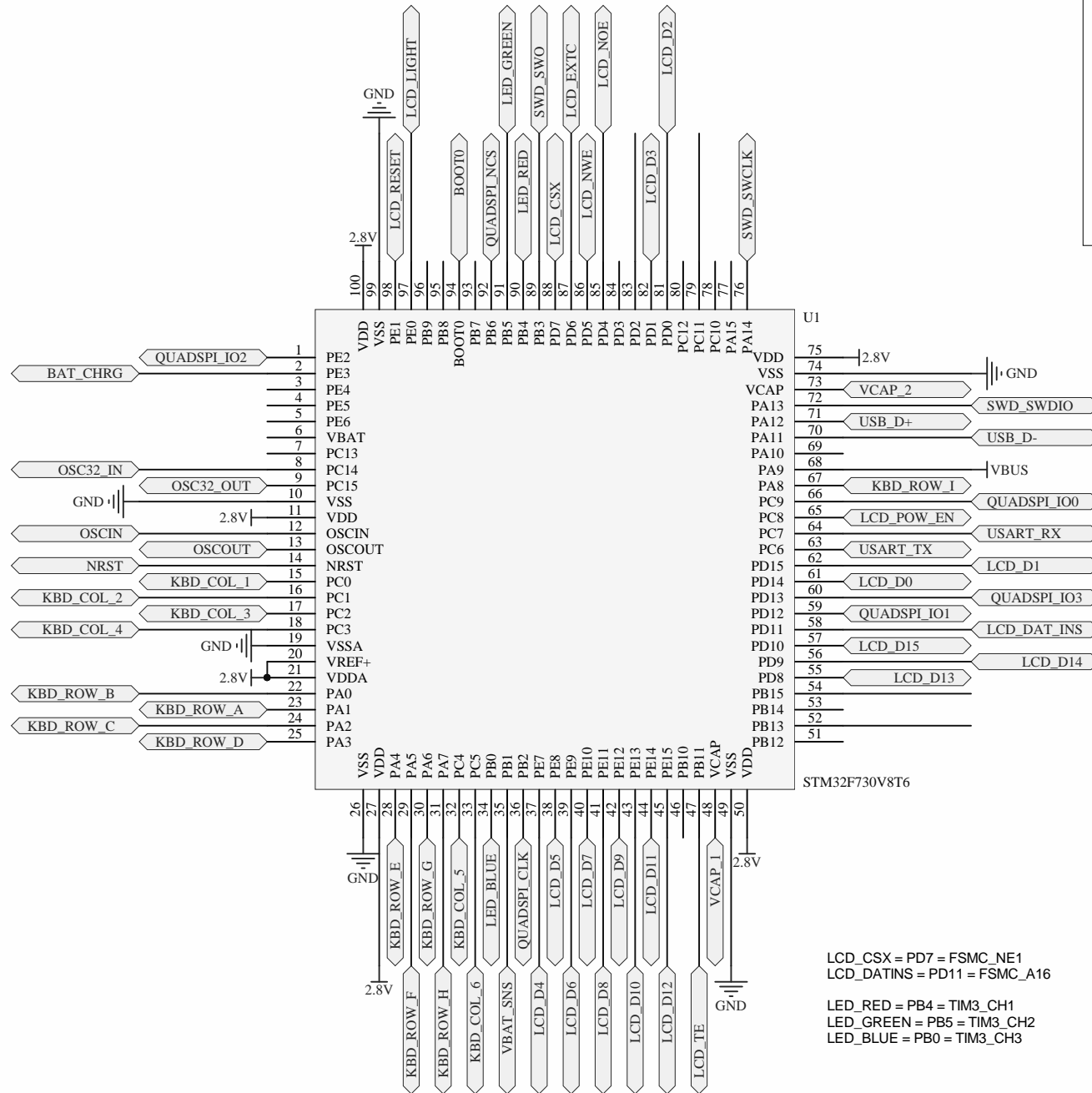


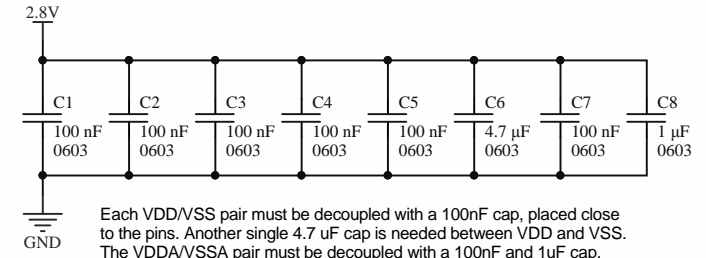
# MCU



LCD\_CSX = PD7 = FSMC\_NE1  
 LCD\_DATINS = PD11 = FSMC\_A16

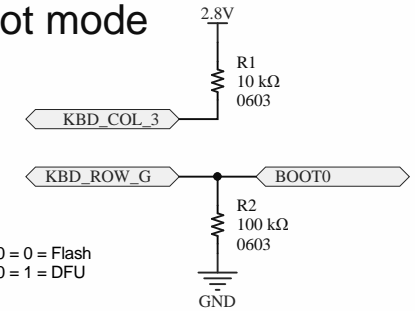
LED\_RED = PB4 = TIM3\_CH1  
 LED\_GREEN = PB5 = TIM3\_CH2  
 LED\_BLUE = PB0 = TIM3\_CH3

## Decoupling



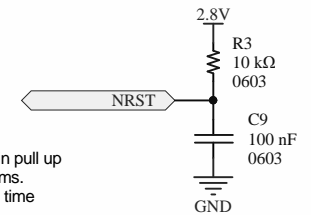
Each VDD/VSS pair must be decoupled with a 100nF cap, placed close to the pins. Another single 4.7 uF cap is needed between VDD and VSS. The VDDA/VSSA pair must be decoupled with a 100nF and 1uF cap.

## Boot mode



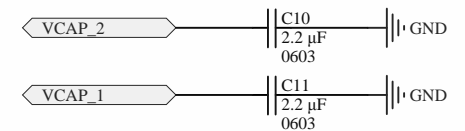
BOOT0 = 0 = Flash  
 BOOT0 = 1 = DFU

## Reset



NRST has a built-in pull up resistor of 40 kOhms.  
 Req = 80ohms. RC time constant = 0.8ms.

## Power supply



Voltage regulator requires a 4.7 uF capacitance on VCAP. On packages with 2 VCAP pins, that value is evenly split.

Title **MCU**

Size: A4

Number: 1

Revision:

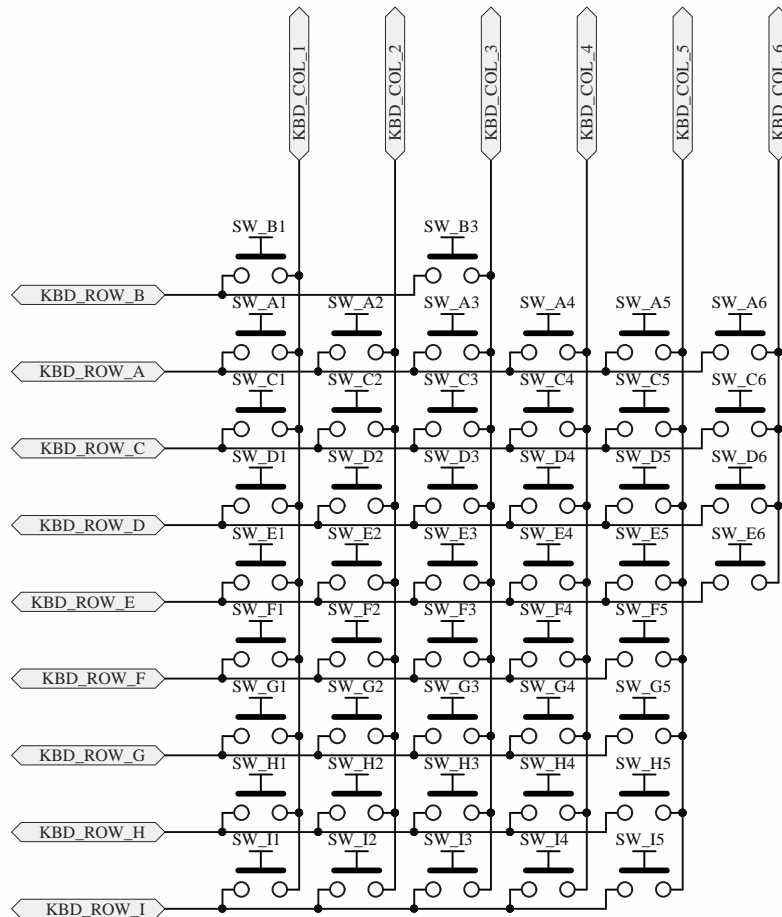
Date: 8/5/2019

Time: 2:37:40 PM

Sheet 1 of 5

# NUMWORKS

## Matrix keyboard



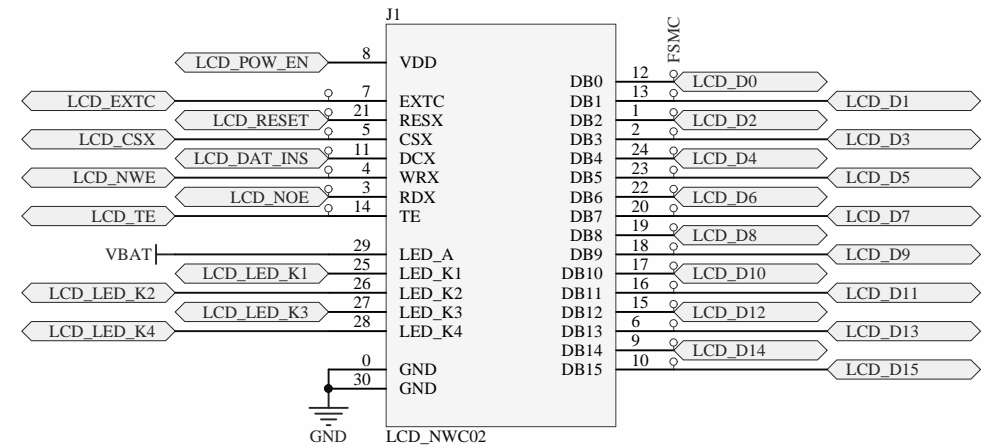
To prevent short-circuits, the software is configured as follows:

- COL are inputs, with a weak pull-up
- ROWs are outputs, which are either driven low or left floating.

This way, it is acceptable to:

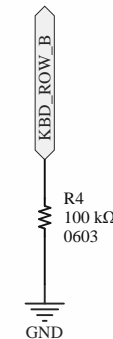
- Add a hard-wired pull-up on COL pins. This will simply reinforce the software-configured one.
- Add a hard-wired pull-down on ROW pins, as long as the resistance is high enough to be compensated by the input pull-up.

## LCD panel



ST7789V draws a max intensity of 7 mA, and STM32 can provide up to 20 mA per pin. That's why we're powering the LCD controller directly from the MCU.

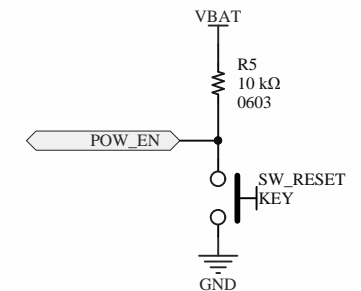
## Power button



In standby mode, KBD\_ROW\_B/PA0 pin will be configured as a WKUP pin waiting for a rising edge. We need to pull it down to make sure it's at zero by default.

When the Power button is pressed, KBD\_ROW\_B gets connected to KBD\_COL\_3 which has a 10K pull-up, which is stronger than the pull-down, therefore generating a rising edge.

