

alimentation ATX pour Raspberry Pi



Sébastien Guerreiro de Brito (France)

Vous avez un RPi qui doit être alimenté, ainsi qu'une vieille alimentation ATX ? Un circuit imprimé, quelques composants et un ATtiny feront l'affaire.

Figure 1. Alimentation ATX. Source : Shutterstock.

L'alimentation pour PC au format ATX (**figure 1**) a été introduite par Intel en 1995. Depuis les processeurs de type Pentium II, c'est le facteur de forme le plus répandu pour les alimentations à découpage pour PC. Toutes ses caractéristiques électroniques, mécaniques, environnementales et autres sont spécifiées par Intel et constituent la norme ATX.

Le but de ce circuit est d'utiliser une alimentation ATX pour alimenter une carte RPi ainsi que les nombreux périphériques qui peuvent y être connectés. L'avantage de cette alimentation est le nombre de tensions différentes disponibles - à des niveaux de puissance intéressants.

Le connecteur d'alimentation ATX

Nous n'allons pas décrire ici le fonctionnement interne de l'alimentation à découpage ATX, mais seulement essayer de la faire fonction-

ner. Pour commencer, examinons en détail le connecteur (**figure 2**) qui doit être relié à la carte mère du PC.

Le connecteur est composé de

- un connecteur femelle Molex Mini-Fit Jr à 24 positions et deux rangées (réf : 39-01-2240) ou équivalent ;
- des contacts femelles Molex Mini-Fit HCS (réf : 44476-1112)

Le **tableau 1** présente le brochage de ce connecteur.

Les signaux intéressants pour piloter l'alimentation sont PWR_OK et PS_ON#. L'alimentation +5 VSB sera également utilisée dans notre montage.

Le signal PWR_OK est un signal « Power good » utilisé par l'alimentation pour indiquer

au système que les sorties +5 VCC, +3,3 VCC et +12 VCC sont présentes et conformes aux seuils.

Le signal PS_ON# est celui qui va nous permettre de démarrer l'alimentation. C'est un signal compatible TTL, actif-bas, qui permet à la carte mère de contrôler l'alimentation à distance. Cela permet l'allumage et l'extinction de l'alimentation par logiciel, le Wake-on-LAN, le wake-on-USB, etc.

Lorsque PS_ON# est tiré vers le bas, l'alimentation doit activer les quatre principaux rails d'alimentation CC : +12 VCC, +3,3 VCC, +5 VCC et -12 VCC. Lorsque PS_ON# est tiré vers un niveau TTL haut, ou un circuit ouvert, les rails de sortie CC ne doivent pas délivrer de courant et doivent être maintenus à un potentiel nul par rapport à la masse. Une alimentation ATX dispose d'un circuit antirebond interne pour empêcher les oscillations

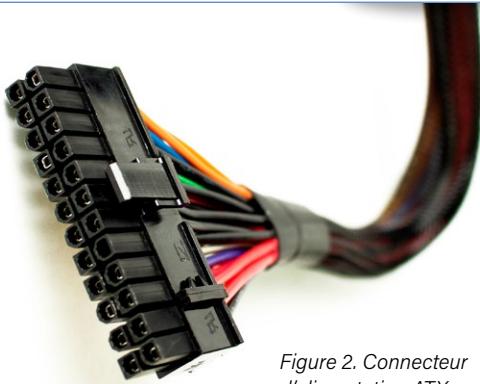


Figure 2. Connecteur d'alimentation ATX.
Source : Shutterstock.

de mise sous/hors tension si le signal PS_ON# est activé par un interrupteur mécanique.

Les caractéristiques du signal PS_ON# sont énumérées dans le **tableau 2** et présentées sous forme de graphiques à la **figure 3**.

Le signal PS_ON# n'a aucun effet sur la sortie +5 VSB. La tension +5 VSB (SB = standby) est présente tant que l'alimentation est connectée au secteur. Cette sortie permet d'alimenter les circuits qui doivent rester opérationnels lorsque les rails d'alimentation principaux sont désactivés.

