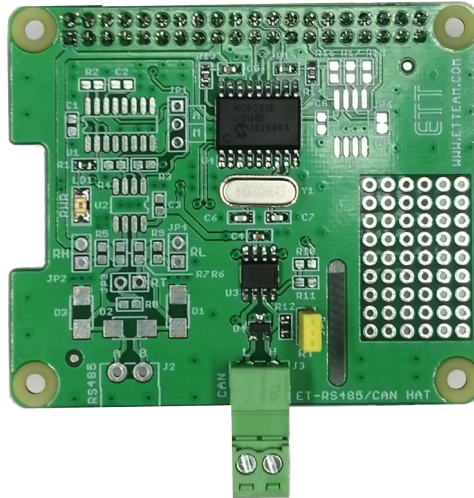


บอร์ด ET-CAN HAT

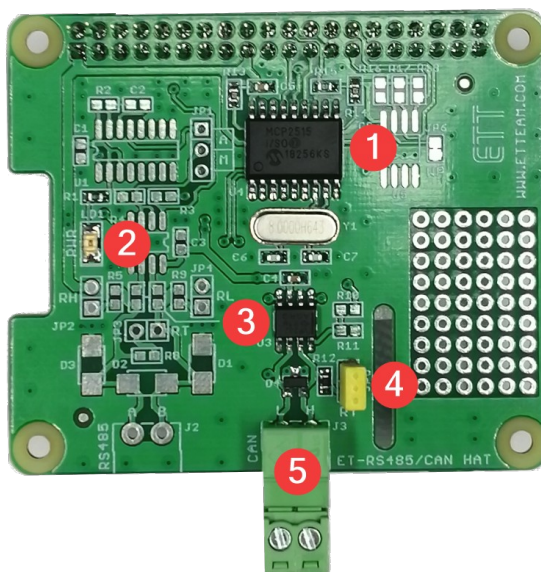


บอร์ด ET-CAN HAT เป็นบอร์ดเชื่อมต่อ CAN ที่ออกแบบมาใช้งานกับบอร์ด Raspberry Pi โดยการเชื่อมต่อผ่านทางพอร์ต SPI

คุณสมบัติของบอร์ด ET-CAN HAT

- สามารถใช้งานร่วมกับบอร์ด Raspberry Pi
- มีส่วนเชื่อมต่อ ID EEPROM (OPTION)
- ใช้ไอซี CAN Controller เบอร์ MCP2515
- ใช้ไอซี CAN Bus Transceivers เบอร์ SN65HVD232
- ใช้คอนเนคเตอร์เชื่อมต่อแบบขายาว ทำให้สามารถต่อบอร์ด HAT อื่นๆ ซ้อนได้
- ขนาดของบอร์ด 6.5 cm x 5.6 cm

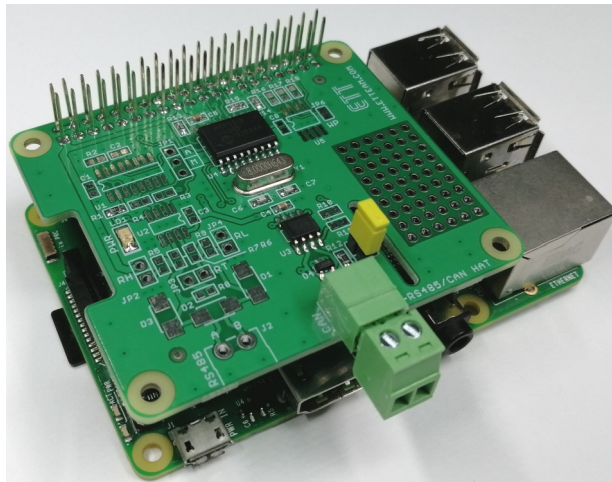
ส่วนประกอบของบอร์ด ET-CAN HAT



- หมายเลข 1 ไอซี CAN Controller เบอร์ MCP2515
- หมายเลข 2 LED แสดงสถานะเมื่อมีไฟเลี้ยงจ่ายเข้าบอร์ด
- หมายเลข 3 ไอซี CAN Bus Transceivers เบอร์ SN65HVD232
- หมายเลข 4 จั๊มเปอร์สำหรับเลือกใช้ตัวต้านทาน Termination (RT) แนะนำให้เลือกใช้เฉพาะบอร์ดต้นทางและปลายทาง
- หมายเลข 5 ขั้วต่อสัญญาณ CAN