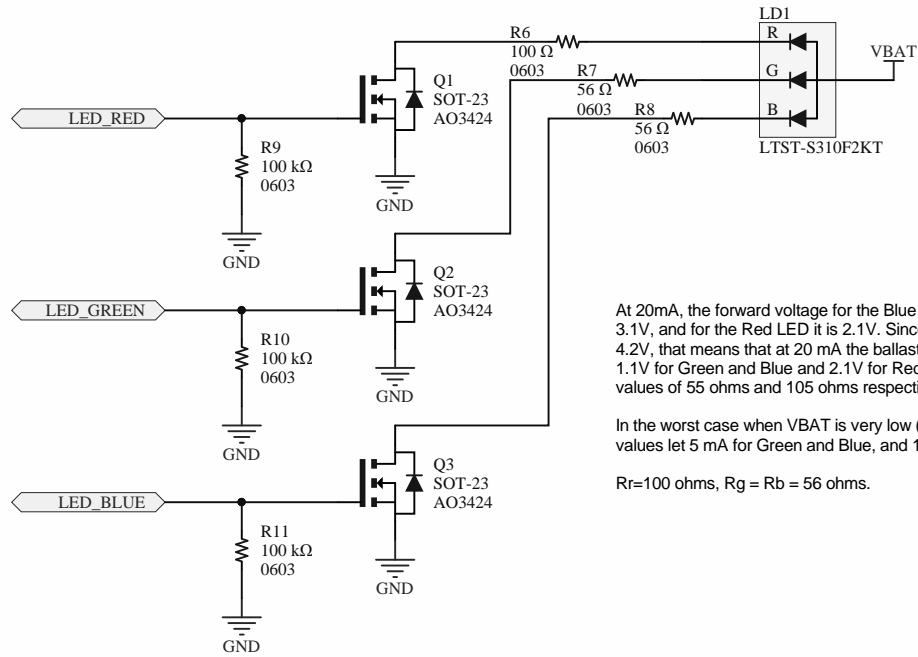
## Reset button



Title <b>UI</b>		
Size: A4	Number: 2	Revision: *
Date: 8/5/2019	Time: 2:37:40 PM	Sheet 2 of 5

# NUMWORKS

# RGB LED

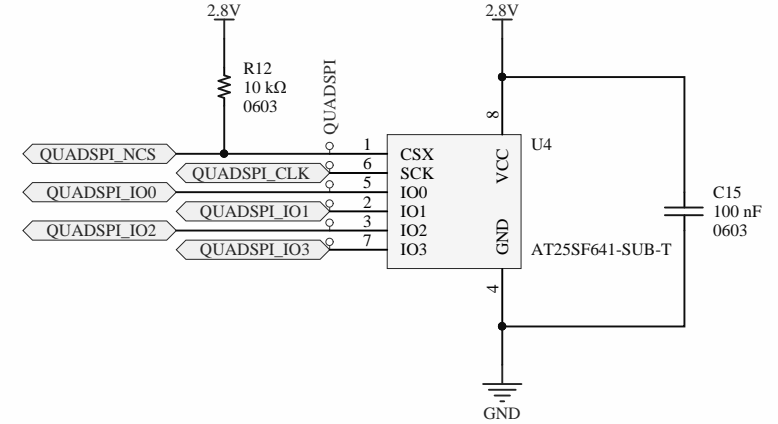


At 20mA, the forward voltage for the Blue and Green LED is 3.1V, and for the Red LED it is 2.1V. Since VBAT will be at most 4.2V, that means that at 20 mA the ballast resistor should drop 1.1V for Green and Blue and 2.1V for Red. That equates to values of 55 ohms and 105 ohms respectively.

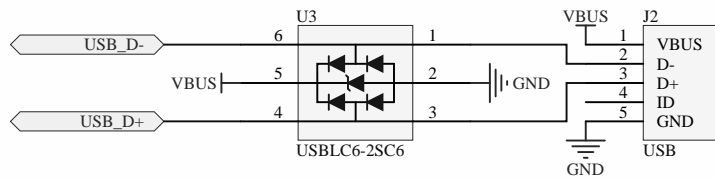
In the worst case when VBAT is very low (below 3.2V), those values let 5 mA for Green and Blue, and 10 mA for Red.

$R_r=100$  ohms,  $R_g = R_b = 56$  ohms.

# External Flash

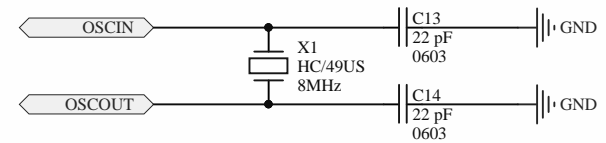


# USB port



We are a USB device, therefore we leave the ID pin floating.

