**Приложение 1**

**Документация на проект № 732**

# ТЕМА:

**ClimaLog** – метеорологична станция, измерваща температура, влажност и дълбочина на снежната покривка, предназначена за използване в планините

# АВТОРИ:

Име: Георги Николов Николов

ЕГН:………………………….

Адрес: Благоевград ул. Цар Борис Първи 29

Телефон: 088 510 1405

Имейл: [georgi.nnikolov07@gmail.com](mailto:georgi.nnikolov07@gmail.com)

Училище: ПМГ „Акад. С. Корольов“

11. клас

# РЪКОВОДИТЕЛ:

Име: д-р Емилия Величкова Николова

Телефон: 089 870 0128

Имейл: emilinikol@gmail.com

Длъжност: старши учител по информатика и ИТ

Училище: ПМГ „Акад. С. Корольов“

# РЕЗЮМЕ:

## Цели:

Всяка година медиите съобщават за инциденти в планините, много от които са следствие от липса на актуална информация за климатичните условия на дадено място.

Идеята на **ClimaLog** е именно да намали случаите на инциденти, като предоставя на потребителите информация в реално време, достъпна от която и да е точка с връзка към глобалната мрежа.

Подобни станции съществуват, но тези, измерващи и дълбочината на снега, са предназначени за професионален weather monitoring, което прави цената им неоправдана за обикновения потребител, имащ вила в планината, или за стопанин на хижа.

## Основни етапи в реализацията:

* Избор на подходящ хардуер и технологии
* Изработка на прототип и тестване на сензорите в реални условия
* Планиране на основните функционалности на всеки модул
* Създаване на Embedded Software за всеки модул. Тестване
* Създаване на web script за съхранение на данните
* Изработка на печатни платки и кутии. Сглобяване
* Създаване на мобилно приложение за визуализиране на последните получени данни
* Тестване на завършения проект. Отстраняване на проблеми

## Ниво на сложност на проекта:

Основните затруднения бяха свързани с обхвата на радиокомуникацията между модулите. Тяхното решение се оказа подмяната на радиата с по-мощни.

Друг проблем беше калибрирането на сензора за дистанция така, че да измерва дистанция до сняг коректно. Това се осъществи чрез използване на функциите за прецизно измерване, вградени в библиотеката, използвана за сензора.

## Логическо и функционално описание на решението:

Проектът се състои от един вътрешен модул (receiver), до шест външни модула (transmitters) и **web** приложение (**backend** – таблица за съхраниние, скриптове за обработка и пренос на данните; **frontend** – мобилно приложение за лесна визуализация)

* + Вътрешен модул:
* Измерва температурата и влажността в помещението
* Приема данните от всички външни модули
* Има интерфейс (OLED дисплей) за визуализиране на актуални данни и промяна на някои основни настройки (височината на монтиране на всеки външен модул)
* Микроконтролер: Espressif ESP32 WROOM
* Захранване: чрез microUSB port
  + Външни модули:
    - Измерват температурата и влажността на въздуха и дълбочината на снежната покривка непосредствено под тях

За измерването на дълбочината се използва лазерен сензор за разстояние

* + - Изпращат данните към вътрешния модул през интервал около 7 минути
* Микроконтролер: Arduino Nano (ATMega328)
  + - Захранване: Li-ion батерия (2800mAh; около 20 дни живот) или жично
  + Web приложение:
    - Получава данните от вътрешния модул по http протокол, през интервал от 1 час.
    - Съхранява ги в google sheets таблица (създават се нови колони за всеки включен външен модул)
    - Мобилно приложение: при отваряне визуализира последния добавен ред от таблицата

## Реализация:

**Използван хардуер:**

* + Измерване на температура и влажност – DHT22
  + Лазерен сензор за дистанция - vl53l0x
  + Радиокомуникация - nRF24L01 PA (long range)
  + OLED дисплей – SSD1306

Embedded програмите на модулите са написани на езика C++. Комуникацията между модулите става посредством радиовълни с честота 2.4GHz, което има определени лимитации – между всеки външен модул и вътрешният не е препоръчително да има стени с метална конструкция, тъй като намаляват силата на сигнала. Ако това е изпълнено, обхватът е около 30 метра – предостатъчен за повечето приложения на проекта.

Следването на принципа обектно-ориентирано програмиране (OOP) на вътрешния модул позволява лесната подмяна на отделните му сегменти (дисплей, радио), както и добавянето на допълнителни външни модули с минимална промяна в кода.

Възможностите на контролера ESP32 – Onboard WiFi и библиотеката HTTPClient, позволяват изпращането на данните до Web приложението, без нужда от допълнителен хардуер.

Web апликацията е създадена върху платформата за автоматизация Google Apps Script. Данните се съхраняват в Google Sheets таблица, в която може да се прави статистика по различни критерии.

Мобилното приложение извежда последните изпратени данни от таблицата, с цел по-интуитивна и бърза визуализация. То е направено на платформата .NET Xamarin.

## Описание на приложението:

За да се използва ClimaLog е необходимо първо да се инсталира. Външните модули се монтират на оптимална локация, както е описано по-долу. Няма изисквания към монтирането на вътрешния модул.

* 1. Инсталиране на Web приложението:

//Да се довърши

//Да се добавят снимки

* 1. Монтиране на външен модул:
     1. Кутията се монтира водоравно на височина от 50 см до 110 см, която се мери от долния й край до земята. Препоръчително е да бъде над равна, открита повърхност без пряка слънчева светлина.
  2. Функции на вътрешния модул:
     1. При включване на захранването първо се появява меню за избиране на една от запазените WiFi мрежи. С еднократно натискане на бутона SET се избира мрежа, а със задържане на SET се потвърждава изборът.
     2. Режим по подразбиране (Визуализира актуални данни)
* Единично натискане на бутона DISP сменява модула, чиито данни се визуализират
* Ако батерията на даден модул падне под 30%, на дисплея се появява икона.
  + 1. Настройване на височина на монтиране на външен модул
* Задържане на бутона SET активира менюто за селектиране на външен модул. Единично натискане на SET сменява селектирания модул, а повторно задържане на SET отваря менюто за избиране на височина.
* В менюто за избор на височина, единично натискане на SET увеличава избраната височина с 5см, а задържане на SET запаметява направения избор.

## Заключение: