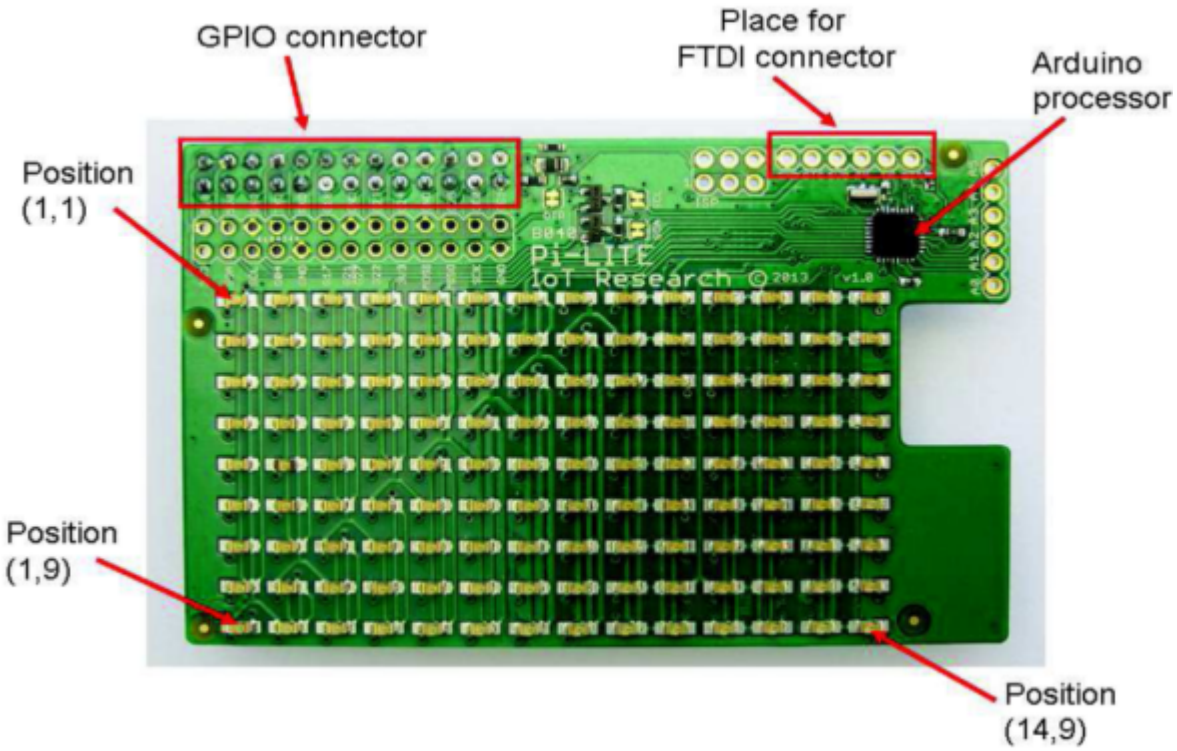
Pi-Lite et raspberry

PI-Lite-Userguide.pdf Ici en anglais

Ci-dessous traduction Google donc plus ou moins juste … de la doc fournie par Henri

Le Pi-Lite est un écran matriciel très polyvalent 9 x 14 LED avec un processeur embarqué Arduino ATmega 328

Chaque pixel est individuellement adressable, de sorte que vous pouvez afficher ce que vous voulez dans la grille. Le Pi-Lite est une carte entièrement assemblée. Il a été principalement conçu pour le RaspberryPi Sheild pour RaspberryPI, , mais fonctionne aussi bien sans le RaspberryPi. L'idée vient du très populaire Arduino LoL Sheild par Jimmy Rodgers et apporte les capacités d'un tel sheild pour le monde du RaspberryPi. LoL signifie « Lots of Leds » ou en Français “Beaucoup de LEDs” bien sûr



Vous pilotez le Pi-Lite via son port série: (UART ou FTDI) 1. Il est vraiment simple d'envoyer du texte et des graphiques aux 126 LED à partir de tout ce qui concerne le port série, un programme ou un moniteur série. 2. vous pouvez programmer le processeur Arduino intégré avec votre propre programme, en remplaçant le logiciel Ciseco pré-chargé

Vous pouvez accéder au port série de différentes façons: 1- branchez le Pi-Lite dans les broches GPIO de votre Raspberry Pi vous y accédez via le port série du RasberryPi à 9600bps.

