

(19)



**Евразийское
патентное
ведомство**

(21) **202390482** (13) **A1**

(12) **ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ЕВРАЗИЙСКОЙ ЗАЯВКЕ**

(43) Дата публикации заявки
2024.07.31

(22) Дата подачи заявки
2022.12.30

(51) Int. Cl. *A01F 25/02* (2006.01)
A01F 25/16 (2006.01)
G08C 17/02 (2006.01)
H04W 4/02 (2018.01)
H04W 84/18 (2009.01)

(54) **БЕСПРОВОДНАЯ БЕСКОНТАКТНАЯ КОНТРОЛЬНО-ИЗМЕРИТЕЛЬНАЯ СИСТЕМА
ДЛЯ ИЗМЕРЕНИЯ ВЕРТИКАЛЬНОГО ПРОФИЛЯ ТЕМПЕРАТУР И ВЛАЖНОСТИ
КОРНЕПЛОДОВ/ОВОЩЕХРАНИЛИЩ/ЗЕРНОХРАНИЛИЩ**

(96) 2022000148 (RU) 2022.12.30

(71)(72) Заявитель и изобретатель:
**ПЕТРУСЕВИЧ СЕРГЕЙ НИКИТИЧ;
КУРИЛЬСКИЙ ДМИТРИЙ
МИХАЙЛОВИЧ; ЖИЛИН ПАВЕЛ
ЛЬВОВИЧ; КОШУРИНА АЛЛА
АЛЕКСАНДРОВНА (RU)**

(74) Представитель:
**Медведев В.Н., Станковский В.М.,
Капустина Ю.В., Яковлев Д.М. (RU)**

(57) Изобретение для оперативного разноуровневого мониторинга условий хранения различных продуктов сельского хозяйства в кагатах или буртах. Может быть использована в сельскохозяйственной отрасли и научно-исследовательской деятельности. Измерительное устройство позволяет получать информацию о вертикальных профилях условий хранения корнеплодов различных сельскохозяйственных культур, в частности о температуре и влажности, также позволяющее дистанционно передавать информацию. Техническим результатом является расширение арсенала технических средств для гидро- и термометрии, на стадии сборки с целью минимизировать потерю урожая при его хранении.

202390482
A1

202390482
A1

G01K 1/02

G01K 1/14

G01K 13/10

ОПИСАНИЕ ИЗОБРЕТЕНИЯ

Беспроводная бесконтактная контрольно-измерительная система для измерения вертикального профиля температур и влажности корнеплодов /овощехранилищ/зернохранилищ

Использование корнеплодов различных сельскохозяйственных культур на промышленные цели связано с хранением сырья продолжительностью 60-160 дней и более на кагатных полях. При этом, например, при хранении сахарной свёклы в кагатах она дышит, потребляя сахарозу и кислород воздуха и выделяя углекислый газ и тепло. При попадании в кагат земли, ботвы условия отвода тепла и углекислого газа ухудшаются, возникают очаги с повышенной температурой, в которых интенсивно развиваются микроорганизмы, вызывающих гниение корнеплодов, деятельность которых является одной из причин потерь свеклы. Болезни корней свеклы во время хранения характеризуются общим названием - кагатная гниль. Потери от кагатной гнили могут быть весьма значительными: на отдельных сахарных заводах потери составляют до 30%. При количестве смеси гнилой массы 8-10% и больше заводы часто совсем не получают кристаллического сахара. Аналогичная ситуация наблюдается с другими культурами.

Известно устройство для измерения температур сыпучих материалов (пат. СССР SU 1527518 A1), содержащее датчики температуры, размещенные вдоль стенки гибкой воздухопроницаемой оболочки со сжатым воздухом, клапаном и штуцером, отличающееся тем, что, с целью повышения надежности и удобства при эксплуатации, оно снабжено тягой в виде замкнутой петли, охватывающей оболочку.

Существенным недостатком аналога является проводной интерфейс передачи данных и невозможность определять значение влажности воздуха.

