Automonous Systems

INTERNET OF THINGS

Fábio Silva
fabiosilva@di.uminho.pt

Index

Internet of Things (IoT)

IoT Communications

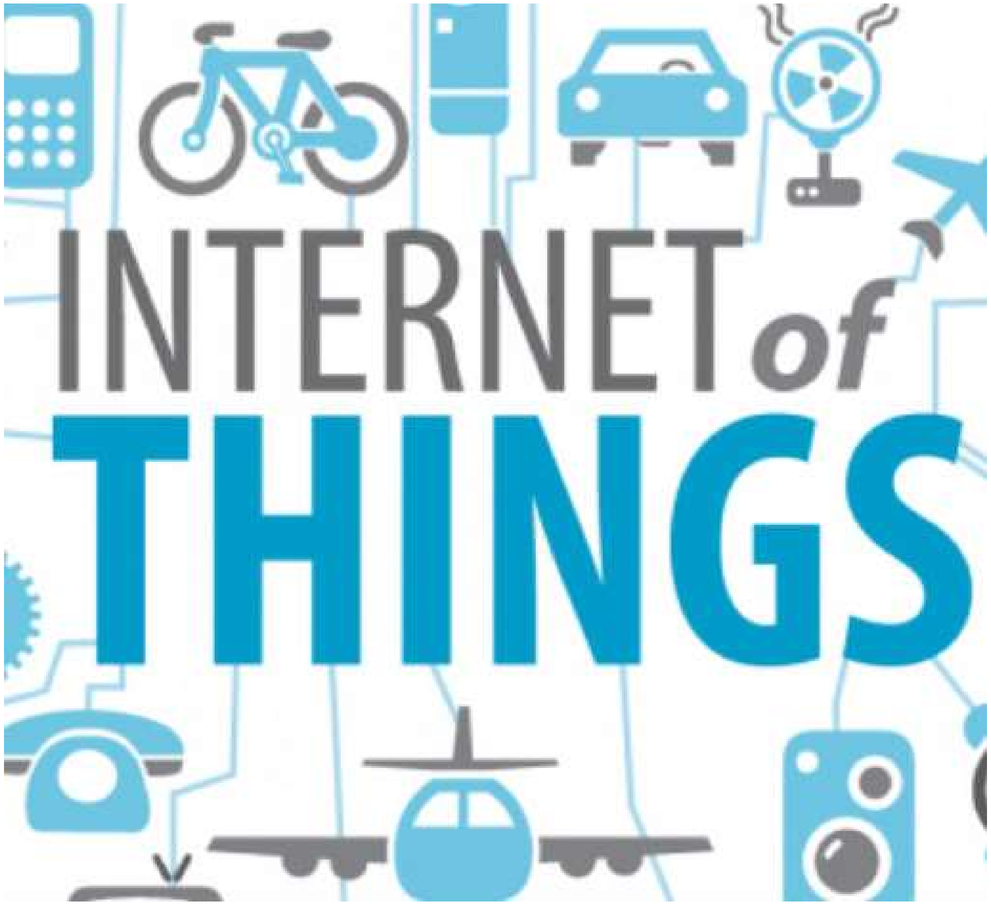
Intelligent Environments

PHESS

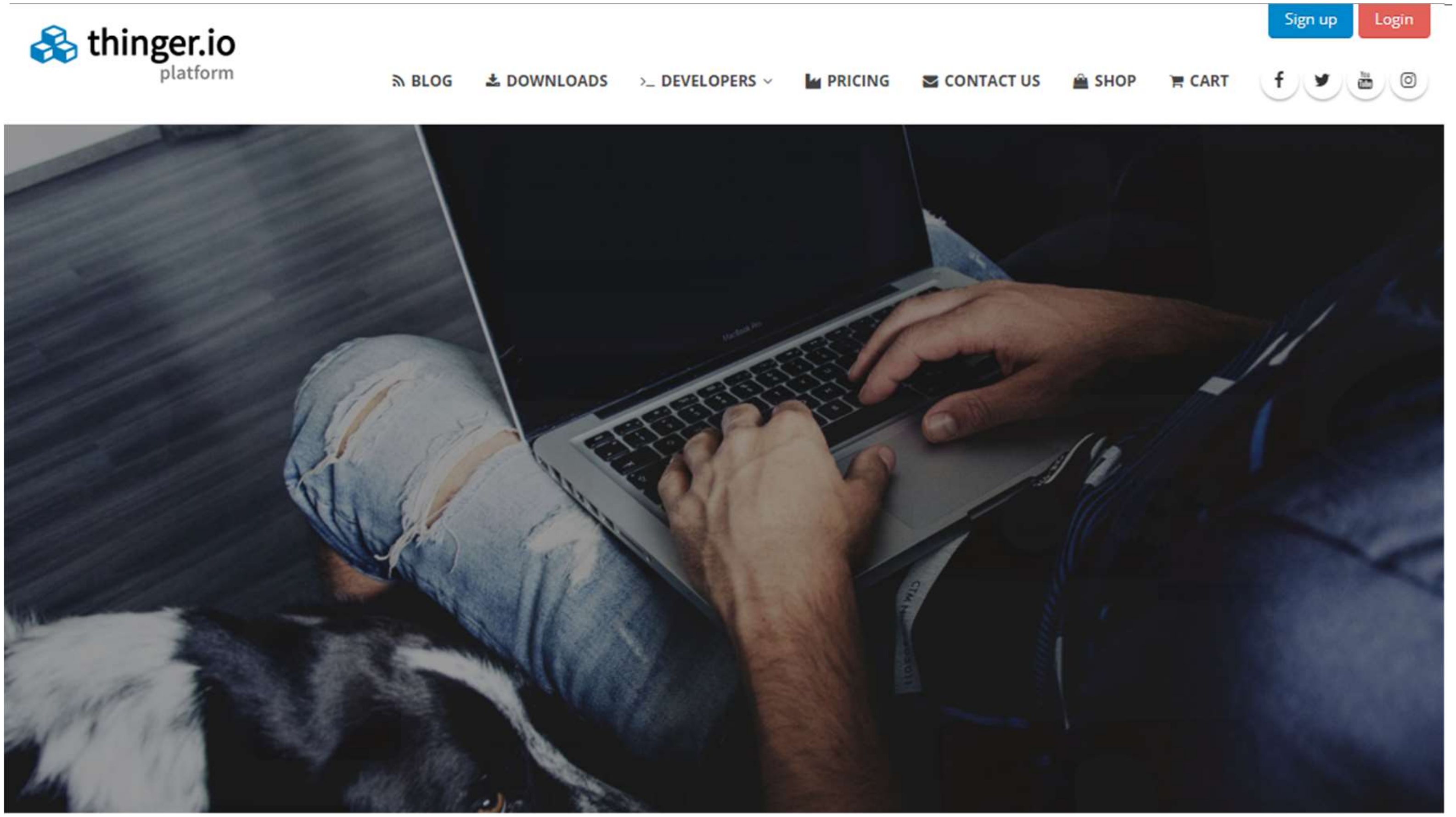
- Platform
- Data Acquisition
- Data Processing
- Actuation

Internet of Things (IoT)

The Internet of Things (IoT) is the network of physical devices, vehicles, home appliances and other items embedded with electronics, software, sensors, actuators, and connectivity which enables these objects to connect and exchange data.



Internet of Things (IoT)



https://thinger.io/

Internet of Things (IoT)

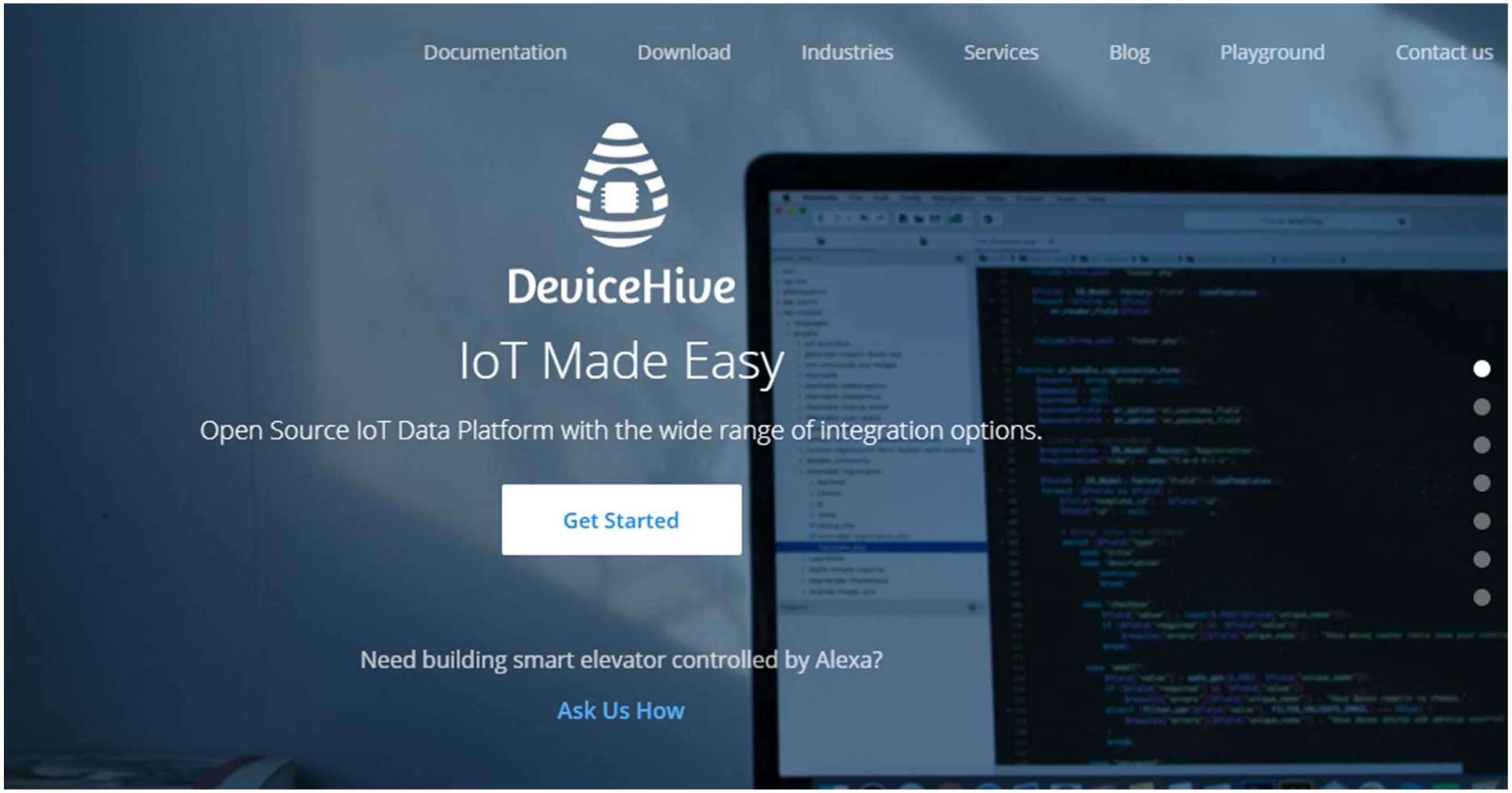
Understand Your Things

The open IoT platform with MATLAB analytics.

