Mini écran connecté

Afficher des messages où que vous soyez avec cet écran connecté à internet.

Difficulté Moyen

① Durée 1 heure(s)



Catégories Électronique, Maison

① Coût 7EUR(€)

Sommaire

Introduction

Video d'introduction

Étape 1 - Fabriquer l'écran connecté

Étape 2 - Paramétrer Adafruit IO

Étape 3 - Paramétrer les identifiants

Étape 4 - Test depuis Adafruit IO

Étape 5 - IFTTT - Afficher les notifications d'un téléphone

Étape 6 - Cacher vos identifiants dans le logiciel Arduino

Étape 7 - Changer les images

Étape 8 - Changer la police d'écriture

Étape 9 - Utiliser son propre serveur MQTT

Notes et références

Commentaires

Introduction

Précédemment, nous avons vu comment afficher une image sur un ST7789, un écran TFT à 3€. (Afficher une image sur un écran ST7789) Nous allons exploiter cette connaissance pour faire un mini écran connecté.

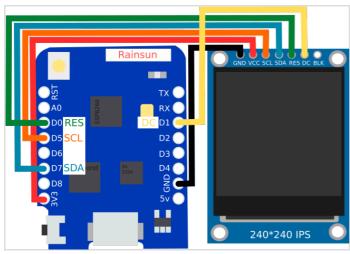
- Cet écran va afficher les messages reçus depuis un serveur MQTT
- Les messages peuvent contenir des caractères accentués
- La transmission sera chiffrée
- Afin d'éviter les attaques MITM (man in the middle), nous allons vérifier l'identité du serveur MQTT à l'aide de son certificat.

À partir de là nous pouvons même utiliser IFTT (ou tasker) afin de transmettre des informations du web / de notre installation domotique ou de notre téléphone sur le serveur MQTT.

Afin de simplifier la partie MQTT, nous allons utiliser Adafruit IO qui permet d'avoir un serveur MQTT gratuitement.



Vous pouvez bien évidemment utiliser votre propre serveur MQTT (la partie IFTT par contre ne marchera plus)



% Afficher une image sur un écran ST7789

Matériaux

- ESP8266
- ST7789
- Câble Wrapping 30 AWG

Outils

- Fer à souder
- Imprimante 3D

Étape 1 - Fabriquer l'écran connecté

Suivez le tutoriel sur l'écran ST7789, avant toute chose, En résumé,

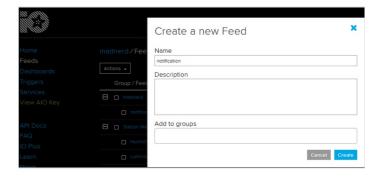
- il faut installer la bibliothèque tft_espi
- Changer le fichier User_Setup.h directement dans la bibliothèque
- Reliez l'écran à l'esp8266

Afficher une image sur un écran ST7789



Étape 2 - Paramétrer Adafruit IO

- Créer un compte sur Adafruit IO: https://io.adafruit.com/
- Cliquer sur Feeds
- Cliquer sur Actions / Create a New Feed
- Dans Name mettez notifications



Étape 3 - Paramétrer les identifiants

Tout d'abord il nous faut récupérer le programme

- Télécharger le croquis ici : https://github.com/maditnerd/st7789_mqtt
- Il nous faut aussi la bibliothèque : Adafruit MQTT

Cette bibliothèque va nous permettre de communiquer avec notre serveur MQTT.

Dans le programme, il y a un fichier arduino_secrets.h, c'est ici que nous allons stocker nos identifiants pour le Wi-Fi et le serveur MQTT Les identifiants sur Adafruit IO se trouvent en cliquant sur View AIO Key

- Dans HOME_SSID mettez le nom de votre réseau WI-FI
- Dans HOME_PASS mettez le mot de passe de votre réseau WI-FI
- Dans ADAFRUIT_MQTT_USERNAME mettez le Username d'adafruit IO
- Dans ADAFRUIT_MQTT_PASS mettez l'Active Key d'adafruit IO

Téléverser le croquis pour tester si tout marche bien.

