

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»



Инженерная школа информационных технологий и робототехники
Направление 09.03.01 «Информатика и вычислительная техника»
Отделение информационных технологий

Отчёт по теме
«Разработка мобильного устройства контроля состояния трубопровода»
по дисциплине «Творческий проект»

Выполнил:
студент гр. 8B11

____.____.____

Елькин В.Е.

Проверил:
Кандидат технических наук отделения ИТ
оценка (до 30 баллов):
к защите допускаю

____.____.____

Мыцко Е. А.

1. ВВЕДЕНИЕ

В настоящее время предприятия нефтегазовой отрасли стремятся осваивать и внедрять новые технологии, направленные на повышение эффективности, безопасности производственного процесса и снижение общих затрат. В технологических процессах транспортировки нефти и нефтепродуктов применяют различные виды средств измерения параметров трубопровода, что позволяет создавать автоматизированное производство и приводит к цифровизации.

Многие составляющие задействованных в транспортировке нефти сооружений, как и сами процессы, требуют мониторинга, но его проведение непосредственно обслуживающим персоналом является весьма неэффективным, такой процесс склонен к неточностям. Более правильный подход заключается в постоянном мониторинге инженерных сооружений, в частности трубопровода, что необходимо для поддержания безаварийной эксплуатации и обеспечения практически непрерывных потоков данных о его состоянии и функционировании.

В связи с этим целью работы является разработка мобильного устройства, позволяющего получать информацию о состоянии трубопроводной системы в режиме реального времени.

Для достижения данной цели необходимо решить следующие задачи:

- Разработка прототипа автономного устройства;
- Разработка принципиальной схемы;
- Разработка корпуса устройства;
- Реализовать ячеистую сеть;
- Реализовать передачу данных в интернет с конечного узла;
- Опытно-промышленные испытания устройства по измерению параметров работы трубопровода (измерение температуры, параметров вибрации и определение пространственного положения объекта).

СОДЕРЖАНИЕ

1. ВВЕДЕНИЕ	2
СОДЕРЖАНИЕ	3
2. РАЗРАБОТКА УСТРОЙСТВА	4
2.1 Топологии сетей.....	4
2.2 ЭЛЕМЕНТЫ СИСТЕМЫ И ПРИНЦИПИАЛЬНАЯ СХЕМА УСТРОЙСТВА.....	5
2.3. СОЗДАНИЕ САЙТА ПРОЕКТА	10
2.3 АВТОНОМНОСТЬ УСТРОЙСТВА	12
3. ОПЫТНО-ПРОМЫШЛЕННЫЕ ИСПЫТАНИЯ УСТРОЙСТВА.....	14
3.1 КОНТРОЛЬ ПРОХОЖДЕНИЯ ОЧИСТНОГО УСТРОЙСТВА ПО ПАРАМЕТРАМ ВИБРАЦИИ	14
3.2 СОЗДАНИЕ БАЗЫ КОЛЕБАНИЙ ТРУБОПРОВОДА В ПРОЦЕССЕ ЭКСПЛУАТАЦИИ.....	16
2.4.3 Контроль температуры трубопровода.....	17
2.4.4 Контроль планово-высотного положения трубопровода	18
4. ЗАКЛЮЧЕНИЕ.....	20
5. ССЫЛКИ	21
6. ПРИЛОЖЕНИЕ	22
Приложение 1. Код работы узла ячеистой сети.	22
Приложение 2. Код работы конечного узла ячеистой сети.	28
Приложение 3. Код отправки данных на сервер.	30
Приложение 4. Код визуализации данных в виде графика.	31

2. РАЗРАБОТКА УСТРОЙСТВА

ТОПОЛОГИИ СЕТЕЙ

Топология – это схема соединения различных устройств в одной сети. С математической точки зрения, каждую сеть можно рассматривать в качестве графе, в котором вершины (узлы сети) - компьютеры, рёбра - связь между узлами (беспроводное, проводное). В компьютерных сетях есть несколько базовых топологий:

- Полносвязная – топология, в которой узлы образуют полный граф, т.е. каждый компьютер соединен друг с другом. Основной недостаток: при большом количестве устройств, сеть трудно реализуема.
- Ячеистая – вариант полносвязной топологии, в которой удалены некоторые соединения.
- Звезда. В данной топологии компьютеры не подключаются друг к другу напрямую, а через центральный узел (роутер). Передача данных между двумя компьютерами также происходит через центральный узел.
- Кольцо. Каждый компьютер соединяется с двумя соседними.
- Дерево. Компьютеры и сетевое оборудование соединяются в дерево, где роутер – корень, остальные узлы – листья.
- Общая шина. Все компьютеры подключены к одному проводу и каждый имеет данные с другого компьютера.

Проанализировав все основные виды топологий, была выбрана смешанная топология для реализации проекта, состоящий из ячеистой и звезды.

2.2 ЭЛЕМЕНТЫ СИСТЕМЫ И ПРИНЦИПИАЛЬНАЯ СХЕМА УСТРОЙСТВА

В рамках проекта разработано устройство для контроля параметров работы трубопровода. Устройство предназначено для использования в нефтегазовой отрасли в рамках цифровизации производства с целью обеспечения высокой точности измерения параметров вибрации, температуры и определения положения в пространстве объектов в жестких условиях (высокие и низкие температуры, интенсивное электромагнитное и статическое поле и т.д.).

Конструктивно устройство включает в себя модули (рисунок 1):

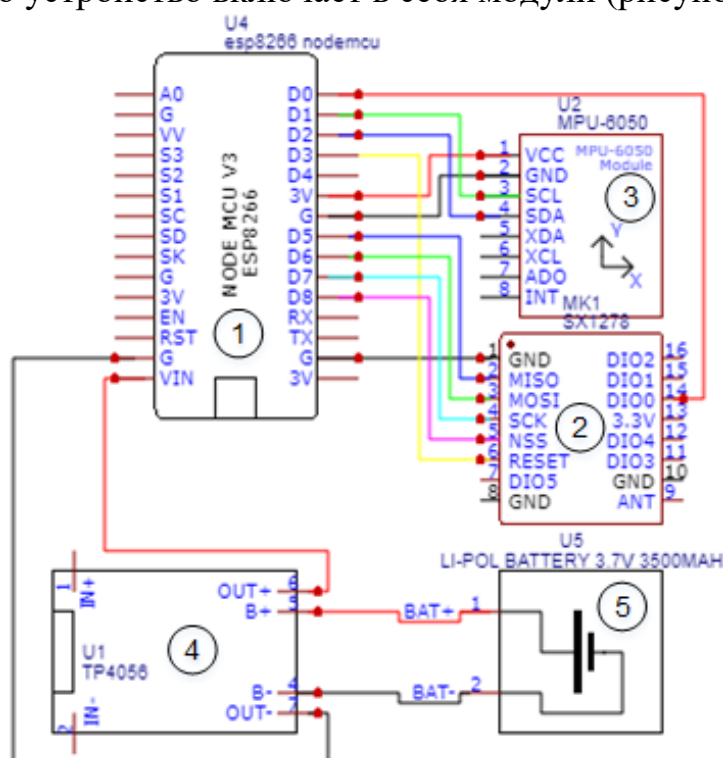


Рисунок 1 – Принципиальная электрическая схема устройства: 1) Микроконтроллер NodeMCU v3 с поддержкой Wi-Fi-интерфейса ESP8266, 2) Модуль передатчика LoRa в диапазоне 868 МГц, 3) Модуль осевого гироскопа и акселерометра MPU-6050 на базе плат Arduino, 4) Модуль зарядного устройства (ЗУ) для литиевых АКБ на TP4056, 5) Литий-полимерный аккумулятор

- Микроконтроллер NodeMCU v3 с поддержкой Wi-Fi-интерфейса ESP8266.

Управление работой всей системы осуществляет микроконтроллер, по его команде происходит аналого-цифровое преобразование сигнала, интегрирование измеряемой величины, цифровая фильтрация и выдача измеренного значения в нормированном виде.

- Модуль приемопередатчика LoRa.

Сеть Lora предназначена для решения задач телеметрии, не связанных с передачей больших объемов информации, а также для управления оконечными устройствами. Под телеметрической информацией подразумевается сбор и передача информации о состоянии любого рода датчиков или устройств.

Для беспроводной передачи результатов измерений от датчиков оказалось оптимальным использование сети LoRaWAN 1.0.2, представляющей собой один из типов LPWAN сетей, расшифровывается как Low-power Wide-area Network – «энергоэффективная сеть дальнего радиуса действия».

