



ТЕХНИЧЕСКИ УНИВЕРСИТЕТ-СОФИЯ

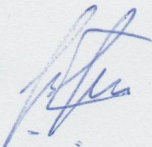


ФАКУЛТЕТ ПО ЕЛЕКТРОННА ТЕХНИКА И ТЕХНОЛОГИИ

ДИПЛОМНА РАБОТА

**ТЕМА: АВТОНОМНА СТАНЦИЯ ЗА
СЛЕДЕНЕ НИВОТО НА РЕКИ
И ВОДОЕМИ В РЕАЛНО ВРЕМЕ**

ДИПЛОМАНТ: Христо Руменов Митрев 

РЪКОВОДИТЕЛ: д-р инж. Любомир Богданов 

София, юли 2021



ТЕХНИЧЕСКИ УНИВЕРСИТЕТ – СОФИЯ



ФАКУЛТЕТ ПО ЕЛЕКТРОННА ТЕХНИКА И ТЕХНОЛОГИИ

КАТЕДРА: **Електронна техника**

Дата на задаване: **25.05.2021 г.**

Дата на предаване: **17.07.2021 г.**

Утвърждавам:

Декан:

(проф. д.т.н. инж. И. Илиев)



ЗАДАНИЕ за дипломна работа

На студента: Христо Руменов Митрев

ф. № 101217010

Образователно-квалификационна степен: бакалавър

Специалност: Електроника

1. **Тема:** Автономна станция за следене нивото на реки и водоеми в реално време.

2. **Заявител на темата:** фирма Некст Лаб ЕООД

(катедра, външна организация)

3. **Изходни данни:**

1. Микроконтролер – ESP32
2. Ултразвуков датчик за разстояние
3. Температурен датчик
4. Датчик за атмосферно налягане
5. Соларен панел за захранване

4. **Обяснителна записка:**

- 4.1 Литературно проучване
- 4.2 Съставяне на блокова схема
- 4.3 Проектиране на принципна схема
- 4.4 Алгоритъм на управляващата програма (и/или сорс код)
- 4.5 Експериментални резултати

5. **Графична част:**

- 5.1 Блокова и принципна схема
- 5.2 Алгоритъм на управляващата програма

Консултант:

(дтн Николай Илиев)

Ръководител:

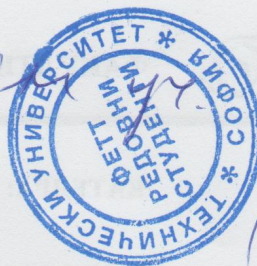
(гл. ас. д-р инж. Любомир Богданов)

Ръководител катедра

“Електронна техника”:

(доц. д-р инж. Петър Якимов)

Зав



20/21

Р

ЗАДАНИЕ

Дипломантът е извършил заданието!
Предлагам да бъде допуснат до защита!

и.ас.д-р инж. Мобомир Богданов

ДЕКЛАРАЦИЯ

Долуподписаният: Христо Бученов Митрев,
декларирам, че съм разработил самостоятелно представената работа. Не съм използвал източници или ресурси, освен посочените. Ясно съм обозначил използваните източници, които са цитирани буквално или по съдържание.

Декларирам и че работата не е представяна в рамките на друга дипломна защита.

Дата: 08.07.2021г.

Подпис: Христо Бученов Митрев

Съдържание

1	Увод	1
2	Измерване на физични величини	2
2.1	Измерване нивото на водоеми	2
2.1.1	MEMS датчици за налягане	2
2.1.2	Ултразвукови датчици за разстояние	4
2.2	Температура на въздуха и атмосферно налягане	5
2.3	Измерване на дъжд	6
3	Заснемане на наблюдавания обект	7
3.1	Цифрова камера със сериен интерфейс	9
4	Технически особености	9
5	Връзка към централен сървър	10
5.1	LoRa	10
5.2	GPRS	11
5.3	Интернет протокол за изпращане на данни	11
6	Хардуер	16
6.1	Принципна схема	16
6.2	Дизайн на платката	19
7	Захранване от соларен панел	20
7.1	Хардуер на захранването	20
7.2	Дизайн на платката	21
7.3	Принципна схема	22
7.4	Оптимизиране на ключовия преобразувател	25
7.5	Усвояване на максимална мощност от соларния панел (MPPT)	26
7.6	Консумация на захранващата платка	26
7.7	Комуникация между станцията и захранването	27
7.8	Обновяване фърмуера на захранването	28

8	Фърмуер на станцията	29
8.1	Операционна система	31
8.2	Драйвер за GPRS модем	32
8.3	Модул за обновяване на фърмуера чрез GPRS	32
9	Трудни за откриване софтуерни и хардуерни проблеми	33
9.1	Отказ на ултразвуковия датчик през деня	33
9.2	Отказ на станцията по време на дъжд	34
9.3	Отказ на станцията след фърмуерно обновяване	35
10	Опитни резултати	36
10.1	Сглобяване на 30 екземпляра	36
10.2	Консумация на енергия по време на работа	38
10.3	Взаимодействие с крайния потребител	39
11	Специални благодарности	42
11.1	Проектиране и опроводяване на платки	42
11.2	Фърмуерни модули	42
11.3	Графичен дизайн на блокови схеми	42
11.4	Отстраняване на правописни грешки	42
11.5	Разработване на bluetooth светкавица	43
11.6	Разработване на устройство за отдалечено откриване на грешки	43
11.7	Производство и инсталиране на станциите	43
11.8	Морални и технически насоки	43
12	Анотация	44

Фигури

1	Блокова схема на LPS33W	6
2	Датчик за дъжд	7
3	Команди, поддържани от VC0706	8
4	Блокова схема на станцията	17
5	Платка на станцията	19
6	Блокова схема на захранването	21
7	Платка на захранването	22
8	Блокова схема на STM32L011F4	23
9	Блокова схема на ESP32	30
10	Сглобена станция	37
11	Партида от 10 станции	37
12	Консумация при качване на един пост	38
13	Местоположение на станциите	39
14	Графика на параметрите	40
15	Снимки от станциите	41

Абревиатури

GPRS General Packet Radio Service

1 Увод

Тази дипломна работа има за цел да опише техническите предизвикателства и конкретните решения, взети при процеса на разработка на IoT устройство, чиято основна цел е да следи в реално време нивото на реки и водоеми и да известява при опасност от наводнения. В процеса на разработка, както в повечето реални ситуации, заданието постепенно бе разширявано и в момента на документиране, станцията има възможност за измерване на количество паднали дъждове, атмосферно налягане, температура на въздуха, температура на водата (при използване на потопяем воден датчик) и освен регистриране на метеорологични параметри, може да заснеме с ниска резолюция състоянието на обекта на измерване.

Освен функционалните задания, има и технически изисквания, като основното е станцията да е абсолютно автономна (захранвана от соларен панел) и да може да издържа дълго време в неблагоприятни метеорологични условия (натрупан сняг върху соларния панел и липса на слънчева светлина), без да нарушава функционалността си,[10] и без да се налага външна намеса.

Още от началото на проекта, трябваше да се направят стратегически компромиси, свързани с цена, обем на електрониката, функционалности и т.н. След множество спорове, в които страните винаги бяха едни и същи - инженерна и мениджмънт, станциите бяха постепенно (а понякога и внезапно) усъвършенствани благодарение на множество малки оптимизации или големи пробиви при дизайна и реализацията.

Най-важното постижение на проекта е успеха при използване на максимума от предоставените ресурси, но освен това идентифицирането на компонентите, при които не трябва да се правят компромиси с техническите параметри (пример за това е автономното захранване).

2 Измерване на физични величини

2.1 Измерване нивото на водоеми

За да има смисъл от станцията и тя да е в състояние да известява при опасност от наводнения, тя трябва да може да измерва водното ниво. Датчикът за водно ниво трябва да е надежден и точен. Нашето изискване е грешката му да не надвишава 1% от пълния мащаб и да работи в температурния обхват от -20°C ¹ до 60°C . Тези две изисквания са безкомпромисни, но освен тях има и придружаващи - датчикът за измерване на водното ниво да не усложнява монтажа на самата станиця и да не консумира голяма мощност по време на работа. Нашият сензор консумира 7 mA и трябва да работи 3 секунди, през които прави 30 измервания.

2.1.1 MEMS датчици за налягане

Едно възможно решение на проблема е да се използва потопяемо устройство със сензор за абсолютно налягане. В днешно време MEMS сензорите са силно развити - прецизни, евтини, с малък корпус и нищожна консумация на енергия, устойчиви на агресивни химични среди и способни да измерват големи диапазони на налягане.

Един от проблемите на тези датчици, който е лесно решим, в някои случаи още от самия производител, е калибрацията и компенсацията на температурната зависимост. Някои производители предлагат готови MEMS датчици с цифров интерфейс, като директно в самия сензор е интегрирано АЦП и има температурна компенсация. Тези датчици са сравнително лесни за използване. Единственото изискване е да се осигури стабилно хранване, за да не се компрометира точността на вграденото АЦП. Едни от производителите на подобни датчици са ST и Bosch. Този клас датчици за налягане, обаче е сравнително скъп.

Друго възможно решение на проблема е използването на чисто аналогов MEMS датчик, който се състои само от мост на Уитстон, съставен

¹В случай, че датчикът е потопяем, се подразбира, че при отрицателни температури, в повечето случаи, водата ще е замръзнала и следователно нивото няма да се променя.

от 4 чувствителни на налягане резистивни структури - две с положителна зависимост и две с отрицателна. Въпросният клас от сензори отново трябва да бъде захранван от стабилно напрежение, но освен това извън самия датчик трябва да се проектира диференциален усилвател с постояннотоково отместване на изхода¹, АЦП и сензор за температурна компенсация. Тези датчици обикновено не са калибрирани при производствения процес и това трябва да стане по време на сглобяване на крайното изделие. За да се осъществи температурната компенсация, много близко до корпуса на датчика се поставя температурен сензор, като по време на дизайна трябва да се внимава двата сензора да са температурно свързани (thermally coupled) колкото се може по-добре. При калибриране на датчика се измерва и температурният коефициент необходим за компенсация, като крайната формула за изчисляване на налягането е:

$$P = P' + T \times K + C \quad (1)$$

P е налягането след компенсация, P' е некомпенсираното налягане, T е температурата, K е коефициента на температурна зависимост, а C е коефициента на отместване. K и C са специфични в зависимост от мерните единици на T и P' .

При използване на аналогов MEMS сензор, трябва да се обърне специално внимание на филтрирането, мащабирането и буферирането на полезния сигнал. Освен това, могат да се приложат и допълнителни софтуерни обработки. Най-ефективната от тях е използване на наддискретизация (oversampling)[5], като така се подобрява динамичният обхват² и броя на ефективните разряди.

При използване на какъвто и да е потопяем датчик за абсолютно налягане, трябва да се обърне внимание, че налягането зависи както от нивото на водния стълб над датчика, така и от атмосферното налягане, което може да се изменя с десетки хектопаскали в зависимост от метеорологичното време. Също атмосферното налягане зависи от надморската височина. Един сантиметър воден стълб грубо съответства

¹Постояннотоковото отместване се използва, за да се компенсира грешката на резистивните структури.

²Динамичният обхват е съотношението между броя на разрядите и нивото на шума при измерване.

на налягане от един хектопаскал. Дори и да коригираме с постоянно отместване надморската височина, при промяна на атмосферното налягане в следствие от промяна на метеорологичното време, грешката при измерване на водния стълб може да достигне десетки сантиметри. За да отстраним този проблем, извън водата трябва да поставим отделен сензор за абсолютно налягане и да извадим неговото показание от това на потопяемия датчик.

2.1.2 Ултразвукови датчици за разстояние

При потопяемите датчици за налягане има проблем, който не е очевиден по време на разработка, но става доста сериозен по време на инсталиране. Те трябва да са стабилно закрепени за дъното на водоема, а от тях до станцията да се прекара и обезопаси дълъг кабел. Самата станция трябва да е монтирана на високо, тъй като разчита на слънчевата светлина за източник на енергия.

Този проблем може да се отстрани напълно, ако вместо потопяем датчик за налягане в самата станция се монтира ултразвуков датчик за разстояние. При този метод за отчитане на водното ниво се появяват други недостатъци, но се оказва, че е по-добре да се откажем от всички предимства на потопяемите датчици с цел да спестим усилия и ресурси при монтажа, особено ако става въпрос за повече от няколко станции.

Ултразвуковите сензори за разстояние имат предавателна част, която генерира кратък звуков импулс с честота над 20 KHz (обикновено 40 KHz), който се разпространява насочено със скоростта на звука във въздух до обекта на измерване. След като достигне обекта на измерване, който трябва да е с много по-голям акустичен импеданс от въздуха, почти целият сигнал се отразява и тръгва в обратна посока към ултразвуковия сензор. При достигане до сензора, се поема обратно от приемната част и се регистрира времето за отиване и връщане от обекта на измерване. Тъй като скоростта на звука във въздух е сравнително стабилна величина, времето за отиване и връщане на импулса може лесно да се преобразува в разстояние.

$$D = \frac{t \times V}{2} \quad (2)$$

D е разстоянието до обекта, t е времето за отиване и връщане на импулса, а V е скоростта на звука.

