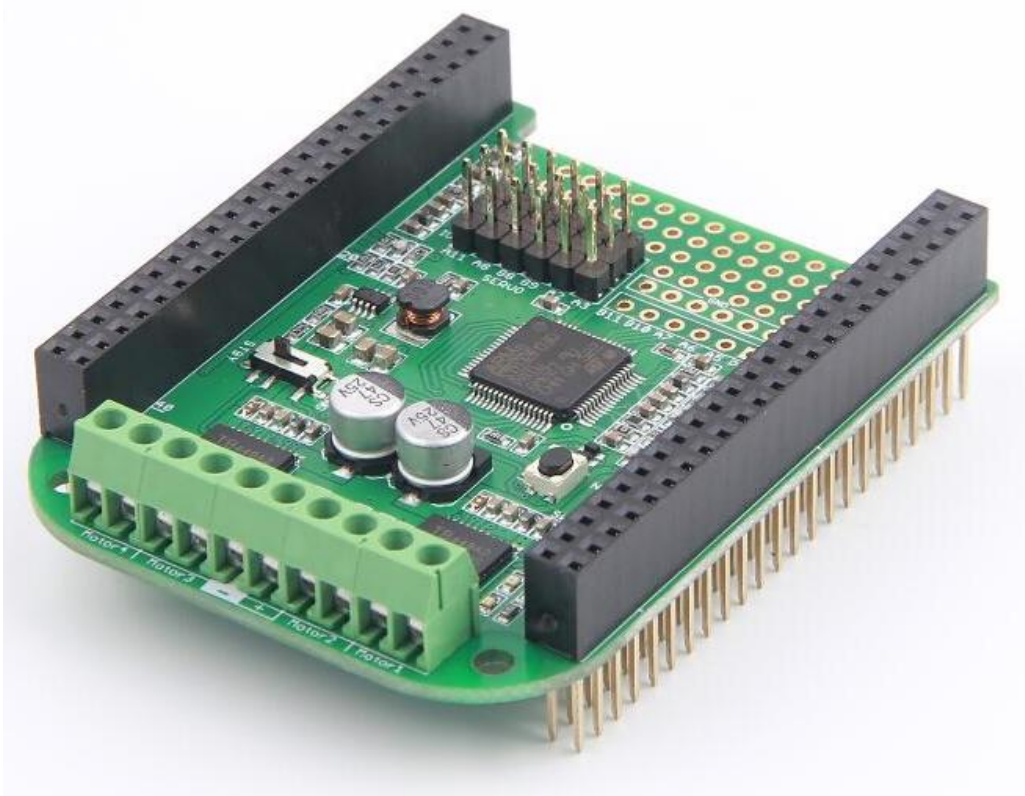


Motor Bridge Cape v1.0



The Motor Bridge Cape features bi-directional motor control using two TB6612FNG integrated dual H-bridge, so it can control two stepper motors or four brushed DC motors with 6 ~15V DC power and about 1A current draw per motor. The cape provides 5V regulated power to BBG or BBB with a max input voltage of 15V. It also has six servo control interfaces and six expand I/O. All the features are provided by the on board STM32F0 coprocessor. The MCU can communicate with BeagleBone by I²C or UART interface.

