

Pay4Meter



Плати за **нужные** ресурсы



Целевой рынок и его ключевые вызовы

На середину 2017 года по различным оценкам в мобильных сетях функционировало 400- 500 миллионов устройств, осуществляющих передачу данных без участия человека и этот рынок растет

В промежутке между 2017 и 2023 годом совокупные темпы годового прироста глобального рынка распределенных реестров и технологии блокчейн составят 57,6%. Как ожидается, к концу обозначенного периода его объем превысит \$5 млрд.

«Проблема оракулов» распространена во всем мире и не позволяет обширное внедрение блокчейн-технологий

Нет ясности как получать достоверные данные для отправки их в блокчейн и работы с ними

Для решения «проблемы оракулов» можно использовать защищенные IoT-датчики для получения достоверной информации

Стоимость аренды должна зависеть от истраченных ресурсов



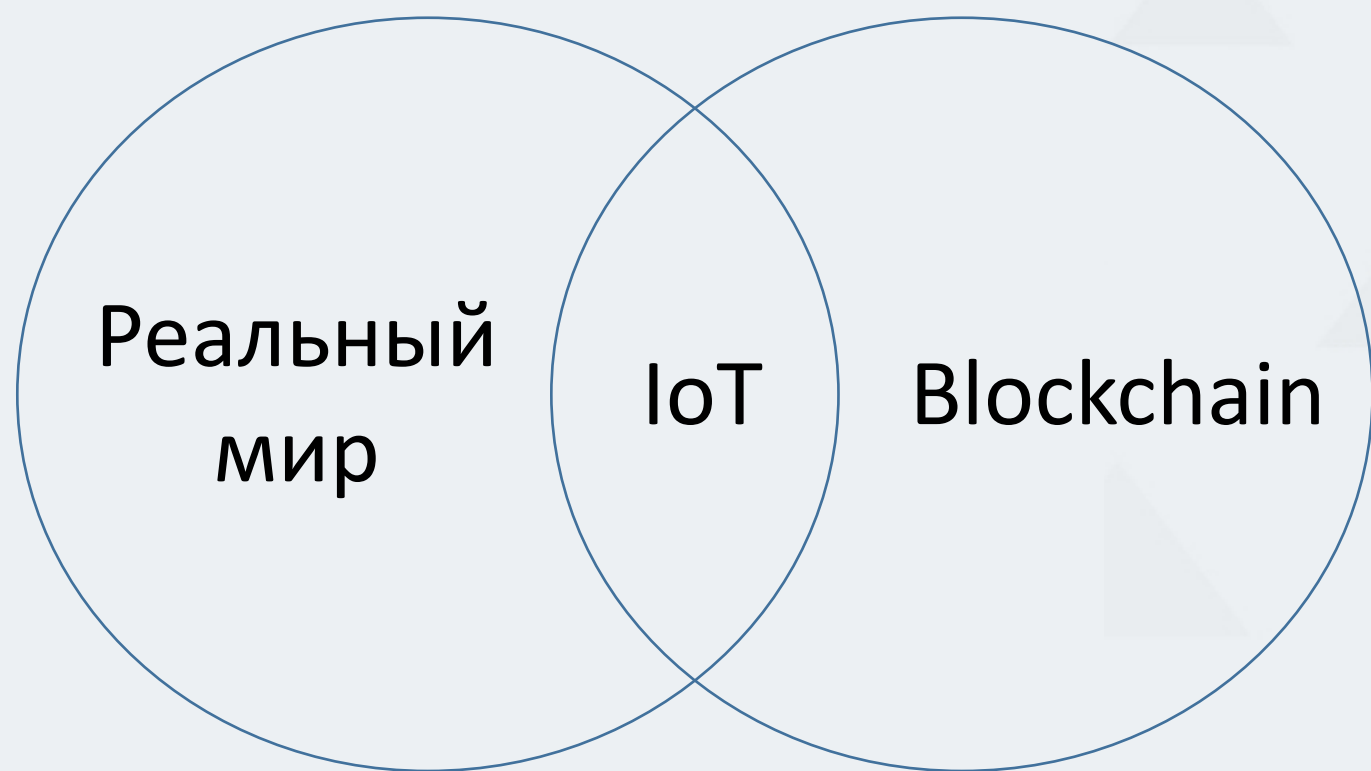
Что привносит проект на рынок?

