**Как организовать выполнение проекта по управлению мобильным роботом**

## Инструкция для преподавателя

**Пререквизиты курса**

- быстрый Wi-Fi в месте проведения мероприятия;

- мобильный робот (один или несколько);

- для каждого мобильного робота:

1. Плата с микроконтроллером - например, плата RobotDyn Mini пропаянная
2. Кабель USB–microUSB
3. Источник питания - например, внешний аккумулятор
4. Два датчика дистанции пропаянных
5. Соединительные провода

- компьютер с установленными средой Arduino IDE и всеми необходимыми библиотеками (об этом далее)

- доступ администратора к «инстансу» ThingWorx + импортированные в платформе сущности для проекта.

Этапы:

1. Предварительная подготовка к занятию
2. Проведение занятия

## Предварительная подготовка

Предварительная подготовка занимает около четырех часов.

**Подготовка мобильного робота**

В плату мобильного робота должен быть загружен скетч **SmartMobileRobot.ino**, который находится по ссылке: <https://drive.google.com/open?id=1rVTbyg1ZKYIo1FRW7cM8kPOScc9P2pK4>

**Подготовка комплектов**

Для проведения этого мастер-класса **каждое** рабочее место необходимо обеспечить оборудованием в следующем составе:

1. Плата RobotDyn Mini пропаянная – 1 шт.
2. Кабель USB–microUSB – 1 шт.
3. Датчик дистанции пропаянный – 1 шт.
4. Соединительный провод – 1 шт.

Для подготовки комплектов к работе выполните пункты 1) - 3), описанные ниже, **на всех компьютерах**, которые будут использоваться на мастер-классе. Каждый комплект нужно настраивать на том компьютере, к которому он будет подключаться во время мастер-класса.

1. **Установка программного обеспечения**
2. Посмотрите код после **USB-TTL** на обратной стороне платы.
3. Если код **CP2104**, перейдите по ссылке <https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers>, скачайте и установите драйвер.
4. Если код **CH340**, перейдите по ссылке <https://sparks.gogo.co.nz/ch340.html>, скачайте и установите драйвер.
5. Перейдите по ссылке <https://www.arduino.cc/en/Main/Software> и скачайте Arduino IDE.
6. Установите Arduino IDE и откройте.
7. Нажмите Файл – Настройки.
8. В поле **Дополнительные ссылки для Менеджера плат** вставьтеадрес <http://arduino.esp8266.com/stable/package_esp8266com_index.json>.
9. Закройте Настройки.
10. Нажмите Инструменты – Плата – Менеджер плат.
11. В строке с поиском введите **esp8266**.
12. Выберите **esp826 by ESP8266 Community** версия **2.1.0** и нажмите **Установить**.
13. Подождите, пока завершится установка, и закройте менеджер плат.
14. Нажмите Инструменты – Плата, и выберите **WeMos D1 R2 & mini** в списке.
15. Нажмите Скетч – Подключить библиотеку – Управлять библиотеками.
16. В строке с поиском напишите **arduinojson.**
17. Выберите **ArduinoJson** последней версии и нажмите **Установить.**
18. Подождите, пока завершится установка, и закройте менеджер библиотек.
19. Нажмите Инструменты - Upload Speed - 115200.

**Если участники работают на мастер-классе за своими ноутбуками, то можно отправить им инструкцию по установке программного обеспечения заранее, чтобы они все сами установили.**