LPWAN сети являются беспроводными и имеют широкий радиус охвата, основным преимуществом таких сетей является низкое энергопотребление, а объем передачи данных в таких сетях измеряется в байтах, но этого достаточно, чтобы передавать необходимую телеметрию с конечного устройства на сервер диспетчера.

Основные преимущества беспроводных сетей LoRaWAN обусловлены использованием широкополосной модуляции LoRa и безлицензионных диапазонов частот. Преимущества LoRaWAN:

- совместимость с существующими сетями/технологиями беспроводной передачи данных;
- высокая помехоустойчивость;
- способны обслуживать десятки и сотни тысяч устройств;
- большая дальность передачи радиосигнала по сравнению с другими беспроводными технологиями, используемыми для телеметрии, достигает 10-15 км;
- низкое энергопотребление у конечных устройств, благодаря минимальным затратам энергии на передачу небольшого пакета данных;
- высокая проникающая способность радиосигнала при использовании частот субгигагерцового диапазона.

- Модуль 3-х осевого гироскопа и акселерометра MPU-6050 на базе плат Arduino.

В основе данного модуля лежит микросхема MPU6050, в которой размещаются два датчика: акселерометр на три координаты и трехосевой гироскоп. Гироскоп измеряет угловые скорости по трем осям с разными пределами измерений: 250, 500, 1000, и 2000 градусов в секунду.

Акселерометр, называемый иначе датчиком вибрации или вибродатчиком, измеряет ударное и вибрационное ускорение в системах диагностики. Акселерометр микросхемы MPU-6050 использует пьезоэлектрический эффект, обладает широкой частотной характеристикой, хорошей чувствительностью и высокой разрешающей способностью.

Схематично, параметры, измеряемые акселерометрами, можно сгруппировать в следующие классы:

- измерение вибрации: объект вибрирует, если он производит колебательные движения относительно положения равновесия;
- измерение ударных ускорений: внезапное возбуждение структуры, создающее резонанс;
- измерение движения: медленное перемещение со скоростью от доли секунды до нескольких минут;
- сейсмоисследования: измерения малых перемещений и низкочастотной вибрации. Такие измерения требуют специализированных малошумящих акселерометров с высокой разрешающей способностью.

Совместное использование акселерометра и гироскопа позволяет определить движение тела в трехмерном пространстве. Их данные проходят предварительную обработку и передаются по последовательному интерфейсу I2C в микроконтроллер.

Для наиболее точного трекинга быстрых и медленных движений возможно запрограммировать необходимые режимы: ± 250 , ± 500 , ± 1000 , $\pm 2000^\circ/\text{сек}$ для гироскопа, а также $\pm 2g$, $\pm 4g$, $\pm 8g$, $\pm 16g$ для акселерометра. Необходимое напряжение питания для модуля от 3,5 до 6 В.

- **Модуль зарядного устройства для литиевых АКБ на TP4056** с дополнительной защитой от токовой перегрузки и переразряда.

- **Литий-полимерный аккумулятор, 3500 мА*ч.**

Принцип работы устройства заключается в том, что модуль 3-х осевого гироскопа и акселерометра MPU-6050 регистрирует вибрации и температуру трубопровода в точке, где установлено устройство. Акселерометр преобразует механические колебания в электрический сигнал, пропорциональный ускорению. Сигнал поступает в микроконтроллер, по его команде происходит аналого-цифровое преобразование сигнала, интегрирование измеряемой величины, цифровая фильтрация и выдача измеренного значения в нормированном виде.

Далее необходимо передать полученную информацию по цепочке другим установленным на трубопроводе устройствам до головного датчика по сети LoRaWAN, с головного – на любое устройство, которое можно подключить по Wi-Fi.

Таким образом, устройство можно разделить на передатчик и приемник: передатчик устанавливается непосредственно на исследуемом объекте для контроля его параметров, приемник находится на диспетчерском пункте и отвечает за передачу информации по Wi-Fi операторам (рисунок 2, 3).

В итоге данные об изменении параметров поступают на сайт (Гл. 3.2).

Питание осуществляется при помощи модуля зарядного устройства (ЗУ), для этого к модулю ЗУ TP4056 подключен литий-полимерный аккумулятор.

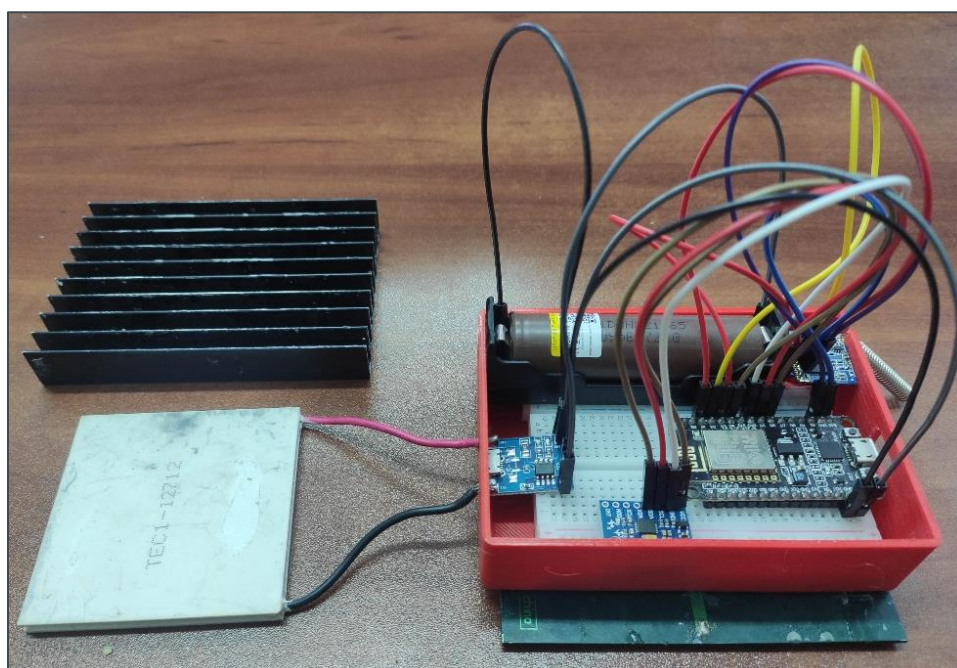


Рисунок 2 – Передатчик прототипа устройства, устанавливаемый на объекте для контроля его параметров.

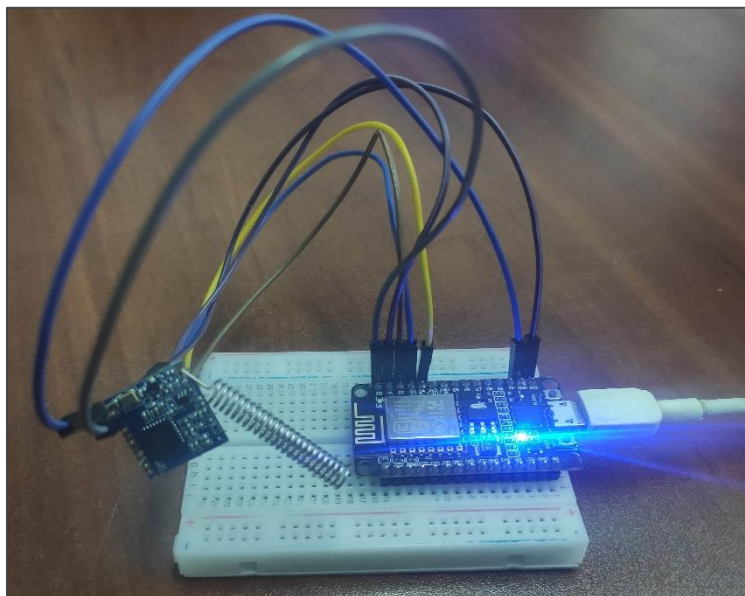


Рисунок 3 – Приемник прототипа устройства на диспетчерском пункте

Среди основных задач, решаемых с помощью рассматриваемого устройства, можно выделить следующие:

- выявление деформированных участков трубопровода (коробление, провисание и другие отклонения от первоначальной формы) в реальном времени;
- уточнение зон высокой сейсмичности;
- ранжирование участков по степени опасности, определение участков для первоочередного диагностического исследования;
- проведение диагностических обследований;
- выработка управленческих решений для предотвращения деформаций трубопровода.

