

PiPhone - Un smartphone basé sur Raspberry Pi

Table of Contents

- [PiPhone - Un smartphone basé sur Raspberry Pi](#)
 - [Matériel](#)
 - [Plan d'assemblage](#)
 - [Installation](#)
 - [PiTFT](#)
 - [Installation de piphone](#)

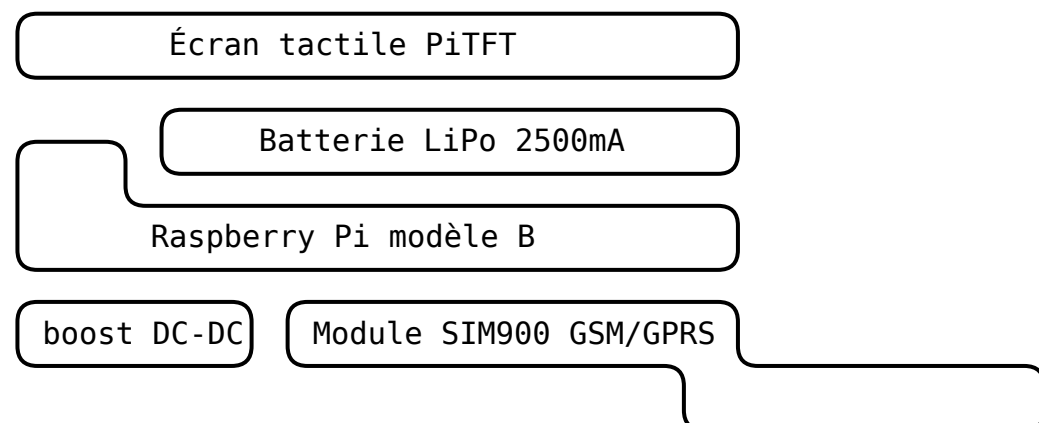
Ce projet a pour objectif de produire un smartphone basé sur un Raspberry Pi. Il utilise une interface à écran tactile Adafruit et un module GSM/GPRS Sim900 pour passer des appels téléphoniques. C'est plus une preuve de concept pour voir ce qui pourrait être fait avec un facteur de forme relativement petit avec des composants prêts à l'emploi (bon marché).

Matériel

Le PiPhone est composé:

1. Raspberry Pi modèle B
2. Écran tactile PiTFT 320 × 240
 - Batterie LiPo 2500mA
3. Module SIM900 GSM/GPRS, qui est connecté via UART au Raspberry Pi
4. Convertisseur boost DC-DC 3.3V - 5V 1A
5. Câbles, connecteurs, interrupteur, etc. - 10 \$

Plan d'assemblage



- L'écrante TFT Adafruit est placé en haut, il sert de clavier numérique, avec un affichage du

numéro à composer en haut, et une icône de téléphone en bas pour passer ou raccrocher l'appel.

- La batterie LiPo s'intègre bien entre l'écran TFT et le Raspberry Pi, afin qu'elle puisse être utilisée de manière autonome, sans aucun fil qui y pend.
- Le module de communication principal, est un module GSM/GPRS Sim900. Cela permet de lui envoyer des commandes AT standard pour passer des appels, raccrocher, envoyer des SMS, des données, etc. Dans l'ensemble, un module très intelligent. Vers le bas du PCB blanc, on peut voir la carte SIM, qui permet au module de s'associer à un réseau GSM local, et il utilise une carte SIM prépayée ordinaire.
- A côté du module GSM, on peut placer l'interrupteur marche/arrêt et un convertisseur DC-DC, qui convertit les 3,7 volts de la batterie LiPoly en 5 volts nécessaires à tout le reste.

<WRAP tip>Le seul problème avec cette configuration est la chaleur. Il fait un peu chaud autour du processeur si on laisse l'appareil allumé pendant plusieurs minutes, car il y a très peu de circulation d'air. Avec un petit ventilateur faisant circuler l'air autour de lui, on peut le laisser allumé pendant de longues périodes pendant le développement, et il était frais au toucher. De plus, l'en-tête à 6 broches du module GSM dépasse un peu, alors on peut dessouder l'en-tête et à souder les fils directement sur le PCB. Cela empêcherait le connecteur de trop dépasser et serait mieux placé lorsqu'il est placé sur une surface plane.<WRAP>

Installation

PiTFT

Étant donné que l'écran PiTFT est assez petit, il faut utiliser des programmes d'interface utilisateur personnalisés. **Pygame** est de loin le moyen le plus simple de le faire.

Installer pip et pygame

```
sudo apt-get install python-pip
sudo apt-get install python-pygame
```

SDL 2.x et SDL 1.2.15-10 présentent de sérieuses incompatibilités avec l'écran tactile. On peut forcer SDL 1.2 en exécutant un script.

Créer le fichier `installSDL.sh` et y coller le texte suivant :

```
#!/bin/bash

# activer les sources de paquets wheezy
echo "deb http://legacy.raspbian.org/raspbian wheezy main" >
/etc/apt/sources.list.d/wheezy.list

# définir stable comme source de paquet par défaut (actuellement buster)
echo "APT::Default-release \"stable\";" >
/etc/apt/apt.conf.d/10defaultRelease
```

```
# définir la priorité de libsdl de wheezy plus élevée que le paquet buster
echo "Package: libsdl1.2debian
Pin: release n=buster
Pin-Priority: -10
Package: libsdl1.2debian
Pin: release n=wheezy
Pin-Priority: 900
" > /etc/apt/preferences.d/libsdl

# installer
apt-get update
apt-get -y --allow-downgrades install libsdl1.2debian/wheezy
```

Exécuter le script

```
sudo chmod +x installSDL.sh
sudo ./installSDL.sh
```

cela forcera l'installation de SDL 1.2

Installation de piphone

Clôner le dépôt:

```
git clone https://github.com/climberhunt/PiPhone.git
```

Usage:

```
cd PiPhone
sudo python piphone.py
```

From:

<http://www.ouarte.garden/> - **dwndoc**

Permanent link:

<http://www.ouarte.garden/doku.php?id=prive:rpi-phone>

Last update: **2025/02/19 10:59**