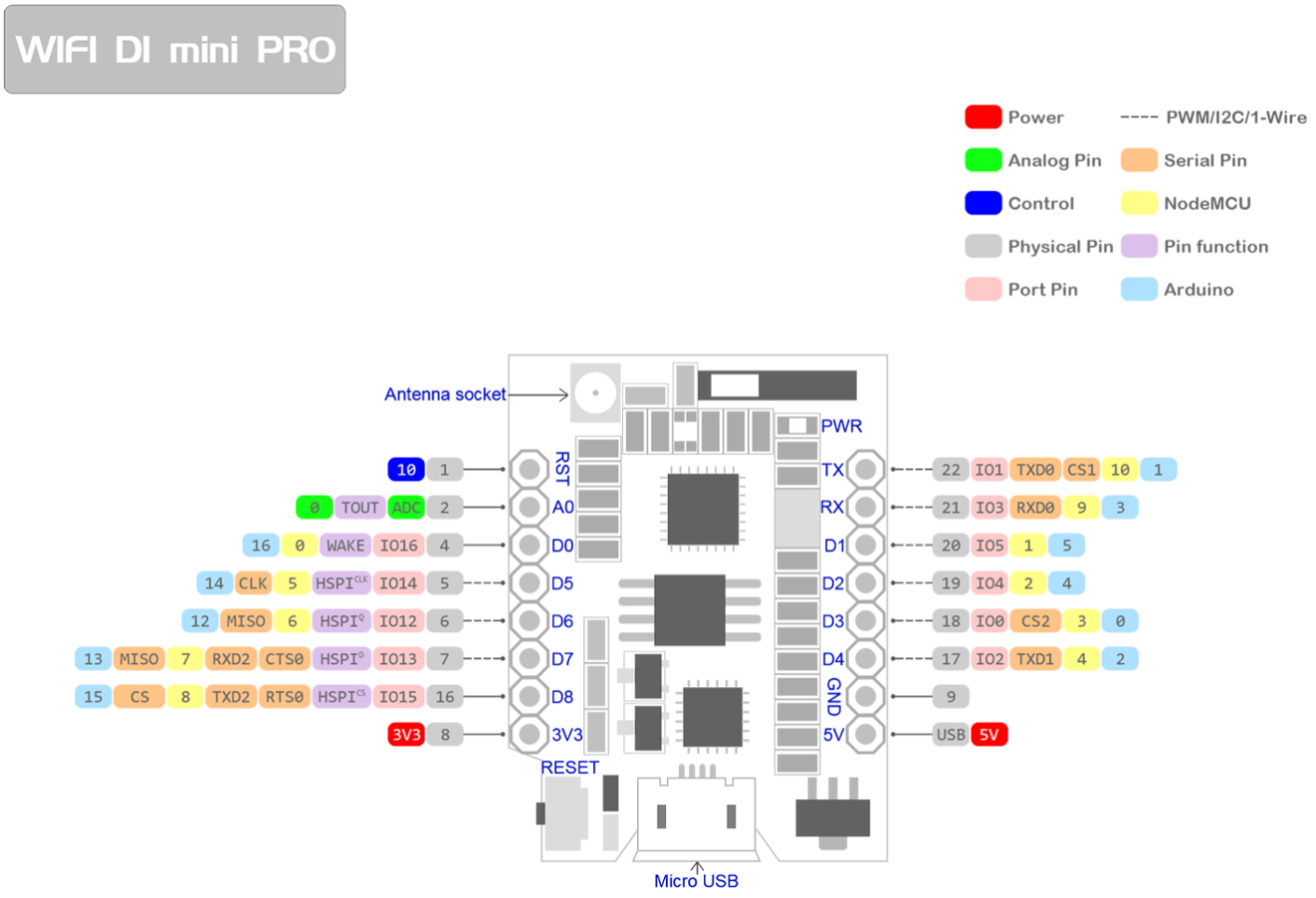
## Загрузка скетча в микроконтроллер платы

1. Перейдите по ссылке <https://drive.google.com/open?id=1rVTbyg1ZKYIo1FRW7cM8kPOScc9P2pK4>
2. Скачайте скетч **SmartBoard.ino.**
3. Подключите плату к компьютеру через кабель microUSB в комплекте.
4. В программе Arduino IDE откройте скетч **SmartBoard.ino**.
5. Нажмите **Инструменты** **-** **Порт** и выберите порт, который соответствует плате.

Его можно определить, отключив плату от компьютера и снова подключив обратно (соответствующий порт исчезнет и вновь появится).

1. В окне со скетчем нажмите в левом верхнем углу для загрузки скетча на плату.
2. Дождитесь, пока скетч загрузится на плату.

## Подключение датчика



1. Подключите соединительный провод к контактам датчика дистанции.
2. Соедините контакт **VCC** датчикас контактом **3V3** контроллера.
3. Соедините контакт **GND** датчикас контактом **GND** контроллера.
4. Соедините контакт **OUT** датчикас цифровым входом **d6** контроллера.
5. Убедитесь, что датчик реагирует на препятствия (загорается синий индикатор). Если нет, то настройте чувствительность датчика, поочередно вращая два настроечных винта на нем.