ตัวอย่างการใช้งานบอร์ด ET-CAN HAT ร่วมกับบอร์ด Raspberry Pi

1. ตัวอย่างการใช้ ET-CAN HAT ร่วมกับบอร์ด Raspberry Pi แสดงดังรูป



2. ให้ทำการเชื่อมต่อจ่ายไฟเข้าบอร์ด Raspberry Pi และเชื่อมต่ออินเทอร์เน็ตให้เรียบร้อยจากนั้นให้ทำการอัปเดตเวอร์ชันของระบบปฏิบัติการเป็นเวอร์ชันล่าสุดก่อนโดยใช้คำสั่ง `sudo apt-get update` และ `sudo apt-get upgrade` จากนั้นให้ทำการ reboot บอร์ด Raspberry Pi โดยใช้คำสั่ง `sudo reboot`

```
pi@raspberrypi: ~  
Setting up libreoffice-script-provider-python (1:6.1.5-3+rpil+deb10u2) ...  
Setting up libreoffice-nlpsolver (0.9+LibO6.1.5-3+rpil+deb10u2) ...  
Setting up libreoffice-report-builder-bin (1:6.1.5-3+rpil+deb10u2) ...  
Setting up libreoffice (1:6.1.5-3+rpil+deb10u2) ...  
Setting up libreoffice-report-builder (1:6.1.5-3+rpil+deb10u2) ...  
Setting up libreoffice-librelogo (1:6.1.5-3+rpil+deb10u2) ...  
Processing triggers for gnome-menus (3.31.4-3) ...  
Processing triggers for libc-bin (2.28-10+rpil) ...  
Processing triggers for man-db (2.8.5-2) ...  
Processing triggers for gnome-icon-theme (3.12.0-3) ...  
Processing triggers for dbus (1.12.16-1) ...  
Processing triggers for shared-mime-info (1.10-1) ...  
Processing triggers for fontconfig (2.13.1-2) ...  
Processing triggers for desktop-file-utils (0.23-4) ...  
Processing triggers for mime-support (3.62) ...  
Processing triggers for hicolor-icon-theme (0.17-2) ...  
Processing triggers for libvlc-bin:armhf (3.0.8-0+deb10u1+rpil) ...  
pi@raspberrypi:~$ sudo reboot
```

3. จากนั้นใช้คำสั่ง `sudo nano /boot/config.txt` เพื่อแก้ไขไฟล์ config.txt ดังรูป

```
pi@raspberrypi: ~  
login as: pi  
pi@192.168.1.108's password:  
Linux raspberrypi 4.19.66-v7+ #1253 SMP Thu Aug 15 11:49:46 BST 2019 armv7l  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/*copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Thu Sep  5 11:45:32 2019  
  
SSH is enabled and the default password for the 'pi' user has not been changed.  
This is a security risk - please login as the 'pi' user and type 'passwd' to set  
a new password.  
  
pi@raspberrypi:~$ sudo nano /boot/config.txt
```

4. ให้มองหาบรรทัดที่มีข้อความ `#dtparam=spi=on` ดังรูป

```
pi@raspberrypi: ~  
GNU nano 3.2 /boot/config.txt  
  
#arm_freq=800  
  
# Uncomment some or all of these to enable the optional hardware interfaces  
#dtparam=i2c_arm=on  
#dtparam=i2s=on  
#dtparam=spi=on  
  
# Uncomment this to enable the lirc-rpi module  
#dtoverlay=lirc-rpi  
  
# Additional overlays and parameters are documented /boot/overlays/README  
  
# Enable audio (loads snd_bcm2835)  
dtparam=audio=on  
  
^G Get Help ^O Write Out ^W Where Is ^X Cut Text ^J Justify ^C Cur Pos  
^M Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

5. ทำการเปิดการใช้งานโมดูล spi โดยการเอาเครื่องหมาย # หน้าข้อความ `#dtparam=spi=on` ออก ดังรูป