Разработанное устройство относится к беспроводным системам мониторинга, позволяющим обеспечить:

- Сбор и обработку информации об изменении контролируемых параметров объектов;
- Беспроводную передачу результатов измерений от датчиков до управляющего оборудования системы на значительные расстояния, с последующей пересылкой информации через радиосети и Интернет на любое необходимое устройство;
- Дистанционный мониторинг технического состояния трубопроводных систем и оборудования;
- Существенную экономию трудовых и финансовых ресурсов при создании системы технического мониторинга объектов за счет отсутствия необходимости прокладки проводов между сенсорными датчиками и управляющим оборудованием системы;

- Возможность установки устройства с минимальным нарушением графика работ.

Преимущества:

- Низкая совокупная стоимость устройства и его монтажа, являющаяся одним из основных требований к конструкции устройства;
- Небольшие габаритно-массовые характеристики;
- Механическая надежность: исключены аварии и сбои в передаче данных о полученных измерениях, вызванные порывом проводов;
- Автоматизация измерений исключает влияние человеческого фактора;
- Отсутствие расходов на прокладку проводов и обслуживающий персонал;
- Небольшое энергопотребление;
- Автономная работа системы и ее элементов;
- Экологичность - отсутствие загрязнения при установке и эксплуатации системы;
- Возможность установки в труднодоступных местах.

Таким образом, техническим результатом, получаемым от внедрения разработанного устройства, является возможность дистанционного мониторинга состояния трубопроводных систем и оборудования.

2.3. СОЗДАНИЕ САЙТА ПРОЕКТА

Существует множество облачных решений, как платных, так и бесплатных, которые можно использовать для публикации данных с датчиков, например Blynk или Arduino IoT Cloud, но создав собственную панель управления мы имеем полный контроль над ней, можем настроить её под свои нужды.

Схема доставки данных выглядит следующим образом:

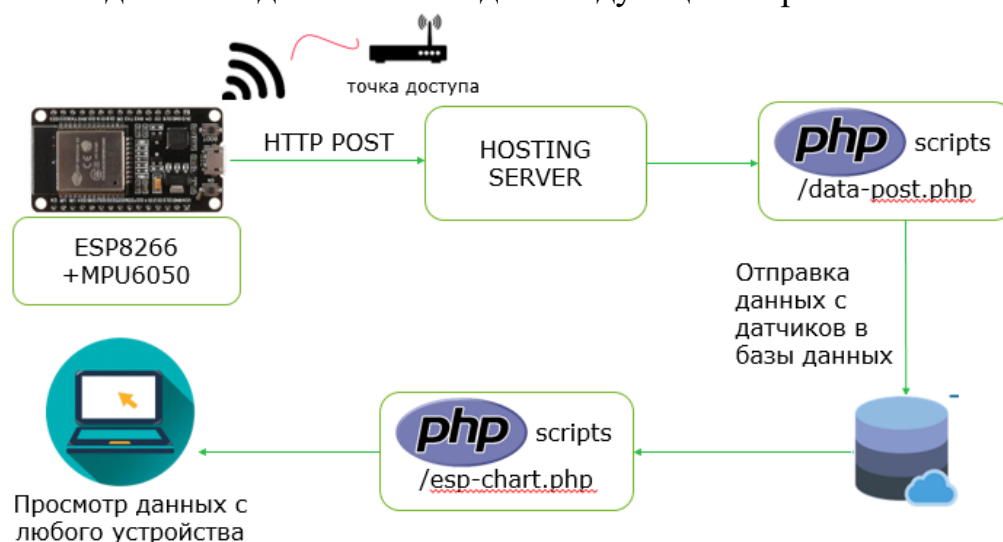


Рисунок 3- Схема отправки данных на сервер

Изначально был создан сервер с помощью программы Open Server, далее с помощью phpmyAdmin настраиваем базу данных.