**Подготовка компьютеров к работе**

1. **Импорт сущностей проекта**

Скачайте файл с сущностями [здесь](https://drive.google.com/open?id=1aM9RpwbP-ijgNWvp4PUDUYCkiCvsXlDf) и импортируйте его в ваш инстанс платформы.

1. **Создание пользователей для учащихся**

Создайте группу (User Group) для учащихся проекта, а также пользователей (Users). Дайте пользователям необходимые права.

**План проведения занятия**

**Теория**

**Вводные слова**

Цель этого мастер-класса – показать новый подход в робототехнике, когда вместо автономных роботов используются роботы, управление которых вынесено в облачное приложение.

На мастер-классе вы

1. поймете, как благодаря новым технологиям даже такой несложный мобильный робот способен стать самостоятельным;
2. создадите простое приложение Интернета вещей и организуете взаимодействие этого приложения с реальным устройством;
3. обучите мобильного робота вести себя «разумно», задав «рефлексы» и «волю».

Все участники мастер-класса смогут продолжить работу над проектом самостоятельно после окончания выставки, например, задать индивидуальность своего питомца или даже научить его «общаться» с другими роботами.

**Подход**

Вынесение управления роботом в облако дает важные преимущества:

1. Вам совершенно необязательно делать дорогого работа с замысловатым техническим оснащением и записывать в его память сложные алгоритмы работы, чтобы он стал вести себя «умно». Даже самые простые вещи можно сделать «умными», если просто научить их общаться с вами или друг с другом через Интернет. Мы можем вынести всю логику в облако, а на борту робота оставить только физическое управление его механизмами и интерфейс для приема и передачи информации. Это дает возможность использовать ресурсы, не ограниченные характеристиками робота, а самого робота сделать простым и дешевым. Поэтому даже такой простой мобильный робот, как этот, оборудованный только двумя датчиками препятствий и маленьким контроллером с Wi-Fi модулем для отправки и приема данных, может иметь разумное поведение.
2. Для внесения изменений в поведение робота не нужно постоянно загружать в него новые алгоритмы работы. Всю настройку можно выполнять откуда угодно, не обязательно находиться рядом с роботом. А робот будет параллельно приобретать обновленное поведение.
3. Робот становится «умнее» за счет того, что «общается» с другими объектами: другими роботами, «умными» вещами и внешними сервисами.
4. Вынесение логики в облако порождает новый подход к программированию поведения робота: событийно-ориентированное программирование. Роботу не задается поведение в виде алгоритма - того, что ему нужно делать постоянно – а в виде реакций на определенные события. Роботу не говорится, что ему делать «просто так», а только в ответ на происходящие события. Такое поведение робота более естественно, потому что именно так ведут себя живые системы. Такой подход к программированию, в свою очередь, более интуитивно понятен по сравнению с алгоритмическим подходом, поэтому ему можно учить даже детей.

**Конфигурация**

На этом мастер-классе мы будем работать с простым мобильным роботом, поведение которого задается через специальное облачное приложение.

