Objets connectés

François Roland

2 novembre 2018

Partage d'informations

http://bit.ly/iot3d

Plan

Règles de conduite

Horaire

- 8:30 – 12:30 (pause vers 10:30)
- 13:30 – 17:30 (pause vers 15:30)

Prise de parole

- Questions et commentaires bienvenus
- Parler suffisamment fort pour l'ensemble du groupe
- Ne pas monopoliser la parole
- Ne pas interrompre

Respect de chacun

- Commentaires discriminatoires non tolérés
- Ne pas perturber le cours (gsm, réseaux sociaux)

François Roland

- Ingénieur Civil en Informatique et Gestion
- Aujourd'hui
  - Chercheur Télécoms UMONS
  - Chargé de projet Fab-IoT-Lab au FabLab Mons
  - Indépendant complémentaire
- Dans le passé
  - Consultant en développement logiciel
  - Take-Eat-Easy (startup)

Mes centres d'intérêt professionnels

- Électronique
- Programmation
- Conception mécanique
- Enseignement

Et vous ?

- Prénom
- Attentes
- Compétences IoT
  - Programmation
  - Électronique
  - Télécommunications
- Projets personnels IoT

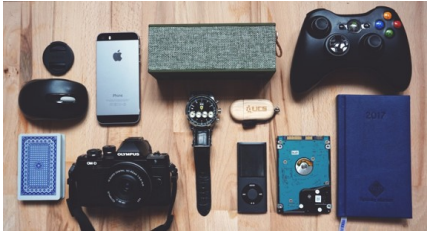
Qu'est-ce que l'Internet des Objets ?

« L'Internet des objets, ou IdO (en anglais Internet of Things, ou IoT), est l'extension d'Internet à des choses et à des lieux du monde physique.

Alors qu'Internet ne se prolonge habituellement pas au-delà du monde électronique, l'Internet des objets connectés représente les échanges d'informations et de données provenant de dispositifs du monde réel avec le réseau Internet. »

`https://fr.wikipedia.org/wiki/Internet_des_objets`

Qu'est-ce qu'un objet connecté ?



Un objet connecté est constitué de :

- Un capteur ou actionneur pour interagir avec son environnement
- Un micro-contrôleur pour transformer les données
- Un moyen de communication pour communiquer avec le monde
- Une source d'énergie pour alimenter les éléments précédents

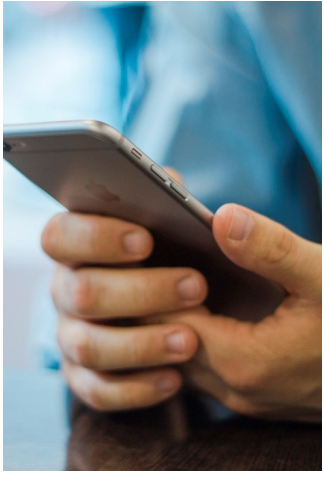
Histoire de l'IoT

L'histoire des objets connectés débute en *1999* lorsque Kevin Ashton, pionnier de la technologie *RFID* (Radio Frequency IDentification – Technologie d'identification automatique), invente l'expression « *Internet des Objets* ».
Cette même année, le concept naît aux États-Unis et particulièrement au *MIT* (Massachusetts Institute of Technology). Ce laboratoire est dédié à la création d'objets connectés à l'aide de l'identification par radiofréquence et les réseaux de capteurs sans fil.

Histoire de l'IoT

En *2003*, Rafi Haladjian, inventeur du premier opérateur Internet en France (Francenet), créé la *lampe DAL*. Une lampe d'ambiance équipée de 9 LEDs, proposant différentes couleurs et commercialisée à *790 euros*.
Deux ans plus tard, l'entreprise du créateur lance le *Nabaztag*, un lapin connecté en WiFi qui lit les mails à haute voix, émets des signaux visuels et diffuse de la musique.