Проект

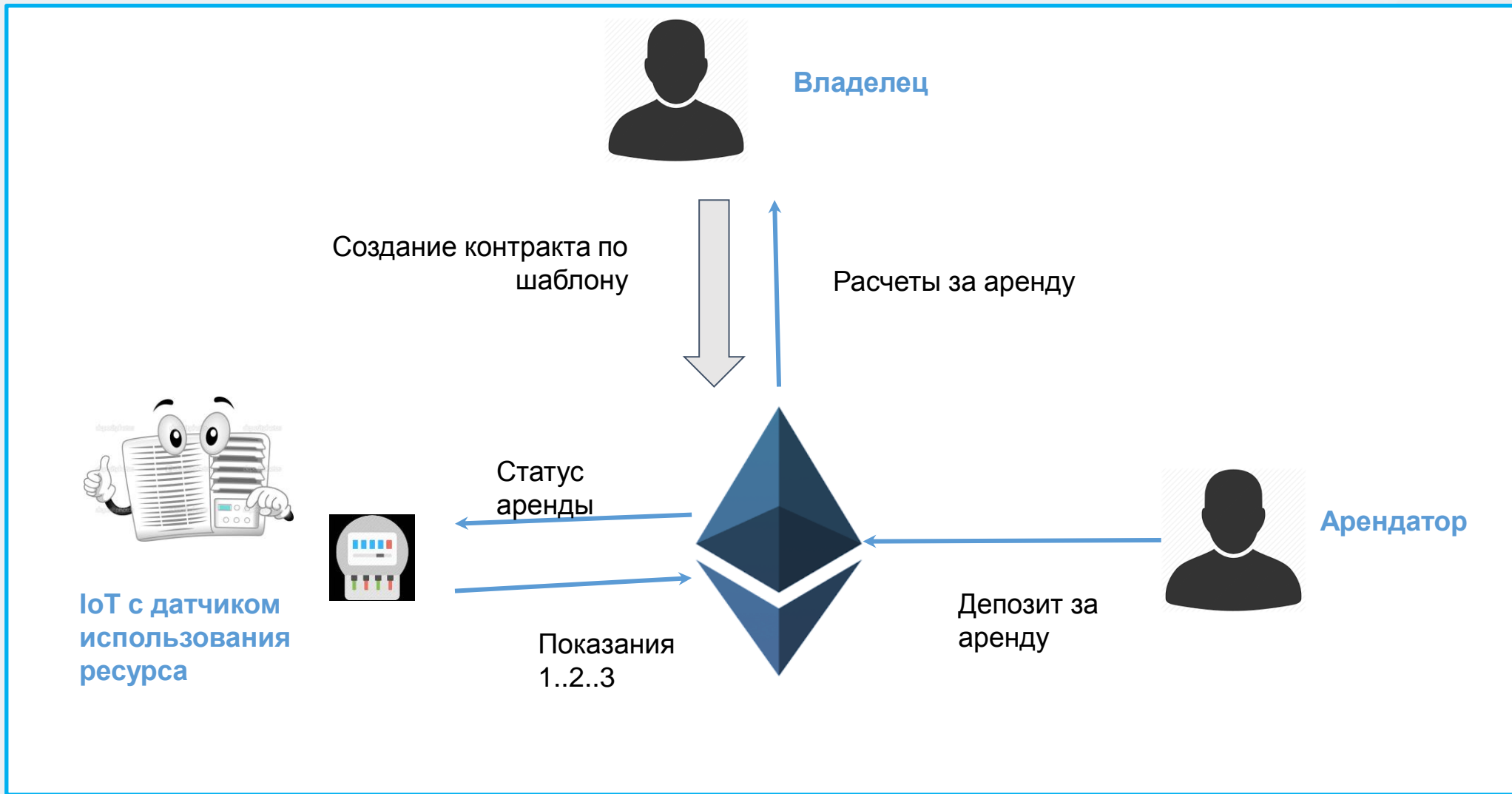
Аренда вещей с учетом потраченных ресурсов