2- En utilisant le port série sur le connecteur FTDI, vous pouvez utiliser d'autres périphériques que le Raspberry Pi, comme un PC, une machine Mac ou Linux.

Avec un programme autonome approprié, le Pi-Lite peut fonctionner seul, sans connexion à un port série. Par exemple, vous pouvez exécuter votre message d'affichage ou de défilement sur la batterie ou le secteur.

Cette section fait référence au logiciel Ciseco livré avec le Pi-Lite. Une fois que le Pi-Lite est sous tension, vous pouvez communiquer avec lui via le port série. Le port série peut être consulté de deux manières différentes: 1. A partir d'un terminal série ou d'un programme sur le Raspberry Pi lorsqu'il est branché sur le GPIO 2. À partir du connecteur FTDI (décrit dans la section suivante). Lorsqu'il est branché sur le Raspberry Pi, le port série est susceptible d'apparaître sous / dev / ttyAMA0, mais sur votre machine ou contrôleur spécifique, il peut s'agir d'un port différent. Vous pouvez utiliser minicom ou un autre programme de terminal similaire pour y accéder: Minicom -b 9600 -o -D / dev / ttyAMA0 Bien sûr, vous pouvez également adresser le port série à partir d'un programme que vous avez écrit sur le Pi. Voir la section sur les exemples d'inspiration. Remarque: vous pouvez trouver le logiciel préchargé dans le carnet de croquis> bibliothèques> LoLShield> exemples> PI_LITE

Règles générales Les règles générales sont les suivantes: 1. Vous pouvez envoyer n'importe quel texte, qui sera défilé sur le Pi-Lite. 2.Lorsque vous recevez $$$, tous les défilements s'arrêtent et le Pi-Lite passe en mode commande. Voir la référence de commande ci-dessous. “$$$” ne sera pas affiché sur l'écran de défilement. 3.Scrolling va recommencer quand un caractère non-commande est reçu. 4. Tous les caractères de contrôle, y compris l'alimentation de ligne (<LF>) sont ignorés à tout moment. La seule exception est le retour chariot (<CR>) qui termine une commande, mais est ignoré autrement. 5.Toute donnée reçue autre que des commandes commute l'affichage en mode défilement et affiche le caractère via le défilement.

Séquence de mise sous tension À la mise sous tension, le Pi-Lite enverra l'identification et la version vers le bas du port série “Pi-Lite version n.n” Notez que cela ne s'affiche pas sur les LED. La séquence suivante est alors exécutée: 1.delay pendant 1 seconde (0,5 pour le bootoloader Uno et un 0,5 supplémentaire pour $$$ si vous souhaitez interrompre en silence pour éviter l'écran de démarrage) 2.scroll une seule ligne vers le bas et vers le haut 3.scroll une colonne unique à droite et à gauche 4.Display Pi-Lite Logo avec le numéro de version du microprogramme (désigné par LED unique en haut à droite) Cette séquence peut être interrompue en envoyant un $$$.

Présentation de l'écran Pixel (1,1) est en haut à gauche et (14,9) est en bas à droite. Le connecteur GPIO se trouve en haut à gauche.

