# Introduction

Dans ce chapitre, nous allons nous intéresser à la partie conception qui est une étape primordiale dans le processus de développement en raison des décisions prises dans cette étape et leurs impacts sur les performances de la solution. Nous allons présenter dans un premier temps l’aspect conceptuel de notre solution**.** Ensuite, nous allons représenter l'environnement matériel et logiciel utilisé. Enfin, nous allons discuter le choix du langage de développement logiciel.

# Spécifications des besoins

L’objectif principal est de développer et concevoir un système qui soit :

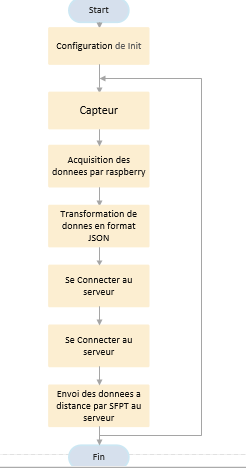
* **Efficace :** le système doit transmettre les données sans interruptions ou coupures.
* **Souple :** il doit être facile à manipuler sans connaissance particulière en informatique ou en électronique.
* **Extensible :** il doit être conçu de telle sorte que nous pouvons le modifier sans besoin de reprendre tout le travail.
* **Flexible :**le système peut être adapté à d'autres cas et situations d'application qui font partie du « smart irrigation».

## 1. Méthodologies

Pour procéder à la conception de l'application, nousl'avons modélisé en utilisant le logiciel de création de diagrammes Microsoft Visio.

* Microsoft Visio

C’est un [logiciel](http://fr.wikipedia.org/wiki/Logiciel) de [diagrammes](http://fr.wikipedia.org/wiki/Diagramme) et de [synoptique](http://fr.wikipedia.org/wiki/Synoptique) pour [Windows](http://fr.wikipedia.org/wiki/Microsoft_Windows) qui fait partie de la suite bureautique [Microsoft Office](http://fr.wikipedia.org/wiki/Microsoft_Office) mais se vend séparément.



Apres la configuration du fichier init pour que le code s’execute à chaque demarrage du raspberry pi ,

la care pi fait l’aquisition des données à chaque periode à travers un capteure de temperature puis les

transforme en fichiers jsons après elle les envoye au serveur par le protocole sftp .

# Environnement matériel

## 1. Système embarqué

Un système embarqué est un système électronique complexe qui intègre du logiciel et du matériel conçus ensemble afin de fournir des fonctionnalités données. Il contient généralement un ou plusieurs microprocesseurs destinés à exécuter un ensemble de programmes définis lors de la conception et stockés dans des mémoires. Le système matériel et l'application logiciel sont intimement liés et immergés dans le matériel et ne sont pas aussi facilement discernables comme dans un environnement de travail classique de type ordinateur de bureau PC (Personal Computer).

En effet, le système embarqué est subdivisé en deux familles : système d’architecture existante et système d’architecture spécifique. Le système à architecture spécifique utilise généralement un microprocesseur combiné avec du matériel et du logiciel pour résoudre un problème de traitement spécifique. Afin d'optimiser les performances et la fiabilité de ces systèmes, des circuits numériques programmables comme les circuits FPGA (Field Programmable Gate Array), comme les circuits dédiés à des applications spécifiques ASIC (Application Specific Integrated Circuits) ou des modules analogiques sont en plus utilisés.

La famille de système à architecture existante est composée de systèmes dédiés et de systèmes généralistes. Les systèmes embarqués d’architecture dédiée sont composés de systèmes embarqués tels que les microcontrôleurs, et de systèmes de traitement numérique de signal composé essentiellement de DSP ou processeurs de traitement numérique de signal.

