IT АКАДЕМИЯ
SAMSUNG

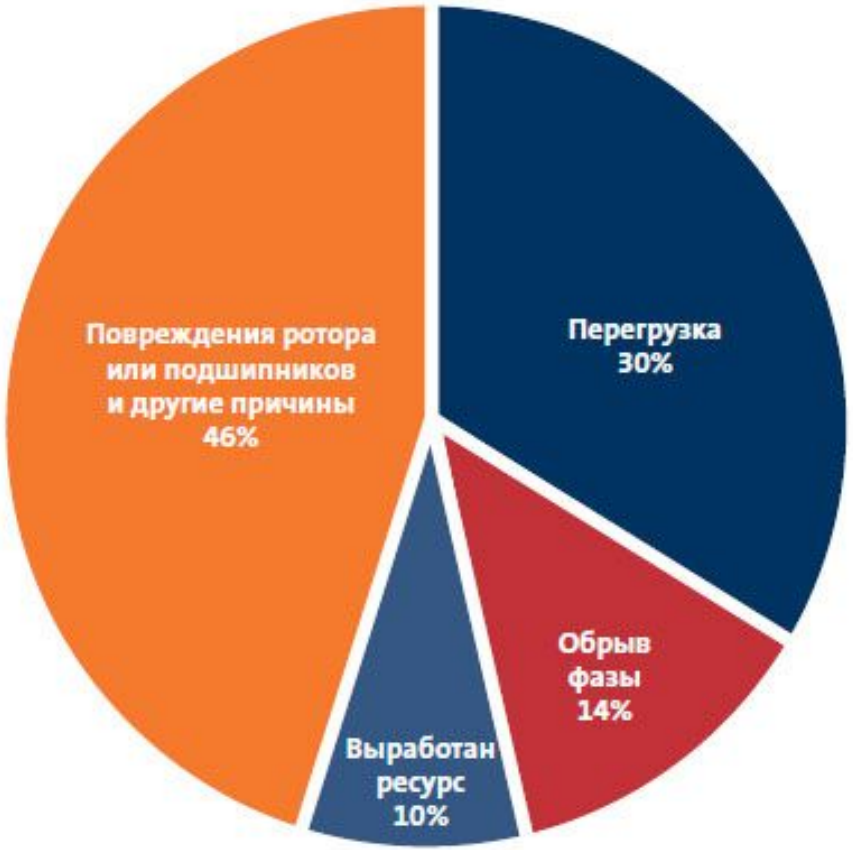
Система акустического и вибрационного мониторинга технического состояния электродвигателя

Проблематика

IT АКАДЕМИЯ SAMSUNG

Сегодня в мире выпускается до 7 млрд электродвигателей. Но в результате аварий ежегодно выходят из строя до 10% применяемых электрических машин.

Выход из строя электроприводов приводит к большому материальному ущербу, связанному с ремонтом и простоем технологического оборудования.



Основные причины отказа электродвигателей, в соответствии с Американской ассоциацией электрических исследований США (Electrical Research Association).

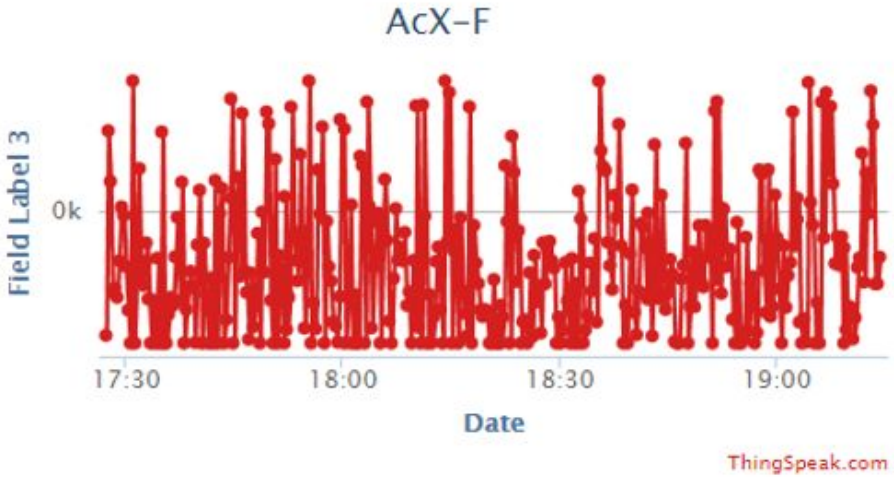
Решение

IT АКАДЕМИЯ SAMSUNG

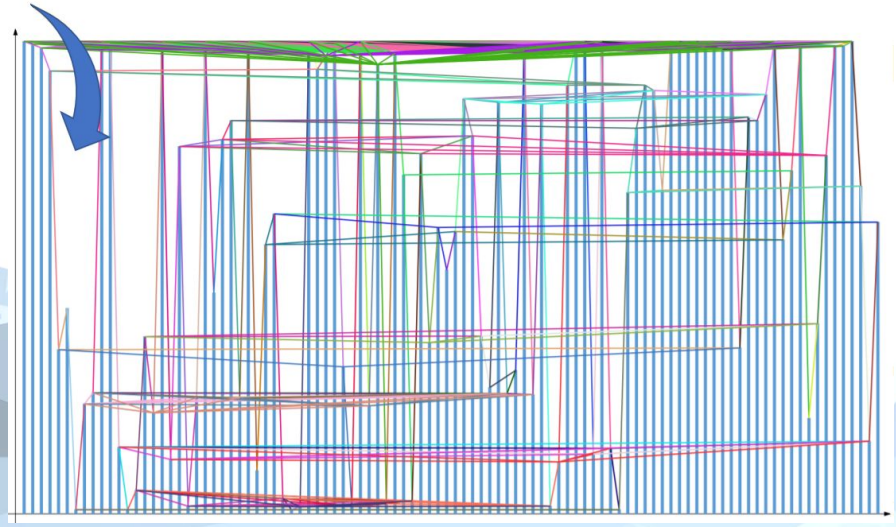
Разработка системы акустического и вибрационного мониторинга технического состояния электродвигателя, для своевременного определения и устранения неисправностей.

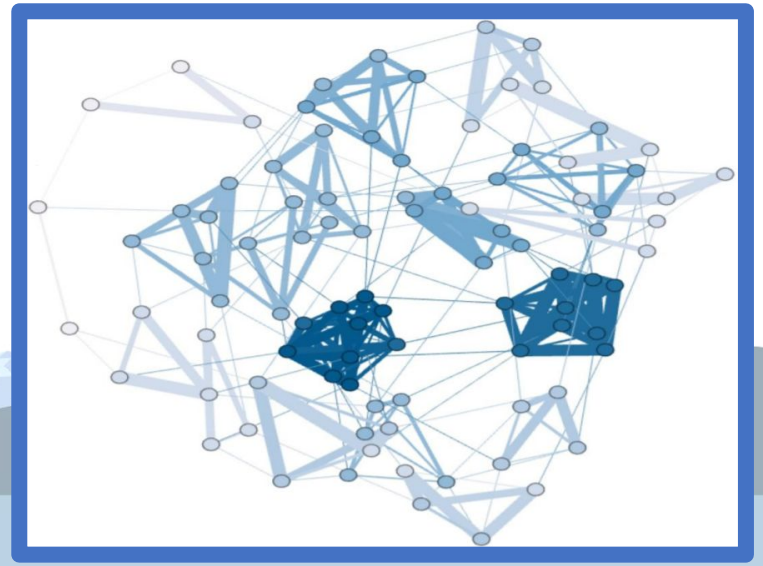
Обмен данными осуществляется по протоколу MQTT по беспроводной сети.

