Shotclock design choices / notes

A product that displays a countdown timer on seven segment displays once a button has been pressed. The countdown will show a maximum of 99 seconds (2 digits), and each of these 2 digits should be present on 4 sides of the product (so a total of 8 seven segment displays).

A buzzer is added to audibily hear when the timer has ran out.

See requirements below for more info.

Table of Contents

Requirements

1. Battery powered, rechargable

    1.1 Li-Po would be nice, since large capacity and shitload of form factors

    1.2 USB type C connector as charge plug (charging may stop usage of product if using li-po)

    1.3 USB type C connector as USB D+/D- to MCU

2. Can drive 2x4 seven (actually, eight segment. They have a dot.) segment displays, with brightness control

3. LED to display battery < 20%

4. LED to display charging status (is charging and is done charging)

5. Switch (latching) to turn product on and off, while still being able to charge product (preferably in both the on and off state. But only off state is also fine for version 1).

6. Three functional buttons (momentary) with hardware debouncing

7. STM32 MCU, with SWD connector

8. UART header

9. Bootloader button to STM

10. Buzzer to indicate that the timer has ran out.

11. EEPROM to store user settings (atleast countdown time and buzzer on/off). So RTC registers should be good enough.

Battery charger & ADC charge status

General info pages used

TP4056 as charging IC (https://dlnmh9ip6v2uc.cloudfront.net/datasheets/Prototyping/TP4056.pdf)

Page used on deciding charge current:

https://www.xtar.cc/news/What-Is-The-Best-Charging-Current-And-How-To-Calculate-The-Charging-Time-Of-Lithium-Battery-154.html

Disable system while charging battery:

https://www.microtype.io/lithium-ion-battery-charger-circuit-load-sharing/

Choosing battery of 1500 or 2000 mAh, since it is good middle-way between size, weight, price and capacity (should last the product more than long enough to be able to only charge it when systemload is off).

I got these in house: https://nl.aliexpress.com/i/1005005148897780.html?gatewayAdapt=glo2nld

With a thermistor in them. Still need to calculate R1 and R2 for the TP4056, by measureing ambient room temperature and measuring the thermistors resistance (used this article: https://www.best-microcontroller-projects.com/tp4056-page2.html)

3 pin JST connector:

https://jlcpcb.com/partdetail/171769-BM03B_SRSS_TB_LF_SN/C160389

I've also added an extra 2 pin connector, since a load of cells don't have a thermistor in them. A jumper is also present to pull the TEMP pin to ground, if no thermistor is to be used.

TEMP Pin NTC: R1, R2 calculation

The batteries I have, have a 10k NTC at room temp. I don't know the specifics, because it is a chinese battery, without good datasheet. Internet says "10K NTC usually has 3977K Thermistor Beta Value" (https://www.eevblog.com/forum/projects/ntc-thermistor-resistance-(li-ion-charging)/).

The datasheet of the TP4056 mentions: "If TEMP pin's voltage is below 45% or above 80% of supply voltage VIN for more than 0.15S, this means temp is to low or high".

So if *assuming* that the following is true:

For 45°C the thermistor resistance is about 4k2.

For 5°C the thermistor resistance is about 26k.

(at ambient room temp of +/- 20°C, the resistance was about 7k8, measured myself, so this assumption sounds about right)

I've converted the resistance calculator from above post: https://www.best-microcontroller-projects.com/tp4056-page2.html into Python code (inserted as OLE object, also found on my GitHub):

Then using the following input values for the Python script:

vratio1 = 0.45
vratio2 = 0.80
ratio_tol1 = 0.01
ratio_tol2 = 0.05
Rntc_min = 4200
Rntc_max = 26000

Gives the following closest E48 resistors (just as in the article. Atleast I know how to calculate this now hahaha):

r1 4780 r2 86600 Rntc1 4.2e3 Rntc2 26e3

I am also adding a jumper, to switch off the whole functionality as well. Both to support 2 pinned batteries, and to disable the temp sensing, if these assumption and/or calculations are wrong.

Battery charge reading & battery lvl status LED

To read the amount of charge in the battery, I want to use the ADC of the STM32.

This article was used as example: https://vivonomicon.com/2019/10/15/reading-battery-voltage-with-the-stm32s-adc/

It mentions that: "if you use smaller resistor values for the voltage divider, you use more current, but the sampling can be faster". A commentor on that post replied: "You can use larger resistors (i.e. 1M) in the divider and still use short sampling times if you add a small capacitor (1n-100n). [between the Analog pin and ground, to buffer so the ADC peripheral gets current more quickly. Instead of via the high resistance path of the voltage divider]".

So that is what is implemented. A 1:1 voltage division (a single Lithium cell get to a max of 4.2V, so divided by 2 will give a max of 2.1V, which the analog pins easily accept), with large resistance values (1M) and a 100n capacitor between the analog ADC peripheral and GND.

Max current leakage would be:

4.2 = I * 2000000
I = 0.0000021 amps (or 2.1 microamps).