В качестве прототипа (патент РФ RU 211225 U1) выбрано устройство для измерения вертикального профиля температур в буртах при компостировании органических отходов представляющее собой прямой стержень из коррозионно-стойкого материала с низкой теплопроводностью, содержащее последовательно расположенные полые цилиндрические сегменты, соединительные втулки, датчики измерения температур, соединенные гибким электрическим проводом, проходящим внутри сегментов и втулок, сообщаемые по протоколу 1-wire с контроллером-логгером, который заключен в корпус держателя, и при помощи разъема считывающее, сохраняющее на съемный носитель информации, обрабатывающее и отображающее в режиме реального времени на графическом дисплее данные от датчиков измерения температур, каждый из которых имеет уникальный цифровой код, заключен в корпус с тремя выводами и размещен в центральной части внутренних продольных каналов соединительных втулок, выполненных из материала с

высокой теплопроводностью, соединяющих между собой корпус держателя, полые последовательно расположенные сегменты первый длиной 350 мм, второй длиной 350 мм, третий длиной 300 мм и конусообразный наконечник.

Недостатками прототипа является то, что в связи с отсутствием датчика влажности возможно проводить контроль лишь одного параметра хранения (температуры), также ввиду отсутствия возможности дистанционной передачи данных, повышается трудоемкость снятия показаний при больших площадях хранения.

Задачей, на решение которой направлена заявляемое изобретение, является оптимизация процесса мониторинга вертикального профиля условий хранения в буртах при складировании сырья сельского хозяйства.

Техническим результатом является расширение арсенала технических средств для гидро- и термо- метрии профилей влажности и температуры воздуха в буртах при хранении корнеплодов и других сыпучих материалов.

Технический результат достигается за счет того, что заявленное устройство для измерения вертикальных профилей температуры и влажности воздуха в буртах при хранении корнеплодов сельскохозяйственных культур представляет собой программно-аппаратный комплекс беспроводной контрольно-измерительной системы температуры и влажности воздуха при хранении корнеплодов. Беспроводная бесконтактная контрольно-измерительная система (ББКИС) на основе построения топологии ячеистой сети, используя микроконтроллер (1) и датчики температуры (2), датчики влажности (3), позволяет нескольким устройствам связываться друг с другом в одной беспроводной сети. Датчики содержат модули индикации, позволяющие определить рабочее состояние и исправность устройств. Используется сетевой протокол, построенный на основе протоколов таких как BLE, Bluetooth Mesh и Wi-Fi Mesh Network. Используя картографическую систему датчики устанавливаются в зависимости установленной схемы (выбранной оператором), или же в хаотичном порядке. Информация с датчиков (2, 3), передается с помощью сетевого протокола передачи данных по беспроводной сети на модуль приема/передачи данных (4), где она обрабатывается с помощью специализированного программного обеспечения. Программное обеспечение взаимодействует с ББКИС, используя протокол, оператор на ПК или любом мобильном устройстве, с возможностью отображения информации получает информацию о температуре и влажности измеряемого объекта в режиме реального времени, те участки, которые выходят за рамки по установленным параметрам температуры и влажности подсвечиваются и сигнализируют оператору об отклонениях показаний в той или иной точке замера, работа ББКИС обеспечивается за счет модуля питания (5), в случае перебоев напряжения в электросети, стабильная работа системы обеспечивается с помощью источника автономного питания (6)

Схема выполнения ББКИС:

- герметичный корпус;
- аккумуляторная батарея Li-ion;
- модуль связи;
- блок датчика влажности;
- блок датчика температуры;
- блок питания/зарядки;
- разъем зарядки/прошивки устройства;

- модуль индикации состояния устройства.