La tendance aujourd’hui consiste à faire gagner du temps et réduire le coût de développement des systèmes embarqués. En effet, plusieurs fabricants ont eu l’idée de développer des systèmes fonctionnels comme système embarqué prêt à l’emploi. Raspberry Pi, BeagleBone et Arduino sont de nouveaux nés de la famille des systèmes embarqués. Ces cartes intègrent un microcontrôleur ou un microprocesseur avec ses accessoires fonctionnels minimales comme mémoire programme, et mémoire RAM et qui puissent exécuter un système d’exploitation en Temps réel (RTOS). Les cartes sont généralement équipées de système de communication USB et Ethernet. Les nouveaux systèmes développés forment une base de travail indispensable pour le concepteur qui doit aujourd’hui s’occuper de concevoir l’application soft.

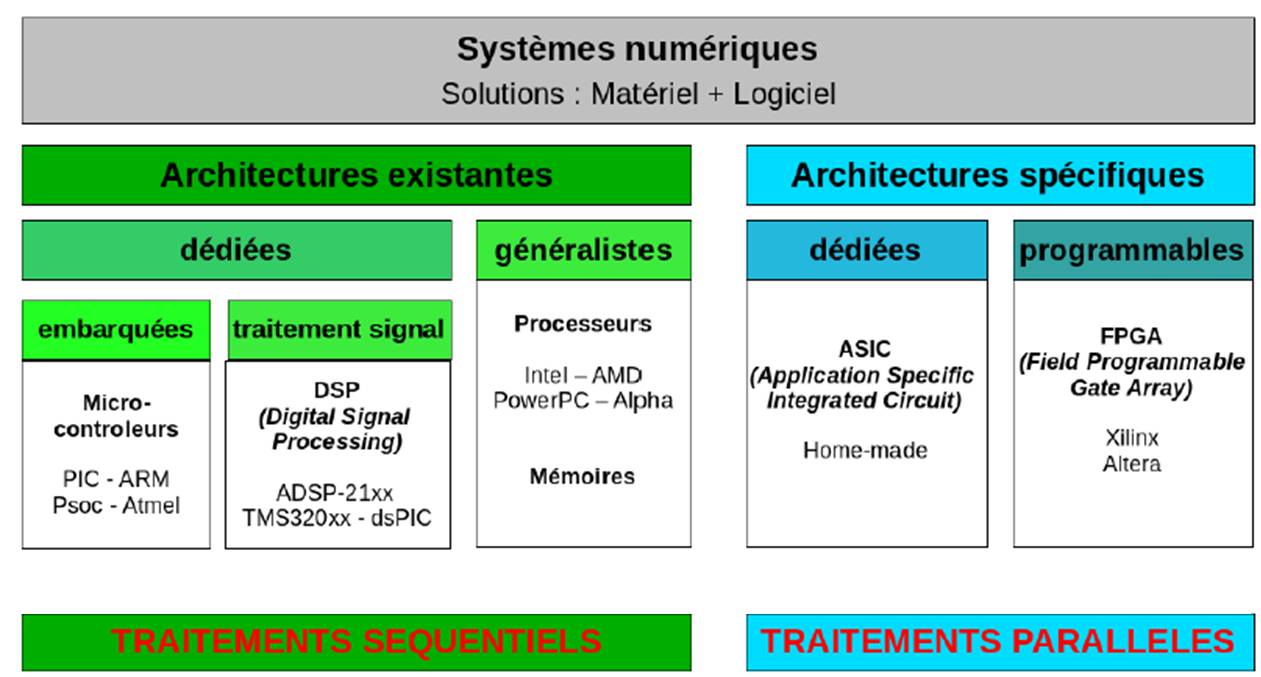


Figure .Architecture des systèmes embarqués

### *Materiel utilis*é

### *Carte Raspberry*

La Raspberry Pi modèle B est un mini-ordinateur à[processeur ARM](http://fr.wikipedia.org/wiki/Processeur_ARM). Cet ordinateur, qui a la taille d'une carte de crédit, permet l'exécution de plusieurs variantes du système d'exploitation libre[GNU](http://fr.wikipedia.org/wiki/GNU)/[Linux](http://fr.wikipedia.org/wiki/Linux" \o "Linux)et des logiciels compatibles. Il est fourni d’une seule carte mère, sans boîtier, sans alimentation, sans clavier, sans souris ni écran dans l'objectif de diminuer les coûts et de permettre l'utilisation de matériel de récupération.