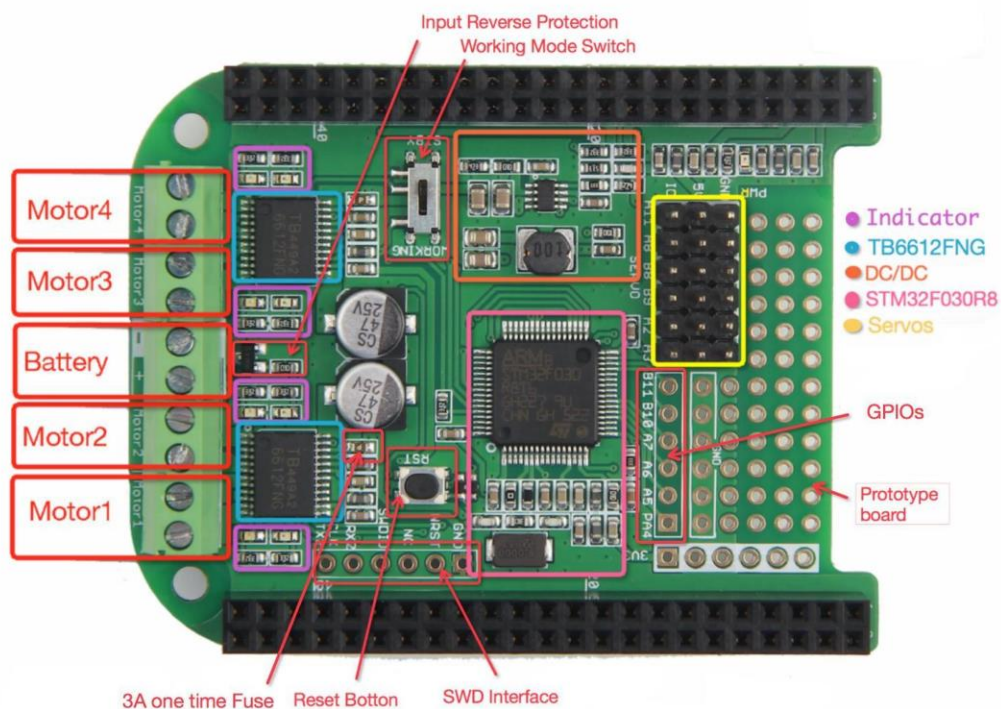
Features

- Can drive 4 DC Motors or 2 Stepper Motors
- Can drive 6 Servos
- Mbed Platform
- STM32F0 coprocessor
- Two TB6612FNG
- 6 expand I/Os
- Communicate with BBG by I²C or UART interface

Specifications

- Battery Input Voltage: 6~15V
- H-bridge Working Voltage: 6~15V
- DC/DC 5V output current: 2A max
- 3V3 output current: 350mA max
- 4 H-bridge driver, each rated current: 1.2A, peak current: 3.2A
- 6 Servo driver, working voltage: 5V, total current is not more than 1.5A
- Input reverse connect protection
- Over current protection: 3A one-off quick fuse

Hardware Overview



- **3A one time Fuse:** Over-current protection
- **Reset Button:** Reset the coprocessor
- **SWD Interface:** Debug interface
- **Prototype board:** Extend other circuits
- **GPIOs:** Extend other circuits
- **Servos:** Servo motor interface
- **STM32F030R8:** The coprocessor
- **DC/DC:** Voltage conversion circuit
- **TB6612FNG:** H-bridge IC
- **Indicator:** Indicating lamp for motor direction
- **Working Mode Switch:** Standby or Working
- **Input Reverse Protection:** Protect circuit
- **Motor4/Motor3:** Drive 2 DC motors or 1 Stepper motor
- **Motor2/Motor1:** Drive 2 DC motors or 1 Stepper motor
- **Battery:** Supply power for motor

Getting Started

After this section, I will show you how to use the Motor Bridge Cape on BGG. Before you get started, please download the Motor Bridge Cape Driver library code from the [Github](#).

To use the Motor Bridge Cape Library simply add the MotorBridge.py file to your project. And import the python file to your project, and create a object of the Motor Bridge Cape.