Вариант исполнения:

Вариант исполнения №1 представлен Фиг. №1;

Микроконтроллер (1) подсоединяется к модулю сбора данных (2) включающему в себя датчики влажности (2.1), датчики температуры (2.2), датчики влажности (2.3), контроллер уровня заряда (2.4), микроконтроллер также подключен к модулю приема/передачи данных (3) (Wi-Fi, BLE, Bluetooth); соединен с модулем питания (4), в качестве которого может выступать - преобразователь или зарядное устройство или устройство защиты источника питания), дополнительно вся эта система имеет источник автономного питания (5). Фиг. №1. принципиальная схема устройства беспроводной контрольно – измерительной системы

Вариант исполнения №2 – Фиг. №2 (Альтернативная схема организации сбора данных):

Микроконтроллер (1), модуль приема/передачи данных (данных Wi-Fi, BLE, Bluetooth) (1.1), включающий в себя датчик уровня заряда (1.1); модуль сбора данных (2), включающий в себя датчик влажности (2.1) и датчик температуры (2.2); модуль питания (3) включающий в себя преобразователь (3.1), зарядное устройство (3.2), устройство защиты источника питания (3.3); источник автономного питания (4).

Фиг. №2. принципиальная схема устройства беспроводной контрольно – измерительной системы используя микроконтроллер с интегрированным беспроводным интерфейсом связи Wi-Fi, Bluetooth, BLE.

Код – №1

```
#include <Adafruit_Sensor.h>
```

```
#include <Adafruit_BME280.h>
```

```
#include "painlessMesh.h"
```

```
#include <Arduino_JSON.h>
```

```
// настройки MESH
```

```
#define MESH_PREFIX "RNTMESH" //введите имя
```

```
#define MESH_PASSWORD "MESHpassword" //и пароль
```

```
#define MESH_PORT 5555 //порт по умолчанию
```

```
//создаем объект BME
```

```

Adafruit_BME280 bme;

//номер ноды
int nodeNumber = 2;

//Строка, которую будем отправлять другим нодам
String readings;

Scheduler userScheduler;

painlessMesh mesh;

// метод-заглушка
void sendMessage() ; // благодаря ему PlatformIO будет работать
String getReadings(); // получение показаний датчика

//Создаем задачи для отправки сообщений и получения показаний
Task taskSendMessage(TASK_SECOND * 5 , TASK_FOREVER, &sendMessage);

String getReadings () {
  JSONVar jsonReadings;
  jsonReadings["node"] = nodeNumber;
  jsonReadings["temp"] = bme.readTemperature();
  jsonReadings["hum"] = bme.readHumidity();
  jsonReadings["pres"] = bme.readPressure()/100.0F;
  readings = JSON.stringify(jsonReadings);
  return readings;
}

```

```

}

void sendMessage () {

String msg = getReadings();

mesh.sendBroadcast(msg);

}

//Запускаем BME280

void initBME(){

if (!bme.begin(0x76)) {

Serial.println("Could not find a valid BME280 sensor, check wiring!");

while (1);

}

}

// Нужно для работы библиотеки painlessMesh

void receivedCallback( uint32_t from, String &msg ) {

Serial.printf("Received from %u msg=%s\n", from, msg.c_str());

JSONVar myObject = JSON.parse(msg.c_str());

int node = myObject["node"];

double temp = myObject["temp"];

double hum = myObject["hum"];

double pres = myObject["pres"];

Serial.print("Node: ");

Serial.println(node);

Serial.print("Temperature: ");

Serial.print(temp);

Serial.println(" C");

```

```

Serial.print("Humidity: ");

Serial.print(hum);

Serial.println(" %");

Serial.print("Pressure: ");

Serial.print(pres);

Serial.println(" hpa");

}

void newConnectionCallback(uint32_t nodeId) {

Serial.printf("New Connection, nodeId = %u\n", nodeId);

}

void changedConnectionCallback() {

Serial.printf("Changed connections\n");

}

void nodeTimeAdjustedCallback(int32_t offset) {

Serial.printf("Adjusted time %u. Offset = %d\n", mesh.getNodeTime(),offset);

}

void setup() {

Serial.begin(115200);

initBME();

//mesh.setDebugMsgTypes( ERROR | MESH_STATUS | CONNECTION | SYNC |
COMMUNICATION | GENERAL | MSG_TYPES | REMOTE ); // выбираем