Для сохранения данных была создана MySQL таблица, чтобы в дальнейшем можно было работать с массивом данных, а также в дальнейшем скачивать данные работы устройства

```
CREATE TABLE `mpu6050` (
  `id` int(11) NOT NULL,
  `vibration` float NOT NULL,
  `temperature` float NOT NULL,
  `charge` float NOT NULL,
  `created_date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

На сервер передаются три переменные vibration – данные о положении датчика MPU6050 (Гц), temperature – температура окружающей среды узла (), charge – заряд аккумулятора (В).

Для сохранения данных и помещения их в таблицу был написан php скрипт, который подключается к базе данных и отправляет данные на сервер [Приложение 3]







				id	vibration	temperature	charge	date
<input type="checkbox"/>	 Изменить	 Копировать	 Удалить	2	1.1	25	3.7	2023-01-17 01:24:21
<input type="checkbox"/>	 Изменить	 Копировать	 Удалить	3	1.2	25	3.7	2023-01-17 01:24:34
<input type="checkbox"/>	 Изменить	 Копировать	 Удалить	4	1.1	25.1	3.7	2023-01-17 01:24:43
<input type="checkbox"/>	 Изменить	 Копировать	 Удалить	5	1	25.1	3.7	2023-01-17 01:24:54
<input type="checkbox"/>	 Изменить	 Копировать	 Удалить	6	7	25.1	3.7	2023-01-17 01:25:01
<input type="checkbox"/>	 Изменить	 Копировать	 Удалить	7	5.8	25	3.7	2023-01-17 01:25:30
<input type="checkbox"/>	 Изменить	 Копировать	 Удалить	8	1.3	25	3.7	2023-01-17 01:25:38

Рисунок 5-Внешний вид базы данных MySQL

Далее с помощью готового js скрипта и написанного php скрипта [Приложение 4], происходит визуализация данных.

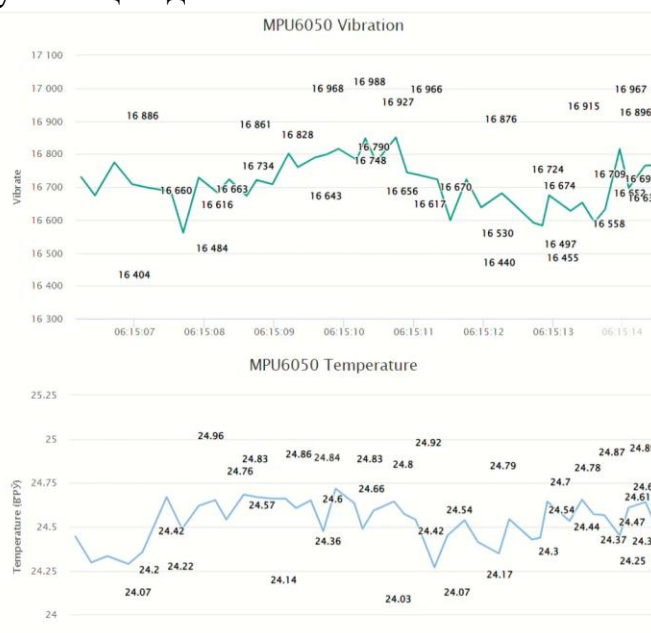


Рисунок 5-Внешний вид страницы сервера

2.3 АВТОНОМНОСТЬ УСТРОЙСТВА

Основная задача, сделать устройство максимально автономным. Сделать это можно двумя способами: программно или физически.

С помощью погружения в сон платы esp8266, можно добиться уменьшения потребляемого тока, и как следствие продлить работу аккумуляторов и увеличить автономность устройства.

При необходимости устройству можно обеспечить бесперебойное питание, которое осуществляется при помощи модуля зарядного устройства (ЗУ), для этого к модулю ЗУ TP4056 подключается литий-полимерный аккумулятор, солнечная панель и элемент Пельтье, благодаря которым происходит питание ЗУ (рисунок 6).

Модуль зарядного устройства также заряжает аккумулятор, и при отсутствии источника питания питание для контроллера поступает от аккумулятора.

Модуль Пельтье ТЕС1-12712 представляет собой полупроводниковое твердотельное устройство, принцип действия которого основан на эффекте Пельтье (охлаждение - нагрев элемента при прохождении через него электрического тока). В разрабатываемом устройстве используется обратный эффект, заключающийся в возникновении электрического тока в замкнутой цепи из двух проводников, контакты которых имеют разную температуру.

Для подключения к элементу Пельтье используется модуль, повышающий постоянное напряжение до 5 Вольт, – преобразователь DC/DC повышающий с 2,5 В на 5 В. Сам элемент Пельтье помещается в теплоизоляции (вспененный каучук) и устанавливается на внешнюю поверхность трубопровода при помощи термопроводящей клейкой пасты, к нему с помощью той же пасты крепится радиатор, над которым расположена солнечная панель. Таким образом, обеспечивается наибольший перепад температур для выработки электрического тока элементом Пельтье.

При помощи такой системы питания обеспечивается автономность разработанного устройства.

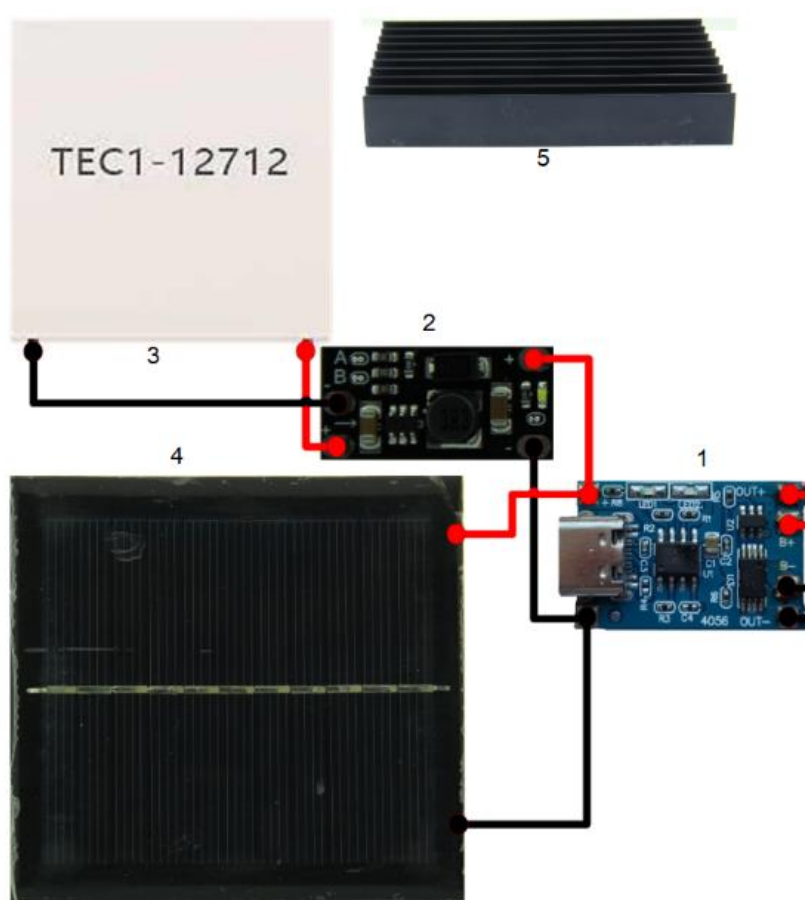


Рисунок 6 – Модули устройства для обеспечения его автономности (1 - Модуль зарядного устройства для литиевых АКБ на TP4056; 2 - Преобразователь DC/DC повышающий с 2,5 В на 5 В; 3 - Модуль Пельтье TEC1-12712; 4 - Солнечная панель размером 95х95 мм, напряжение 5,5 В, мощность 1 Вт; 5 - Радиатор, 100х70 мм)

3. ОПЫТНО-ПРОМЫШЛЕННЫЕ ИСПЫТАНИЯ УСТРОЙСТВА

3.1 КОНТРОЛЬ ПРОХОЖДЕНИЯ ОЧИСТНОГО УСТРОЙСТВА ПО ПАРАМЕТРАМ ВИБРАЦИИ

При движении очистное устройство воздействует на стенки и внутренние сварные швы трубопровода, возбуждая волновой процесс и ударные импульсы, которые распространяются по стенке трубы на значительные расстояния. Их форма и амплитуда зависят от типа внутритрубного объекта (ВТО), скорости его движения, размеров внутренних выступов сварных швов и демпфирующих (физико-механических) свойств внешней среды трубопровода.

Вибрации трубопровода во время и при отсутствии движения ВТО принципиально отличаются по амплитудно-частотному спектру возбуждаемого в стенке трубопровода волнового процесса [6].

В рамках проектной работы 2021 года на Сузунском производственном участке (СПУ) на межпромысловом трубопроводе «Сузун-Ванкор» проводились опытно-промышленные испытания (ОПИ) первого прототипа разрабатываемого устройства, регистрировалось прохождение скребка.

Для испытаний был выбран участок трубопровода, находящийся в 200 метрах от камеры пуска очистного устройства (ОУ). На участке трубы проведен демонтаж изоляции в месте сварного шва и установлен прототип устройства, произведена фиксация макетной платы с контроллером, которая подключена к аккумулятору (рисунок 7).



Рисунок 7 – Замер вибрации первым прототипом устройства во время прохождения скребка

Далее была проведена запись вибрации после пуска скребка. После прохождения скребка точки, в которой установлен прототип устройства, он был

демонтирован, произвели монтаж теплоизоляции. Для проверки результата также установили прототип в 10 км от УПН, сняли показания колебаний с датчика. В процессе испытаний получили данные о прохождении очистного устройства точки, в которой установлен прототип.

По показаниям колебаний в 200 м от УПН построена акселерограмма вибрации (график зависимости ускорения колебаний от времени) (рисунок 8). Представленная акселерограмма состоит из низкоуровневого шума, являющегося следствием эксплуатации трубопровода, и нескольких довольно сильных ударных воздействий, которые появились в результате прохождения очистного устройства точки, где установлен датчик, и сварных швов до и после этой точки. Наибольшая величина получена в момент времени $t = 486$ с (примерно 8 мин), и составляет 22,6.

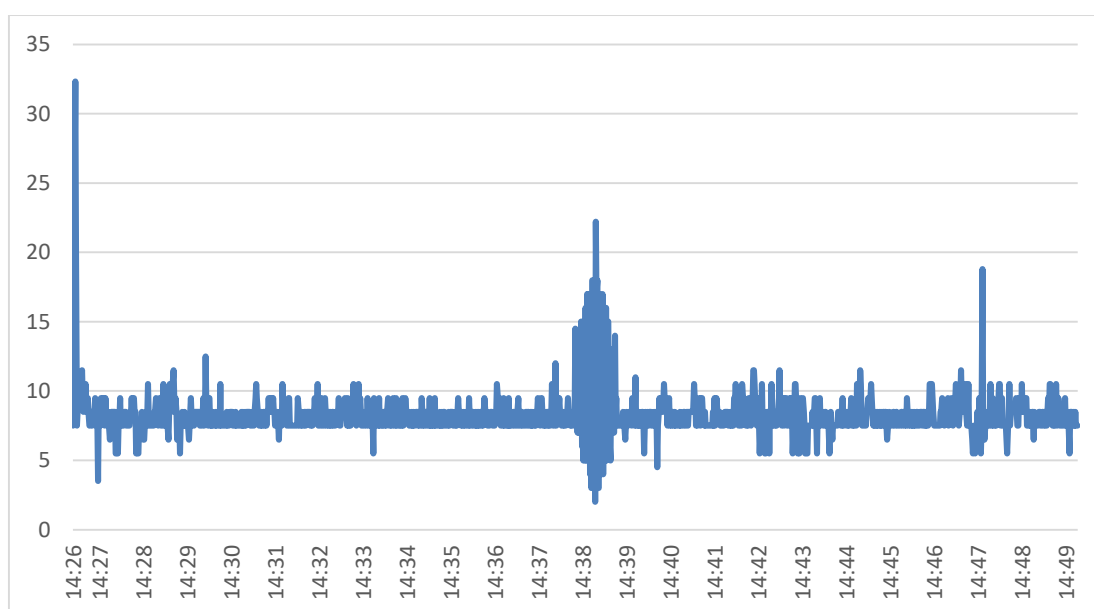


Рисунок 8 – Акселерограмма вибрации при прохождении очистного устройства

Таким образом, по пиковым значениям ускорения регистрируется прохождение внутритрубного снаряда точки установки датчика и сварных швов. Также, зная время установки и включения устройства, определено, что через 6 часов 45 минут после запуска очистного устройство оно прошло точку 10 км.

Далее в рамках ОПИ проведены замеры напряжения и силы тока элемента Пельтье и солнечной батареи, установленных на трубопроводе, для определения фактической вырабатываемой мощности системы с целью обеспечения ее автономности (рисунок 9). Полученные данные подтвердили теоретический расчет и возможность автономного использования устройства.



Рисунок 9 – Замер напряжения и силы тока элемента Пельтье и солнечной батареи

Таким образом, устройство позволяет установить факт и момент прохождения ВТО через контрольные точки, что значительно сокращает район его поиска при незапланированной остановке.

3.2 СОЗДАНИЕ БАЗЫ КОЛЕБАНИЙ ТРУБОПРОВОДА В ПРОЦЕССЕ ЭКСПЛУАТАЦИИ

Для создания базы данных колебаний трубопровода при тех или иных процессах первый прототип устройства был установлен на трубопроводе Сузунского месторождения. С него получены данные о вибрации и проведен их анализ.

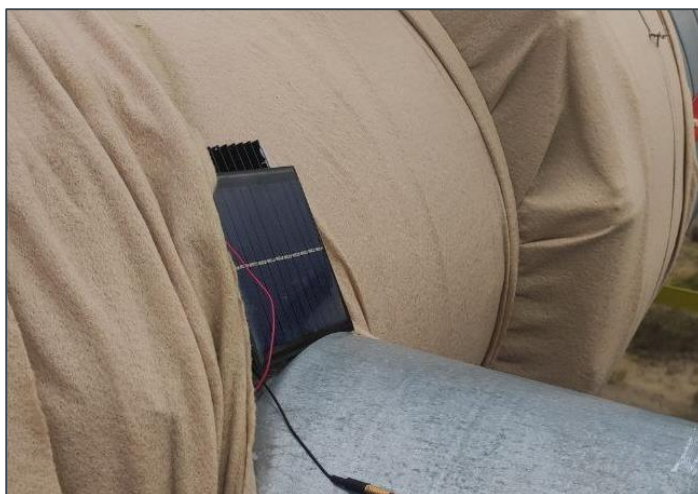


Рисунок 10 – Установленное на трубопроводе устройство

По данным с устройства построена акселерограмма вибрации (рисунок 11). На акселерограмме помимо низкоуровневого шума, являющегося следствием эксплуатации трубопровода, выделяется период времени с высоким уровнем колебаний (выделено черным квадратом на рисунке 11). Данное увеличение амплитуды предположительно вызвано прохождением газовой пробки, что подтверждается наблюдениями.

