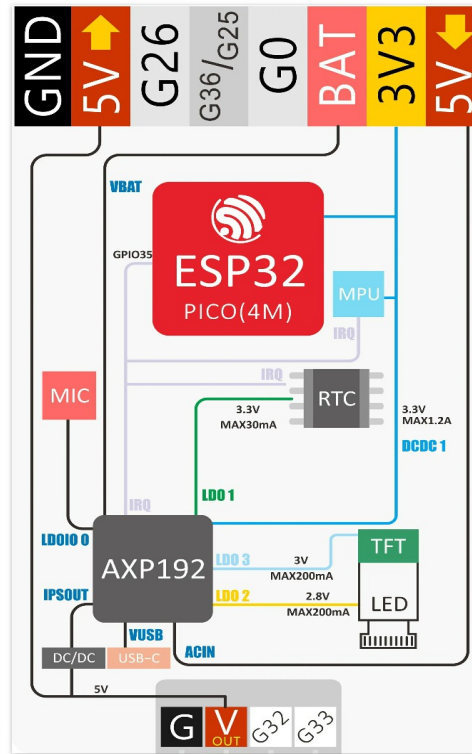
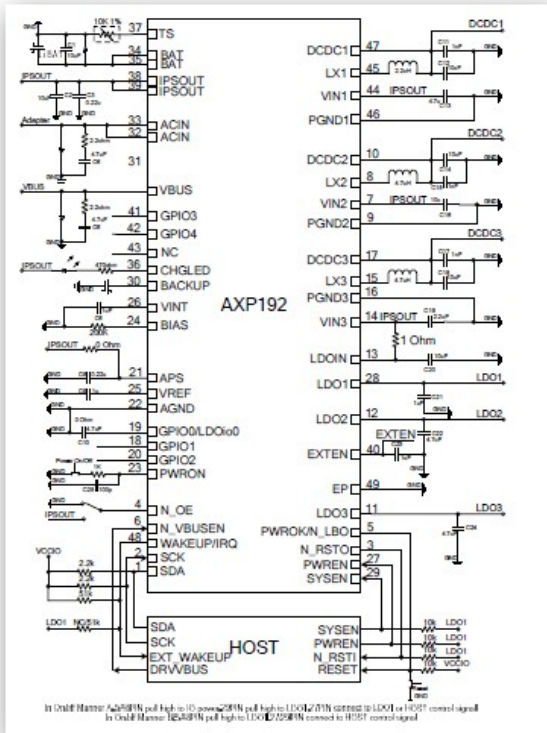


AXP 192 - PowerManagementIC

Der AXP 192 ist nur im M5Stick C / Plus und M5Stack Core 2 verbaut. Seine Aufgaben sind:

- Betriebsspannungen für die verschiedenen Bauteile erzeugen
- Zwischen 2 Eingangsspannungen und Akkubetrieb umschalten
- Den Akku laden
- Verschiedene Spannungen und Ströme messen.

AXP 192 - PowerManagementIC



Links ist die Beschaltung der verschiedenen Spannungswandler und Spannungsregler zu sehen.

In der Mitte die Verwendung innerhalb des M5Stick.

Dieser Bereich ist für uns weniger interessant.

AXP 192 - PowerManagementIC

Uns interessiert vor allem die Überwachung der Stromversorgung.

Die Methoden für den **AXP 192** finden wir im Modul **hardware/axp192**.

Vorher

`from m5import import *`
nicht vergessen!

```
>>> from m5import import *
>>> dir(axp)
['__class__', '__init__', '__module__',
 '__qualname__', '__dict__', 'addr', 'deinit',
 'powerAll', 'setLDO2State', 'clearAllIRQ', 'i2c',
 '_regChar', 'CURRENT_100MA', 'CURRENT_190MA',
 'CURRENT_280MA', 'CURRENT_360MA',
 'CURRENT_450MA', 'CURRENT_550MA',
 'CURRENT_630MA', 'CURRENT_700MA',
 'getChargeState', 'setChargeState',
 'getBatVoltage', 'getBatCurrent',
 'getVinVoltage', 'getVinCurrent',
 'getVBusVoltage', 'getVBusCurrent',
 'getTempInAXP192', 'powerOff', 'setLDO2Volt',
 'setLDO3Volt', 'setLDO3State',
 'setLcdBrightness', 'setChargeCurrent',
 'disableAllIRQ', 'enableBtnIRQ', 'btnState',
 '_read12Bit', '_read13Bit', '_read16Bit']
```

AXP 192 - PowerManagementIC

Zu Micropython AXP 192 habe ich leider keine Informationen im Internet gefunden.