Mode commande Entrer en mode commande: Quand $$$ est reçu, le Pi-Lite répond avec “OK” et entre en mode commande. Toutes les commandes doivent être terminées par un retour chariot <Command> <CR> Si <command> <CR> n'est pas reçu dans quelques secondes de $$$, la commande est terminée et ignorée. Après avoir reçu et exécuté une commande, le Pi-Lite quitte le mode de commande et revient en mode de défilement. Si des données non valides sont reçues en attendant un caractère de commande, la commande est ignorée et le mode de commande terminé.

commandes	**commentaires**
SPEED(value) $$$SPEED200[RETURN]	LA VITESSE Définir le délai de défilement en mS 1 défile très rapidement, 1000 défile très lentement. La vitesse par défaut est 80
F(value) example (pacman) $$$ F00000000000000000000 1110000 111111100111111101 111111111111 0111111110111 101100011001 1000110000000 00000000000000000000000000 00000	Frame buffer (un chiffre par pixel - 126 chiffres au total) (Valeur) est une chaîne de 126 caractères dans un et des zéros pour représenter chaque état de LED (on ou off)
B(column),(value) examples $$$B1,89[RETURN] - column 1 set to 89% $$$B14,20[RETURN] - column 14 set to 20%	Bargraph (Colonne) 1 à 14 (Valeur) 0 à 100% NB ces commandes ne pas sur l'écriture de l'écran, c'est par la conception pour autoriser les modes mixtes, si vous voulez un écran clair, émettre un ALL, OFF premier