Научная новизна



Для обработки данных с устройства используется сетевой метод анализа нелинейных временных рядов.





Модель конвертирования временных рядов в сети.

Ценность предложения

IT АКАДЕМИЯ SAMSUNG

- Прогнозирование выхода из строя электродвигателей.
- Исключение остановок производства для диагностики.
- Эффективное распределение ресурсов служб технического обслуживания.
- Минимизация издержек из-за выхода из строя оборудования.

Цели и задачи

IT АКАДЕМИЯ SAMSUNG

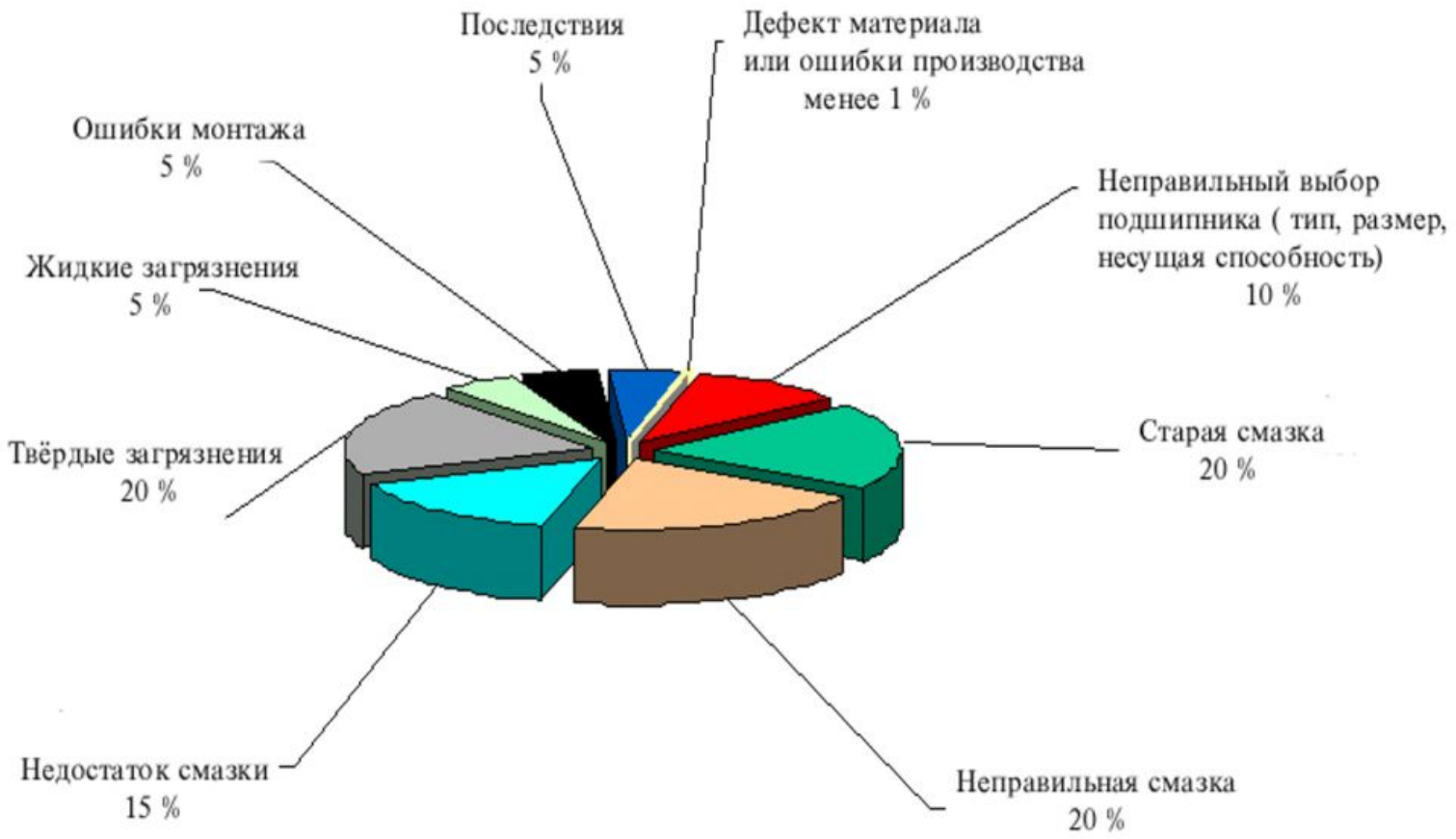
Цели:

- Разработать систему акустического и вибрационного мониторинга технического состояния электродвигателя.

Задачи:

- Определить функционал проектируемого устройства.
- Спроектировать и изготовить печатную плату устройства.
- Разработать технологию фиксации датчиков на корпусе электродвигателя.
- Собрать функциональный прототип для сбора данных.
- Разработать математическую модель для анализа данных.
- Сравнить с аналогами.

Причины выхода из строя



Показатели износа подшипников

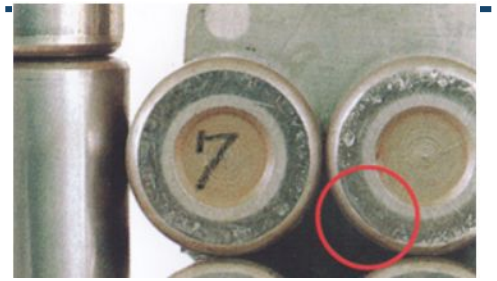
IT АКАДЕМИЯ SAMSUNG

Отслаивание

Локализация:
Внутреннее кольцо сферического роликоподшипника
Причина:
Чрезмерная осевая нагрузка

Задиры

Локализация:
Ролики двухрядного цилиндрического роликоподшипника
Причина: Недостаточное количество смазки и чрезмерная осевая нагрузка

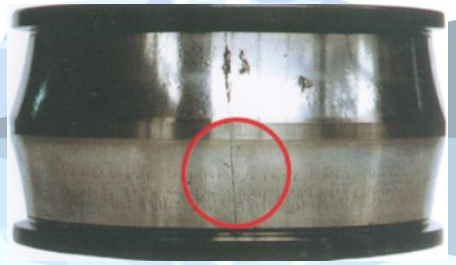


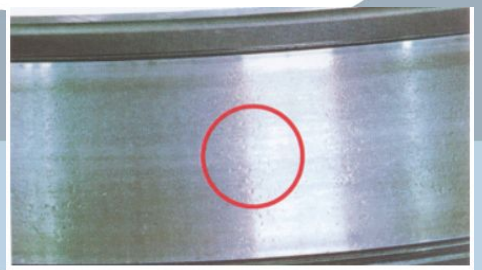
Размытие поверхности

Локализация:Наружное кольцо цилиндрического роликоподшипника
Причина:
Недостаточная радиальная нагрузка, проскальзывание роликов, вызванное чрезмерным количеством смазки

Вмятины и трещины

Локализация:
Внутреннее кольцо конического роликоподшипника
Причина:
Попадание инородных веществ и грязи на поверхность