Този на пръв поглед лесен принцип за измерване на разстояние носи със себе си не малко проблеми и за успешното му прилагане трябва да се вземат съображения. Също е много трудно, а в някои случаи дори и невъзможно да се постигне точност на измерване близка до потопяемите датчици за налягане. При измерване в идеални условия, точността, която може да се постигне, обикновено е една дължина на вълната на ултразвуковия импулс. При честота 40 KHz и приблизителна скорост 344 m/s на звука във въздух¹, дължината на вълната на звука е малко под 1 сантиметър. За жалост към минималната грешка от 1 сантиметър се натрупват и поредица грешки, които са свързани със зависимостта на скоростта на звука от поредица параметри (температура, налягане, влажност), като най-критичният параметър се оказва температурата на въздуха:

$$v = 331 \text{ m/s} + 0.6 \frac{\text{m/s}}{^{\circ}\text{C}} \times T \quad [7] \quad (3)$$

Грешката, зависеща от температурата на въздуха, може да се компенсира, но това води до проблем с измерване температурата на въздуха при пряко облъчване на станцията от слънчева светлина.

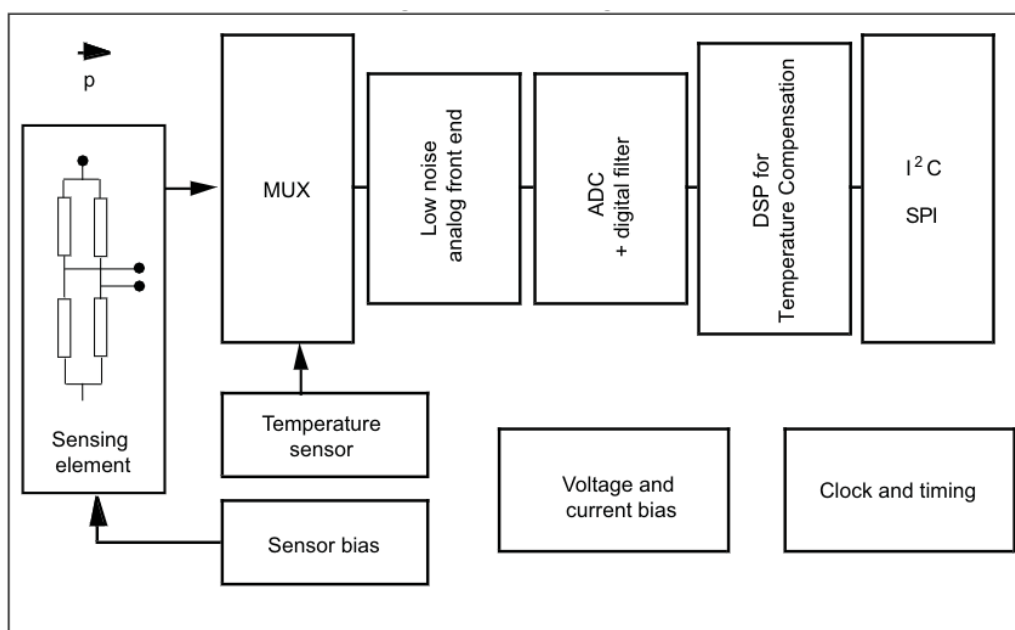
2.2 Температура на въздуха и атмосферно налягане

Измерването на температурата на въздуха може да се окаже проблемно, ако няма лесен начин да подсигуриим, че температурният ни датчик ще е винаги на сянка. Тъй като станциите са разположени на открито, е почти сигурно, че слънцето ще влияе директно върху незащитен датчик за температура. В този случай най-добрият подход е да се използва екран на Стивънсън. Поради съображения за цена и обем, на нашата

¹тази скорост е измерена при температура 20°C, атмосферно налягане от 101,325kPa и относителна влажност 50%.

станция не се използва подобна конструкция, но е положено старание да се оптимизира позицията на датчика.

Повечето прецизни цифрови MEMS датчици за измерване на атмосферно налягане включват и температурен датчик за температурната им компенсация, както бе обяснено в секция 2.1.1, което прави логично да се използва един и същ датчик за атмосферно налягане и температура на въздуха. В нашия случай бе избран LPS33W[4], тъй като е малък, маломощен, устойчив и лесен за интеграция. За подобни датчици възможността за взаимодействие с тях посредством I²C е голямо предимство при ограничен брой изводи на избрания микроконтролер.

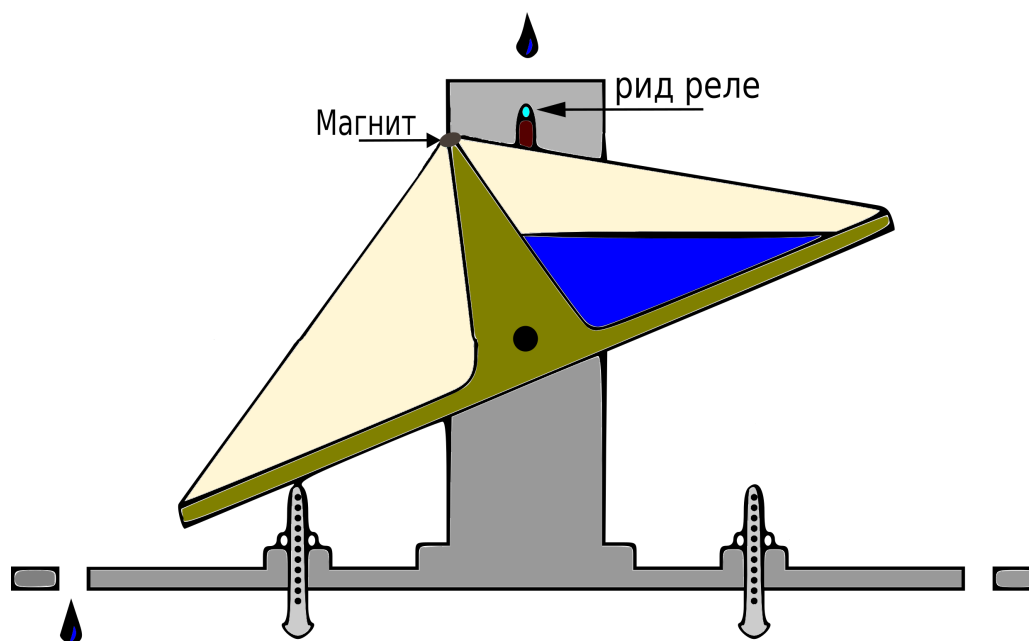


Фигура 1: Блокова схема на LPS33W

2.3 Измерване на дъжд

Дъждът е един от основните фактори влияещи на нивото на реките и водоемите. Ако имаме информация за него в реално време, ще можем да известим за потенциална опасност още преди тя да е настъпила. Датчика за дъжд е сравнително интересен компонент, тъй като няма как количествено да се окачестви падналият дъжд, без да има механич-

но взаимодействие с него. Механичното взаимодействие представлява поток на част от дъждовната вода през датчика, което създава предпоставка за много механични проблеми. За осигуряване на максимална надежност на подобен датчик трябва да се намалят до минимум движещите се компоненти и да се осигури път без препятствия за потока на течност. Датчикът, който се използва в станцията, е базиран на рид реле и една единствена подвижна "кобилица".



Фигура 2: Датчик за дъжд

3 Заснемане на наблюдавания обект

Станцията има възможност да се конфигурира да измерва нивото на водата на всяка минута. При тази конфигурация се извършват над 1400 измервания в денонощие. При надеждност на датчика за измерване на нивото от 99.9%, можем да очакваме 3 грешни измервания на всеки два дена. Тук е важно да се отбележи, че обикновено грешки се появяват заради израстнала растителност непосредствено до обекта на измерване. За да се потвърди истинността на измерванията, станцията заснема

обекта на наблюдение през конфигурируем интервал от време.

Command Definition	Command Byte	Instructions
GEN_VERSION	0x11	Get Firmware version information
SET_SERIAL_NUMBER	0x21	Set serial number
SET_PORT	0x24	Set port
SYSTEM_RESET	0x26	System reset
READ_DATA	0x30	Read data register
WRITE_DATA	0x31	Write data register
READ_FBUF	0x32	Read buffer register
WRITE_FBUF	0x33	Write buffer register
GET_FBUF_LEN	0x34	Get image lengths in frame buffer
SET_FBUF_LEN	0x35	Set image lengths in frame buffer
FBUF_CTRL	0x36	Control frame buffer register
COMM_MOTION_CTRL	0x37	Motion detect on or off in communication interface
COMM_MOTION_STATUS	0x38	Get motion monitoring status in communication interface
COMM_MOTION_DETECTED	0x39	Motion has been detected by communication interface
MIRROR_CTRL	0x3A	Mirror control
MIRROR_STATUS	0x3B	Mirror status
COLOR_CTRL	0x3C	Control color
COLOR_STATUS	0x3D	Color status
POWER_SAVE_CTRL	0x3E	Power mode control
POWER_SAVE_STATUS	0x3F	Power save mode or not
AE_CTRL	0x40	Control AE
AE_STATUS	0x41	AE status
MOTION_CTRL	0x42	Motion control
MOTION_STATUS	0x43	Get motion status
TV_OUT_CTRL	0x44	TV output on or off control
OSD_ADD_CHAR	0x45	Add characters to OSD channels
DOWNSIZE_CTRL	0x54	Downsize Control
DOWNSIZE_STATUS	0x55	Downsize status
GET_FLASH_SIZE	0x60	Get SPI flash size
ERASE_FLASH_SECTOR	0x61	Erase one block of the flash
ERASE_FLASH_ALL	0x62	Erase the whole flash
READ_LOGO	0x70	Read and show logo
SET_BITMAP	0x71	Bitmap operation
BATCH_WRITE	0x80	Write mass data at a time

Фигура 3: Команди, поддържани от VC0706

3.1 Цифрова камера със сериен интерфейс

За камера бе избрана VC0706. Предимството на тази камера е, че не използва масовия Serial Camera Control Bus интерфейс, който изисква използването на поне 10 извода за вход и изход. Вместо SCCB, VC0706 използва UART и се нуждае само от два извода за вход и изход. Комуникацията с камерата също е опростена, като се свежда до изпращане на команди в предефиниран формат и съответно получаване на предефинирани отговори[8].

4 Технически особености

Камерата, GPRS модема, bootloader-а на захранването и два от датчиците, които използваме, поддържат единствено SCI. Този интерфейс е предпочитан, защото е много лесен за използване. Също цифровите интерфейси са предпочитани пред аналоговите заради тяхната шумова устойчивост и защото улесняват схемотехниката (не се нуждаем от аналогова част, която да буферира, филтрира или мащабира сигналите). Колкото по-рано преобразуваме аналоговите величини в цифрови, толкова по-добре. В нашия случай всички аналогови величини се преобразуват в цифрови още в самите датчици.

Проблемът при използването предимно на SCI интерфейс е, че в нашия контролер разполагаме само с 3 хардуерни модула поддържащи UART. Един от тях е зает за програмиране и извеждане на информация за отстраняване на грешки. Още един от тях трябва постоянно да се задели за управление на GPRS модема. Това ни оставя с един свободен модул и поне 3 устройства, които разчитат на него.

Този проблем е лесно решим ако извън контролера поставим допълнителна схемотехника за преобразуване на SCI към друг интерфейс. Това обаче противоречи на първоначалния ни избор на цифрови вместо аналогови датчици с цел опростяване на схемотехниката.

Тъй като не се налага никоя двойка от устройствата да се обслужва едновременно, би било най-ефективно да се използва един хардуерен модул, който да се преконфигурира преди обслужването на всяко от уст-

ройствата. Допълнително сме улеснени от възможността UART модула да използва всеки един от изводите на нашия контролер.

За да се улесни софтуерната разработка на драйверите, базирани на SCI, е реализиран централен софтуерен модул, който инициализира вътрешната структура и позволява смяна на контрол между отделните драйвери с минимална преинициализация на периферията.

След инициализиране на UART модула, всеки драйвер който трябва да го ползва, сменя изводите свързани към него, честотата на предаване на информация и пререгистрира своята индивидуална подпроцедура за приемане на данни.

5 Връзка към централен сървър

Прецизното измерване на водното ниво на реките е половината от задачата на станцията. Другата половина е предаването на резултатите в реално време. За тази цел, на нашата станция трябва да се осигури връзка към централен сървър. Това може да стане чрез достъп на станциите до интернет или чрез алтернативни методи за връзка (LoRa, Sigfox).

5.1 LoRa

LoRa(Long Range) е частен стандарт, дефиниращ физическия слой за безжично предаване на информация на дълги разстояния с ниска консумация. Системи базирани на LoRa спецификациите се използват за безжично отчитане на водомери и други подобни функции, които нямат необходимост от предаване на големи количества данни през кратки интервали. LoRaWAN е придружаващия стандарт, който дефинира протоколите за предаване на информация. За жалост тези стандарти не позволяват предаване на информация с висока скорост или през малки интервали от време¹. Освен ограниченията от протокола, в България няма добре развита инфраструктура за подобни устройства.

¹колкото повече устройства използват една мрежа, толкова по-рядко индивидуалните устройства имат право да предават информация

Нашата станция, освен да отчита нивото на водоеми, трябва да изпраща и снимки. Средният обем на една снимка е 100 kB. Това прави невъзможно използването на протокол от рода на LoRa за предаване на толкова количество информация.

