must be at least 2^7 = 128 times higher than the nyquist frequency. The bandwidth of interest is 10Hz, so the nyquist frequency is 20Hz, and the sample frequency must thus be at least 2.56kHz. However, this is only the sample frequency per channel. Since there are two channels, the system sample rate will be twice as high, or 5.12kHz. We sacrifice a little processing power for a nice 16MHz clock division at 6.4kHz.

From measurements, the minimum full scale input current is 1μA. The integration capacitor must be charged up to the full scale ADC voltage with a full scale input current within one period time. Then:
$Q = C \cdot V \Rightarrow C = I \cdot T / V = I / (V \cdot f) = 1\mu A / (5V \cdot 6.4kHz) = 32pF$
Allowing 10% margin, the capacitance value should be higher than
$C' = C / (1-\eta) = 32pF / (1-10\%) = 35.5pF$
39pF was selected as the closest E12 value.

There are several nonidealities which may mess up measurement:
- Capacitor leakage
- Controller noise (voltage & current)
- Resistor noise (voltage & current)
- Stability
- Bandwidth limitations
- Aliasing

Some simple models were formed to investigate the effects. ToDo: add them to this sheet.




TU Delft
Trades bandwidth to increase resolution
By Arthur Admiraal & Daan de Groot
**POxIM**
Sheet: /Noise shaper/
File: NoiseShaper.sch
***Title: Noise shaping***
Size: A4 Date: 2020-07-16 **Rev: A.0**
KiCad E.D.A. kicad (5.1.0-0) Id: 2/6

# Operation

The LEDs are multiplexed in digits. There are three numeric digits, 0 to 2, and a single auxiliary digit. Setting a single common pin high selects the digits with the corresponding number. Setting all the common pins low selects the auxiliary digit. See also the segment map and truth table.

# Why buffering?

I/O pins directly drive the segments. Transistors are used as buffers to sink and source the common currents. It would be even simpler to directly use I/O pins, but we found that the maximum average current of I = Io,max / Nled = 20mA / (4 · 7) = 0.71mA per LED does not give enough brighness for good readability. Buffering the common currents raises the maximum average current to I = Io,max / Ndigit = 20mA / 4 = 5mA, which is more than the maximum available current through the 100mA fuse.

# Power budget

There is 74mA available current for the display. Of this, 7 · 10mA is allocated to the LEDs, and 4mA to driving the buffer transistors.

# Why bipolar buffers?

There are many other possibilties to buffering the common currents. MOSFETs would be significantly less power hungry than bipolar transistors. The reasone we specified bipolar transistors, is that a circuit topology optimised for bipolar transistors (with base resistors) is compatible with MOSFETs, but a MOSFET-optimised topology (without base resistors) would not be compatible with bipolar transistors, thus restricthing the pool of suitable parts.

# Valid states

| Common 0 | Common 1 | Common 2 | Selected digit |
|---|---|---|---|
| 1 | 0 | 0 | Digit 0 |
| 0 | 1 | 0 | Digit 1 |
| 0 | 0 | 1 | Digit 2 |
| 0 | 0 | 0 | Digit AUX |

# Segment map

Digit AUX

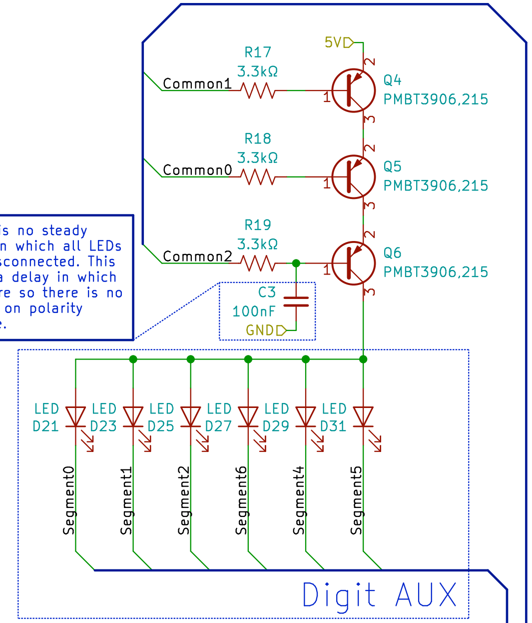
Segment 1 | Segment 0 | Segment 2

Digit 0 / Digit 1 / Digit 2: Segment 0, Segment 1, Segment 2, Segment 3, Segment 4, Segment 5, Segment 6