Collect Analyze Act

https://thingspeak.com/

Internet of Things (IoT)



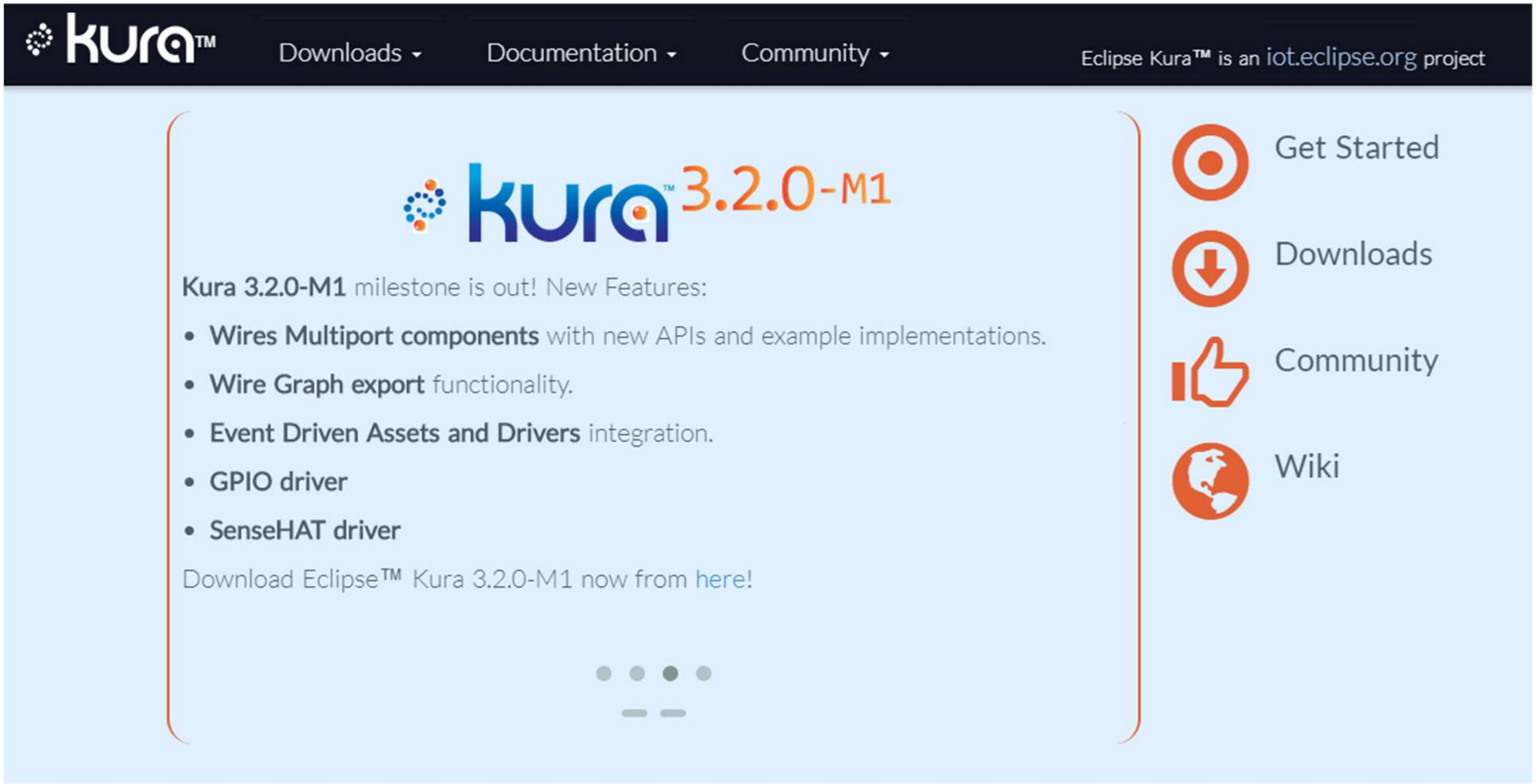
IoT Made Easy

Open Source IoT Data Platform with the wide range of integration options.

Need building smart elevator controlled by Alexa?

https://www.devicehive.com/

Internet of Things (IoT)



Kura 3.2.0-M1 milestone is out! New Features:

- **Wires Multiport components** with new APIs and example implementations.
- **Wire Graph export** functionality.
- **Event Driven Assets and Drivers** integration.
- **GPIO driver**
- **SenseHAT driver**

Download Eclipse™ Kura 3.2.0-M1 now from here!

https://www.eclipse.org/kura/

Internet of Things (IoT)

IoT Software Platform	Device management?	Integration	Protocols for data collection	Analytics	DB
Kaa IoT Platform	Yes	Portable SDK available to integate any particular platfrom, REST API	MQTT, CoAP, XMPP, TCP, HTTP	Real time IoT Data Analytics and Visualization with Kaa, Apache Cassandra and Apache Zappelin	MongoDB, Cassandra, Hadoop, Oracle NoSQL
SiteWhere	Yes	REST API, Mule AnyPoint, and more	MQTT, AMQP, Stomp, WebSockets, and direct socket connections	Real-time analytics (Apache Spark)	MongoDB, HBase , InfluxDB
ThingSpeak	No	REST and MQTT APIs	HTTP	MATLAB Analytics	MySQL
DeviceHive	*Unknown	REST AP, MQTT APIs	REST API, WebSockets or MQTT	Real-time analytics (Apache Spark)	PostgreSQL ,SAP Hana DB
Zetta	No	REST APIs	HTTP	Using Splunk	Unknown
Distributed Services Architecture (DSA)	NO	REST APIs	HTTP	No	ETSDB – Embedded Time Series
Thingsboard.io	Yes	REST APIs	MQTT, CoAP and HTTP	Real time analytics(Apache Spark, Kafka)	Cassandra
Thinger.io	Yes	REST APIs	MQTT, CoAP and HTTP	Yes	MongodB
WSo2	Yes	REST APIs	HTTP, WSO2 ESB, MQTT	Yes, WSO2 Data Analytics Server	Oracle, PostgreSQL, MySQL, or MS SQL

IoT Communications

REST

- REST, or REpresentational State Transfer, is an architectural style for providing standards between computer systems on the web, making it easier for systems to communicate with each other.

- REST-compliant systems, often called RESTful systems, are characterized by how they are stateless and separate the concerns of client and server.

IoT Communications

MQTT

- MQTT (Message Queue Telemetry Transport) was originally developed out of IBM's pervasive computing team and their work with partners in the industrial sector.
- Over the past couple of years the protocol has been moved into the open source community, seen significant growth in popularity as mobile applications have taken off, and it is in the process of moving into the hands of a standards body.
- http://mqtt.org/

IoT Communications

AMQP

- AMQP stands for Advanced Message Queuing Protocol.
- As the name implies, it provides a wide range of features related to messaging, including reliable queuing, topic-based publish-and-subscribe messaging, flexible routing, transactions, and security.
- AMQP exchanges route messages directly—in fanout form, by topic, and also based on headers.
- https://www.amqp.org/

Intelligent Environment

Definitions

- “Intelligent environments (IE) are spaces with embedded systems and information and communication technologies creating interactive spaces that bring computation into the physical world and enhance occupants experiences”
  Juan C Augusto; Vic Callaghan

- “Intelligent environments are spaces in which computation is seamlessly used to enhance ordinary activity. One of the driving forces behind the emerging interest in highly interactive environments is to make computers not only genuine user-friendly but also essentially invisible to the user”
  A. Steventon; S. Wright

Intelligent Enviroment

Data Acquisition

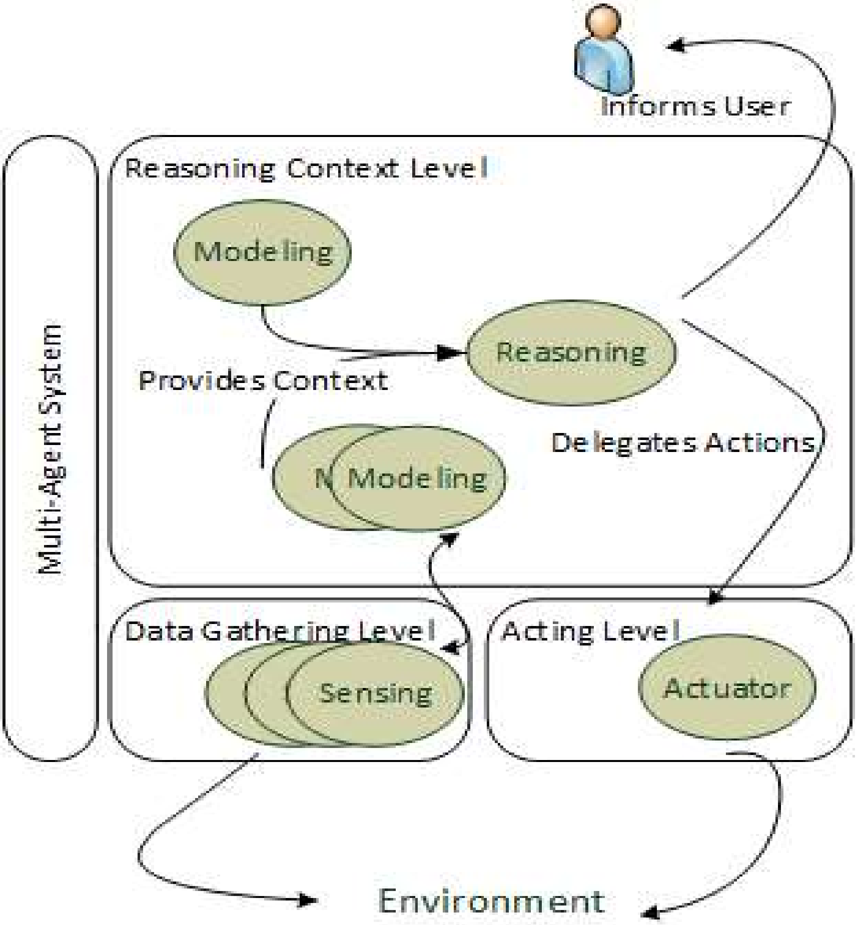
- Sensors
- Services
- Data processing

Reasoning

- Data Modeling
- Machine Learning
- Decision Models

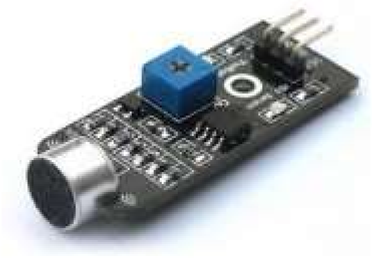
Actuation

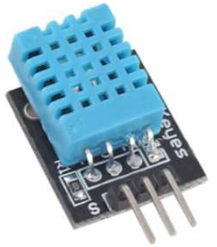
- Notifications
- Interactions
- Actions