```
pi@raspberrypi: ~  
GNU nano 3.2 /boot/config.txt Modified  
  
#arm_freq=800  
  
# Uncomment some or all of these to enable the optional hardware interfaces  
#dtparam=i2c_arm=on  
#dtparam=i2s=on  
dtparam=spi=on  
  
# Uncomment this to enable the lirc-rpi module  
#dtoverlay=lirc-rpi  
  
# Additional overlays and parameters are documented /boot/overlays/README  
  
# Enable audio (loads snd_bcm2835)  
dtparam=audio=on  
  
^G Get Help ^O Write Out ^W Where Is ^X Cut Text ^J Justify ^C Cur Pos  
^M Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

6. จากนั้นให้ทำการเพิ่มข้อความ `dtoverlay=mcp2515-can0,oscillator=8000000,interrupt=25` ล่างบรรทัด `dtparam=spi=on` ดังรูป

```
pi@raspberrypi: ~  
GNU nano 3.2 /boot/config.txt Modified  
  
#arm_freq=800  
  
# Uncomment some or all of these to enable the optional hardware interfaces  
#dtparam=i2c_arm=on  
#dtparam=i2s=on  
dtparam=spi=on  
dtoverlay=mcp2515-can0,oscillator=8000000,interrupt=25  
  
# Uncomment this to enable the lirc-rpi module  
#dtoverlay=lirc-rpi  
  
# Additional overlays and parameters are documented /boot/overlays/README  
  
# Enable audio (loads snd_bcm2835)  
dtparam=audio=on  
  
^G Get Help ^O Write Out ^W Where Is ^X Cut Text ^J Justify ^C Cur Pos  
^M Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

7. ทำการบันทึกไฟล์ config.txt โดยการกดปุ่ม CTRL+X และตอบ Y
8. จากนั้นให้ reboot บอร์ด Raspberry Pi โดยใช้คำสั่ง `sudo reboot`
9. ทำการตรวจสอบว่าโมดูล spi ได้ทำงานหรือยัง โดยใช้คำสั่ง `dmesg | grep -i spi` จะเห็นข้อความ ดังรูป

```
pi@raspberrypi:~$ login as: pi
pi@192.168.1.100's password:
Linux raspberrypi 4.19.66-v7+ #1253 SMP Thu Aug 15 11:49:46 BST 2019 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Sep  6 10:27:00 2019

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.

pi@raspberrypi:~$ dmesg | grep -i spi
[ 7.471490] mcp251x spi0.0 can0: MCP2515 successfully initialized.
pi@raspberrypi:~$
```

10. ทำการตรวจสอบว่าโมดูล can ได้ทำงานหรือยัง โดยใช้คำสั่ง `dmesg | grep -i can` จะเห็นข้อความ ดังรูป

```
pi@raspberrypi:~$ dmesg | grep -i spi
[ 7.471490] mcp251x spi0.0 can0: MCP2515 successfully initialized.
pi@raspberrypi:~$ dmesg | grep -i can
[ 7.421316] CAN device driver interface
[ 7.471490] mcp251x spi0.0 can0: MCP2515 successfully initialized.
pi@raspberrypi:~$
```

11. หลังจากนั้นให้ทำการติดตั้งโมดูล python-can สำหรับเขียนโปรแกรมภาษา Python เพื่อติดต่อกับ CAN โดยใช้คำสั่ง `sudo pip install python-can` โดยที่คำสั่งนี้จะติดตั้งโมดูลสำหรับ Python 2 แต่ถ้าผู้ใช้ต้องการติดตั้งโมดูลสำหรับ Python 3 ก็ให้ใช้คำสั่ง `sudo pip3 install python-can`
12. เมื่อติดตั้งโมดูล python-can เสร็จเรียบร้อยแล้ว ให้ทำการ reboot บอร์ด Raspberry Pi โดยใช้คำสั่ง `sudo reboot`
13. เมื่อทุกอย่างเรียบร้อยแล้ว จะเป็นการทดสอบรับส่งข้อมูลผ่าน CAN ซึ่งในตัวอย่างนี้ จะใช้บอร์ด Raspberry Pi จำนวน 2 บอร์ด ซึ่งได้ทำการติดตั้งบอร์ด ET-CAN HAT เรียบร้อยแล้ว โดยที่จะให้บอร์ดแรกเป็นตัวส่งข้อมูลและอีกบอร์ดเป็นตัวรับข้อมูล ซึ่งทั้ง 2 บอร์ดนี้จะเชื่อมต่อเข้าด้วยกันทางหัวต่อ CAN (CANH ต่อกับ CANH และ CANL ต่อกับ CANL) จากนั้นบอร์ดแรกตัวส่งข้อมูลให้ทำการรันโปรแกรม `can_send.py` โดยใช้คำสั่ง `sudo python can_send.py` และบอร์ดที่สองตัวรับข้อมูลให้ทำการรันโปรแกรม `can_receive.py` โดยใช้คำสั่ง `sudo python can_receive.py` จากนั้นที่บอร์ดตัวรับ จะเห็นข้อมูลที่รับเข้ามาทางพอร์ต CAN ได้ ดังรูป