Digit AUX

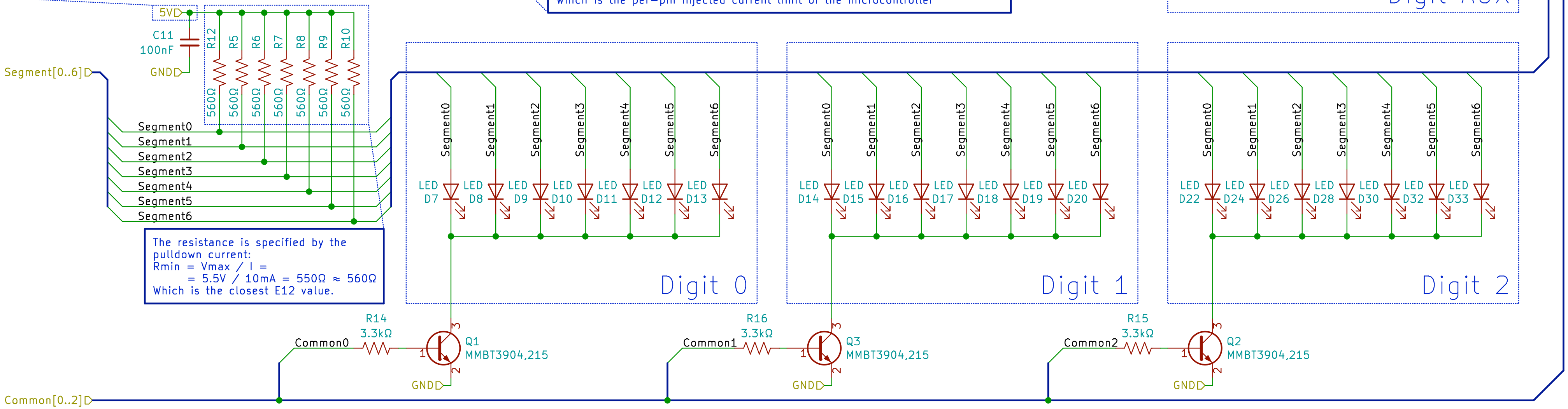
Segment 4 | Segment 6 | Segment 5

There is no steady state in which all LEDs are disconnected. This gives a delay in which they are so there is no flicker on polarity change.




Pullups are used to decrease the current ripple through the power supply of the microcontroller. Segment 5 and 6 are open-drain only, so pullups are used. Play close attention to the supply path. It is a potential source of EMI issues, since high-frequency currents flow through here. This costs a factor 2 in brightness, but saves buffering.

The microcontroller relies on the forward voltage of the LEDs to prevent the voltage rising above its supply. In case of a single point failure, the resistor will limit the current from exceeding:
Imax = (Vmax-Vs) / R = (5.5V - 3.3V) / 560Ω = 3.9mA
Which is the per-pin injected current limit of the microcontroller




The resistance is specified by the pulldown current:
Rmin = Vmax / I =
= 5.5V / 10mA = 550Ω ≈ 560Ω
Which is the closest E12 value.

# **Warning! This sheet contains bugs.**

This sheet contains major design oversights related to driving the auxiliary segments:

- Because the microcontroller runs from 3.3V, but the PNP transistors from 5V, they cannot shut off. Current is injected into the microcontroller pins.
- Because the outputs are open drain with pullup resistors, the auxiliary LEDs are driven without current limiting resistors, which will damage the microcontroller and the LEDs.
- The microcontroller relies on the LEDs for clamping the voltage on its pins to below the supply. However, the auxiliary digit doesn't provide this functionality, so that too much current is injected into the pins.