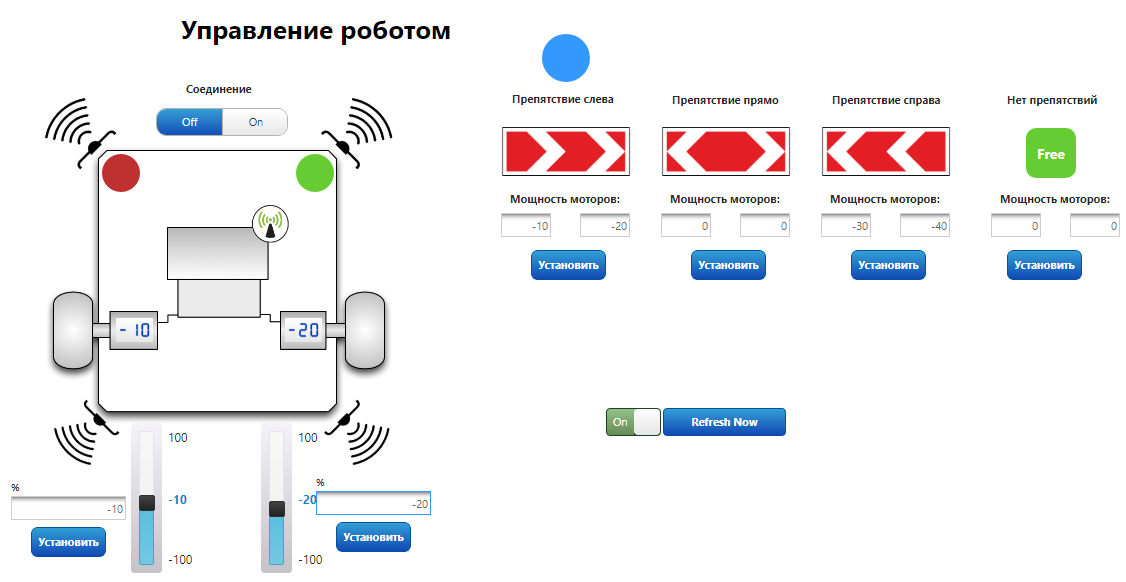
1. **Робот**

Для работы робота не него устанавливается специальный модуль для сбора данных о наличии объектов около него, отправки их в Интернет и управления сервоприводами моторов робота.

Модуль включает в себя:

1. источник питания робота;
2. два датчика дистанции, установленные в передней части робота, которые определяют наличие объектов около него;
3. ардуино-совместимый микроконтроллер с Wi-FI, который собирает данные с датчиков и отправляет эти данные в Интернет. Кроме того, контроллер получает из облака целевые значения мощностей моторов робота и управляет подключенными к нему сервоприводами. Задание мощностей обоих моторов мобильного робота полностью определяет его движение.
4. **Облачное приложение**

Поведение робота задается через специальное облачное приложение.



В этом приложении:

1. показывается текущее состояние робота:
   1. наличие соединения;
   2. состояния датчиков дистанции;
   3. мощности вращения колес;
2. задаются поведение робота через задание скоростей вращения его колес:
   1. ручное задание скоростей;
   2. реакция на событие – объект слева;
   3. реакция на событие – объект справа;
   4. реакция на событие – объект прямо по курсу;
   5. реакция на событие – объект не обнаружен (движения по умолчанию).

Скорость колеса:

* -100 – «полный назад»
* 0 – колесо не вращается
* +100 – «полный вперед»

**Примечание:**

Для юстировки колес робота нужно при необходимости подкрутить винты обоих сервоприводов робота так, чтобы при задании нулевой скорости вращения колеса робота не вращались.

**Демонстрация поведения робота**

Демонстрируем поведение робота:

* показываем, как он реагирует на объекты;
* пробуем задавать разные реакции на обнаружение объектов
* предлагаем участникам попробовать то же со своими роботами (если есть свои за их рабочими местами)

**Создание приложения Интернета вещей**

Такое облачное приложение для задания поведения робота сделано на платформе Интернета вещей ThingWorx. Покажем, как создаются такие приложения. Для этого сделаем новое приложение для получения данных с датчика дистанции и свяжем его с физическим датчиком. Датчики, подключенные к плате для отправки данных с него в Интернет, находятся за каждым рабочим местом. Такое же оборудование установлено на нашем мобильном роботе.

После того, как научитесь отправлять данные с датчиков, сможете сами добавить задание поведения мобильного робота, как это сделано в нашем приложении.

**Получение доступа к платформе Thingworx (при использовании версии EEx)**

На ваших компьютерах уже открыта форма для регистрации нового пользователя.

1. Введите ваши личные данные:

First Name – имя

Last Name – фамилия

I am studying… - напишите **iot**

University or Organization - образовательное учреждение

Email – напишите вашу почту

Create a username… - придумайте логин, **используйте латинские буквы**

Create a password… - придумайте пароль

Поставьте галочку напротив **I agree…**

1. Нажмите **Register Now**.

Появится личный кабинет, через который можно заходить в платформу.

Кроме того, в личном кабинете на вкладке **Academic Resources** вам доступны полезные образовательные ресурсы, включая бесплатные онлайн-курсы от университета Интернета вещей (IoT University).

1. Нажмите **Launch ThingWorx**.
2. В появившемся окне введите логин и пароль, которые вы указали при регистрации.
3. Дождитесь загрузки платформы Thingworx.

Доступ к платформе по этому логину и паролю останется у вас и после мастер-класса.