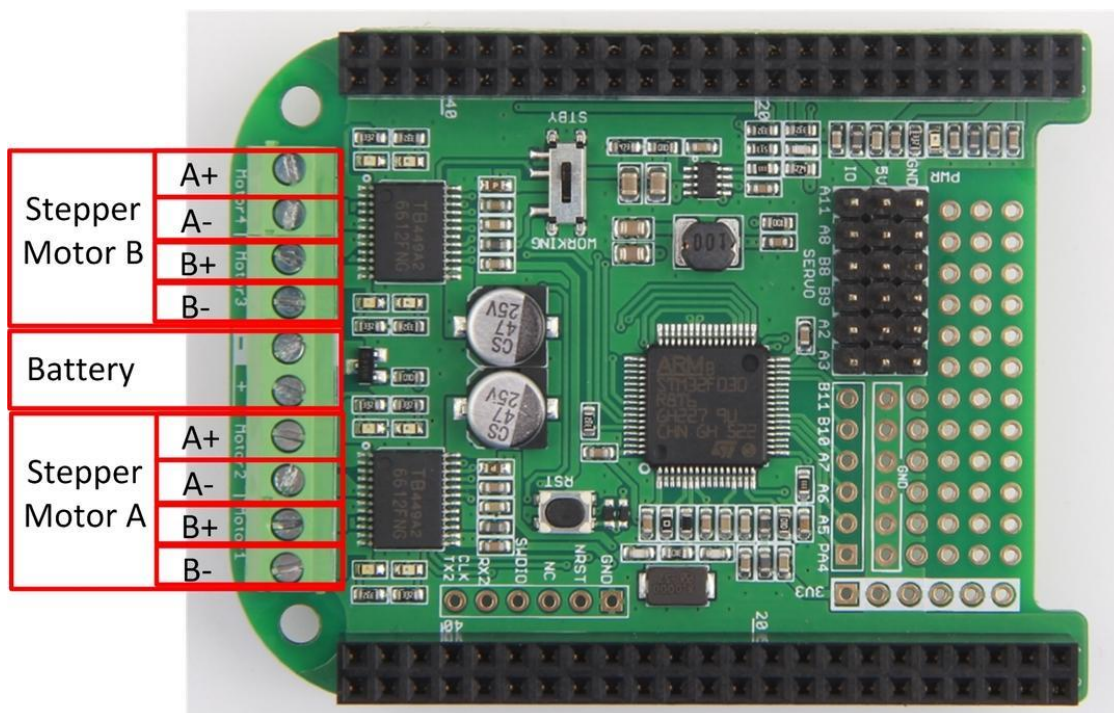
```
1 import MotorBridge
2 motor = MotorBridge.MotorBridgeCape ()
```

Install the dependencies

```
1 sudo apt-get update
2 sudo apt-get install build-essential python-pip python-dev python-smbus git
3 sudo pip install Adafruit-GPIO
```

Stepper Motor

The Stepper Motor Interface of Motor Bridge Cape just as the picture below shows.



The functions of Stepper Motor

Here is the brief description about the function of stepper motor.

StepperMotorAInit()

Description: Init the Stepper Motor A port.

StepperMotorAMove(MoveSteps,StepDelayTime)

Description: Drive the Stepper Motor A.

MoveSteps: How many steps the stepper motor will move. **Positive** stands for clock wise direction. **Negative** stands for counter-clock wise direction.

StepDelayTime: The dead time for every step. unit: us.

StepperMotorBInit()

Description: Init the Stepper Motor B port.

StepperMotorBMove(MoveSteps,StepDelayTime)

Description: Drive the Stepper Motor B.

MoveSteps: How many steps the stepper motor will move. **Positive** stands for clock wise direction. **Negative** stands for counter-clock wise direction.

StepDelayTime: The dead time for every step. unit: us.