TUDelft

3 7-segment digits and indicator LEDs

By Arthur Admiraal & Daan de Groot

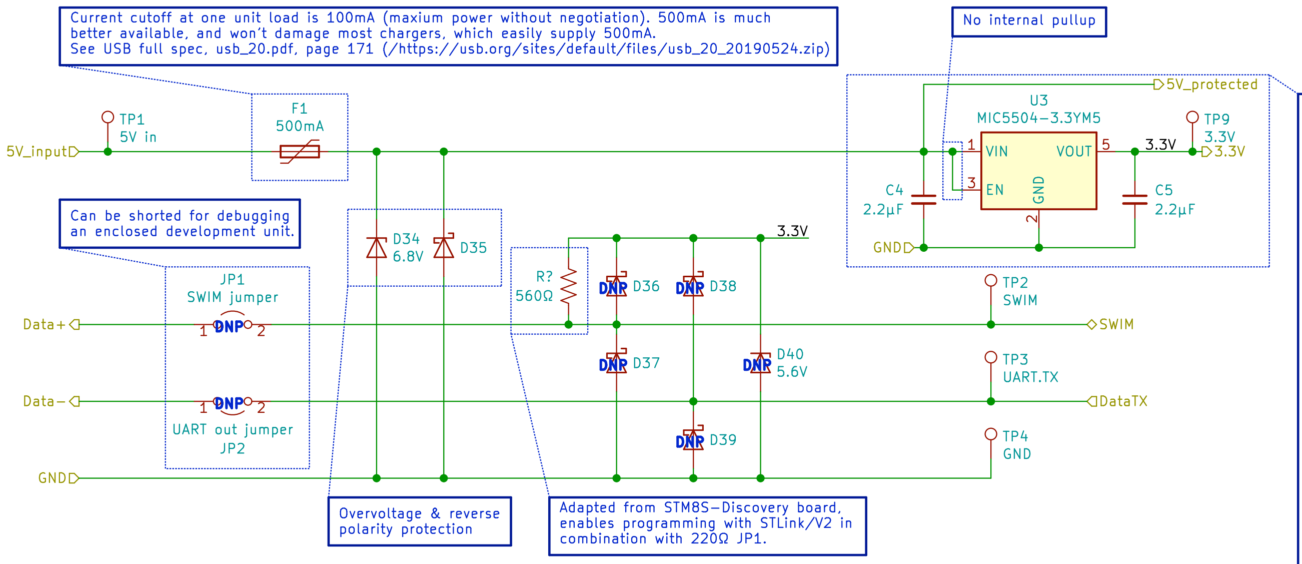
POxIM

Sheet: /Display/

File: Display.sch

**Title: Discrete LED display**

| | | |
|---|---|---|
| Size: A4 | Date: 2020-07-16 | Rev: A.1 |
| KiCad E.D.A. kicad (5.1.0-0) | | Id: 3/6 |




At first we thought no LDO would be needed, since all circuitry can operate at 5V. However, LDOs do more than just voltage conversion. They also reject power supply ripple and present a low output impedance, as we found out the hard way.

Since the STM8 has no separate analog supply, voltage ripple on the supply is mixed directly with the input, so noise in the 0Hz to 100Hz band should be >100dB down. Sadly, the LEDs are driven with significant components at these frequencies, and a 10mArms component at 3.3V would require a <3.3mΩ supply impedance, which is not easily achievable.

Hence, a two-step approach was taken. First, a <1Ω resistance was ensured for the polyfuse. This places the noise at 10mA · 1Ω / 3.3V = 50dB down. Furthermore, the LED current was not sourced by the microcontroller, but only sunk. Then, an LDO was used to isolate the microcontroller supply from the main supply. This gives an additional >60dB of ripple rejection to achieve >100dB PSRR.

However, achieving this PSRR is actually even more complicated. Since the microcontroller also doesn't have a seperate analog ground, the resistance of the ground bond wire (assumed <50mΩ) wire will induce excissive ripple at 10mA · 76mΩ / 3.3V = −75dB. Buffering could be used to reduce the current through the microcontroller, but would create a supply dependence on a logic IC or greatly increase the transistor part count. A more elegant solution is that this ripple can be filtered out if it only appears in one location in the delta-sigma loop. The noise coupling to the ADC is unavoidable, but the DAC output can be buffered. This reduces the noise by >40dB, bringing it back >100dB down.