```

типы mesh.setDebugMsgTypes(ERROR | STARTUP); // установите перед функцией init()
чтобы выдавались приветственные сообщения

```
mesh.init( MESH_PREFIX, MESH_PASSWORD, &userScheduler, MESH_PORT );
```

```
mesh.onReceive(&receivedCallback);
```

```
mesh.onNewConnection(&newConnectionCallback);
```

```
mesh.onChangedConnections(&changedConnectionCallback);
```

```
mesh.onNodeTimeAdjusted(&nodeTimeAdjustedCallback);
```

```
userScheduler.addTask(taskSendMessage);
```

```
taskSendMessage.enable();
```

```
}
```

```
void loop() {
```

```
// функция также запустит планировщик
```

```
mesh.update();
```

```
}
```

Альтернативная реализация ПО №2 – по WiFi:

```
#include <WiFi.h>
```

```
#include <PubSubClient.h>
```

```
#include <Wire.h>
```

```
#include <Adafruit_Sensor.h>
```

```
#include <Adafruit_BME280.h>
```

```
/* --- Настройки сети --- */
```

```
const char* ssid = "BBKIS";
```

```
const char* password = "01234567";
```

```

/* --- адрес сервера --- */
const char* mqtt_server = "192.168.1.12";

float temperature;
float humidity;
float pressure;
/*#include <SPI.h>
#define BME_SCK 18
#define BME_MISO 19
#define BME_MOSI 23
#define BME_CS 5*/

#define SEALEVELPRESSURE_HPA (1013.25)

Adafruit_BME280 bme; // I2C
//Adafruit_BME280 bme(BME_CS); // hardware SPI
//Adafruit_BME280 bme(BME_CS, BME_MOSI, BME_MISO, BME_SCK); // software SPI

/* create an instance of PubSubClient client */
WiFiClient espClient;
PubSubClient client(espClient);

/*LED GPIO pin*/
const char led = 2;

/* topics */
#define TEMP_TOPIC "smarthome/room1/temp"
#define HUMI_TOPIC "smarthome/room1/humidity"
#define PRESS_TOPIC "smarthome/room1/pressure"
#define LED_TOPIC "smarthome/room1/led" /* 1=on, 0=off */

```

```

long lastMsg = 0;
char msg[20];

void receivedCallback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Message received: ");
  Serial.println(topic);

  Serial.print("payload: ");
  for (int i = 0; i < length; i++) {
    Serial.print((char)payload[i]);
  }
  Serial.println();
  /* we got '1' -> on */
  if ((char)payload[0] == '1') {
    digitalWrite(led, LOW);
  } else {
    /* we got '0' -> on */
    digitalWrite(led, HIGH);
  }
}

void mqttconnect() {
  /* Loop until reconnected */
  while (!client.connected()) {
    Serial.print("MQTT connect ...");
    /* client ID */
    String clientId = "ESP32Client";
    /* connect now */
    if (client.connect(clientId.c_str())) {

```

```

Serial.println("connected");
/* subscribe topic with default QoS 0*/
client.subscribe(LED_TOPIC);
} else {
Serial.print("failed, status code =");
Serial.print(client.state());
Serial.println("try again in 5 seconds");
/* Wait 5 seconds before retrying */
delay(5000);
}
}
}

void setup() {
Serial.begin(115200);
// We start by connecting to a WiFi network
Serial.println();
Serial.print("Connecting to ");
Serial.println(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
/* set led as output to control led on-off */
pinMode(led, OUTPUT);

Serial.println("");
Serial.println("WiFi connected");

```

```

Serial.println("IP address: ");
Serial.println(WiFi.localIP());

/* configure the MQTT server with IPaddress and port */
client.setServer(mqtt_server, 1883);
/* this receivedCallback function will be invoked
when client received subscribed topic */
client.setCallback(receivedCallback);

bool status;
// default settings
// (you can also pass in a Wire library object like &Wire2)
status = bme.begin(0x76);
if (!status) {
Serial.println("Could not find a valid BME280 sensor, check wiring!");
while (1);
}
}

void loop() {
/* if client was disconnected then try to reconnect again */
if (!client.connected()) {
mqttconnect();
}
/* this function will listen for incoming
subscribed topic-process-invoke receivedCallback */
client.loop();
/* we measure temperature every 3 secs
we count until 3 secs reached to avoid blocking program if using delay()*/
long now = millis();