Normalement vous devriez arriver jusqu'à l'écran Notifications

a

Vous pouvez vérifier sur le moniteur série (Baudrate : 115200) s'il y a un problème.











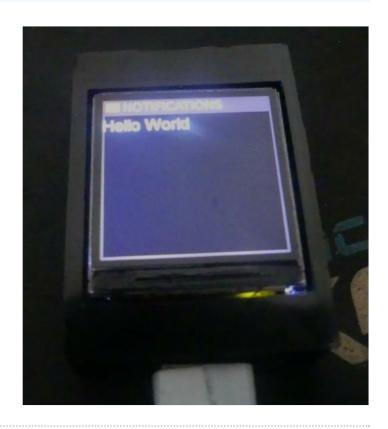
Étape 4 - Test depuis Adafruit IO

Adafruit IO nous permet de créer une donnée qui sera affichée sur l'écran.

- Cliquer sur Feeds / Notifications
- Puis cliquer sur Add Data
- Écrivez un message
 - 0

Si un message est envoyé alors que l'écran n'est pas connecté, celui-ci sera ignoré. MQTT permet toutefois de garder en mémoire les messages avec la fonction Retain





Étape 5 - IFTTT - Afficher les notifications d'un téléphone

IFTTT est un service qui permet d'automatiser des tâches, Adafruit IO est compatible avec celui-ci.



Si vous voulez utiliser IFTTT avec votre propre serveur, il y a des webhooks qui permettent de faire cela.

Nous allons voir comment afficher les notifications d'un smartphone sur notre écran.

Bien que ce soit amusant comme projet, n'oubliez pas que vous allez donner accès à vos notifications à deux services sur internet. Même si les communications sont en théorie sécurisée, niveau vie privée c'est une très mauvaise idée.

- Créer un compte sur IFTTT
- Installer l'application android

if

- Sur l'interface web d'IFTTT, cliquer sur My Applets
- Cliquer sur New Applet
- Choisissez le service Android Device
- Choisissez Notification Received

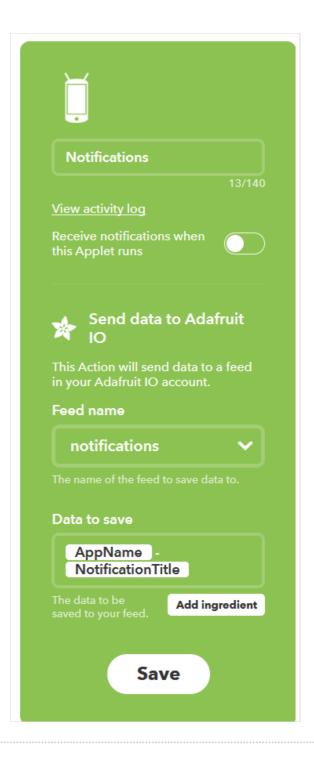
then

- Choisissez Adafruit
- Choisissez Send data to Adafruit IO
- Dans Feed Name mettez notifications
- Dans Data to save choisisez AppName et Notification Title



L'ESP8266 va se déconnecter (puis se reconnecter) du serveur MQTT, si le message est trop long.

Votre applet devrait ressembler à ceci. Aller sur votre téléphone, lancer IFTTT et autoriser **l'accès aux notifications**.



Étape 6 - Cacher vos identifiants dans le logiciel Arduino

Nous avons utilisé le fichier **arduino_secrets.h** qui est dans notre croquis pour **sauvegarder nos identifiants**.



Il vaut mieux éviter de faire cela, car vous risquez de partager accidentellement vos mot de passe!

Une solution pour éviter ça et de créer une **bibliothèque** pour nos identifiants.

L'autre avantage c'est que nos identifiants seront **accessibles pour tous nos croquis**!