Should be reasonable for a 1500mAh battery (or is it smart to increase the resistance values & capacitor value even more, to reduce the current leakage even more?).
And for the battery level status led:
Assuming a forward voltage drop of 2.0 volts and a forward current of 10 mA, we can calculate the resistor value using Ohm's Law (V = IR). Rearranging the formula, we get R = V / I.

R = (Vsupply - Vf) / I
R = (3.3V - 2.0V) / 0.01A
R = +/- 130 Ω (130 is closest E48 std. value.)

Voltage regulator (3v3)

TPS63000, variable output DC/DC buck boost.

Variable output calculation

From datasheet:

8.2.2 Detailed Design Procedure

8.2.2.1 Programming the Output Voltage

Within the TPS6300x family, there are fixed and adjustable output voltage versions available. To properly configure the fixed output voltage devices, the FB pin is used to sense the output voltage. This means that it must be connected directly to VOUT. At the adjustable output voltage versions, an external resistor divider is used to adjust the output voltage. The resistor divider must be connected between VOUT, FB and GND. When the output voltage is regulated properly, the typical value of the voltage at the FB pin is 500 mV. The maximum recommended value for the output voltage is 5.5 V. The current through the resistive divider should be about 100 times greater than the current into the FB pin. The typical current into the FB pin is 0.01 µA, and the voltage across the resistor between FB and GND, R2 , is typically 500 mV. Based on those two values, the recommended value for R2 should be lower than 500 kΩ, in order to set the divider current at 1 µA or higher. TI recommends to keep the value for this resistor in the range of 200 kΩ. From that, the value of the resistor connected between VOUT and FB, R1 , depending on the needed output voltage (VOUT), can be calculated using Equation 1. (1) If as an example, an output voltage of 3.3 V is needed, a 1-MΩ resistor should be chosen for R1 . To improve control performance using a feedforward capacitor in parallel to R1 is recommended. The value for the feedforward capacitor can be calculated using Equation 2.

R2 calculation

$$R_1 = R_2 \times \left(\frac{V_{OUT}}{V_{FB}} - 1 \right)$$

Wanted Vout is 3v3.

R1 = 1M (according to datasheet 8.2.2.1)

R2 = ?

Vout = 3.3V

Vfb = 0.5V (typical value from datasheet 8.2.2.1)

Solving for R2:

R2 = (Vfb * R1) / (Vout - Vfb)

R2 = (0.5 * 1000000) / (3.3 – 0.5) = 178571,4285714286 ohms.

Solved for Vout (to make sure I did not make a solving mistake):

Vout = (Vfb * (R1 + R2)) / R2

Vout = (0.5 * (1000000 + 178000)) / 178000 = 3,308988764044944 = 3.3v.

So chosen R2 = 178k.

Cff calculation

$$C_{ff} = \frac{2.2\ \mu s}{R_1}$$

2.2us = 0.0000022 s

R1 = 1000000 ohm

Cff = 2.2us / R1 = 0.0000022 / 1000000 = 0.0000000000022 F

0.0000000000022 F = 2.2 pF

So Cff should be 2.2pF

Vbat pin on MCU

I'd like to connect the Vbat pin on the MCU, to the Li-Ion battery. This way I don't have to put an extra coin-cell in, and I can still use the RTC functionality on the MCU.

The datasheet mentions the following important information on the RTC and backup registers (2.3.16, but I've marked important keywords here):

"+/- 2.2 uA power usage at 3v3 (see figure 16 of datasheet)

VBAT Backup operating voltage min: 1.8V and max 3.6V.

The RTC and the backup registers are supplied through a switch that takes power either on VDD supply when present or through the VBAT pin. The backup registers are forty-two 16-bit registers used to store 84 bytes of user application data when VDD power is not present. They are not reset by a system or power reset, and they are not reset when the device wakes up from the Standby mode. The real-time clock provides a set of continuously running counters which can be used with suitable software to provide a clock calendar function, and provides an alarm interrupt and a periodic interrupt. It is clocked by a 32.768 kHz external crystal, resonator or oscillator, the internal low-power RC oscillator or the high-speed external clock divided by 128. The internal low-speed RC has a typical frequency of 40 kHz. The RTC can be calibrated using an external 512 Hz output to compensate for any natural quartz deviation. The RTC features a 32-bit programmable counter for long term measurement using the Compare register to generate an alarm. A 20-bit prescaler is used for the time base clock and is by default configured to generate a time base of 1 second from a clock at 32.768 kHz."

So:

1. No higher than 3.6V (Li-Po can get up to 4.2V, so we need a lower voltage).

2. Low power consumption, which is nice. LDO would be good option

3. 84 bytes of user data is more than enough. This means no extra EEPROM chip is needed.

4. Can be clocked by internal clock, or High-Speed External. Since I see no use for a really precise, non-drifting clock, this means that I can omit the 32.768kHz crystal from the design.

Only problem, is the voltage.