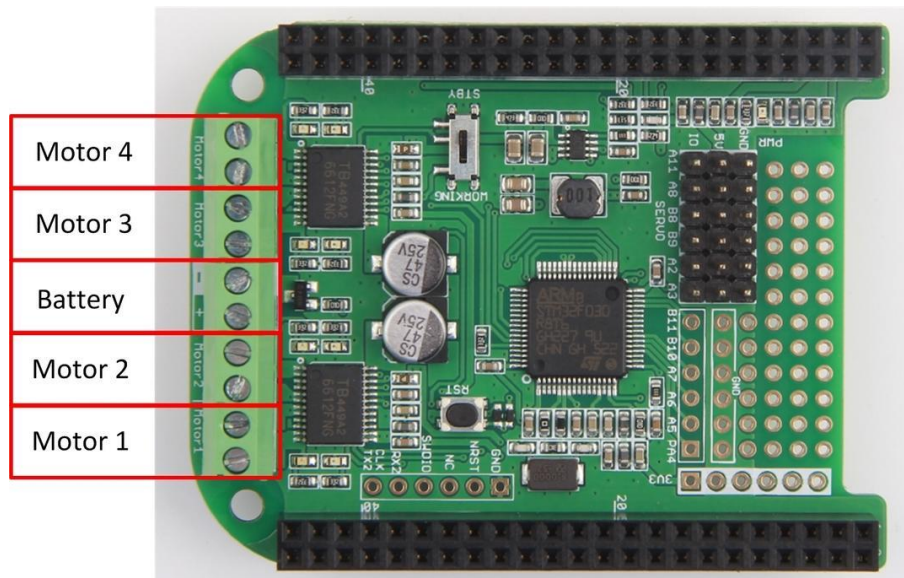
Example of Stepper Motor

Copy the following code to your project and save it as a python file.

```
1 import MotorBridge
2 import time
3
4 def StepperMotorATest():
5     print 'Hello From MotorBridge'
6     motor.StepperMotorAInit()
7     motor.StepperMotorAMove(1000,1000) # 1000 steppers 1000us every step
8     time.sleep(1)
9     motor.StepperMotorAMove(-1000,1000) #1000 steppers 1000us every step
10    time.sleep(1)
11
12 def StepperMotorBTest():
13    print 'Hello From MotorBridge'
14    motor.StepperMotorBInit()
15    motor.StepperMotorBMove(1000,1000) # 1000steppers 1000us every step
16    time.sleep(1)
17    motor.StepperMotorBMove(-1000,1000) # 1000 steppers 1000us every step
18    time.sleep(1)
19
20
21 if __name__=="__main__":
22    motor = MotorBridge.MotorBridgeCape()
23    StepperMotorATest()
24    StepperMotorBTest()
```

DC Motor

The DC Motor Interface of Motor Bridge Cape just as the picture below shows.



The functions of DC Motor

Here is the brief description about the functions of DC motor.

DCMotorInit(MotorName,Frequency)

Description: Init the DC Motor and set the frequency.

MotorName: 1 ~ 4 stands for Motor1 ~ Motor4.

Frequency: Set the frequency of the DC motor.

Note

If you change the DC motor frequency, the other DC Motors's frequency will change too.

DCMotorMove(MotorName,Direction,PWMDuty)

Description: Drive the DC Motor. Set the direction and PWMDuty.

MotorName: 1 ~ 4 stands for Motor1 ~ Motor4.

Direction: 1 stands for clock wise. 2 stands for counter-clock wise. 3 Stop the motor.

PWMDuty: 0 ~ 100 stands for 0%~ 100% of the pwm duty.

motor.DCMotorStop(MotorName)

Description: Stop the DC Motor.

MotorName: 1 ~ 4 stands for Motor1 ~ Motor4.