Histoire de l'IoT

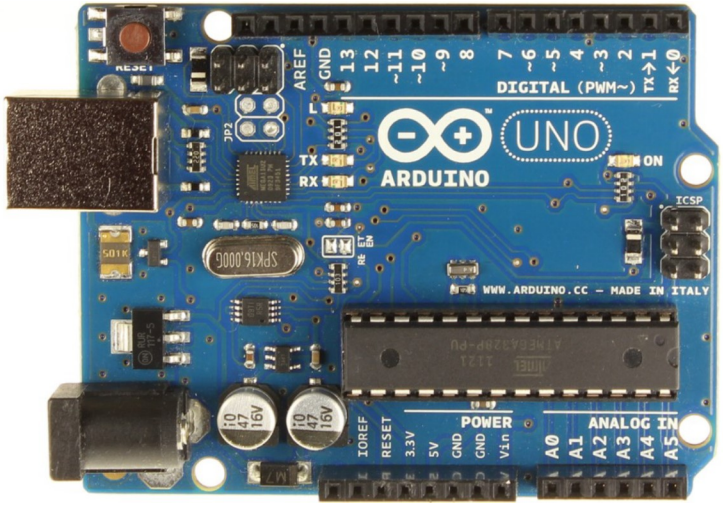


C'est néanmoins en *2007* que le phénomène de l'IoT a pris de l'ampleur, avec la démocratisation des *smartphones* et la sortie du premier iPhone par Apple. La dématérialisation est en marche.

Nous vivons actuellement une rupture forte. Celle nous permettant de connecter Internet au moindre objet de notre quotidien. Ces objets génèrent une quantité de données telle que ses perspectives d'exploitation semblent sans limite.

Le sujet des data suscite des questions éthiques délicates sur le respect de la vie privée.

Arduino


RESET
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM~)
UNO
ARDUINO
ON
TX
RX
ICSP
WWW.ARDUINO.CC – MADE IN ITALY
POWER
ANALOG IN
IOREF
RESET
3.3V
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5

Programmation C pour Arduino

```
Code == texte;
Exécution ligne par ligne;
// Commentaire
```

Variable

```
1 int pin = 13;
```

- Un type
  - `int`
  - `float`
  - `char[]`
- Un nom
- Une valeur de départ (*optionnel*)

Fonction

```
1 int myMultiplyFunction(int x, int y) {
2   return x * y;
3 }
4
5 void loop() {
6   int k = myMultiplyFunction(2, 3);
7
8   Serial.println(k); // 6
9 }
```

- Un nom
- Des paramètres (*optionnel*)
- Un type de retour
- Un code d'implémentation

Fonctions spéciales

```
1 void setup() {
2   ...
3 }
```

est exécuté une seule fois au démarrage.

```
4 void loop() {
5   ...
6 }
```

est exécuté en boucle après `setup()`.

Condition

```
1  if (x > 120) {
2    digitalWrite(LEDpin1, HIGH);
3    digitalWrite(LEDpin2, HIGH);
4  }
5
6  if (y == 150) {
7    digitalWrite(LEDpin4, LOW);
8  } else {
9    digitalWrite(LEDpin4, HIGH);
10 }
```

- Une condition à tester
- Un bout de code à exécuter si la condition est vérifiée
- Un bout de code à exécuter si la condition n'est pas vérifiée (*optionnel*)

Boucle for

```
1 for (int i=0; i <= 255; i++) {
2   analogWrite(PWMpin, i);
3   delay(10);
4 }
```

À utiliser quand on sait combien de fois il faut répéter la boucle.

