



Swakeup

PROJECT REPORT

within the lecture of
Programming Embedded Systems

at Uppsala University
in the Departement of Information Technology

Elmar van Rijnsouw (Elmar.Vanrijnsouw.9818@student.uu.se)
and Maximilian Stiefel (Maximilian.Stiefel.8233@student.uu.se)

Deadline: 2017-05-31 24:00

Processing Period:
Supervisor:

January 2017 - May 2017
Philipp Rümmer (philipp.ruemmer@it.uu.se)

Contents

1	Introduction	1
2	Background and Analysis	2
3	Design	3
4	Implementation	4
4.1	Hardware	4
4.2	Software	4
5	Testing	5
6	Conclusion	6
	Bibliography	IV
A	Schematics Logic Board	V
B	Schematics Power Board	VI
C	USART interrupt generation	XI

List of Figures

List of Tables

1 Introduction

It is a well-known fact, that Sweden is a country with big variations of the daily hours of sun throughout the year. Far up north the sun does not set anymore in January. Even in Stockholm in January, the most dark month of the year, the sun rises at 8:47 am and sets at 2:55 pm (cf.[1]). According to *Sveriges Radio* "many Swedes suffer from the winter blues or seasonal affective disorder" (cf. [2]). In a strong winter every source of light is a source of happiness. This is why a wakeup light, which is based on a strong light source (at least 10 W RGB LED), is able to give one the optimal start into a dark winter day with an artificial sunrise as bright as a real sun shining through the window.

This report describes the Swakeup (from engl. "Swedish Wakeup Light"), a device communicating to the user not only through light. It does not simply wake one up, but also gives one information about social media, latest mails, calendar and weather. The user interface consists besides of a high-power LED of an OLED screen. Swakeup is also part of the IoT as it has the ability to communicate via IEEE 802.11. This of course enables a lot of possibilities e.g. connecting your phone to the wakeup light. A lot of effort has been put into the designing maxim, that everything should be as small as possible. The whole electronics fit on an base area of 5 cm x 4 cm. So the Swakeup fits smoothly on the bedside table. And honestly: What is the last thing people are doing before they go to sleep? Right! They look on your phone. That is why Swakeup comes with a USB charger for your e.g. phone as well. Another design maxim of this product is cheapness. Everybody should be able to buy one. As all engineering work is available online, it gives people (with the corresponding knowledge) the opportunity to build a wakeup light themselves, look up what this device is doing with their personal data or even contribute to the product.