Показатели износа подшипников

IT АКАДЕМИЯ SAMSUNG

Фреттинг-коррозия

Локализация:
Внутреннее кольцо радиального шарикоподшипника
Причина:
Вибрация

Ложное бринеллирование

Локализация:
Внутреннее кольцо радиального шарикоподшипника
Причина:
Вибрации от внешнего источника при стационарном положении

Проскальзывание

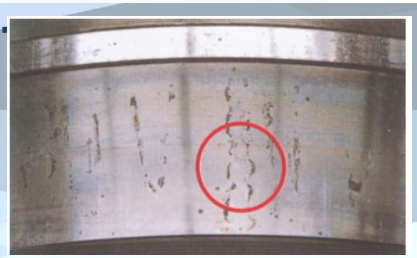
Локализация:
Внутреннее кольцо сферического роликоподшипника
Причина:Недостаточный натяг

Электрическая коррозия

Локализация:
Внутреннее кольцо конического роликоподшипника
Причина:Разность электрических потенциалов внутреннего и наружного колец

Ржавчина и коррозия

Локализация:
Внутреннее кольцо сферического роликоподшипника
Причина:
Попадание воды в смазку

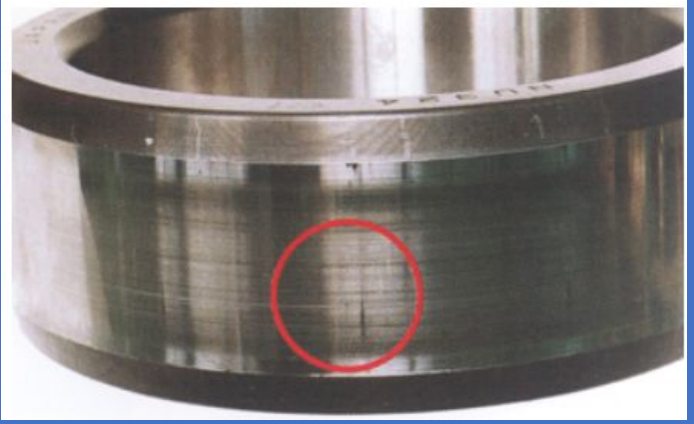


Показатели износа подшипников

IT АКАДЕМИЯ SAMSUNG

Ошибки при монтаже

Локализация:
Внутреннее кольцо цилиндрического роликоподшипника
Признак:
Осевые царапины на поверхности качения
Причина:
Наклон внутреннего и наружного колец во время монтажа

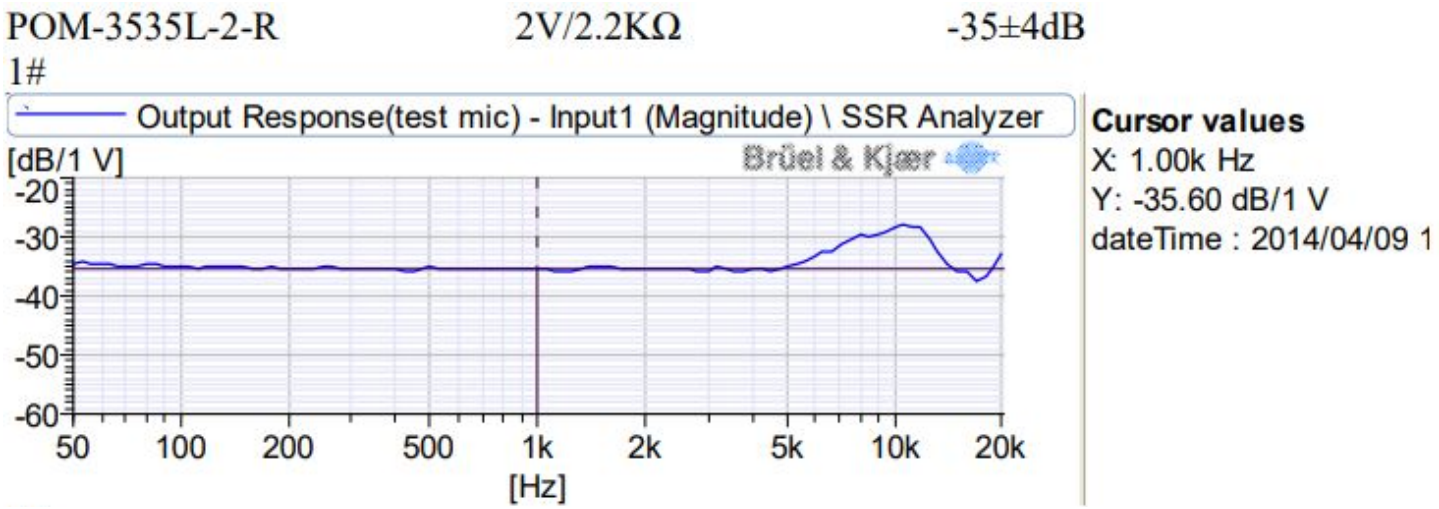


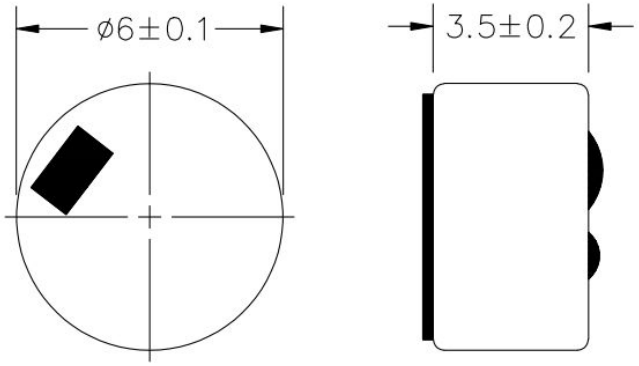
Обесцвечивание

Локализация:
Внутреннее кольцо шарикоподшипника
Признак:
Появление голубого или фиолетового оттенка на поверхности дорожки качения
Причина:
Тепловыделение, вызванное недостаточной смазкой