- Un compteur
- Un bout de code à répéter

Boucle while

```
1 int value = 0;
2 while(value < 10) {
3   digitalWrite(LEDpin3, HIGH);
4   delay(200);
5   digitalWrite(LEDpin3, LOW);
6   value = analogRead(TRIMpin);
7 }
```

À utiliser lorsque l'on veut répéter une boucle tant qu'une condition est vérifiée

- Une condition à vérifier
- Un code à répéter

Découvrons les kits

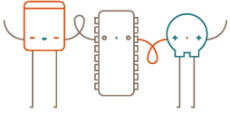
Installation de l'IDE Arduino

ARDUINO

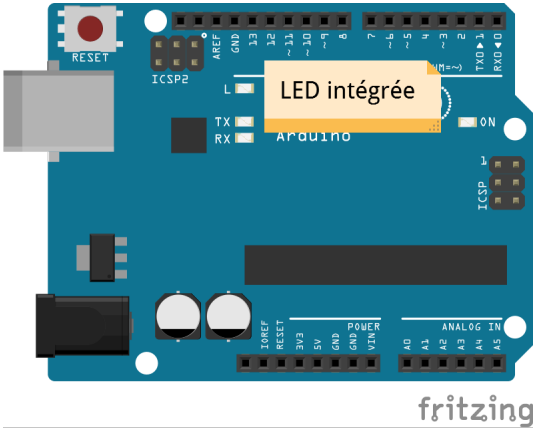
Genuino

AN OPEN PROJECT WRITTEN, DEBUGGED, AND SUPPORTED BY ARDUINO.CC AND THE ARDUINO COMMUNITY WORLDWIDE

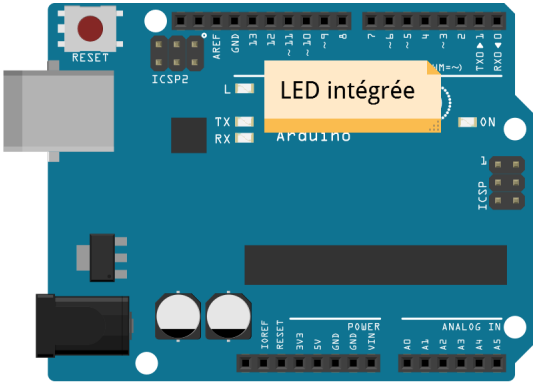
LEARN MORE ABOUT THE CONTRIBUTORS OF ARDUINO.CC on arduino.cc/credits



LED interne

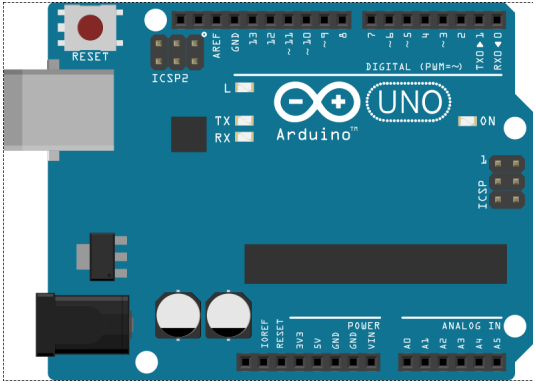

LED intégrée
RESET
ICSP2
L
TX
RX
ON
ICSP
POWER
ANALOG IN
fritzing

LED interne

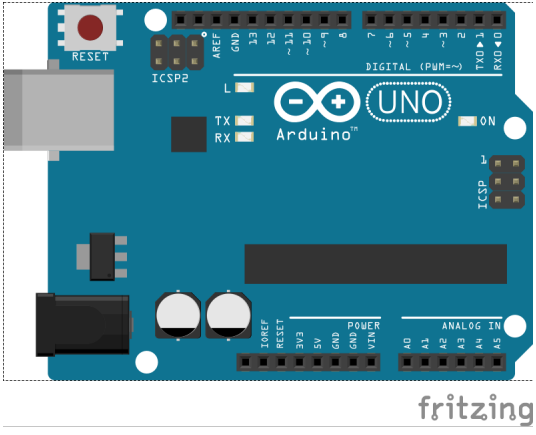



```
1  void setup() {
2    pinMode(LED_BUILTIN, OUTPUT);
3  }
4
5  void loop() {
6    digitalWrite(LED_BUILTIN, HIGH);
7    delay(500);
8    digitalWrite(LED_BUILTIN, LOW);
9    delay(500);
10 }
```