5.2 GPRS

General Packet Radio Service (GPRS) е стандарт, надграждащ 2G и 3G, поддържан от консорциума 3GPP. GPRS мрежите са широко разпространени из целия свят и позволяват директен достъп до интернет и предаване на големи количества информация. Директната интернет връзка елиминира необходимостта от междинно устройство-посредник.

Освен това, при използване на GPRS модем, цялата свързаност ще е базирана на AT команди между модема и главния микроконтролер. Тъй като AT командите са стандартизирани, станцията става независима от GPRS модема и в бъдеще той може много лесно да бъде заменен с друг.

5.3 Интернет протокол за изпращане на данни

За да се качат данните към сървър, бе избран HTTP. Този протокол е сравнително неефективен, защото към всеки пост към и отговор от сървър се добавят около 300 байта контролна информация. Това не е малко в сравнение със средния размер на постове от около 600 байта.

Предимството на този подход е, че улеснява разработчиците на сървър, тъй като могат директно да използват сървърната програма на Apache и не се налага да боравят с друг програмен език освен PHP. Тук няма да се съсредоточаваме върху сървърната част, тъй като е извън нашата област на познания.

Данните, които се изпращат чрез HTTP са структурирани по стандарта JSON. Това също добавя ненужен обем към полезната информация, но прави данните човешко-четими.

Друга алтернатива на JSON е XML. В началото станциите използваха XML, но при него контролната информация, която се добавя, е още повече. Причината е че всеки XML таг огражда полезната информация,

използвайки се два пъти - в началото и в края¹. Този проблем е решен при JSON.

Тъй като сървърът е публично достъпен, към формата на данните има добавено допълнително поле, в което се записва поредица от 8 символа², уникални за всеки пост. Поредицата се изчислява с алгоритъм за шифриране, като на него се подават полета, които са задължителни да присъстват във всеки пост - дата, час и име на устройство. Това е задължителна мярка за защита от злонамерени атаки към сървъра. По този начин постове могат да качват само тези, които знаят алгоритъма за шифриране.

¹ полето **"water_level" : 63.00**, при използване на XML ще стане: **<water_level>63.00</water_level>**.

² Тези 8 символа кодират 4 байта полезна информация, представена в шестнасетичен вид. Това означава че за да се пробие бариерата с груба сила, среднестатистически ще трябва над 2 милиарда опита за всеки отделен пост.

Ето така изглежда един примерен пост към нашия сървър:

```
POST /xxx/xx/xxxx HTTP/1.1
Host: xxx.xxxxxxxxxx.xxx:xx
Connection: close
User-Agent: A9_GPRS_Module
Content-Length: 903

[{"date" : "2021-01-21 16:39:06",
"devID" : "ACWA17",
"serial" : "8CAAB592F824",
"hash" : "d2f208d5",
"params" :
{
"water_level" : 63.00,
"distance" : 209.5,
"distance2" : 217.0,
"battery" : 4.144,
"temperature" : 0.46,
"ambient_press" : 950.10,
"current_version" : 6.13,
"gprs_signal_quality" : 28,
"percent_battery" : 84,
"rain_lm2" : 0.00,
"mm_rain_per_hour" : 0.000
},
"alarms" :
{
"water_level" : 1
},
"debug" : "---previous-post-ok!"
}]
```

Първите 5 реда са HTTP служебни данни, които указват домейна и адреса за изпращане на данните, версията на HTTP и размера на полезните данни. След тези служебни данни трябва да има един празен ред по стандарт.

След това следват полезните данни, структурирани по стандарта JSON. Всеки пост е задължително да съдържа дата, час, сериен номер на устройството и уникален хеш. След това следват параметрите. Тази конкретна станция е снабдена с два датчика за разстояние. Нивото на водата се изчислява косвено чрез едната дистанция - тя се изважда от предварително зададена константа (разстоянието от датчика до дъното на реката). Освен основните параметри, станцията изпраща също температура на въздуха, атмосферно налягане, качество на сигнала, натрупан дъжд и интензитет на дъжда. След параметрите, станцията може да изпрати сигнал за аларма когато даден параметър излезе извън предварително зададени граници. Накрая е добавена информация за намиране и отстраняване на грешки.

Когато сървърът получи успешно поста, непосредствено след това връща отговор към станцията. Отново първите 6 реда са HTTP служебни данни. След това се връща актуална дата и час, по която станцията си коригира натрупаната грешка от предходния пост. Това се прави защото за такт на вътрешния часовник за реално време (RTC) на станцията се използва RC тактов генератор.

Най-важната част от отговора е тази, в която се връщат параметри, конфигурирани от крайния потребител. Чрез тези параметри, той може да променя поведението на станцията.

Също се изпраща параметър, указващ най-новата версия на фърмуера, налична на сървъра (6.13 в този случай). При наличие на по-нова версия, станцията автоматично преминава в режим на обновяване, изтегля новата версия, проверява я за грешки при изтегляне, записва я в енергонезависимата памет и се рестартира.

Ето така изглежда един примерен отговор след успешен пост:

```
HTTP/1.1 200 OK
Date: Thu, 21 Jan 2021 14:41:53 GMT
Server: Apache/2.4.18 (Ubuntu)
Content-Length: 579
Connection: close
Content-Type: application/json; charset=utf-8

{
  "dateProcessed":"2021-01-21 16:41:53",
  "status":200,"bench":"15.299 ms \ / 0.322 sec.",
  "samplesProcessed":1,
  "data": {
    "devID":"ACWA17",
    "imgs":null,
    "paramsProcessed":11,
    "alarmsProcessed":1,
    "conf": {
      "refresh":"30",
      "battery_threshold":"50",
      "firmware_ver":"6.13",
      "WL_ALR":"50",
      "REFRESH_ALR":"15",
      "rain_tare":"6",
      "Message":"ACWA17 SMS in event of alarm",
      "Phone1":"0877XXXXXX",
      "Phone2":"0888XXXXXX",
      "Phone3":"0887XXXXXX",
    },
    "log":1
  },
  "message":"OK"
}
```


В този случай, потребителят е настроил поведението на станцията по следния начин:

- Станцията да качва нов пост през 30 минути ("refresh" : "30")
- След 6 часа без дъжд, да се нулира брояча на натрупаните валежи ("rain_tare" : "6")
- При ниво на реката над 50 сантиметра, станцията да премине в аварийен режим ("WL_ALR" : "50")
- При навлизане в аварийен режим, станцията да изпрати SMS на 3 телефонни номера ("Phone1" : "0877XXXXXX", ...)
- При навлизане в аварийен режим, станцията да започне да качва нов пост през 15 минути ("REFRESH_ALR" : "15")

6 Хардуер

6.1 Принципна схема

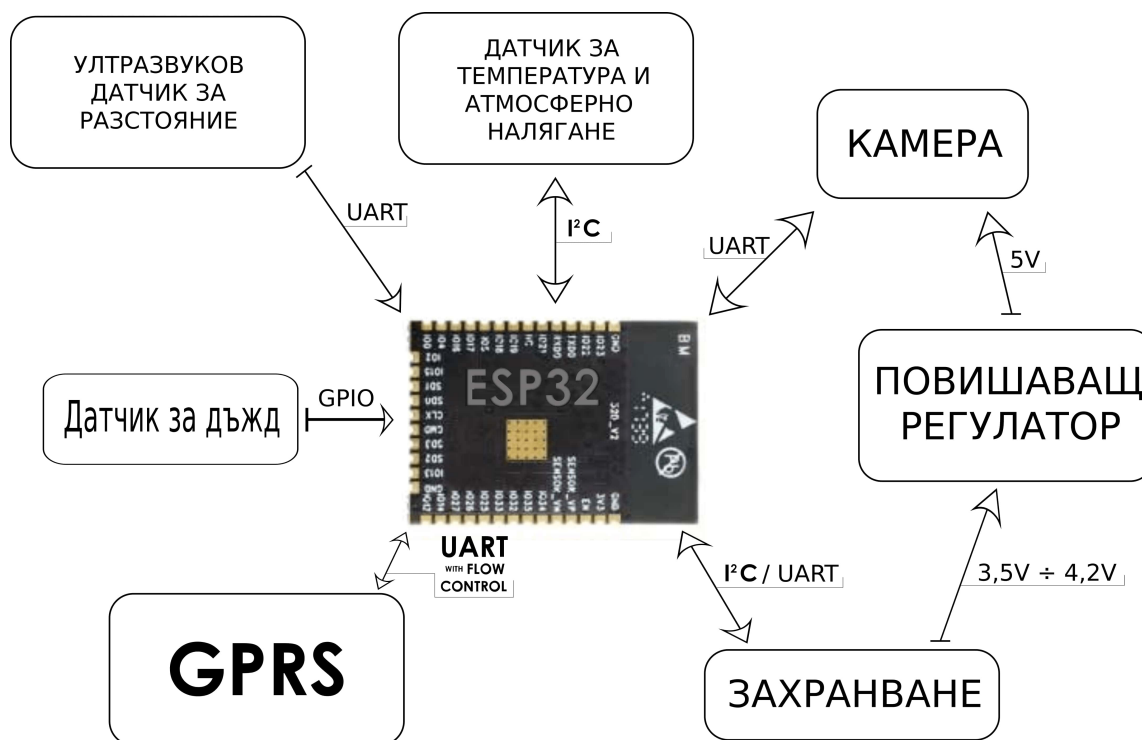
Схемата на станцията е оптимизирана за възможно най-ниска консумация. Захранването на всеки един датчик се управлява посредством транзисторни ключове.

Системата е проектирана да се захранва от една литиево-йонна клетка. Паралелно на захранващото напрежение са свързани последователно два суперкондензатора. Използват се в тази конфигурация, защото не съществуват суперкондензатори, които да поддържат напрежения над 2.7 V. Сумарното им номинално напрежение и капацитет са 5.4 V и 1.5 F. Специално бяха тествани кондензатори от няколко различни производителя и бяха избрани тези заради ниското им вътрешно съпротивление - сумарно 150 $m\Omega$. За да бъдат балансирани суперкондензаторите, паралелно на всеки от тях е поставен резистор. Токът през двата резистора, които са с еднаква стойност, трябва да е около 10 пъти по-голям от разликата в токовете на утечка на двата суперкондензатора.