**Краткий обзор платформы**

ThingWorx – ведущая платформа в отрасли Интернета вещей.

На главной странице все объекты платформы сгруппированы по разделам: **Modeling** (моделирование), **Visualization** (визуализация), **Data Storage** (хранение данных) и т.д.

На этом мастер-классе нам понадобится три раздела: Modeling, Visualization и Security.

Сейчас многие объекты физического мира собирают и отправляют в Интернет огромное количество информации. В связи с этим появляется смысл говорить об отдельных объектах цифровой действительности – **«цифровых двойниках»** (Digital Twins) – «копий» физических объектов, которые объединяют в себе информацию о соответствующих им физических объектах. Любая физическая вещь может быть представлена в информационном мире с помощью одного или нескольких виртуальных вещей, однако виртуальная вещь может существовать и без какой бы то ни было соответствующей ей физической вещи.

В платформе ThingWorx такие «цифровые двойники» называются **Things (вещи)**. Они находятся в разделе **Modeling**.

Создание приложения Интернета вещей на платформе ThingWorx состоит из трех основных этапов:

1. Создание виртуальных вещей и организация взаимодействия между ними. Этот пункт соответствует разделу **Modeling**.
2. Визуализация необходимых данных на веб-странице. Этот пункт соответствует разделу **Visualization**.
3. Организация взаимодействия виртуальных вещей с физическими. Этот пункт частично соответствует разделу **Security**, частично **Modeling**.

В результате получается система, в которой поведение физических объектов задается поведением их цифровых двойников, а вся необходимая информация отображается и задается на веб-странице.

Сейчас давайте подробно рассмотрим каждый из этих этапов.

**Далее участники создают приложение по инструкции (см. инструкцию по созданию приложения).**

**Моделирование**

**Создание вещи**

Вещь (Thing) – это тот самый «цифровой двойник» физического объекта. Давайте создадим новую вещь – цифровую копию нашего мобильного робота.

1. На вкладке **Home** (Домой), разделе **Modeling** (Моделирование), выберите **Things** (Вещи) и нажмите **New** (Новая).
2. Введите имя **RobotThing** в поле Name.
3. В поле **ThingTemplate** напишите **generic** и выберите общий шаблон **GenericThing**.

**Добавление свойств к вещи**

Давайте зададим свойства робота: состояние датчика дистанции и имя робота. Эти свойства соответствуют свойствам физического мобильного робота. Названия свойств принято писать в одно слово, каждое слово с большой буквы.

1. Перейдите на вкладку **Properties**.
2. Нажмите **+Add My Property.**
3. В появившемся окне напишите **RobotName** в поле **Name** (имя).
4. В поле **Description** (описание) напишите **Имя робота.**
5. В поле **Base Type** (тип) выберите **String.**
6. Поставьте галочку напротив **Persistent** (сохраняемый).
7. Нажмите **Done and Add.**
8. В появившемся окне напишите **BarrierSensor** в поле **Name** (имя).
9. В поле **Description** (описание) напишите **Состояние датчика дистанции.**
10. В поле **Base Type** (тип) выберите **Number.**
11. Поставьте галочку напротив **Persistent** (сохраняемый).
12. Нажмите **Done.**
13. Нажмите **Save** вверху экрана**.**

**Визуализация**

**Создание мэшапа**

Мэшап («сборка») – веб-страница для визуализации данных приложения Интернета вещей. Мэшап может объединять данные из нескольких совершенно разных источников. В платформе ThingWorx мэшапы создаются с помощью специального инструмента – Mashup Builder.

Давайте сделаем мэшап для вывода информации о состоянии датчика дистанции нашего мобильного робота.

1. На вкладке **Home** (Домой), разделе **Vizualization** (Визуализация), выберите **Mashups** (Мэшапы) и нажмите **New** (Новый).
2. Оставьте предопределенные настройки (Mashup: Page, Layout Options: Responsive) и нажмите **Done**.

Это Mashup Builder. На белой (рабочей) области по центру показывается, как будет выглядеть готовый мэшап. Слева есть список виджетов, которые перемещаются на рабочую область и используются для вывода и ввода нужной информации. Давайте сохраним мэшап и продолжим работу над ним.

