

Cherry blossom,

GETTING STARTED GUIDE FOR FIRST RUN

Rev 3

20210626



What Is Cherry blossom?

Cherry Blossom is a single-board computer running embedded Linux. It has onboard eMMC that contains the Linux OS, as well as a microSD slot to run a different OS, or burn it to the eMMC.

It has lots of I/O pins, including analog-to-digital converter (ADC) and pulse-width-modulation (PWM) pins. It also has an HDMI output, a USB plug (for a camera, keyboard/mouse, etc.) and another USB for connection to a computer, ethernet, on-board LEDs, and a 5V input jack all Via Stacker board enhancements

This guide will cover everything you need to know to get started with your Cherry Blossom, including SSH-ing to it from your computer, connecting to the internet from your Cherry Blossom through your internet connected computer, and getting started using the onboard Linux OS.

What You Need

Pre requests

Mini usb b cable

External 5V 2A (specifically needed when using HAT boards)

Ethernet cable , for Ethernet Hat

FTDI driver install

Ftdi FT232R TTL Cable

<https://etcher.io/>

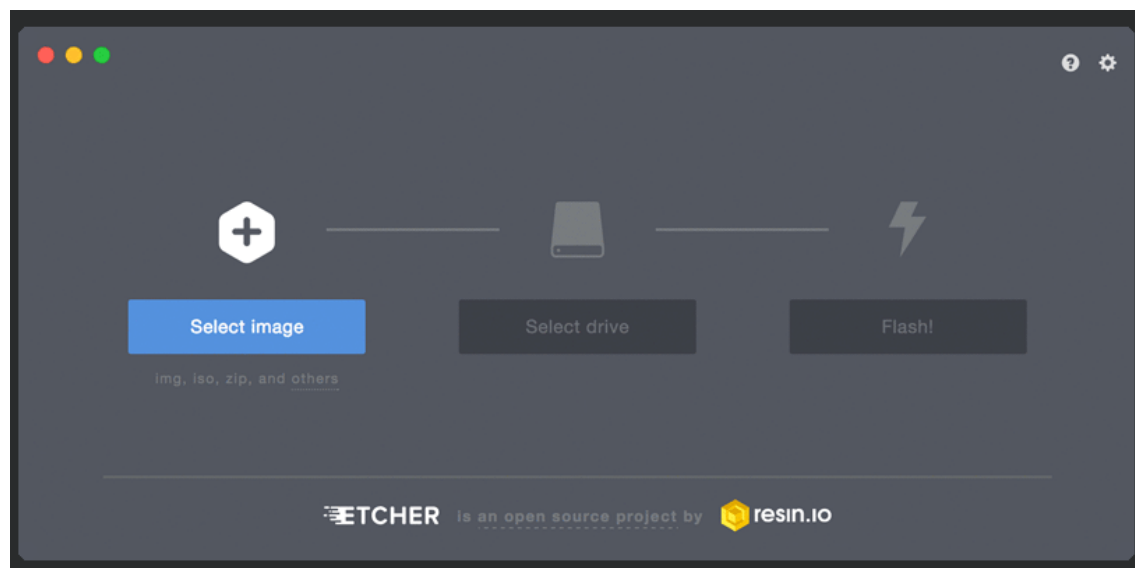
Download images at

<https://github.com/AADCherryBlossom/CB>

Loading Image to card,

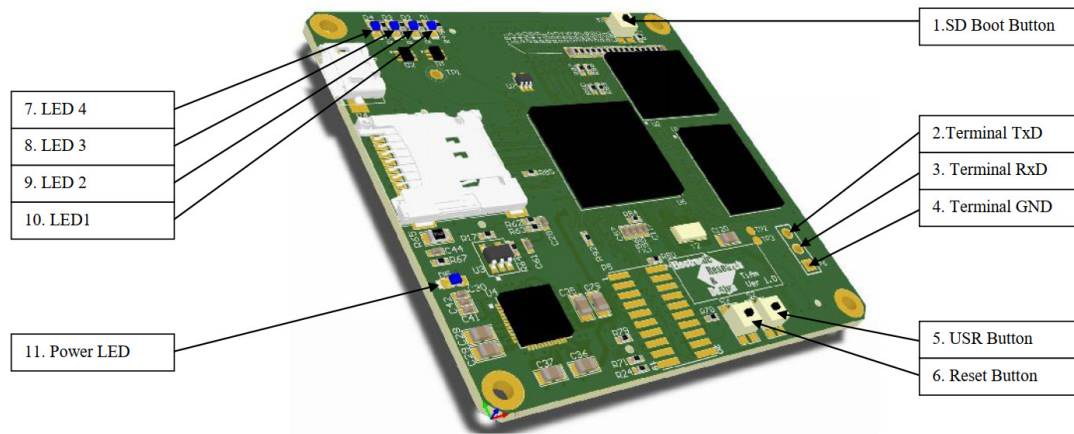
Loading the images on SD card you will need to use Etcher program, ensure the SD card is the only external storage on the system before continuing. Follow steps as per etcher program , recommendation is to use the IOT image for requirements that want to use the on board EMMC .