2 Background and Analysis

3 Design

4 Implementation

4.1 Hardware

4.2 Software

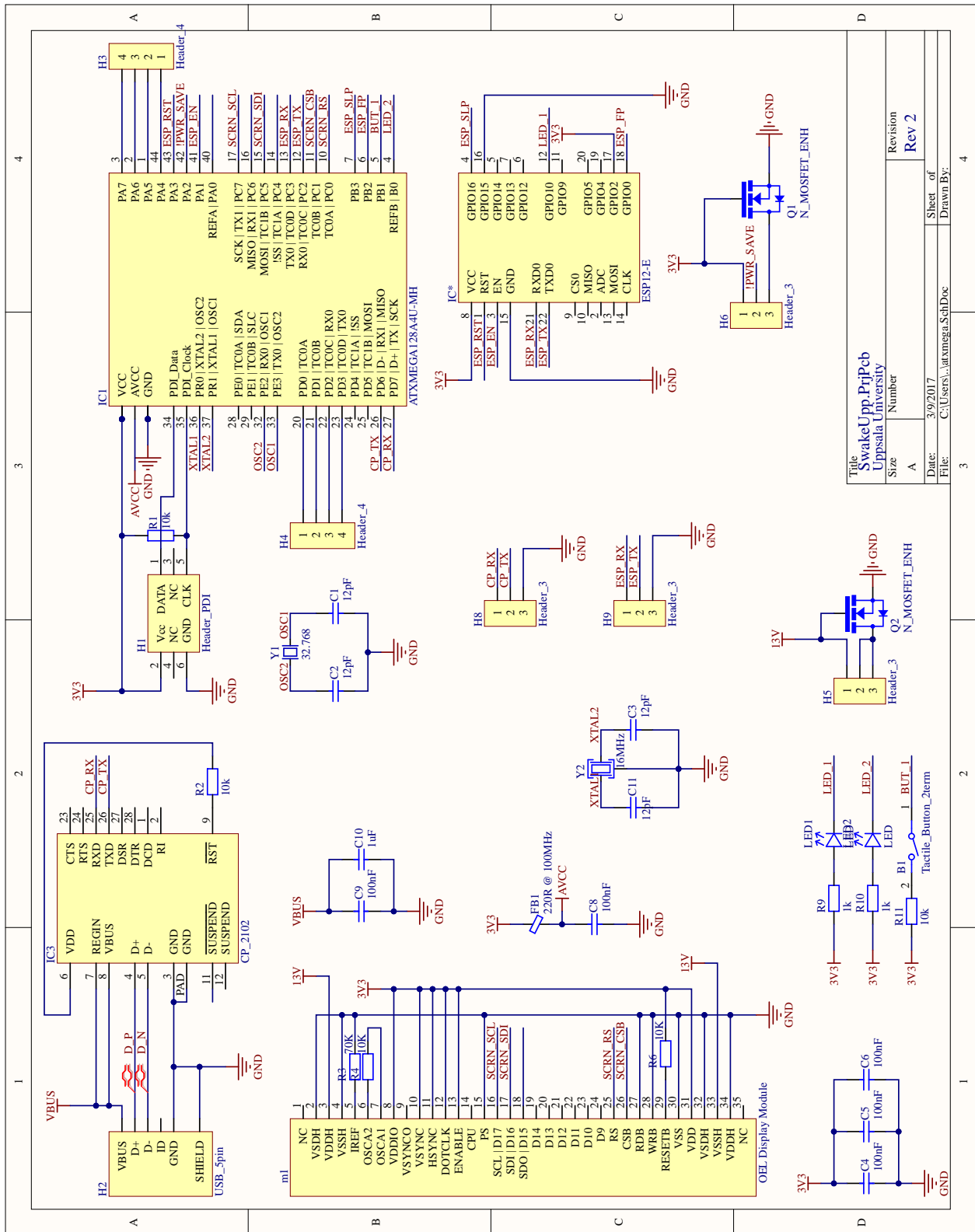
5 Testing

6 Conclusion

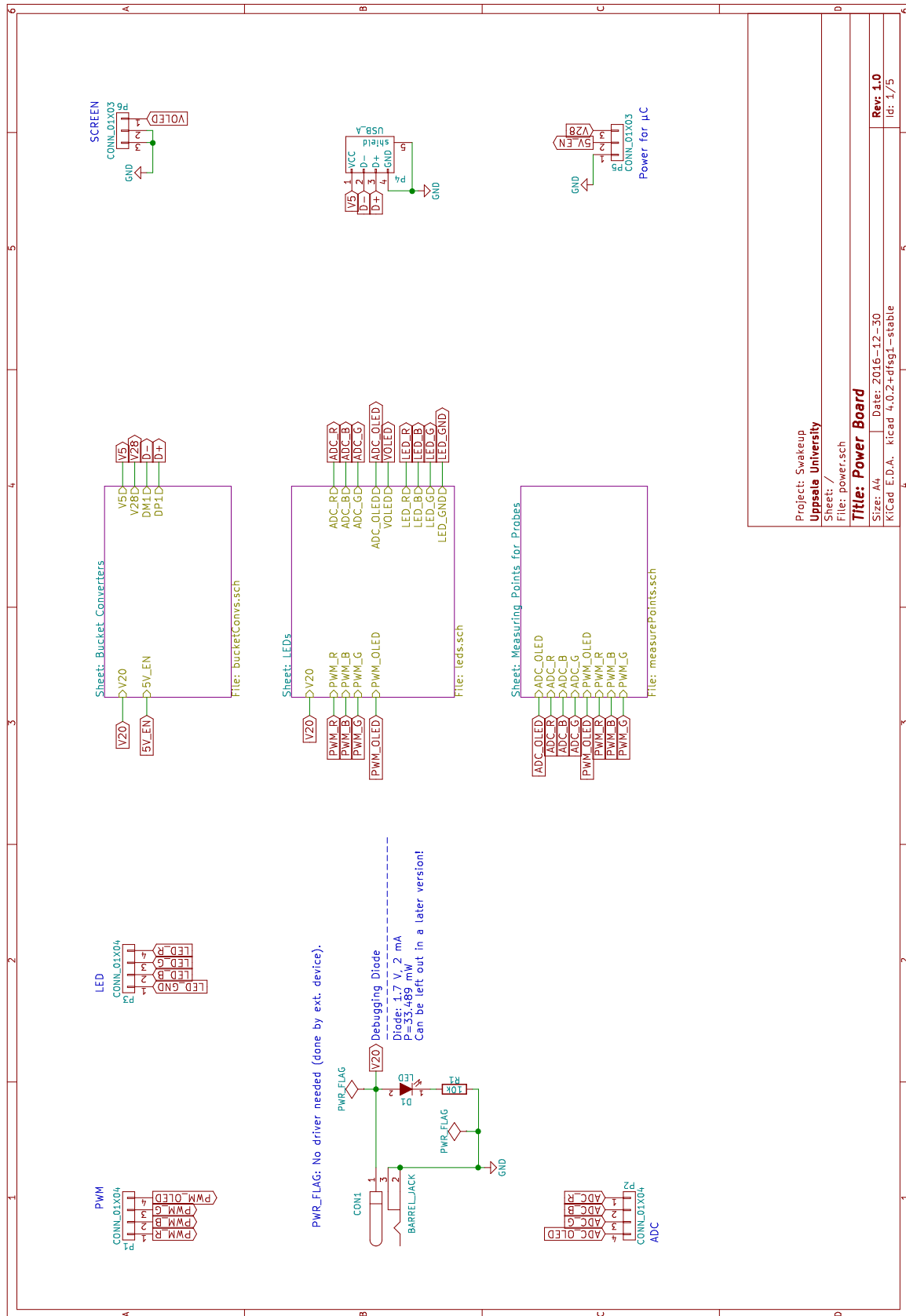
Bibliography

- [1] *Visit Sweden*. (2017). Time and daylight in Sweden, [Online]. Available: <https://visitsweden.com/time-and-daylight-hours/>.
- [2] *Sveriges Radio*. (2014). Are Swedes more suicidal than most? [Online]. Available: <http://sverigesradio.se/sida/artikel.aspx?artikel=5924063>.