C'est Andreas Spiesss qui a eu cette idée https://www.youtube.com/watch? v=CAGQ8h8PKX4&t=347s

- Créer dans Documents/Arduino/libraries un dossier arduino secrets
- Copier arduino_secrets.h dans ce dossier

Dans le croquis, changez l'include au début du code

//#include "arduino_secrets.h"

include <arduino_secrets.h>



Étape 7 - Changer les images

Si vous voulez changer les images affichées, elles sont sauvegardées dans bitmaps.h

Pour convertir vos images en code, suivez ce tutoriel

Afficher une image sur un écran ST7789

Les images en version vectorielle SVG et PNG sont disponibles dans le dossier images

Pour afficher une image, il faut juste utiliser cette commande:

tft.pushImage(0,0,240,240,adafruit_io);

Pour écrire du texte sur plusieurs lignes, positionnez le texte puis écrivez le

tft.setCursor(0,50);

tft.println(data); //Print text

Vous pouvez aussi écrire une seule ligne avec cette commande

tft.drawString(HOME_SSID, 25, 20);







Étape 8 - Changer la police d'écriture

Vous avez peut-être remarqué que la police d'écriture est incluse dans le croquis sous le nom arial 12 pt8b.h

Il se trouve que par défaut, il n'y a pas de police d'écriture avec les accents qui puissent être intégrés directement dans le code.

J'ai donc dû la générer à partir d'une police d'écriture (.ttf).

Vous pouvez télécharger une police d'écriture (par ex: sur https://www.dafont.com/), et la convertir en code.

Pour cela il vous faut fontconvert, le code source est disponible dans Adafruit_GFX mais vous pouvez trouver une version prêt à l'usage (pour Windows) dans la description de cette vidéo

https://www.youtube.com/watch?v=L8MmTISmwZ8

Pour générer la police d'écriture il vous suffit de taper cette commande:

fontconvert arial.ttf 12 32 255 > arial12pt8b.h

Pour changer la taille de la police, changer le premier chiffre (ici 12)

👔 Le 8b signifie que la police utilise 8bytes et contient donc les caractères accentués (entre autres)

Étape 9 - Utiliser son propre serveur MQTT

Le croquis est configuré pour un serveur qui utilise SSL pour chiffrer les communications.

En plus de changer les identifiants / le serveur / le port dans arduino secrets.h, il faut récupérer le certificat SSL de votre serveur. Pour cela il vous faut openssl, vous pouvez le télécharger ici (Windows): https://slproweb.com/products/Win32OpenSSL.html Puis tapez cette commande (avec l'adresse de votre serveur mqtt)