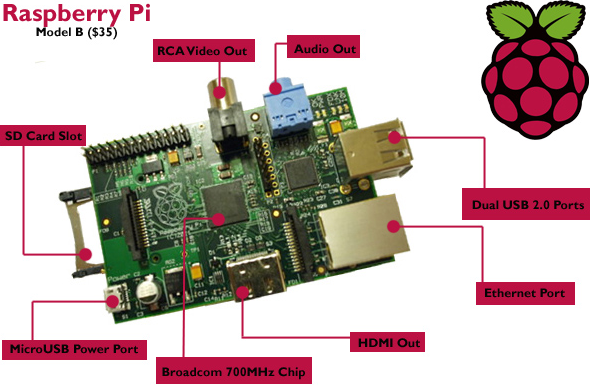
****

Figure . *Une carte Raspberry*

Cemini-ordinateur est destiné à encourager l'apprentissage de la programmation informatique. Il contientdeux ports USB et un port RJ45 et intègre un puissant processeur ARM cadencé à 700 MHz et une mémoire vive de 512Mo.

## . Choix de la plateforme

Il est simple et difficile de choisir entre les plates-formes existantes. Il est simple de choisir s’il s’agit de voir le coût comme paramètre de sélection par exemple. Mais il est difficile de choisir si l’application à développer impose des contraintes surtout le temps-réel.

Dans le dernier cas, plusieurs paramètres entrent en jeu : taille mémoire, rapidité du processeur, système d’exploitation, …

## ****1.Raspberry Pi est petit…****

De la **taille d’une carte de crédit**, une carte Raspberry Pi est parfaitement adaptée pour être

intégrée dans un robot. Pas besoin d’ajouter de la place supplémentaire, ou une charge

additionnelle de 2 kg pour rendre votre robot intelligent.

## ****2.Pas cher – moins de 40€****

Oui, vous avez bien lu : moins de 40€ pour un ordinateur complet. **Raspberry Pi est**

**vraiment accessible.** De ce fait, acheter une carte pour votre robot ne vous ruinera pas, et

n’augmentera pas de façon considérable le prix final du robot. Vous pourrez bénéficier de

nombreuses fonctionnalités très utiles, pour un prix plus que raisonnable.

3.**Raspberry Pi est puissant**

Malgré sa petite taille, ne sous-estimez pas les capacités d’une carte Raspberry Pi ! Celle-ci

est aussi puissante qu’un smartphone (pas les dernières modèles). La dernière version en date,

Raspberry Pi 3, comprend un **processeur 4 coeurs ainsi que 1GB de RAM**. Vous pouvez y

installer un vrai système d’exploitation, comme par exemple Raspbian, Ubuntu ou Windows

IoT, de quoi faire tourner la plupart des applications que vous utilisez sur votre ordinateur

portable.

## ****4. Beaucoup d’entrées/sorties****

Une carte Raspberry Pi n’est pas seulement un ordinateur. Avec **40 pins GPIO**, vous pouvez

facilement connecter votre carte avec de nombreux capteurs et composants électroniques.

En utilisant des protocoles comme **serial, i2c, ou encore spi**, vous êtes assuré de pouvoir

utiliser un maximum de composants électroniques. Ceci est un point important, car cela rend

Raspberry Pi **compatible avec la plupart des appareils et composants** que vous trouverez

pour vos projets robotiques.

## ****5.Une multitude d’appareils connectés****

## ****6. Très facile d’utilisation par rapport aux technologies industrielles****

**Tout le monde peut utiliser une carte Raspberry Pi.** Tout ce que vous avez à faire, c’est de

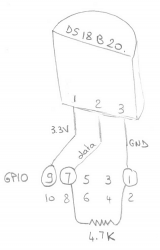
télécharger un système d’exploitation, de l’écrire sur une carte microSD, de connecter votre

Raspberry Pi à un écran, et c’est tout !

