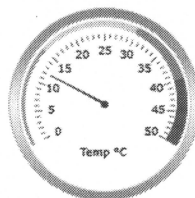


Communication entre un navigateur et une carte Arduino

1. Présentation

On souhaite établir une communication entre un navigateur, sur lequel a été chargée l'interface graphique ci-contre, et une carte Arduino.

Celle-ci contient un programme qui, à la demande, mesure la température ambiante ou commande un dispositif de chauffage dont l'état (on, off) est identifié par une LED.



12

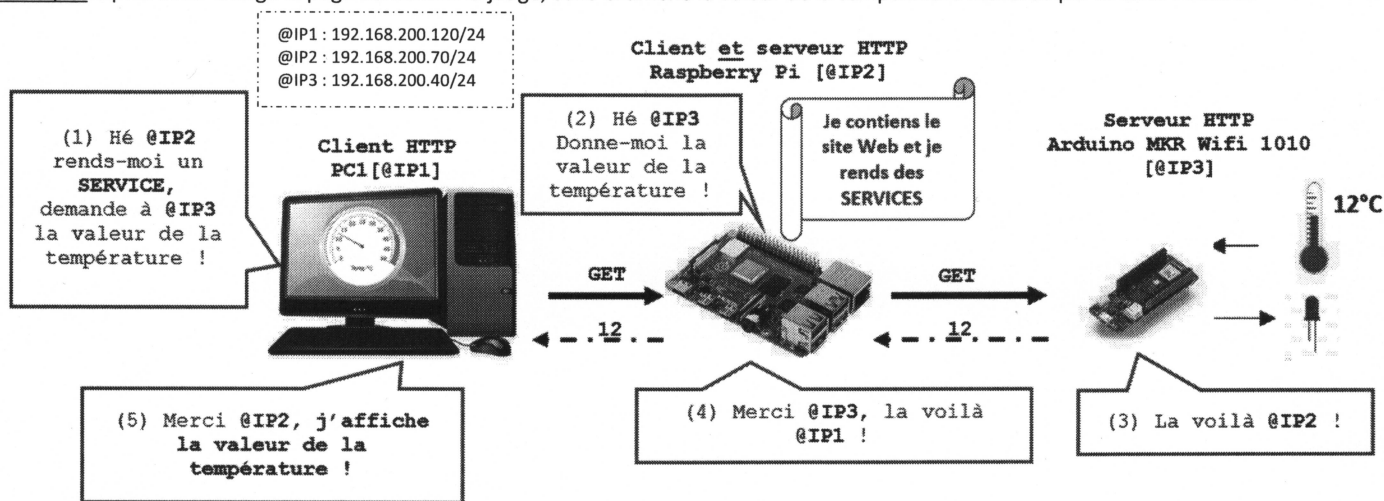
Commande de la led L et de D0 sur MKR Wifi 1010

On Off



2. Principe retenu en projet

Exemple : après avoir chargé la page contenant la jauge, celle-ci affiche la valeur de la température mesurée par la carte Arduino.



GET

C'est la **méthode** la plus courante pour **demandeur une ressource**. Une **requête** GET est sans effet sur la ressource.

3. Les méthodes spécifient le type de requête

Dans le protocole **HTTP**, une **méthode est une commande** spécifiant un type de **requête**, c'est-à-dire qu'elle demande au serveur d'effectuer une action. En général, l'action concerne une ressource identifiée par l'**URL** qui suit le nom de la méthode.

URL (Uniform Resource Locator)

protocol ://[user:password@]host[:port][[/path]][/ressource][?nom₁=val₁&nom₂=val₂...&nom_n=val_n]

La **requête HTTP** est créée à partir de l'**URL** entrée dans le navigateur ou **CONSTRUITE** avec du code (JAVASCRIPT, PHP, etc.). Elle est constituée :

1. de l'adresse du serveur (host),
2. du numéro de port [:port], codé sur **16 bits**
3. du chemin d'accès [/path] vers la ressource [/ressource] suivi d'éventuelles données passées sous la forme **nom=valeur** séparées par **&**.