If the requirement is more resource intense , it is recommended to use a class10 16GB SD card high temp grade for longer operation and life conditions. Industrial SD card



Starting Up the board

Board definitions



Before connecting USB/POWER cable , press the SD boot button.

While the button is pressed, insert USB/POWER cable

Once cable is inserted release the boot button, it will take a while but it will enumerate a USB Ethernet connection under windows. "RNDIS" description

It is best to use the command line environment to determine the network interface over USB,

It will give you your IP address and the IP address of a gateway, Use Putty or similar program to access the unit by using the gateway ip address

```
C:\Windows\System32>ipconfig /all

Windows IP Configuration

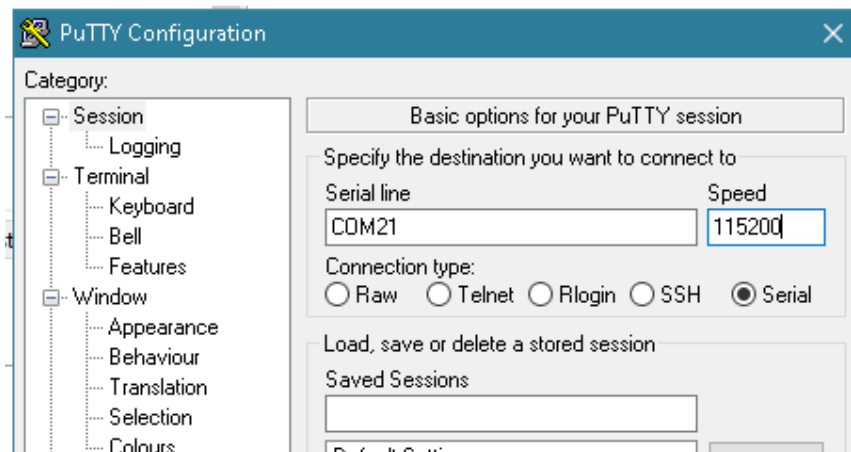
Host Name . . . . . : aadjnblgxbmm32
Primary Dns Suffix . . . . . : altron.com
Node Type . . . . . : Hybrid
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No
DNS Suffix Search List. . . . . : altron.com


Ethernet adapter Local Area Connection 5:

Connection-specific DNS Suffix . : 
Description . . . . . : RNDIS
Physical Address. . . . . : 6C-EC-EB-68-84-91
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::7d91:fd2d:1604:b23%82<Preferred>
IPv4 Address. . . . . : 192.168.7.1<Preferred>
Subnet Mask . . . . . : 255.255.255.252
Lease Obtained. . . . . : 03 July 2017 11:46:07 AM
Lease Expires . . . . . : 03 July 2017 11:48:06 AM
Default Gateway . . . . . : 
DHCP Server . . . . . : 192.168.7.2
DHCPv6 IAID . . . . . : 1382870251
DHCPv6 Client DUID. . . . . : 00-01-00-01-1D-52-8E-28-80-00-0B-89-76-

DNS Servers . . . . . : fec0:0:0:ffff::1%1
                       : fec0:0:0:ffff::2%1
                       : fec0:0:0:ffff::3%1
NetBIOS over Tcpip. . . . . : Enabled
```

The unit also enumerates a USB to serial interface , Putty can also be used to access the unit through serial



If the ether net Hat is used it will start the DHCP client and the unit can be accessed over the Ethernet. Use a IP scanner like Angry IP scanner to do a search of through a IP range of IPs. The Host name cherry blossom should come up under description