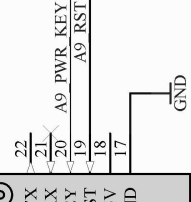
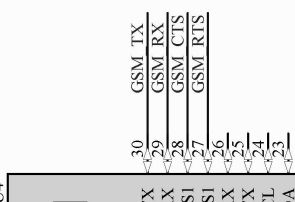
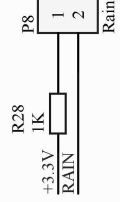
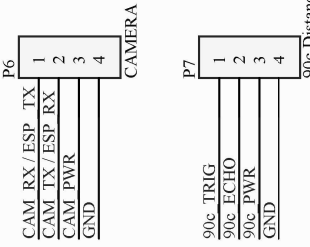
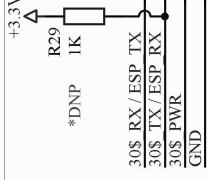
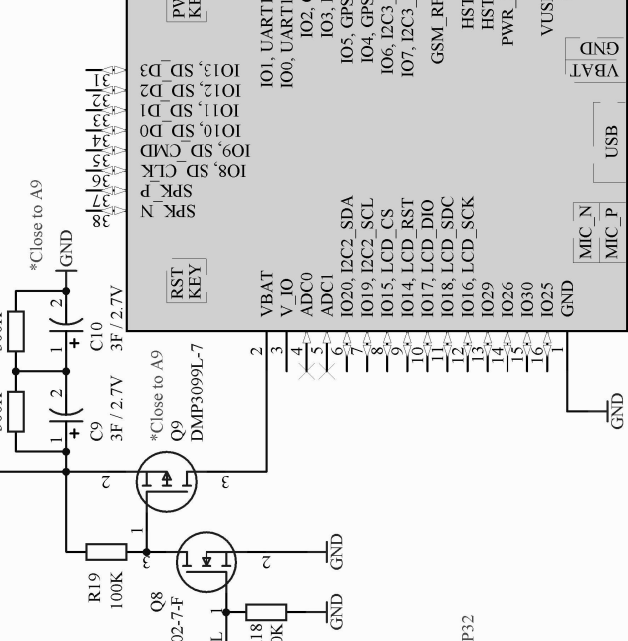
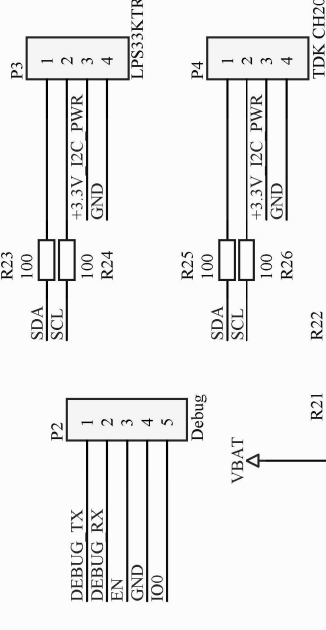
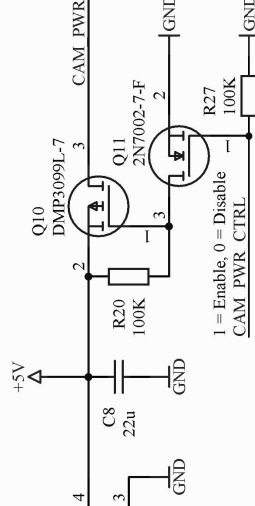
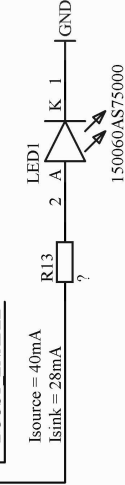
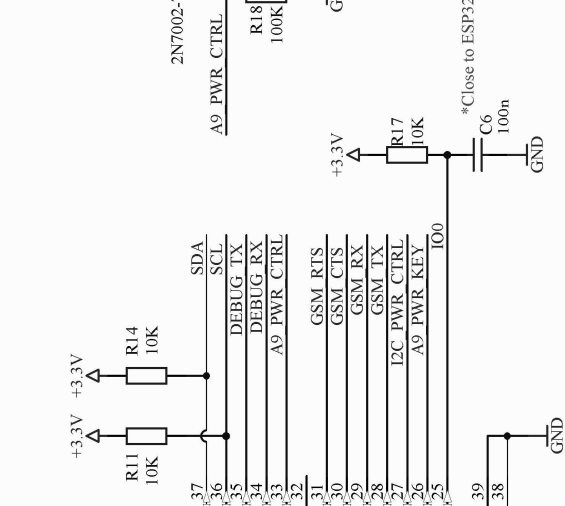
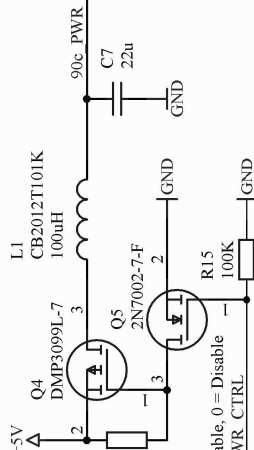
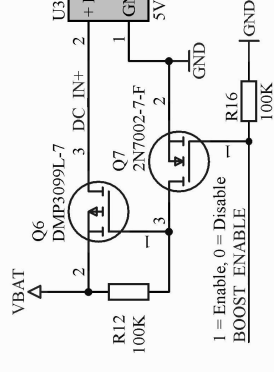
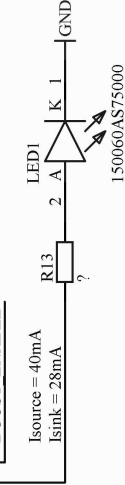
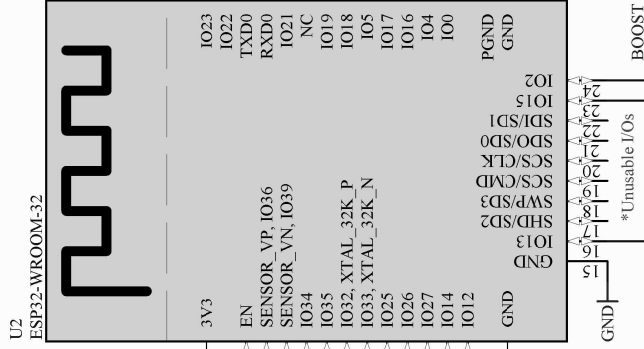
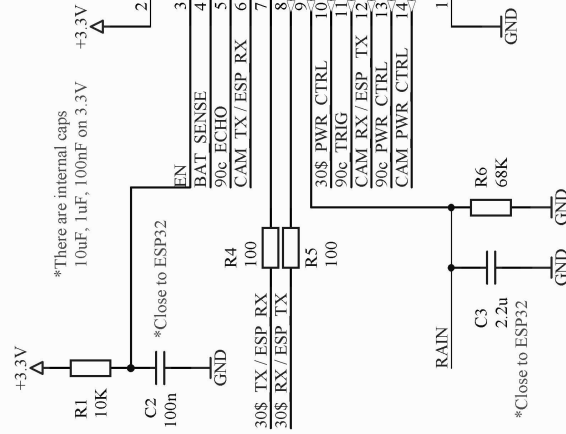
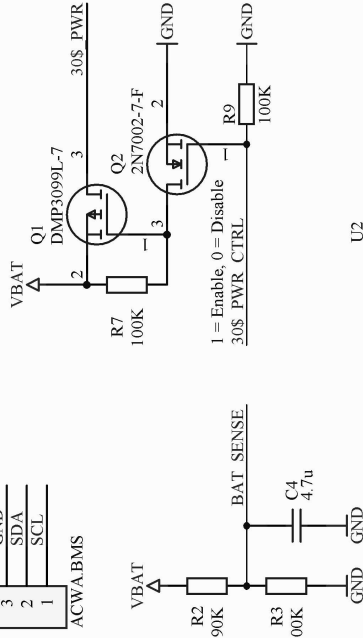
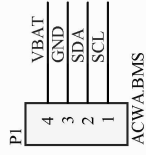
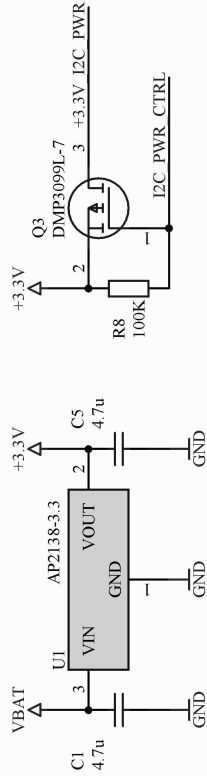
В повечето ситуации, станцията може да функционира и без суперкондензатори. Когато батерията е заредена и не се използва в среда с много ниски температури, тя може да осигури оптимално хранване и включването на големи консуматори като камерата и GPRS модемът не представлява проблем.

Нашата станция обаче трябва да може да работи при заряд на батерията, малко над минималния и *при температури под -20°C* . При толкова ниски температури, напрежението на батерията се понижава, а вътрешното съпротивление нараства. В такива ситуации суперкондензаторите допринасят за стабилността на системата.

Тъй като камерата и един от датчиците се нуждаят от 5 V за да функционират, трябва да се добави повишаващ ключов регулатор, който също се управлява с транзисторен ключ, за да не допринася към паразитната консумация на станцията, когато не се нуждаем от 5 V захранваща релса.



Фигура 4: Блокова схема на станцията



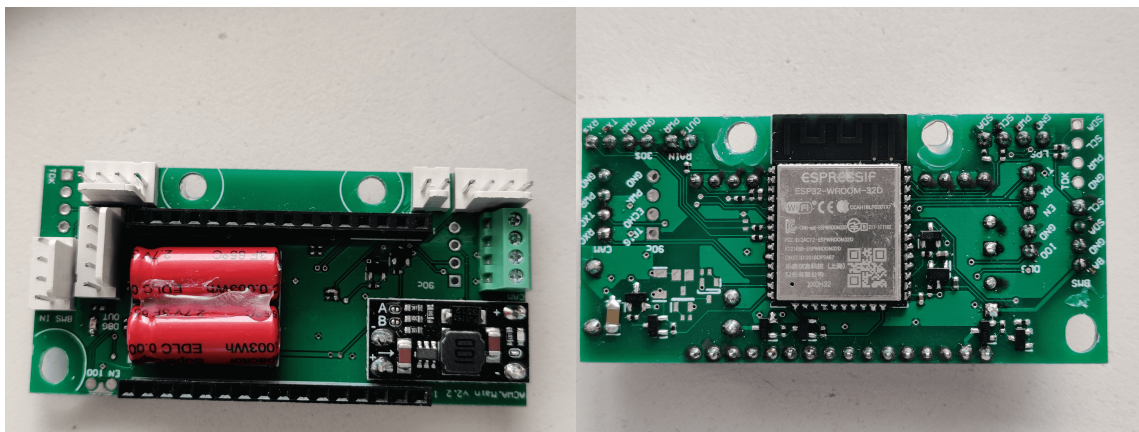
6.2 Дизайн на платката

Платката е оптимизирана за минимална площ. Това налага разположението на компонентите да е двустранно. Използвани са готови модули за 3 от блоковете - микроконтролерът, GPRS модемът и повишаващият регулатор.

Тъй като микроконтролерният модул включва и антена, специално внимание трябва да се отдели за да няма разположени компоненти в близост до нея.

При тези размери на платка е трудно да се спази изискването за антената. В нашия случай, GPRS модемът е поставен на рейки, а изводите, които не се използват, са отстранени, за да може на тяхно място да се разположи микроконтролерът.

При кратката си пикова консумация, GPRS модема може както да предизвика нежелани спадания на захранващото напрежение, така и силни електромагнитни смущения. Тези смущения са толкова интензивни, че могат да предизвикат нежелано рестартиране на микроконтролера, ако в близост до неговия извод за рестартиране, не се сложи кондензатор, който да филтрира смущенията. Кондензатори трябва да бъдат поставени и близо до всеки извод, който се използва за вход.



Фигура 5: Платка на станцията

7 Захранване от соларен панел

За източник на енергия бе избран монокристален соларен панел, съставен от 12 соларни клетки, свързани последователно. Всяка клетка има площ от 2940 mm^2 , което прави сумарната площ на 12^{те} клетки 353 cm^2 .

Приблизително можем да очакваме около 20 mW изходна мощност от квадратен сантиметър, което при 353 cm^2 , се равнява на 7.06 W . Нашият соларен панел по спецификации е 7 W .

Тази мощност на соларния панел е при идеални условия. През зимата трябва да очакваме до 5 пъти по-малко енергия заради късите дни и ниския интензитет на слънчевата радиация. Също трябва да сме подготвени за неблагоприятно разположение на станциите и респективно на соларния панел, както и за дълги периоди, през които панелът ще е затрупан със сняг. Тези фактори ни принуждават да разчитаме на значително по-малко енергия спрямо максималната възможна за такъв панел и трябва да ги запомним за да може после да си обясним несъответствието между средна консумация на станцията и максимална мощност на соларния панел.

Тъй като имаме 12 клетки свързани последователно, изходното напрежение ще е приблизително $12 \times 0.5 \text{ V} = 6 \text{ V}$ при товар на изхода и максимално $12 \times 0.7 \text{ V} = 8.4 \text{ V}$ при липса на товар. Това напрежение трябва да се понижи към $3 \text{ V} \div 4.2 \text{ V}$ за да можем безопасно да зареждаме литиево-йонна батерия.

7.1 Хардуер на захранването

Захранването може много лесно да се осъществи с понижаващ импулсен преобразувател, настроен да регулира изхода си до 4.2 V . Този подход не е ефективен. Когато напрежението на батерията е под 4.2 V , регулатора ще си повиши коефициента на запълване до 100% и максималният ток ще се ограничи от соларния панел. Проблема в този случай е работната точка на соларния панел - той ще отдава максимален ток, но напрежението му ще бъде под оптималното - 4.2 V вместо 6 V . Оттук можем да заключим че грубо ще губим $\frac{6 \text{ V} - 4.2 \text{ V}}{6 \text{ V}} = 30\%$ от

максималната мощност, която може да се усвои от панела. За да успеем да усвоим по-голяма мощност, бе използван микроконтролер, който следи зарядния ток към батерията и на базата на него оптимизира коефициента на запълване, управляващ ключовия преобразувател между панела и батерията. Освен това микроконтролерът си комуникира със станцията посредством I²C и от него може да се извлече информация за усвоената енергия и напреженията на панела и батерията.

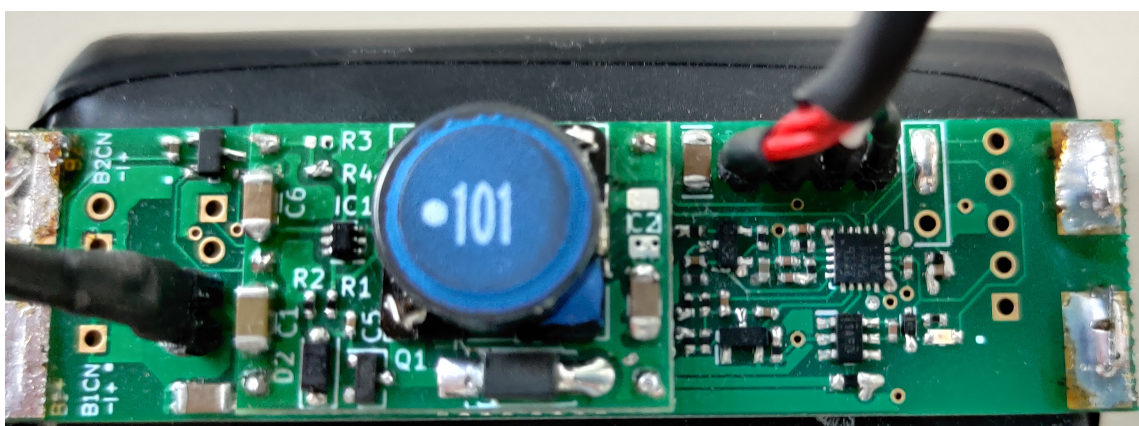


Фигура 6: Блокова схема на захранването

7.2 Дизайн на платката

Платката на захранването е проектирана да се използва с една или две литиево-йонни клетки, които се запояват директно към нея. Също понижаващият ключов регулатор е проектиран като модул на отделна платка, за да може лесно да бъде заменян с алтернативни дизайни, оптимизирани за други соларни панели. Токът към батерията се следи посредством шунт и диференциален усилвател INA181[3]. Усилвателят

е поставен възможно най-близо, както до шунта, така и до микроконтролерът (STM32L011F4[6]), за да се минимализират електромагнитните смущения, индуцирани в кръговете шунт-усилвател и усилвател-АЦП.