1. Нажмите кнопку **Info** (Информация) вверху экрана.
2. Введите имя **RobotMashup** в поле Name.
3. Нажмите **Save** вверху экрана.
4. Нажмите **Edit** (Редактировать) вверху экрана, затем нажмите **Design**.

**Добавление виджетов к мэшапу**

На мэшапе будет дисплей для отображения данных с датчика дистанции, подпись к этому дисплею и виджет для автообновления данных.

1. На вкладке **виджеты** (Widgets) отфильтруйте или найдите виджет **LED Display**.
2. Перетащите его на рабочую область мэшапа.
3. В появившемся окне нажмите **Yes** для добавления на мэшап панели.
4. На вкладке **виджеты** (Widgets) отфильтруйте или найдите виджет **Label**.
5. Перетащите его на рабочую область мэшапа и расположите над виджетом LED Display.
6. В свойствах виджета **Label** (левая нижняя область экрана) найдите поле **Text** инапишите там **Датчик дистанции**.
7. На вкладке **виджеты** (Widgets) отфильтруйте или найдите виджет **Auto Refresh**.
8. Перетащите его на рабочую область мэшапа.
9. В свойствах виджета **Auto Refresh** (левая нижняя область экрана) найдите поле **RefreshInterval** (интервал между обновлениями в секундах), напишите там **3** и нажмите **Enter**.

**Связывание виджетов и сервисов**

Для отображения информации на виджетах нужно связать их с данными. Мы свяжем виджет **LED Display** со свойством **BarrierSensor** вещи **RobotThing**, чтобы дисплей выводил текущее значение с датчика дистанции.

1. На вкладке **Data** (правая часть экрана) нажмите на зеленый "плюс".
2. В появившемся окне в поле **Select Entity** начните писать имя вещи, которую вы создали (**RobotThing**) и выберите ее из списка результатов поиска (Search Results).
3. В поле **Select Services**, напишите **getproperties** для фильтра сервиса получения свойств GetProperties**.**
4. Нажмите на голубую стрелку, указывающую вправо, чтобы выбрать сервис получения свойств **GetProperties.**
5. Поставьте галочку у **MashupLoaded?** (запускать сервис при загрузке мэшапа) и нажмите **Done**.
6. В правой части экрана разверните красный заголовок, нажав на знак **+** слева от **GetProperties**.
7. Привяжите виджет **LED Display**:
   1. Перетащите **BarrierSensor** под красным сервисным заголовком “GetProperties” на виджет **LED Display** на правом верхнем прямоугольнике рабочей области.
   2. Выберите **Data** как цель привязки (Binding Target) на всплывающем окне.
8. Привяжите виджет **Auto Refresh**:
9. Кликните на виджете **Auto Refresh** на рабочей области.
10. Наведите на стрелочку в левом верхнем углу виджета.
11. Выберите событие **Refresh**и перетащите его на красный сервисный заголовок **GetProperties**на вкладке Data (правая часть экрана).
12. Нажмите **Save** вверху экрана.
13. Нажмите **Cancel Edit** вверху экрана.
14. Нажмите **View Mashup** вверху экрана.

В браузере откроется новая вкладка с веб-страницей приложения.

**Проверка работы мэшапа**

1. Откройте вещь **RobotThing** и перейдите во вкладку **Properties**.
2. Нажмите **Set** справа от свойства **RobotName**.
3. В появившемся окне введите имя робота, какое хотите, и нажмите **Set**.
4. Нажмите **Set** справа от свойства **BarrierSensor**.
5. В появившемся окне введите значение **1** и нажмите **Set**.
6. Откройте вкладку в браузере с веб-страницей приложения и убедитесь, что значения, которые вы задали, изменились.

**Организация связи модуля с платформой**

Мы создали цифровую «копию» нашего мобильного робота и вывели информацию о ней на веб-странице. Теперь мы свяжем эту цифровую «копию» с физическим устройством, и тогда на мэшапе будет отображаться информация об этом физическом устройстве.

**Создание сервиса для взаимодействия с реальным устройством**

Для отправки данных с датчика дистанции реальное устройство генерирует POST запрос, который запускает сервис в платформе, посылая свойство, отвечающее за состояние датчика дистанции, в качестве входного параметра этого сервиса. Давайте сделаем этот сервис.