Подбор микрофона

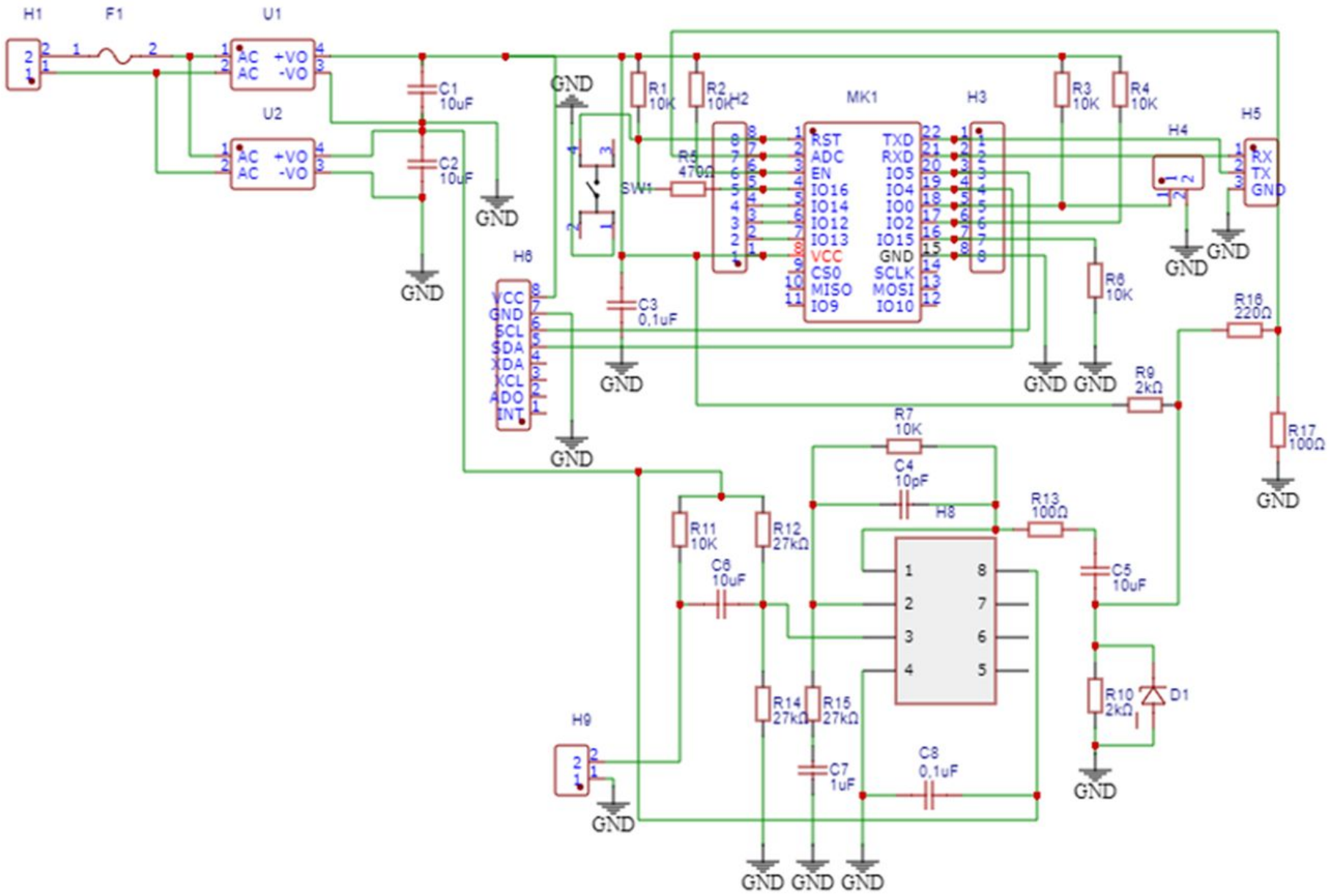
IT АКАДЕМИЯ SAMSUNG





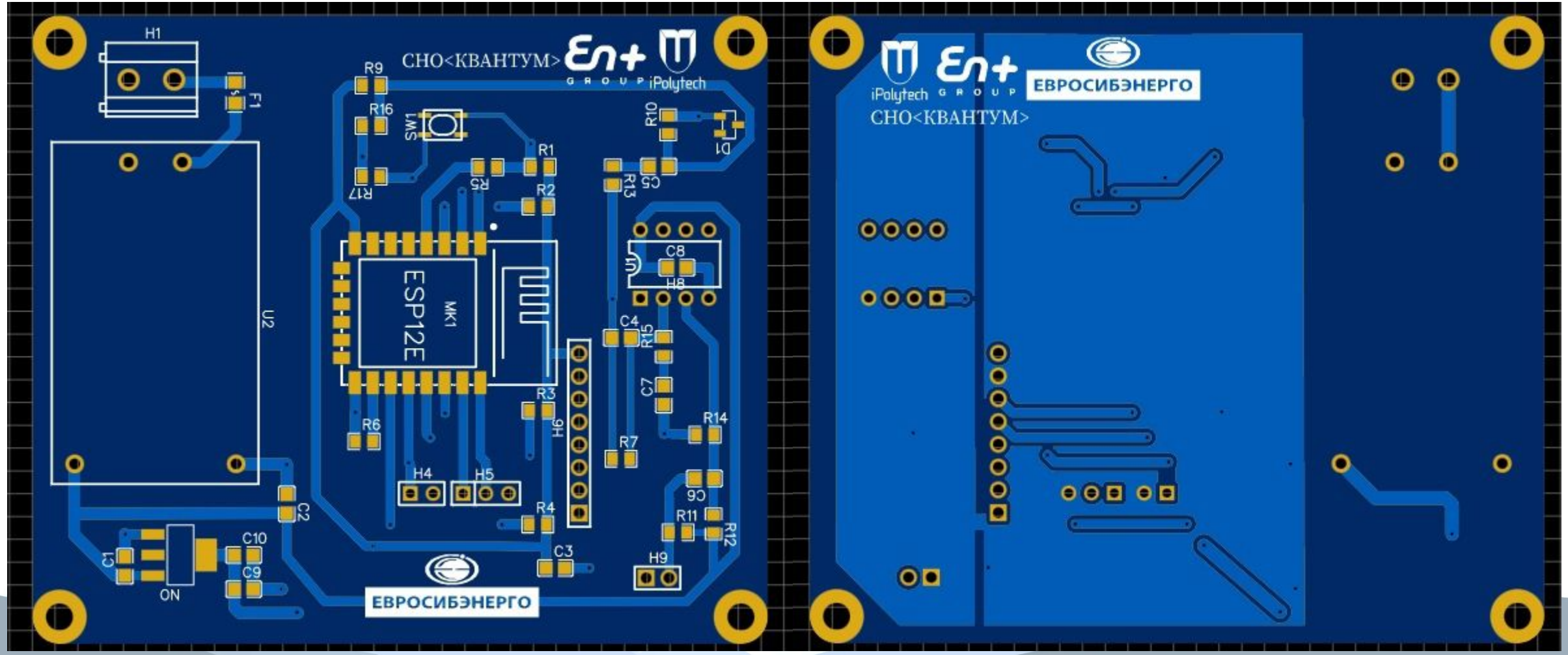
PARAMETERS	VALUES	UNITS
DIRECTIVITY	OMNI	–
SENSITIVITY	−35±4	dB
STANDARD OPERATING VOLTAGE	2	Vdc
MAX OPERATING VOLTAGE	10	Vdc
CURRENT CONSUMPTION (MAX)	0.5	mA
IMPEDANCE	2.2	kOhm
SIGNAL TO NOISE RATIO (MIN.)	70	dB
TERMINAL	SOLDER PAD	–
INTERNAL CAPACITOR	N/A	pF

