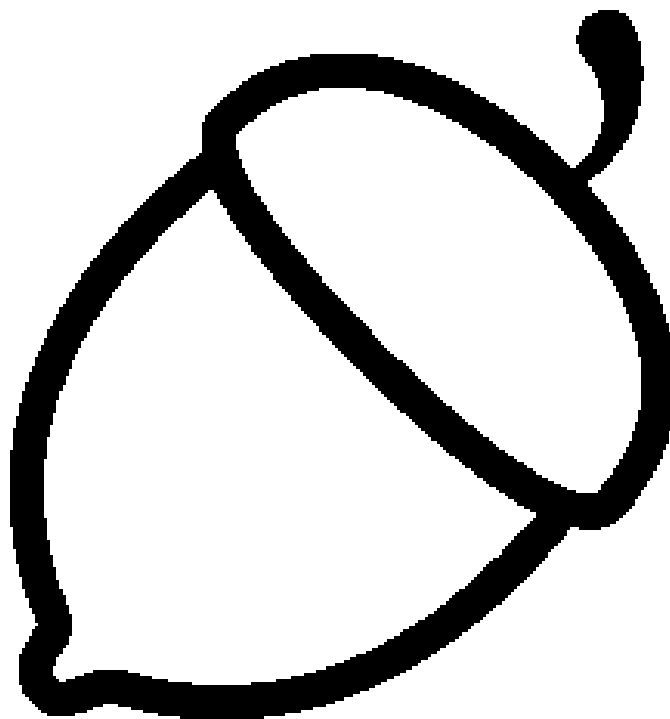


*NutsBoard*

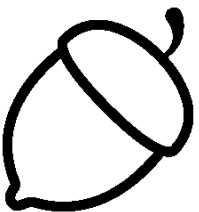


**NutsBoard**

Quick Start Guide

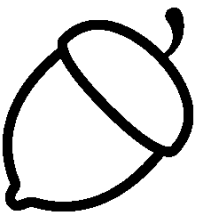
# Chapter

1. Walnut I/O List
2. Writing SDCard Images
  - [Linux](#)
  - [Windows](#)
3. How to debug
4. How to boot up
5. Q&A

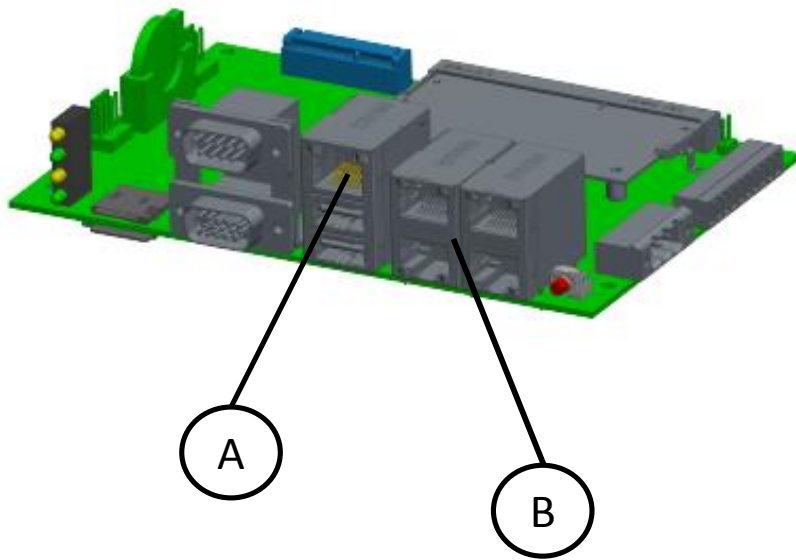


# Walnut I/O List

- [Wired networking](#)
- [USB](#)
- [UART](#)
- [SD Card](#)
- [LED lights](#)
- [Reset System](#)
- [Power input connector](#)
- [Digital In/Out](#)
- [204 pin SO-DIMM connector](#)
- [CAN BUS/Debug Console](#)
- [RTC battery slot](#)
- [SIM Card slot](#)
- [Jumper of boot selection](#)
- [Mini PCI Express](#)

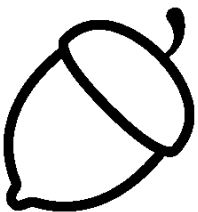


# Wired networking



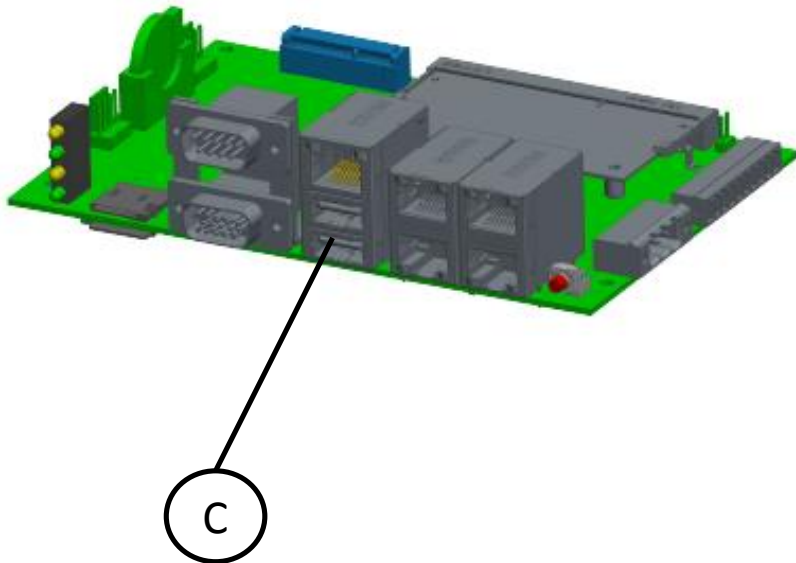
A): 1 Wide area network (WAN)  
Gigabit Port. Support , 10/100/1000  
Speed.

B): 4 Local Area Network (LAN)  
Megabit ports. Support 10/100 Speed.