1. Выберите вещь **RobotThing**.
2. Перейдите на вкладку **Services**.
3. Нажмите **+Add My Service**.
4. В поле **Name** напишите **InputService.**

Теперь нужно добавить входные параметры. Плата генерирует запрос в виде строки, в которой значению на датчике дистанции соответствует свойство **d6** (датчик подключен к выходу d6 платы). Свойство d6 и будет входным параметром сервиса.

1. Перейдите во вкладку **Inputs/Outputs**.
2. Нажмите **+Add** справа от Inputs.
3. В появившемся окне в поле **Name** напишите **d6.** Тип (Base Type) оставьте **String**.
4. Нажмите **Done** (готово).
5. В поле **Script** (скрипт) напишите **me.BarrierSensor = d6;**
6. Нажмите **Done** (готово).
7. Нажмите **Save** вверху экрана.

**Создание ключа приложения**

Для обеспечения безопасности передачи данных каждый запрос содержит также

«ключ приложения» (AppKey).

Давайте создадим ключ приложения.

1. В поисковой строке в верхней части экрана напишите **“+a”**и выберите команду**+Application Key**.
2. В поле **Name**в появившемся окне напишите **RobotAppKey**.
3. В поле**User Name Reference**начните писать ваш логин и выберите появившийся объект.
4. Нажмите **Save**вверху экрана.

После сохранения в поле **keyId**появится ключ.

**Ключ приложения будет использоваться при подключении реального устройства к платформе.**

**Настройка микроконтроллера**

1. Подключите плату к компьютеру через кабель microUSB, если не подключена.
2. Откройте скетч **SmartBoard\_RobotProject.ino** с помощью программы **Arduino IDE**.
3. В программе нажмите **Инструменты** - **Монитор порта**.
4. Нажмите кнопку **Reset** на плате.

В Мониторе порта вы должны увидеть надпись:

“*Press any key for setup or wait 5 sec for start”.*

1. **У вас есть пять секунд для того, чтобы перейти в меню настроек.** По истечении этого срока плата переключится в режим штатной работы с облаком.
2. Кликните на строку запросов в мониторе порта, нажмите **Пробел** и затем **Enter.**

Появится меню настроек платы.

1. Если вы **не успели** перейти в меню настройки платы, и начался поиск сети, снова нажмите **Reset** на плате и затем выполните действия пункта **4b.**
2. Убедитесь, что в мониторе порта выбран режим **«Нет конца строки».**
3. Нажмите **1** и **Enter**.
4. Напишите **название Wi-Fi сети** (SSID) , к которой плата будет подключаться,в формате
   1. **<название сети >#**, и нажмите **Enter**.
   2. Например, если имя сети dlink, напишите dlink# и нажмите Enter.
5. Нажмите **2** и **Enter**.
6. Напишите **пароль от Wi-Fi сети** в формате **<пароль >#** и нажмите **Enter**.
7. Нажмите **3** и **Enter**.
8. Напишите имя вещи в формате **<имя вещи>#** и нажмите Enter.
   1. **Имя вещи – это имя той самой вещи, которую Вы создали (включая окончание с вашим логином).**
9. Нажмите **4** и **Enter**.
10. Напишите **InputService#** и нажмите **Enter**.
11. Нажмите **5** и **Enter**.
12. Напишите ключ приложения (AppKey) в формате **<ключ>#** и нажмите Enter.
13. **Ключ приложения – этот код в поле keyId ключа RobotAppKey, который вы создали.**
14. Нажмите **0** и **Enter**.
15. Нажмите **y** и **Enter** для подтверждения выхода из меню.

**Просмотр данных с модуля на мэшапе**

Теперь данные, передаваемые в платформу устройством, можно увидеть на мэшапе.

Для этого откройте вкладку в браузере с веб-страницей приложения и убедитесь, что значения с датчика дистанции подключенного модуля отображаются.