Exemple 1 : URLs entrées dans le navigateur pour obtenir **DIRECTEMENT** la valeur de la température (val1) auprès de la carte Arduino :

<http://192.168.200.36/lire/val1>



GET /lire/val1 HTTP/1.1
Host: 192.168.200.40:80
En-tête de la requête
(Capturé avec **Advanced REST client**)

ou éclairer la LED.

<http://192.168.200.36/ecrire/valA?val=1>

Exemple 2 : URLs entrées dans le navigateur pour obtenir **INDIRECTEMENT** la valeur val1 auprès de la carte Arduino ou éclairer la LED.

<http://192.168.200.70/scripts/lireVal1.php>

<http://192.168.200.70/scripts/ecrireValA.php?commande=1>

4. Fonctionnalités que doit assurer le code (JAVASCRIPT + PHP) pour répondre au principe retenu

Construire les requêtes GET afin :

- d'interroger périodiquement la carte Arduino pour connaître la valeur de la température,
- d'envoyer un ordre à la carte Arduino lorsqu'on clique sur un des deux boutons.

Proposer la possibilité de

- produire des actions en réponse aux informations données par la carte Arduino,
- gérer les erreurs de communication.

Lors de la mise au point des projets, on utilise des requêtes **GET** pour tester la communication à partir du navigateur.

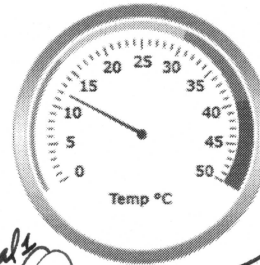
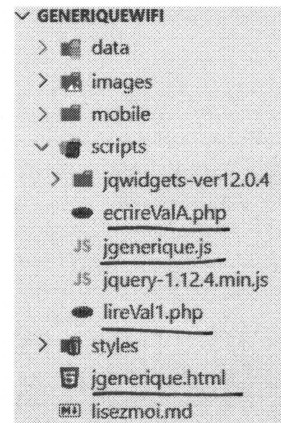
```

<!DOCTYPE html>
<html>
<head>
  <title id='Description'>Générique Wifi</title>
  <meta charset="UTF-8">
  <link rel="stylesheet" href="scripts/jqwidgets-ver12.0.4/styles/jqx.base.css" type="text/css" />
  <script type="text/javascript" src="scripts/jquery-1.12.4.min.js"></script>
  <script type="text/javascript" src="scripts/jqwidgets-ver12.0.4/jqxcore.js"></script>
  <script type="text/javascript" src="scripts/jqwidgets-ver12.0.4/jqxchart.js"></script>
  <script type="text/javascript" src="scripts/jqwidgets-ver12.0.4/jqxgauge.js"></script>
  <script type="text/javascript" src="scripts/jqwidgets-ver12.0.4/jqxbuttons.js"></script>
</head>

<body>
  <div id="jaugeCirculaire"></div>
  <div id="val1"></div>
  <p>Commande de la led L et de D0 sur MKR Wifi 1010</p>
  <form>
    <input type="button" id="On" name="commande" value="On" />
    <input type="button" id="Off" name="commande" value="Off" />
  </form>
  <script type="text/javascript" src="scripts/jgenerique.js"></script>
</body>
</html>

```

jgenerique.html



Commande de la led L et de D0 sur MKR Wifi 1010

On Off

jgenerique.js

```

$(document).ready(function () {
  $('#jaugeCirculaire').jqxGauge({
    ranges: [{ startValue: 0, endValue: 15, style: { fill: '#C9C9C9', stroke: '#C9C9C9' }, endWidth: 3, startWidth: 1 },
    { startValue: 15, endValue: 30, style: { fill: '#FCF06A', stroke: '#FCF06A' }, endWidth: 6, startWidth: 3 },
    { startValue: 30, endValue: 40, style: { fill: '#FCA76A', stroke: '#FCA76A' }, endWidth: 9, startWidth: 6 },
    { startValue: 40, endValue: 50, style: { fill: '#FC6A6A', stroke: '#FC6A6A' }, endWidth: 12, startWidth: 9 },
    ticksMinor: { interval: 1, size: '5%' },
    ticksMajor: { interval: 5, size: '9%' },
    max: 50,
    labels: { interval: 5 },
    caption: { value: "Temp °C" },
    width: 200,
    height: 200
  });
});