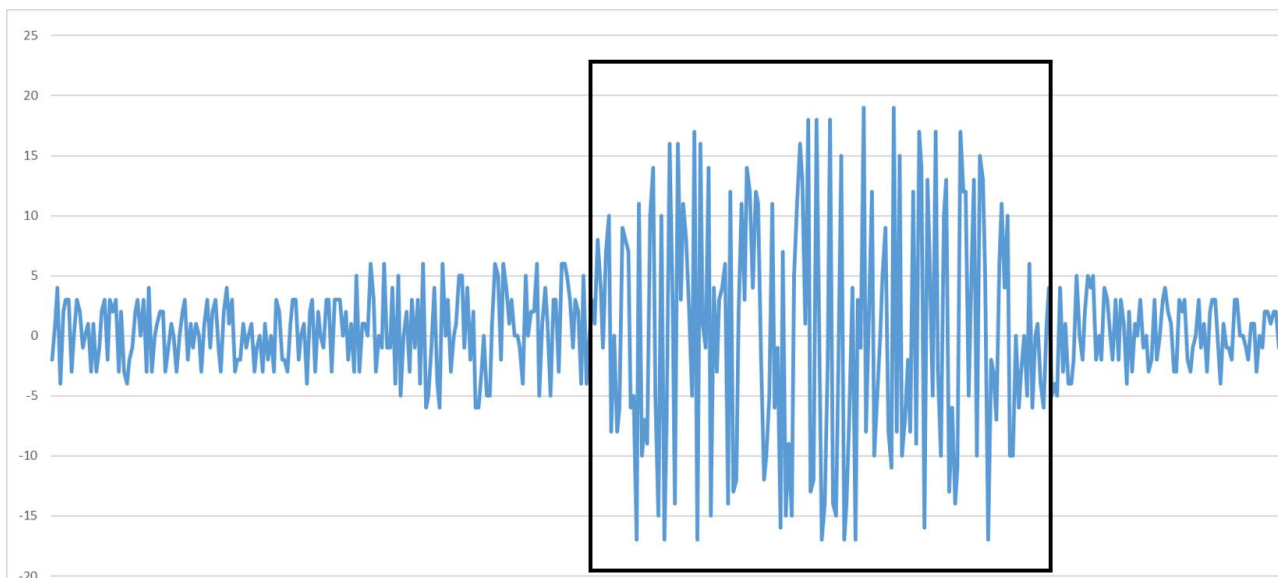


Рисунок 11 – Акселерограмма вибрации при эксплуатации трубопровода

Таким образом, на данный момент формируется база данных колебаний для проведения исследования работы трубопровода, что позволит определять протекание тех или иных процессов.

2.4.3 Контроль температуры трубопровода

Зимой 2022 года прототип устройства был установлен на участке трубопровода КП11.4 Сузунского месторождения для контроля температуры одной из труб, в которой происходит уменьшение скорости течения транспортируемого продукта, что приводит к его замерзанию.

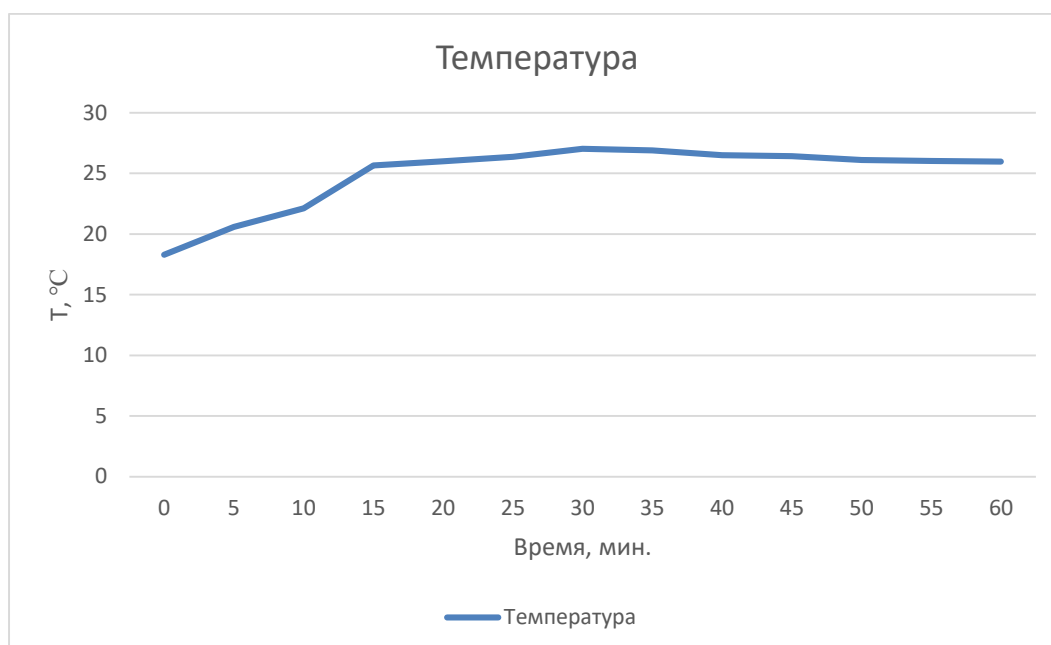


Рисунок 13 – Динамика изменения температуры трубопровода.

2.4.4 Контроль планово-высотного положения трубопровода

Трубопроводы являются протяженными объектами, трассы которых прокладываются в разнообразных геологических, гидрогеологических и климатических условиях, в зонах влияния которых происходит заметная активизация процессов, связанных с изменением свойств грунтов, деградацией многолетнемерзлых пород, изменением границ сезонно-талого слоя, режима поверхностных и грунтовых вод, потерей устойчивости грунта на склонах, что, с одной стороны, приводит к активизации опасных геологических процессов, часто не свойственных данной территории, а с другой – обуславливает дополнительные нагрузки на трубопровод.

Значительная часть трубопроводов сооружается в надземном исполнении – на опорах или эстакадах, обеспечивающих некоторый зазор между трубой и грунтом. В таких трубопроводах создаются благоприятные условия для формирования стоячих волн в пролетах трубы между соседними креплениями, которые характеризуются набором собственных частот, зависящих от длины пролетов, материала и диаметра трубы, толщины ее стенок и др. При определенных условиях на этих частотах могут возникать резонансы, повышающие вероятность повреждения участка трубопровода.

К возникновению резонансов могут приводить как природные (ветровая нагрузка, осадки, сейсмические воздействия и т. д.), так и техногенные (движение транспорта, различные технологические процессы, воздействие на трубу потока транспортируемого вещества и др.) факторы. Особенно опасны низкочастотные резонансы, вызываемые мощными периодическими или квазипериодическими воздействиями – динамическими нагрузками

трубопроводов от вращающихся неуравновешенных роторов насосов и турбоагрегатов, мощными вибрациями различных механизмов, например, в процессе интенсификации добычи нефти и т. д. Частичная или полная потеря устойчивости опор трубопровода, например вследствие выпучивания или просадки грунта под ними, приводят, наряду с возрастанием изгибных напряжений в трубах, к фактическому увеличению длины пролетов и, как следствие, к понижению их собственных частот. Это еще больше увеличивает опасность возникновения низкочастотных резонансов, которые могут привести к повреждению или даже к разрушению участка трубы [5].

В данном проекте рассмотрена возможность оценки собственных частот и форм колебаний участков трубопроводов посредством разработанного устройства. Предлагаемая система диагностики включает в себя непрерывный мониторинг с получением данных от автоматизированных постов наблюдения (датчиков: акселерометра и гироскопа) о положении и вибрации трубопровода при влиянии различных процессов с целью обеспечения контроля и прогнозирования планово-высотного положения.

Зимой 2022 года прототип устройства был установлен на участке трубопровода Сузунского месторождения для контроля его пространственного положения. Рассматриваемый участок подвержен провисам.



Рисунок 14 – Прототип устройства, установленный на трубопроводе.

4. ЗАКЛЮЧЕНИЕ

В проекте предложена технология, показана практика применения мобильного устройства для контроля параметров трубопровода на участках прокладки в сложных геологических условиях, обеспечивающей непрерывный контроль как отдельных элементов, так и всего объекта в целом на основе анализа данных, полученных от автоматизированных постов наблюдения и в результате периодических обследований.

Разработанное устройство относится к беспроводным системам мониторинга, позволяющим обеспечить дистанционный мониторинг технического состояния объектов, а также существенную экономию трудовых и финансовых ресурсов.

Проведены опытно-промышленные испытания устройства на Сузунском производственном участке, в рамках которых регистрировалось прохождение ОУ, планово-высотное положение участка трубопровода и его температура.

По полученным данным определено время прохождения ОУ двух контрольных точек, построена акселерограмма вибрации. Также в рамках ОПИ проведены замеры напряжения и силы тока элементов питания системы для подтверждения ее автономности. Проведена регистрация температуры участка трубопровода, отслеживалась ее динамика. На данный момент устройство установлено на трубопроводе для контроля его положения в пространстве.

В результате доказаны эффективность и надежность применения устройства в реальных условиях.

По результатам оценки экономической эффективности реализация данного проекта на Сузунском производственном участке оказывает положительный экономический эффект. Срок окупаемости проекта составляет менее одного года.

5. ССЫЛКИ

1. Акуленко Л. Д., Коровина Л. И., Нестеров С. В. Собственные колебания участка трубопровода // Изв. РАН. Механика твердого тела. – 2011. – № 1.
2. Алексеев В.А., Донченко В.А., Шапарев В.Я., Сынков С.А. Сигнализаторы прохождения внутритрубных объектов // Диагностика трубопроводов: Тез. докл. III Междунар. конф. – М., 2001. – С. 161–166.
3. Березнев А. В. Влияние внутреннего гидростатического давления на частоты свободных колебаний криволинейного участка трубопровода // Изв. вузов. Нефть и газ. – 2011. – № 3.
4. ГОСТ 32388-2013 «Трубопроводы технологические. Нормы и методы расчета на прочность, вибрацию и технологические воздействия».
5. Курочкин А. А. Вибродиагностическое сопровождение внутритрубных объектов / А. А. Курочкин, Н. Н. Бочкарев // Известия Томского политехнического университета [Известия ТПУ]. — 2010. — Т. 317, № 2: Математика и механика. Физика. — [С. 73-77].
6. Прокофьев А. Б. Расчет собственных частот и форм колебаний трубопроводов с помощью программного комплекса // Изв. Самарского науч. центра РАН. – 1999. – № 2.
7. РТМ 38.001-94 «Указания по расчету на прочность и вибрацию технологических стальных трубопроводов».
8. СА 03-003-07 «Расчеты на прочность и вибрацию стальных технологических трубопроводов».
9. Супрунчик В.В., Коновалов Н.М., Мызников М.О. Система сопровождения внутритрубных снарядов «ССВС001» // Трубопроводный транспорт. – 2003. – № 12. – С. 9–12.
10. Bu N., Ueno N., Koyanagi S., Ichiki M., Fukuda O., Akiyama M. Experimental studies on vibration testing of pipe joints using metal gaskets, Proceedings of the 6th WSEAS International Conference on instrumentation, Measurement, Circuits & Systems, Hangzhou, China, April 15-17, 2007.
11. Li B. H., Gao H. S., Zhai H. B., Liu Y. S., Yue Z. F. Free vibration analysis of multispan pipe conveying fluid with dynamic stiffness method, Nuclear Engineering and Design, 2011, Vol. 241.
12. Mao Q., Zhang J. H., Luo Y. S., Wang H. J., Duan Q. Experimental studies of orifice-induced wall pressure fluctuations and pipe vibration, International Journal of Pressure Vessels and Piping, 2006, Vol. 83.
13. Xie J. H., Tian K., He L., Yang T. R., Zhu X. H. Modal experiment research on fluid-solid coupling vibration of hydraulic long-straight pipeline of shield machine, Applied Mechanics and Materials, 2012, Vol. 105-107.

6. ПРИЛОЖЕНИЕ

Приложение 1. Код работы узла ячеистой сети.