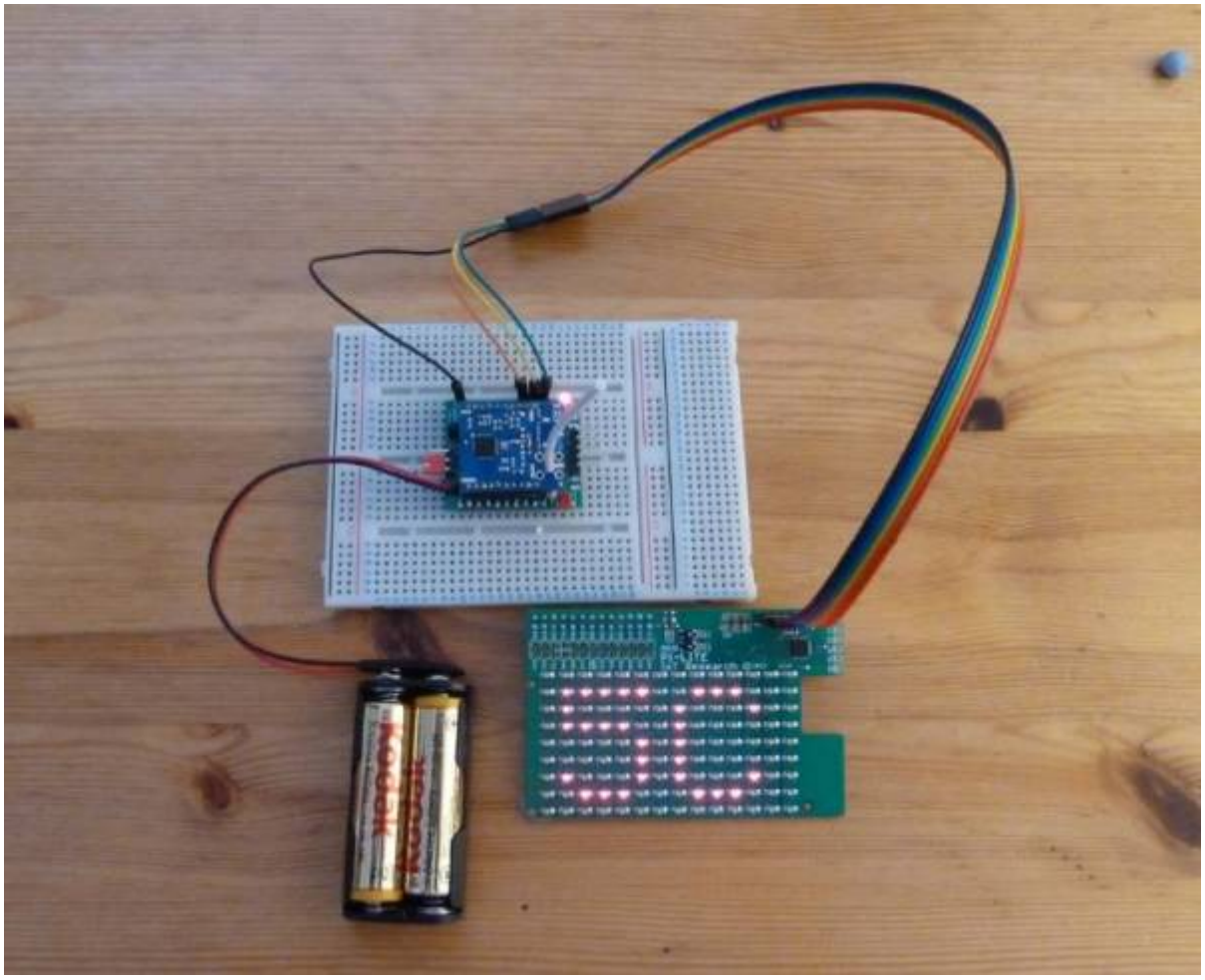
commandes	commentaires
V(row),(value) examples $$$V1,89[RETURN] - bar 1 set to 89% $$$V2,20[RETURN] - bar 2 set to 20%	VU mètre (2 lignes horizontales) placer la rangée r (1-2) à val en (rangée) 1 ou 2 (Valeur) 1 à 100% NB ces commandes ne pas sur l'écriture de l'écran, c'est par la conception pour autoriser les modes mixtes, vous voulez un écran clair, émettre un ALL, OFF premier
P(column),(row),(action) examples $$$P4,4,ON[RETURN] $$$P4,4,OFF[RETURN] $$$P4,4,TOGGLE[RETURN]	Pixel définit un état de pixel (Colonne) 1 à 14 (Ligne) 1 à 9 (Action) ON, OFF, TOGGLE
ALL,(value) examples $$$ALL,ON $$$ALL,OFF	ALL (ALL) permet d'activer ou de désactiver tous les pixels (Valeur) ON ou OFF
SCROLL(value) examples $$$SCROLL-4[RETURN]	Défilement des colonnes val gauche ou droite (Valeur) -1 à -14 se déplacera vers la droite (Valeur) 1 à 14 se déplacera vers la gauche Une fois qu'un pixel est éteint 'affichage, il ne reviendra pas lorsque vous faites défiler vers l'arrière
T(column),(row),(character)examples $$$T1,1,Z[RETURN] - display a Z at location 4,4	Affichage textuel à l'emplacement x, y un caractère texte