Миссия и видение

Решение для P2P рынков аренды с расчетом по объему потребленного ресурса. Прозрачность и безопасность сделок между владельцем и арендатором.



Описание работы технологии



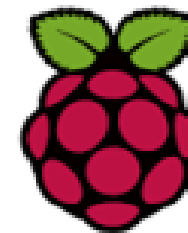
Датчик: Микрокомпьютер Raspberry

Функциональность: Считывание, обработка, отправка в blockchain полученных данных.
Вкл/выкл доступа к ресурсам

Стоимость: \$35

Преимущества: Скорость передачи в Blockchain
Возможность блокировки
Автономность

Недостатки: Необходима защита и
невозможность вмешательства



Внесение параметров своего прибора и измеряемых данных

Set your parameters

PARAMETER 1:

Name

Value

PARAMETER 2:

Name

Value

HEIGHT:

LAST BLOCK UTXO ROOT HASH:

CODE:



Получение индивидуального смарт-контракта с подключенным датчиком

```
browser/pay4meter (test).sol x
1 pragma solidity ^0.4.0;
2 contract Pay4meter {
3
4     address customer;
5     address sensor;
6     address owner;
7
8     bool lease_status; //true - now leasing; false - now not leasing
9     uint256 price; //price for 1 measurement
10    uint256 measurement; // measurement of sensor
11    uint256 begin_measurement;
12    uint256 balance; //now available
13
14    uint256 amountForOwner;
15
16    struct Leaser {
17        uint256 depositToReturn;
18        //address leaser;
19    }
20
21    mapping(address => Leaser) leasers;
22
23    function Pay4meter(address sensor_init, uint256 price_init) public {
24        measurement=0;
25        balance=0;
26        amountForOwner=0;
27        lease_status=false;
28        owner = msg.sender;
29        sensor = sensor_init;
30        price = price_init;
31    }
32
33    function createLease () public payable {
34        require(msg.value > price);
35        require(!lease_status);
36
37        customer = msg.sender;
38        begin_measurement = measurement;
39        lease_status = true;
40        balance = msg.value;
```