Description de la carte électronique

Pour permettre à l'alimentation ATX d'être commandée par le RPi, on a conçu un circuit imprimé (PCB) [1]. Il contient la logique nécessaire pour permettre le démarrage et l'arrêt du RPi en y ajoutant une petite UC. Cela offre aussi d'autres avantages intéressants. Comme nous disposons d'une alimentation ATX, nous utiliserons le rail 5 V pour ajouter un jeu de ports USB au PCB, ce qui permettra aux périphériques exigeant des courants plus élevés d'en profiter. La **figure 4** montre le PCB en action.

| Paramètre | Minimum | Maximum |
|-----------------------|---------|----------------------|
| VIL | 0 V | 0.8 V |
| IIL | - | -1.6 mA ¹ |
| VIH | 2.0 V | - |
| VIH en circuit ouvert | - | -5.25 V |
| Ondulation/Bruit | | 400 mV crête |

¹ Remarque : un courant négatif indique que le courant circule de l'alimentation à la carte mère.

| Pin | Signal | Wire color |
|-----|----------|------------|
| 1 | +3.3 VDC | Orange |
| 2 | +3.3 VDC | Orange |
| 3 | COM | Black |
| 4 | +5 VDC | Red |
| 5 | COM | Black |
| 6 | +5 VDC | Red |
| 7 | COM | Black |
| 8 | PWR_OK | Grey |
| 9 | +5 VSB | Pink |
| 10 | +12 V1DC | Yellow |
| 11 | +12 V1DC | Yellow |
| 12 | +3.3 VDC | Orange |
| 13 | +3.3 VDC | Orange |
| 14 | -12 VDC | Blue |
| 15 | COM | Black |
| 16 | PS_ON# | Green |
| 17 | COM | Black |
| 18 | COM | Black |
| 19 | COM | Black |
| 20 | Reserved | NC |
| 21 | +5 VDC | Red |
| 22 | +5 VDC | Red |
| 23 | +5 VDC | Red |
| 24 | COM | Black |

Tableau 1. Brochage du connecteur ATX.

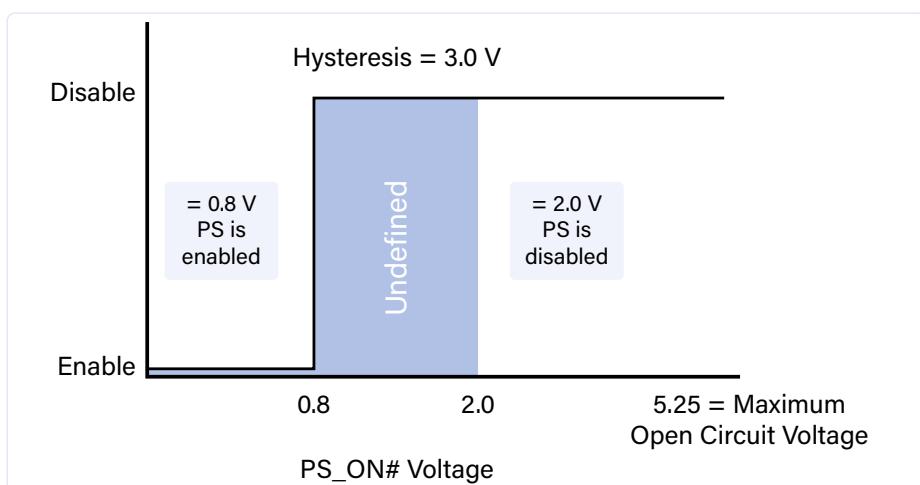


Figure 3. Caractéristiques du signal PS_ON#. (Source : ATX/ATX12V Power Supply Design Guide Version 1.1, Section 3.3.2, figure 3)

