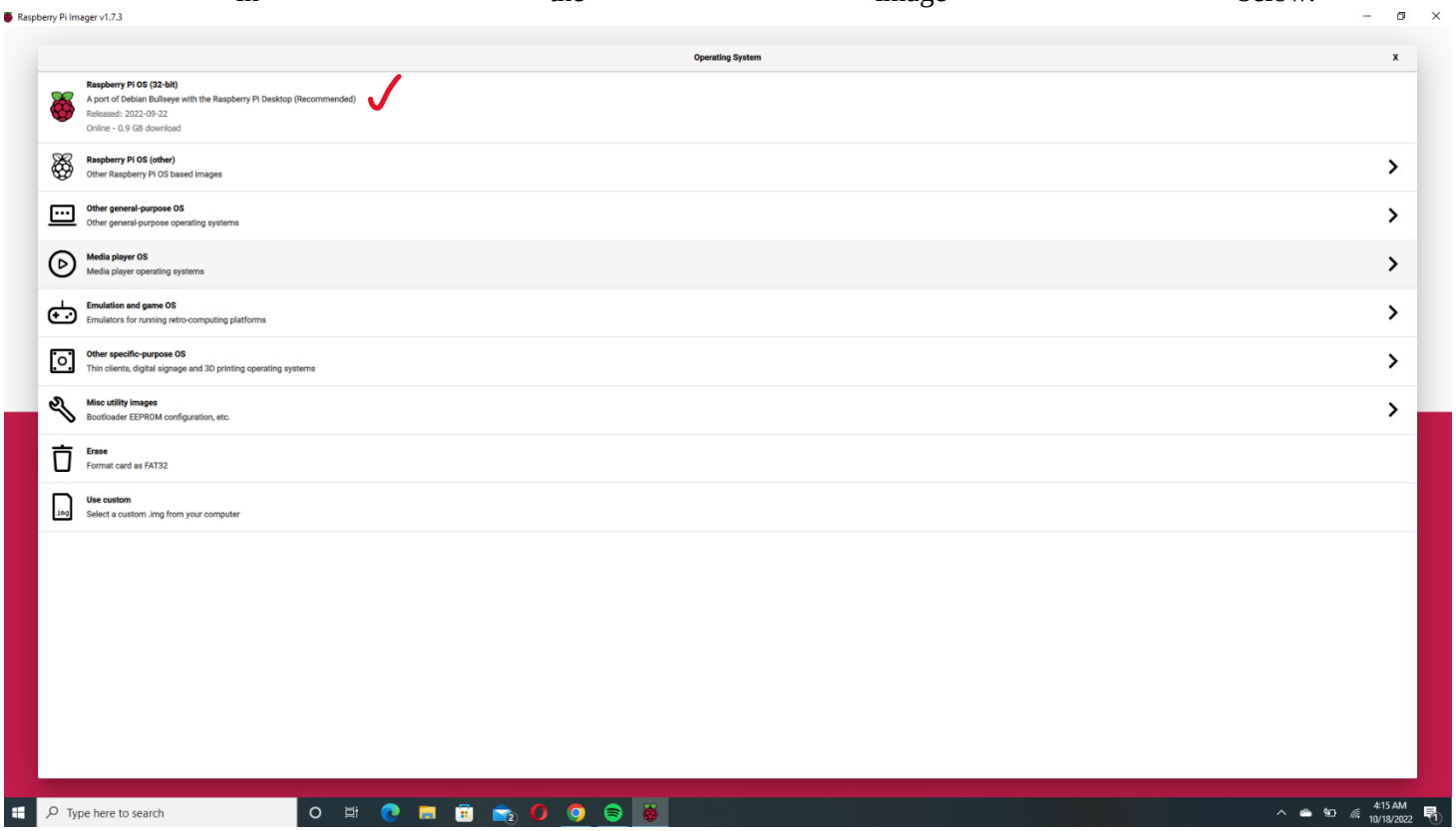
DTN & ION software installation instructions