Сдача в аренду

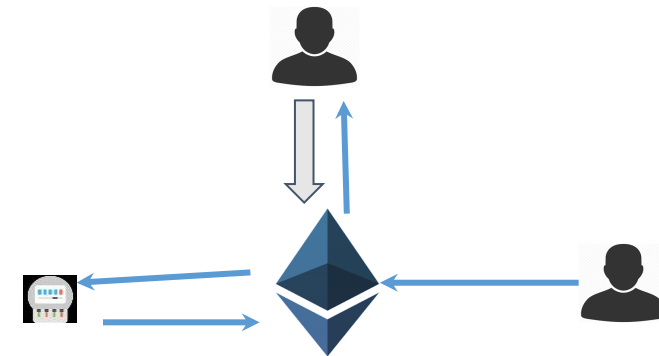
Покупка и установка нужного датчика



Создание индивидуального смарт-контракта

```
1 pragma solidity >=0.4.0;
2 contract Pay4meter {
3
4     address customer;
5     address sensor;
6     address owner;
7
8     bool lease_status; //true - now leasing; false - now not leasing
9     uint256 price; //price for 1 measurement
10    uint256 measurement; // measurement of sensor
11    uint256 begin_measurement;
12    uint256 balance; //now available
13
14    uint256 amountForOwner;
15
16-    struct Leaser {
17-        uint256 depositToReturn;
18-        //address leaser;
19-    }
20
21    mapping(address => Leaser) leasers;
22
23-    function Pay4meter(address sensor_init, uint256 price_init) public {
24-        measurement=0;
25-        balance=0;
26-        amountForOwner=0;
27-        lease_status=false;
28-        owner = msg.sender;
29-        sensor = sensor_init;
30-        price = price_init;
31-    }
32
33-    function createlease () public payable {
34-        require(msg.value > price);
35-        require(!lease_status);
36-
37-        customer = msg.sender;
38-        begin_measurement = measurement;
39-        lease_status = true;
40-        balance = msg.value;
```

Сдача собственности в аренду



Структура расходов

- Создание датчиков для регистрации разных типов ресурсов
- Защита датчиков
- Разработка платформы

Потоки доходов

- Установка датчика
- Создание индивидуального смарт-контракта
- Процент с транзакции

Конкурентные преимущества и ключевые факторы успеха

Что предлагают на рынке?

При аренде не
учитываются ресурсы
Непрозрачность
потребляемых ресурсов

Что предлагает проект?

Внедрение смарт-контракта для
прозрачного взаимодействия и
экономии ресурсов

Ключевые факторы успеха

Смарт-контракт
(прозрачность и
надежность)

IoT-датчик
(связь реального
мира с blockchain)

Работающий
прототип

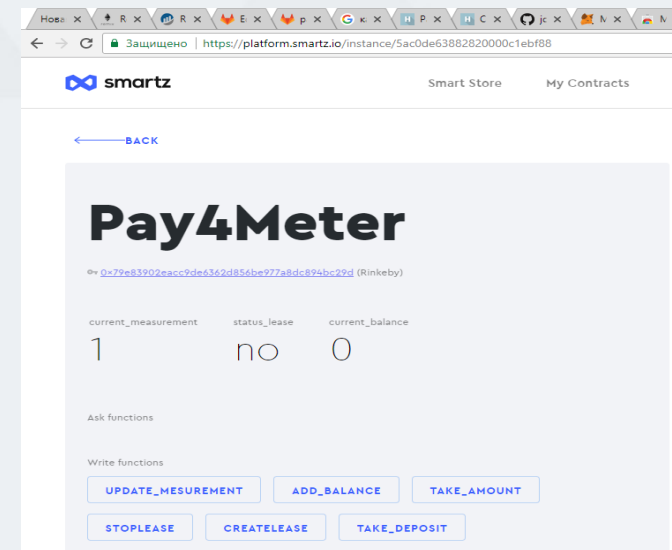
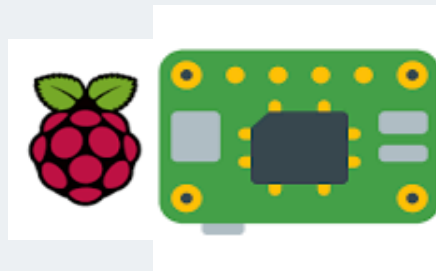
Что уже сделано?

Смарт контракт на Solidity
(Ethereum)

Интеграция микропроцессора
в ноду Rikenby с передачей
показаний и обратной связью

Шаблон контракта

```
browser/pay4meter (test).sol *
1 pragma solidity ^0.4.0;
2 contract Pay4meter {
3
4     address customer;
5     address sensor;
6     address owner;
7
8     bool lease_status; //true - now leasing; false - now not leasing
9     uint256 price; //price for 1 measurement
10    uint256 measurement; // measurement of sensor
11    uint256 begin_measurement;
12    uint256 balance; //now available
13
14    uint256 amountForOwner;
15
16    struct Leaser {
17        uint256 depositToReturn;
18        //address leaser;
19    }
20
21    mapping(address => Leaser) leasers;
22
23    function Pay4meter(address sensor_init, uint2
24        measurement=0;
25        balance=0;
26        amountForOwner=0;
27        lease_status=false;
28        owner = msg.sender;
29        sensor = sensor_init;
30        price = price_init;
31    }
32
33    function createLease () public payable {
34        require(msg.value > price);
35        require(!lease_status);
36
37        customer = msg.sender;
38        begin_measurement = measurement;
39        lease_status = true;
40        balance = msg.value;
```