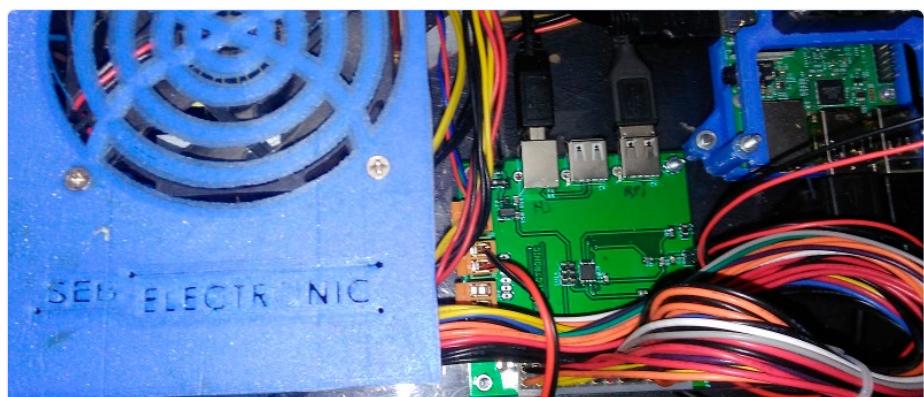
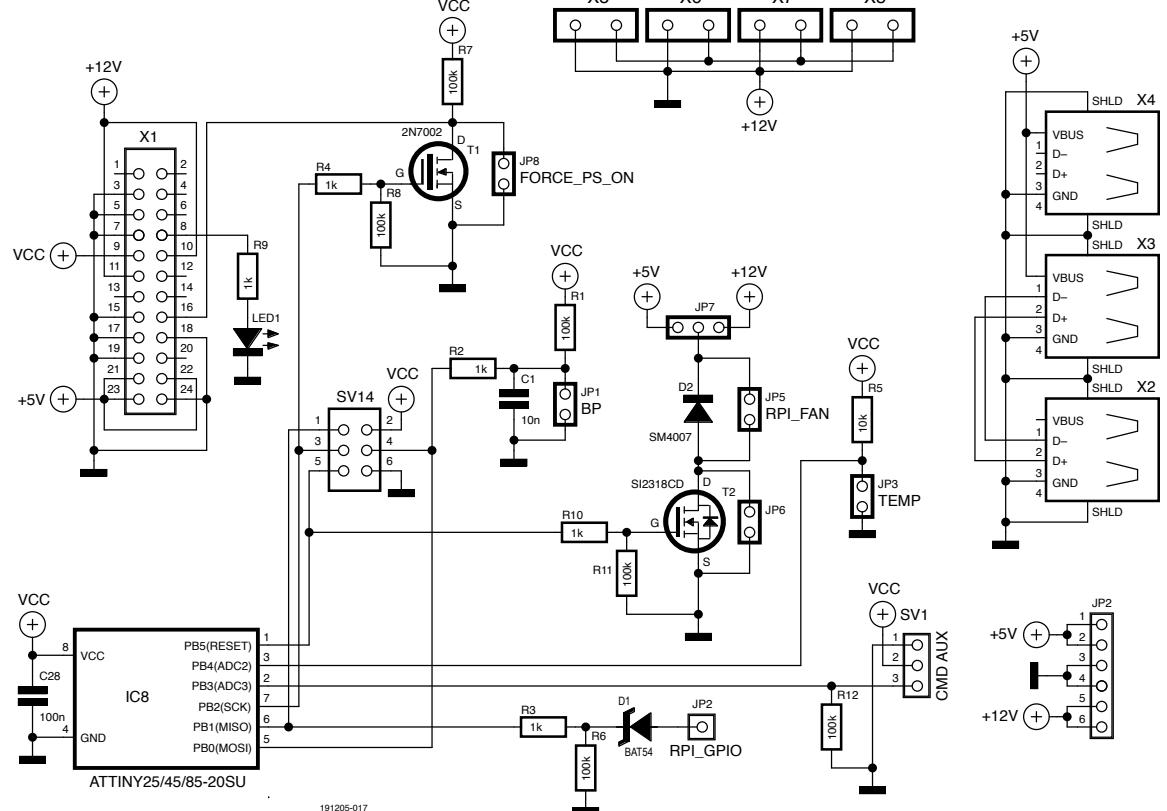


Figure 4. Alimentation ATX connectée au PCB.

Tableau 2. Caractéristiques du signal PS_ON#.

Figure 5. Schéma.



On peut voir le schéma du PCB à la **figure 5**. Le principe de commande de l'alimentation ATX est relativement simple : il est basé sur l'utilisation d'un microcontrôleur ATTiny85 alimenté directement par +5 VSB. L'avantage d'utiliser un microcontrôleur est que, bien que la commande de l'alimentation soit simple, on peut y ajouter d'autres fonctionnalités.