openssl s_client -showcerts -connect io.adafruit.com:8883

Récupérer le dernier certificat affiché et copier le dans connection.h

```
ĎTÉRSKYXTMWJNOVb61RBEØM3gBBq1iAiy800xu1Nq07jdCIQkBsNpFtzbIZhsDS
£T2IGWP4WEmBQ3067c+ZXKFr2DcrXBEtHam80Gp2SNhou2U5U7UesDL/xgLK6/0d7
£T1EVMSUVJkZ8VeZr+IUIIvoLrtjLbqugb0T3OYXW+CQU0kBAgMBAAGjggFAMIIB
£PDAdBgNVHQ4EFgQUIE/UXYvkp0KmgP792PkA760+A1cwHwYDVR0JBBgWFOAUT1JU
BIVSUNUSG-6+rkS7QYXjzkwDgYDVR0PAQH/BAQDAgGGMB0GA1UdJQQMMBQGCCSG
AQUEBwMBBggrBgEFBQcDAjASBgNVHRNBAF8ECDAGAQH/AgEAMDQCCSGAQUEBwEB
£CgwJjAkBggrBgEFBQcwAYYY3HR0CDOVL29jc3AuZ6lnaWNlcnQwY29tMEIGA1Ud
hQYMDkwM6A10DGWMH04H64(99jcmwztmRp72]JZXJ0ElmWb59EaWGPQ2VydEds
£D1bFJvb3RHM15jcmwvPQYDVR0gBDYwNDAyBgRVHSAAMCowKAYIKwYBBQUHAgEW
£GH06HdHBZ018vd3d3LmRpZ2JJZXJ6LmNvb59DUFMwQQYJKOZITNvCMDELBQADgEB
£T1EBDQ6C6WpyGUSKAJjACYwSK41iGF7Kw609711R7z1kwZhRONGOFVU118Vnj2B
£4+/SXmnHJk3mlPyL1xuIAt9sMeC7+vrcRIF5wFBC0MCNSbHwhNN1JzkbifNeP5
£0+Z0+ZdQdFmKCO+nBSKR6HqZA+LMTMCMMUX2khGguPHmtDz44GmEGVZLYFF8EQD3A
                        exbYxTMwVJWoVb6lrBEgM3gBBqiiAiy800xu1Nq07JdCIQkBsNpFtZbIZhsD
               uV6k2Cr/N3XxFpT3hRpt/3usU/Zb9wfKPtWpoznZ4/44c1p9rzFcZYrWkj3A+7
B1E6GmP2fhXhP1D/XVfTW/h0yCJGE1V9Glm/uGOa3DXH1mbAcxSyCRraG+ZBkA
4SeM6YB1/7MBRpPc2618Y=
---END CERTIFICATE-----
                  s:/C=US/O=DigiCert Inc/OU=www.digicert.com/CN=DigiCert Global Root G2
i:/C=US/O=DigiCert Inc/OU=www.digicert.com/CN=DigiCert Global Root G2
     IIDjjCCAnagAwIBAgIQAzrx5qcRqaC7KGSxHQn65TANBgkqhkiG9w0BAQsFADBh
                        wCQVDVQQGEwJVUzEWBBIGA1UEChMMRG1naUNIcnQgSW5jMRkwFwYDVQQLEx83
zZG1nawN1cnQuY29tMSAwHgYDVQQDExdEaWdpQ2VydCBHbG9iYWwgUm9vdCBH
eFw0xMzA4MDExMjAwMDBaFw0zODAxMTUxMjAwMDBaMGExCzAJ8gNVBAYTA1VT
 IJAEFw0xMzA4MDExMjAwMDBaFw0zODAxMTUxMjAwMDBaMGEXCzAJBgNVBAYTAlVT
IRUwEwDVQQKEwsaWdpQzVydCBJbmMxATAXBgNVBASTEHd3dySkaWdpYZVydC5j
20x1Dae9gNVBAMTF0RpZzlDZXJ0IEdsb2JhbCBSb290IEcyMIIBIJANBgkqhki6
bw0BAQEFAAOCAQ8AMIIBCgKCAQEAuzfNNNX7a8myaJCtSnX/RrohCgiNORIUyfuI
20uBjqJkTx65qsGGmvPrC3oXgkkRLpimn7Wo6h+4FRIIAWSULecYxpsMNzaHxmx
1x7e/dfgySSDNo7sH0N03Xss0r0up5/kqbit0tSzpLY16ZtrAgCSYP9PIUkY92eQ
12EGnI/yuum06ZIYa7XzV+hd68ZMHauVBJV38zUtluNJbd13d/tJS7SsVQepjSWz
12COTTG1F8PapsplwHr2HnVYwnS1cUffkdzXOS9XzKBgyMWNGPHgm+F6HmICr0g+UQ
1/IOLCsRnKPZzFBQ9RnbDhxSJITRNrw9FDKZJobq7nMwxM4MphQIDAQABo0IwQDAP
1/2NNNB4F8EBTADAQH/MA4GA1UdDwEB/wQEAWIBhjAdBgNVHQ4EFgQUTIJJUBIV
1/YUNDS/6+FNS7QYXJZkwDQYJNCZIHVCNAQELBQADggEBAGBANKJRVDKhj6zHd6mcY
1/Y19PMWLSn/pvtsr59+wXSN3KjITOYFnQoQj3kVnNeyIv/iPsGEMNKSuIEyExtxW
1/4F2d4m0rvHRAif6zZ0JFrabA0UWTW98kmth/Ssw1HKJ2L7tctu7XUIOGZXIW
1/4F0XNU+MS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMS4MGTAGAUAYMSAMGTAGAUAYMSAGAUAYMSAGAUAYMSAGAUAYMSAGAUAYMSAGAUAYMSAGAUAYMSAGA
```

```
//Adafruit IO Certificate
const char digicert[] = R"EOF(
    -BEGIN CERTIFICATE-
MIIDjjCCAnagAwIBAgIQAzrx5qcRqaC7KGSxHQn65TANBgkqhkiG9w0BAQsFADBh
```