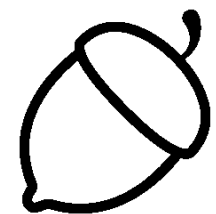


**NutsBoard**

# USB

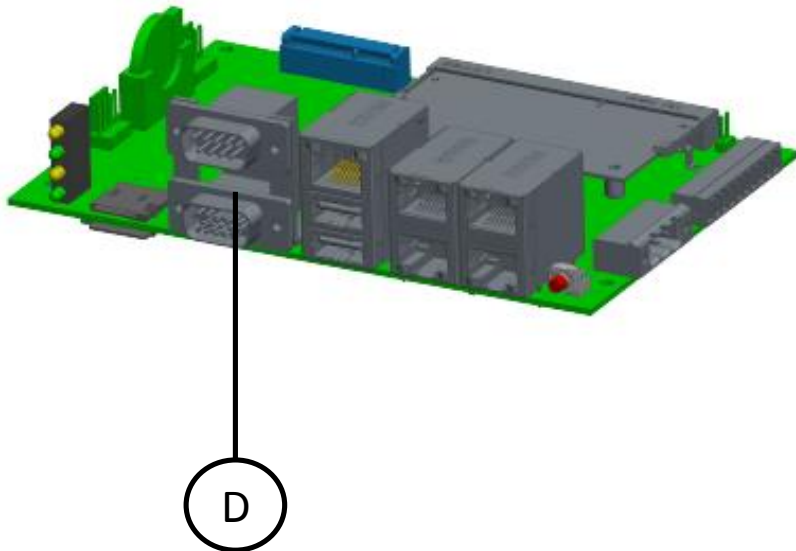


C): 2 USB 2.0 port



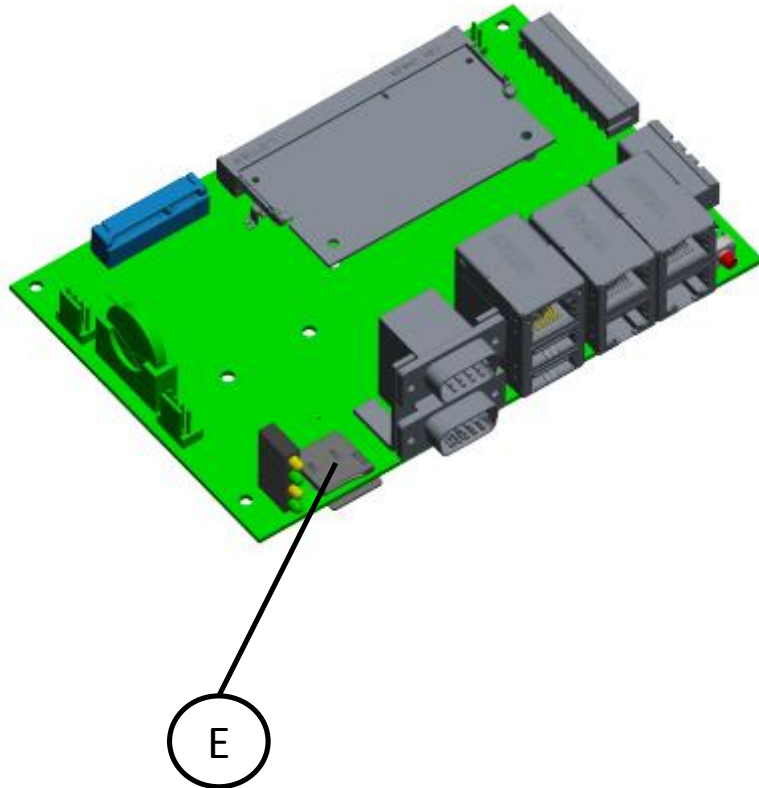
**NutsBoard**

# UART

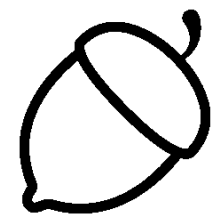


D): 2 DB9 UART connectors.

# SD Card

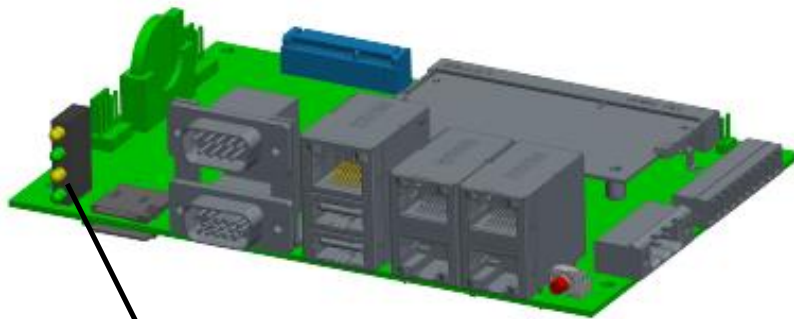


E): Micro SD card slot  
Maximum size: 32GB



**NutsBoard**

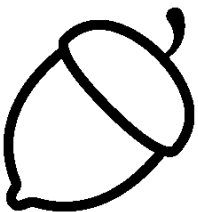
# LED lights



F

F): LED lights including:

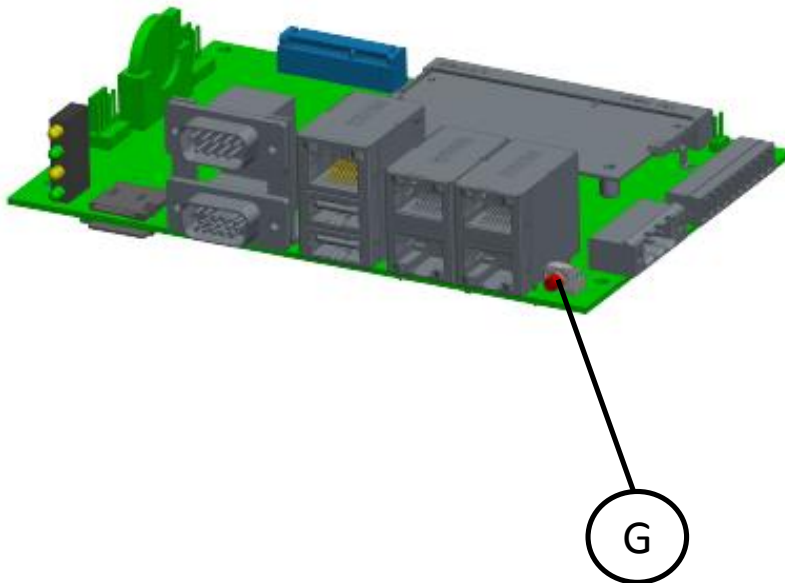
1. Red
2. Yellow
3. Blue
4. Green



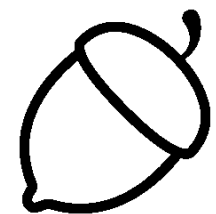
**NutsBoard**



# Reset button

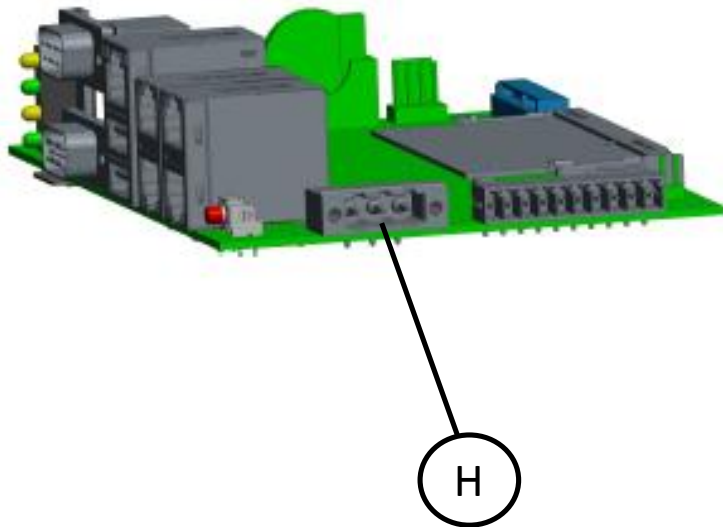


G): Hardware reset button

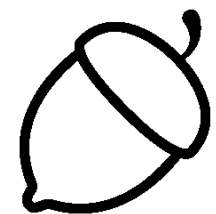


**NutsBoard**

# Power input connector

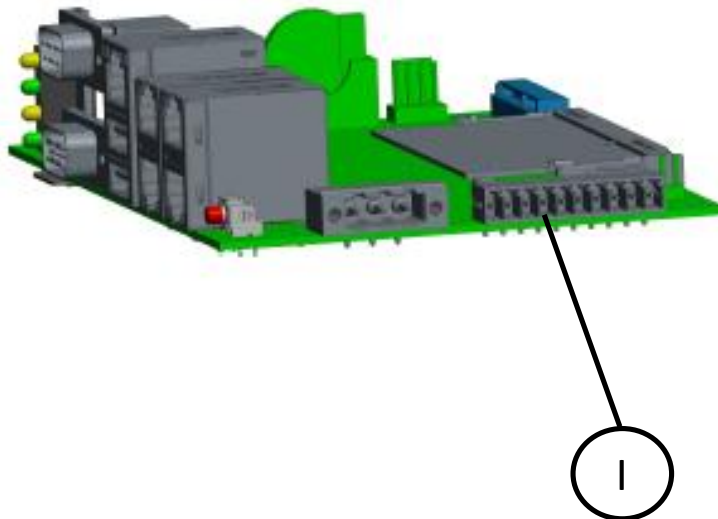


H): 12VDC POWER input connector



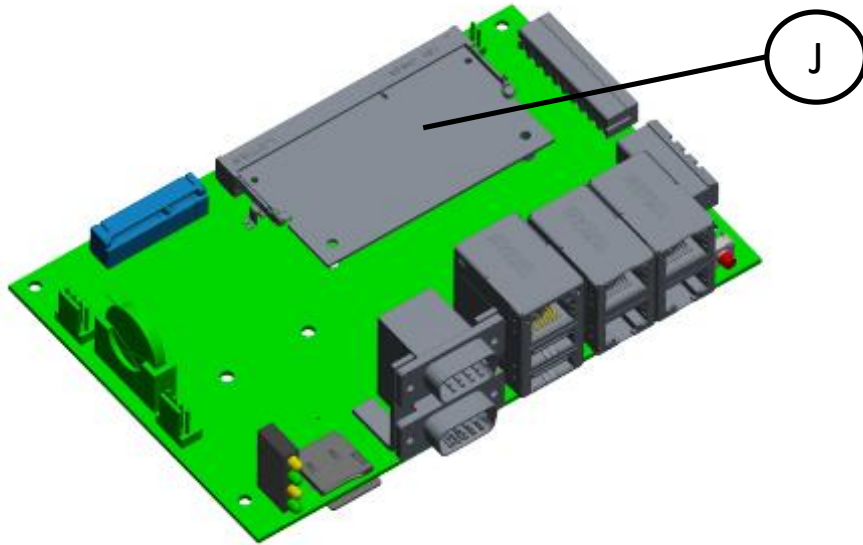
**NutsBoard**

# Digital In/Out



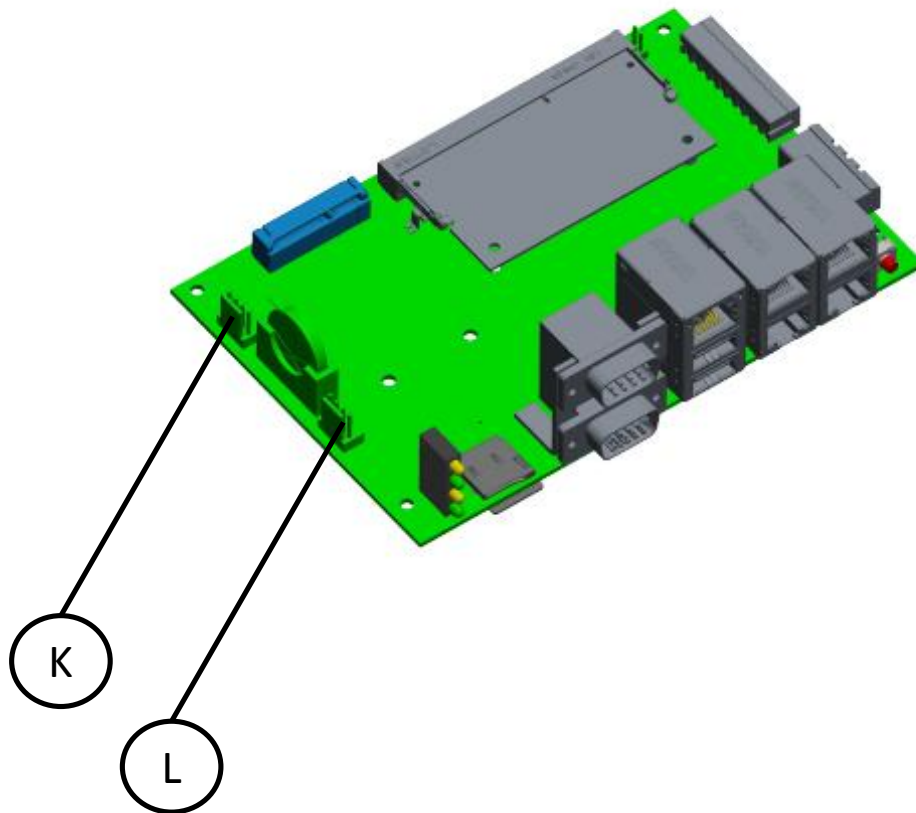
I): Digital In/Out connector

# 204 pin SO-DIMM connector



J): SO-DIMM connector with Almond SoM compatible

# CAN bus/ Debug console

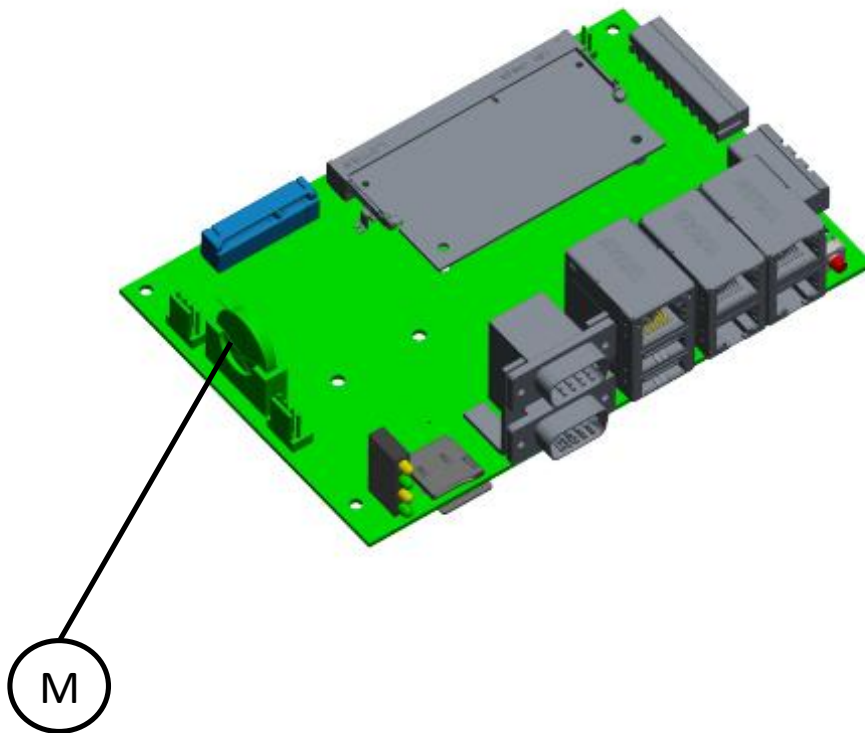


K): CAN bus connector

L): Debug console connector

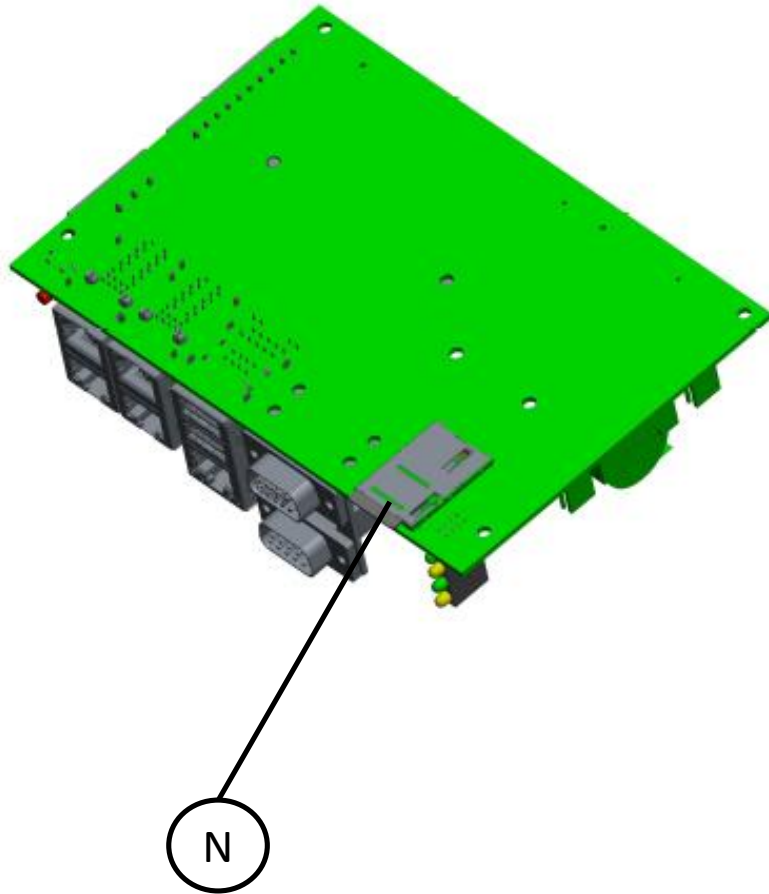