A voltage division would probably not work here, since the voltage would keep changing when the battery drains. This would probably give weird clock issues (I don't need a really stable clock, but a sort-off stable clock would be nice).

So next best option would probably to use a small LDO. The RTC pin does not use to much current, so the inefficiency does not matter to much.

LD39015M33R is chosen, since this has input range of 1.8 to 5.5v and a stable output of 3v3. Very low quiescent current (18 µA typ. at no load, 38 µA typ. at 150 mA load, 1 µA max. in OFF mode) and also 150mA output current, which should be way more than enough.

Momentary buttons and hardware debouncing

I want 3 digital input momentary buttons, with hardware debouncing integrated into the PCB.

Schmitt Trigger IC chosen for debouncing: STM m74hc14yrm13tr.

Because the internet said this IC is great for button removing hysterisis at 3v3 levels since it is a really fast schmitt trigger IC. The IC doesn't need additional configuration. Also the IC is ok in terms of availability.

https://www.st.com/resource/en/datasheet/m74hc14.pdf

I've used 'snapeda.com' to create a schematic symbol (footprint was already available). Quite useful stuff: https://www.snapeda.com/parts/M74HC14YRM13TR/STMicroelectronics/view-part/

Buzzer

To make sure the buzzer does not pull to much current from the MCU, an NPN transistor is used:

Also a flyback diode is placed between buzzer pins, to act as a bypass if the magnetic field collapses, to not damage the transistor.

A current limiting resistor of 1k ohm is used for the base pin of the transistor.

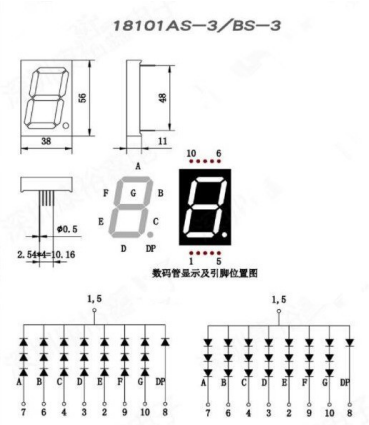
Then 3v3 is connected to GND, via the transistor.

Connectors to seven (eight) segment displays

The used segment displays are common cathode: https://t.ly/1rQHq (short URL, because AliExpress creates huge URLs).

There are 8 segments (7 for the digit, 1 for a dot), and two ground pins per display:

PIN 1 = GROUND
PIN 5 = GROUND
PIN 2 = E
PIN 3 = D
PIN 4 = C
PIN 6 = B
PIN 7 = A
PIN 8 = DP
PIN 9 = F
PIN 10 = G



The displays will be mounted on seperate PCB's for easy product assembly (the displays have pins, so the PCB's will have header sockets). Then, each segment PCB will be connected to the main shotclock PCB, also using header pins/sockets.

To minimize noise, low impedance paths and reduces the chances of ground bounce or voltage fluctuations, I'll try to add as much ground pins between the two PCB's as possible.

Each segment PCB will mount two segment displays (each display should be able to show a different digit). Each shotclock will have 4 segment PCB's mounted in 90° offset rotation from main PCB center. These all show the same 2 sets of digits (hense, 16 signals needed in total). The transistors will be mounted on the segment display PCB, not on the main PCB. Main PCB will only provide power, GND & PWM signals.

Since there are 16 (PWM) signal lines needed between the boards, I'll try and fit 16 ground pins, together with 16 signal pins, and 1 3v3 pin between each PCB. This comes to a total of 33 connections between the boards.

When using 2.54mm pitch header pins/sockets, the physical space needed for each connector is: 2.54 * 33 = 83,82mm, which should easily fit in a one row setup. The chosen segment displays are 40mm wide per display, so the product must be and will be atleast 100mm wide.

If 1x33 header pins/sockets are difficult to source, I could also use 2 rows, which would mean they would be 83,82mm/2 = 41,91mm wide. For now, I'll just use one row of 33 pins.

Thoughts and questions

1. Does the LDO for the Vbat pin and batt+ connection make sense? Or should I just use voltage division or a switching regulator for this? (high clock accuracy is not needed, just a clock, and a way to power backup registers when 3v3 is off)

2. Is it smart to add reverse polarity protections, or ESD protections? If so, where?

3. Any handy places to put a test point, that I have forgotten?

4. For the ADC voltage divider, what stops me from increasing resistance and buffer capacitor even more, to get less current leakage?

5. Is it a good idea to create a hard shut-off for the li-ion when it is for example < 3.0V? If so, what is a good way to do so? (the charge IC TP4056 does not have such functionality built-in).

6. Should my jumper system, to shut off the TEMP sensing of the TP4056 work? By just pulling the ‘IC_TEMP’ label to GND, the whole ‘BATT_NTC’ label, and that voltage divider will effectivly be ‘disabled’ right? Or am I also pulling the ‘Vbus’ to GND in that case?

7. Is one ground pin per PWM a bit overkill? Or does this sound about right?