L'arrivée du Raspberry Pi, il a deux ans, a ouvert un champ d'exploration pour l'initiation à l'informatique de la nouvelle génération. Alors que depuis des années la formation initiale se cantonne à l'enseignement de la bureautique, pompeusement baptisée "informatique" le Raspberry Pi offre aux enseignants un outil accessible et ludique. En particulier, la disponibilité en standard de Scratch permet aux élèves de développer des programmes sous forme graphique en s'appropriant les bases de l'algorithmique, tout en les aidant à créer et à travailler ensemble. Dans cet article vous apprendrez comment utiliser Scratch pour commander la carte d'interface numérique PiFace et agir sur le monde réel.

Prérequis : Savoir écrire un programme en Scratch

Présentation de la carte PiFace

L'idée de la carte PiFace a été lancée par Andrew Robinson, un chercheur de l'université de Manchester, spécialisé dans les processeurs embarqués et passionné par l'éducation. La carte a été conçue pour être facilement utilisable, initier les enfants à l'électronique et leur permettre d'interagir rapidement avec le monde réel.

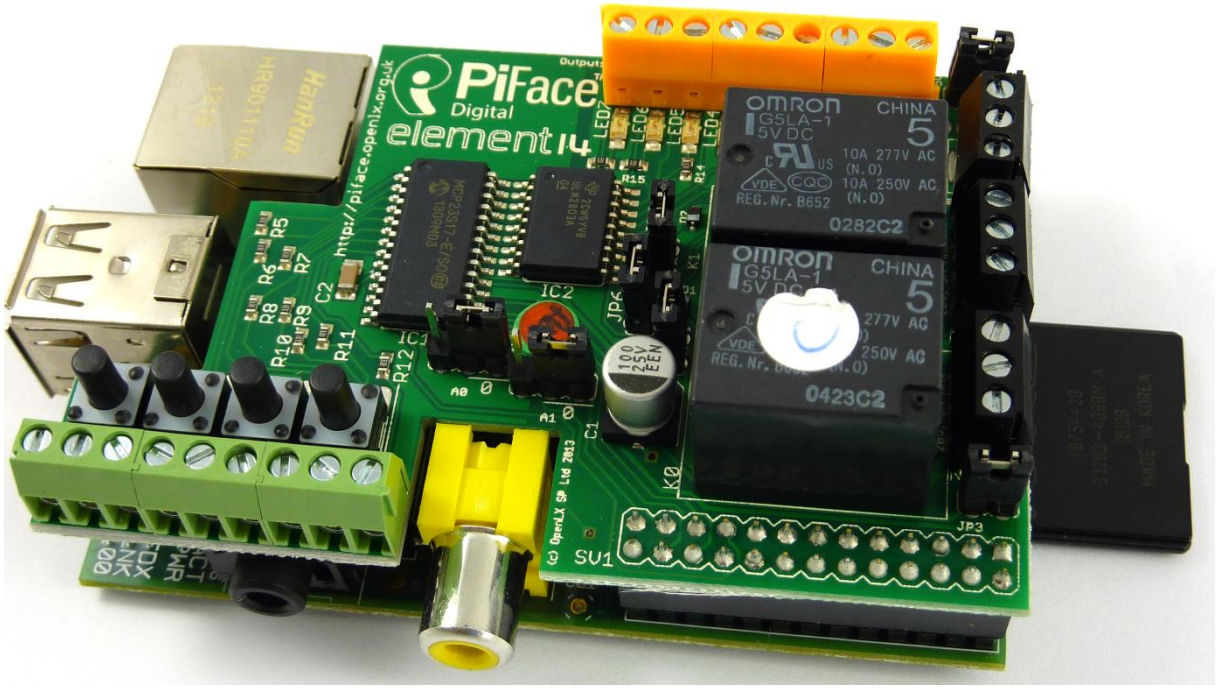

Fig. 1 : La carte PiFace montée sur le Raspberry Pi (photo FM)

La PiFace se connecte directement sur le connecteur GPIO du Raspberry Pi. Les entrées et sorties se raccordent sur des borniers situés le long des côtés de la carte.
La carte est équipée d'un MCP23S17 connecté sur le bus SPI du GPIO, et offrant 16 entrées/sorties numériques. Les 16 I/O se répartissent en 8 entrées (dont 4 équipées de boutons poussoirs) et 8 sorties.

Sorties de la carte PiFace

Les sorties (bornes jaunes) sont bufferisées par un ULN2803 et munies chacune d'une LED de contrôle. Chaque sortie est un Darlington en collecteur-ouvert qui supporte 50 V et peut commander une charge jusque 500 mA. Les sorties 0 et 1 sont équipées chacune d'un relai dont les contacts (NO/NF) sont accessibles sur un bornier à vis (noir).
Des cavaliers présents sur la carte permettent un certain nombre de réglages : adresse de la carte, alimentation externe, diodes anti-retour, désactivation des relais, désactivation des sorties.