Разработка схемы

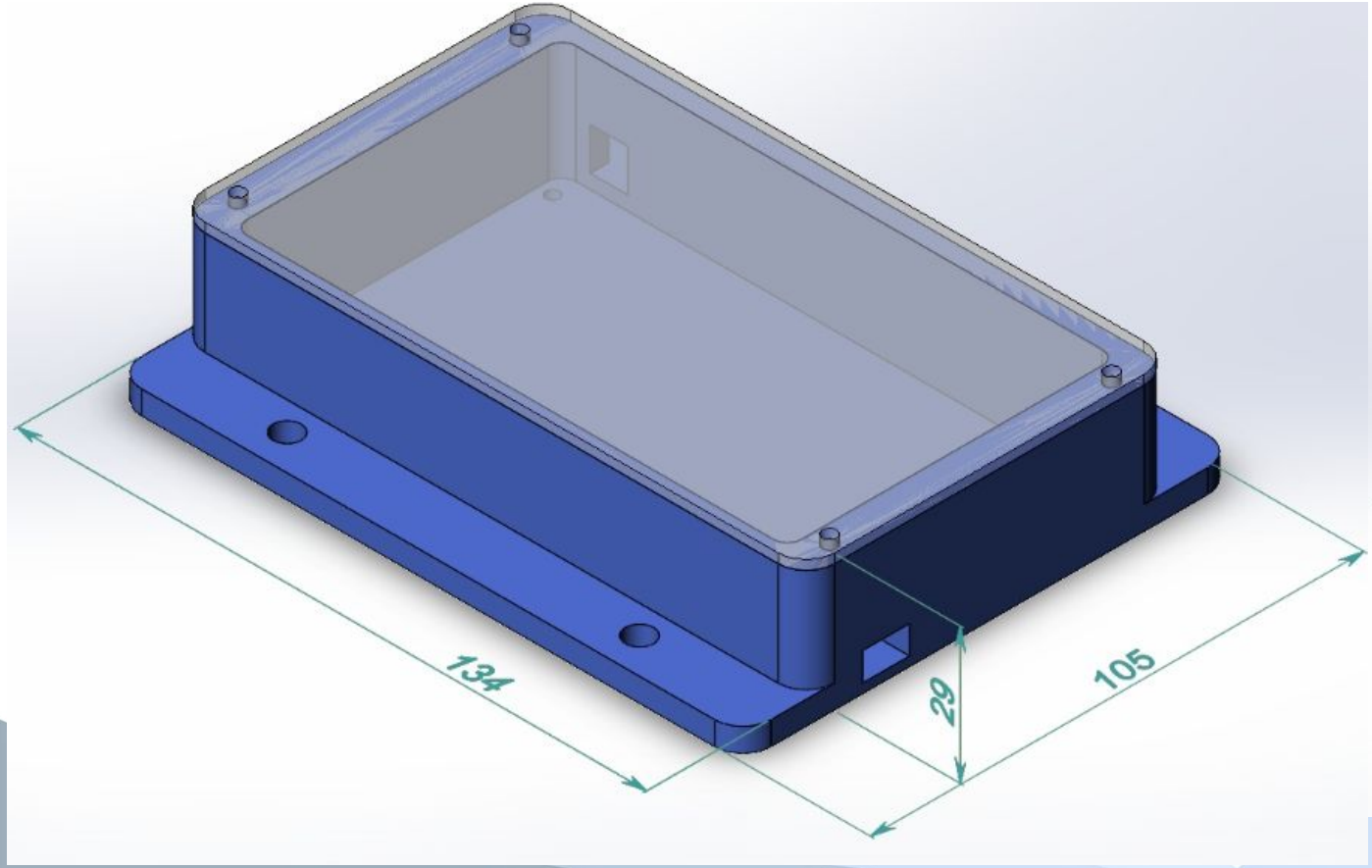


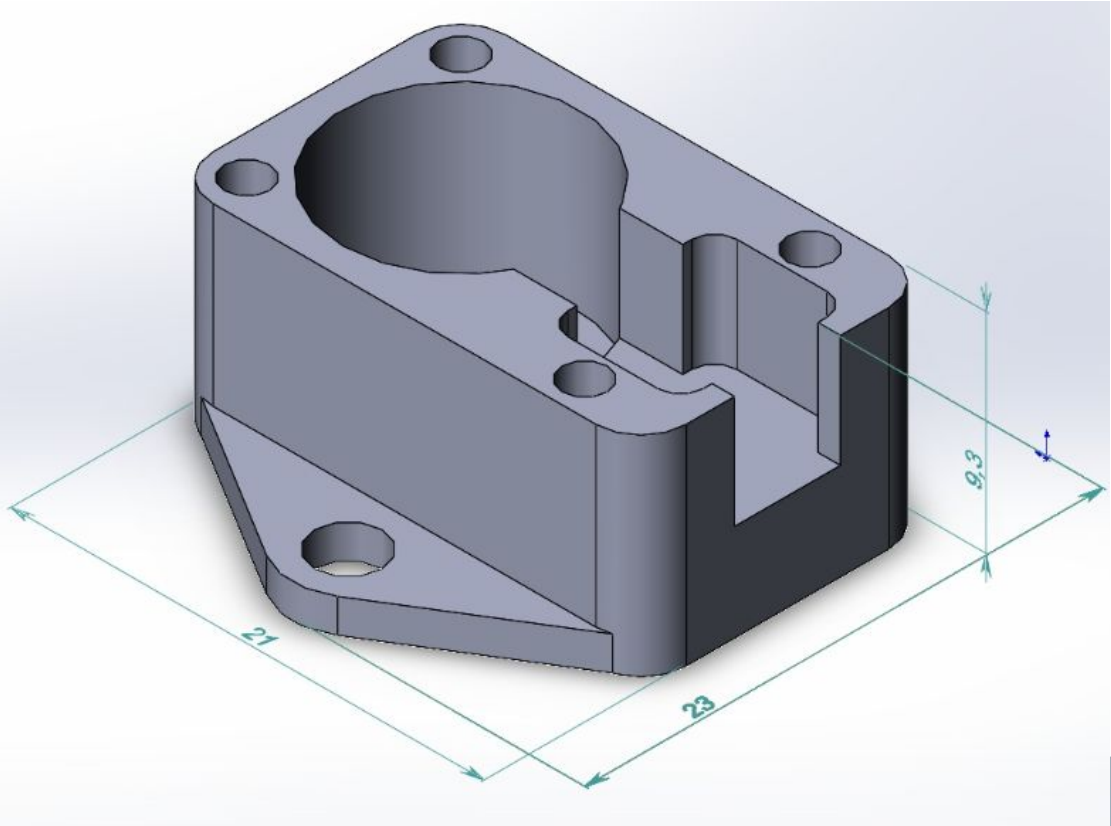
Разводка платы

IT АКАДЕМИЯ SAMSUNG

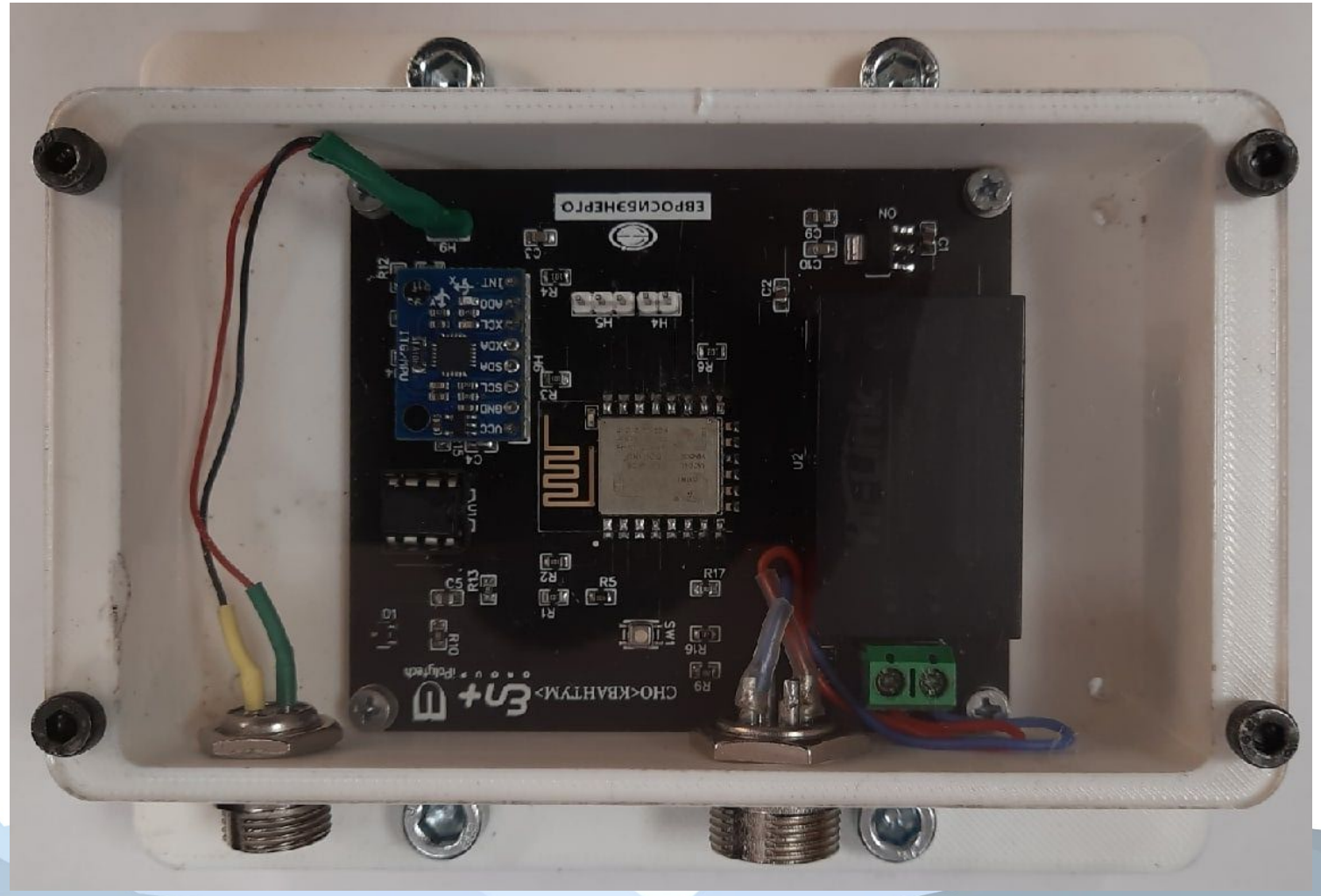


Разработка крепления



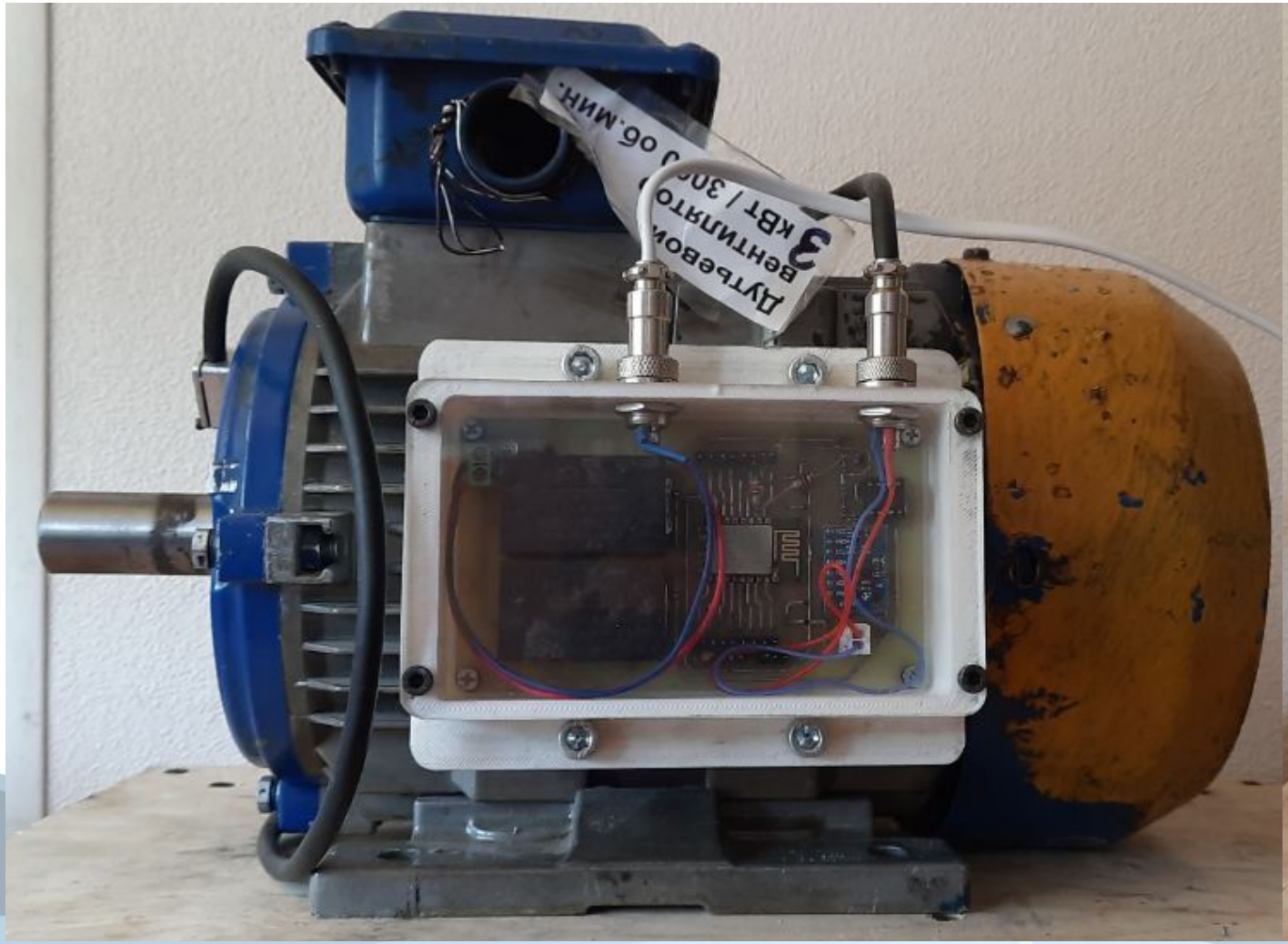


Крепление и плата в сборе



Крепление на электродвигатель

IT АКАДЕМИЯ SAMSUNG



Разработка ПО

IT АКАДЕМИЯ SAMSUNG

```
#include <Wire.h>
#include <ESP8266WiFi.h>
#include <ESP8266mDNS.h>
#include <WiFiUdp.h>
#include <ArduinoOTA.h>

//----------------------Настройки передачи данных-----------------
String apiKey = "JJDF988985JRJF77";           //API канала
const char *ssid =  "name";                 //имя интернета
const char *pass =  "pasword";             //пароль от интернета
const char* server = "api.thingspeak.com";    //сайт для отправки данных
WiFiClient client;                            //создаёт клиента для подключение к сети
//-----------------------------------------------------------------
#define WINDOW_SIZE 5                                  //Размер окна для фильтра скользящего среднего
int SUM,READUNGS[WINDOW_SIZE],INDEX=0;        //Переменные для фильтра скользящего среднего
int SUM1[3],READUNGS1[3][WINDOW_SIZE],INDEX1=0;
const int MPU_addr=0x68;                      //I2C address of the MPU-6050 (для акслерометра: SDA - D2, SCL - D1)
float AcX,AcY,AcZ,Accel[3],sounds[2];                      //переменные для акселерометра и микрофона(A0)
unsigned long tim;

void setup(){
  Serial.begin(9600);
  WiFi.begin(ssid, pass);                                  //инициализация работы вайфая