```

**LE JAVASCRIPT
(jQuery) s'exécute
sur le CLIENT
(NAVIGATEUR)**

Gestion des boutons

```

$('input').click(function () {
  var valeur = '0';
  if ($(this).attr('value') == 'On') {
    valeur = '1';
  }
  else if ($(this).attr('value') == 'Off') {
    valeur = '0';
  }

  $.ajax({
    url: 'scripts/ écrireValA.php',
    type: 'get',
    data: $(this).attr('name') + '=' + valeur,
    success: function () {
      // pour aller plus loin : création de messages
    },
    timeout: 3000,
    error: function () {
      // pour aller plus loin : gestion des erreurs
    }
  });
});

```

valeur ← 1

data = donnée à transmettre à écrireValA.php

ici : commande = 1

commande = 1

Commentaires

- (a) précise le fichier de destination de la donnée et construit la donnée : propriété = valeur
- (b) s'exécute lors d'un retour d'information (requête réussie)
- (c) s'exécute au bout d'un temps t = timeout si la requête n'a pas abouti.

Gestion de la demande periodique de val1

```
function loadVal1() {
  $.ajax({
    url: 'scripts/lireVal1.php',
    type: 'get',
    success: function (data) {
      $("#val1").html(data); // Pour les tests
      var val1 = parseInt(data);
      $("#jaugeCirculaire").jqxGauge({ value: val1 });
    },
    timeout: 4000,
    error: function () {
      // pour aller plus loin : gestion des erreurs
    }
  })
}

loadVal1(); // Exécuté une fois au chargement de la page

setInterval(function () { // Boucle infinie, s'exécute
  loadVal1() // toutes les 5s
}, 5000);
};
```

Info prof

1. présentation (p1)
2. fonctionnement (p4)
3. tracer les chemins
4. détail du code

**LE PHP s'exécute sur
le SERVEUR
(RASPBERRY PI)**

```
<?php
// URL à modifier en fonction de l'@IP et du programme de la carte Arduino
$service_url = "http://192.168.200.36/lire/val1";
$curl = curl_init($service_url);

// Envoi cURL à la carte Arduino
curl_setopt($curl, CURLOPT_IPRESOLVE, CURL_IPRESOLVE_V4);
curl_exec($curl);
curl_close($curl);
?>
```