Nous avons fourni une gamme d'exemples pour vous montrer quelques-unes des choses que vous pouvez faire avec le Pi-Lite. Vous pouvez utiliser ces exemples pour créer vos propres solutions, car le code est open source.

Le Pi-Lite dispose d'un Arduino embarqué avec un logiciel pré-chargé. En fournissant De la carte Pi-Lite, vous pouvez utiliser soit le port série (broche GPIO 10, UART-RXD et GPIO broche 8, UART- TXD), ou les broches FTDI pour envoyer des données et des commandes pour piloter l'affichage. Vous devez veiller à vous assurer d'utiliser la signalisation série 3.3V ou 5V.

Exemple: une unité d'affichage sans fil.

Par exemple, l'image ci-dessous montre le Pi-Lite (avec sa valeur par défaut). Firmware) connecté à un XRF, après que nous lui avons envoyé la chaîne de caractères 5C $$$ d'un PC. Cela défile dans le texte 5C, puis s'arrête. Défilement en raison de l'initiateur de commande: $$$



Pi-Lite connecté à XRF

Nous avons relié le Pi-Lite via les broches du connecteur FTDI à une carte de déploiement XRF (XBBO) Avec un XRF connecté. Cela permet de placer facilement l'ensemble du système sur une breadboard. Voici comment Nous avons relié les deux ensemble:

Pi-Lite FTDI pin	XRF pin
GNG	10(GND)
CTS	n/c
5VO	1(VDD)
RX	2(DOUT)
TX	3(DIN)
DTR	n/c

Nous avons ensuite utilisé un module SRF connecté à un PC (non représenté sur l'image) pour envoyer des données sans fil et envoyer des commandes au Pi-Lite pour l'affichage. Vous pouvez bien sûr envoyer des données à partir un programme aussi. Pourquoi essayez vous pas de lui faire afficher des messages Twitter, par exemple? Notez que le Pi-Lite fonctionnera avec le 3V fourni.

Programmation du Pi-Lite

Si vous voulez exécuter autre chose que le programme préconfiguré sur le Arduino sur le conseil Pi-Lite, vous pouvez bien sûr le faire. Le Pi-Lite peut étre programmé à partir de n'importe quel ordinateur avec l'IDE Arduino installé sur celui-ci.

Programmation de tout sauf le Raspberry Pi

Si vous utilisez un ordinateur autre que le Raspberry Pi, vous n'avez rien à faire, sinon connecter correctement le Pi-Lite au port série de l'ordinateur que vous utilisez. Il suffit de connecter le Pi-Lite à votre ordinateur de deux façons: 1.Via les broches FTDI sur le Pi_Lite. Vous pouvez aimer souder une prise ou des broches dans la carte Pi-Lite pour ce faire. 2.Via le port série, fourni sur la carte Pi-Lite. Encore une fois ajoutant une prise ou des broches facilite la vie. Vous pouvez ensuite utiliser votre IDE Arduino comme vous le faites pour tout autre panneau basé sur Arduino.

Programmation de la framboise Pi Si vous souhaitez utiliser votre Raspberry Pi pour programmer l'Arduino sur la carte Pi-Lite via le GPIO, Vous devrez régler le Pi de framboise correctement avant que vous puissiez le faire. En particulier, vous devez faire ce qui suit: 1. Installez l'IDE Arduino avec des enveloppes spéciales 2. Installez les bibliothèques Pi-Lite 3. Selon les bibliothèques que vous avez installées, vous devez appliquer un correctif pour Raspberry Pi continue de démarrer et de s'arrêter correctement. Nous vous indiquerons chaque étape ci-dessous.

Étape 1: Installation de l'IDE Arduino avec des wrappers sur le Raspberry Pi

Veuillez suivre attentivement les instructions de ce guide pour installer l'IDE Arduino avec ses enveloppes Et assurez-vous qu'ils sont correctement configurés pour une utilisation via le GPIO. Vous devez suivre ces étapes pour que les choses fonctionnent correctement et avant de progresser à l'étape 2 Étape 2: Installation des bibliothèques Ensuite, vous devez installer la bibliothèque LoL pour le Pi-Lite. Bien que le Pi-Lite fonctionne très bien avec les bibliothèques LoL de Jimmy Rodger, nous vous recommandons d'installer la bibliothèque LoL mise à niveau de Ciseco. C'est en partie parce que le compilateur sur le Pi de framboise est plus pointilleux que les compilateurs plus anciens en usage lorsque la bibliothèque de Jimmy a été libéré. La bibliothèque Ciseco

1.Inclut le programme par défaut que nous avons installé sur le Pi-Lite avant l'expédition 2.Inclure un certain nombre d'améliorations et d'améliorations qui font le Pi-Lite ne LoL choses ne peuvent pas; 3.Includes un correctif qui assure que votre Pi continuera à démarrer correctement une fois que vous avez téléchargé votre propre programme.

Donc, si vous utilisez la bibliothèque LoL Ciseco, vous n'avez pas à effectuer l'étape 4 ci-dessous! La bibliothèque Ciseco LoL est disponible à partir du github ici: https://github.com/CisecoPlc/PiLite.