Intelligent Enviroment

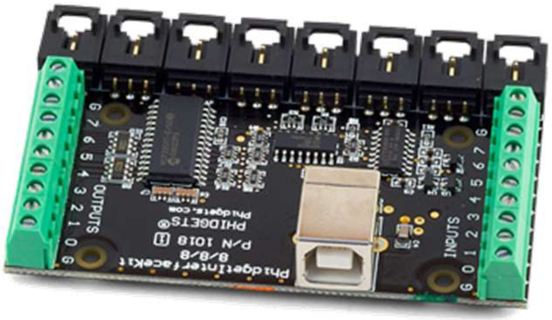
Data Aquisition





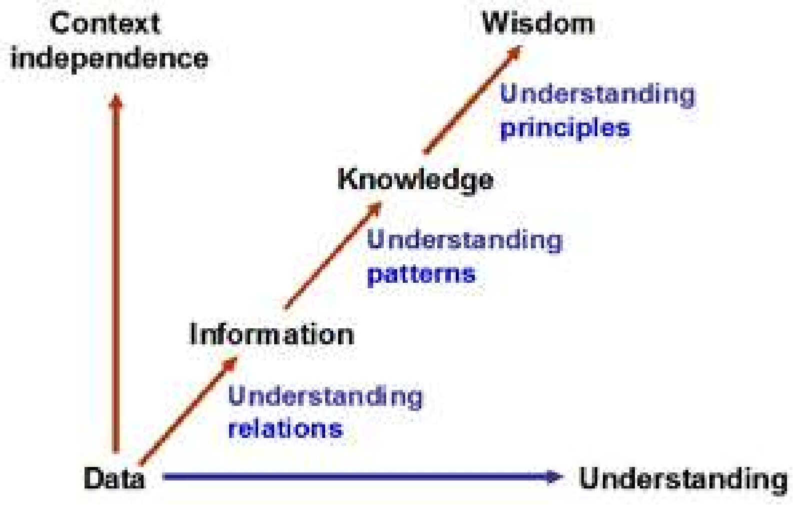
Intelligent Enviroment

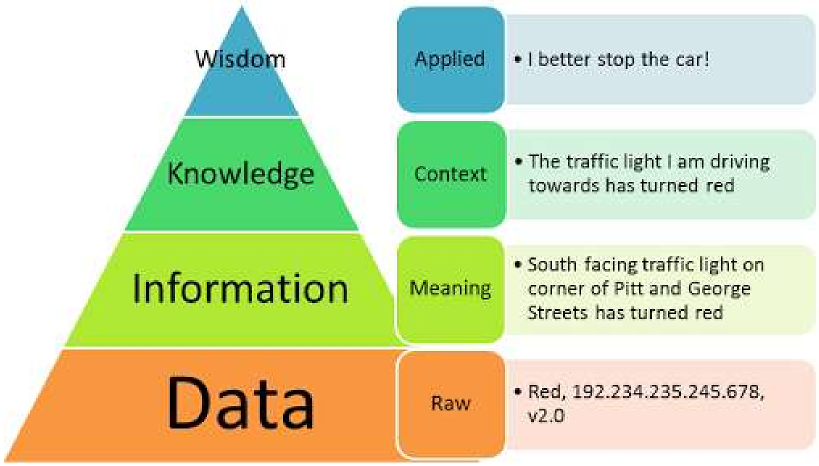
Data Aquisition



Intelligent Enviroment

Reasoning





Applied	• I better stop the car!
Context	• The traffic light I am driving towards has turned red
Meaning	• South facing traffic light on corner of Pitt and George Streets has turned red
Raw	• Red, 192.234.235.245.678, v2.0