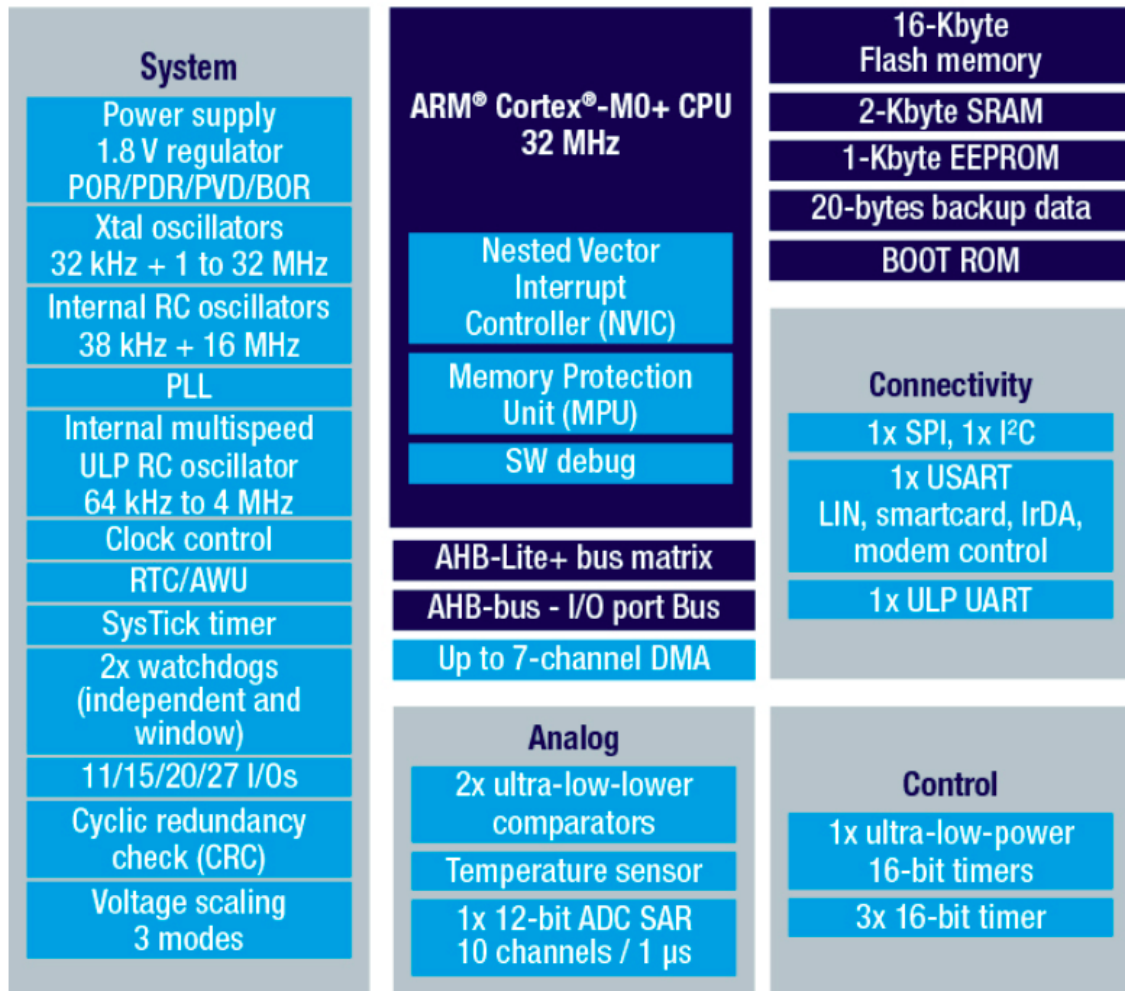
Фигура 7: Платка на захранването

7.3 Принципна схема

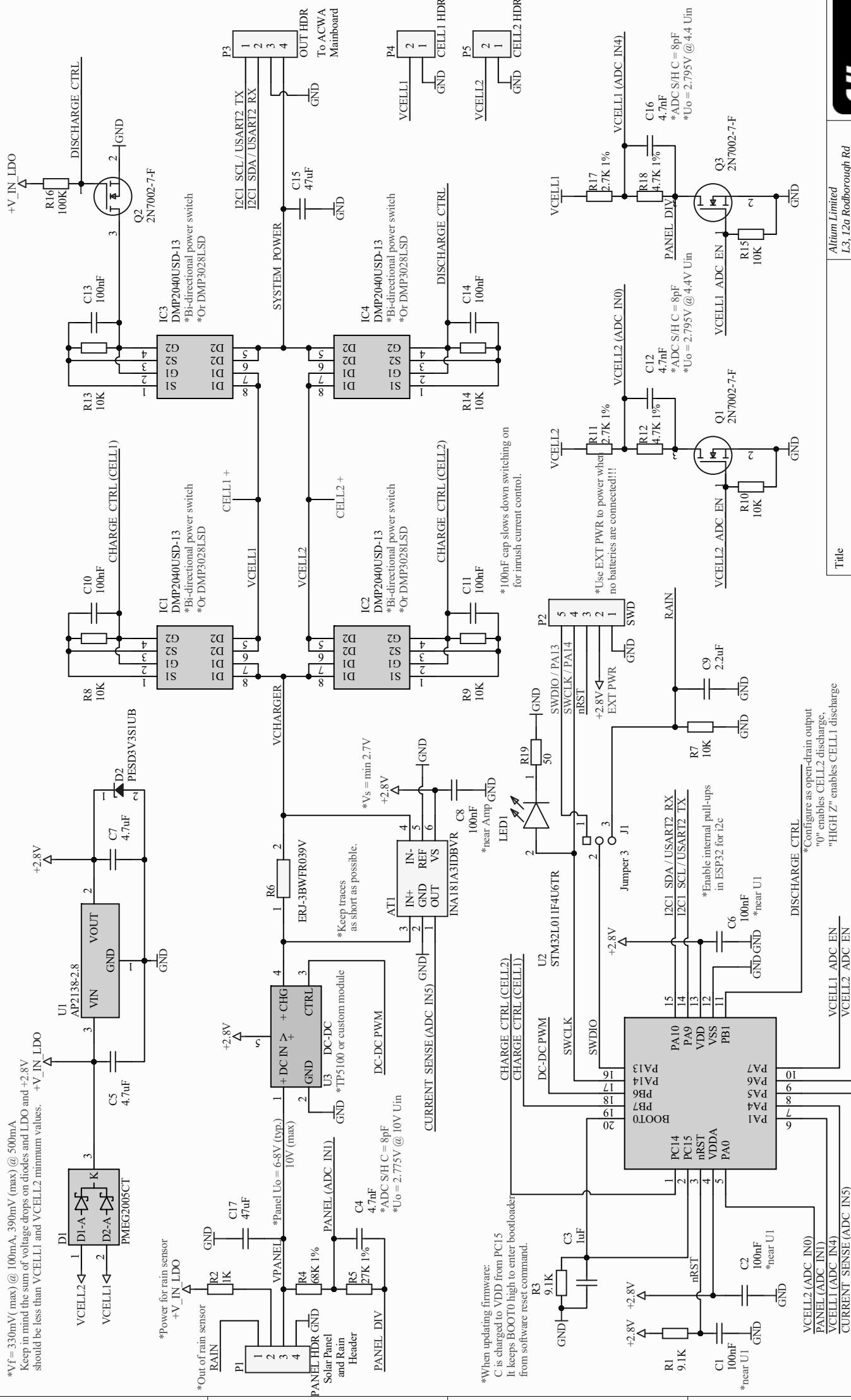
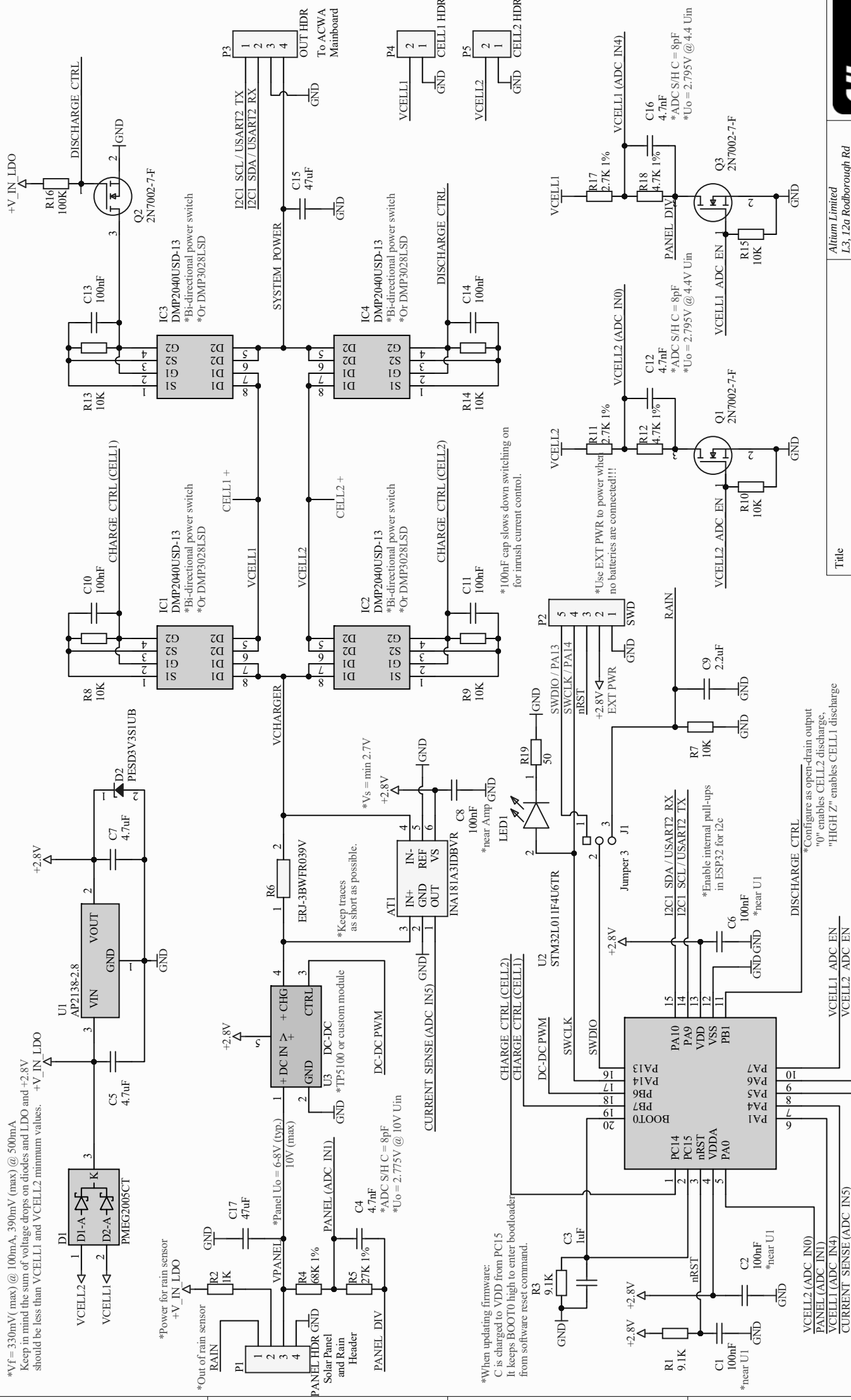
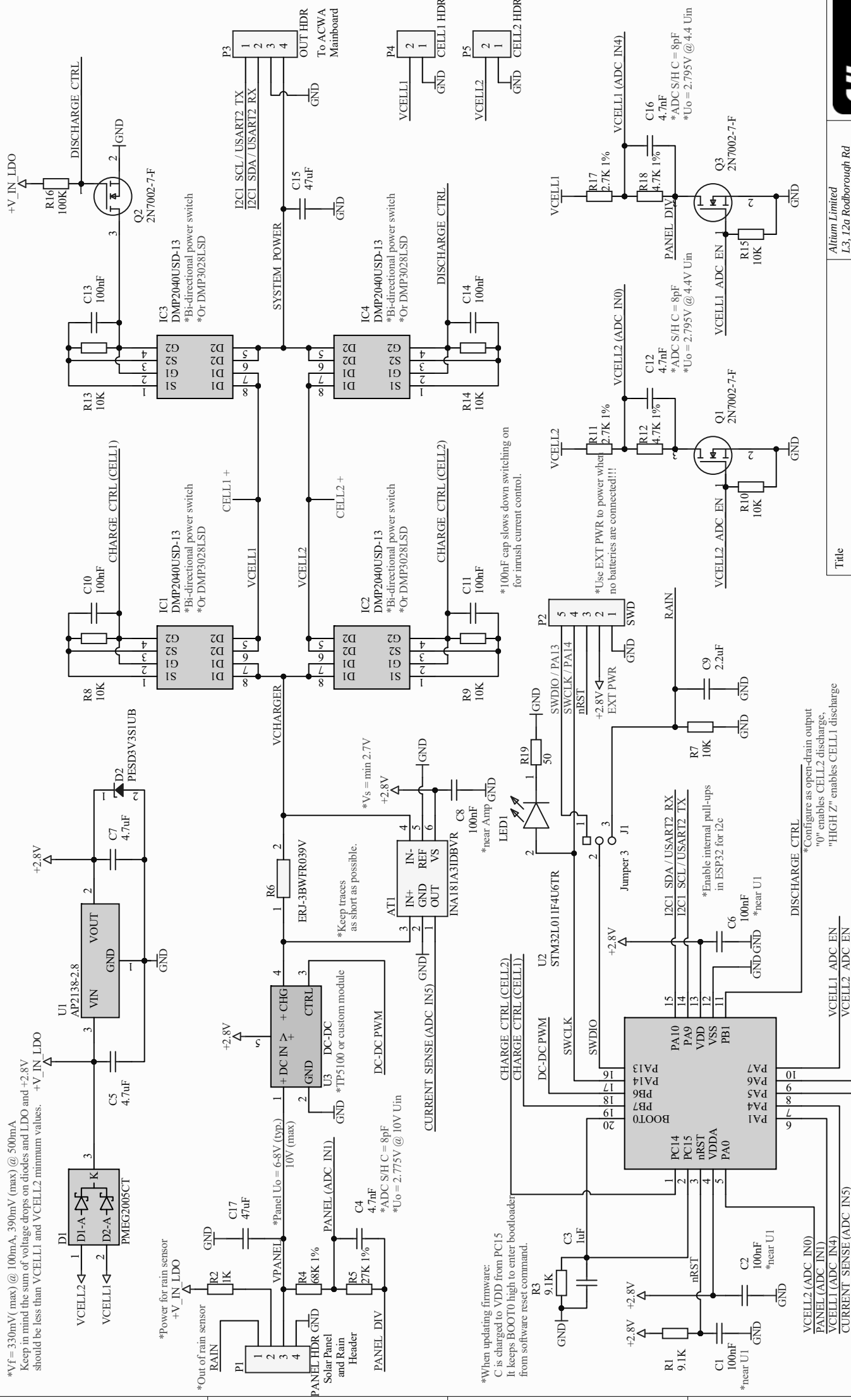
За да осигурим захранващо напрежение от 2.8 V, използваме два диода на Шотки. Анодите им са свързани към всяка от 2^{те} батерии, а катодите им към линейния понижаващ регулатор. Схемата е реализирана така, за да можем да осигурим захранване при липса на коя да е от двете батерии.