```
pi@raspberrypi:~$
Timestamp: 1567743490.207508 ID: 0123 S DLC: 8 00 01 02 03 04 05 06 07 Channel: can0
Timestamp: 1567743491.209793 ID: 0123 S DLC: 8 00 01 02 03 04 05 06 07 Channel: can0
Timestamp: 1567743492.211968 ID: 0123 S DLC: 8 00 01 02 03 04 05 06 07 Channel: can0
Timestamp: 1567743493.214035 ID: 0123 S DLC: 8 00 01 02 03 04 05 06 07 Channel: can0
Timestamp: 1567743494.216116 ID: 0123 S DLC: 8 00 01 02 03 04 05 06 07 Channel: can0
Timestamp: 1567743495.218145 ID: 0123 S DLC: 8 00 01 02 03 04 05 06 07 Channel: can0
Timestamp: 1567743496.220437 ID: 0123 S DLC: 8 00 01 02 03 04 05 06 07 Channel: can0
Timestamp: 1567743497.222601 ID: 0123 S DLC: 8 00 01 02 03 04 05 06 07 Channel: can0
Timestamp: 1567743498.224652 ID: 0123 S DLC: 8 00 01 02 03 04 05 06 07 Channel: can0
Timestamp: 1567743499.226725 ID: 0123 S DLC: 8 00 01 02 03 04 05 06 07 Channel: can0
Timestamp: 1567743500.228829 ID: 0123 S DLC: 8 00 01 02 03 04 05 06 07 Channel: can0
Timestamp: 1567743501.231057 ID: 0123 S DLC: 8 00 01 02 03 04 05 06 07 Channel: can0
Timestamp: 1567743502.233284 ID: 0123 S DLC: 8 00 01 02 03 04 05 06 07 Channel: can0
Timestamp: 1567743503.235292 ID: 0123 S DLC: 8 00 01 02 03 04 05 06 07 Channel: can0
Timestamp: 1567743504.237287 ID: 0123 S DLC: 8 00 01 02 03 04 05 06 07 Channel: can0
```

รายละเอียดของโปรแกรม can_send.py

```
import os
import can
import time

os.system("sudo ip link set can0 type can bitrate 500000")
os.system("sudo ifconfig can0 up")

can0 = can.interface.Bus(channel = 'can0', bustype = 'socketcan_ctypes')# socketcan_native

print("Test CAN send")
print("Press CTRL-C to exit")

try:
    while True:
        msg = can.Message(arbitration_id=0x123, data=[0, 1, 2, 3, 4, 5, 6, 7], extended_id=False)
        can0.send(msg)
        time.sleep(1)

except KeyboardInterrupt:
    os.system("sudo ifconfig can0 down")
```

รายละเอียดของโปรแกรม can_receive.py

```
import os
import can
import time

os.system("sudo ip link set can0 type can bitrate 500000")
os.system("sudo ifconfig can0 up")

can0 = can.interface.Bus(channel = 'can0', bustype = 'socketcan_ctypes')# socketcan_native

print("Test CAN receive")
print("Press CTRL-C to exit")

try:
    while True:
        msg = can0.recv(10.0)
        print (msg)
        if msg is None:
            print("Timeout occurred, no message.")

except KeyboardInterrupt:
    os.system("sudo ifconfig can0 down")
```