Entrées de la carte PiFace

Les entrées de la carte PiFace sont maintenues à l'état haut (1) par une résistance de tirage activée par défaut. Les entrées sont mises à 0 (activées) par un bouton poussoir présent sur la carte (entrées 0 à 3) ou par un dispositif extérieur mettant l'entrée à la masse (entrées 0 à 7).

Remarque : Les entrées et sorties sont numérotées de 1 à 8 sur la carte et non pas de 0 à 7 comme il est d'usage en informatique. Sur l'émulateur graphique, par contre les boutons de commande des sorties sont numérotés… de 0 à 7.

Installation et test de la carte

Installation et mise à jour du système Raspbian

Installez une version "propre" de Raspbian sur une carte SD de 4 Go minimum (*2014-01-07-wheezy-raspbian.zip* au moment de l'écriture de l'article). Mettez en place la carte PiFace sur le Raspberry Pi, en veillant à ce qu'il ne soit pas alimenté (la prise d'alimentation micro-USB ne soit pas connectée). Insérez la carte SD dans son support et branchez la prise d'alimentation. Le système démarre et affiche l'outil de configuration *raspi-config*. Dans le choix 4 Internationalisation Options, réglez les locales sur fr_FR.UTF8 UTF8 puis la zone sur Europe-Paris (si vous résidez en France), enfin configurez le clavier en AZERTY et redémarrez le Raspberry Pi.
Mettez le système à jour :

```
sudo apt-get update
sudo apt-get upgrade
```

Test de la PiFace en Python

Les versions récentes de Raspbian intègrent les outils de gestion de la carte PiFace par défaut. Les programmes de test en Python sont donc disponibles et vous pouvez lancer le programme *blink.py* qui fera clignoter la LED 7 une fois par seconde et permettra de vérifier que tout est en ordre :

```
pi@raspberrypi ~ $ python3 /usr/share/doc/python3-pifacedigitalio/examples/blink.py
```

Le clignotement d'une LED est souvent utilisé en tant que "Hello World" de l'embarqué… La lecture des programmes présents dans le dossier *examples* constitue une bonne introduction au pilotage de la PiFace en Python.

Installation de l'émulateur graphique

Maintenant que nous avons vérifié que tout est en ordre, passons à l'installation de l'émulateur graphique.

```
pi@raspberrypi ~ $ sudo apt-get install python3-pifacedigital-emulator
```

Cette installation va occuper environ 108 Mo supplémentaires sur la carte SD. Apres la fin de l'installation, redémarrez le Raspberry Pi :

```
pi@raspberrypi ~ $ sudo reboot
```

L'icône de l'émulateur PiFace est maintenant présente sur le bureau de LXDE.

Fig. 2 : L'icône de l'émulateur PiFace sur le bureau LXDE

Double cliquez sur l'icône nouvellement installée pour l'interface graphique.