IP Range - Angry IP Scanner						
Scan Go to Commands Favorites Tools Help						
IP Range: 192.168.66.120 to 192.168.66.130		IP Range				
Hostname: CCDELL6540		IP1 Netmask		Start		
IP	MAC Address	Comments	Hostname	Ports [0+]	Ping	
192.168.66.120		[n/a]		[n/s]	2 ms	
192.168.66.121	D0:5F:B8:DD:82:D3	[n/a]	cherryblossom	[n/s]	0 ms	
192.168.66.122	[n/a]	[n/a]	[n/a]	[n/s]	[n/a]	
192.168.66.123	[n/a]	[n/a]	[n/a]	[n/s]	[n/a]	
192.168.66.124	[n/a]	[n/a]	[n/a]	[n/s]	[n/a]	
192.168.66.125	[n/a]	[n/a]	[n/a]	[n/s]	[n/a]	
192.168.66.126	[n/a]	[n/a]	[n/a]	[n/s]	[n/a]	
192.168.66.127	[n/a]	[n/a]	[n/a]	[n/s]	[n/a]	
192.168.66.128	[n/a]	[n/a]	[n/a]	[n/s]	[n/a]	
192.168.66.129	[n/a]	[n/a]		[n/s]	[n/a]	
192.168.66.130	[n/a]	[n/a]	[n/a]	[n/s]	[n/a]	

When connection is established, use the following username and password

username: cherry

password: blossom

or

username: debian

password: temppwd

or

username :Ubuntu

password :temppwd

cherry@CHERRYBLOSSOM: ~

```
login as: cherry
cherry@192.168.66.148's password:

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: Sun Nov  8 15:05:33 2020 from 192.168.66.110
cherry@CHERRYBLOSSOM:~$
```

After password success you need to do the next steps in root access

Type **su root** and then for the password type **blossom** again.

```
cherry@CHERRYBLOSSOM: ~  
login as: cherry  
cherry@192.168.66.148's password:  
  
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: Sun Nov  8 15:05:33 2020 from 192.168.66.110  
cherry@CHERRYBLOSSOM:~$ sudo su root  
[sudo] password for cherry:  
root@CHERRYBLOSSOM:/home/cherry#
```

NOTE , not compulsory but recommended for IOT image

It is recommended to load a develop all applications first before you do the next steps , this way the image will be loaded with all you programs loaded on the internal memory

RELOAD OF ONBOARD EMMC IMAGE

Once the previous steps are completed and the programs are loaded follow the below steps to make the current SD card we have been working a more permanent and load it into the EMMC.

Navigate to the directory as below image shows

```
cherry@cherryblossom:~$ cd /boot  
cherry@cherryblossom:/boot$
```

Once in the directory edit the following file

```
cherry@cherryblossom:/boot$ sudo nano -w /boot/uEnv.txt
```

Unmask the very last line in the file , this will cause the unit to load the image on the SD card to internal memory

```
cmdline=init=/opt/scripts/tools/eMMC/init-eMMC-flasher-v3.sh
```

Save the change and exit , press ctrl+x then enter

Now type reboot

```
cherry@cherryblossom:/$ sudo reboot
```

Once reboot is done , leave the SD card inserted and remove the power

Give it about 5 seconds,

Hold SD boot button and insert the USB/Power cable. Once USB/Power cable is inserted ,release SD boot button. The board will start up and the LED will go a bit crazy / running LED while the image is transferred to the EMMC internal memory.

Once all this is done ALL LEDS may be on then off, restart the device by doing a power cycle and REMOVE the SD card.

If all LEDS flash there was an error in transfer of image to emmc. Possible reason on board memory is too small for image .

Add a new SD card that does not contain an image for extra storage.

Applications to consider installing and having

```
//##### changing unix root password cherry blossom  
#####3
```

```
sudo passwd root
```

```
//##### move to root user cherry blossom  
#####
```

```
sudo su root
```

```
//##### cherry blossom default username and password  
#####
```

```
user: debian  
pass: tempwd
```

```
user: cherry
```



```

pass: blossom

user: ubuntu
pass: temppwd

//##### install set programs some cherry blossom
#####

***** expanding beagle bone blk sd card
image *****

sudo apt-get update
sudo apt-get upgrade

//expand the files system
cd /opt/scripts/tools/
git pull
sudo ./grow_partition.sh
sudo reboot

sudo apt-get update
sudo apt-get upgrade
sudo apt-get -f install

sudo apt-get update
sudo apt-get upgrade

//=====
//grow sd card size
//

sudo /opt/scripts/tools/grow_partition.sh

https://elinux.org/Beagleboard:Expanding\_File\_System\_Partition\_On\_A\_microSD

***** sequence important here *****

sudo apt-get update
sudo apt-get upgrade

sudo apt-get install gnome-tweak-tool

sudo apt-get install exfat-fuse exfat-utils

sudo apt-get install software-properties-common

sudo apt-get install build-essential python-dev python-setuptools python-
pip python-smbus -y

sudo pip install Adafruit_BBIO

sudo apt-get update
sudo apt-get upgrade

sudo apt-get install xz-utils