Installation process for Pi OS

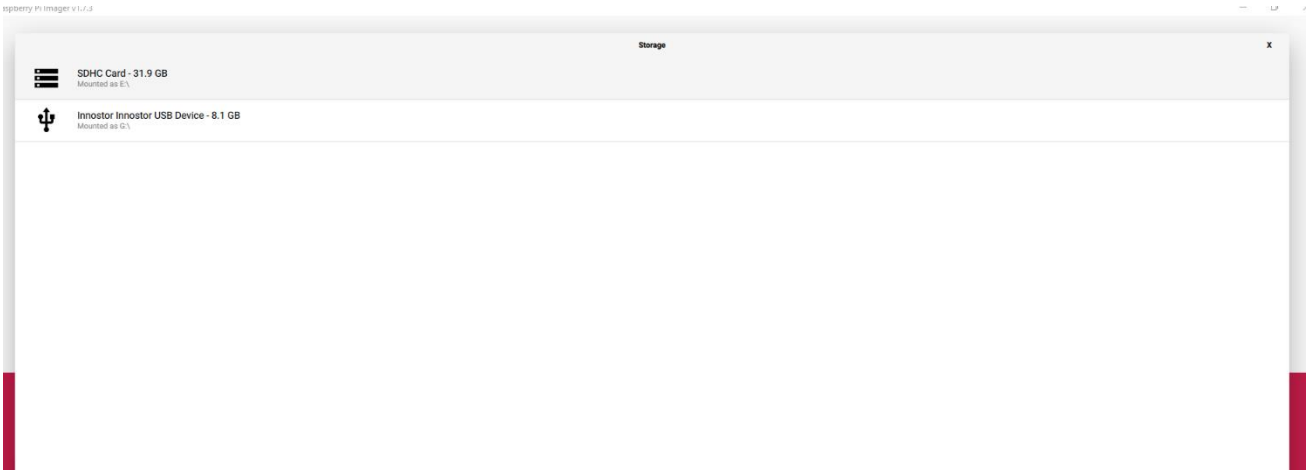
For this process you will need to go to https://www.raspberrypi.com/software/ and download the raspberry pi imager. Once the program in downloaded and running in your PC please insert the SD card that you desire to use to install the software.

Installing the Operative System

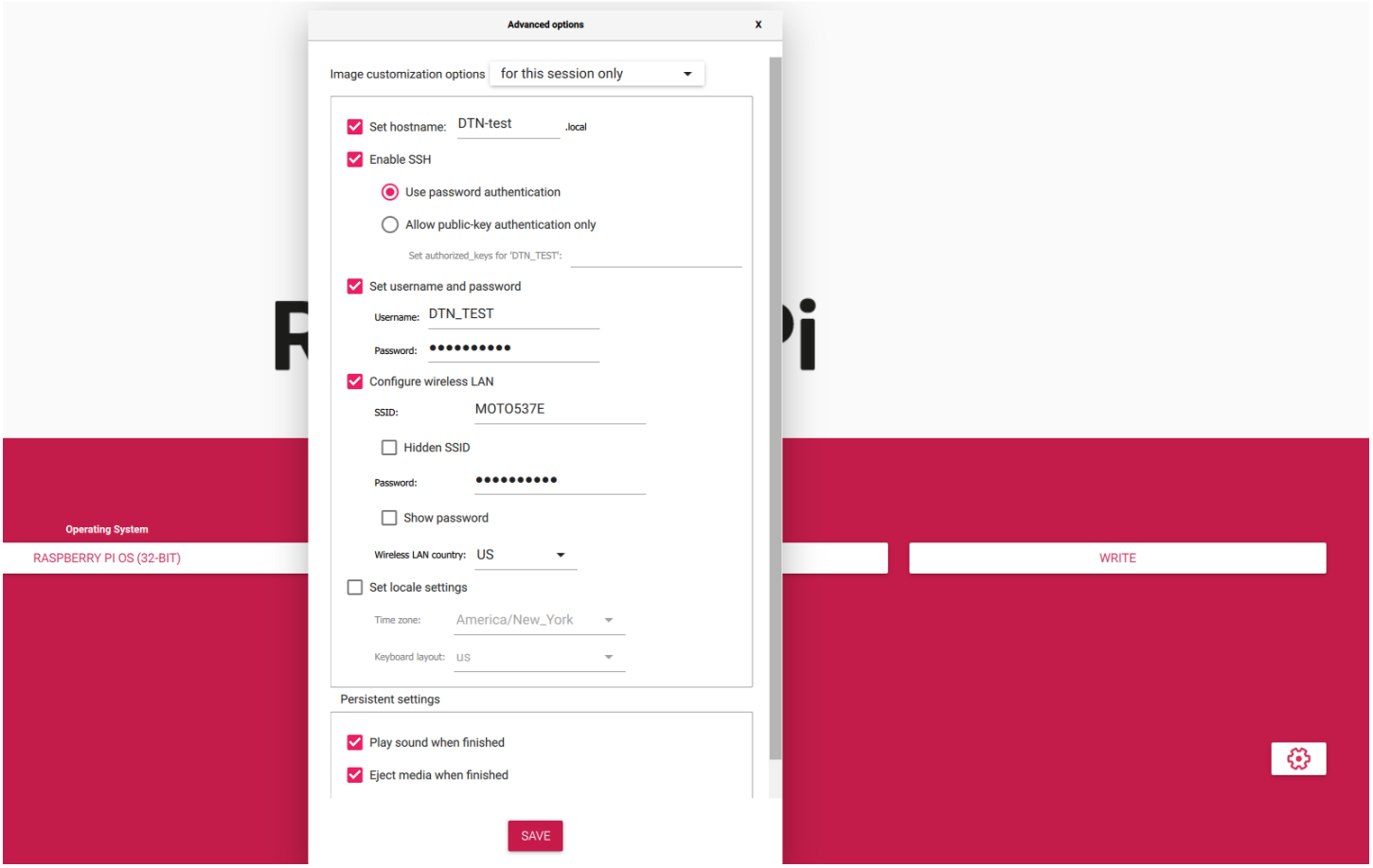
1. Run the "Raspberry Pi Imager", click on CHOOSE OS and select the first option as shown in the image below.