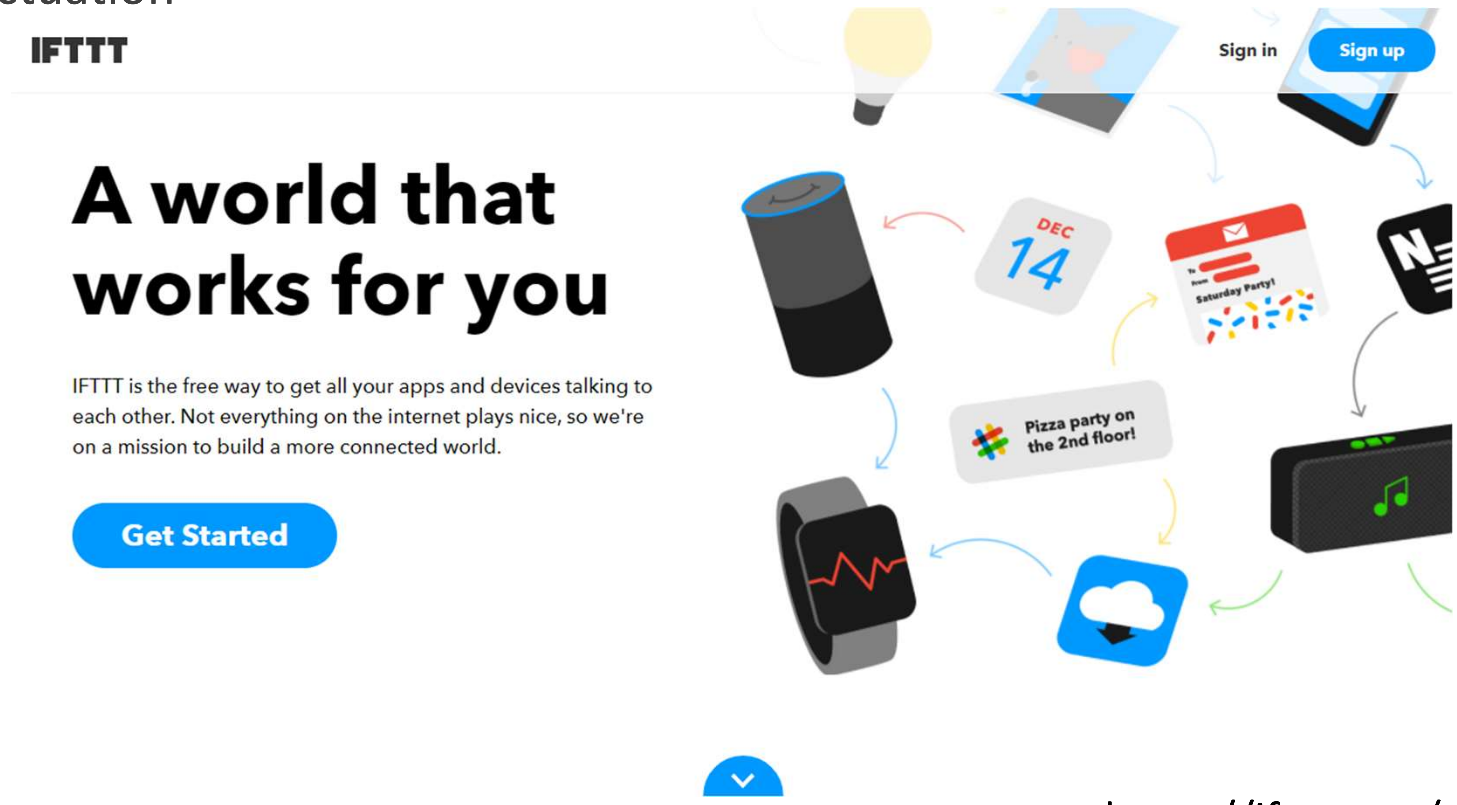
Intelligent Enviroment

Actuation

- Act upon the environment
- Act upon users
- Act upon objects

Intelligent Enviroment

Actuation



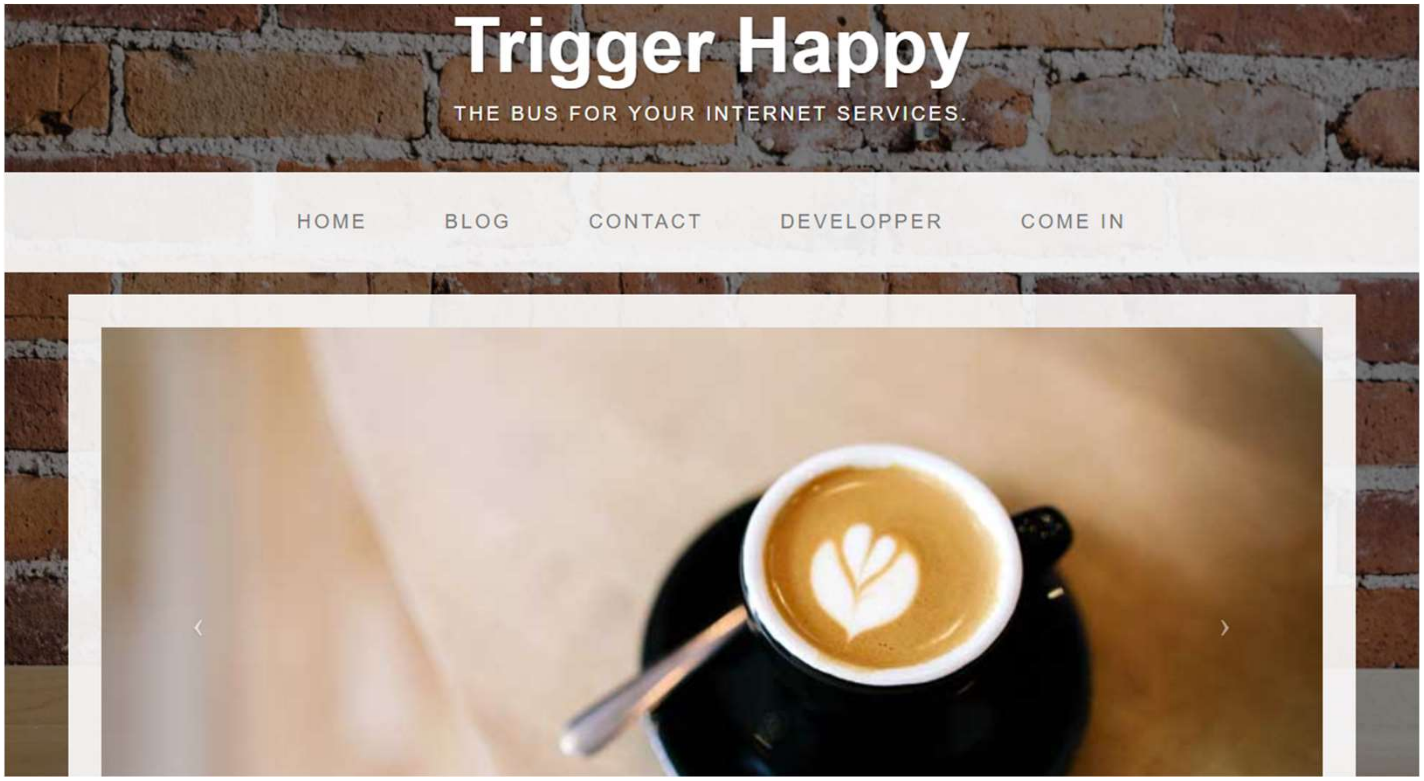
A world that works for you

IFTTT is the free way to get all your apps and devices talking to each other. Not everything on the internet plays nice, so we're on a mission to build a more connected world.

https://ifttt.com/

Intelligent Enviroment

Actuation



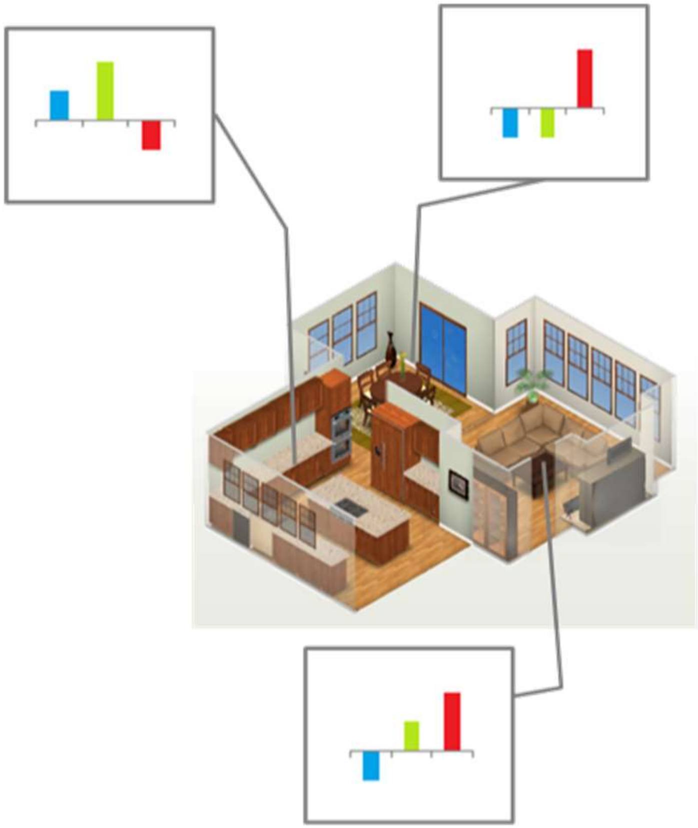
https://trigger-happy.eu/

PHESS - Platform

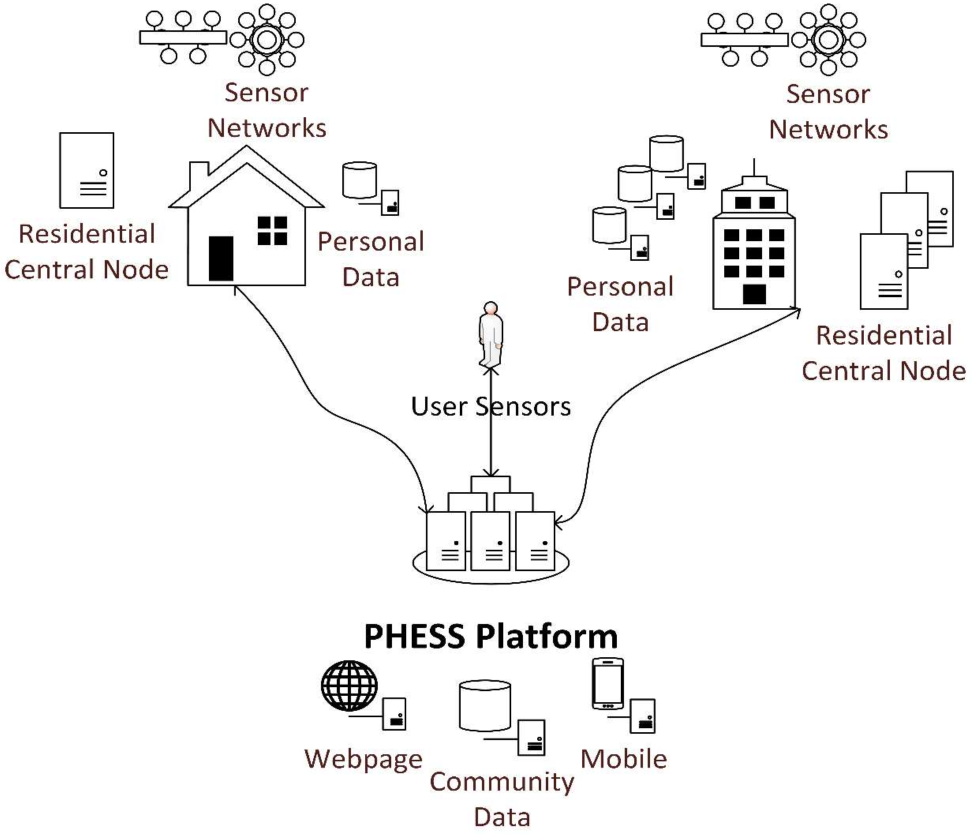
Developed in the context of Inteligent Enviroments

Uses Agent Oriented Programming based on JADE

Allows interaction with external sensors and web platforms



PHESS - Platform



PHESS - Platform

Java API sample:

```
PhessSensors s = new PhessSensors();
s.connectToPHESS("10.0.0.10","1099", "test");
              try {
System.out.println("List of Agents: ");
List<String> res = s.getTotalAgents();
for(String e : res){
             System.out.println(e);
}
System.out.println("List of Services: ");
res = s.getSensorsAgentServices("IslabSensor");
for(String e : res){
             System.out.println(e);
}
System.out.println("Sensor value: ");
Map<String,String> resM = s.getSensorValueFromAgent("ISlabSensor");
for(String e : resM.keySet()){
             System.out.println(e + " : " + resM.get(e))
}
} catch (InterruptedException | PhessTimeoutException e) {
             e.printStackTrace();
                          }
 finally {
              try {
                          s.disconnect();
             } catch (StaleProxyException | InterruptedException | PhessTimeoutException e) {
                          e.printStackTrace();
             }
 }
```

PHESS - Platform

Java Launch Sensor

in the platform

Inside Launcher in PhessSensors

```
public static void main(String args[]) {
    Launcher mc = new Launcher();
    mc.initRemoteContainer( “host", "1099");
    mc.startAgentInPlatform("SensorName", "agents.Sensor ");
    }
```

PHESS - Platform

Important Methods from the SensorAgentTemplate

```
//required methods to implement
public abstract void setup();

//optional methods
public void handleTimedSensorEvent(){;}
public boolean handleMessage(){return false;}
public void handleDfNotification(DFAgentDescription[]
notification) {;};
```

PHESS - Platform

Mapping sensors

```
private void setupSystem() {
getContentManager().registerLanguage(codec);
getContentManager().registerOntology(ontology);

try{
this.sensors = new HashMap<String, String>();
this.sensors.put("IslabSound", String.valueOf(0));
this.sensorDefinition = new HashMap<String,Sensor>();

}catch(Exception e){
e.printStackTrace();
}
}
```

PHESS - Platform

Definition of a sensor

```
public void registerSensorsInDatabase(){
Sensor sound = new DefaultSensor();
sound.setSensorId(1);
sound.setSensorName("IslabSound");
sound.setSensorRefreshRate(Integer.valueOf("10000"));
sound.setSensorType("SOUND");
sensorDefinition.put("IslabSound", sound);
}
```

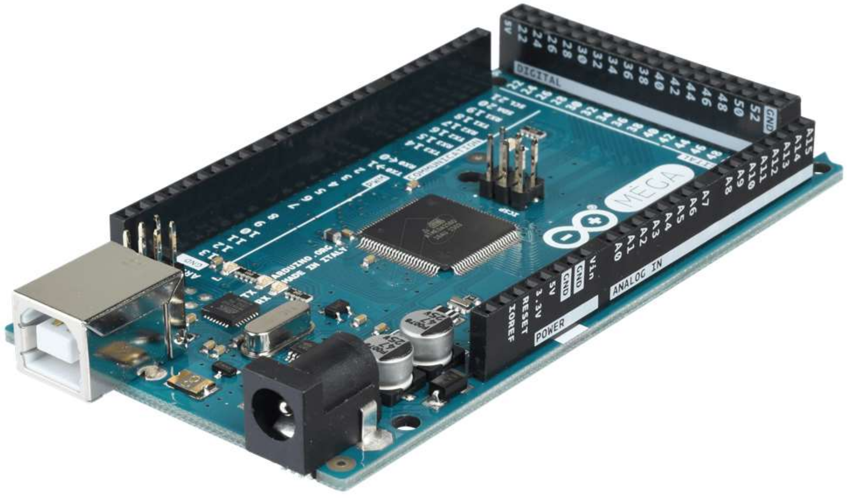
PHESS - Platform

Assignment:

- Register sample sound sensor with PHESS demo platform

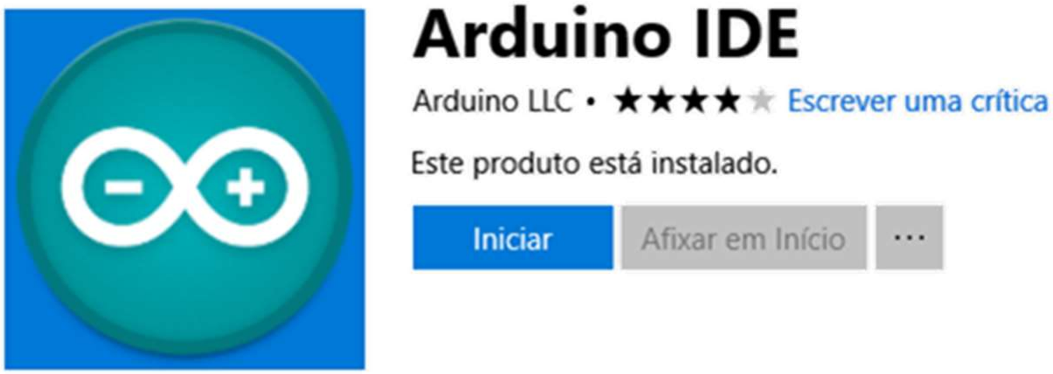
PHESS – Data Acquisition

Arduino Mega



PHESS – Data Acquisition

Arduino Mega



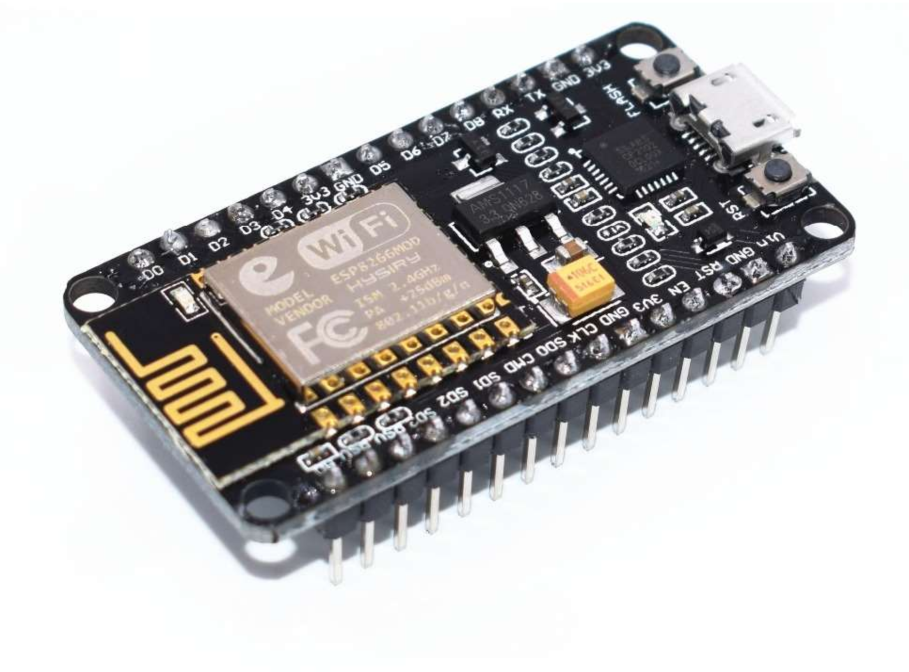
Arduino IDE

Arduino LLC • ★★★★☆ Escrever uma crítica

Este produto está instalado.

PHESS – Data Acquisition

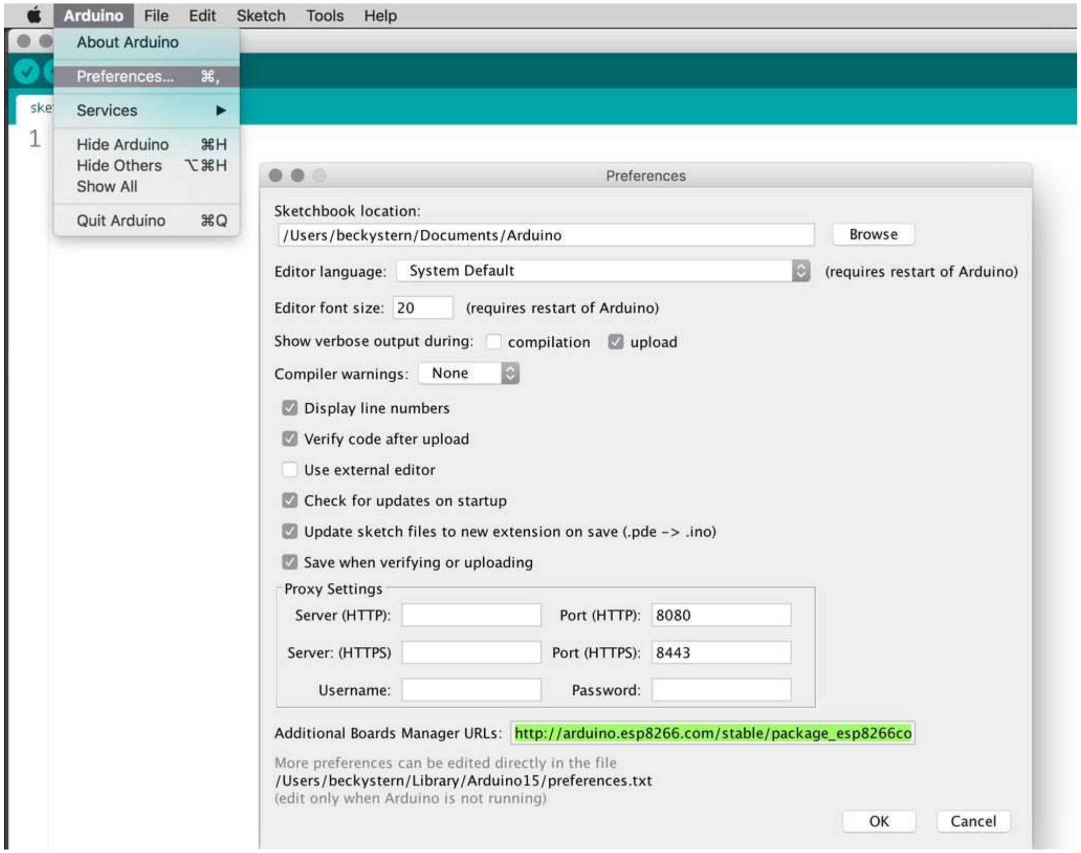
ESP8266



PHESS – Data Acquisition

ESP8266

- http://arduino.esp8266.com/stable/package_esp8266com_index.json



Sketchbook location: /Users/beckystern/Documents/Arduino

Editor language: System Default (requires restart of Arduino)

Editor font size: 20 (requires restart of Arduino)

Show verbose output during: □ compilation ☑ upload

Compiler warnings: None

☑ Display line numbers

☑ Verify code after upload

□ Use external editor

☑ Check for updates on startup

☑ Update sketch files to new extension on save (.pde -> .ino)

☑ Save when verifying or uploading

Proxy Settings

Server (HTTP): Port (HTTP): 8080

Server: (HTTPS) Port (HTTPS): 8443

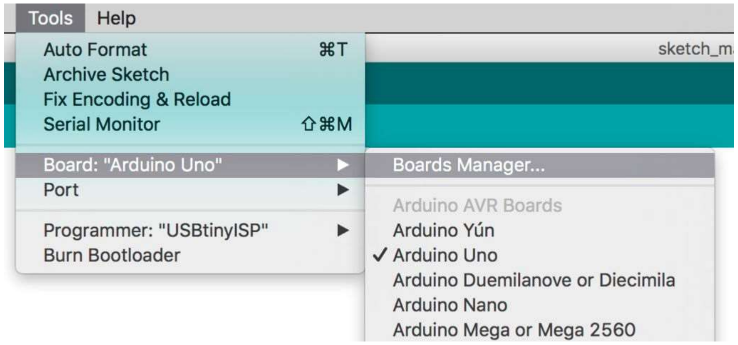
Username: Password:

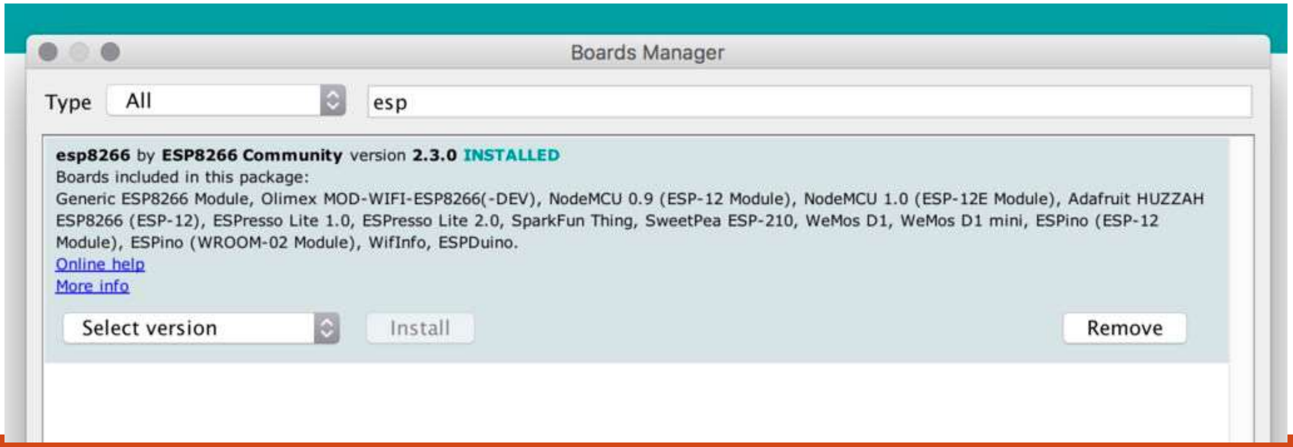
Additional Boards Manager URLs: http://arduino.esp8266.com/stable/package_esp8266co