Also note that the transistor drive currents were not diverted away from the microcontroller supply, since they can be kept period. That way, they will only have DC and >100Hz components, so that they don't interfere with the measurement.

TUDelft

Prevents excessive voltages and currents

By Arthur Admiraal & Daan de Groot

**POxIM**

POXIM

Sheet: /USB interface/

File: USB.sch

**Title: USB interfacing**

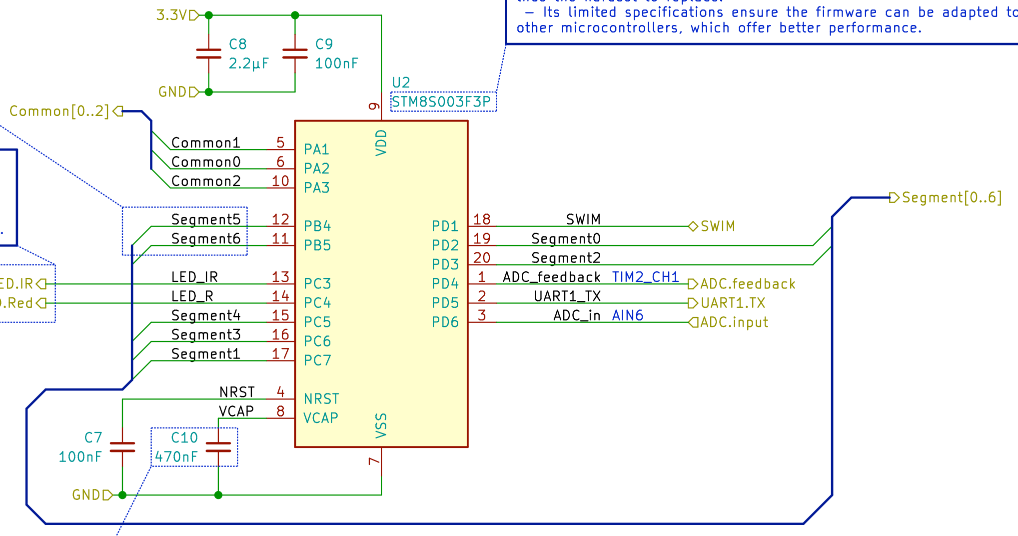
| | | |
|---|---|---|
| Size: A4 | Date: 2020-07-16 | **Rev: A.1** |
| KiCad E.D.A. kicad (5.1.0-0) | | Id: 4/6 |

STM8 microcontrollers are commonly seen as a pretty poor. However,
it has some advantages which made it the perfect choice for this design:
– It is one of the most widely available microcontrollers (Dutch Farnell data), with >1M units available in Europe alone accross distributors. This is of paramount importance, since it is the most integrated part in the design, and thus the hardest to replace.
– Its limited specifications ensure the firmware can be adapted to most other microcontrollers, which offer better performance.

PB4 and PB5 are open–drain only. This is taken into account in the LED display sheet.

Two–level 80Hz PWM of thresholded sine and cosine. The modulation frequency was chosen to provide sufficient channel separation to 50Hz – 60Hz line voltages and harmonics thereof and to provide sufficient channel separation to the DC component and subharmonics. Furthermore, it was verified that there would be no perceptable flicker on the red LED at this frequency, since it is visible to the user.

80Hz, I TIM1_CH3
80Hz, Q TIM1_CH4




100nF capacitors could be replaced with 470nF for BOM consolidation, but since we're still far under the maximum unique part count of most assemblers, we keep the 100nF capacitors, which are 25% cheaper at scale. For production quantities > 1000, it is also not worth replacing the 470nF capacitor with 5 100nF capacitors.

## Important

High speed internal RC oscillator (HSI) should be user–trimmed with CLK_HSITRIMR register at given TA and VDD conditions to attain 1.0% accuracy.