  while (WiFi.status() != WL_CONNECTED) delay(500);        //ожидание подключения к сети
  Wire.begin();                                            //инициализация библиотеки Wire
  Wire.beginTransmission(MPU_addr);                        //начало отправки данных на MPU
  Wire.write(0x6B);                                        // PWR_MGMT_1 register (пока не измеряет выключен)
  Wire.write(0);                                           // установка 0 (активируем MPU-6050)
  Wire.endTransmission(true);                              //окончание отправки
  ArduinoOTA.begin();
}

//------------------Функция измерения-----------------------------
void measure(){
  Wire.beginTransmission(MPU_addr);        //начинает передачу данных на MPU
  Wire.write(0x3B);                        //какие данные отправяться
  Wire.endTransmission(false);             //окончание отправки
  Wire.requestFrom(MPU_addr,14,true);      //запрос байт у ведомого устройства
  AcX=float(Wire.read()<<8|Wire.read());          //запись данных акселерометра
  AcY=float(Wire.read()<<8|Wire.read());
  AcZ=float(Wire.read()<<8|Wire.read());
  AccelAverage(AcX,AcY,AcZ);
  sounds[0]=SoundAverage(analogRead(A0));
  sounds[1]=analogRead(A0);
}
```

Разработка ПО

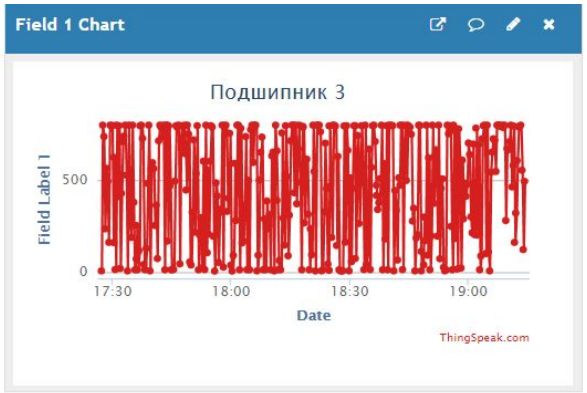
IT АКАДЕМИЯ SAMSUNG

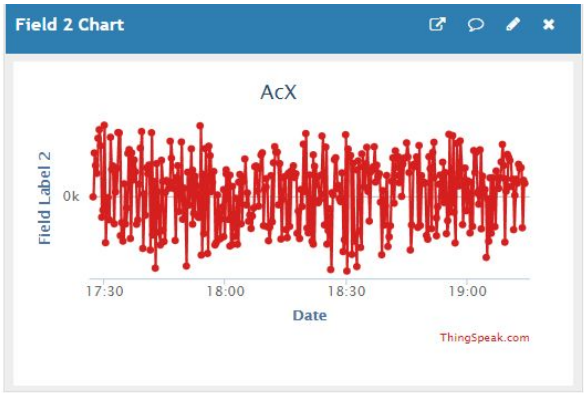
```
//-----------------Основная-----------------------------------------
void loop(){
  ArduinoOTA.handle();
  measure();                          //функция измерения