```

```

if (now - lastMsg > 10000) {
lastMsg = now;

temperature = bme.readTemperature();
humidity = bme.readHumidity();
pressure = (bme.readPressure() / 100.0F); //в hPa
Serial.print("Влажность = ");
Serial.println(humidity);
/*
Serial.print("Approx. Altitude = ");
Serial.print(bme.readAltitude(SEALEVELPRESSURE_HPA));
Serial.println(" m");
*/
if (!isnan(temperature)) {
/* publish the message */
client.publish(TEMP_TOPIC, String(temperature, 1).c_str());
client.publish(HUMI_TOPIC, String(humidity, 0).c_str());
client.publish(PRESS_TOPIC, String((0.75*pressure), 0).c_str()); // *0,75 - переводим в
мм.рт.ст.
}
else {
ESP.restart();
}
}
}
}

```

Альтернативная реализация ПО №3:

```

#ifndef MY_DEBUG
#define MY_DISABLED_SERIAL
#endif

#define MY_RADIO_NRF5_ESB
#define MY_NRF5_ESB_PA_LEVEL (0x4UL)

```

```

int16_t mtwr;

#define MY_TRANSPORT_WAIT_READY_MS (mtwr)

#define SN "EFEKTA MICRO T&H E-Ink"
#define SV "1.4"

#define TEMP_ID 1
#define HUM_ID 2
#define SIGNAL_Q_ID 100
#define BATTERY_VOLTAGE_ID 101
#define SET_TIME_SEND_ID 102
#define SET_BATT_SEND_ID 103
#define MY_SEND_RESET_REASON 105
#define SET_COLOR_ID 106
#define SET_FONTS_ID 107
#define SET_BIND_ID 108

#include <MySensors.h>

MyMessage msgTemp(TEMP_ID, V_TEMP);
MyMessage msgHum(HUM_ID, V_HUM);
MyMessage sqMsg(SIGNAL_Q_ID, V_VAR1);
MyMessage bvMsg(BATTERY_VOLTAGE_ID, V_VAR1);
MyMessage setTimeSendMsg(SET_TIME_SEND_ID, V_VAR1);
MyMessage setBattSendMsg(SET_BATT_SEND_ID, V_VAR1);
MyMessage sendMsg(MY_SEND_RESET_REASON, V_VAR1);
MyMessage setColor(SET_COLOR_ID, V_VAR1);
MyMessage setFonts(SET_FONTS_ID, V_VAR1);
MyMessage setBind(SET_BIND_ID, V_VAR1);

uint32_t PIN_BUTTON_MASK;
volatile byte buttIntStatus = 0;

```

```
#define APP_GPIOTE_MAX_USERS 1
extern "C" {
#include "app_gpiote.h"
#include "nrf_gpio.h"
}
static app_gpiote_user_id_t m_gpiote_user_id;
```

Оптимальным ПО является Версия ПО № 3, за счет максимального времени работы устройства.