Moniteur série

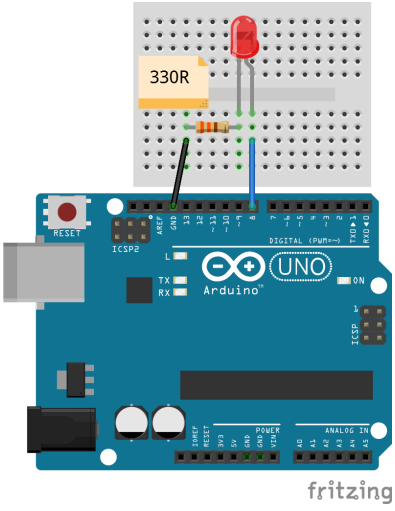


Moniteur série




```
1  int i = 0;
2
3  void setup() {
4    Serial.begin(115200);
5    Serial.println("setup DONE");
6  }
7
8  void loop() {
9    Serial.print("Loop #");
10   Serial.println(i);
11   i++;
12   delay(1000);
13 }
```

LED externe


330R
RESET
ICSP2
DIGITAL (PWM~)
UNO
Arduino
TX
RX
ON
ICSP
POWER
ANALOG IN
fritzing

Lois électriques

Loi d'Ohm

$$U = R \cdot I$$

$$I = \frac{U}{R}$$

$$R = \frac{U}{I}$$

Puissance électrique

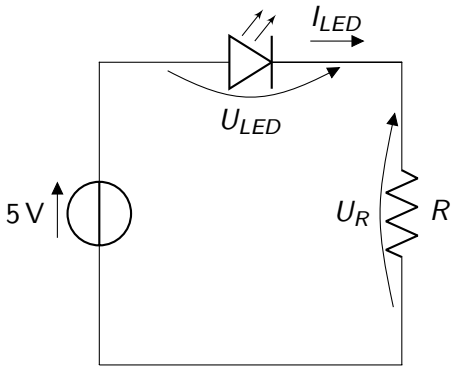
$$P = U \cdot I \quad (1)$$

$$P = R \cdot I^2 \quad (2)$$

$$P = R \cdot I^2 \quad (3)$$

où U est la tension, I le courant, R la résistance et P la puissance.

Résistance de limitation du courant



Pour une LED rouge,

$$U_{LED} = 1.7\,\text{V} \quad \text{et} \quad I_{LED} = 10\,\text{mA}$$

donc la tension aux bornes de la résistance vaut

$$U_R = 5\,\text{V} - 1.7\,\text{V} = 3.3\,\text{V}$$

et sa valeur

$$R = \frac{U_R}{I} = \frac{3.3\,\text{V}}{10\,\text{mA}} = 330\,\Omega$$

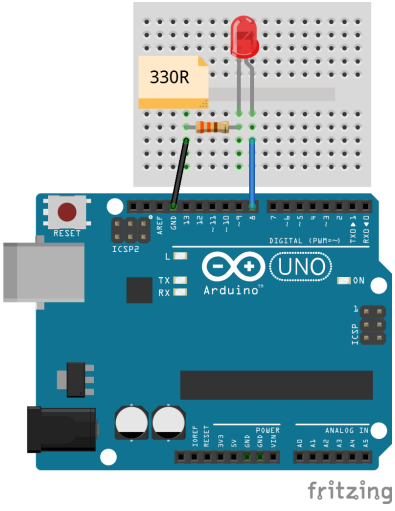
par application de la loi d'Ohm.