# RTC battery slot

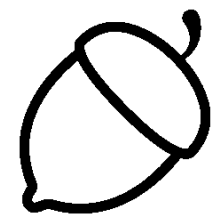


M): RTC battery slot

# SIM Card slot

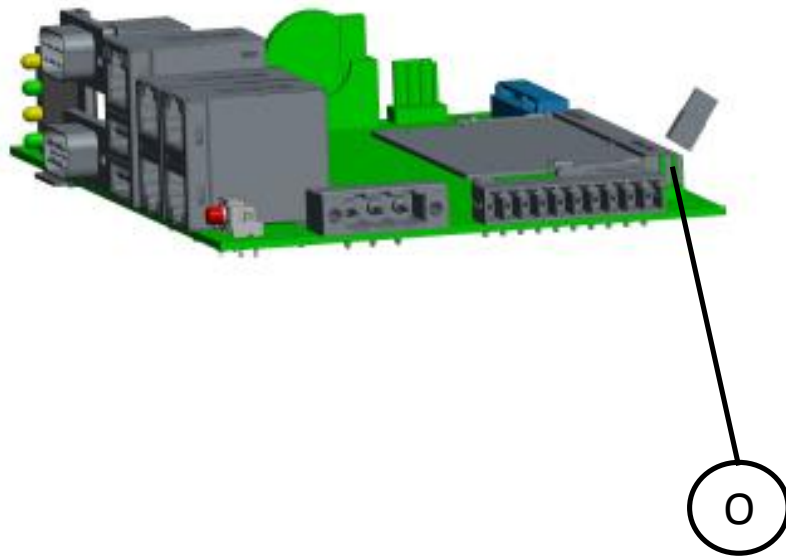


N): Subscriber Identity Module (SIM) slot

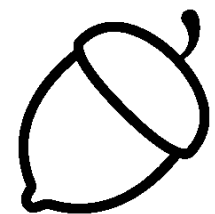


**NutsBoard**

# Jumper of boot selection



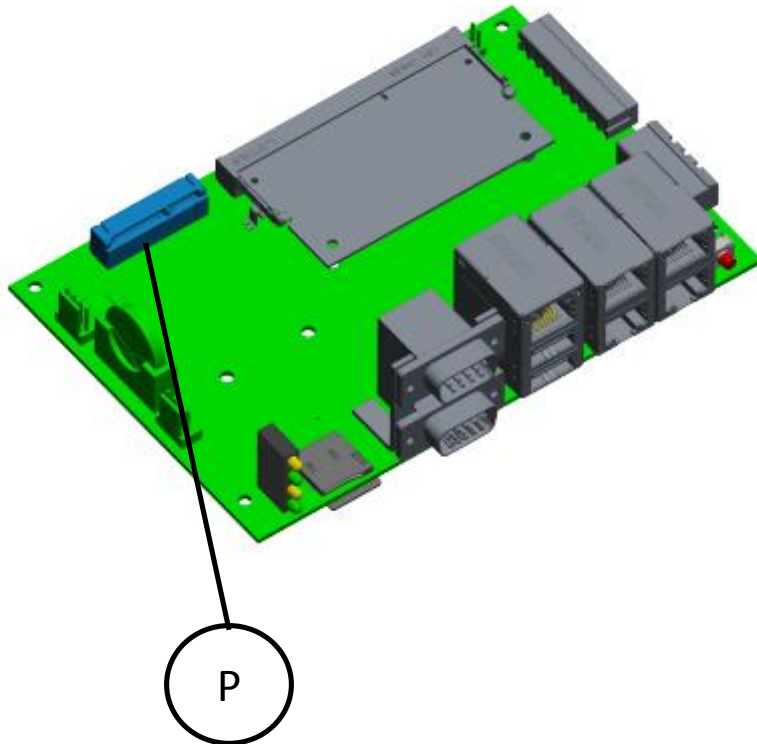
0): SD boot with jumper.  
NAND boot without jumper.



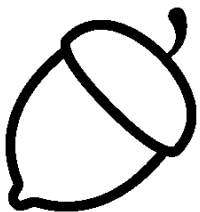
**NutsBoard**



# Mini PCI Express



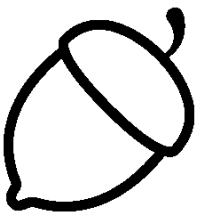
P): mPCIe connector with 3G/4G communication modules.



**NutsBoard**

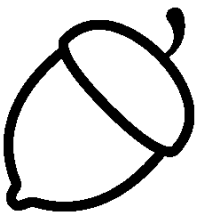
# Writing SDCard Images

- Linux
- Windows



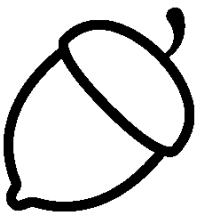
# Writing SDCard Images

- **Preparing the bootable microSD card for your Nutsboard**
  - The microSD card that is created below will contain the Nutsboard operating system. A large number of demo runtime images are available.
- **Procedures to get you started**
- a) Download your preferred Nutsboard runtime image.
- <https://goo.gl/kBcdij>
- b) Extract the file that you just downloaded
  - click on the file “Through your browser to download”.
  - The extracted files will contain a file ending in *.zip*
  - Please uncompress this file.



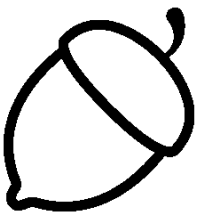
# Writing SDCard Images

- Linux
  - This paragraph explains how to create a SD card using Linux desktop or notebook. The SD card can be made using a standard terminal.
  - “`dd if=*.img of=/dev/sdd bs=1M`”
  - replace \*.img with the full name of the SD card image and replace /dev/sdd with your SD card device".