G01K 1/02

G01K 1/14

G01K 13/10

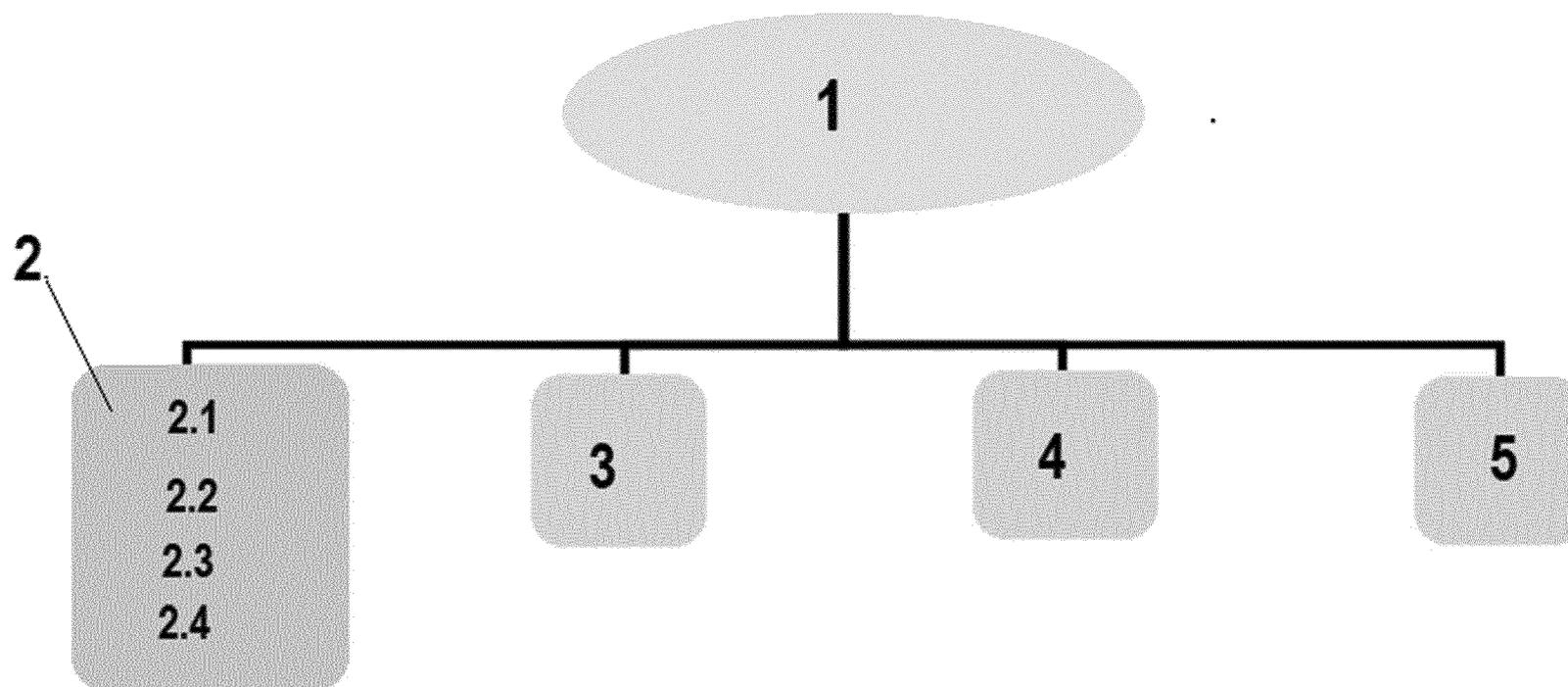
ФОРМУЛА ИЗОБРЕТЕНИЯ

Беспроводная бесконтактная контрольно-измерительная система для измерения вертикального профиля температур и влажности корнеплодов

1. **Беспроводная бесконтактная контрольно-измерительная система для измерения вертикального профиля температур и влажности в буртах при хранении корнеплодов** на основе программно-аппаратного комплекса, состоящая из микроконтроллера, модуля приема/передачи данных Wi-Fi, BLE, Bluetooth, датчика уровня заряда; модуля сбора данных; модуля питания; источника автономного питания и устройства с возможностью введения, анализа и отображения информации, с помощью протокола BLE -Bluetooth Low Energy, позволяет нескольким устройствам связываться друг с другом в одной беспроводной сети, используя картографическую систему датчиков в зависимости от установленной схемы получает информацию о температуре и влажности измеряемого объекта в режиме реального времени,
2. **Беспроводная бесконтактная контрольно-измерительная система для измерения вертикального профиля температур и влажности в буртах при хранении корнеплодов по п.1.** отличающаяся тем, что модулем сбора данных являются датчики: влажности, и датчики температуры;
3. **Беспроводная бесконтактная контрольно-измерительная система для измерения вертикального профиля температур и влажности в буртах при хранении корнеплодов по п.1.** отличающаяся тем, что модулем питания представлены преобразователь или зарядное устройство или устройство защиты источника питания.

