

TP n°3 : Utilisation d'un capteur I2C sous Raspberry Pi

L'objectif de ce TP va consister à lire des données sur un capteur BMP280 branché sur une Raspberry PI et l'envoyer sur une la plateforme Cloud ThingSpeak.

1. – Création et configuration du système :

Vous devez disposer d'une Raspberry Pi3 avec une carte micro SD Windows 10 IoT Standard/Core, d'un poste Windows 10 avec Visual Studio 2019 (session 3iL normale).

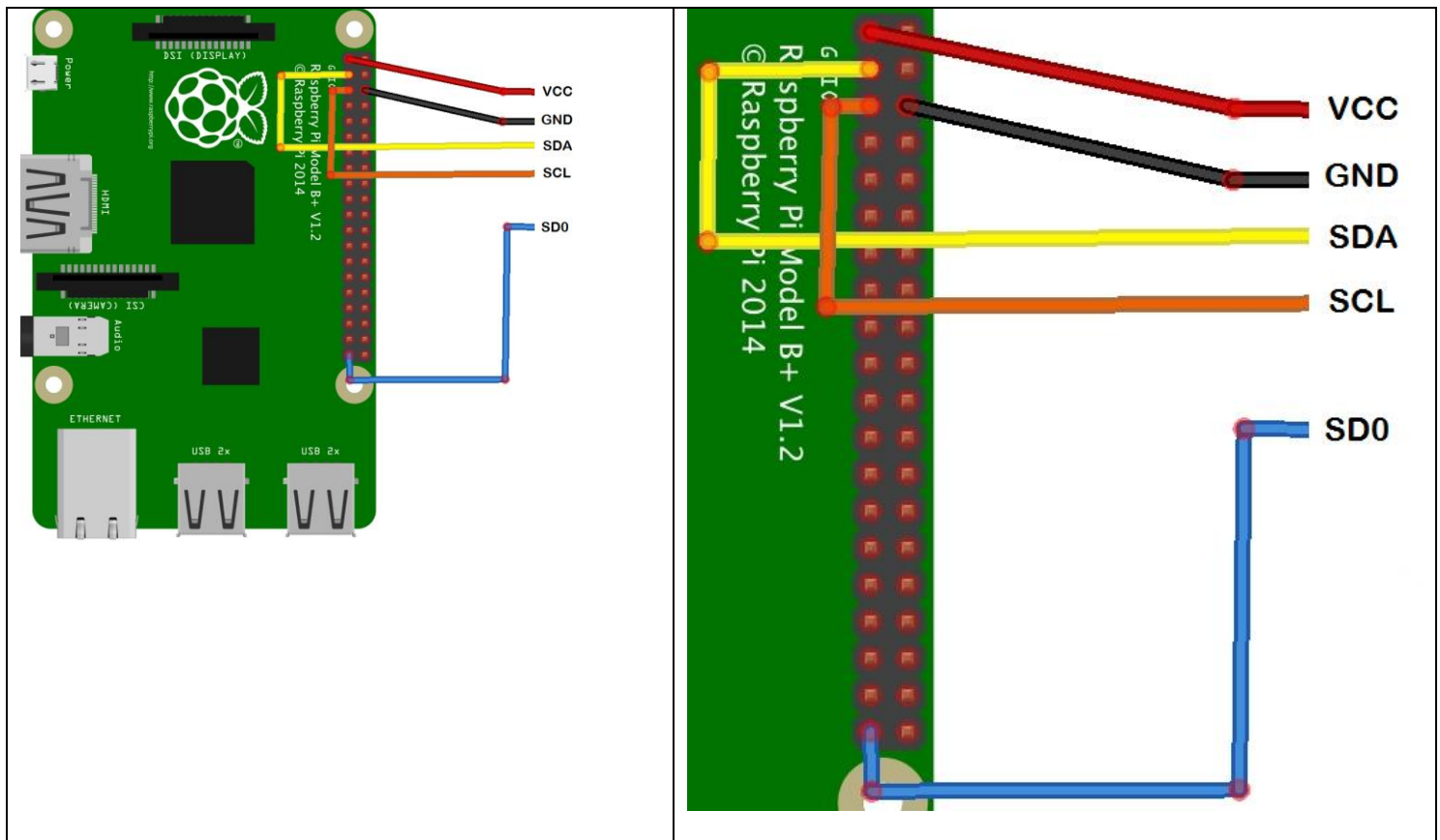
Connectez physiquement votre Raspberry à un écran, à un câble ethernet et à une souris.