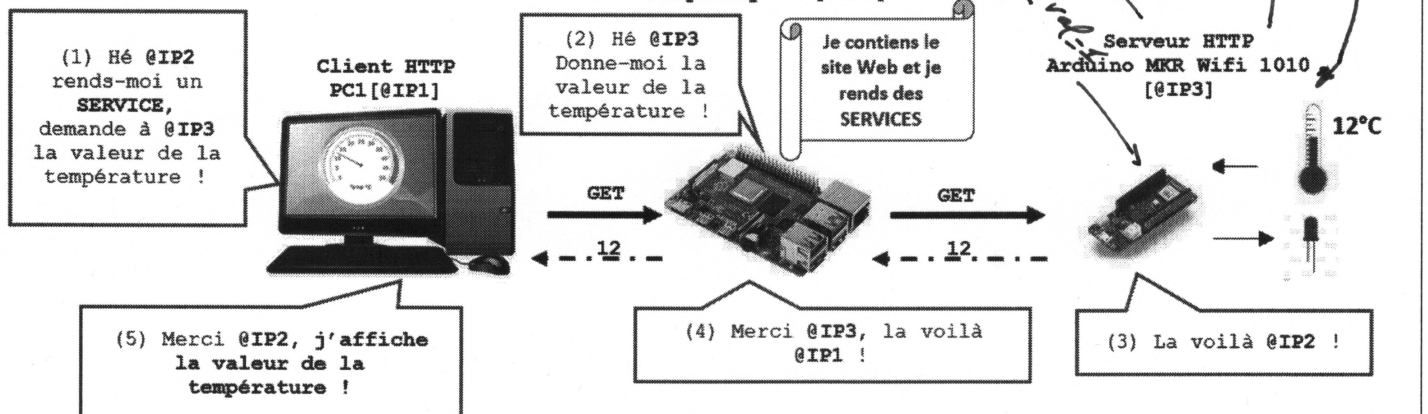
lireVal1.php

ecrireValA.php

```
<?php
// URL à modifier en fonction de l'@IP et du programme de la carte Arduino
$service_url = 'http://192.168.200.36/ecriture/vala ?val='.$_GET["commande"];
$curl = curl_init($service_url);

// Envoi cURL à la carte Arduino
curl_setopt($curl, CURLOPT_IPRESOLVE, CURL_IPRESOLVE_V4 );
$curl_response = curl_exec($curl);
curl_close($curl);
?>
```

Client et serveur HTTP
Raspberry Pi [@IP2]



Requêtes GET et leurs réponses

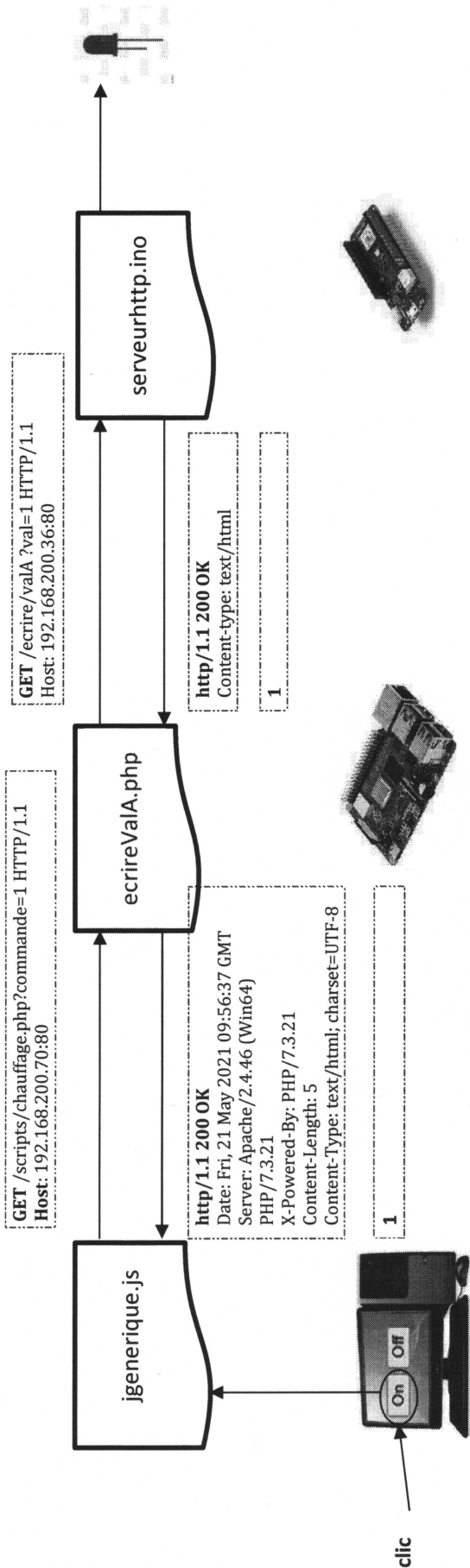
Requête

Quelle que soit la méthode invoquée, la syntaxe d'une requête HTTP a le même format de base composé de **deux parties** : l'**en-tête** et le **corps**.