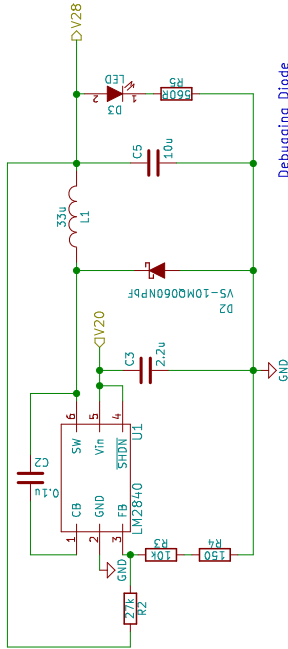
A Schematics Logic Board



B Schematics Power Board

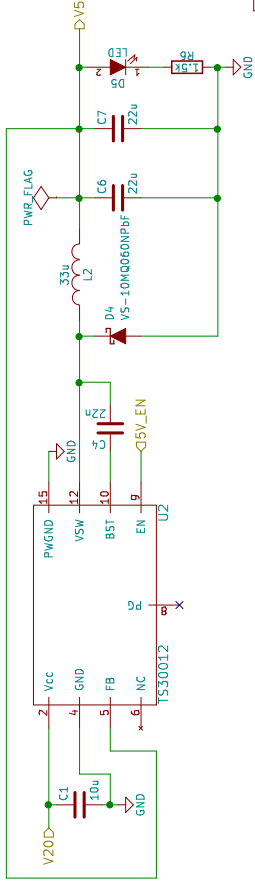


2.8 V for μC
 $V_{out} = 0.765 V \cdot (1 + (27k / 10.15k)) = 2.8 V$
 $27k / 10.15k = 2.66$

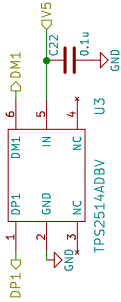


Debugging Diode
 Diode: 1.7 V, 2 mA
 Power: 2.64 mW
 Can be left out in a later version!