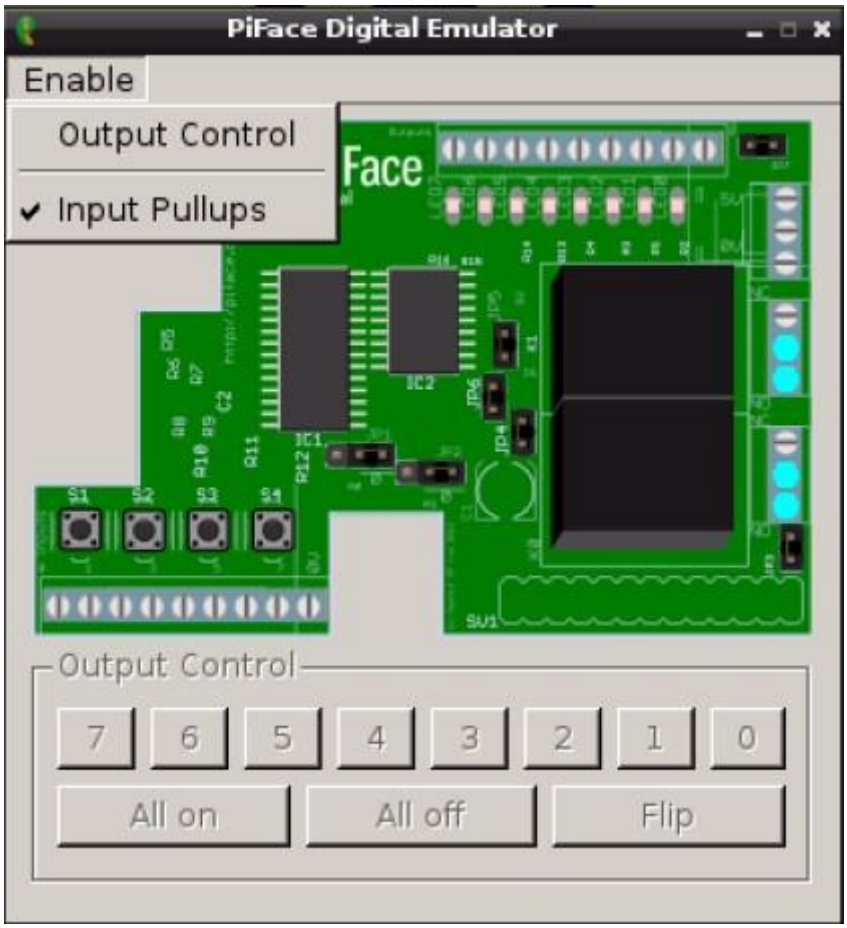

Fig. 3 : Interface graphique de l'émulateur PiFace

L'interface graphique représente la carte PiFace vue de dessus. Par défaut, les sorties sont dévalidées. Pour les mettre en service, cliquez sur **Enable** puis sur **Output Control** pour les mettre en service. En bas de la fenêtre, la zone **Output Control** qui était grisée et non accessible avant la validation des sorties, devient opérationnelle. Les *boutons* numérotés permettent d'activer/désactiver chaque sortie séparément. Vous pouvez vérifier le bon fonctionnement en observant l'allumage/extinction des LED, et pour les sorties 0 et 1, vous entendrez simultanément le bruit des relais.
Le bouton **All on** active toutes les sorties, **All off** désactive toutes les sorties et **Flip** inverse l'état de chaque sortie.
Testez les différentes possibilités en contrôlant que les informations fournies par l'émulateur coïncident avec ce que vous observez sur la carte.

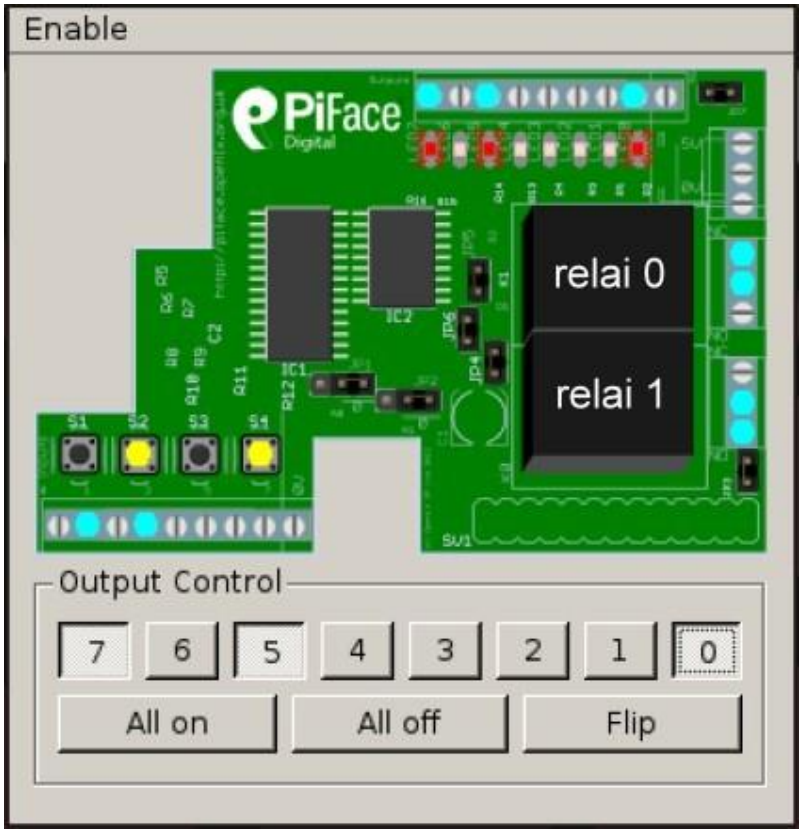

Fig. 4 : Emulateur PiFace avec des entrées et des sorties activées.

Sur la Fig. 4, les poussoirs S2 et S4 sont appuyés, les entrées correspondantes sont indiquées par un cercle coloré qui remplace le dessin de la vis.
Les boutons appuyés apparaissent en jaune, et les LED allumées brillent en rouge.
Le relai 0 (situé sous la rangée de LED) est collé. Le contact est établi entre les deux bornes supérieures. Le relai 1 est au repos, le contact est établi entre les deux bornes basses de son bornier.

Configurer Scratch pour piloter la carte PiFace

Scratch est un environnement de développement graphique et un langage de programmation utilisant des blocs prédéfinis. Les blocs ont des formes qui ne permettent d'encastrer que les blocs autorisés, éliminant de fait les erreurs de syntaxe.
D'origine, Scratch anime des lutins (sprites) sur une scène. Cependant, dans le but d'autoriser le jeu multi-joueurs, une méthode nommée Mesh permet de partager entre plusieurs projets Scratch des variables et des messages diffusés en broadcast, même si ces projets sont hébergés sur des ordinateurs différents. Mesh envoie un message à tous les programmes connectés, chaque fois qu'un broadcast est envoyé dans Scratch, ou qu'une variable globale est modifiée.
Dans notre cas, le serveur Mesh sera hébergé sur l'ordinateur local, et c'est sur le port 42001 que viendra se connecter le programme de gestion de la carte PiFace (Fig. 5).