TUDelft

Performs signal processing and control

By Arthur Admiraal & Daan de Groot

**POxIM**

POXIM

Sheet: /Microcontroller/

File: Microcontroller.sch

**Title: Microcontroller**

| | | |
|---|---|---|
| Size: A4 | Date: 2020–07–16 | **Rev: A.1** |
| KiCad E.D.A. kicad (5.1.0–0) | | Id: 5/6 |

All pins are designed to survive three scenarios:
- Hotplugging of the 4 pole 3.5mm jack
- Short-circuits to ground
- Application of line audio

When the 3.5mm jack is hotplugged, the worst that could happen is that outputs short together or to ground. Hence, to satisfy this requirement, the inputs need to handle short-circuits to voltages between 0V and 5V. This also satisfies the short-circuit to ground requirement.

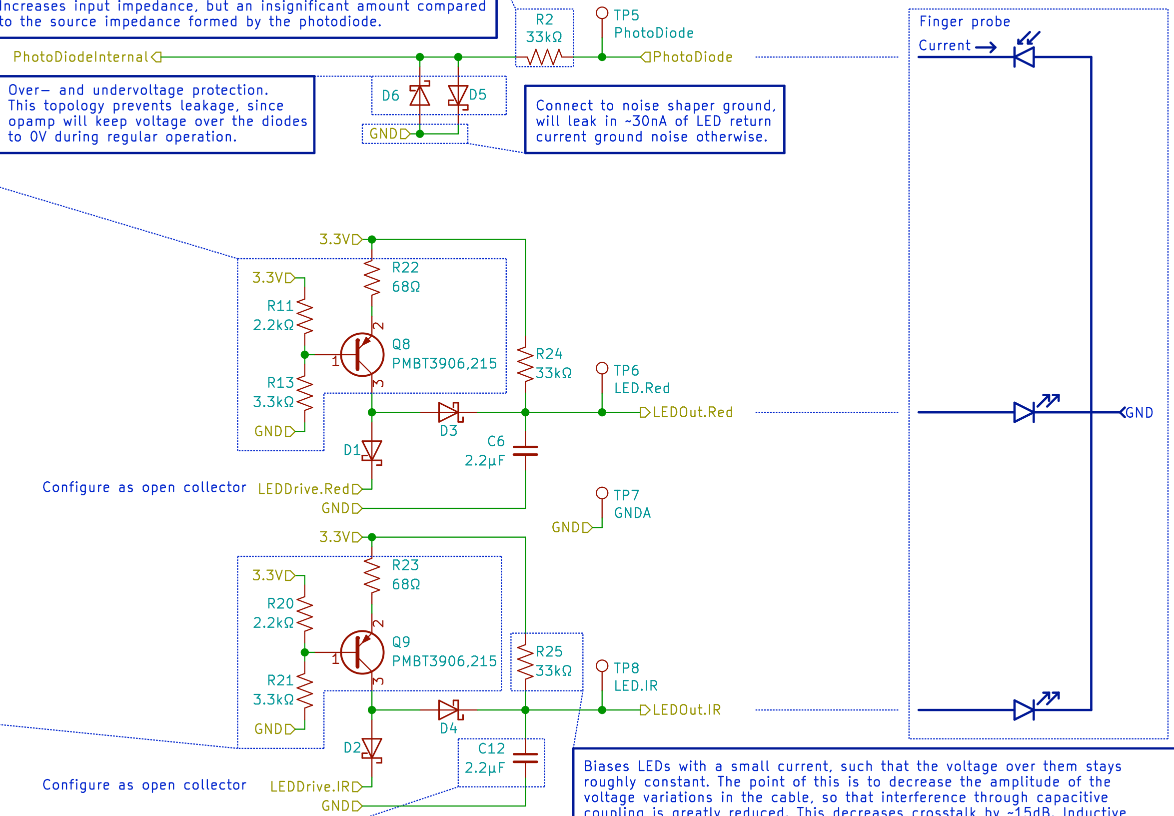
The voltage level of common 3.5mm headphone jacks is <2V in amplitude (see also https://electronics.stackexchange.com/questions/28404/what-is-the-voltage-range-of-a-standard-headphone-jack-from-a-phone). Thus, all pins need to handle shorts to voltages ∈ [−2V; 2V].

Hence, all three requirements are satisfied if all pins can handle short-circuits to voltages between ∈ [−2V; 5V].