Valeurs standards LED

Les valeurs caractéristiques d'une LED dépendent de sa couleur.

couleur	*chute de tension*	*courant maximal*
rouge	1.7 V	10 mA
jaune	2.1 V	10 mA
verte	2.2 V	10 mA
bleue	3.2 V	20 mA
blanche	3.6 V	20 mA

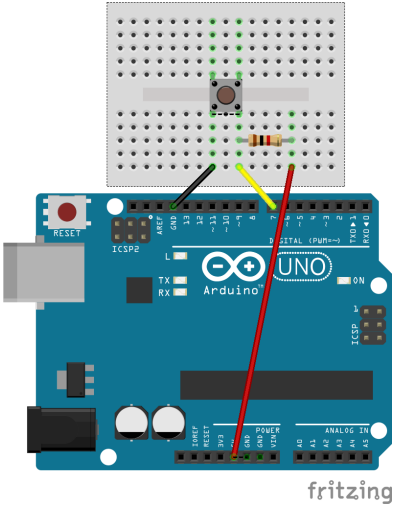
LED externe



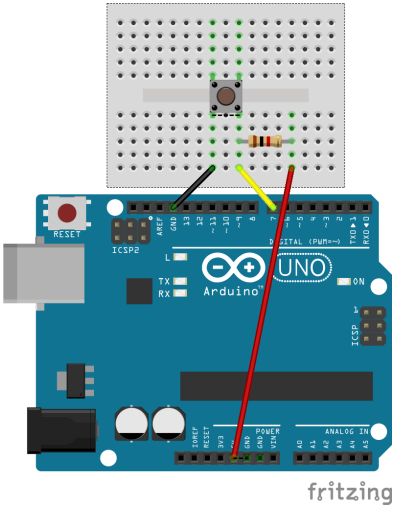

```arduino
void setup() {
  pinMode(8, OUTPUT);
}

void loop() {
  digitalWrite(8, HIGH);
  delay(500);
  digitalWrite(8, LOW);
  delay(500);
}
```

Bouton poussoir



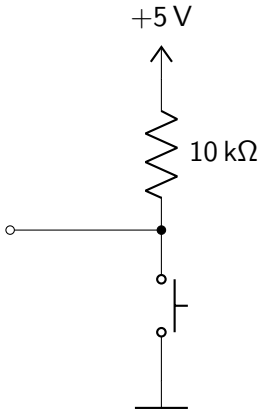
Bouton poussoir



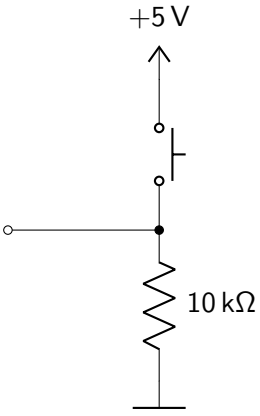

```
1  void setup() {
2    Serial.begin(115200);
3    pinMode(7, INPUT);
4  }
5
6  void loop() {
7    if (digitalRead(7) == LOW) {
8      Serial.println("button pushed");
9    }
10   delay(200);
11 }
```

Résistance de tirage

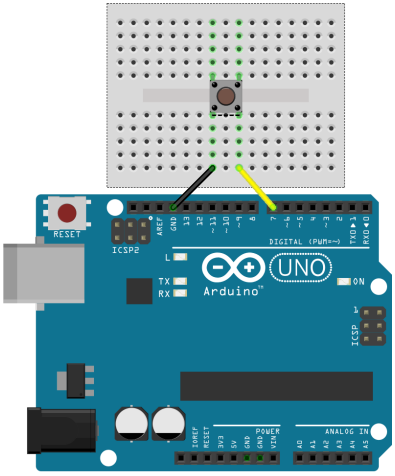
Résistance pullup



Résistance pulldown

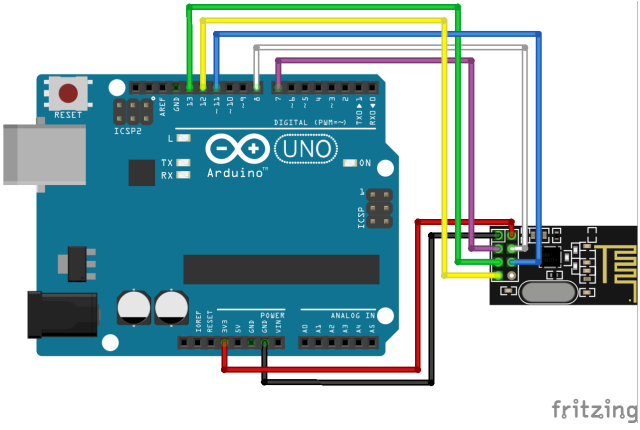


Bouton poussoir avec pullup interne




```
1  void setup() {
2    Serial.begin(115200);
3    pinMode(7, INPUT_PULLUP);
4  }
5
6  void loop() {
7    if (digitalRead(7) == LOW) {
8      Serial.println("button pushed");
9    }
10   delay(200);
11 }
```