5V for Mobile Phone
 $V_{out} max = 5 V$
 $A_{out} max = 2 A$

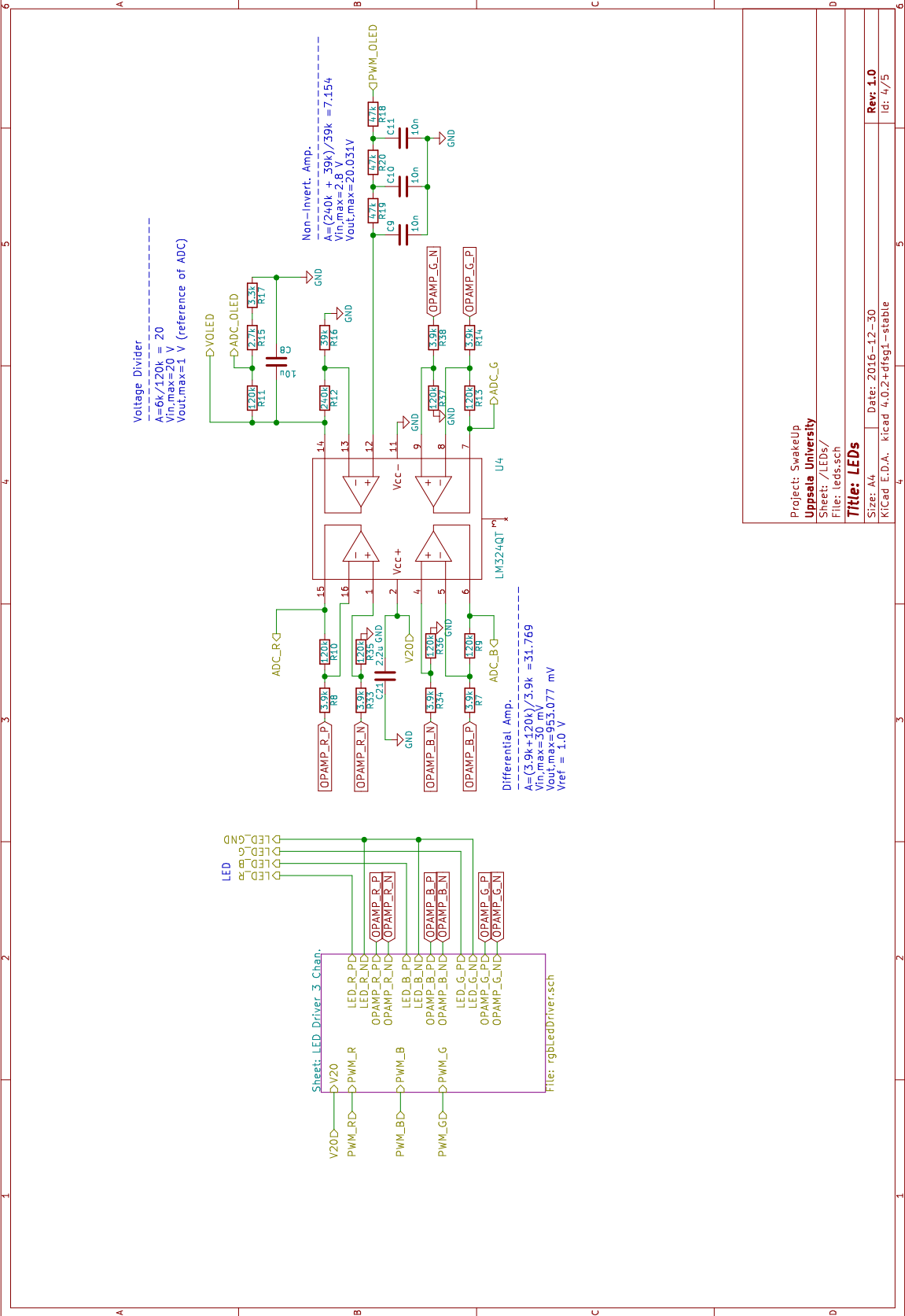


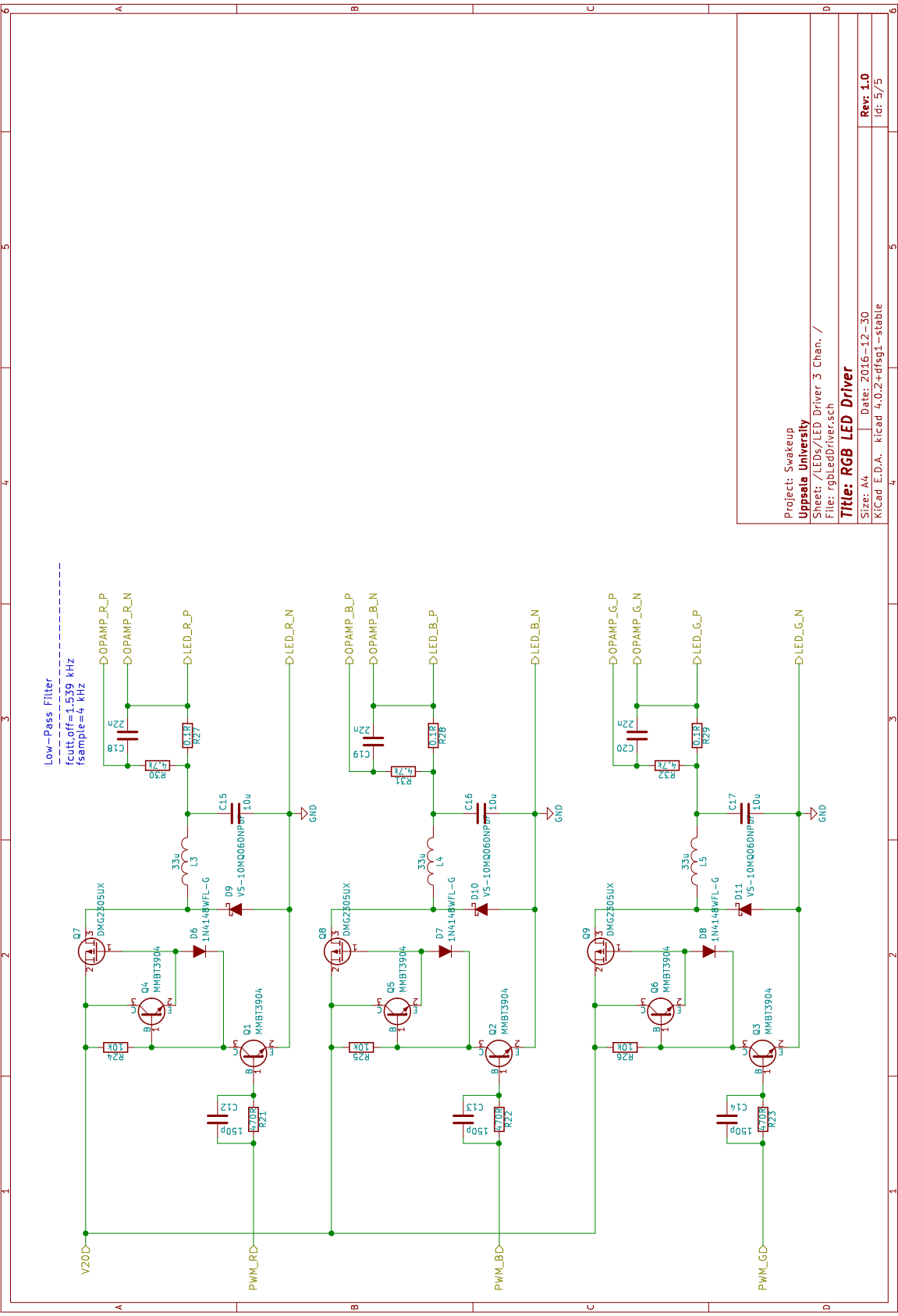
Debugging Diode
 Diode: 1.7 V, 2 mA
 Power: 2.64 mW
 Can be left out in a later version!