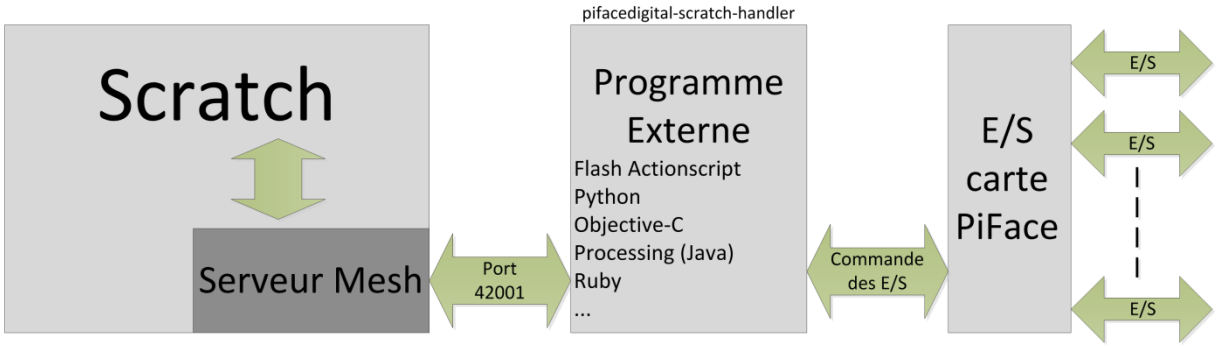

Fig. 5 : Mesh permet à des programmes externes d'interagir avec Scratch.

Mise en route de Mesh dans Scratch

Par défaut Mesh n'est pas activé dans Scratch. Normalement, il faut cliquer sur **Partage** dans la barre de menu, en appuyant sur Shift, pour accéder au menu permettant d'activer Scratch… Enfin ça c'est la théorie, parce que Mesh n'est pas accessible par défaut. Je vous propose donc de voir la procédure pour activer Mesh. Respectez bien toutes les étapes pour obtenir le résultat attendu. Si à l'issue de toutes les manipulations Mesh n'est pas activé, soyez persévérant, et recommencez en suivant scrupuleusement chaque étape.

Fig. 6 : Lorsque Mesh est activé, les deux dernières lignes du menu apparaissent

1 – Ne démarrez Scratch en cliquant sur l'icône du bureau

Démarrez Scratch en super-utilisateur.

Ouvrez un terminal LXterminal et tapez en ligne de commande :

```
pi@raspberrypi ~ $ sudo /usr/bin/scratch
```

L'ouverture de Scratch avec *sudo* permettra après l'activation de Mesh, de sauvegarder les modifications. Si vous démarrez avec l'icône du bureau, les modifications ne seront pas enregistrées et il faudra recommencer la procédure.

2 – Accédez au menu caché de Scratch

Fig. 7 : Accès au menu caché en cliquant sur le rond du R

L'accès au (premier) menu caché de Scratch se fait en maintenant la touche Shift appuyée et en cliquant dans le rond formé par la lettre R de Scratch en haut à gauche de l'écran (Fig. 7).

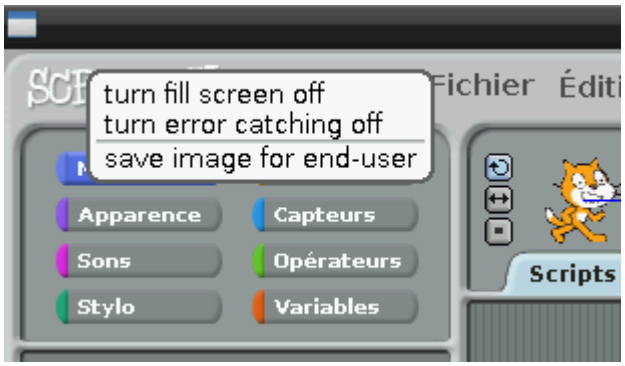

Fig. 8 : Menu caché de Scratch

Dans le menu qui s'ouvre, cliquez sur **turn fill screen off**. Ceci a pour effet de réduire la taille de la zone occupée par l'interface graphique Scratch dans la fenêtre qu'il occupe. L'accès au second menu caché est maintenant possible.

3 – Accédez au second menu caché de Scratch

Déplacez la souris au bas de l'interface graphique de Scratch. L'action précédente a libéré un peu de place dans le bas de la fenêtre.

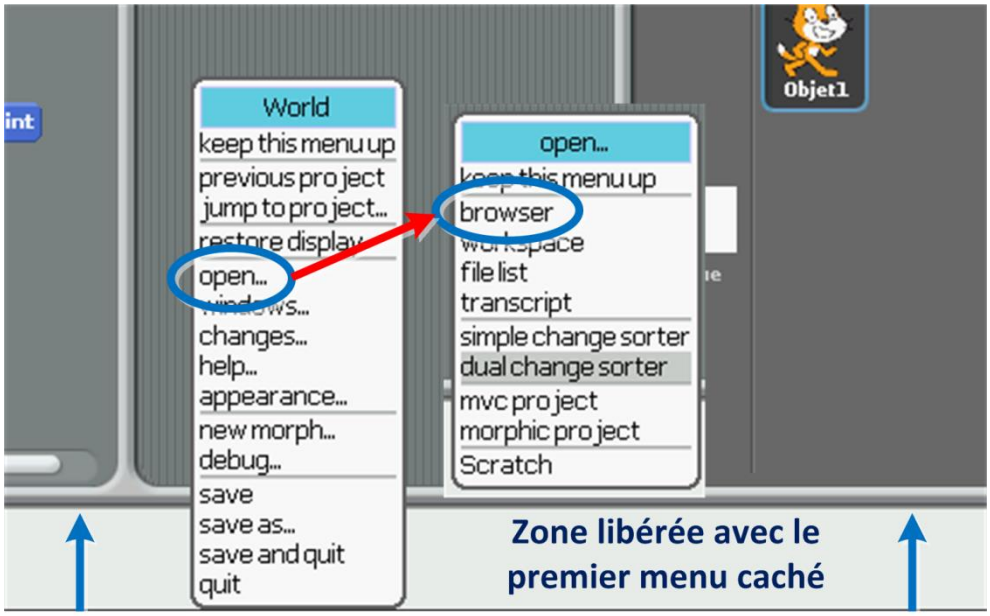

Fig. 9 : Second menu caché permettant d'ouvrir le navigateur d'objet interne à Scratch

Cliquez dans la zone blanche libérée sous l'interface graphique. Un menu s'ouvre : cliquez sur l'item **open** puis dans le menu suivant cliquez sur **browser**. Ceci vous donne accès au navigateur d'objets de Scratch qui va (enfin) vous permettre d'activer Mesh.

4 – Modifier addServerCommandTo

Après ouverture du navigateur d'objets, Cliquez successivement sur **Scratch-UI-Panes** => **ScratchFrameMorph** => **menu/button actions** => **addServerCommanTo:**
Une zone texte apparait dans la partie basse du navigateur. Remplacez le texte "**t2 ← true**" par le texte "**t2 ← false**".

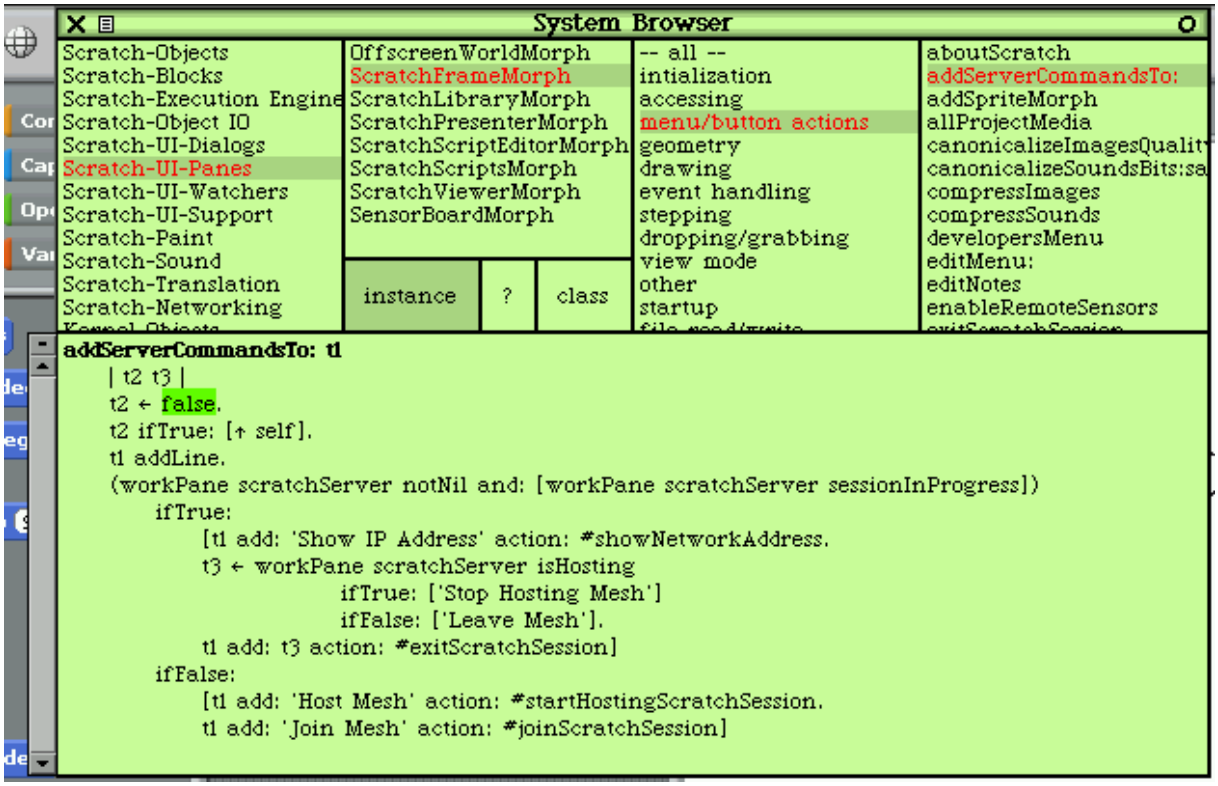

Fig. 10 : Modification de addServerCommandTo:

La modification est faite, il faut maintenant la sauvegarder. Juste à gauche de **addServerCommandTo:** et au-dessus de la flèche de défilement, figure un rectangle contenant un signe **-**. Cliquez sur ce rectangle pour ouvrir le menu permettant de valider la modification que vous avez effectuée.

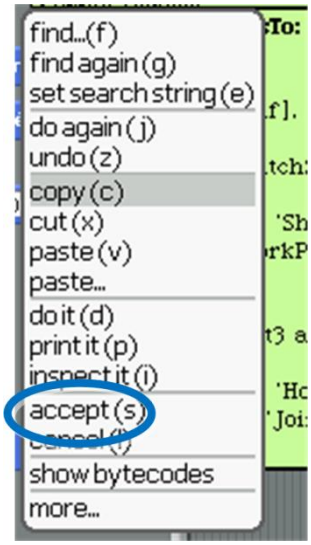

Fig. 11 : Menu permettant de valider les modifications

Cliquez sur **accept(s)** pour valider les modifications que vous avez apportées à **addServerCommandTo:**. Le menu se ferme et vous pouvez quitter le brower d'objets en cliquant sur la croix située à gauche de la barre de titre **System Browser**.

5 – Enregistrer les modifications

Pour enregistrer les modifications que vous venez d'apporter à Scratch, il faut ouvrir à nouveau le premier menu caché, en maintenant la touche SHIFT appuyée et en cliquant dans le rond de la lettre R. Dans un premier temps cliquez sur **turn fill screen on** pour que l'éditeur graphique remplisse à nouveau la fenêtre. Ensuite cliquez sur **save image for end user** pour enregistrer les modifications.

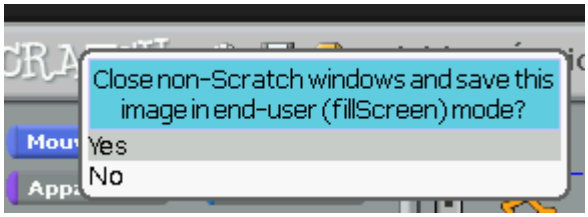

Fig. 12 : Sauvegarde des modifications de Scratch

Répondez **Yes** pour confirmer que vous voulez sauver vos modifications. A l'issue de la sauvegarde, la fenêtre de Scratch se ferme, et vous revenez à LXTerminal.

On ne peut pas dire que la mise en service de Mesh soit facilement accessible, mais si vous suivez l'ordre des opérations minutieusement, le serveur devrait être opérationnel sur le port 42001.

Test du fonctionnement de Mesh

Pour vérifier le bon fonctionnement de Mesh, ouvrez cette fois Scratch en cliquant sur l'icône du bureau. Maintenez la touche SHIFT appuyée et cliquez sur le menu **Partage**. Si Mesh fonctionne, deux lignes supplémentaires apparaissent en bas du menu (Fig. 6).

Relevé de l'adresse du serveur Mesh

Cliquez sur Show IP Address, une fenêtre s'ouvre et vous indique l'adresse du serveur Mesh (Fig. 13).

Fig. 13 : Adresse du serveur Mesh

Notez cette adresse, c'est celle qui permettra au programme *pifacedigital-scratch-handler* de dialoguer avec Mesh. Vous pouvez choisir d'arrêter Mesh en cliquant sur **Stop Hosting Mesh**. La remise en route se fait dans le même menu, en cliquant sur **Host Mesh**, ce qui affiche automatiquement la fenêtre indiquant l'adresse IP du serveur Mesh.

Lancement du programme interface entre Scratch et la carte PiFace

Laissez Scratch ouvert, ou réduisez la fenêtre, mais Scratch doit fonctionner pour que le programme puisse se connecter au serveur Mesh. Ouvrez un terminal LXTerminal et saisissez :

```
pi@raspberrypi ~ $ pifacedigital-scratch-handler 127.0.1.1
```

L'adresse IP sera bien entendu celle que vous aurez relevée à l'étape précédente, sur votre propre Raspberry Pi. Le programme doit démarrer et se connecter au serveur Mesh (fig. 14).