Communication point-à-point

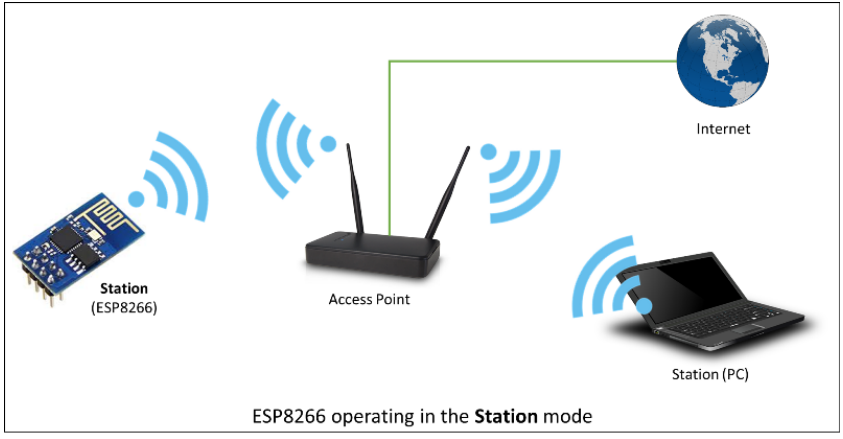


`http://tmrh20.github.io/RF24/`

3.3 V

Ce module ne supporte pas une tension de plus de 3.3 V

Communication WiFi



https://arduino-esp8266.readthedocs.io/en/latest/esp8266wifi/readme.html

WiFi

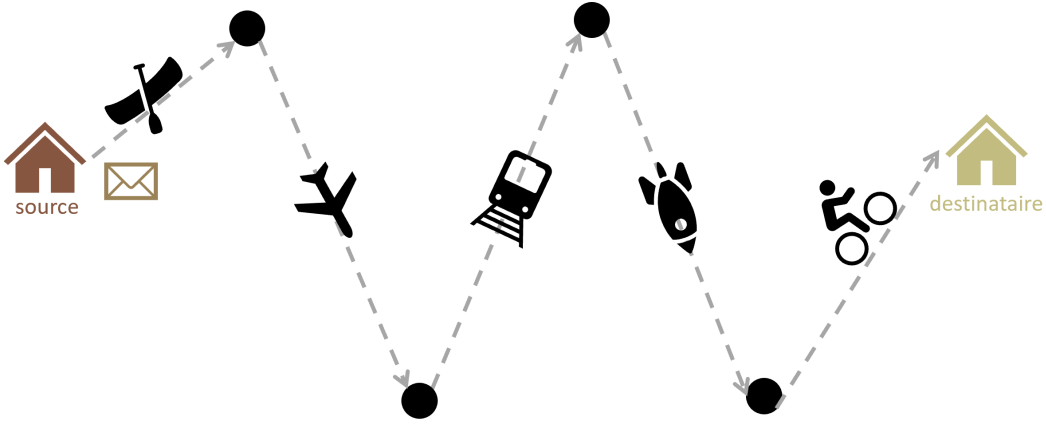
Avantages

- Vitesse de transfert ++
- Portée +
- Coût des transmetteurs −

Inconvénients

- Consommation d'énergie ++
- Facilité de configuration −

Protocole Internet (IP)