Limits input current compared on overvoltages and undervoltages. Increases input impedance, but an insignificant amount compared to the source impedance formed by the photodiode.

Over- and undervoltage protection. This topology prevents leakage, since opamp will keep voltage over the diodes to 0V during regular operation.

Connect to noise shaper ground, will leak in ~30nA of LED return current ground noise otherwise.




This is a constant current source based on an emitter follower buffering a constant voltage over a constant resistance. When the LED is off, the current is diverted to the open colelctor I/O of the microcontroller. Hence, a continuous current is draw, so that low current ripple is achieved.

The current is set by the voltage over the constant resistance. This voltage should be lower than Vs − Vf,LED,max − Vsat = 3.3V − 2.2V − 0.3V = 0.8V. For 10mA output current, this gives R < V / I = 0.8V / 10mA = 80Ω. We select 68Ω for some margin. Now for a PNP transistor, the output current is given by:
Io = (Vs − Vb − 0.81) / R
Using a voltage divider, Vb = α · Vs. Then:
Io = (Vs − α · Vs − 0.81) / R = ((1 − α) · Vs − 0.81) / R
Solving for α:
((1 − α) · 3.3V − 0.81) / 68Ω < 10mA
α > 1 − (10mA · 68Ω + 0.81) / 3.3V = 0.59
Which is achieved with a 2.2kΩ and 3.3kΩ resistor, giving:
α = 3.3kΩ / (2.2kΩ + 3.3kΩ) = 0.6
Then:
Io = ((1 − 0.77) · [4.3V; 5.5V] − 0.81) / 47Ω = [3.8mA; 9.6mA]
Which is acceptable. The maximum total current of It,max = Io,max + Ib,max = = Io,max + Vs,max / R = 9.6mA + 5.5V / (820Ω + 3.3kΩ) = 10.9mA is slightly out of the budget. However, the actual current will be decreased due to the current flowing through the voltage divider into the bias pin. Hence in reality, it will be in budget.

A PMOS can also be used, there:
Io = (Vs − Vb − Vth) / R = ((1 − α) · Vs − Vth) / R
α > 1 − (10mA · 47Ω + Vth) / 5.5V
Giving:
Io < [7.8mA; 10mA] − [4.6mS; 0mS] · Vth
So, for operation down to the lowest expected input, let
Io,min = Io,max / 4 = 2.5mA, so that:
Vth,max = 7.8mA − 2.5mA / 4.6mS = 1.1V

# Why the constant current drive?

At first the LEDs were driven through a simple resistance. However, the current ripple of the LEDs caused a supply voltage ripple. This ripple mixed with the LED drive signal to cause channel bleeding. To eliminate the ripple, the LED drive current is only diverted away from the LEDs, not halted, so that a continuous current is achieved.

Filter LED PWM harmonics above:
f = 1 / (2π · R · C) =
= 1 / (2π · 2.5Ω · 2.2µF) = 29kHz
Prevents sourcing significant power at frequencies where the cable is a good antenna. Resistance is small-signal resistance of LED.

Biases LEDs with a small current, such that the voltage over them stays roughly constant. The point of this is to decrease the amplitude of the voltage variations in the cable, so that interference through capacitive coupling is greatly reduced. This decreases crosstalk by ~15dB. Inductive coupling can't be avoided, since the current must be varied to create brightness variations in the LED.

The biasing doesn't work if the LED drive pins force no voltage over de LED when it isn't driven. Hence, they must be configured as open collector.

The bias current is I = (Vs − Vf) / R = (5V − 1.85V) / 33kΩ = 0.1mA. This is comparable to the minimum brightness setting of
Imin = Ifs / N = 20mA / 166 = 0.12mA, so it won't hamper brightness scaling.


TUDelft
Performs filtering, prevents ESD damage
By Arthur Admiraal & Daan de Groot
POxIM

POXiM

Sheet: /Signal conditioning/
File: SignalConditioning.sch

**Title: Audio jack signal conditioning**

| | | |
|---|---|---|
| Size: A4 | Date: 2020-07-16 | Rev: A.1 |
| KiCad E.D.A. kicad (5.1.0-0) | | Id: 6/6 |