More preferences can be edited directly in the file
/Users/beckystern/Library/Arduino15/preferences.txt
(edit only when Arduino is not running)

PHESS – Data Acquisition

ESP8266



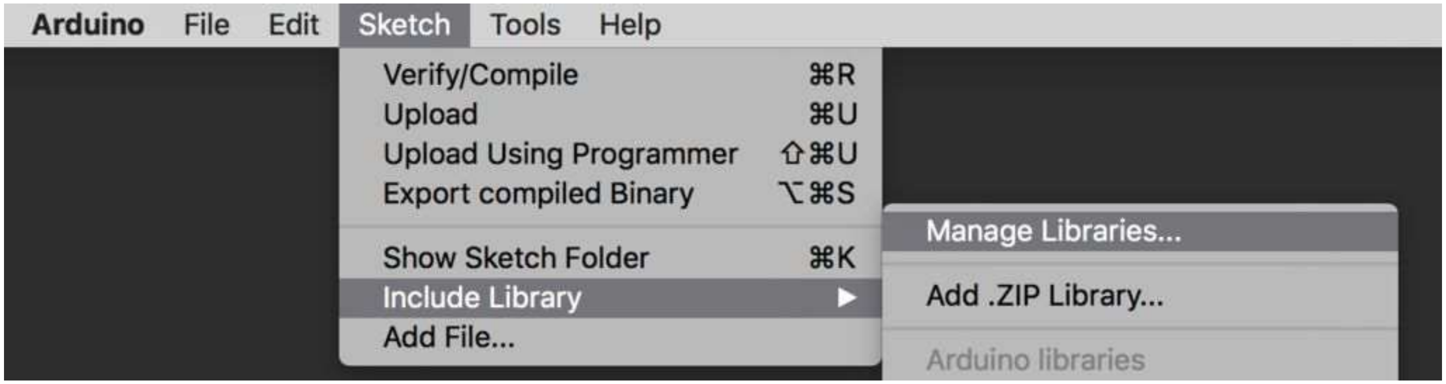


esp8266 by **ESP8266 Community** version **2.3.0** **INSTALLED**
Boards included in this package:
Generic ESP8266 Module, Olimex MOD-WIFI-ESP8266(-DEV), NodeMCU 0.9 (ESP-12 Module), NodeMCU 1.0 (ESP-12E Module), Adafruit HUZZAH ESP8266 (ESP-12), ESPresso Lite 1.0, ESPresso Lite 2.0, SparkFun Thing, SweetPea ESP-210, WeMos D1, WeMos D1 mini, ESPino (ESP-12 Module), ESPino (WROOM-02 Module), WifInfo, ESPDuino.
Online help
More info

PHESS – Data Acquisition

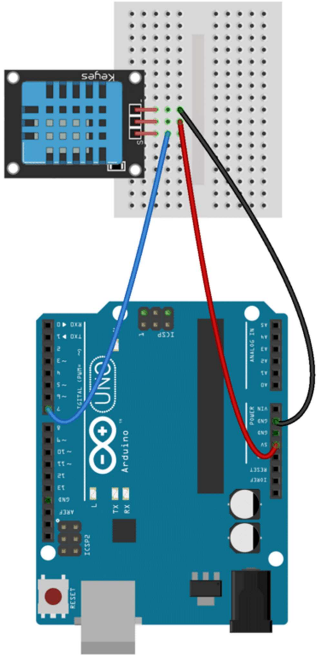
Arduino Mega and ESP8266

- Libraries:
  - ArduinoHttpClient
  - Adafruit IO Arduino
  - Adafruit MQTT



PHESS – Data Acquisition

- Arduino and ESP8266
  - DTH11
  - http://www.circuitbasics.com/how-to-set-up-the-dht11-humidity-sensor-on-an-arduino/



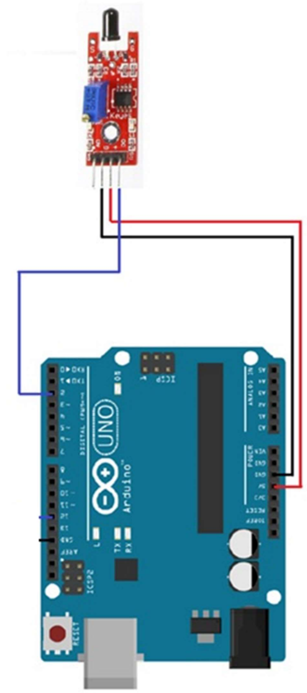
```
#include <dht.h>
dht DHT;
define DHT11_PIN 7

void setup(){
  Serial.begin(9600);
}

void loop()
{
  int chk = DHT.read11(DHT11_PIN);
  Serial.print("Temperature = ");
  Serial.println(DHT.temperature);
  Serial.print("Humidity = ");
  Serial.println(DHT.humidity);
  delay(1000);
}
```

PHESS – Data Acquisition

- Arduino and ESP8266
  - Flame Sensor
  - http://www.theorycircuit.com/arduino-flame-sensor-interface/



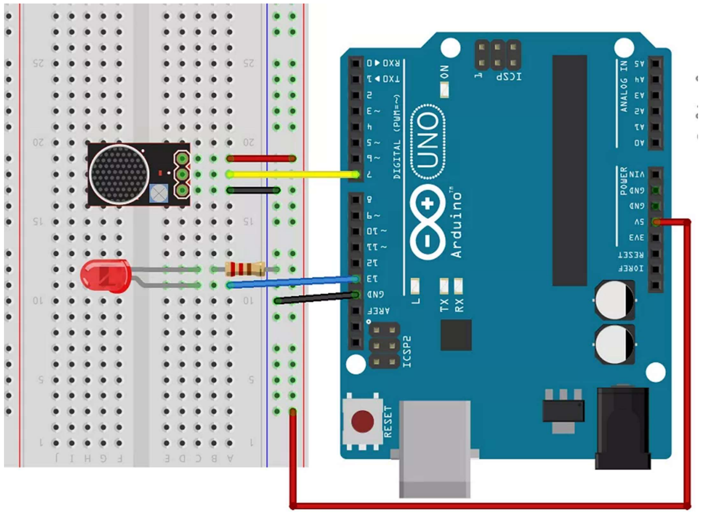
```
int FlamePin = 2;  // This is for input pin
int Flame = HIGH;  // HIGH when FLAME Exposed

void setup() {
  pinMode(Buzzer, OUTPUT);
  pinMode(FlamePin, INPUT);
  Serial.begin(9600);
}

void loop() {
  Flame = digitalRead(FlamePin);
  if (Flame== HIGH)  {
    Serial.println("HIGH FLAME");
    digitalWrite(Buzzer, HIGH);
  }
  else  {
    Serial.println("No flame");
    digitalWrite(Buzzer, LOW);
  }
}
```

PHESS – Data Acquisition

- Arduino and ESP8266
  - Sound Sensor
  - https://randomnerdtutorials.com/guide-for-microphone-sound-sensor-with-arduino/



```
int ledPin=13;
int sensorPin=7;
boolean val =0;

void setup(){
  pinMode(ledPin, OUTPUT);
  pinMode(sensorPin, INPUT);
  Serial.begin (9600);
}

void loop (){
  val =digitalRead(sensorPin);
  Serial.println (val);
if (val==HIGH) {
    digitalWrite(ledPin, HIGH);
  }   else {
    digitalWrite(ledPin, LOW);
  }
}
```

PHESS – Data Acquisition

- Arduino Communication
  - RXTX Library
    - Send data to Java

```
String inputString = "";         // a String to hold incoming data
boolean stringComplete = false;

void setup() {
  // initialize serial:
  Serial.begin(9600);
  // reserve 200 bytes for the inputString:
  inputString.reserve(200);
}

void loop() {
  // print the string when a newline arrives:
  if (stringComplete) {
    Serial.println(inputString);
    // clear the string:
    inputString = "";
    stringComplete = false;
  }
}
```

PHESS – Data Acquisition

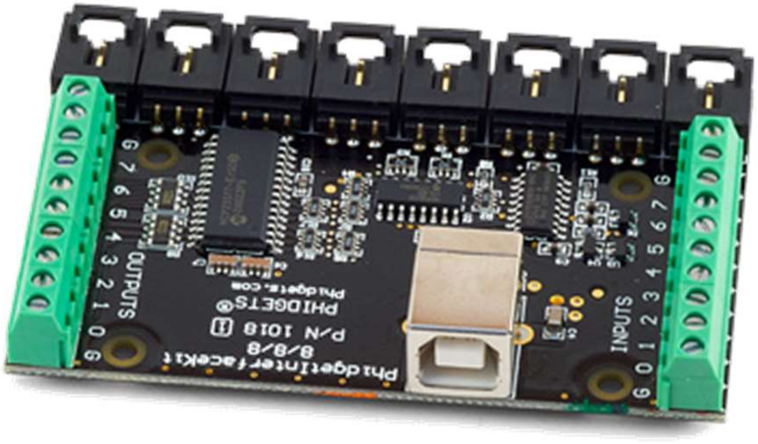
- Arduino Communication
  - RXTX Library
    - Send data to Java

```
/*   SerialEvent occurs whenever a new data comes in the
hardware serial RX. This   routine is run between each time
loop() runs, so using delay inside loop can   delay response.
Multiple bytes of data may be available. */

void serialEvent() {
  while (Serial.available()) {
    // get the new byte:
    char inChar = (char)Serial.read();
    // add it to the inputString:
    inputString += inChar;
    // if the incoming character is a newline, set a flag so the
main loop can
    // do something about it:
    if (inChar == '\n') {
      stringComplete = true;
    }
  }
}
```