```
#include "LoRaWan_APP.h"
#include "Arduino.h"
#include "Wire.h"

//----- MPU6050 Measurement & Filtering Range -----
-----

#define AFS_SEL 2 // Accelerometer Configuration Settings AFS_SEL=2, Full Scale
Range = +/- 8 [g]
#define DLPF_SEL 0 // DLPF Configuration Settings Accel BW 260Hz, Delay 0ms /
Gyro BW 256Hz, Delay 0.98ms, Fs 8KHz
#define IMUAddress 0x68 // адрес датчика

//----- Variables for gravity -----
-----
int accX, accY, accZ; // Accelerometer values
long Cal_accX, Cal_accY, Cal_accZ; // Calibration values
float GaccX, GaccY, GaccZ; // Convert accelerometer to gravity value
float Min_GaccX=0, Max_GaccX=0, PtoP_GaccX, Min_GaccY=0, Max_GaccY=0, PtoP_GaccY,
Min_GaccZ=0, Max_GaccZ=0, PtoP_GaccZ; // Finding Min, Max & Peak to Peak of gravity
value
float Min = 0, Max = 0; // Initial value of Min, Max
int cnt; // Count of calibration process
float Grvt_unit; // Gravity value unit
long period, prev_time; // Period of calculation
int tempRaw;
int dataToSend = 0;
// Lora and LoraWan Settings
#ifndef LoraWan_RGB
#define LoraWan_RGB 0
#endif

#define RF_FREQUENCY 868000000 // Hz
#define TX_OUTPUT_POWER 14 // dBm
#define LORA_BANDWIDTH 0 // [0: 125 kHz,
// 1: 250 kHz,
// 2: 500 kHz,
// 3: Reserved]
#define LORA_SPREADING_FACTOR 7 // [SF7..SF12]
#define LORA_CODINGRATE 1 // [1: 4/5,
// 2: 4/6,
// 3: 4/7,
// 4: 4/8]
#define LORA_PREAMBLE_LENGTH 8 // Same for Tx and Rx
#define LORA_SYMBOL_TIMEOUT 0 // Symbols
#define LORA_FIX_LENGTH_PAYLOAD_ON false
#define LORA_IQ_INVERSION_ON false
```