# *capteur DS18B20*

Le capteur DS18B20 est un capteur à coût réduit (10 € les 5 capteurs étanches avec câble de 1 mètre ou 8 € les 10 capteurs de base), qui utilise le protocole 1-wire. On peut connecter plusieurs capteurs DS18B20 en série sur un même port du Raspberry Pi. Nous allons voir ici comment connecter un seul thermomètre et en écrire la valeur dans un fichier txt par un script Shell ou PHP.

## *Connecter le capteur DS18B20*



Le capteur peut mesurer des températures entre -55°C et +125°C et reste précis à 0.5°C sur

l’intervalle -10°C à +85°C. Chaque capteur a un numéro de série unique (64 bits) et peut donc

être identifié précisément.

Pour **une seule sonde**, on câble comme dans le schéma de droite.

On peut mettre **plusieurs capteurs en série** (en théorie autant qu’on veut, en pratique il

semble qu’il vaille mieux se limiter à 10 capteurs). Dans ce cas, on connecte une seule

résistance 4.7 K et plusieurs sondes selon le schéma ci-dessous.

## *Paramétrer le Pi pour lire la / les températures*

Editer /boot/config.txt pour qu’il contienne la ligne suivante **puis redémarrer le Pi** :

dtoverlay=w1-gpio

### *Tester le capteur*

### En ligne de commande, taper les lignes suivantes

sudo modprobe w1-gpio

sudo modprobe w1-therm

cd /sys/bus/w1/devices

ls

cd 28-xxxx (changer pour correspondre aux numéros de série qui s'affichent)

cat w1\_slave

Si le capteur est correctement câblé, on voit apparaître quelque chose qui ressemble à ce qui

suit, où t=20187 signifie que la température est 20.187°C :

43 01 4b 46 7f ff 0d 10 bd : crc=bd YES

43 01 4b 46 7f ff 0d 10 bd t=20187

S’il y a plus d’un capteur de température,  ls dans /sys/bus/w1/devices affichera plusieurs répertoires.

### *Installer les drivers sur le Pi*

modprobe w1-gpio et modprobe w1-therm sont les drivers pour les capteurs de

température. Ils doivent démarrer lorsque le Pi démarre. Pour celà, ajouter les lignes suivantes à /etc/modules  :

w1‐gpio

w1‐therm

On redémarre le Pi et les drivers sont maintenant chargés.

1. ***Environnement logiciel***

**Python**  est un [langage de programmation](https://fr.wikipedia.org/wiki/Langage_de_programmation) [interprété](https://fr.wikipedia.org/wiki/Langage_interpr%C3%A9t%C3%A9), multi-[paradigme](https://fr.wikipedia.org/wiki/Paradigme_(programmation)) et [multiplateformes](https://fr.wikipedia.org/wiki/Plate-forme_(informatique)). Il favorise la [programmation impérative](https://fr.wikipedia.org/wiki/Programmation_imp%C3%A9rative) [structurée](https://fr.wikipedia.org/wiki/Programmation_structur%C3%A9e), [fonctionnelle](https://fr.wikipedia.org/wiki/Programmation_fonctionnelle) et [orientée objet](https://fr.wikipedia.org/wiki/Programmation_orient%C3%A9e_objet). Il est doté d'un [typage dynamique](https://fr.wikipedia.org/wiki/Typage_dynamique) [fort](https://fr.wikipedia.org/wiki/Typage_fort), d'une gestion automatique de la mémoire par [ramasse-miettes](https://fr.wikipedia.org/wiki/Ramasse-miettes_(informatique)) et d'un [système de gestion d'exceptions](https://fr.wikipedia.org/wiki/Syst%C3%A8me_de_gestion_d%27exceptions) ; il est ainsi similaire à [Perl](https://fr.wikipedia.org/wiki/Perl_(langage)), [Ruby](https://fr.wikipedia.org/wiki/Ruby), [Scheme](https://fr.wikipedia.org/wiki/Scheme" \o "Scheme), [Smalltalk](https://fr.wikipedia.org/wiki/Smalltalk" \o "Smalltalk) et [Tcl](https://fr.wikipedia.org/wiki/Tool_Command_Language).