Deshalb müssen wir uns ansehen, was die UIFlow-IDE zu bieten hat.

Rechts die für uns interessanten Methoden:

```
'getChargeState', 'setChargeState',  
'getBatVoltage', 'getBatCurrent',  
'getVinVoltage', 'getVinCurrent',  
'getVBusVoltage', 'getVBusCurrent',  
'getTempInAXP192', 'setChargeCurrent'
```

```
'CURRENT_100MA', 'CURRENT_190MA',  
'CURRENT_280MA', 'CURRENT_360MA',  
'CURRENT_450MA', 'CURRENT_550MA',  
'CURRENT_630MA', 'CURRENT_700MA',
```

AXP 192 - PowerManagementIC

Zur Batterieüberwachung stehen 4 Methoden zur Verfügung:

axp.getChargeState()

gibt an, ob geladen wird (True) oder nicht (False).

axp.getBatVoltage()

gibt die Batteriespannung zurück (float).

axp.getBatCurrent()

gibt den Lade- / Entladestrom zurück (float).

map_value((axp.getBatVoltage()), 3.7, 4.1, 0, 100)

Berechnet den Ladezustand in %

```
ladestatus = axp.getChargeState()
```

```
batterie_prozent = map_value((axp.getBatVoltage()), 3.7, 4.1, 0, 100)
```

```
batterie_Spannung = axp.getBatVoltage()
```

```
batterie_strom = axp.getBatCurrent()
```

AXP 192 - PowerManagementIC

Hier ein kleines Testprogramm zum Anzeigen von Batteriespannung und -strom:

```
>>> from m5import import *
```

```
>>> print(exp.getBatVoltage(), exp.getBatCurrent())
```

```
4.1415 0.0
```

```
>>>
```

```
# mit Maßeinheiten:
```

```
>>> print(exp.getBatVoltage(), 'V', exp.getBatCurrent(), 'mA')
```

```
4.0678 V 89.00001 mA
```

AXP 192 - PowerManagementIC

V_{BUS} ist die Spannung die über die USB-Buchse eingespeist wird.

Auch diese Stromversorgung kann überwacht werden. Die Anwendung entspricht der bei der Batterie:

```
>>> print(axp.getVBusVoltage(), 'V', axp.getVBusCurrent(), 'mA')
```

```
4.9844 V 168.0 mA
```

```
>>>
```

AXP 192 - PowerManagementIC

V_{in} ist die Stromversorgung die über den 5 V Eingangspin der 8-poligen Buchsenleiste zugeführt werden kann.

Da die Stromversorgung bei dieser Messung über USB erfolgt, wird nur 0.0 ausgegeben.

```
>>> print(axp.getVinVoltage(), 'V', axp.getVinCurrent(), 'mA')
```

```
0.0 V 0.0 mA
```


AXP 192 - PowerManagementIC

Die Einstellung des Ladestroms darf nur mit größter Vorsicht vorgenommen werden.

Es kann zur Zerstörung des AXP192 und des Akkus kommen. Im schlimmsten Fall kann ein Feuer ausbrechen!

Rechts die Konstanten zur Einstellung des Ladestroms.

CURRENT_100MA

CURRENT_190MA

CURRENT_280MA

CURRENT_360MA

CURRENT_450MA

CURRENT_550MA

CURRENT_630MA

CURRENT_700MA

Diese Werte sind nicht implementiert:

CURRENT_780MA

CURRENT_880MA

CURRENT_960MA

CURRENT_1000MA

CURRENT_1080MA

CURRENT_1160MA

CURRENT_1240MA

CURRENT_1320MA

AXP 192 - PowerManagementIC

Der Ladestrom wird mit der Methode **axp.setChargeCurrent(const)** eingestellt.