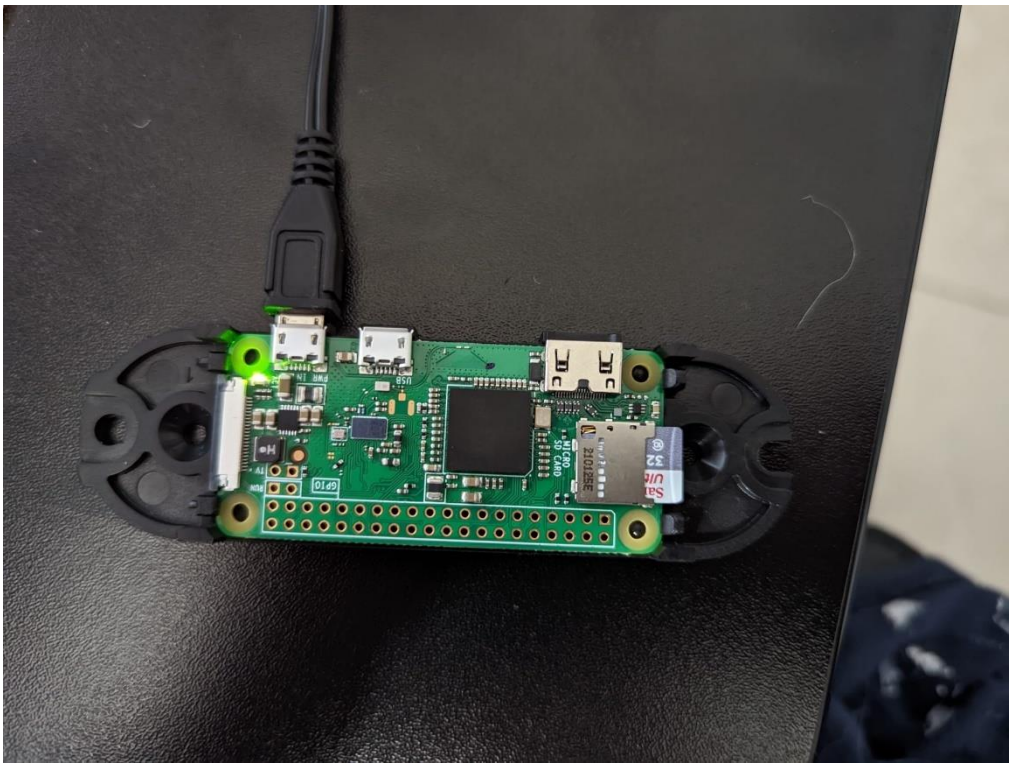
2. Select the microSD card where you want to install the Operative System



3. Click the configuration button and fill the required information. Make sure that the option "Enable SSH" is ON and set it up as "use password authentication"; The credentials entered here are the ones that will be used to login to the machine later through SSH.