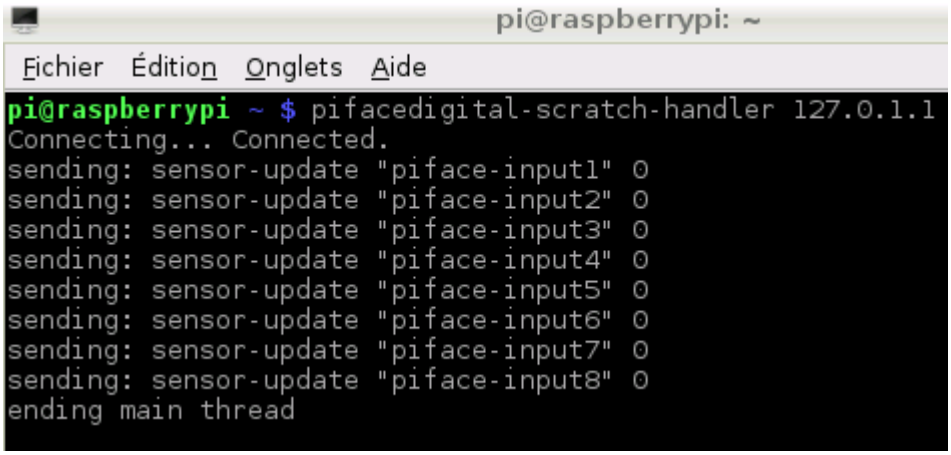

Fig. 14 : Démarrage de pifacedigital-scratch-handler

Le programme initialise également la valeur des variables correspondant aux 8 entrées numériques de la carte PiFace. Attention : la configuration de Scratch est enregistrée et Mesh sera lancé à chaque démarrage de Scratch. Par contre, il faudra répéter le lancement de *pifacedigital-scratch-handler* chaque fois que vous démarrerez Scratch pour l'utiliser avec la carte PiFace. Vous pouvez cependant automatiser le lancement en écrivant un script shell.

Commande d'une LED à partir de Scratch

Dans Scratch, Cliquez sur le bouton **Variables** puis sur **Nouvelle variable**.
Dans la fenêtre qui s'ouvre saisissez le nom de la variable : **piface-output1**. Respectez absolument la façon de nommer la variable, c'est ce qui permet au programme *pifacedigital-scratch-handler* de la reconnaitre ! Laissez coché le bouton radio **Pour tous les objets**, et cliquez sur **OK** pour enregistrer votre variable (Fig. 15).

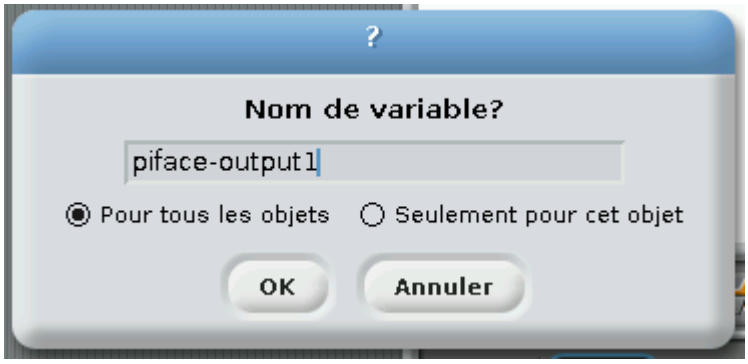

Fig. 15 : Création de la variable piface-output1

Vous disposez maintenant d'une variable qui est liée avec la sortie 1 de la carte PiFace. Déplacez le bloc **à piface-output1 attribuer 0** dans la zone des scripts (zone centrale de l'interface graphique). Modifiez la valeur de la variable dans le bloc que vous venez de déplacer, en remplaçant le 0 par un 1. Cliquez sur le bloc pour appliquer la modification. La sortie 0 de la carte PiFace est activée : la LED s'allume et le relai correspondant colle. Si vous remplacez le 1 par un zéro et cliquez à nouveau sur le bloc, la sortie repasse à 0 et le relai retombe (Fig. 16).

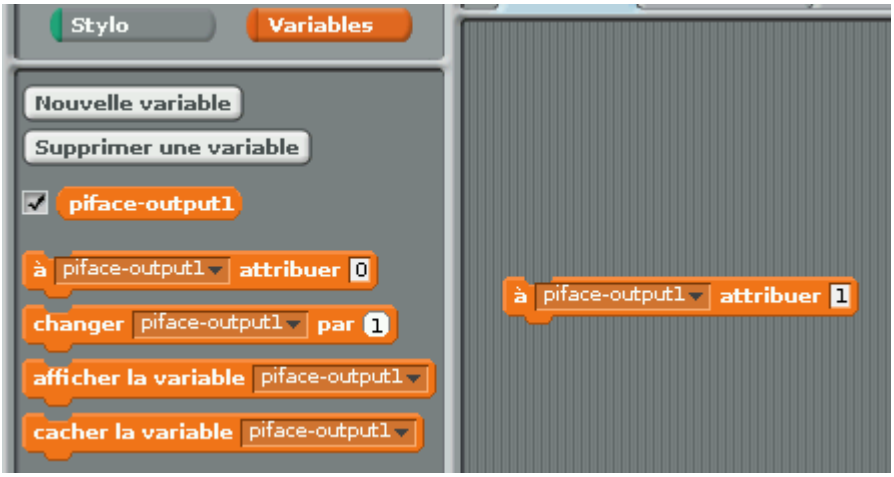

Fig. 16 : Utilisation de la variable piface-output1

Lecture de l'état d'une entrée de la carte PiFace

Dans Scratch, cliques sur le bouton **Capteurs**. Cliquez sur le bloc **valeur du capteur potentiomètre** et déplacez-le dans la zone des scripts. Ouvrez la liste déroulante en cliquant sur la flèche située à droite de **potentiomètre** (fig. 17)