Windows 10 IoT Core pour Raspberry doit déjà être installé sur la carte micro SD. Sinon il suffit d'aller sur le centre de téléchargement sur <https://developer.microsoft.com/en-us/windows/iot/downloads> et d'installer « Windows 10 IoT Core Dashboard ».

2. – Connexion du capteur BMP280 :

Vous devez aller câbler un capteur numérique I2C SPI BMP280 3.3V qui mesure la température et la pression, ainsi que l'altitude pour certaines versions.

Vous devez connecter votre capteur en utilisant le schéma suivant. Attention à bien avoir couper l'alimentation de la Raspberry et à bien respecter le câblage.



Le capteur BMP 280 existe sous 2 versions :



Votre configuration est prête. Démarrez votre système.

Notez son adresse ip (ifconfig) :

3. - Lecture des données sur le capteur :

Démarrez Visual Studio 2019 et créez un nouveau projet C# / Universal Windows / Application Vide (ou Blank Projet). Laissez par défaut les versions proposées par Visual Studio.

Configurez Visual Studio pour une exécution distante avec Débuggage. Pour cela, il faut configurer la cible en spécifiant l'IP de votre système IoT. Déroulez la liste déroulante affichant « Local Machine » et sélectionnez « Remote Machine ». Dans la partie « Adress », saisissez l'IP de votre Raspberry et faites « Select ». Enfin sélectionnez « ARM » comme type de processeur.

Lancez votre application sur votre Raspberry et vérifiez qu'elle s'exécute bien.

Ajoutez un bouton « Lecture » et un « TextBox », interceptez la méthode « Click » de votre bouton.

Récupérez le fichier « ClassBMP280 » (consignes données en TP) et ajoutez le à votre projet.

Changez le « namespace » de cette classe pour lui affecter celui de votre projet (pour que cette classe soit visible dans le projet).

Créez une variable globale bmp_280 de type BMP280

Dans le code de votre bouton « Lecture », instanciez cette variable avec l'opérateur « new » et appelez en « await » la fonction d'initialisation du capteur.

Avant de lire les informations disponible, il convient d'adapter la classe « ClassBMP280 ».

A la classe « BMP280 » complétez la méthode « ReadTemperature » qui permet de lire la température. Les registres du capteur permettant de lire la température sont : BMP280_REGISTER_TEMPDATA_MSB, BMP280_REGISTER_TEMPDATA_LSB et BMP280_REGISTER_TEMPDATA_XLSB

Il faudra aussi convertir la valeur en appelant la méthode « BMP280_compensate_T_double ».

Si la méthode « ReadPressure » a besoin d'appeler « ReadTemperature » pour initialiser des registres, la même opération n'est pas nécessaire dans cette dernière méthode.

Créez une variable température de type « float », appelez la méthode « ReadTemperature » en « await » et affichez la température dans la TextBox. Faites la même chose pour la pression. Testez votre application et vérifiez que la température est correcte.

Regardez sur Internet la pression du jour, la valeur correspond t-elle et pourquoi ? :

Ajoutez une correction votre pression pour la ramener au niveau de la mer sachant que la pression diminue d'un 1 hPa tous les 8 m et que l'altitude de 3iL est d'environ 270 m. Testez votre application et vérifiez que la pression est correcte.

4. - Exportation des données vers la plateforme :

Allez sur la plateforme <https://thingspeak.com/> et créez un compte. Lorsque vous êtes connecté, créez un nouveau « Channels », donnez lui un nom et définissez les 2 champs « Field1 » et « Field2 » comme Température et Pression. Sauvegardez.

Revenez à votre application. Ajoutez un bouton « Envoi » et interceptez sa méthode « Click ».

