

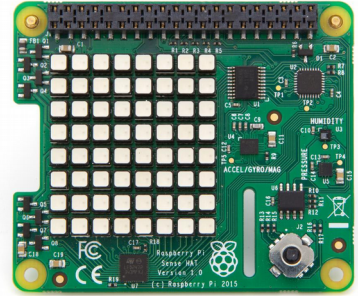
# La carte Sense-Hat du Raspberry-Pi

## 1 - Introduction :

« Sense-Hat » est une carte d'extension pour le Raspberry-Pi. Elle lui permet d'être sensible à un certain nombre d'éléments et d'afficher des informations en utilisant sa matrice d'affichage LED 8x8.

Pour l'utilisation de cette carte, il est nécessaire d'utiliser son gestionnaire qui s'implémente sous la forme d'un objet en C++ : la classe **SenseHat**.

Pour l'ensemble des fonctionnalités d'écrites dans ce document il est indispensable d'ajouter les 2 lignes suivantes dans le code source du programme :

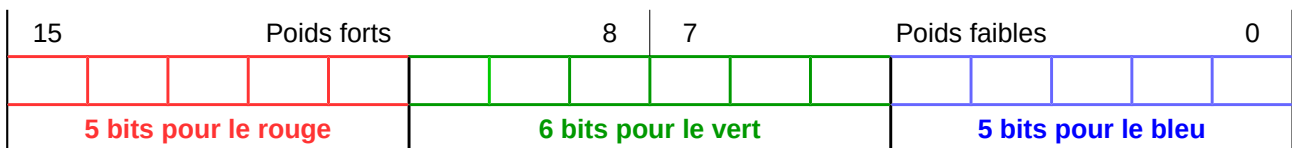


```
#include <SenseHat.h> // à placer au début du fichier source.
```

```
SenseHat sense ; // Déclaration du gestionnaire de la carte Sense-Hat avant son utilisation (instance de la classe SenseHat)
```

## 2 - Utilisation de la matrice de Leds :

Cette matrice est composée de 8 lignes et de 8 colonnes variant chacune de 0 à 7 soit au total 64 leds ou pixels pouvant émettre une lumière codée sur 16 bits en RGB565 soit :



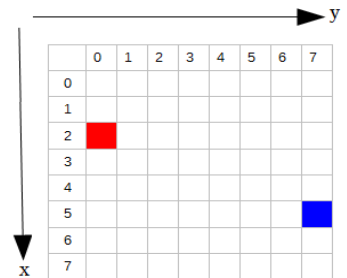
### 2.1 Contrôle individuel de chacun des pixels

```
void SenseHat::AllumerPixel(int x, int y, uint16_t couleur)
```

La matrice peut être ainsi représentée par un tableau

```
#include <SenseHat.h>
```

```
SenseHat sense ;  
sense.AllumerPixel(2,0,ROUGE) ;  
sense.AllumerPixel(5,7,BLEU) ;
```



Dans la fonction **AllumerPixel**, le premier paramètre représente le numéro de la ligne de **0 à 7**, le deuxième représente le numéro de la colonne également de **0 à 7**. Le troisième paramètre correspond à la couleur sur 16 bits non signés de **0x0000 à 0xFFFF**. Un certain nombre de couleurs sont déjà prédéfinies :

ROUGE	0xF800	BLEU	0x07E0	NOIR	0x0000
VERT	0x001F	BLANC	0xFFFF	ORANGE	0xFC00
CYAN	0x87FF	MAGENTA	0xF81F	JAUNE	0xFFE0

La couleur **NOIR** correspond à l'extinction d'une led, le paramètre couleur est à zéro.

### **uint16\_t SenseHat::ObtenirPixel(int x, int y)**

La fonction **ObtenirPixel** retourne, sous la forme d'un entier sur 16 bits non signé, la couleur du pixel dont les coordonnées sont passées à la fonction.

```
#include <SenseHat.h>

SenseHat sense ;
uint16_t couleur ;
couleur = sense.ObtenirPixel(2,0) ;
```

### **uint16\_t SenseHat::ConvertirRGB(uint8\_t rouge, uint8\_t vert, uint8\_t bleu)**

### **uint16\_t SenseHat::ConvertirRGB( uint8\_t tabRVB [ ] )**

La fonction **ConvertirRGB** sous ses deux formes, permet de convertir une couleur exprimée sous la forme de trois entiers non signés sur 8 bits en un entier représentant la couleur codée en RGB565 sous la forme d'un entier sur 16 bits. La première forme reçoit 3 entiers en paramètre, la seconde reçoit un tableau contenant les 3 entiers.

```
#include <SenseHat.h>

SenseHat sense ;
uint16_t couleur1 ;
uint16_t couleur2 ;
uint8_t bleu[3] = {0,0,255} ;
couleur1 = sense.ConvertirRGB(255,255,255) ;
couleur2 = sense.ConvertirRGB(bleu) ;
```

## **2.2 Contrôle de l'ensemble de la matrice de leds**

Les méthodes qui vont suivre agissent sur la matrice de leds dans sa globalité. Elles permettent d'afficher une image, un texte, un caractère, de faire pivoter l'affichage ou d'effacer la matrice.

### **uint16\_t SenseHat::AfficherMotif( uint16\_t tabRVB [ ][8] )**

La méthode **AfficherMotif** permet d'afficher une image ou un motif passé en paramètre sous la forme d'un tableau 8x8 contenant la valeur de chaque pixel.

```
#include <SenseHat.h>
#define N NOIR // redéfinition des couleurs
#define J JAUNE
#define V VERT
SenseHat sense ; // déclaration de l'instance
uint16_t image[8][8] = {
    {N,J,J,J,J,J,J,N},
    {J,V,J,J,J,J,V,J},
    {J,J,J,J,J,J,J,J},
    {J,J,J,V,J,J,J,J},
    {J,J,J,J,J,J,J,J},
    {J,J,V,J,J,V,J,J},
    {N,J,J,V,V,J,J,N},
    {N,N,J,J,J,J,N,N},
};
sense.AfficherMotif(image) ; // appel de la méthode
```

	0	1	2	3	4	5	6	7
0		J	J	J	J	J	J	N
1	J	V	J	J	J	J	V	J
2	J	J	J	J	J	J	J	J
3	J	J	J	J	J	J	J	J
4	J	J	J	V	J	J	J	J
5	J	J	J	J	J	J	J	J
6	N	J	J	V	V	J	J	N
7	N	N	J	J	J	J	N	N

### **unit16\_t SenseHat::Effacer( uint16\_t couleur )**

Par défaut, la méthode **Effacer** provoque l'extinction de tous les pixels, la couleur par défaut est NOIR. En passant en paramètre une autre couleur, le fond d'écran prend la couleur proposée.

```
#include <SenseHat.h>

SenseHat sense ;

sense.Effacer() ;           // la matrice est éteinte
sense.Effacer(ROUGE) ;      // la matrice devient rouge
```

### **unit16\_t SenseHat::PivoterMotif( int rotation )**

La méthode **PivoterMotif** permet de faire tourner le motif présent sur la matrice de 90°, 180° ou 270° soit quart de tour, demi-tour, trois quarts de tour. Seules les valeurs 90,180 et 270 passées en paramètre ont un effet sur l'image. La rotation s'effectue dans le sens horaire.

```
#include <SenseHat.h>

SenseHat sense ;

sense.PivoterMotif(90) ;    // la matrice fait 1/4 de tour
sense.PivoterMotif(180) ;   // la matrice fait 1/2 de tour
sense.PivoterMotif(270) ;   // la matrice fait 3/4 de tour
```

## **3 - Ressentir l'environnement :**

La carte Sense-Hat dispose d'un certain nombre de capteurs ou senseurs lui permettant de détecter les conditions autour d'elle.

### **3.1 La pression**

La pression, "Pressure" en anglais, est la force appliquée sur une surface par unité de surface.

La pression atmosphérique, également appelée "pression barométrique" est la pression exercée par le poids de l'air de l'atmosphère sur la terre. Elle est le plus souvent mesurée en **hectopascals** (hPa). La valeur de 1000 hPa représente la pression atmosphérique normale exercée sur la terre au niveau de la mer.

La pression atmosphérique diminue lentement avec l'altitude. Cette pression diminue donc progressivement depuis le niveau de la mer jusqu'aux limites de l'espace.

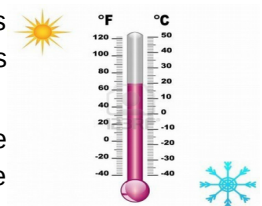


### **3.2 La température**

La température, "Temperature" en anglais, est une grandeur physique qui est reliée à la sensation de chaud et de froid dans la vie courante.

Le capteur de température de la carte Sense HAT peut mesurer des températures aussi basses que -40 degrés Celsius et jusqu'à +120 degrés Celsius.

La carte Sense HAT dispose de deux capteurs de températures. L'un d'entre eux est incorporé au capteur d'humidité et l'autre est incorporé au capteur de pression.

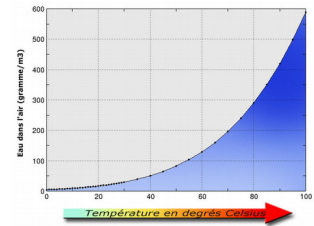


### 3.3 L'humidité

L'humidité, "Humidity" en anglais, est la quantité de vapeur d'eau dans l'air. La vapeur d'eau est l'état gazeux de l'eau. La quantité de vapeur d'eau en suspension dans l'air dépend de la température.

- Plus la température est élevée et plus il y a de vapeur d'eau en suspension dans l'air.
- Moins la température est élevée et moins de vapeur d'eau peut rester en suspension dans l'air.

L'humidité **absolue** est la masse totale de vapeur d'eau en suspension dans un volume d'air donné. La température n'est pas prise en considération. L'humidité absolue est exprimée en gramme d'eau par mètre cube d'air.



L'humidité **relative**, elle, est exprimée en pourcentage. Pour une température donnée de l'air, il y a une quantité maximum de vapeur d'eau que l'air "contenir" en suspension. L'humidité relative est le pourcentage actuel de vapeur d'eau présente par rapport à la quantité maximum que l'air peut contenir (pour cette température). C'est sous cette forme que le capteur d'humidité de la carte fournit les informations mesurées.

### 3.4 Méthodes pour relever ces trois grandeurs :

La classe SenseHat offre 3 méthodes simples pour obtenir la pression, la température ou le taux d'humidité dans l'air.

**float SenseHat::ObtenirPression( )**

**float SenseHat::ObtenirTemperature( )**

**float SenseHat::ObtenirHumidite( )**

```
#include <SenseHat.h> // à placer au début du fichier source.

SenseHat sense ;      // Déclaration de l'instance du gestionnaire de la carte Sense-Hat.
float pression ;
float temperature ;
float humidite ;

pression = sense.ObtenirPression() ;      // retourne la pression en hPa
temperature = sense.ObtenirTemperature() ; // retourne la température en degrés Celsius
humidite = sense.ObtenirHumidite() ;      // retourne le pourcentage d'humidité relative
```

Remarque : Les capteurs étant placés à proximité des autres composants du Raspberry Pi, les valeurs mesurées peuvent être erronées en particulier la température.

[http://wiki.mchobby.be/index.php?title=RASP-SENSE-HAT-ASTRO-PI-D%C3%A9buter#Afficher\\_des\\_images](http://wiki.mchobby.be/index.php?title=RASP-SENSE-HAT-ASTRO-PI-D%C3%A9buter#Afficher_des_images)