```

```

sudo apt-get install samba ..... windows file sharing

sudo apt-get install ssh ..... ssh telnet server
sudo apt-get install vsftpd .....ftpserver

sudo apt-get install make
sudo apt-get install gcc
sudo apt-get install python
sudo apt-get install python3

sudo apt-get install git-core
sudo apt-get install cron ..... task sheduler
sudo apt-get install mc ..... midnight
comander similar to norton comander/totalcommander

```

==== if midnight commander does not exist in the apt-get install follow the following lines =====

```

nano /etc/apt/sources.list
A source.list file opens, add these lines in the file and save and exit
from the file.
deb http://ftp.debian.org/debian/ stable main contrib non-free
deb http://ftp.de.debian.org/debian jessie main

```

```

then run
sudo apt-get update
sudo apt-get upgrade
=====
=====

```

```

sudo apt-get install minicom
sudo apt-get install picocom

```

```

sudo apt-get install network-manager

```

```

sudo apt-get install wput

```

```

sudo apt-get install aptitude

```

//##### creating the rc.local service to use it to execute startup code

create the file if it does not exist

```

sudo nano -w /etc/rc.local

```

add the folloing in the file

```

#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#

```

```
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

exit 0
```

Close and save the file with the above content

Create the service

```
nano /lib/systemd/rc.local.service
Edit the above file as necessary to invoke the different functionalities
like network. Enable these only if the code needs that particular service.
Disable unwanted ones to decrease boot time.
```

```
[Unit]
Description=rc.local startup
After=syslog.target network.target
[Service]
Type=simple
ExecStart=/etc/rc.local
[Install]
WantedBy=multi-user.target
```

Create a symbolic link to let the device know the location of the service.

```
cd /etc/systemd/system/
ln /lib/systemd/rc.local.service rc.local.service
```

Make systemd reload the configuration file, start the service immediately (helps to see if the service is functioning properly) and enable the unit files specified in the command line.

```
systemctl daemon-reload
systemctl start rc.local.service
systemctl enable rc.local.service
```

Restart cherry blossom immediately to see if it runs as intended.

reboot

```
##### resolve.conf ##### optional only if no
ping route can be resolved
```

this is your dns name you want to use to resolve addresses

```
echo "nameserver 8.8.8.8" > /etc/resolv.conf
```

```
invoke-rc.d hostname.sh start
invoke-rc.d networking force-reload
invoke-rc.d network-manager force-reload
```

```
##### cloning a SD card after setup is complete#
##### when your ready to make a copy of your setup and want
to role it out to another
```

find the device sunder /DEV ensure the disks are unmounted before
prosessing the dd command

ls /ev will list thed devices on the pc
do this before inserting the disc and again after inserting the disc, try
to find the new device name
usually this will come up as mmcblk0

now using the root permissions do the following

```
sudo dd if=/dev/mmcblk0 | gzip > bbb_sd_image_clone.gz .....
this take some time so be patient
```

after process is complete , insert the new SD card and follow the following
process

```
sudo gzip -dc bbb_sd_image_clone.gz | sudo dd of /dev/mmcblk1
.....mmcblk1 refers to the new sd card inserted
```

again wait until proces is complete , eject the new card and test.

on different linux box
dd if=/dev/sdi of=~/bbb_clone.img bs=1M count=4000
where sdi is the drive on the different linux box
bs is block sizes , and count is how many blocks do you want , 1M x 4000 =
4GB size

```
//=====
=====
//controlling GPIO using echo
```

Controlling the GPIO with the file system
You can use the following commands to control the GPIO with the file
system.