PHESS – Data Acquisition

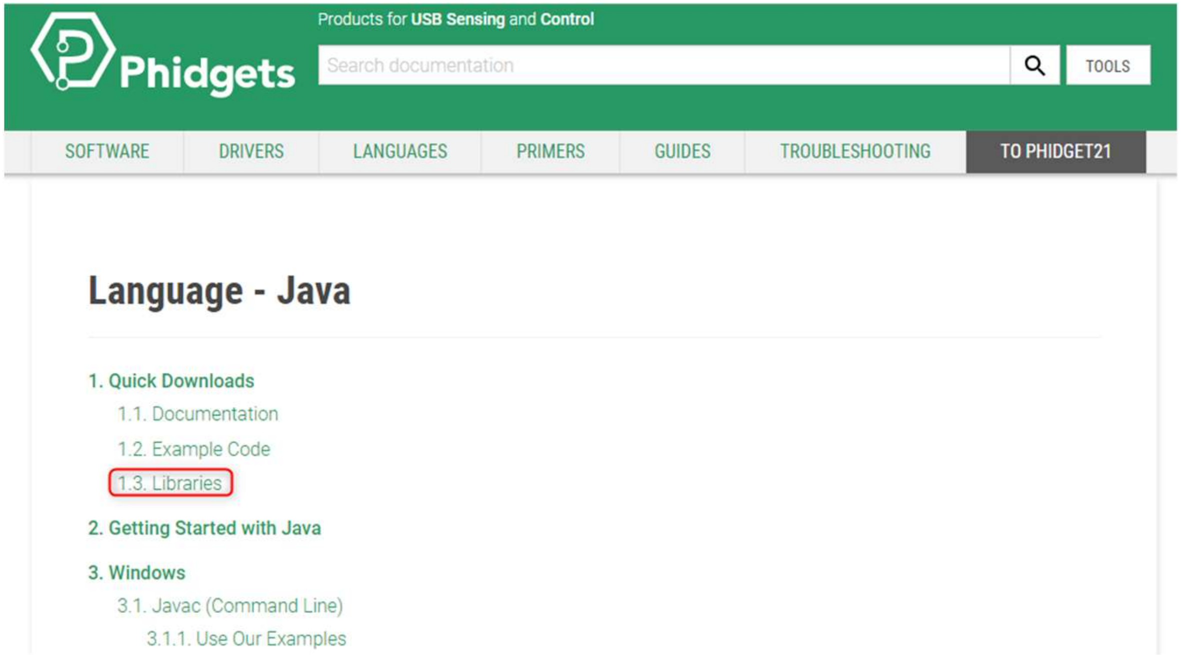
Phidgets



PHESS – Data Acquisition

Phidgets

- Install Drivers
  - https://www.phidgets.com/docs/Language_-_Java



Language - Java

1. Quick Downloads
   1.1. Documentation
   1.2. Example Code
   1.3. Libraries
2. Getting Started with Java
3. Windows
   3.1. Javac (Command Line)
      3.1.1. Use Our Examples

PHESS – Data Acquisition

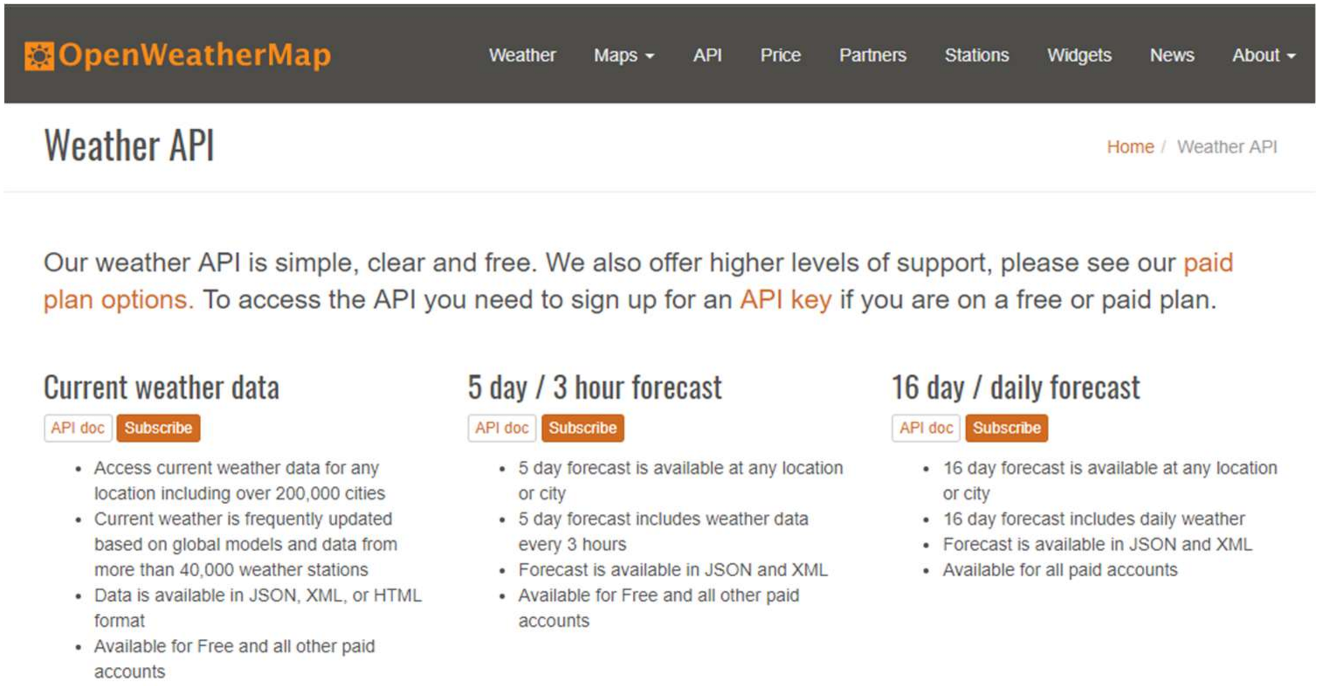
Phidgets

- Temperature
- Luminosity
- RFID Tag

PHESS – Data Acquisition

Web Services

- Use open web services to monitor condition
  - https://openweathermap.org/api



Weather API

Our weather API is simple, clear and free. We also offer higher levels of support, please see our paid plan options. To access the API you need to sign up for an API key if you are on a free or paid plan.

Current weather data

API doc Subscribe

- Access current weather data for any location including over 200,000 cities
- Current weather is frequently updated based on global models and data from more than 40,000 weather stations
- Data is available in JSON, XML, or HTML format
- Available for Free and all other paid accounts

5 day / 3 hour forecast

API doc Subscribe

- 5 day forecast is available at any location or city
- 5 day forecast includes weather data every 3 hours
- Forecast is available in JSON and XML
- Available for Free and all other paid accounts

16 day / daily forecast

API doc Subscribe

- 16 day forecast is available at any location or city
- 16 day forecast includes daily weather
- Forecast is available in JSON and XML
- Available for all paid accounts

PHESS – Data Acquisition

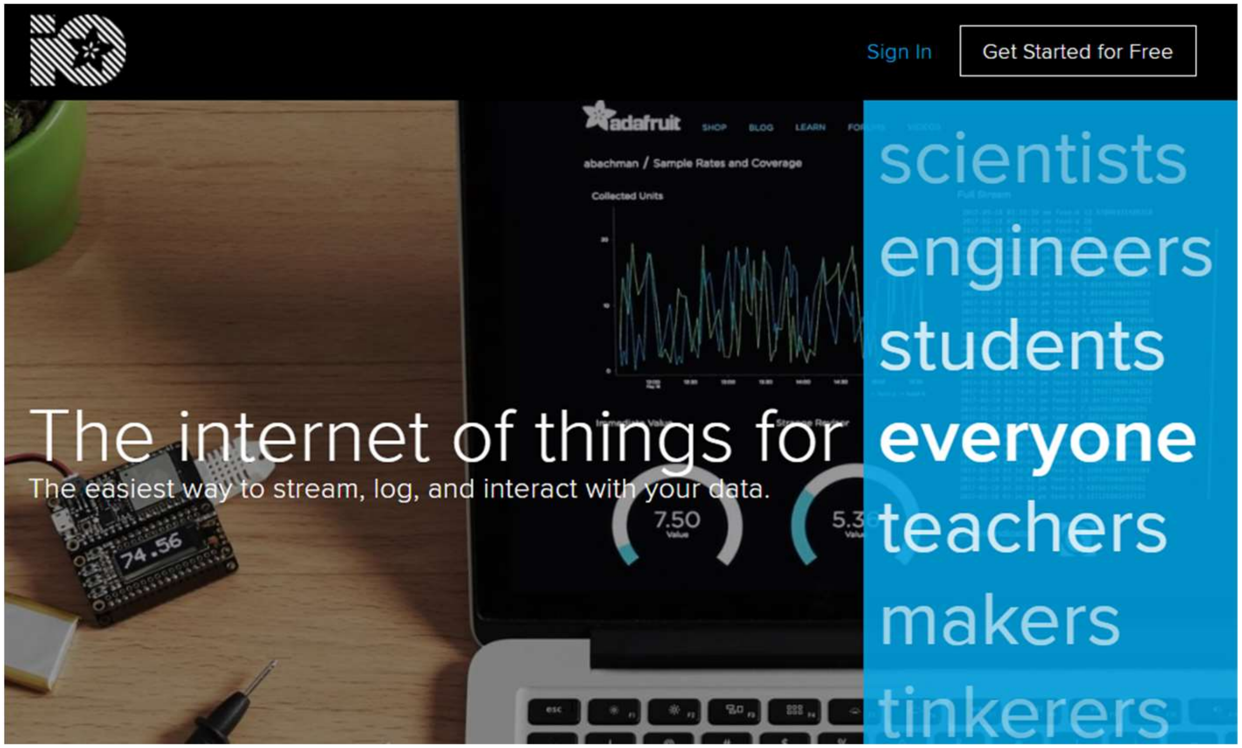
Assignment:

- Register a sensor based on hardware and a Web Service

PHESS – Data Acquisition

Register at AdafruitIO

- https://io.adafruit.com/



The internet of things for scientists engineers students **everyone** teachers makers tinkerers

The easiest way to stream, log, and interact with your data.