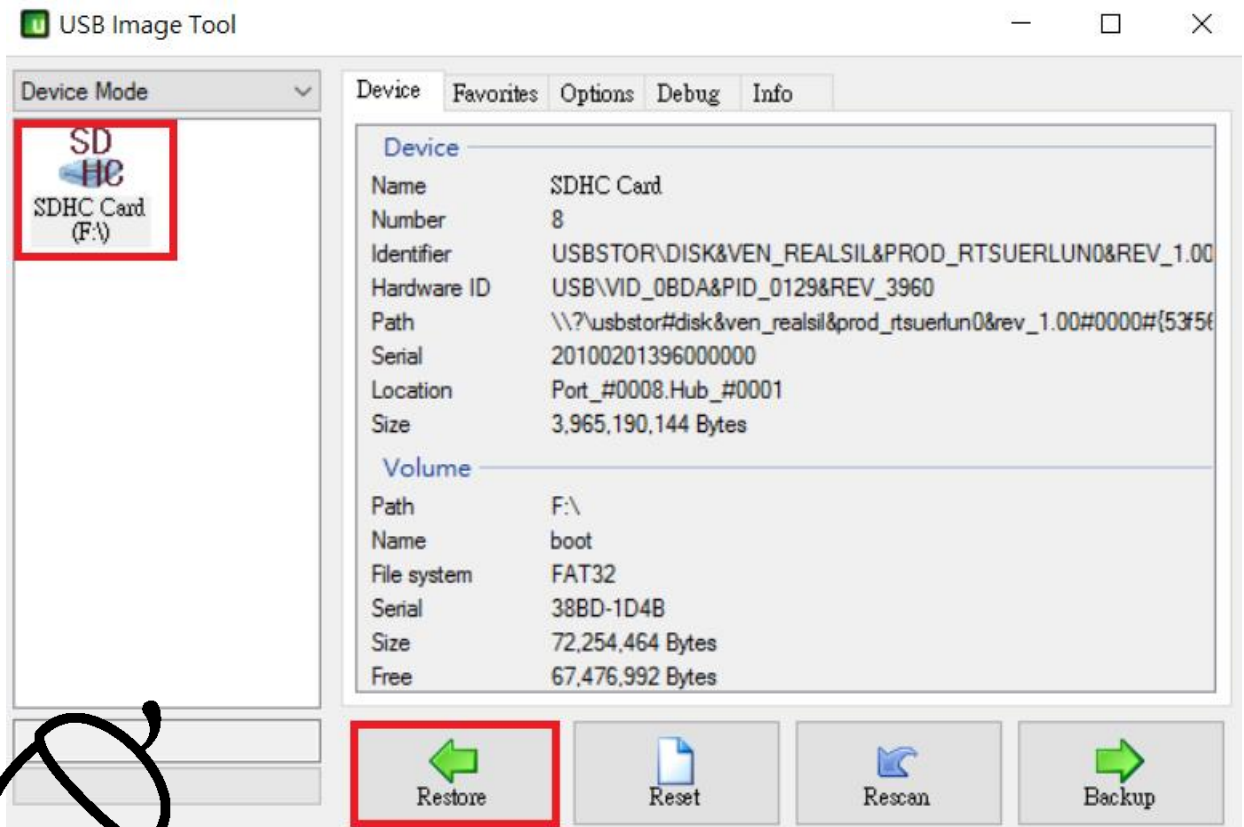
# Writing SDCard Images

- Windows
  - For Windows we suggest writing sdcard images with this tool “usbit.zip”, Please Download file.
  - [https://mega.nz/#!NsUkha4Q!NlyvA-bSPDTfGFNg8IhSAbv7t9IGexmQlFDV3qt\\_b8](https://mega.nz/#!NsUkha4Q!NlyvA-bSPDTfGFNg8IhSAbv7t9IGexmQlFDV3qt_b8)
  - click on the file “Through your browser to download”.



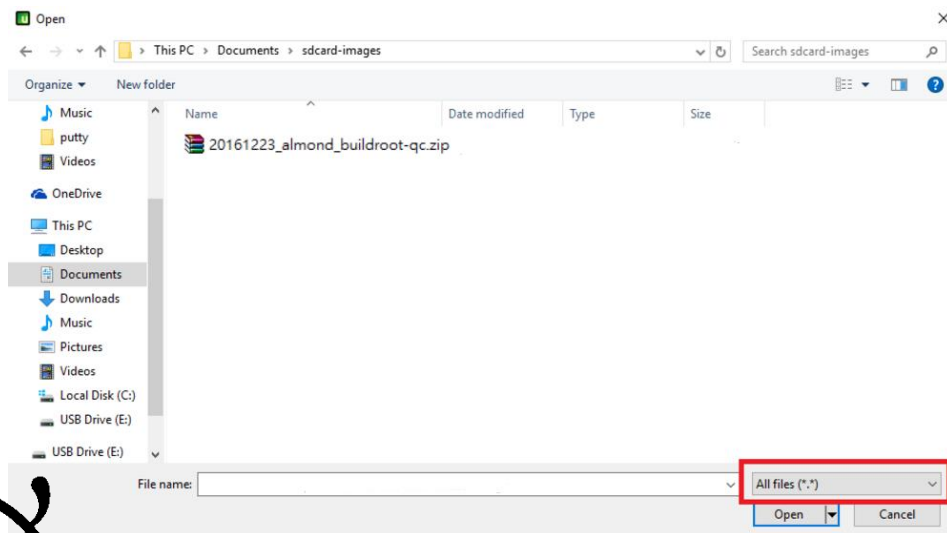
# Writing SDCard Images

- When you open the application, you should see on the left your sdcard. Then you'll need to press the "Restore" button to write a sdcard image to it.