# High-speed oscillator



$$CX = 2 * (CL - Cstray) = 2 * (16pF - 5pF) = 22pF$$

Title *Peripherals*

Size: A4

Number: 4

Revision: \*

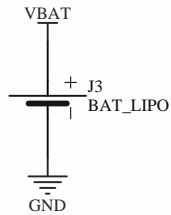
Date: 8/5/2019

Time: 2:37:41 PM

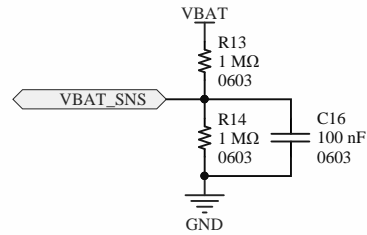
Sheet 4 of 5

# NUMWORKS

# Battery

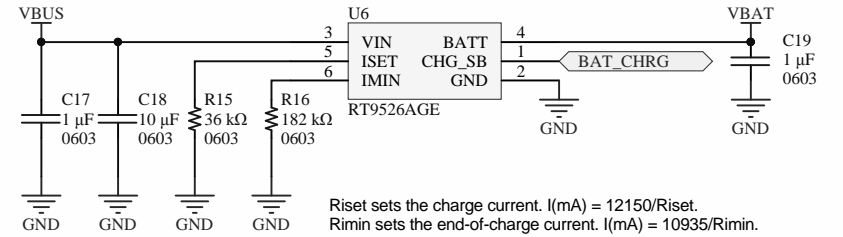


# Battery voltage measurement



VBAT being around 3.7V, the divider will waste 2uA. Which would take 74 years to deplete a 1200 mAh battery.

# LiPo USB charger

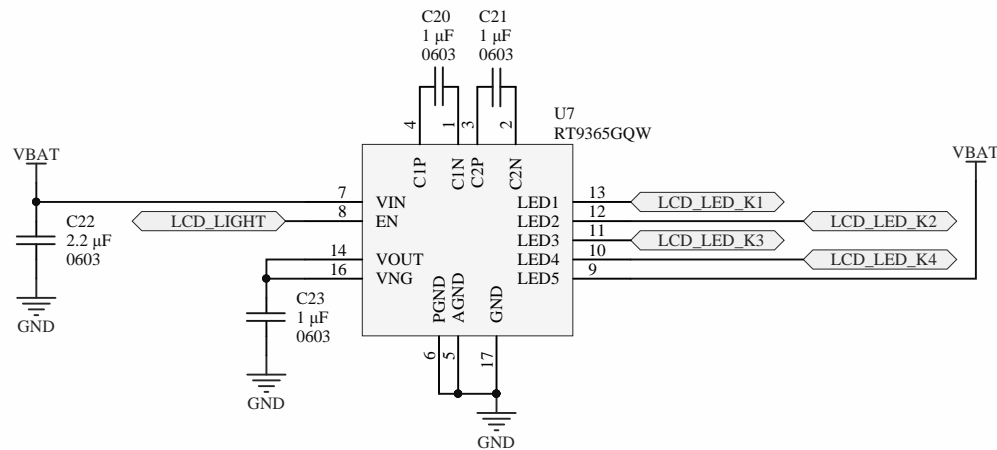


Riset sets the charge current.  $I(\text{mA}) = 12150/\text{Riset}$ .  
Rimin sets the end-of-charge current.  $I(\text{mA}) = 10935/\text{Rimin}$ .

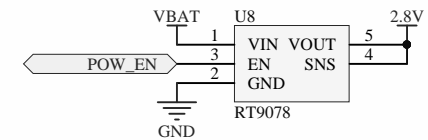
Riset = 36 kOhms, Iset = 337 mA  
Rimin = 182 kOhms, Imin = 60 mA.

We pick Imin = 60mA = I\_Batt(15mA) + 2\*I\_LED (2\*20) + I\_MCU(5mA)

# LCD backlight power supply



# Logic power supply



Title **Power**

Size: A4

Number:5

Revision:\*

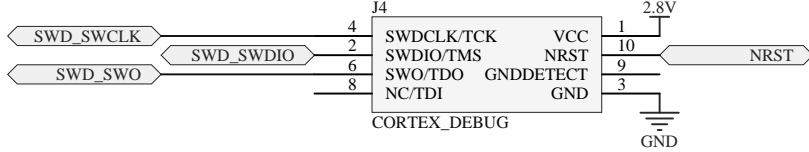
**NUMWORKS**

Date: 8/5/2019

Time: 2:37:41 PM

Sheet 5 of 5

### Cortex debug port



Title <i>Extras</i>		
Size: A4	Number:*	Revision:*
Date: 8/5/2019	Time: 2:37:41 PM	Sheet* of *

# NUMWORKS