PHESS – Data Acquisition

Update Sensor Agents to publish sensor data with the MQTT protocol in the AdafruitIO platform

```
import org.eclipse.paho.client.mqttv3.MqttClient;
import org.eclipse.paho.client.mqttv3.MqttConnectOptions;
import org.eclipse.paho.client.mqttv3.MqttException;
import org.eclipse.paho.client.mqttv3.MqttMessage;
import org.eclipse.paho.client.mqttv3.persist.MemoryPersistence;

public class MQTT_Test {

public static void main(String[] args) {
    String topic        = “topic";
    String content      = "Message from MqttPublishSample";
    int qos             = 2;
    String broker       = "tcp://io.adafruit.com:1883";
    String clientId     = "JavaSample";
    MemoryPersistence persistence = new MemoryPersistence();
                                                        (…)
```

PHESS – Data Acquisition

Update Sensor Agents to publish sensor data with the MQTT protocol in the AdafruitIO platform

```
                                        (…)

try {
        MqttClient sampleClient = new MqttClient(broker, clientId, persistence);
        MqttConnectOptions connOpts = new MqttConnectOptions();
        connOpts.setCleanSession(true);
        connOpts.setUserName(“username");
        connOpts.setPassword(“key".toCharArray());
        connOpts.setSSLProperties(new Properties());
        System.out.println("Connecting to broker: "+broker);
        sampleClient.connect(connOpts);
        System.out.println("Connected");
        System.out.println("Publishing message: "+content);
        MqttMessage message = new MqttMessage(content.getBytes());
        message.setQos(qos);
        sampleClient.publish(topic, message);
        System.out.println("Message published");
        sampleClient.disconnect();
        System.out.println("Disconnected");
        System.exit(0);
                                        (…)
```

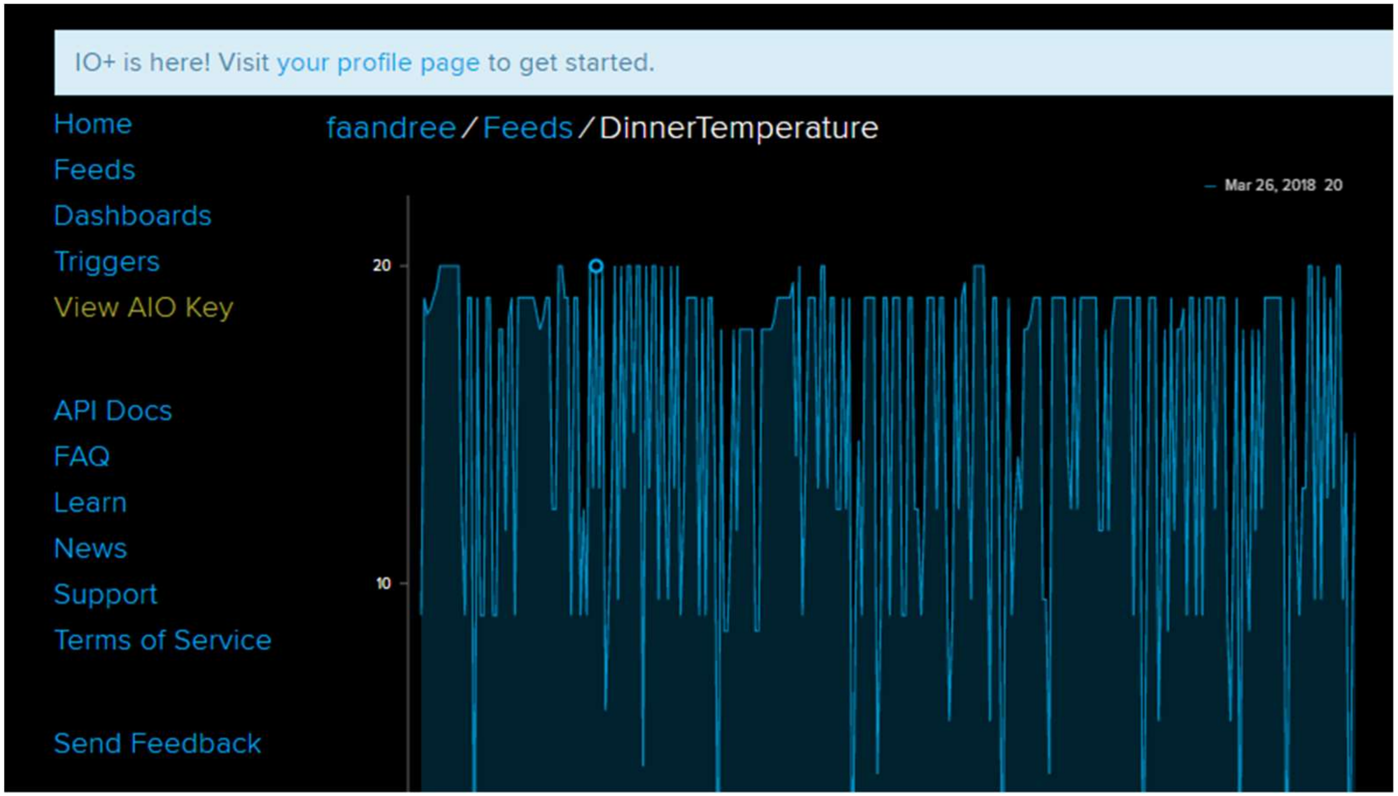
PHESS – Data Acquisition

Update Sensor Agents to publish sensor data with the MQTT protocol in the AdafruitIO platform

```
                                        (…)
} catch(MqttException me) {
        System.out.println("reason "+me.getReasonCode());
        System.out.println("msg "+me.getMessage());
        System.out.println("loc "+me.getLocalizedMessage());
        System.out.println("cause "+me.getCause());
        System.out.println("excep "+me);
        me.printStackTrace();
    }
  }
```

PHESS – Data Acquisition

Analyse Feeds and Dashboards in the AdafruitIO platform



IO+ is here! Visit your profile page to get started.

faandree / Feeds / DinnerTemperature

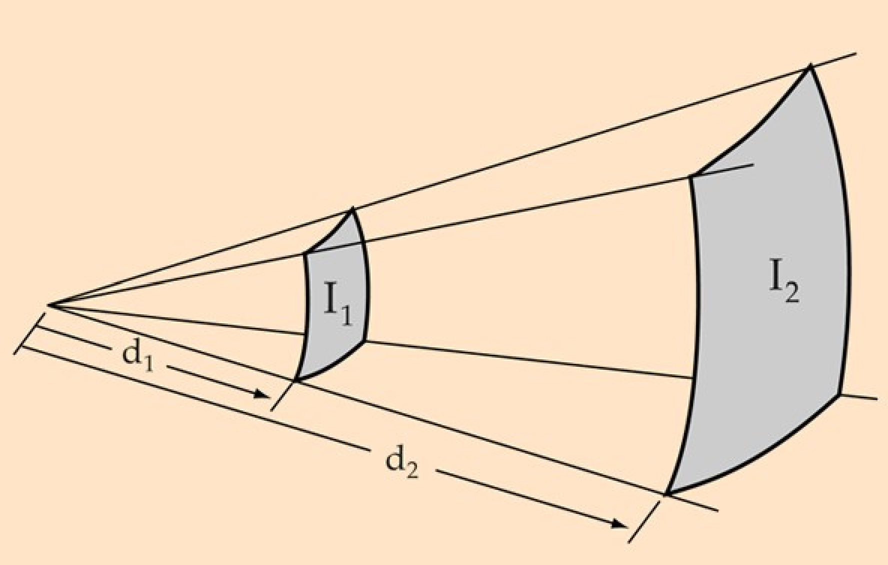
PHESS – Data Processing

Sound Analysis

I – Intensity in db

d – distance in meters

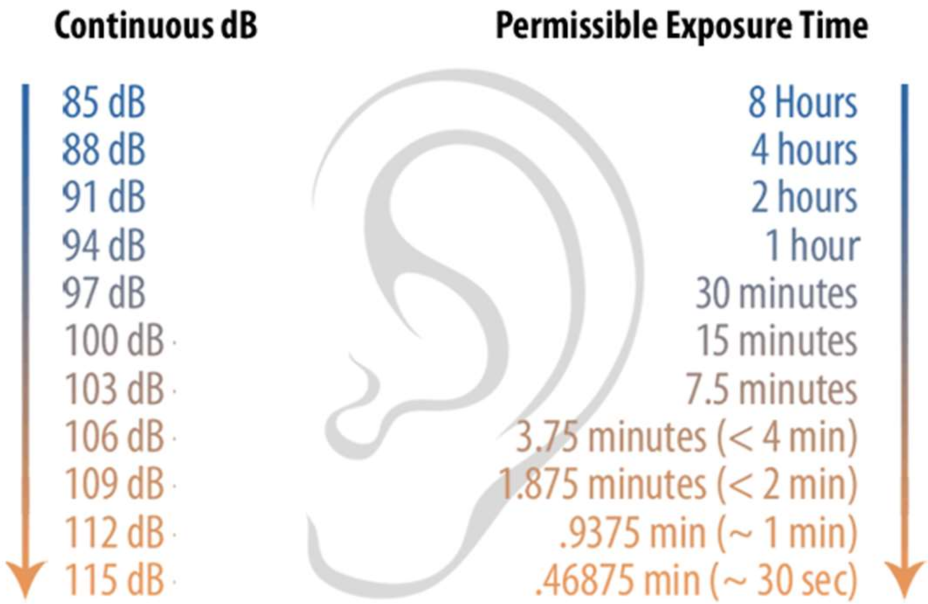
$$\frac{I_2}{I_1} = \left[\frac{d_1}{d_2}\right]^2$$



Fonte: http://hyperphysics.phy-astr.gsu.edu/hbase/acoustic/isprob2.html

PHESS – Data Processing

Sound Analysis



Continuous dB	Permissible Exposure Time
85 dB	8 Hours
88 dB	4 hours
91 dB	2 hours
94 dB	1 hour
97 dB	30 minutes
100 dB	15 minutes
103 dB	7.5 minutes
106 dB	3.75 minutes (< 4 min)
109 dB	1.875 minutes (< 2 min)
112 dB	.9375 min (~ 1 min)
115 dB	.46875 min (~ 30 sec)

PHESS – Data Processing

Temperature , Humidity and Light Analysis

- Look for comfortable parameters
- Safety parameters
- Estimate presence of AC
- Estimate lights turned on

PHESS – Data Processing

Assignment:

- Register use the theory to create new metrics based on the sensor values;
- Store the metrics created in the AdafruitIO platorm through MQTT messages

PHESS – Actuation

Next Steps

- …

Automonous Systems

INTERNT OF THINGS

Fábio Silva
fabiosilva@di.uminho.pt