Méthode URL Version En-tête : valeur <i>En-tête : valeur</i> <i>etc.</i>	Méthode = (GET, POST, etc.) URL placée dans le champ d'en-tête Host Version = HTTP1.1 En-tête : valeur sont des lignes facultatives (Connection, Accept-, User-Agent, etc.) à l'exception de Host !
Corps de la requête	Le corps est utilisé par exemple pour transmettre des données (POST)

Réponse

Version CodeRéponse En-tête : valeur En-tête : valeur <i>etc.</i>	Version = HTTP/1.1 CodeRéponse = (200 OK, 204 NO RESPONSE, 404 NOT FOUND...) En-tête : valeur sont des lignes facultatives (type de serveur, type MIME, etc.)
Corps de la réponse	Corps : données HTTP (généralement, la page WEB au format HTML)



Nom : prof

Classe : _____

[Accueil] serveur HTTP V4 (MKR Wifi 1010)

Fiche de test et personnalisation de votre projet

En entrant l'@IP de la carte MKR1010 WIFI de test dans un navigateur, vous obtenez les commandes qu'elle reconnaît.

Commandes reconnues:

Activation de (la LED L et de la sortie D0) <- valA ou de la sortie D1 <- valB

- - http://@IP/écriture/valA?val=1
- - http://@IP/écriture/valB?val=1

Désactivation de (la LED L et de la sortie D0) <- valA ou de la sortie D1 <- valB

- - http://@IP/écriture/valA?val=0
- - http://@IP/écriture/valB?val=0

Lecture des entrées analogiques A0 -> val1 et A1 -> val2

- - http://@IP/lire/val1 renvoie la valeur sur l'entrée A0
- - http://@IP/lire/val2 renvoie la valeur sur l'entrée A1
- - http://@IP/lire/vals1et2 renvoie les valeurs sur les entrées A0 et A1 (format JSON)

1. Quelle URL permet d'obtenir DIRECTEMENT la valeur de la grandeur physique mesurée par la carte de test ? Testez cette URL.

@IP Arduino : 192.168.200.40 URL : http://192.168.200.40/lire/val1

2. Comment pensez-vous écrire la partie [/path][document] dans votre carte Arduino afin de l'adapter à votre projet ?
Remarque : le texte utilisé doit être représentatif de la mesure réalisée.

projet : mesurer luminosité => /lire/luminosite

3. Dans votre projet, remplacez le nom du fichier **lireVal1.php** de l'exemple afin qu'il soit représentatif de la grandeur physique que vous mesurez.

____ nouveau nom ____ : luminosite.php



Le code JavaScript doit également être modifié.

Vous avez modifié le nom du fichier **lireVal1.php** !

4. Quelle URL permet d'obtenir INDIRECTEMENT (en passant par le serveur) la valeur de la grandeur physique mesurée par la carte de test ? Testez cette URL.

Remarque : pendant les tests, le serveur est soit le NAS_SIN, soit le Raspberry Pi du groupe.

@IP serveur : 192.168.200.70

URL : http://192.168.200.70/scripts/luminosite.php

5. On souhaite que la LED de la carte de test s'éclaire. Quelle URL permet de la commander DIRECTEMENT ? Testez cette URL.

http://192.168.200.40/écriture/valA?val=1

6. Si vous avez des boutons dans votre page, comment pouvez-vous renommer le fichier **ecritureValA.php** de l'exemple pour qu'il soit représentatif des commandes dans votre projet ?

____ nouveau nom ____ : eclairage.php (c'est eclaireage dans le projet)



Le code JavaScript doit également être modifié.

Vous avez modifié le nom du fichier **ecritureValA.php** !

7. On souhaite que la LED de la carte de test s'éteigne. Quelle URL permet de la commander INDIRECTEMENT (en passant par le serveur) ? Testez cette URL.

Remarque : une donnée (l'état du bouton) doit être passée dans cette URL. Revoir la syntaxe donnée en page 1.

http://192.168.200.70/scripts/eclairage.php?commmande=0