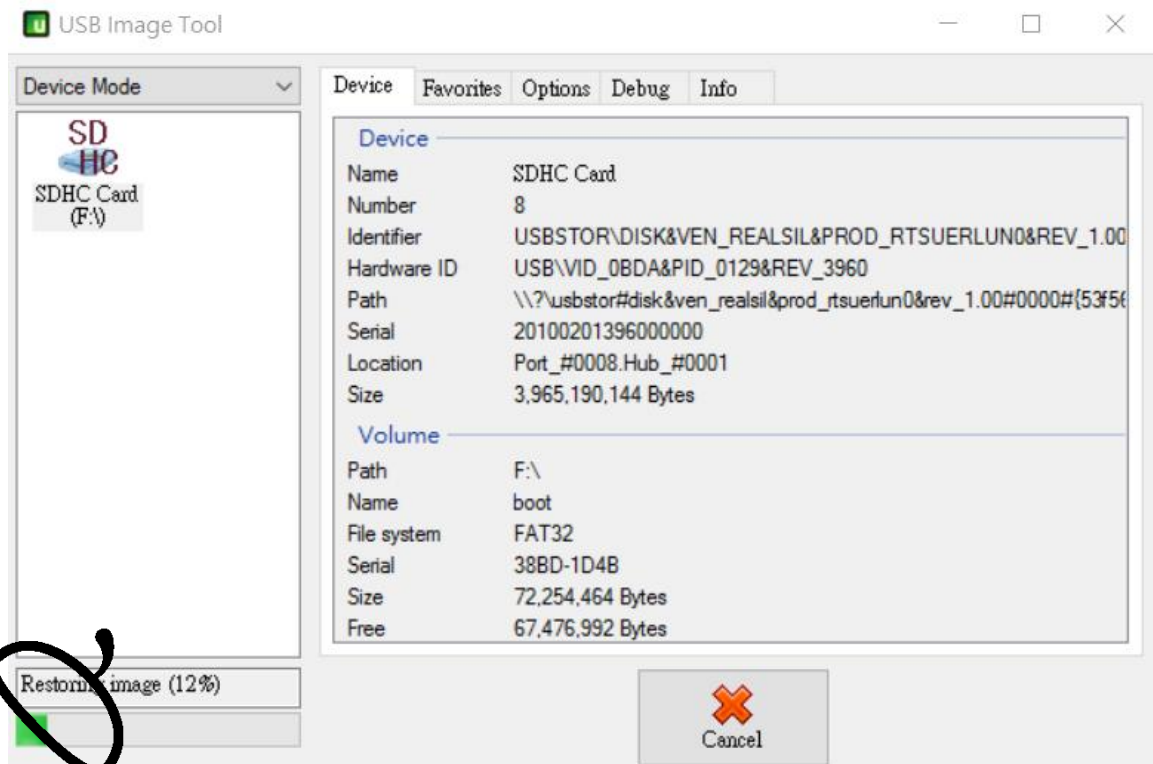
# Writing SDCard Images

- You will need to assign sdcard image location on disk, then select “All Files (\*.\*)” from the drop down on the bottom right – this behavior will help us to find .sdcard or .sdcard.gz files. In the end, select the image and click “Open”:



# Writing SDCard Images

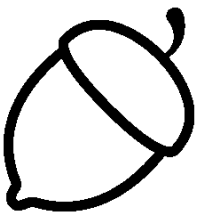
- Once you hit the Open button, the sdcard writing should start. You'll see a status bar on the bottom Bottom left corner as it progresses. Once it's at 100% you're good to go:





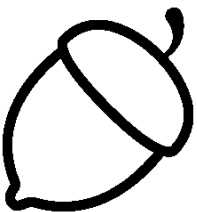
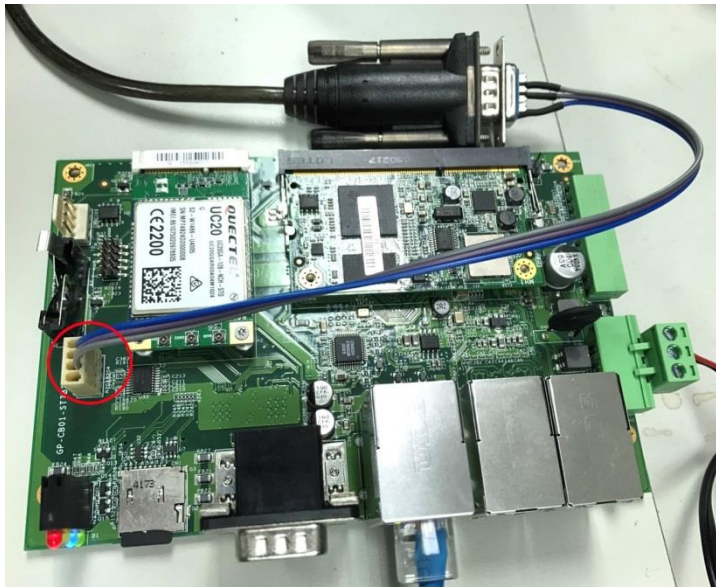
# How to debug

- Linux
- Windows



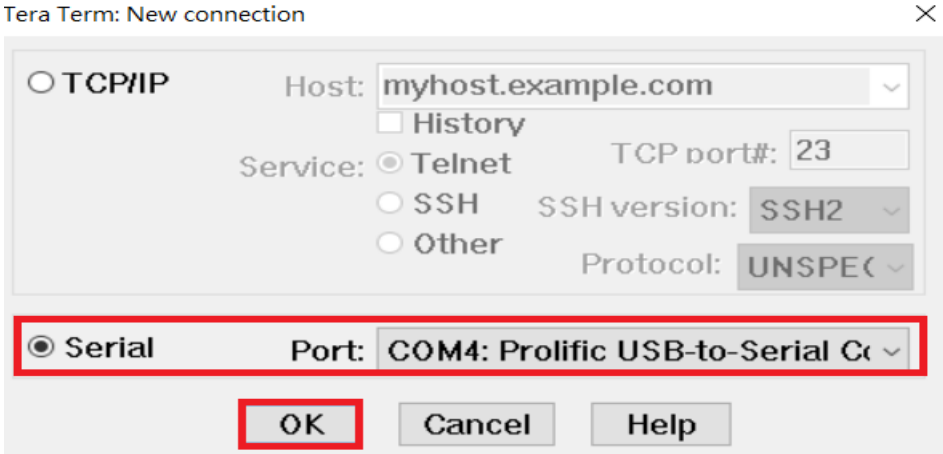
# How to debug

- If you want to control or display the debug console, see the following instructions:
  - RS232: Ping 2, 3, 5 Connecting line  
To J13: 2, 3, 4.(as the picture shows).



# How to debug

- Windows
  1. Install TeraTerm Tool to your PC.
  2. Choose correct serial port:

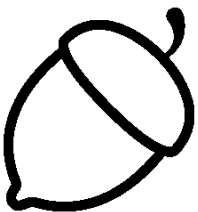


Tera Term: New connection

☐ TCP/IP    Host: myhost.example.com    ☐ History  
Service: ☒ Telnet    TCP port#: 23  
☐ SSH    SSH version: SSH2  
☐ Other    Protocol: UNSPEC

☒ Serial    Port: COM4: Prolific USB-to-Serial Co

OK    Cancel    Help



# How to debug

- Windows

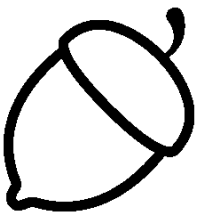
1. Open the TeraTrem and set Baud rate to 115200

- setup → Serial Port → Baud Rate  
“115200”