Adresse IP

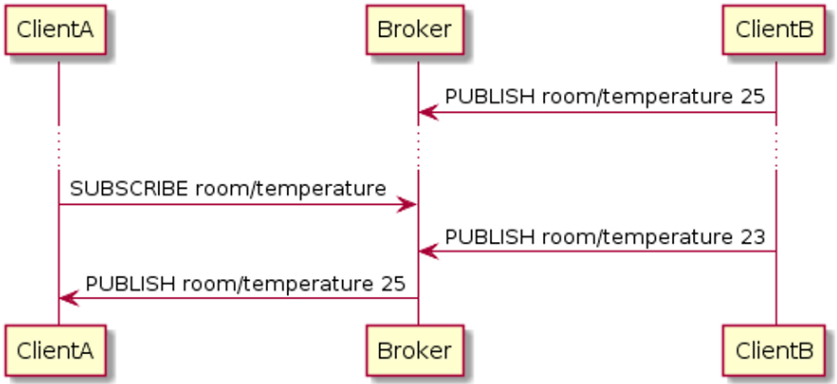
Adresse statique

- Adresse : 192.168.243.11
- Passerelle : 192.168.243.1
- Masque de sous-réseau : 255.255.255.0

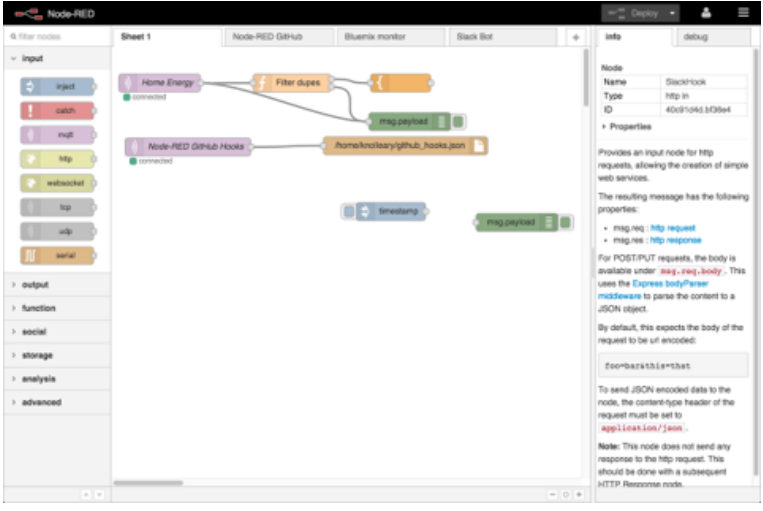
Adresse dynamique (DHCP)

- Attribution automatique
- Configuration réseau

MQTT


ClientA
Broker
ClientB
PUBLISH room/temperature 25
SUBSCRIBE room/temperature
PUBLISH room/temperature 23
PUBLISH room/temperature 25
ClientA
Broker
ClientB

Node-RED


Node-RED
Deploy
filter nodes
input
inject
catch
mqtt
http
websocket
tcp
udp
serial
output
function
social
storage
analysis
advanced
Sheet 1
Node-RED GitHub
Bluemix monitor
Slack Bot
Home Energy
connected
Filter dupes
msg.payload
Node-RED GitHub Hooks
connected
/home/knolleary/github_hooks.json
timestamp
msg.payload
info
debug
Node
Name
Slack-lock
Type
http in
ID
40c91c4d.b38e4
Properties
Provides an input node for http requests, allowing the creation of simple web services.
The resulting message has the following properties:
msg.req : http request
msg.res : http response
For POST/PUT requests, the body is available under msg.req.body. This uses the Express body-Parser middleware to parse the content to a JSON object.
By default, this expects the body of the request to be url encoded:
foo=bar&this=that
To send JSON encoded data to the node, the content-type header of the request must be set to application/json.
Note: This node does not send any response to the http request. This should be done with a subsequent HTTP Response node.

Ce que nous avons vu

- Programmation Arduino
- Capteurs et actionneurs
- Multimètre et console série
- Communication sans fil

Ce que nous n'avons pas abordé

- Sécurité
- Mise à jour
- Gestion d'énergie
- Communication sans fil basse consommation
- Packaging (chaleur, vibration)