Le langage Python est placé sous une [licence libre](https://fr.wikipedia.org/wiki/Licence_libre) proche de la [licence BSD](https://fr.wikipedia.org/wiki/Licence_BSD)[6](https://fr.wikipedia.org/wiki/Python_(langage)#cite_note-6) et fonctionne sur la plupart des plates-formes informatiques, des [smartphones](https://fr.wikipedia.org/wiki/Smartphone" \o "Smartphone) aux [ordinateurs centraux](https://fr.wikipedia.org/wiki/Ordinateur_central)[7](https://fr.wikipedia.org/wiki/Python_(langage)#cite_note-7), de [Windows](https://fr.wikipedia.org/wiki/Windows) à [Unix](https://fr.wikipedia.org/wiki/Unix) avec notamment [GNU/Linux](https://fr.wikipedia.org/wiki/GNU/Linux)en passant par [macOS](https://fr.wikipedia.org/wiki/MacOS" \o "MacOS), ou encore [Android](https://fr.wikipedia.org/wiki/Android" \o "Android), [iOS](https://fr.wikipedia.org/wiki/IOS_(Apple)" \o "IOS (Apple)), et peut aussi être traduit en [Java](https://fr.wikipedia.org/wiki/Java_(langage)) ou [.NET](https://fr.wikipedia.org/wiki/Microsoft_.NET). Il est conçu pour optimiser la productivité des programmeurs en offrant des outils de [haut niveau](https://fr.wikipedia.org/wiki/Langage_de_haut_niveau) et une syntaxe simple à utiliser.

## Pourquoi Python est-il si populaire ?

Un certain nombre de raisons sont à l’origine de ce phénomène.

## 1. Simplicité

Python est un**langage orienté objet simple à utiliser et à mettre en oeuvre**, il est donc logique qu’un grand nombre de développeurs affluent pour **programmer**avec. De plus, le code Python est propre et facile à comprendre.

## 2. Polyvalence

Python**fonctionne sur plusieurs types de systèmes et de plateformes,** cela fait partie des points sur lesquels il se distingue des autres langages de programmation. Sa force réside également dans le fait qu’il peut être utilisé pour réaliser diverses typologies de projets, pour n’en citer que quelques uns :

• développement web, développement de logiciel

• très souvent utilisé dans la **Data Science**

• pour les **opérations de systèmes**

• **coder** des**jeux vidéo avancés**

• certains l’ont même expérimenté avec l’**Intelligence Artificielle**

Il n’y a pas de limites aux applications Python.

## 3. Possibilités web

Python dispose d’un code propre et d’une énorme documentation. Cela aide les développeurs **à créer et à personnaliser leur code facilement et efficacement**. La syntaxe facilite le processus de révision du code. Python possède également plusieurs **GUI frameworks comme par exemple, Django,** qui permettent la création d’applications graphiques, la réutilisation de code et aussi la réalisation d’applications multi-plateformes.

## 4. Promis à un bel avenir

Python domine les concepts émergents tels que la Data Science, l’Intelligence Artificielle et le **Machine Learning**. Ces technologies deviennent de plus en plus importante pour les industries qui génèrent beaucoup de données et qui oeuvrent abondamment sur l’automatisation. Python a prouvé qu’il était capable de **répondre à ces nouvelles problématiques**, c’est pourquoi il est de plus en plus populaire auprès des développeurs.

Le protocole utilisée :

SFTP signifie SSH File Transfer Protocol ou Secure File Transfer Protocol. Comme l’indique la première définition, SFTP fait partie de SSH ou Secure Shell. Il s’agit d’un remplaçant sûr pour l’établissement d’une session de terminal sur des machines UNIX. SFTP est le composant de ce protocole SSH qui assure le transfert de fichiers.