```

#define RX_TIMEOUT_VALUE          1000
#define BUFFER_SIZE                40 // Define the payload size
here

char txpacket[BUFFER_SIZE];
char rxpacket[BUFFER_SIZE];

static RadioEvents_t RadioEvents;
int16_t gyroX;
int16_t gyroY;
int16_t gyroZ;

uint16_t boardID = 1, txNumber;

int fracPart(double val, int n)
{
    if( val >= 0 )
        return (int)((val - (int)(val))*pow(10,n));
    else
        return (int)((((int)(val) - val)*pow(10,n));
}

void setup() {

    Serial.begin(115200);

    init_MPU6050();

    Gravity_Range_Option();

    Calib_MPU6050(); // Calculating calibration value

    txNumber = 0;

    Radio.Init( &RadioEvents );
    Radio.SetChannel( RF_FREQUENCY );
    Radio.SetTxConfig( MODEM_LORA, TX_OUTPUT_POWER, 0, LORA_BANDWIDTH,
                        LORA_SPREADING_FACTOR, LORA_CODINGRATE,
                        LORA_PREAMBLE_LENGTH, LORA_FIX_LENGTH_PAYLOAD_ON,
                        true, 0, 0, LORA_IQ_INVERSION_ON, 3000 );
}

void init_MPU6050(){
    //MPU6050 Initializing & Reset
    Wire.beginTransaction(IMUAddress);
    Wire.write(0x6B); // PWR_MGMT_1 register
    Wire.write(0);    // set to zero (wakes up the MPU-6050)
}

```

```

Wire.endTransmission(true);

//MPU6050 Clock Type
Wire.beginTransmission(IMUAddress);
Wire.write(0x6B); // PWR_MGMT_1 register
Wire.write(0x03); // Selection Clock 'PLL with Z axis gyroscope reference'
Wire.endTransmission(true);

//MPU6050 accelerometer Configuration Setting
Wire.beginTransmission(IMUAddress);
Wire.write(0x1C); // accelerometer Configuration register
if(AFS_SEL == 0) Wire.write(0x00); // AFS_SEL=0, Full Scale Range = +/- 2 [g]
else if(AFS_SEL == 1) Wire.write(0x08); // AFS_SEL=1, Full Scale Range = +/-
4 [g]
else if(AFS_SEL == 2) Wire.write(0x10); // AFS_SEL=2, Full Scale Range = +/-
8 [g]
else Wire.write(0x18); // AFS_SEL=3, Full Scale Range = +/- 10 [g]
Wire.endTransmission(true);

//MPU6050 DLPF(Digital Low Pass Filter)
Wire.beginTransmission(IMUAddress);
Wire.write(0x1A); // DLPF_CFG register
if(DLPF_SEL == 0) Wire.write(0x00); // acccel BW 260Hz, Delay 0ms / Gyro BW
256Hz, Delay 0.98ms, Fs 8KHz
else if(DLPF_SEL == 1) Wire.write(0x01); // acccel BW 184Hz, Delay 2ms /
Gyro BW 188Hz, Delay 1.9ms, Fs 1KHz
else if(DLPF_SEL == 2) Wire.write(0x02); // acccel BW 94Hz, Delay 3ms / Gyro
BW 98Hz, Delay 2.8ms, Fs 1KHz
else if(DLPF_SEL == 3) Wire.write(0x03); // acccel BW 44Hz, Delay 4.9ms /
Gyro BW 42Hz, Delay 4.8ms, Fs 1KHz
else if(DLPF_SEL == 4) Wire.write(0x04); // acccel BW 21Hz, Delay 8.5ms /
Gyro BW 20Hz, Delay 8.3ms, Fs 1KHz
else if(DLPF_SEL == 5) Wire.write(0x05); // acccel BW 10Hz, Delay 13.8ms /
Gyro BW 10Hz, Delay 13.4ms, Fs 1KHz
else Wire.write(0x06); // acccel BW 5Hz, Delay 19ms / Gyro BW 5Hz, Delay
18.6ms, Fs 1KHz
Wire.endTransmission(true);
}

void Gravity_Range_Option(){
  switch(AFS_SEL) { // Selecting Gravity unit value
    case 0:
      Grvt_unit = 16384;
      break;
    case 1:
      Grvt_unit = 8192;
      break;
    case 2:
      Grvt_unit = 4096;

```

```
        break;
    case 3:
        Grvt_unit = 3276.8;
        break;
    }
}
```

```
void getValues() {
    /* Update all the values */
    Wire.beginTransmission(IMUAddress);
    Wire.write(0x3B); // starting with register 0x3B (ACCEL_XOUT_H)
    Wire.endTransmission(false);
    Wire.requestFrom(IMUAddress, 14, true); // request a total of 14 registers
    accX = Wire.read() << 8 | Wire.read(); // 0x3B (ACCEL_XOUT_H) & 0x3C
    (ACCEL_XOUT_L)
    accY = Wire.read() << 8 | Wire.read(); // 0x3D (ACCEL_YOUT_H) & 0x3E
    (ACCEL_YOUT_L)
    accZ = Wire.read() << 8 | Wire.read(); // 0x3F (ACCEL_ZOUT_H) & 0x40
    (ACCEL_ZOUT_L)
    tempRaw = Wire.read() << 8 | Wire.read(); // 0x41 (TEMP_OUT_H) & 0x42
    (TEMP_OUT_L)
    tempRaw = ((float)tempRaw + 12412.0) / 340.0;
    gyroX = Wire.read() << 8 | Wire.read(); // 0x43 (GYRO_XOUT_H) & 0x44
    (GYRO_XOUT_L)
    gyroY = Wire.read() << 8 | Wire.read(); // 0x45 (GYRO_YOUT_H) & 0x46
    (GYRO_YOUT_L)
    gyroZ = Wire.read() << 8 | Wire.read(); // 0x47 (GYRO_ZOUT_H) & 0x48
    (GYRO_ZOUT_L)
    //temp = ((float)tempRaw + 12412.0) / 340.0;
}
```

```
void loop()
{

    getValues();

    Calc_Grvt();

    Display_Grvt();

    char str[160];
    uint32_t starttime = millis();
    getValues();
    //int temp = 0;

    //delay(500);
    dataToSend = (accX + accY + accZ)/3;
    //if( ++txNumber >= 10000) txNumber = 0;
```

```
sprintf(txpacket, "%d#%d", dataToSend, tempRaw);
Serial.printf("sending \"%s\\", length %d\\r\\n", txpacket, strlen(txpacket));

Radio.Send( (uint8_t *)txpacket, strlen(txpacket) ); //send the package out

//turnOnRGB(bOn ? COLOR_SEND : COLOR_RECEIVED, 0); //change rgb color
//bOn = !bOn;

delay(10);
//dataToSend = 0;
}

void Calib_MPU6050() {
    for(int i = 0 ; i < 2000 ; i++) { // Summing Iteration for finding calibration
value
        if(i % 200 == 0) { // Display progress every 200 cycles
            cnt++;
            if(cnt == 1) { // Characcters to display first
                Serial.print("Calculating .");
                */
            }
            else { // Display progress by point
                Serial.print(".");
            }
        }
    }

    getValues(); // Read acccelerometer data

    delay(10);

    // Sum data
    Cal_accX += accX;
    Cal_accY += accY;
    Cal_accZ += accZ;
}

// Average Data
Cal_accX /= 2000;
Cal_accY /= 2000;
Cal_accZ /= 2000;

// Serial Print
Serial.println("");
Serial.println("End of Calculation");
Serial.print("Cal_accX = "); Serial.print(Cal_accX);
Serial.print(" | Cal_accY = "); Serial.print(Cal_accY);
Serial.print(" | Cal_accZ = "); Serial.println(Cal_accZ);
}
```

```
void Calc_Grvt() {
    accX = (accX - Cal_accX); // Calibrated acccelerometer value
    accY = (accY - Cal_accY); // Calibrated acccelerometer value
    accZ = (accZ - Cal_accZ); // Calibrated acccelerometer value

    GaccX = accX / Grvt_unit; // Converting the Calibrated value to Gravity value
    GaccY = accY / Grvt_unit; // Converting the Calibrated value to Gravity value
    GaccZ = accZ / Grvt_unit; // Converting the Calibrated value to Gravity value

    //----- Calculating Min, Max & Peak to Peak of Gravity -----
    -----

    Min_GaccX = min(Min_GaccX, GaccX);
    Max_GaccX = max(Max_GaccX, GaccX);
    PtoP_GaccX = Max_GaccX - Min_GaccX;