Exporting a pin:

```
echo 40 > /sys/class/gpio/export
Setting a pin OUTPUT:
```

```
echo out > /sys/class/gpio/gpio40/direction
Writing a pin HIGH:
```

```
echo 1 > /sys/class/gpio/gpio40/value
Writing a pin LOW:
```

```
echo 0 > /sys/class/gpio/gpio40/value
Setting a pin INPUT:
```

```
echo in > /sys/class/gpio/gpio40/direction
Reading the value from an INPUT pin (returns 1 for HIGH and 0 for LOW):
```

```
cat /sys/class/gpio/gpio40/value
```

```
//=====
//Controlling the GPIO with BoneScript
//You can use the following BoneScript commands to control the GPIO.
//
//Loading a BoneScript module:

var b = require('bonescript');
Setting a pin OUTPUT:

b.pinMode("P9_14", b.OUTPUT);
Writing a pin HIGH:

b.digitalWrite("P9_14", b.HIGH);
Writing a pin LOW:

b.digitalWrite("P9_14", b.LOW);
Setting a pin INPUT:

b.pinMode("P8_11", b.INPUT);
Reading the value from a digital INPUT pin (returns HIGH or LOW):

b.digitalRead("P8_11");
Setting a pin for pulse-width modulation (PWM) with 50 percent duty cycle:

b.pinMode('P9_14', b.OUTPUT);
b.analogWrite('P9_14', 0.5);
Reading the value from an analog INPUT pin (returns a value between 0 and
1):

b.analogRead('P9_40');

//=====
//Controlling the GPIO with Python
//You can use the following Python commands to control the GPIO.

Importing Adafruit's BeagleBone Input Output Library:

import Adafruit_BBIO.GPIO as GPIO
Setting a pin OUTPUT:

GPIO.setup("P9_14", GPIO.OUT)
Writing a pin HIGH:

GPIO.output("P9_14", GPIO.HIGH)
Writing a pin LOW:

GPIO.output("P9_14", GPIO.LOW)
Setting a pin INPUT:

GPIO.setup("P8_11", GPIO.IN)
Reading the value from a digital INPUT pin (returns HIGH or LOW):

GPIO.input("P8_11")
```

Setting a pin for PWM with 50 percent duty cycle:

```
import Adafruit_BBIO.PWM as PWM
PWM.start("P9_14", 50)
Setting an analog INPUT:
```

```
import Adafruit_BBIO.ADC as ADC
ADC.setup()
Reading the value from an analog INPUT pin (returns a value between 0 and 1):
```

```
analogReading = ADC.read("P9_40")
```

```
//=====
==
//adafruit gpio setup
```

```
sudo ntpdate pool.ntp.org
sudo apt-get update
sudo apt-get install build-essential python-dev python-pip python-smbus -y
git clone git://github.com/adafruit/adafruit-beaglebone-io-python.git
cd adafruit-beaglebone-io-python
sudo python setup.py install
cd ..
sudo rm -rf adafruit-beaglebone-io-python
```

```
//=====
=== optional if required
// Sakis3g connection internet usb modem
```

always recommend running package update before we start installing anything:

```
sudo apt-get update
```

```
sudo apt-get install ppp
```

```
wget "http://raspberry-at-home.com/files/sakis3g.tar.gz"
```

```
sudo mkdir /usr/bin/modem3g
sudo chmod 777 /usr/bin/modem3g
sudo cp sakis3g.tar.gz /usr/bin/modem3g
cd /usr/bin/modem3g
sudo tar -zxvf sakis3g.tar.gz
sudo chmod +x sakis3g
```

```
sudo ./sakis3g connect
```

Please select APN by using APN variable, or by enabling interactive mode.

```
$ /usr/bin/modem3g/sakis3g --interactive "connect"
```

Available options are:

```
Internet          (Internet)
CUSTOM_APN        Custom APN...
```

Example:

```
$ /usr/bin/modem3g/sakis3g APN="Internet"
```

When you're done with creation of /etc/sakis3g.conf you can now run:
sudo ./sakis3g connect
sudo ./sakis3g disconnect

```
/usr/bin/modem3g/sakis3g --interactive "menu" "console"
```

```
//=====
===== optional if required
// LIB QMI setup
//
```

```
apt-get install libqmi-utils
```

<https://techship.com/faq/how-to-step-by-step-set-up-a-data-connection-over-qmi-interface-using-qmicli-and-in-kernel-driver-qmi-wwan-in-linux/>

<http://www.linuxfromscratch.org/blfs/view/svn/general/libqmi.html>

```
//===== install mosquitto
=====
```

```
sudo apt update
```

```
sudo nano -w /etc/apt/sources.list
Add "deb http://httpredir.debian.org/debian stretch main contrib non-free"
to the bottom of the file. Write and close the file.
```

```
sudo apt-get update
sudo apt-get install mosquitto
sudo apt install mosquitto mosquitto-clients
```

extra resource :
<https://thenewstack.io/off-the-shelf-hacker-mosquitto-on-the-bone/>
<https://www.digitalocean.com/community/tutorials/how-to-install-and-secure-the-mosquitto-mqtt-messaging-broker-on-ubuntu-18-04>