  if(millis()-tim>=15000){            //задержка 15 секунд
    if (client.connect(server,80)){
      String DATA = apiKey;
      DATA +="&field1=";            //номер графика для отправки
      DATA += String(sounds[0]);       //переменаая для отправки
      DATA +="&field2=";            //номер графика для отправки
      DATA += "\r\n\r\n";           //окончание записи данных

      client.print("POST /update HTTP/1.1\n");                          //начало отправки данных на сервер
      client.print("Host: api.thingspeak.com\n");
      client.print("Connection: close\n");
      client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");
      client.print("Content-Type: application/x-www-form-urlencoded\n");
      client.print("Content-Length: ");
      client.print(DATA.length());
      client.print("\n\n" );
      client.print(DATA);                                          //конец отправки данных на сервер
      tim=millis();                                                //запись текущего времени работы
    }
  }
}

//---------------фильтры скользящего среднего-----------------------
float SoundAverage(int a){
  SUM-=READUNGS[INDEX];
  READUNGS[INDEX]=a;
  SUM+=READUNGS[INDEX];
  INDEX=(INDEX+1)%WINDOW_SIZE;
  return (SUM/WINDOW_SIZE);
}

void AccelAverage(float a, float b,float c){
  float res[]={a,b,c};
  for(int i=0;i<3;i++){
    SUM1[i]-=READUNGS1[i][INDEX1];
    READUNGS1[i][INDEX1]=res[i];
    SUM1[i]+=READUNGS1[i][INDEX1];
    INDEX1=(INDEX1+1)%WINDOW_SIZE;
    Accel[i]=SUM1[i]/WINDOW_SIZE;
  }
```