    Min_GaccY = min(Min_GaccY, GaccY);
    Max_GaccY = max(Max_GaccY, GaccY);
    PtoP_GaccY = Max_GaccY - Min_GaccY;

    Min_GaccZ = min(Min_GaccZ, GaccZ);
    Max_GaccZ = max(Max_GaccZ, GaccZ);
    PtoP_GaccZ = Max_GaccZ - Min_GaccZ;
}

void Display_Grvt() {
    //----- Serial print -----
    -----

    Serial.print("accX= " + String(accX));
    Serial.print(" |accY= " + String(accY));
    Serial.println(" |accZ= " + String(accZ));

    //----- LCD Display -----
    -----

    period = millis() - prev_time;

    if(period > 1000) {

        prev_time = millis();
        Min_GaccX = 0;
        Max_GaccX = 0;
        Min_GaccY = 0;
        Max_GaccY = 0;
        Min_GaccZ = 0;
        Max_GaccZ = 0;
    }
}
```

Приложение 2. Код работы конечного узла ячеистой сети.

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

// Replace with your network credentials
const char* ssid = "Redmi";
const char* password = "";

#define BAND          868E6;
#define BUFFER_SIZE  32 // Define the payload size here

bool bOn = false;
uint16_t rxLen = 0;
char dbgBuf[128];
char rxpacketVibrate[BUFFER_SIZE] = {0};
char rxpacketTemp[BUFFER_SIZE] = {0};
char rxpacket[BUFFER_SIZE] = {0};
int accX, accY, accZ, tempRaw;
bool flag = 0;
float data1;

// Create AsyncWebServer object on port 80
AsyncWebServer server(80);
String data(){
    int k= random(16384,17000);
    return String(k);
}
String RawTemperature(){
    int8_t temp = ((float)tempRaw + 12412.0) / 340.0;
    return String(temp);
}
String data2(){
    data1 = random(24, 25);          // генерируем целые числа
    float data3 = random(0, 100);    // генерируем числа после запятой
    float k = data1 + data3 / 100;   // складываем полученные значения
    //float k = random(24.5, 25);
    return String(k);
}
void setup(){

    // Serial port for debugging purposes
    Serial.begin(115200);
    Heltec.begin(true /*DisplayEnable Enable*/, true /*Heltec.LoRa Disable*/,
        true /*Serial Enable*/, true /*PABOOST Enable*/, BAND /*long BAND*/);
    Heltec.LoRa.setPreambleLength(8);
    Heltec.LoRa.setCodingRate4(5);
    Heltec.LoRa.setSpreadingFactor(7);
    Heltec.LoRa.setSignalBandwidth(125E3);
```

```

Heltec.LoRa.disableInvertIQ();

bool status;
LoRa.onReceive(onReceive);
// put the radio into receive mode
LoRa.receive();
// Initialize SPIFFS
if(!SPIFFS.begin()){
    Serial.println("An Error has occurred while mounting SPIFFS");
    return;
}

// Connect to Wi-Fi
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi..");
}

// Print ESP32 Local IP Address
Serial.println(WiFi.localIP());
int rxNum, id;
char showBuf[40];

if( rxLen > 0 )
{
    bOn = !bOn;
    digitalWrite(LED, bOn ? LOW : HIGH);
    Serial.println(dbgBuf);
    rxLen = 0;

    sscanf(rxpathet, "%d.%d.%d", &accX, &accY, &accZ);
    Serial.println(accX);
}

// Route for root / web page
server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){
    request->send(SPIFFS, "/index.html");
});
server.on("/vibration", HTTP_GET, [](AsyncWebServerRequest *request){
    request->send_P(200, "text/plain", String(rxpathetVibrate).c_str());
});
server.on("/temperature", HTTP_GET, [](AsyncWebServerRequest *request){
    request->send_P(200, "text/plain", String(rxpathetTemp).c_str());
});
server.on("/charge", HTTP_GET, [](AsyncWebServerRequest *request){
    request->send_P(200, "text/plain", String(rxpathetCh).c_str());
});
// Start server

```

```

    server.begin();
}

void loop(){

}

void onReceive(int packetSize)
{
    int i;

    // read packet
    for(i = 0; i < packetSize; i++){
        rxpacket[i] = (char)LoRa.read();
        if(flag){
            if(rxpacket[i] == '#'){
                flag = 1;
            }
            rxpacketVibrate[i] = rxpacket[i];
        }
        else{
            rxpacketTemp[i] = rxpacket[i];
        }
    }
    rxpacket[packetSize] = 0;
    rxLen = sprintf(dbgBuf, "Rcv: %s %s with RSSI %d", rxpacketVibrate,rxpacketTemp
, LoRa.packetRssi());
}

```

Приложение 3. Код отправки данных на сервер.

```

<?php
class mpu6050{
    public $link='';
    function __construct($vibration, $temperature, $charge){
        $this->connect();
        $this->storeInDB($vibration, $temperature, $charge);
    }

    function connect(){
        $this->link = mysqli_connect('localhost','root','') or die('Cannot connect to the
DB');
        mysqli_select_db($this->link,'vankor') or die('Cannot select the DB');
    }

    function storeInDB($vibration, $temperature, $charge){
        $query = "insert into mpu6050 set vibration='".$vibration."',
temperature='".$temperature."',charge='".$charge.'";

```

```

$result = mysqli_query($this->link,$query) or die('Errant query:  '.$query);
}

}
if($_GET['temperature'] != '' and $_GET['vibration'] != '' and $_GET['charge'] !=
''){
    $mpu6050=new mpu6050($_GET['vibration'],$_GET['temperature'],$_GET['charge']);
}

?>

```

Приложение 4. Код визуализации данных в виде графика.

```

<!DOCTYPE HTML><html>
<!-- Rui Santos - Complete project details at https://RandomNerdTutorials.com

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files.
The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software. -->
<head>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <script src="https://code.highcharts.com/highcharts.js"></script>
    <style>
        body {
            min-width: 310px;
            max-width: 800px;
            height: 400px;
            margin: 0 auto;
        }
        h2 {
            font-family: Arial;
            font-size: 2.5rem;
            text-align: center;
        }
    </style>
</head>
<body>
    <h2>ESP32 Lora Station</h2>
    <div id="chart-vibration" class="container"></div>
    <div id="chart-temperature" class="container"></div>
    <div id="chart-pressure" class="container"></div>
</body>
<script>
var chartT = new Highcharts.Chart({
    chart:{ renderTo : 'chart-vibration' },
    title: { text: 'MPU6050 Vibration' },

```

```

series: [{
  showInLegend: false,
  data: []
}],
plotOptions: {
  line: { animation: false,
    dataLabels: { enabled: true }
  },
  series: { color: '#059e8a' }
},
xAxis: { type: 'datetime',
  dateTimeLabelFormats: { second: '%H:%M:%S' }
},
yAxis: {
  title: { text: 'Vibrate' }
  //title: { text: 'Temperature (Fahrenheit)' }
},
credits: { enabled: false }
});
setInterval(function ( ) {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      var x = (new Date()).getTime(),
          y = parseFloat(this.responseText);
      //console.log(this.responseText);
      if(chartT.series[0].data.length > 40) {
        chartT.series[0].addPoint([x, y], true, true, true);
      } else {
        chartT.series[0].addPoint([x, y], true, false, true);
      }
    }
  };
  xhttp.open("GET", "/vibration", true);
  xhttp.send();
}, 400 ) ;

var chartH = new Highcharts.Chart({
  chart: { renderTo: 'chart-temperature' },
  title: { text: 'MPU6050 Temperature' },
  series: [{
    showInLegend: false,
    data: []
  }],
  plotOptions: {
    line: { animation: false,
      dataLabels: { enabled: true }
    }
  }
},

```



```
xAxis: {
  type: 'datetime',
  dateTimeLabelFormats: { second: '%H:%M:%S' }
},
yAxis: {
  title: { text: 'Temperature (Celseus)' }
},
credits: { enabled: false }
});
setInterval(function ( ) {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      var x = (new Date()).getTime(),
          y = parseFloat(this.responseText);
      x+=216000;
      //console.log(this.responseText);
      if(chartH.series[0].data.length > 40) {
        chartH.series[0].addPoint([x, y], true, true, true);
      } else {
        chartH.series[0].addPoint([x, y], true, false, true);
      }
    }
  };
  xhttp.open("GET", "/temperature", true);
  xhttp.send();
}, 400 ) ;

</script>
</html>
```