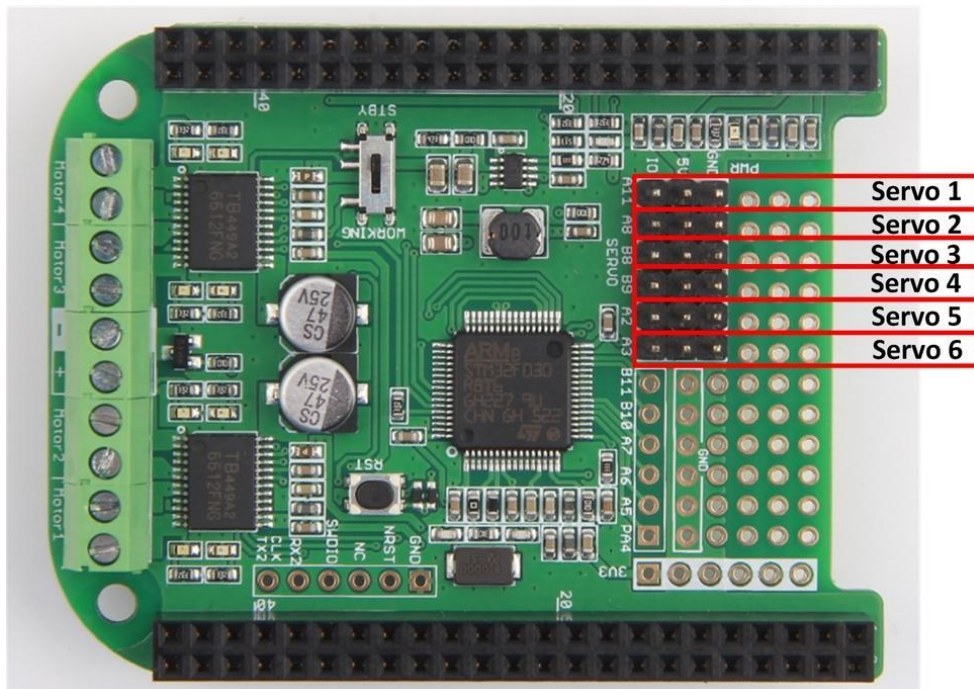
Example of DC Motor

Copy the following code to your project and save it as a python file.

```
1 import MotorBridge
2 import time
3
4 MotorName      = 1
5 ClockWise      = 1
6 CounterClockWise = 2
7 PwmDuty        = 90
8 Frequency      = 1000
9
10 if __name__ == "__main__":
11     motor = MotorBridge.MotorBridgeCape ()
12     motor.DCMotorInit (MotorName, Frequency)
13     while True:
14         motor.DCMotorMove (MotorName, ClockWise, PwmDuty)
15         time.sleep (2)
16         motor.DCMotorMove (MotorName, CounterClockWise, PwmDuty)
17         time.sleep (2)
18         print "hello"
19         motor.DCMotorStop (MotorName)
20         time.sleep (2)
```

Servo

The Servo Interface of Motor Bridge Cape just as the picture below shows.



The functions of Servo

Here is the brief description about the functions of Servo.

Servoinit(ServoName,Frequency)

Description: Init the Servo and set the frequency.

ServoName: 1 ~ 6 stands for Servo1 ~ Servo6.

Frequency: Set the frequency of the Servo, the default value is 50 Hz.

ServoMoveAngle(ServoName,Angle)

Description: Drive the Servo. Set the Servo angle.

ServoName: 1 ~ 6 stands for Servo1 ~ Servo6.

Angle: 0 ~ 180 stands for 0 degrees to 180 degrees.

Example of Servo

Copy the following code to your project and save it as a python file.

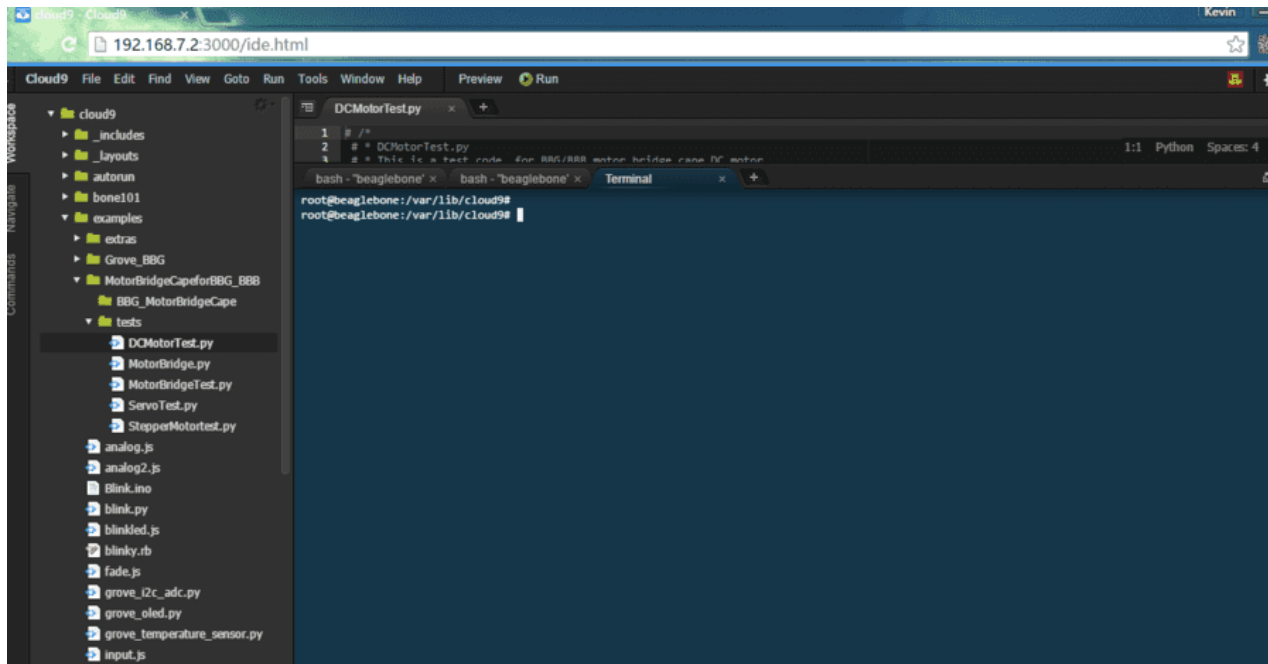
```
1 import MotorBridge
2 import time
3 ServoName = 2
4 Frequency = 50
5 Angle1 = 20
6 Angle2 = 160
7
8 if __name__ == "__main__":
9     motor = MotorBridge.MotorBridgeCape()
10    motor.Servoinit(ServoName, Frequency)
11
12    while True:
13        print 'Servo Test'
14        motor.ServoMoveAngle(ServoName, Angle1)
15        time.sleep(2)
16        motor.ServoMoveAngle(ServoName, Angle2)
17        time.sleep(2)
```