Резисторните делители се управляват с транзисторни ключове. Така можем да ги изчислим да работят с по-голям ток (567 $\mu\text{A max}$) и да ги изключваме когато не се използват. При по-голям ток през резисторните делители, намаляваме влиянието на входната схема за следене и запомняне на АЦП модула. Това позволява използване на наддискретизация (oversampling) с множество измервания през кратки интервали, без входният ток към АЦП-то да повлияе на измерването.

STM32L011x4



Фигура 8: Блокова схема на STM32L011F4



7.4 Оптимизиране на ключовия преобразувател

Най-важното решение, което трябва да вземем при оптимизацията на понижаващия ключов преобразувател е, на каква честота ще работи. От това решение зависи избора на схемата за управление на униполярен транзистор (gate driver), дросел, тактова честота на управляващия микроконтролер и разделителна способност на ШИМ-а. Ако изберем висока честота, ще можем да използваме дросел с по-ниска индуктивност (и по-малки размери), но ще трябва да изберем по-бърз gate driver, да вдигнем тактовата честота на микроконтролера и/или да намалим разделителната способност на ШИМ-а. За предпочитане е честотата на преобразувателя да е над 20 kHz за да не чува той по време на работа.

Нашият преобразувател работи на 28 kHz, а коефициента на запълване на управляващия сигнал може да се изменя със 150 дискретни стъпки. Това означава че такта към таймера, генериращ управляващия сигнал, трябва да е поне $150 \times 28 \text{ kHz} = 4.2 \text{ MHz}$.

Дроселът, който бе избран е с индуктивност 100 μH . При напрежение на панела от 6 V и напрежение на батерията 3 V, пулсациите на тока ще са:

$$\Delta I_L = \frac{(V_{in} - V_{out}) \times DutyCycle}{f_s \times L} \quad [1] \quad (4)$$

При заместване във формулата, получаваме 590 mA¹.

Вътрешното съпротивление на дросела е 110 m Ω , което означава, че при средноквадратичен ток през него от 1 A, ще губим 110 mW. При изходно напрежение 3 V, ни остава запас за загуби от около 250 mW в останалите компоненти, за да успеем да постигнем КПД над 90%.

Дроселът е избран с възможно най-големи размери, за да може да е с по-голяма индуктивност (така постигаме по-нисък ток на пулсации) и по-ниско вътрешно съпротивление (постигат се по-малки загуби).

¹Приемаме, че ефективността на преобразувателя е 90% (което ще доведе до коефициент на запълване от 55%) и изходният ток е над 300 mA (преобразувателят работи в режим на непрекъснат поток през дросела - CCM)

7.5 Усвояване на максимална мощност от соларния панел (MPPT)

За да успеем да усвоим колкото се може повече енергия от соларния панел, трябва да реализираме алгоритъм за следене точката на максимална мощност(MPPT)[9]. Този алгоритъм работи на следния принцип:

1. Измерва се напрежението и тока през соларния панел
2. Коефициента на запълване се събира с променлива, която първоначално е инициализирана с 0x01
3. Изчаква се 50 ms (за да се стабилизируют параметрите)
4. Отново се измерват напрежението и тока през соларния панел
5. Изчислява се мощността преди и след промяната на коефициента на запълване
6. Ако мощността след промяната е по-малка, сменяме знака на променливата, с която преизчисляваме коефициента на запълване
7. Цикълът започва от начало

Тъй като на изхода на ключовия преобразувател има батерия, можем да приемем, че промяната на напрежението ѝ ще е незначителна между отделните измервания. Това позволява да се опрости алгоритъмът. Вместо да се изчислява входната мощност, измервайки напрежението и тока на входа, може да изчисляваме изходната мощност като мерим единствено тока на изхода към батерията. Измерването на напрежението може да се прави през много по-големи интервали от време.

7.6 Консумация на захранващата платка

Захранващата платка консумира 4.1 mA по време на нормална работа. Тази консумация е около 2 декади повече от тази на станцията в нискоенергиен режим (35 μ A). Това не е проблем, защото по време на

нормална работа, захранването оптимизира тока на зареждане, който може да се повиши с над 200 mA благодарение на MPPT алгоритъма. През нощта, обаче, тази консумация е неоправдана, тъй като не усвояваме енергия от соларния панел.

За да намалим консумацията на захранващата платка, трябва да следим зарядния ток към батерията. Когато той е по-малко от 4 mA, батерията се разрежда. В този случай, тъй като използваме вътрешен преконфигурируем тактов генератор, можем да намалим тактовата честота, на която работим. В нашия случай, превключваме от 4.2 MHz към 1 MHz. След преминаване към 1 MHz, консумацията намалява наполовина (2.1 mA). При смяната на тактовата честота, ключовият преобразувател започва да работи на 6.7 kHz. Това не е проблем, защото токът на заряд е много малък. За да предотвратим постоянно превключване между двата режима при заряден ток от 4 mA, е реализиран хистерезис. Захранването си понижава тактовата честота при заряден ток под 4 mA и я повишава при заряден ток над 12 mA. Освен това, следим и състоянието на батерията. Ако тя е заредена до горната си граница, няма нужда да преминаваме към по-ниска тактова честота (токът е малък за да не презаредим батерията, а не защото е нощ).

В бъдеще трябва да се реализира по-агресивен режим за намаляване на консумацията през нощта. Вместо намаляване на тактовата честота, микроконтролерът, който използваме има възможност да спре такта си изцяло. Това ще усложни комуникацията между захранването и станцията. Също трябва да се реализира и тества по-сложен алгоритъм за събуждане и тестване на зарядния ток. Тези две предизвикателства не са решени все още заради ограниченото време за разработване на устройството и наличието на други ключови компоненти за оптимизиране.

7.7 Комуникация между станцията и захранването

Използването на микроконтролер в захранващата платка позволява по-прецизен контрол върху процеса на зареждане. За да позволим оператора на станцията да наблюдава и дори да управлява този процес, между станцията и захранващата платка има двупосочна комуникация.

За комуникационен интерфейс е избран I²C тъй като всички изводи на контролера се използват, а I²C шината може да се споделя с датчика за атмосферно налягане и температура. Захранващата платка работи в подчинен на главния микроконтролер режим, но освен това в бъдеще също би могла да комуникира с датчика за температура и атмосферно налягане.

Захранващата платка с настоящия си фърмуер поддържа над 20 команди за промяна режима на работа или за прочитане на параметър, който измерва и/или изчислява. В бъдеще командите лесно могат да бъдат разширени до 256 (или повече ако се смени адресирането да не е 8 бита).

По-интересните команди са:

- Задай горна граница за напрежението при зареждане на батерията
- Прочети текущото напрежение на батерията
- Прочети текущият заряден ток на батерията
- Прочети напрежението на соларният панел
- Стартирай ново принудено измерване на напреженията на соларният панел и батерията
- Спри процеса на зареждане
- Стартирай процеса на зареждане
- Прочети текущата версия на фърмуера на контролера
- Рестартирай контролера в режим на bootloader

7.8 Обновяване фърмуера на захранването

Последните две команди, които се поддържат от захранващата платка, не са случайни. Тяхната цел е да помогнат при реализиране обновяване на фърмуера без да се използват никакви допълнителни изводи

освен двата извода от I²C интерфейса. Когато има нужда да се обнови фърмуера на захранващата платка, компилираният двоичен файл се внедрява във фърмуера на главния микроконтролер¹. След това главният микроконтролер се обновява посредством GPRS модема и проверява текущата версия на фърмуера на захранващата платка. Ако тя е по-ниска от версията на внедрения двоичен файл, главният микроконтролер изпраща команда към захранването за влизане в режим на bootloader. След това I²C изводите на главния микроконтролер се преинициализират в изводи на UART интерфейс, тъй като bootloader-а поддържа само този интерфейс. После се стартира обновяване на фърмуера на захранващата платка. След успешно обновяване и сверяване на новозаписаният фърмуер, на bootloader-а се изпраща команда за стартиране изпълнение на фърмуера от адрес 0x00, което съответства на софтуерно рестартиране.

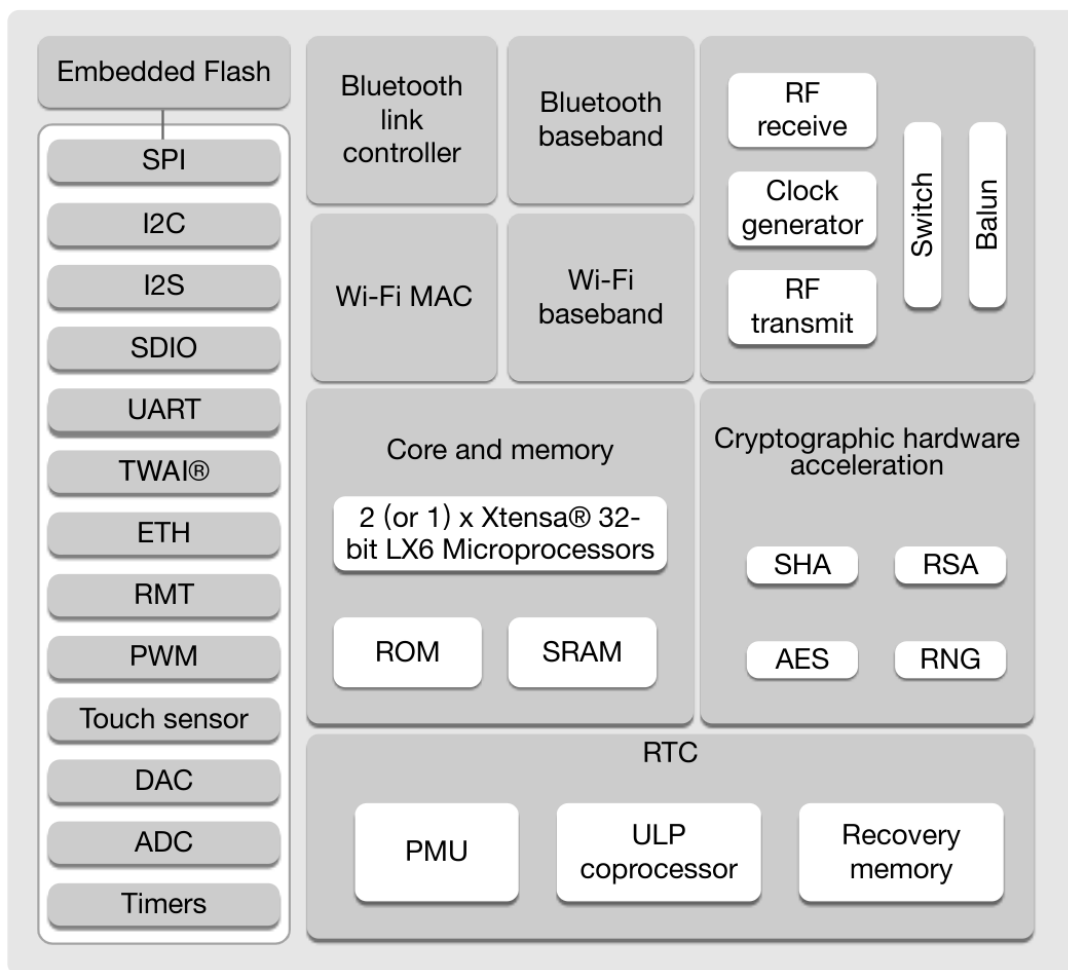
8 Фърмуер на станцията

В станцията за главен микроконтролер е избран ESP32rev3[2] с 32 мегабита външна SPI програмна памет и вградена оперативна памет от 512 kiB. Този микроконтролер предоставя огромно количество ресурси на програмиста. Това със себе си носи и огромно количество техническа документация за използване на тези ресурси.