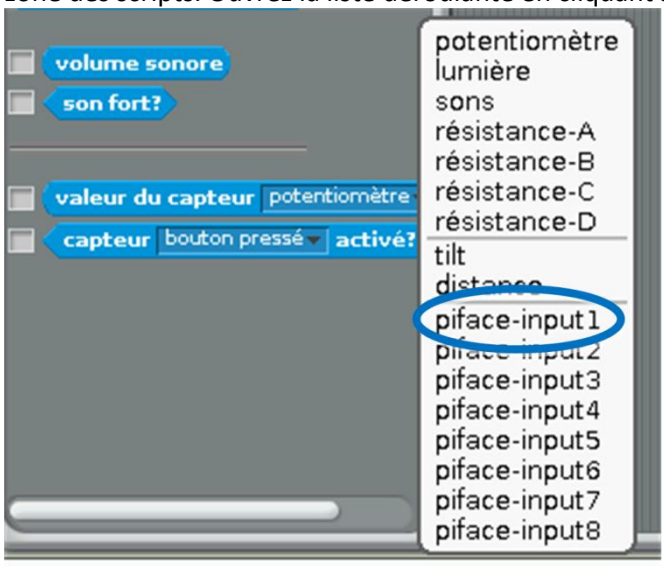

Fig. 17 : choix de la variable d'entrée

Choisissez l'entrée **piface-input1**. Pour utiliser la valeur de cette entrée, renvoyée par le bloc **valeur du capteur**, il va falloir écrire un court programme (fig. 18).

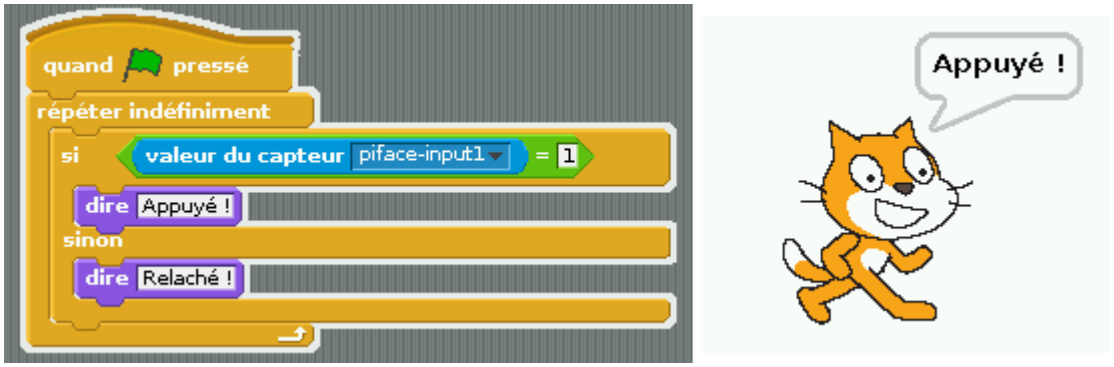

Fig. 18 : Programme de test de la valeur de l'entrée piface-input1 (ici le bouton poussoir est appuyé)

Répéter
    Si le bouton est appuyé
        Le chat dit : Appuyé !
    Sinon
        Le chat dit : Relaché !

En appuyant sur le bouton 1 de la carte PiFace, puis en le relâchant, le chat annonce le résultat en permanence.

Programme en Scratch

Tous les outils nécessaires à la réalisation d'un programme Scratch utilisant la carte PiFace sont maintenant disponibles :

- Installation et test de la carte PiFace
- Installation de l'interface avec Scratch
- Communication entre Scratch et les sorties
- Communication entre Scratch et les entrées

Organisation du programme

```
Annoncer le mode d'emploi du programme
Si le bouton 4 n'est pas actionné
        Si le bouton 1 est actionné
                Annoncer que le chenillard fonctionne
                Activer le chenillard (10 balayages des diodes 1 à 4)
        Si le bouton 2 est actionné
                Annoncer que le clignotant fonctionne
                Activer le clignotant (10 clignotements des diodes 1 à 4)
Sinon
        Dire au revoir
```

La conversion de cet algorithme en Scratch ne pose pas de problème particulier, les blocs du programme correspondent aux blocs de l'algorithme.
A partir de cet exemple, vous pouvez maintenant développer vos propres idées : jeu de dé, commande d'un moteur avec Scratch….

Fig. 19 : Programme utilisant les entrées sorties de la carte PiFace.

Sources :

http://wiki.scratch.mit.edu/wiki/Mesh

http://mchobby.be/wiki/index.php?title=PiFace-Scratch-Demarrer

http://www.piface.org.uk/guides/Install_PiFace_Software/Installing_PiFace_Digital_modules/
http://pi.cs.man.ac.uk/download/EnablingMeshInScratch.pdf