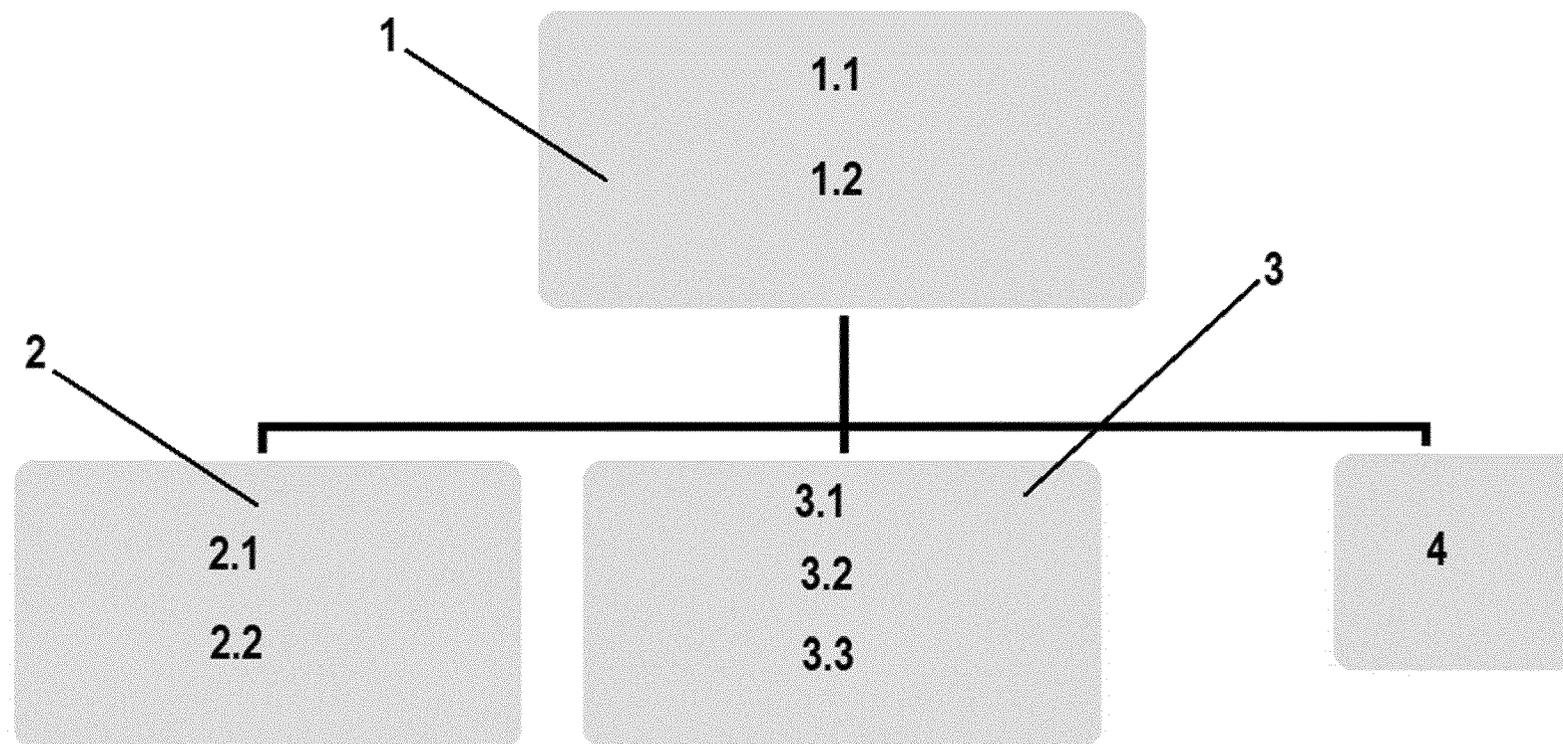
ЧЕРТЕЖИ

Беспроводная бесконтактная контрольно-измерительная система для измерения вертикального профиля температур и влажности корнеплодов /овощехранилищ/зернохранилищ