Téléchargez d'abord le fichier zip de la bibliothèque. Pour ce faire, appuyez sur l'icône ZIP et enregistrez le fichier. Vous devriez voir apparaître un fichier nommé PiLite-master.zip. Si vous n'avez pas été séduit par l'exécution de l'IDE Arduino pour la première fois, faites-le maintenant. Cela créera un dossier appelé sketchbook dans votre répertoire personnel. Nous avons besoin de ce répertoire pour installer les bibliothèques. Fermez l'IDE Arduino pour le moment. Puis jetez un coup d'oeil dans votre dossier de carnet de croquis, susceptible d'être dans / home / pi / sketchbook. S'il ya un dossier appelé bibliothèques dans le carnet de croquis, tout va bien. S'il n'existe aucun dossier de créer un. Double-cliquez sur le fichier PiLite-master.zip que vous avez téléchargé plus tôt. Cela ouvre le Xarchiver. Sélectionnez Action> Extraire et puis cliquez sur Extraire, pour extraire le dossier PiLite-master. Ouvrez le dossier PiLite-master et vous verrez un dossier appelé LoLShield. Vous devez copier ce dossier dans votre dossier de bibliothèques dans / home / pi / sketchbook / libraries. Cela devrait suffire à “installer” la bibliothèque. Démarrez maintenant l'IDE Arduino et vous devriez pouvoir voir un ensemble d'exemples LoL apparaître dans la liste déroulante des exemples de l'IDE Arduino. Si c'est le cas, vous avez installé la bibliothèque avec succès. Si vous avez choisi d'utiliser la bibliothèque LoL de Ciseco, vous avez terminé. Essayez quelques exemples et utilisez-les pour vous inspirer à faire votre propre programme faire exactement comme vous voulez. Si vous avez choisi d'utiliser la bibliothèque LoL de Jimmy Rodgers, vous devez continuer et exécuter l'étape 4 ci-dessous pour vous

assurer que tout fonctionne sans problème. BTW, il ya quelques informations sur le site Arduino sur la façon d'installer des bibliothèques si vous êtes intéressé.

Étape 3: LoL libary fix pour éviter les problèmes de démarrage et d'arrêt Si vous choisissez d'utiliser les bibliothèques LoL de Jimmy Rodger sur le Raspberry Pi, vous devez faire un petit changement dans un fichier pour vous assurer que le démarrage et l'arrêt de Raspberry Pi fonctionnent correctement lorsque le Pi-Lite avec votre croquis Arduino est connecté. Voici ce qu'il faut faire: Recherchez le fichier Charlieplexing.cpp dans le répertoire / home / pi / sketchbook / libraries / LoLShield Allez à la ligne 104, qui ressemblera à ceci: #define MEASURE_ISR_TIME Commenter cette ligne en ajoutant deux barres obliques à l'avant de celle-ci. Ça devrait ressembler à ça: *# define MEASURE_ISR_TIME Enregistrez le fichier et quitter et, c'est tout. Pour ceux d'entre vous curieux sur le pourquoi et le wherefores de cette étape, voici une brève explication: Lorsque les bottes de framboise Pi démarre il cherche à voir si la broche 5 sur le GPIO est faible. Si c'est le cas, il démarre en mode sans échec, au lieu de démarrer normalement. Alternativement, quand le Pi de framboise est Arrêté, il peut être réveillé en amenant la broche 5 vers le bas. Le Pi-Lite a une option matérielle pour travailler en mode SCI et utilise la broche 5 pour SCL. Cela peut interférer avec l'amorçage ou l'arrêt de la framboise Pi, de sorte que le changement de Charlieplexing.cpp garantit que cela n'arrivera pas.*

From:
http://chanterie37.fr/fablab37110/ - **Castel'Lab le Fablab MJC de Château-Renault**

Permanent link:
http://chanterie37.fr/fablab37110/doku.php?id=start:compte_rendu:2017-01-09

Last update: **2023/01/27 16:08**