Команда проекта



Антон Печеркин

Роль в команде:

- 10 Frontend
- 10 Маркетинг
- 10 Бизнес-модель и аналитика

Контакты:

- 10 Email: aa.pecherkin@yandex.ru
- 10 Тел: +7 (965) 241-27-08
- 10 Telegram: @AAPecherkin



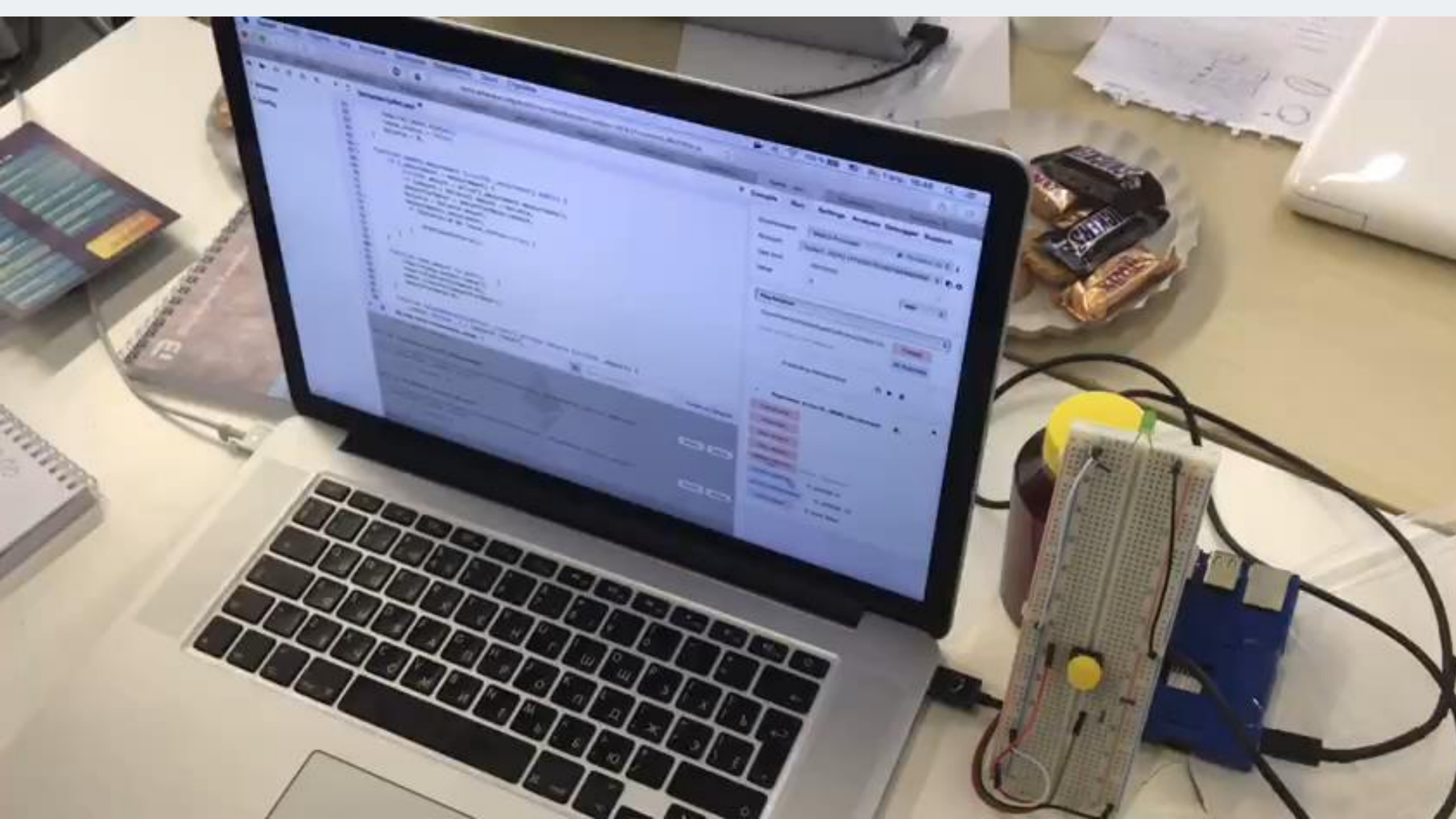
Александр Дауркин

Роль в команде:

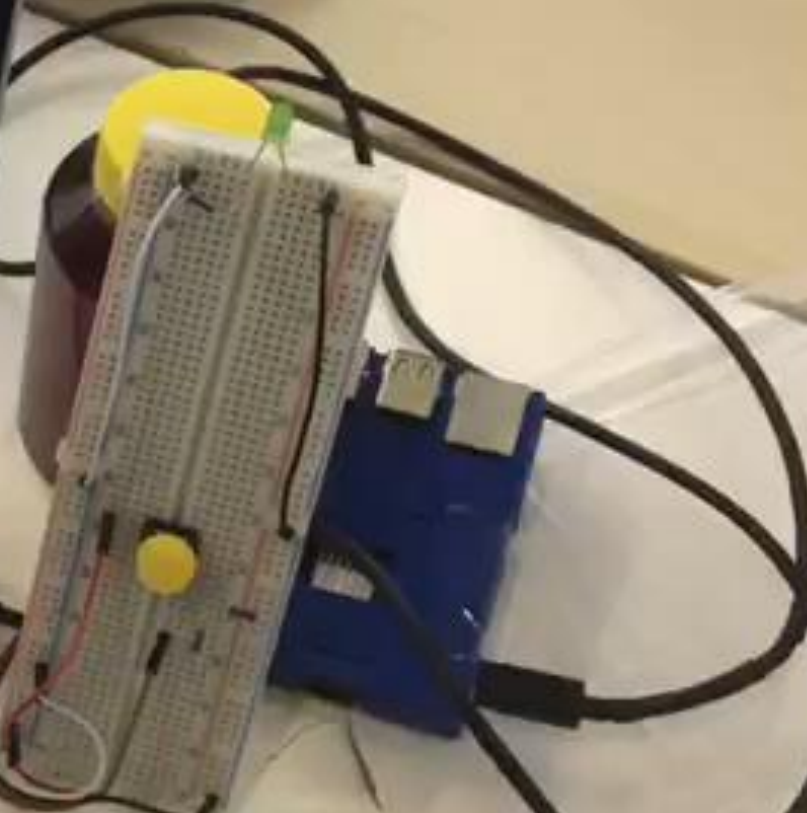
- 10 Backend
- 10 Моделирование Raspberry
- 10 Аналитика

Контакты:

- 10 Email: adaurkin@rambler.ru
- 10 Тел: +7 (917) 503-79-63
- 10 Telegram: @adaurkin



```
int main() {
    Serial.begin(9600);
    while (1) {
        if (Serial.available()) {
            char c = Serial.read();
            if (c == '1') {
                digitalWrite(LED_PIN, HIGH);
            } else if (c == '0') {
                digitalWrite(LED_PIN, LOW);
            }
        }
    }
}
```





Pay4Meter

Рады сотрудничеству!

Контакты:

⑩ Email: aa.pecherkin@yandex.ru

⑩ Тел: +7 (965) 241-27-08

⑩ Telegram: [@AAPecherkin](https://t.me/AAPecherkin)