Upgrade the Firmware

If there is something wrong with your cape, try to upgrade it. This section show how to upgrade the Motor Bridge Cape Firmware using BeagleBone Green. It also works on BBGW and BBB.

1.Insert the Motor Bridge Cape to BBG/BBGW/BBB, and connect BBG to your computer via USB cable.

2.Let your BBG connect to the internet, and access it via SSH as the picture below shows.



3. Download the code from Github by execute following commands

```
1 git clone https://github.com/Seeed-Studio/MotorBridgeCapeFirmware
```

```
root@beaglebone:/var/lib/cloud9#
root@beaglebone:/var/lib/cloud9# git clone https://github.com/Seeed-Studio/MotorBridgeCapeFirmware
Cloning into 'MotorBridgeCapeFirmware'...
remote: Counting objects: 21, done.
remote: Compressing objects: 100% (19/19), done.
remote: Total 21 (delta 7), reused 11 (delta 2), pack-reused 0
Unpacking objects: 100% (21/21), done.
root@beaglebone:/var/lib/cloud9#
```

4. Navigate to "MotorBridgeCapeFirmware" and execute "make flash" command to flash the firmware.

```
1 cd MotorBridgeCapeFirmware/ && make flash
```

After a few seconds, you can see "Verification OK" information from the terminal

5. Check if the firmware is flash well by execute following command

```
1 i2cdetect -y -r 1
```



```
root@beaglebone:/var/lib/cloud9/MotorBridgeCapeFirmware# i2cdetect -y -r 1
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  4b  --  --  --  --  --
50:  --  --  --  --  UU  UU  UU  UU  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
root@beaglebone:/var/lib/cloud9/MotorBridgeCapeFirmware#
```

If you find the I2C address 0x4b, it means you have already refreshed the firmware successfully.

Known issues

Can't find the I2C address

Q: I have already updated the firmware, but can't detect the i2c address?

A: Be sure that the P9_23 Pin is High, since the P9_23 is connect to Reset pin of STM32, I set the P9_23 pin to high level in the initialization function of MotorBridgeCape class.

Can't update the firmware

Q: When I update the firmware, the error information says that can't find the UART2?

A: You should enable the BB-UART2, since the BBG flash the firmware to Motor Bridge Cape by UART2..

```
vi /boot/uEnv.txt
```

Then uncomment "cape_enable=capemgr.enable_partno=BB-UART2". Save and exit the editor, at last reboot your board.

The Cape is still not working

Q: I have already updated the firmware successfully and can detect the I2C address, but why I still can't get the cape working?

A: Please mention that there's a working mode switch on the board, be sure the switch is turn to working. If you still have other questions, please come to our forum.

fail to run make flash

Q: Can't run make flash with error code "Can't init. Ensure BOOT0=1 BOOT1=0, and reset Device"

A: Please follow below steps to solve the issue.

- Step 1: Run `sudo nano /boot/uEnv.txt` and then add below 2 lines at bottom of the uEnv.txt.

```
1 uboot_overlay_addr0=/lib/firmware/BB-UART2-00A0.dtbo
2 enable_uboot_cape_universal=1
```

- Step 2. Restart the BBGW
- Step 3. Run the Sudo make flash again.

Resources

- [Schematic in PDF](#)
- [PCB in Eagle File](#)
- [Schematic in Eagle File](#)
- [Motor Bridge Cape Library](#)
- [Motor Bridge Cape Firmware Source Code](#)

Tech Support

Please submit any technical issue into our [forum](#) or drop mail to techsupport@seeed.cc.