En variable globale ajoutez en constante globale les 2 chaîne de caractères suivantes :

```
private const string _url = "http://api.thingspeak.com/";  
private const string _APIKey = "____KEY_____";
```

La valeur de « _APIKey » est disponible dans votre espace « ThingSpeak » sous l'onglet « API Keys » : valeur de « Write API Key »

Dans la méthode de votre bouton, créez 2 variables « float » pour la température et la pression. Instanciez une variable de type « StringBuilder » (namespace « System.Text ») avec l'opérateur « new ». Construisez votre chaîne de connexion en concaténant (avec la méthode append) :

- L'url de l'API ThingSpeak définie en constante globale
- L'instruction « update » suivi d'un « ? »
- La valeur « key= » suivi de votre valeur « _APIKey »
- La caractère « & »
- La valeur du champ température précédé de « field1= »
- La caractère « & »
- La valeur du champ température précédé de « field2= »

Ajoutez l'espace de nom « using System.Net; »

Ajoutez la méthode suivante :

```
private static string GoToThingSpeak(string QueryString)  
{  
    StringBuilder sbResponse = new StringBuilder();  
    byte[] buf = new byte[8192];  
    HttpWebRequest myRequest = (HttpWebRequest)WebRequest.Create(QueryString);  
    HttpWebResponse webResponse = (HttpWebResponse)myRequest.GetResponse();  
    try  
    {  
        Stream myResponse = webResponse.GetResponseStream();  
        int count = 0;  
        do  
        {  

```

```

        count = myResponse.Read(buf, 0, buf.Length);
        if (count != 0)
        {
            sbResponse.Append(Encoding.ASCII.GetString(buf, 0, count));
        }
    }
    while (count > 0);
    return sbResponse.ToString();
}
catch (WebException ex)
{
    return "0";
}
}

```

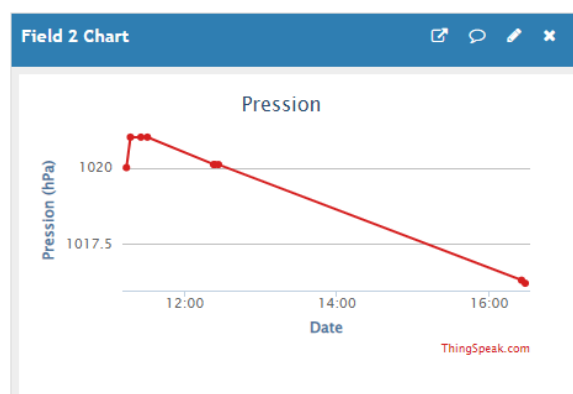
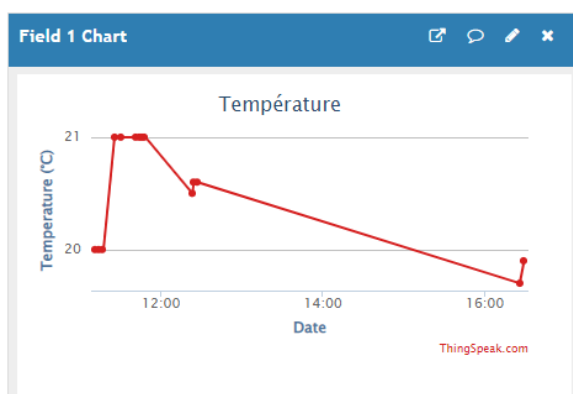
Appelez cette méthode en lui passant en paramètre la chaîne de caractères représentant l'URL à destination de votre plateforme.

Testez votre application et vérifiez que votre espace « ThingSpeak » reçoit les 2 valeurs.

Attention il faut passer par le bouton « Lecture » pour initialiser l'interface du capteur. Envoyez plusieurs valeurs pour commencer à obtenir des courbes.

Affichez les données lues et la valeur renvoyée par la méthode « GoToThingSpeak ».

Que représente cette valeur ? : _____



5. - Automatisation de l'exportation des données :

Créez un 3^e bouton « Envoi périodique » et interceptez la méthode « Click ». Ce bouton doit démarrer un Timer qui va permettre d'envoyer les valeurs mesurées toutes les 30 secondes. un Timer nécessite la création d'une Thread dédiée (afin d'éviter d'occuper la thread principale de l'application). Pour cela vous devez utiliser le code suivant :

```

    TimeSpan period = TimeSpan.FromSeconds(30);
    ThreadPoolTimer PeriodicTimer = ThreadPoolTimer.CreatePeriodicTimer((source) =>
    {
        // Envoie des données
    }, period);

```

Ce code nécessite d'inclure l'espace de nom « Windows.System.Threading; ». Attention, une Thread ne peut pas accéder à l'interface de l'application, il n'est donc pas possible d'afficher directement les données envoyées dans le TextBox.