Фиг. 1 Схема устройства беспроводной контрольно-измерительной системы

Беспроводная бесконтактная контрольно-измерительная система для измерения вертикального профиля температур и влажности
корнеплодов /овощехранилищ/зернохранилищ



Фиг. №2. принципиальная схема устройства беспроводной контрольно – измерительной системы используя микроконтроллер с интегрированным беспроводным интерфейсом связи Wi-Fi , Bluetooth, BLE

ОТЧЕТ О ПАТЕНТНОМ ПОИСКЕ
(статья 15(3) ЕАПК и правило 42 Патентной инструкции к ЕАПК)

Номер евразийской заявки:

202390482

А. КЛАССИФИКАЦИЯ ПРЕДМЕТА ИЗОБРЕТЕНИЯ:

МПК:

A01F 25/02 (2006.01)
A01F 25/16 (2006.01)
G08C 17/02 (2006.01)
H04W 4/02 (2018.01)
H04W 84/18 (2009.01)

СПК:

A01F 25/02
A01F 25/16
Y02A 40/51
G08C 17/02
H04W 4/023
H04Q 2209/40
H04W 84/18

Б. ОБЛАСТЬ ПОИСКА:

Просмотренная документация (система классификации и индексы МПК)

A01F 25/00, 25/02, 25/16, G05D 27/02, G06F 13/00, G08C 17/00, 17/02, 21/00, H04W 4/02, 84/00, 84/18

Электронная база данных, использовавшаяся при поиске (название базы и, если, возможно, используемые поисковые термины)

ЕАПАТИС, Espacenet, Google Patents, «Поисковая платформа» Роспатент

В. ДОКУМЕНТЫ, СЧИТАЮЩИЕСЯ РЕЛЕВАНТНЫМИ

Категория*	Ссылки на документы с указанием, где это возможно, релевантных частей	Относится к пункту №
Y	WO 2012/100773 A1 (WEBSTECH APS и др.) 2012.08.02, реферат, со строки 5 стр. 3 по строку 7 стр. 10, фиг. 7	1-3
Y	RU 2732678 C1 (ООО «ИННОВАТИКА+») 2020.09.21, реферат, пункты 1, 13, 14 и 23 формулы	1-3
A, D	RU 211225 U1 (ООО «ДАРВИН») 2022.05.26, весь документ	1-3
A	WO 2008/151635 A1 (UNIV AARHUS и др.) 2008.12.18, реферат, формула	1-3
A	CN 206629484 U (UNIV TIANJIN SCIENCE & TECH) 2017.11.14, реферат, аб. 15, 20	1-3

последующие документы указаны в продолжении

* Особые категории ссылочных документов:

«А» - документ, определяющий общий уровень техники

«D» - документ, приведенный в евразийской заявке

«E» - более ранний документ, но опубликованный на дату подачи евразийской заявки или после нее

«O» - документ, относящийся к устному раскрытию, экспонированию и т.д.

"P" - документ, опубликованный до даты подачи евразийской заявки, но после даты испрашиваемого приоритета"

«Т» - более поздний документ, опубликованный после даты приоритета и приведенный для понимания изобретения

«Х» - документ, имеющий наиболее близкое отношение к предмету поиска, порочащий новизну или изобретательский уровень, взятый в отдельности

«У» - документ, имеющий наиболее близкое отношение к предмету поиска, порочащий изобретательский уровень в сочетании с другими документами той же категории

«&» - документ, являющийся патентом-аналогом

«L» - документ, приведенный в других целях

Дата проведения патентного поиска: **08/11/2023**

Уполномоченное лицо:

Начальник отдела механики,
физики и электротехники

 Д.Ф. Крылов