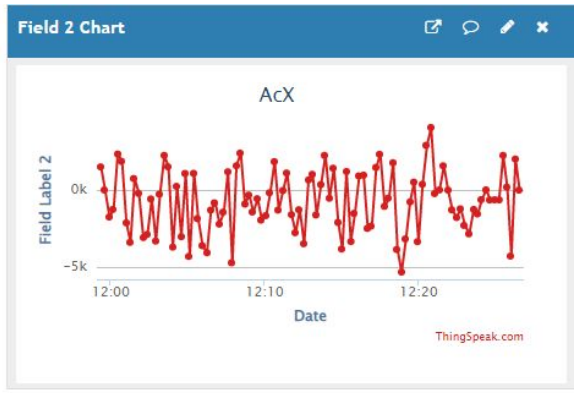
Выходные данные

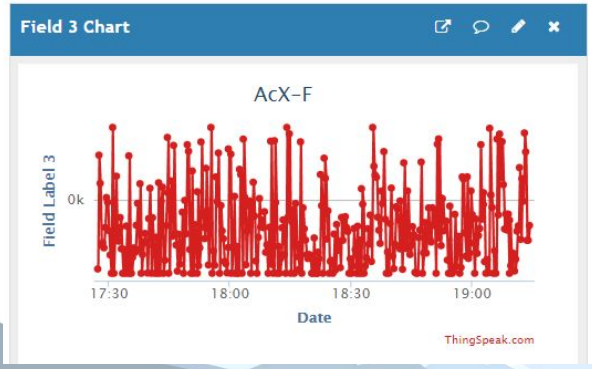
IT АКАДЕМИЯ SAMSUNG

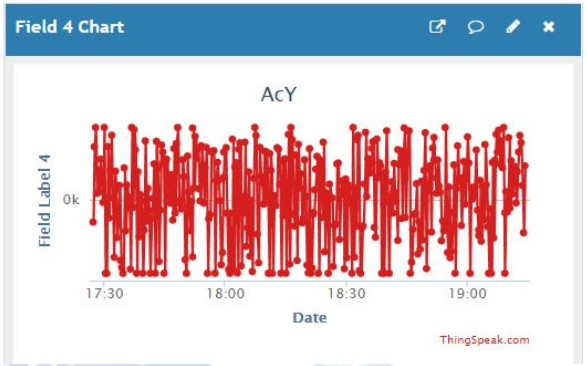
Запись результатов по Wi-fi на сервер

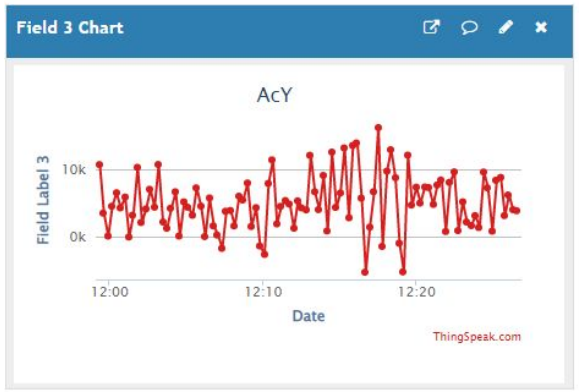


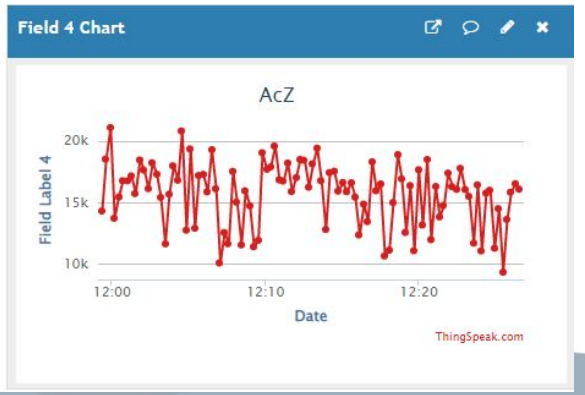












Конкуренция

IT АКАДЕМИЯ SAMSUNG

	VAMS	Weg Motor Scan	ABB Ability™ Smart Sensor	REMM	AnomAlert
Электрические параметры	☐	☐	✔	✔	✔
Температура	☐	✔	✔	✔	☐
Вибрации	✔	✔	✔	✔	☐
Мобильное приложение	☐	✔	✔	☐	☐
Web-сервис	✔	✔	✔	☐	☐
Приложение для ПК	☐	✔	✔	✔	✔
Система оповещения	✔	✔	✔	✔	✔
Анализ и предсказание выхода оборудования из строя	✔	✔	✔	✔	✔
Масштабирование	✔	✔	нет данных	☐	✔
Сбор статистики	✔	✔	✔	✔	✔
Стоимость, $	200	420	620	4900	2000

Дорожная карта

IT АКАДЕМИЯ SAMSUNG

Концептуальная проработка

07.02.2022 – 29.02.2022

Разработка технического задания

03.03.2022 – 30.06.2022

Создание прототипа

05.04.2022 – 12.06.2022

Испытания прототипа

06.03.2022 – 15.11.2022

Доработка

14.11.2022 – 28.11.2022

Подготовка к производству

31.11.2022 – 23.12.2022

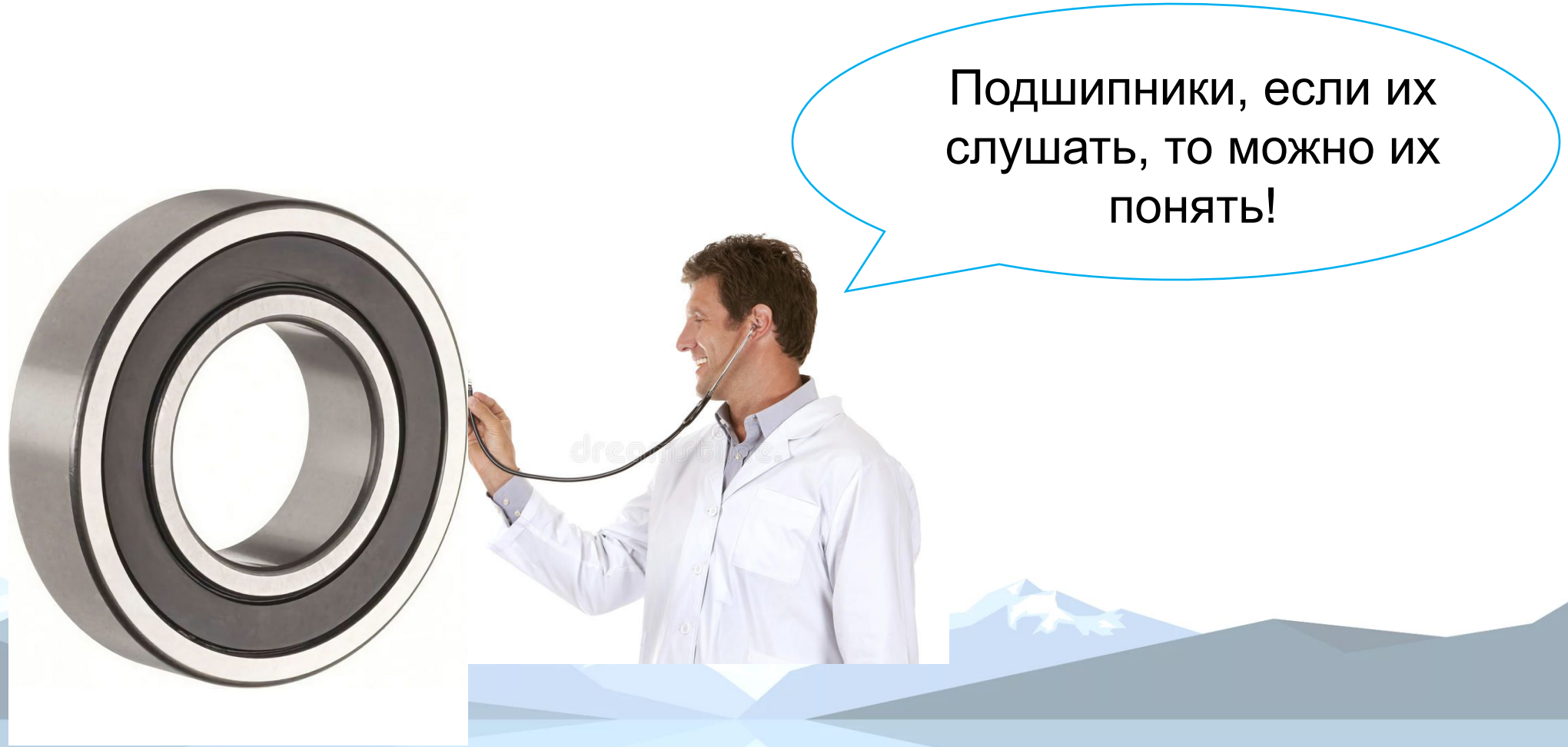
Команда

IT АКАДЕМИЯ SAMSUNG

Кононенко Роман Владимирович	**Волошин Никита Анатольевич**	**Кравчук Данил Александрович**	**Левин Йонатан Игорь**
Руководитель, кандидат технических наук, доцент	Программист, студент 3 курса	Схемотехник, студент 4 курса	Инженер-конструктор, студент 4 курса

Заключение

IT АКАДЕМИЯ SAMSUNG

В ходе работы было разработана система контроля и диагностики технического состояния электродвигателя в реальном времени. Система соответствует заявленному функционалу. В дальнейшем будут производится испытания и отладка данной системы в условиях производства.

Спасибо за внимание