Фърмуерът е базиран на опрационната система за реално време FreeRTOS и е разделен на следните модули:

- Модул за инициализиране
- Модул за извеждане на информация (посредством UART) за откриване и отстраняване на грешки
- Модул за комуникация с I²C периферия
- Модул за следене количеството паднал дъжд
- Модул за обслужване на ултразвуков датчик посредством UART

¹това не води до осезаема разлика в размера на програмата на главния контролер, тъй като съотношението между обема на фърмуера на двата контролера е над 20 към 1



Фигура 9: Блокова схема на ESP32

- Модул за обслужване на камера посредством UART
- Модул за измерване на физични величини посредством свързаните датчици и съхранение на данните при отпадане на GPRS мрежата
- Модул за съхранение на служебни конфигурируеми от потребителя параметри
- Модул за управление на GPRS модем
- Модул за обновяване фърмуера на контролера посредством GPRS модема

- Модул за обновяване фърмуера на захранващата платка
- Модул за комуникация със сървър

Обемът на фърмуера надхвърля 7 хиляди реда код на ANSI C. Това е далеч от грандиозен проект, но също няма как да се разгледа детайлно в настоящата дипломна работа. Вместо това ще се концентрираме върху съществените и интересни части и предизвикателства. Тук е важно да се отбележи, че както в почти всеки проект 90% от усилията за усъвършенстване *не са свързани с писане на нови подпрограми*, а с отстраняване на грешки от настоящите. Това погледнато от друг ъгъл означава, че целият фърмуер е написан сумарно за няколко месеца, а след това години са били нужни за отстраняване на грешки. Проектът за станциите, ще навърши 3 години от старта си през септември 2021 г.

8.1 Операционна система

Използването на операционна система утежнява фърмуера, но ако нейните функционалности се оползотворят по оптимален начин, тя може да позволи системата да стане по-гъвкава и дори по-бърза противно на очакванията.

В нашата система ограничаващият фактор за скоростта е GPRS модема, който освен че консумира най-много енергия, отнема най-дълго време за инициализация и установяване на връзка с мобилна клетка. По време на инициализацията на GPRS, операционната система ни позволява паралелно да инициализираме други периферии и да извечем информация от свързаните към станцията сензори.

Freertos е тясно интегрирана в средата за разработване на софтуер за нашия микроконтролер. Нейният диспечер се стартира още при най-ранна инициализация на микроконтролера и много от вградените функции в средата разчитат на нея. Основният плюс (заради интегрирането на Freertos) е добавената абстракция от системните прекъсвания, които изпращат сигнали към работни нишки, в които може много прецизно да се приоритизира обработката на тези сигнали.

8.2 Драйвер за GPRS модем

Най-важният модул във фърмуера е драйверът за GPRS модема. На него е базирано дистанционното обновяване на фърмуера, сверяването на вътрешния часовник (RTC) на микроконтролера и изпращане на акумулираните измервания и снимки към централен сървър.

При отказ на връзката със сървърът, станцията става безполезна. Проблемите в другите модули могат сравнително лесно да се отстраняват чрез фърмуерно обновяване, но при проблем с драйвера на GPRS модема, отказва и модулът за обновяване, който зависи от него.

Драйверът се базира основно на комуникация между контролера и модема посредством сериен интерфейс. Комуникацията е чрез AT команди, които са стандартизирани от консорциума 3GPP. Много е важно да се използват функционалностите на двупосочната комуникация, а именно контролера трябва обработва всеки отговор на модема. При възникване на проблем, информацията която трябва да се качи на сървъра се съхранява във вътрешната енергонезависима памет за да се изпрати при следващото свързване.

8.3 Модул за обновяване на фърмуера чрез GPRS

Станцията разполага с 4 МВ енергонезависима памет. Тази памет е разделена на отделни региони. Два от тези региони са предвидени за фърмуера на станцията. Използват се два региона именно за да може да се осъществи безопасно обновяване на фърмуера. Когато станцията функционира, програмата се зарежда от единия регион, а когато започне да се обновява, записва новият фърмуер в неизползвания регион. При потвърждение валидността на фърмуера, станцията се рестартира, зареждайки програмата от новообновения регион.

За да се осъществи фърмуерно обновяване, се използват два отделни bootloader-а. Първият е записан във вградена енергонезависима памет, интегрирана в самия микроконтролер. Той инициализира външната SPI памет и стартира втория bootloader от адрес 0x1000. Вторият bootloader прочита таблицата на регионите и от нея открива адреса на региона със служебна информация. В този дял от паметта, при последното обновя-

ване е указано къде е записан новия фърмуер. Последно bootloader-а отново проверява първия "магически" байт от фърмуера - той трябва да е 0xE9¹. Ако първият байт съвпадне, се зарежда новият фърмуер. А ако не съвпадне, bootloader-а се връща към региона със служебна информация и търси друг дял, от който да зареди работен фърмуер.

Нашият модул за обновяване се свързва към централния сървър, след което започва да тегли новата версия на фърмуера като паралелно с това проверява нейната валидност и я записва в SPI паметта. При успешно изтегляне и презаписване на новата програма, се обновява региона със служебна информация, указвайки на bootloader-а при следващото рестартиране да зареди новозаписания дял от паметта.

9 Трудни за откриване софтуерни и хардуерни проблеми

Тук ще дискутираме най-интересните проблеми при разработката на софтуера и хардуера на станцията. Някои от тях изглеждат тривиални след откриването им, но преди това са предизвиквали необяснимо поведение на станциите за дълги периоди от време.

9.1 Отказ на ултразвуковия датчик през деня

Ултразвуковият датчик, както вече беше обяснено, ипраща кратък звуков импулс и го регистрира, след като се отрази от обекта на измерване.

Аналоговата част, която регистрира отразения сигнал, е ключов елемент при работата на датчика. Тя се състои от усилвател с лентов филтър, който усилва само отразения сигнал. Сигналят затихва пропорционално на разстоянието, което е пропътувал, което от своя страна е пропорционално на времето, изминало от момента на изпращане. Поради това явление, повечето добри ултразвукови датчици добавят

¹Тази проверка се прави в случай че при някаква неизправност в програмата се окаже невалиден адрес за стартиране или се презапише настоящият адрес с невалидна информация

компенсация на усилването спрямо изминалото време от момента на излъчване на импулса. Това означава, че непосредствено след излъчване, усилвателят има малък коефициент на усилване за да не детектира фалшив сигнал, но пропорционално на времето, коефициента се увеличава за да можем да регистрираме дори и слаби отразени сигнали от далечни обекти.

Независимо от ултразвуковия датчик, захранващата платка зарежда батерията посредством ключов регулатор, винаги когато батерията не е напълно заредена и има слънце.

Когато батерията се зарежда с голям ток, това е придружено от усилване на смущенията от ключовия регулатор. Тези смущения не пречат на повечето блокове от системата, но могат да предизвикат много специфичен отказ на ултразвуковия датчик.

Докато ултразвуковият датчик се използва за детектиране на обекти на разстояние до метър и половина, усилването на алоговият приемник на импулсите, не надвишава критичната граница, при която смущенията от захранването му влияят. Ако разстоянието надвиши метър и половина, обаче, датчика започва да показва фалшиви измервания, поради фалшиво предварително задействане на алоговия приемник. Тези измервания са със случайна стойност между метър и половина и два метра.

За да се предотврати този проблем, зареждането на батерията се преустановява чрез команда към зареждащата платка по време на работа на ултразвуковия датчик. Това почти не влияе на енергията, която можем да усвоим от панела, тъй като датчика се влючва за около 3 секунди през половин час или в най-лошия случай по време на авария - през няколко минути.

9.2 Отказ на станцията по време на дъжд

Станцията изпраща пост към сървъра през конфигурируем интервал, обикновено половин час. Времетраенето при качване на нов пост е между 20 секунди и 3 минути, в зависимост дали трябва да се изпратят SMS-и при аларма и/или да се направи и качи снимка на измерваният

обект. През останалото време главният микроконтролер е в режим на ниска консумация.

По време на работа на станцията, входът свързан към датчика за дъжд, е конфигуриран да генерира прекъсване, а преди станцията да влезе в режим на ниска консумация, този вход се конфигурира да събуди станцията при появяване на импулс от датчика. Също преди да влезе в режим на ниска консумация, станцията записва времеви маркер в този момент.

При събуждане, преди станцията да започне инициализация, проверява дали е събудена от таймер или от входа за дъжд. В случай, че е събудена от входа за дъжд, изчислява времето изминало от времеви маркер до текущия момент в програмата. На базата на това време (което отговаря на времето, прекарано в режим на ниска консумация), преизчислява колко остава до заданието за нов пост, увеличава брояча за дъжд и отново навлиза в режим на ниска консумация.

В много редки случаи, събуждането от датчика за дъжд настъпва непосредствено преди събуждането от таймер. Докато програмата стигне до момента, в който изчислява времето прекарано в режим на ниска консумация, вече са минали няколко десети от секундата. Това прави изчисленото време, малко по-голямо от оригиналното задание. След увеличаване на брояча за дъжд, се извиква следната функция:

deep_sleep(assigned_time_to_sleep_us - time_slept_us);

При изваждането резултата трябва да е строго положителен, но тъй като вторият операнд е по-голям от първия, настъпва препълване и станцията заспива за месеци.

Този проблем е тривиален за решение - преди да заспим, трябва да направим проверка дали вторият операнд е по-голям от първия. В случай че е, просто продължаваме към останалата част от програмата, както бихме направили ако се събудим от таймер.

9.3 Отказ на станцията след фърмуерно обновяване

ESP32 има 512kiB оперативна памет. По време на режим на ниска консумация, повечето от тази памет се изключва, за да може консума-

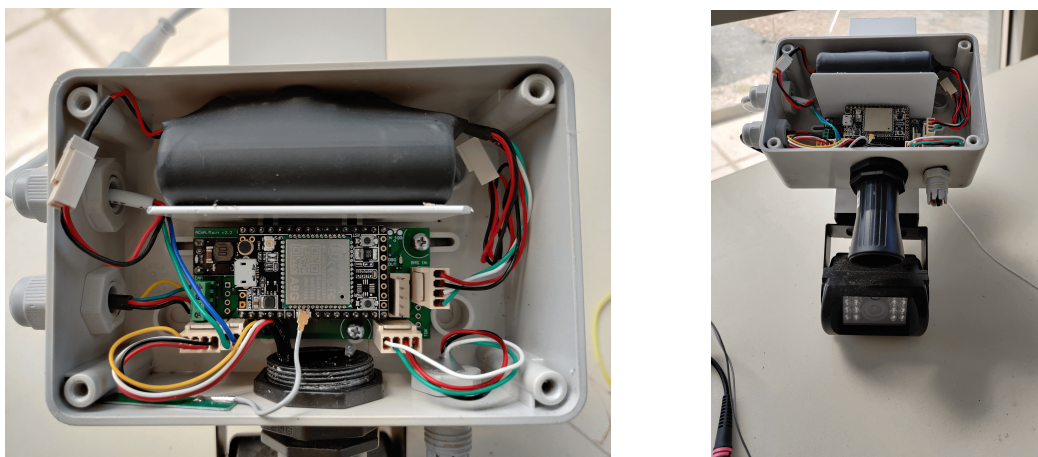
цията да спадне под 100 μ A. Малка част от паметта (16 kiB) остава активна. Нейната стойност се губи само при отпадане на захранването или хардуерно рестартиране. Тази памет се използва за съхранение на различни служебни параметри. Един от тези служебни параметри е указател към масив, записан в енергонезависимата памет. Този указател се преинициализира само след отпадане на захранването или хардуерно рестартиране. При компилиране на фърмуера, обаче, в някои нови версии, компилаторът променя адреса на този масив. След обновяване на фърмуера, при опит на програмата да достъпи масива чрез необновения указател, се стига до неопределено поведение.

Отново можем да решим проблема много лесно - преинициализираме указателя независимо от причината за рестартиране на контролера.

10 Опитни резултати

10.1 Сглобяване на 30 екземпляра

При дизайна на станцията бе обърнато внимание да се проектира възможно най-компактна и лесна за сглобяване с минимално обучение на неквалифициран персонал. В кутията се виждат два щуцера от ляво, през които влизат проводниците на камерата, датчика за дъжд и соларния панел. Камерата и датчика за дъжд са свързани към основната платка (разположена в центъра), а соларният панел е свързан към захранващия модул (разположен над основната платка), който е обвит в черен термошлаух. Между захранващия модул и главната платка има 4 проводна връзка за комуникация и захранване. Под нея има 4 проводна връзка към датчика за атмосферно налягане и температура който е поставен в щуцер. В дъното на кутията до датчика за температура е поставен ултразвуков датчик за дистанция. Зад него е монтирана камерата.