2. See the result:

```
Starting logging: OK
Initializing random number generator... [ 10.916867] random: dd: uninitialized urandom read (512 bytes read, 109 bits of entropy available)
done.
Starting network: OK
[ 11.368401] random: ssh-keygen: uninitialized urandom read (32 bytes read, 5 bits of entropy available)
Starting sshd: [ 11.489799] random: sshd: uninitialized urandom read (32 bytes read, 116 bits of entropy available)
OK

Welcome to Buildroot
buildroot login: [ 12.628765] random: python: uninitialized urandom read (32 bytes read, 126 bits of entropy available)
[ 12.947066] random: nonblocking pool is initialized
* Running on http://0.0.0.0:8080/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger pin code: 196-455-687
```



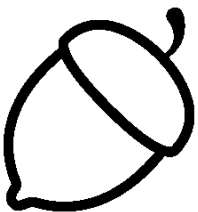
# How to debug

- Linux
  - Install minicom first

```
chris@ubuntu:~$ sudo apt install minicom
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libappindicator1 libindicator7 libpango1.0-0 libpangox-1.0-0
```

- Setting the minicom configuration

```
chris@ubuntu:~$ ls /dev/ttyUSB0
/dev/ttyUSB0
chris@ubuntu:~$ sudo minicom -s
```



# How to debug

- Linux
  - minicom configuration,
    - press “A”, Modify “ttyUSB0”
    - press “F”, change “N”
    - press “Enter”

```
+-----[configuration]-----+
| Filenames and paths          |
| File transfer protocols      |
| Serial port setup            |
| Modem and dialing            |
| Screen and keyboard          |
| Save setup as dfl             |
| Save setup as..              |
| Exit                          |
| Exit from Minicom            |
+-----+-----+

```

```
A - Serial Device                /dev/ttyUSB0
B - Lockfile Location            /var/lock
C - Callin Program
D - Callout Program
E - Bps/Par/Bits                 115200 8N1
F - Hardware Flow Control       No
G - Software Flow Control       No

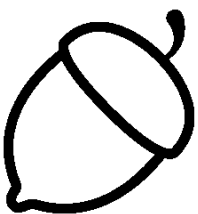
Change which setting? █

+-----+-----+
| Screen and keyboard          |
| Save setup as dfl             |
| Save setup as..              |
| Exit                          |
| Exit from Minicom            |
+-----+-----+

```

```
+-----[configuration]-----+
| Filenames and paths          |
| File transfer protocols      |
| Serial port setup            |
| Modem and dialing            |
| Screen and keyboard          |
| Save setup as dfl             |
| Save setup as..              |
| Exit                          |
| Exit from Minicom            |
+-----+-----+

```



# How to debug

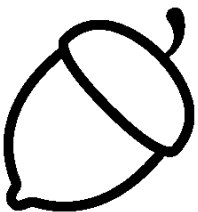
- Linux
  - Login

```
Welcome to minicom 2.7

OPTIONS: I18n
Compiled on Feb  7 2016, 13:37:27.
Port /dev/ttyUSB0, 14:48:38

Press CTRL-A Z for help on special keys

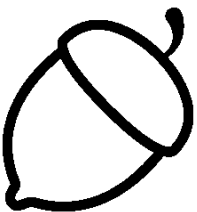
Welcome to Buildroot
buildroot login: /root/qc-web
```



# How to boot up

- Please follow these the steps from Chapter 1:

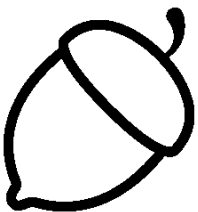
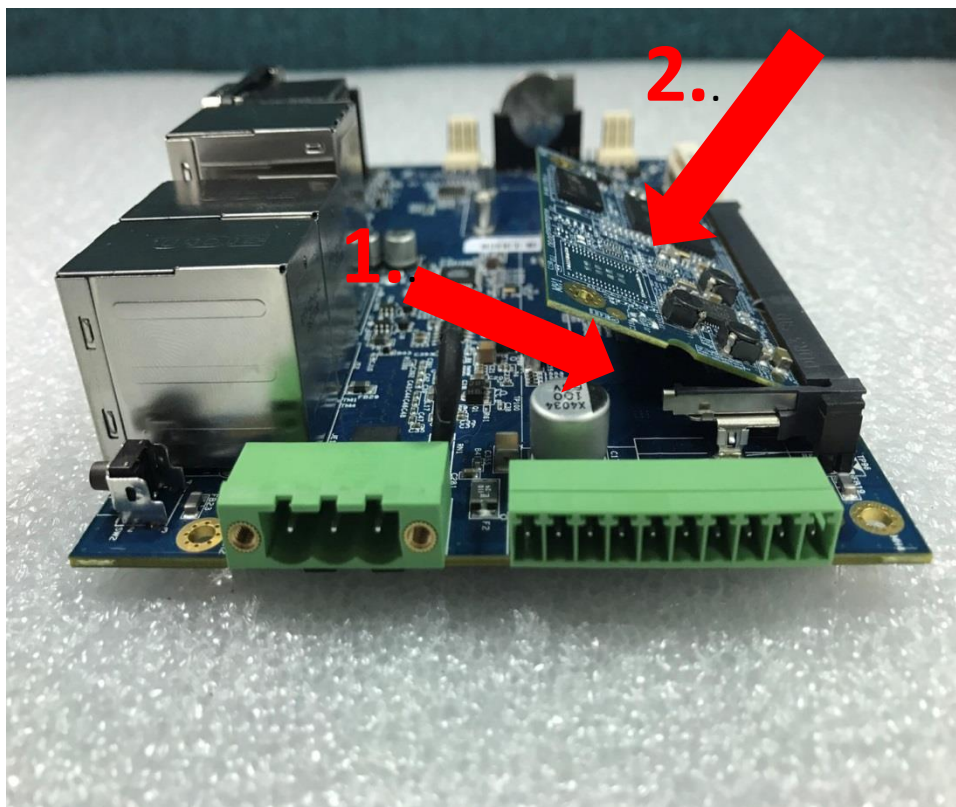
J → A → L → E → H





# Q&A

- Install your Almond into Walnut
  - Set 1: Tilt installation, Insert into connector as 45 degrees.
  - Set 2: Push down.



# Q&A

- Installation 204 pin SO-DIMM connector
  - Set 3: push the side to fixed your Almond SoM.