Démarrage de l'alimentation électrique

L'ATTiny reçoit l'ordre de démarrer l'alimentation par un bouton poussoir externe soudé au cavalier JP1.

Lorsque le bouton poussoir est activé, le microcontrôleur commande le transistor T1, qui démarre l'alimentation. Le cavalier JP8 permet de démarrer l'alimentation en mode automatique.

Gestion de la température du processeur du RPi

Nous dotons la carte du RPi d'un refroidissement par air supplémentaire. Pour cela, on y place dessus une thermistance NTC de 10 kΩ et, selon la température mesurée, la carte déclenchera un ventilateur. On sélec-

tionne la tension du ventilateur (12 V ou 5 V) avec le cavalier JP7.

Communication avec le RPi

Pour pouvoir éteindre l'alimentation ATX depuis le RPi, comme sur un PC, nous avons mis en place un petit stratagème.

Le signal RPI_GPIO est connecté au GPIO27 sur la carte RPi. On voit le connecteur GPIO du RPi sur la **figure 6**.

Au niveau du système d'exploitation, il faut d'abord s'assurer que le GPIO27 est accessible. Pour cela, nous créons le fichier d'initialisation de GPIO (exécuté au démarrage) avec l'éditeur nano :

```
nano S75gpioinit
```

Ensuite, nous entrons le code du **listage 1**.

Nous rendons le fichier exécutable et le plaçons dans le dossier d'initialisation.

```
sudo chmod +x S75gpioinit
sudo mv S75gpioinit /etc/init.d
```

Vous remarquerez dans le script d'initialisation que le GPIO27 est initialisé avec une valeur de 1. Ainsi, lorsque nous éteindrons le

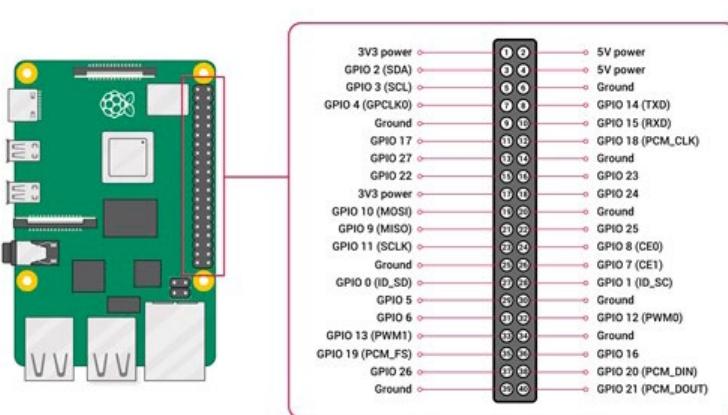


Figure 6. Brochage du RPi (Source : [2]).

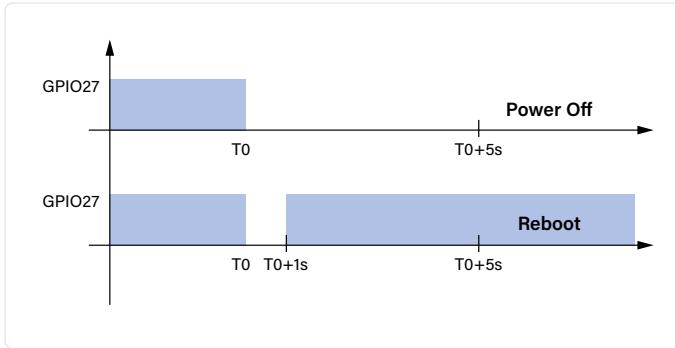


Figure 7. Gestion de la broche GPIO27.

RPi via le système d'exploitation (poweroff, par exemple), cette entrée repassera à 0 et le microcontrôleur saura que c'est le moment d'éteindre l'alimentation ATX.

Mais, plus amusant, nous voulons pouvoir redémarrer la carte RPi par réinitialisation de l'alimentation électrique. Pour cela, l'astuce

que nous utilisons est de faire varier le GPIO27 afin de signaler au microcontrôleur que ce n'est pas une mise hors tension, mais une réinitialisation, qui est demandée.