MQswCQYDVQQGEwJVUzEVMBMGAlUEChMMRGlnaUNlcnQqSW5jMRkwFwYDVQQLExB3 d3cuZGlnaWNlcnQuY29tMSAwHgYDVQQDExdEaWdpQ2VydCBHbG9iYWwgUm9vdCBH MjAeFw0xMzA4MDExMjAwMDBaFw0zODAxMTUxMjAwMDBaMGExCzAJBqNVBAYTA1VT MRUwEwYDVQQKEwxEaWdpQ2VydCBJbmMxGTAXBgNVBAsTEHd3dy5kaWdpY2VydC5j b20xIDAeBqNVBAMTF0RpZ2lDZXJ0IEdsb2JhbCBSb290IEcyMIIBIjANBqkqhkiG 9w0BAQEFAAOCAQ8AMIIBCgKCAQEAuzfNNNx7a8myaJCtSnX/RrohCgiN9RlUyfuI 2/Ou8jqJkTx65qsGGmvPrC3oXgkkRLpimn7Wo6h+4FRlIAWsULecYxpsMNzaHxmx lx7e/dfgy5SDN67sH0N03Xss0r0upS/kqbitOtSZpLY16ZtrAGCSYP9PIUkY92eQ q2EGnI/yuum06ZIya7XzV+hdG82MHauVBJVJ8zUtluNJbd134/tJS7SsVQepj5Wz tCO7TG1F8PapspUwtP1MVYwnS1cUf1KdzXOS0xZKBgyMUNGPHgm+F6Hm1cr9g+UQ vIOlCsRnKPZzFBQ9RnbDhxSJITRNrw9FDKZJobq7nMWxM4MphQIDAQABo0IwQDAP BqNVHRMBAf8EBTADAQH/MA4GAlUdDwEB/wQEAwIBhjAdBqNVHQ4EFqQUTiJUIBiV 5uNu5g/6+rkS7QYXjzkwDQYJKoZIhvcNAQELBQADggEBAGBnKJRvDkhj6zHd6mcY 1Y19PMWLSn/pvtsrF9+wX3N3KjITOYFnQoQj8kVnNeyIv/iPsGEMNKSuIEyExtv4 NeF22d+mQrvHRAiGfzZ0JFrabA0UWTW98kndth/JswlHKj2ZL7tcu7XUIOGZX1NG Fdtom/DzMNU+MeKNhJ7jitralj41E6Vf8PlwUHBHQRFXGU7Aj64GxJUTFy8bJZ91 8rGOmaFvE7FBcf6IKshPECBV1/MUReXgRPTqh5Uykw7+U0b6LJ3/iyK5S9kJRaTe pLiaWN0bfVKfjllDiIGknibVb63dDcY3fe0Dkhvld1927jyNxFlWW6LZZm6zNTfl

----END CERTIFICATE----

```
eer signing digest: SHA512
erver Temp Key: ECDH, P-256, 256 bits
```

SL handshake has read 4401 bytes and written 433 bytes

ew, TLSv1/SSLv3, Cipher is ECDHE-RSA-AES256-GCM-SHA384 erver public key is 2048 bit ecure Renegotiation IS supported

pression: NONE ension: NONE

---END CERTIFICATE--

Session:
Protocol : TLSv1.2
Cipher : ECDHE-RSA-AES256-GCM-SHA384
Session-ID: C704AB36DE3C11EB9DD4DF95B1AB2554A3562EF21068CF149B1002697155442D

ubject=/C=US/ST=New York/L=New York/O=Adafruit Industries LLC/CN=*.adafruit.com .ssuer=/C=US/O=DigiCert Inc/OU=www.digicert.com/CN=GeoTrust TLS RSA CA G1

Notes et références

Voilà ce tutoriel est fini, j'espère que cela vous donnera des idées pour faire des applications sympas avec.

Évidemment dans l'idéal

- Il faudrait pouvoir paramétrer notre écran depuis une interface web
- Éviter d'utiliser des images de la taille de l'écran quand ce n'est pas nécessaire

Mais cela devrait vous donner un bon point de départ.

Suivez-moi sur Twitter si vous voulez être au courant des prochains tutoriels en avance : https://twitter.com/m4dnerd