Wenn man mit dem Ladestrom experimentieren will, so sollte unbedingt eine automatische Ladestromreduzierung eingebaut werden, um den AXP192 zu schützen!

```
axp.setChargeCurrent(CURRENT_190MA)

# automatische Ladestrom Reduzierung

If axp.getTempInAXP192() > 70.0:
    axp.setChargeCurrent(CURRENT_100MA)

# Die maximale Chiptemperatur(tj)
# beträgt gemäß Datenblatt 130 °C.
```

AXP 192 - PowerManagementIC

Eine **Erhöhung des Ladestroms** macht bei dem eingebauten Akku keinen Sinn. Dieser hat eine Kapazität von 120mAh. Damit dürfte sein maximaler Ladestrom mit der default Einstellung von 100mA erreicht sein.

Interessant wird diese Möglichkeit, wenn man einen **externen Lilon-Akku** parallel zum eingebauten Akku anschalten will. Das ist ohne weiteres möglich. M5Stack verkauft den **M5StickC 18650C** der einen 18650 Akku enthält, der parallel zum internen Akku angeschlossen ist.

Der AXP192 ist default so eingestellt, dass er bei Spannungen $< 4,2$ Volt die Ladung beendet. Z.B. 4,192 V oder auch 4,179 V.

AXP 192 - PowerManagementIC



Links: M5StickC
18650C und eine
18650 Zelle
provisorisch am
M5Stick.

Rechts: Anschluss
der 18650 Zelle
parallel zum
internen Akku.



AXP 192 - PowerManagementIC

axp.getChargeState()

Liefert True oder False zurück, wenn geladen oder nicht geladen wird.

axp.setChargeState()

Sollte das Laden ein- bzw. Ausschalten. Tut's in der REPL nicht.

```
>>> axp.getChargeState()  
True  
>>> print(axp.getBatVoltage(), 'V', axp.getBatCurrent(), 'mA')  
3.9094 V 90.0 mA  
>>> axp.setChargeState(False)  
>>> print(axp.getBatVoltage(), 'V', axp.getBatCurrent(), 'mA')  
3.9149 V 90.0 mA
```

AXP 192 - PowerManagementIC

```
# Python Script im M5Stick
```

```
while True:
    print(axp.getChargeState())
    print(axp.getBatVoltage(), 'V', axp.getBatCurrent(), 'mA')
    axp.setChargeState(False)
    print(axp.getChargeState())
    print(axp.getBatVoltage(), 'V', axp.getBatCurrent(), 'mA')
```

```
# Die Ausgabe zeigt, das es auch im Script nicht funktioniert:
```

```
True
4.1811 V 45.5 mA
True
4.1811 V 45.5 mA
```

AXP 192 - PowerManagementIC

axp.getTempInAXP192()

Liefert die interne Temperatur des AXP192 zurück.

axp.powerOff()

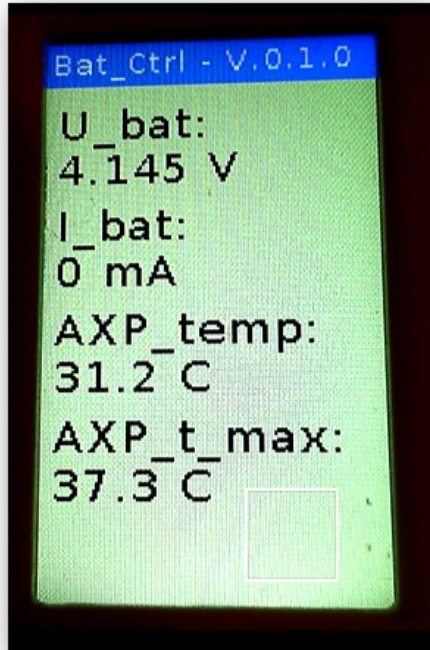
Schaltet den M5Stick aus. Einschalten manuell.

axp.setLcdBrightness(0 ... 100)

Setzt die Helligkeit des Displays.

AXP 192 - PowerManagementIC

Anzeige des Scripts **Bat_Ctrl.py**.



Das Script kann von der Seite

https://wiki.attraktor.org/Micropython_Kurs_2022

als Zip-Datei heruntergeladen werden.

Es ist eine Zip-Datei, weil das Wiki keine *.py Dateien akzeptiert.

AXP 192 - PowerManagementIC

ENDE