USB Dedicated Charging Port Control.
 Simple SOT-23-6 IC for detecting proprietary and open standards used by a device and providing the corresponding electrical signature at the data lines (voltage or impedance).

Uppsala University	
Sheet: /Bucket Converters/	
File: bucketConv.sch	
Title: Bucket Converters for 5 V and 2.8 V	
Size: A4	Date: 2016-12-30
KiCad E.D.A.	kiCad 4.0.2+dfsg1--stable
Rev: 1.0	Id: 2/5





Project: Swakeup
Uppsala University
Sheet: /LEDs/LED Driver 3 Chan. /
File: rgbLedDriver.sch
Title: RGB LED Driver
Size: A4 Date: 2016-12-30
KiCad E.D.A. KiCad 4.0.2+dfsg1-stable
Rev: 1.0
Id: 5/5

C USART interrupt generation

```
1  #define USARTRXCISR(NAME, PORT, USART_ID, REC_FC) \
2  ISR(NAME##_RXC_vect) { \
3      uint8_t read = PORT.DATA; \
4      if (writeInBuf(read, &PORT)) { \
5          REC_FC(read); \
6          uint8_t i = 0; \
7          for (; i < UART_MAX_DELIMITERS; i++) { \
8              if (delimiters[USART_ID][i].delimiter != 0) { \
9                  delimiters[USART_ID][i].length++; \
10                 if (read == delimiters[USART_ID][i].delimiter) { \
11                     delimiters[USART_ID][i].port = &PORT; \
12                     event_fire(&EVENT_UART_DELIMITER, \
13                             SYSTEM_ADDRESS_CAST (&delimiters[USART_ID][i])); \
14                 } \
15             } \
16         } \
17     } else { /*buffer full */ \
18         CP_PORT.CTRLA &= ~(USART_RXCINTLVL_LO_gc); \
19     } \
20 } \
21 \
22 #define USARTDREISR(NAME, PORT, USART_ID)\
23 ISR(NAME##_DRE_vect) { \
24     uint8_t size = uartStatus[USART_ID].outBuffer_size; \
25     if (size > 0) { \
26         if (softlock(USART_ID)) {\
27             uint8_t tail = uartStatus[USART_ID].outBuffer_tail;\
28             PORT.DATA = outBuffer[USART_ID][tail]; \
29             uartStatus[USART_ID].outBuffer_size--;\
30             tail++; \
31             if (tail >= UART_MAX_OUT_BUFFER) tail = 0;\
32             uartStatus[USART_ID].outBuffer_tail = tail;\
33             unlock(USART_ID); \
34         } \
35     } else {\
36         sending[USART_ID] = 0;\
37         PORT.CTRLA &= ~(USART_DREINTLVLO_bm);\
38     } \
39 }
```