On peut voir sur la **figure 7** comment est géré le GPIO27 : Lorsque le microcontrôleur voit la broche GPIO27 à 0, il attend 5 secondes, puis,

si la broche est toujours à 0, cela signifie que le RPi a été éteint, donc il éteint l'alimentation ATX. Sinon, c'est qu'il reçoit une commande de redémarrage et provoque l'arrêt de l'alimentation ATX pendant 2 secondes, avant de la redémarrer.

Pour que tout cela soit possible, nous devons

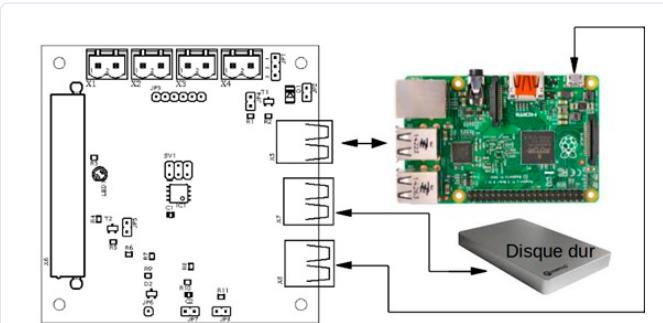


Figure 8. Connexion de la carte RPi.



Listage 1. Script Shell.

```
#!/bin/sh
#####
# IO Init Script #
# Author : Sébastien Guerreiro #
# Versions : #
# ----- #
# Mars 2019 V1.0 Création #
#####
$CDEV_GPIO=/sys/class/gpio
#-----
register()
{
#Registers the outputs
if [ $SENS="out" ]; then
#Outputs
echo "$NUM" > $CDEV_GPIO/export
echo "$SENS" > $CDEV_GPIO/gpio$NUM/direction
echo "$VALEUR" > $CDEV_GPIO/gpio$NUM/value
chmod g+w $CDEV_GPIO/gpio$NUM/value
else
#Inputs
echo "$NUM" > $CDEV_GPIO/export
echo "$SENS" > $CDEV_GPIO/gpio$NUM/direction
fi
ret=$?
if [ $ret -eq 0 ]; then
#echo_success;
else
echo_failure;
exit $ret;
fi
}
#-----
else
# echo_failure;
echo "Registering GPIO$NUM : ERROR"
exit $ret;
fi
#
unregister()
{
echo "$NUM" > /sys/class/gpio/unexport
ret=$?;
if [ $ret -eq 0 ]; then
echo_success;
else
echo_failure;
exit $ret;
fi
}
#-----
start()
{
echo "Registering GPIO--Setting Power GPIO ON"
#Output for power GPIO
NUM=27; SENS=out ; VALEUR=1
register;
}
#-----
start
exit 0
```



Listage 2. Le script d'arrêt.

```
#!/bin/sh
#####
# Reboot Poweroff Management #
# Author : Sebastien Guerreiro (www.sebelectronic.com)#
# Versions : #
# -----
# Fev 2020 V1.0 Création #
#####
case $1 in
-r)
echo 0 > /sys/class/gpio/gpio27/value
sleep 1
echo 1 > /sys/class/gpio/gpio27/value
sleep 1
/sbin/shutdownSeb $@
;;
*)
/sbin/shutdownSeb $@
;;
esac
```

intervenir avec quelques commandes Linux. Du côté du système d'exploitation, nous renommons les instructions shutdown et reboot contenues dans sbin en shutdownSeb et rebootSeb, respectivement

```
sudo mv /sbin/shutdown /sbin/
shutdownSeb
sudo mv /sbin/reboot /sbin/rebootSeb
```

Ensuite, nous créons deux scripts : shutdown et reboot, que nous plaçons dans le dossier `/sbin`. Dans le terminal, tapez :

```
nano shutdown
```

et saisissez ce qui suit (**listage 2**) dans le fichier. Enregistrez le fichier et quittez nano.

Dans le terminal, tapez :

```
sudo cp shutdown /sbin
```

Ceci copiera votre script dans `/sbin` afin que vous puissiez l'exécuter plus tard.