Фигура 10: Сглобена станция

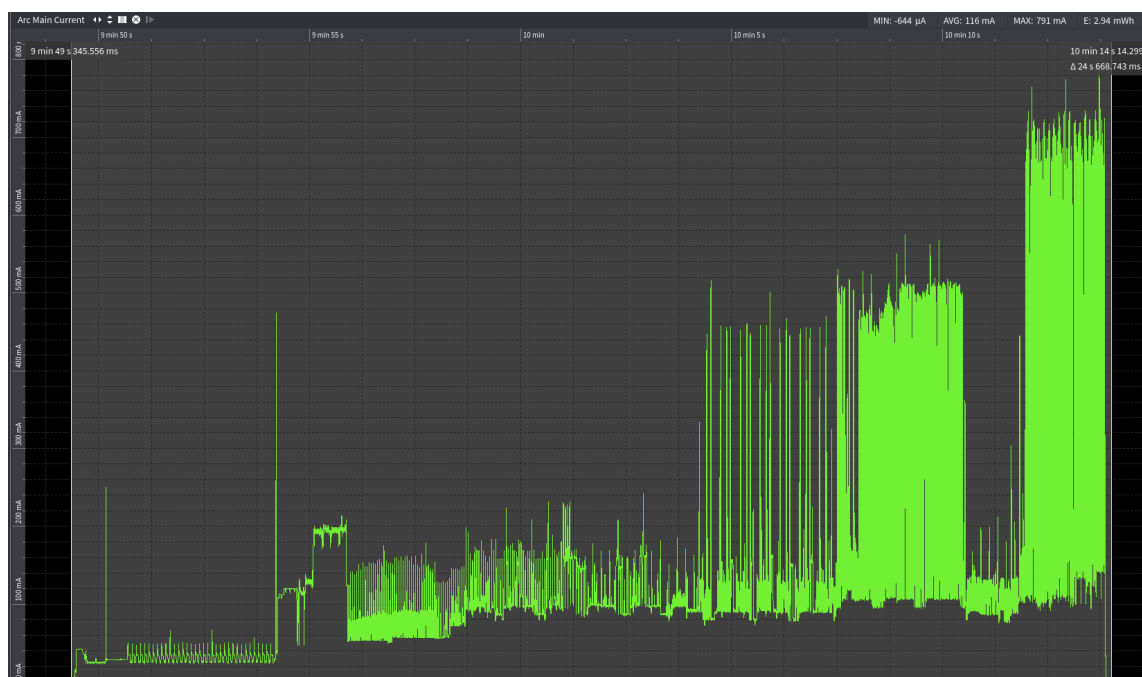


Фигура 11: Партида от 10 станции

Станциите бяха сглобени и инсталирани на 3 отделни партиди от 10 станции, като след всяка партида бяха идентифицирани и оптимизирани най-бавните стъпки при сглобяване и монтиране. При последната партида (Фиг. 11), процесът на сглобяване беше завършен за един работен ден от двама души. Това означава, че средно са необходими около 2

човеко-часа за сглобяване на една станция (без насищане на платките). Този процес може да бъде допълнително забързан при производство на по-големи партии.

10.2 Консумация на енергия по време на работа



Фигура 12: Консумация при качване на един пост

Графиката показва консумирания ток, при 3,7 V захранващо напрежение. От лявата страна се виждат два кратки пика от 250 mA и 480 mA. Те съответстват на включването на ултразвуковия датчик (по-малкия пик) и GPRS модема (по-големия пик). Причината за тях е зареждането на кондензаторите на входовете на тези два модула. След включването на GPRS модула, се забелязва силно завишаване на консумацията, свързано, първо с инициализацията и регистрирането на модема към клетка на оператора, а после с качването на пост към централния сървър и получаване на отговор от него.

Средната консумация е 116 mA за времетраене от 25 секунди. Енергията за един пост е малко под 3 mW/h. При използване на батерия с

капацитет от 10 W/h, можем успешно да осъществим 330 поста, без необходимост от зареждане на батерията. Това означава, че през зимата, можем да изпращаме приблизително по един пост на всеки два часа, за един месец, при положение, че соларният панел е затрупан със сняг.

За да можем да издържим дълго време без приток на енергия, трябва също да оптимизираме и режима на ниска консумация. Нашата станция използва 35 μ A в режим на ниска консумация.

10.3 Взаимодействие с крайния потребител

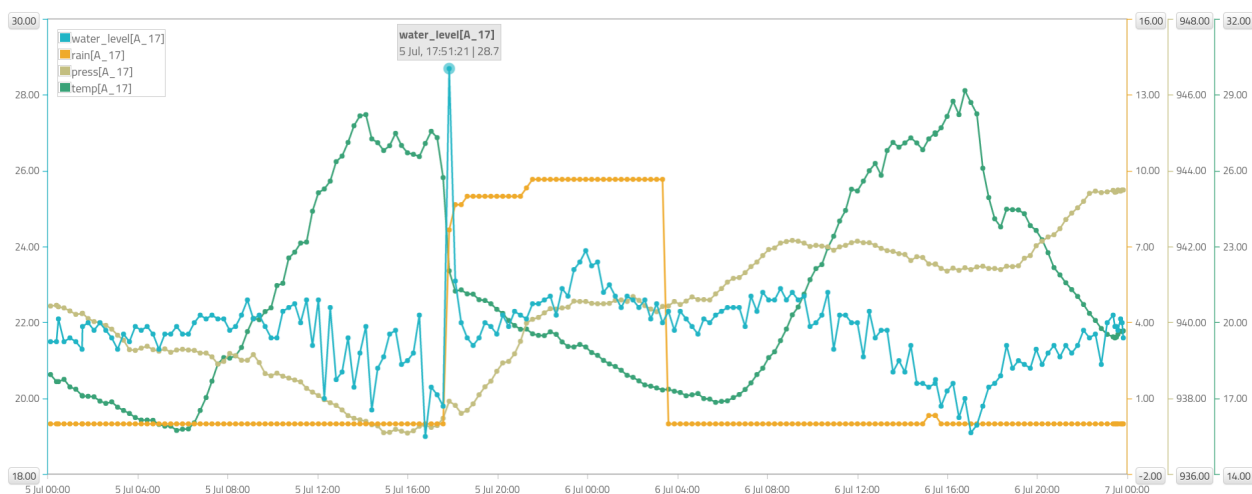


Фигура 13: Местоположение на станциите

Когато потребителят влезе в системата, първо вижда карта, на която са изобразени всички местоположения с инсталирани станции. При избор на конкретна станция, може да види колко време остава до следващия пост, какво е текущото водно ниво, снимка, мини графика и кога е бил последният пост на станцията.

При избор на режим на показване на графики, потребителят може да види графика на измерваните параметри за един или повече дни. Всеки един отделен пост се изобразява като точка върху графиката. При поставяне на курсора върху точката, се вижда точната стойност, дата и час.

На показаната графика можем да видим, как на 5^{ти} юли 2021 г. около 18 часа е валил дъжд, първоначално с голям интензитет, а след това с по-малък. Станцията е регистрирала повишаване на водното ниво от около 22 см. до 29 см. Освен това може да се види как температурата на въздуха спада по време на дъжда. Атмосферното налягане е било в локален минимум и постепенно се е покачило до 943 хектопаскала



Фигура 14: Графика на параметрите

При избор на конкретна станция, потребителят може да разглежда снимките от нея на голям екран. Разделителната способност на снимките е 1024x768 пиксела. През нощта се включват инфрачервени светодиоди, но въпреки това се оказва че водата поглъща почти цялото инфрачервено лъчение.



Фигура 15: Снимки от станциите

Много скоро, на част от станциите беше инсталирана допълнителна светкавица, управлявана през Bluetooth Low Energy (BLE), която значително подобрява нощните снимки. В момента на писане на дипломната работа, светкавиците още се тестват, но е планирано в бъдеще да се инсталират на всяка станция.

Използва се Bluetooth, защото така светкавиците могат да са самостоятелни устройства, отстранявайки се нуждата от модификации на вече инсталираните станции. Освен това, по-късно може да ми трябва и тема за магистърска дипломна работа :)

11 Специални благодарности

Аз, Христо Митрев, съм изключително благодарен на всеки който ми помогна при изпълнението на проекта.

11.1 Проектиране и опроводяване на платки

Благодаря на Велизар Мазнев, който опроводи 2 отделни версии на печатна платка¹. Също благодаря за дизайна на схемата и опроводяването на захранващата платка.

11.2 Фърмуерни модули

Благодаря на Георги Жéлезов, който работеше паралелно с мен² за реализирането първо на модула за фърмуерно обновяване, а после и за реализирането на модула за управление на камерата.

11.3 Графичен дизайн на блокови схеми

Благодаря на Тихомир Делчев, който ми помогна да направя графично оформление на блоковите схеми (за да ги добавя в дипломната работа) на главната платка и на захранващата платка.

11.4 Отстраняване на правописни грешки

Благодаря на Любомир Богданов, Георги Жéлезов, Ивелина Митрева и Ива Станчева, които прочетоха цялата дипломна работа или част от нея и ми помогнаха да намеря и отстраня множеството правописни и стилистични грешки³

¹Първата версия беше проектирана и тествана няколко месеца за да се идентифицират възможни проблеми. След това беше проектирана втората версия, в която почти всичко е преработено

²Аз трябваше да работя по други части от фърмуера, докато той самостоятелно реализира двата модула

³Оказва се че решението ми да се занимавам с инженерна дейност, вместо с лингвистична, е било много добър избор.

11.5 Разработване на bluetooth светкавица

Благодаря на проф. Рачо Иванов и д-р инж. Любомир Богданов за разработването на автономна светкавица (захранвана от соларен панел и управлявана чрез bluetooth), която значително подобри качеството на снимките от станциите през нощта.

11.6 Разработване на устройство за отдалечено откриване на грешки

Благодаря на д-р инж. Любомир Богданов, който разработи устройство, което записва информацията за откриване на грешки, изваждана на един от изводите на станцията. Това устройство ми позволи дистанционно да следя част от станциите посредством GPRS и да отстраня множество трудни за откриване проблеми.

11.7 Производство и инсталиране на станциите

Благодаря на Велизар Мазнев, който насити платките на станцията и захранването и подготви датчиците за атмосферно налягане и температура да бъдат монтирани при процеса на сглобяване на крайното изделие. Благодаря и на Калин Сакалийски, който сглоби и монтира повечето станции.

11.8 Морални и технически насоки

Благодаря на моя началник д-р Николай Илиев, който за дълго време ме освободи с цел да напиша възможно най-добрата дипломна работа. Благодаря и за житейските съвети и напътствия, както и за помощните указания от високо ниво при технически проблеми.

12 Анотация

Септември 2019г. беше започнат проект за разработване на автономна станция за следене нивото на реки и водоеми. Постепенно бяха тествани различни датчици, модеми за връзка към централен сървър, методи за предаване на информация към сървъра, камери за заснемане обекта на измерване, соларни панели и захранванваща електроника.

Кулминацията при разработката на това устройство беше през 2021г. когато след смяна на няколко различни дизайна, бяха проектирани финалните платки на станцията и захранването, а през март, бяха сглобени и инсталирани 30 отделни станции по проект, спечелен на търг от Софийска община¹.

По време на разработване на станциите имаше много интересни моменти с проблеми в хардуера и софтуера. Постепенно проблемите бяха отстранявани, а станциите подобрявани.

Освен отстраняване на проблеми, при натрупване на повече опит с архитектурата на използваните микроконтролери, фърмуера беше разширяван. Бяха реализирани и методи за обновяване на фърмуера и на двата използвани микроконтролера, за да могат станциите да бъдат оптимизирани и в бъдеще.

Също бяха използвани интересни решения за споделяне на един периферен модул между няколко устройства и смяна на протокола за комуникация със захранването, за да можем да използваме по-малко изводи за комуникация и обновяване.

Най-важното постижение на проекта е успеха при използване на максимума от предоставените ресурси, но освен това индентифицирането на компонентите, при които не трябва да се правят компромиси с техническите параметри (пример за това е автономното захранване).

¹Страница за общ достъп, с ограничени права за преглед може да се намери на: <https://acwa.app/mapWidget>.

Литература

- [1] *Basic Calculation of a Buck Converter's Power Stage*. 2015. URL: www.ti.com/lit/an/slva477b/slva477b.pdf.
- [2] *ESP32 Datasheet*. URL: www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf.
- [3] *INA181 Datasheet*. URL: www.ti.com/lit/ds/symlink/ina181.pdf.
- [4] *LPS33W Datasheet*. URL: www.st.com/resource/en/datasheet/lps33w.pdf.
- [5] Maithil Pachchigar. *Increase Dynamic Range of SAR ADCs Using Oversampling*. 2015. URL: www.analog.com/en/technical-articles/increase-dynamic-range-of-sar-adcs-using-oversampling.
- [6] *STM32L011F4 Overview*. URL: www.st.com/en/microcontrollers-microprocessors/stm32l011f4.html.
- [7] *Temperature and the speed of sound*. URL: www.nde-ed.org/Physics/Sound/tempandspeed.xhtml.
- [8] *VC0706 Protocol*. URL: www.adafruit.com/datasheets/VC0706protocol.pdf.
- [9] *What is Maximum Power Point Tracking (MPPT)*. URL: www.solar-electric.com/learning-center/mppt-solar-charge-controllers.html.
- [10] *Правила за писане на запетая пред съюза "и"*. URL: www.kaks epishe.com/zapetaya-pred-i.