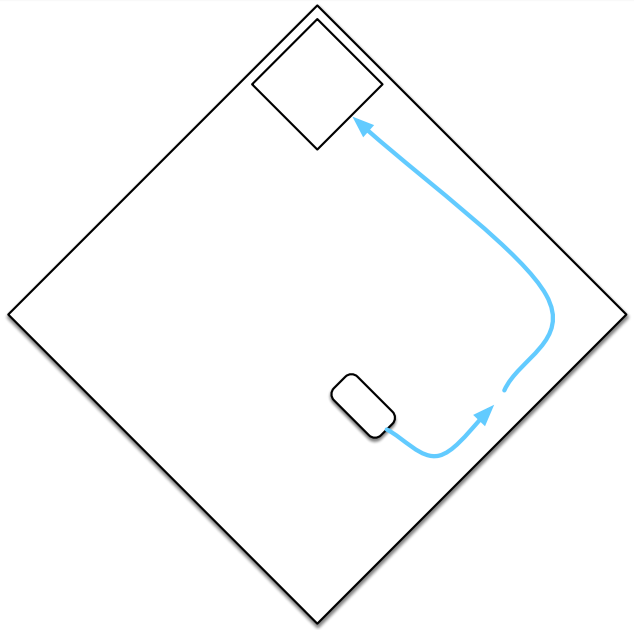
**Дополнительное задание**

Кто закончил раньше, можно доделать мэшап: добавить картинку робота с помощью виджета **Image** и вывести имя робота с помощью виджета **Label**.

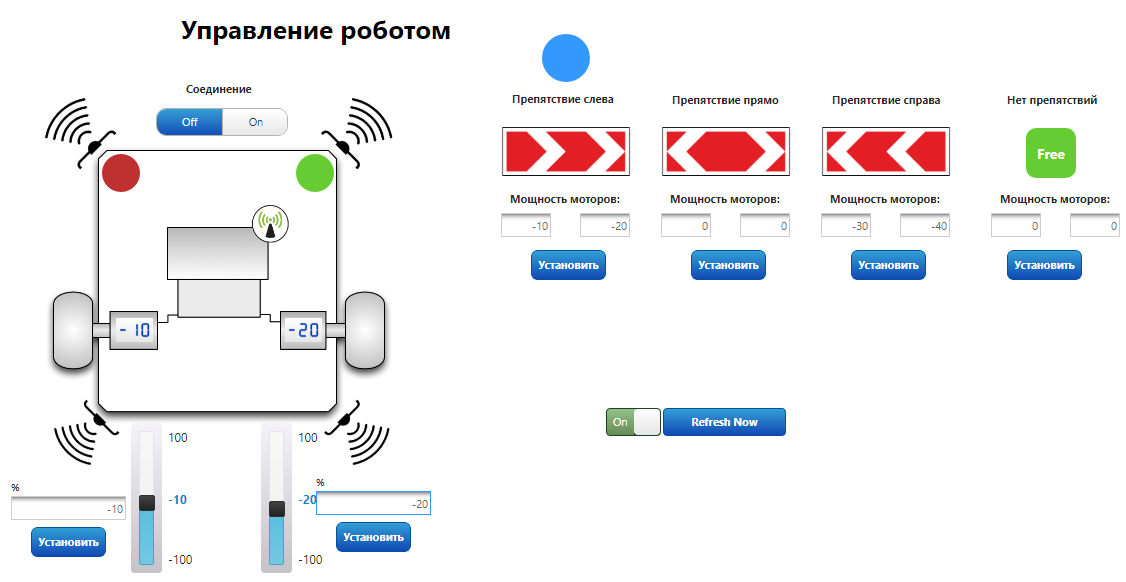
**Работа с мобильным роботом**

Мобильный робот использует ту же плату, которую мы сейчас использовали. Для экономии времени мы заранее сделали мэшап для управления и настройки робота.

В этой части задания вы через веб-страницу зададите «рефлексы» робота, которые помогут ему добраться в зону финиша. Робот оборудован двумя датчиками дистанции, которые позволяют ему обнаруживать объекты.



Веб-страница для мониторинга и настройки робота

**Вход в приложение - ДЕМО**

1. Перейдите по адресу <https://academic-educatorsextension.portal.ptc.io/Thingworx/FormLogin/My_Robot>
2. В поле **Name** напишите **Robot**.
3. В поле **Password** напишите **Robot**.
4. Нажмите **Submit**.

В приложении вам нужно задать реакции робота на четыре события:

1. объект слева;
2. объект справа;
3. объект прямо по курсу;
4. объект не обнаружен (движение по умолчанию)

так, чтобы роботу удалось добраться в зону финиша без столкновений с препятствиями.

Для этого нужно подобрать мощности моторов, которые устанавливаются как реакция на наступление этих событий.