Ensuite, nous devons faire le script de reboot. Toujours dans le terminal, entrez :

```
nano reboot
```

et saisissez le script suivant (**listage 3**) dans ce fichier. Enregistrez le fichier et quittez nano. Ensuite, dans le terminal, tapez :

```
sudo cp reboot /sbin
```

Ceci copiera votre script de redémarrage dans `/sbin` et lui permettra d'être exécuté plus tard.

C'est tout !

Connexion des périphériques

L'utilisation d'une alimentation ATX nous permet de connecter la carte à un disque dur IDE, par exemple, ou à un lecteur de CD-ROM IDE dans mon cas. En bref, l'avantage est que vous pouvez utiliser les périphériques d'un vieux PC pour vos nouveaux projets.



Listage 3. Le script de redémarrage.

```
#!/bin/sh
#####
# Reboot Management #
# Author : Sebastien Guerreiro #
# Versions #
# -----
# Fev 2020 V1.0 Création #
#####
echo 0 > /sys/class/gpio/gpio27/value
sleep 1
echo 1 > /sys/class/gpio/gpio27/value
sleep 1
#/sbin/rebootSeb
/sbin/shutdownSeb -r now
```



PRODUITS

- **Raspberry Pi 4 2 GB (SKU 18965)**
www.elektor.fr/18965
- **Velleman VTSS220 station de soudage avec contrôle de température (SKU 19865)**
www.elektor.fr/19865

LIENS

- [1] Projet sur Elektor Labs : <https://elektormagazine.fr/labs/atx-powersupply-for-rpi>
- [2] Image Source : <https://raspberrypi.com/documentation/computers/images/GPIO-Pinout-Diagram-2.png>

La connexion se fait comme vous pouvez le voir sur la **figure 8**.

Si vous souhaitez en savoir plus sur le projet, vous pouvez visiter la page Elektor Labs [1].

(191205-04) — VF : Denis Lafourcade

Des questions, des commentaires ?

Vous avez des questions ou des commentaires sur cet article ? Contactez Elektor à l'adresse redaction@elektor.fr ou laissez un commentaire sur la page des projets à Elektor Labs.



LISTE DES COMPOSANTS

Résistances (0805, 0,1 W)

R1, R3, R6, R7, R8 = 1 k, 1%
R2, R4, R5, R9, R10 = 100 k, 1%
R11 = 10 k, 1%

Condensateurs

C1 = 100 nF, 0805
C2 = 10 nF, 0805

Semi-conducteurs

D1 = SM4007
D2 = BAT54
IC1 = ATTINY85
T1 = SI2318CDS

T2 = 2N7002

LED1 = LED, verte, 3 mm

Divers

SV1, JP7 = connecteur mâle 3 points, pas

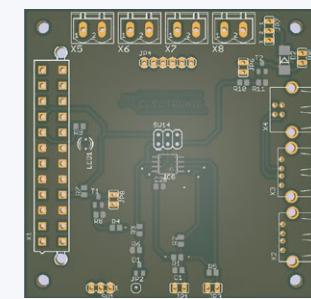
2,54 mm

JP1, JP3, JP5, J6, JP8 = connecteur mâle 2 points, pas 2,54 mm

JP2 = connecteur mâle 1 point, pas 2,54 mm

SV14 = connecteur mâle 2 rangées, 6 points, pas 2,54 mm

X5, X6, X7, X8 = connecteur MSTBVA 2.5 / 2-G-5.08 à 2 broches



X4 = Connecteur USB Type B

X1 = Connecteur Molex Mini-Fit Jr droit, 5566, 24 voies, 2 rangées

X3, X2 = Connecteur USB Type A

Circuit imprimé

Advertisement

Rejoignez la communauté Elektor



- accès à l'archive numérique depuis 1978 !
- 8x magazine imprimé Elektor
- 8x magazine numérique (PDF)
- 10 % de remise dans l'e-choppe et des offres exclusives pour les membres
- accès à plus de 5000 fichiers Gerber



Devenez membre maintenant !



www.elektormagazine.fr/membres

elektor