4. Click on “WRITE” and wait until the process is completed. Once the process is done take the microSD and insert it on the raspberry pi, connect the raspberry pi to the power and wait until it has a solid green light.



5. Go to *“This PC”* and check the external devices connected your computer, you should be able to see that the name of the microSD card has now changed to *“boot”*.
6. Login to your modem and check with devices are connected to the network, you should see the just created “Raspberry pi” device connected in your network.

Access

2.4 GHz | 5 GHz

List of Connected Devices

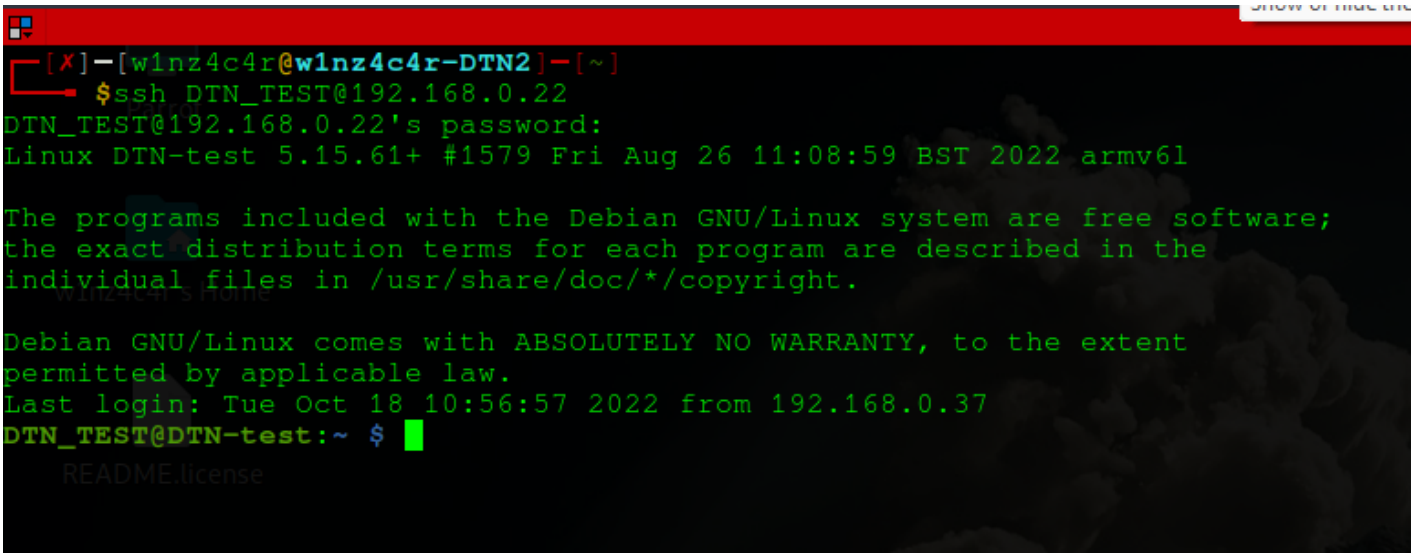
MAC Address	Time Connected	Signal Strength(dBm)	IP Address	Host Name	Mode	Speed(Kbps)
B8:27:EB:B2:95:78	0d 0h 0m	-35	192.168.0.22	DTN-test	802.11n	58500
92:4F:B1:8A:C3:1F	0d 0h 22m	-33	192[redacted]-S7		802.11n	6000
00:E9:3A:39:DD:C8	0d 1h 55m	-59	192[redacted]1700		802.11n	1000
D0:05:2A:62:3A:92	0d 4h 12m	-46			802.11n	1000
FA:AF:0C:EB:F7:4A	0d 4h 58m	-63	192.[redacted]		802.11n	1000
68:14:01:96:96:85	1d 6h 23m	-67	192.[redacted]9685		802.11n	5500
B4:C9:B9:B5:7D:2D	6d 14h 7m	-43	192.[redacted]F37		802.11n	72222

Getting Connected to the Raspberry Pi.

You will need to use a tool such as Putty or a linux virtual machine for ease, in my case im going to be using my own linux virtual machine to connect to the raspberry pi.

1. Connect to the machine using SSH, the credentials are the ones that you set up on the installation process.

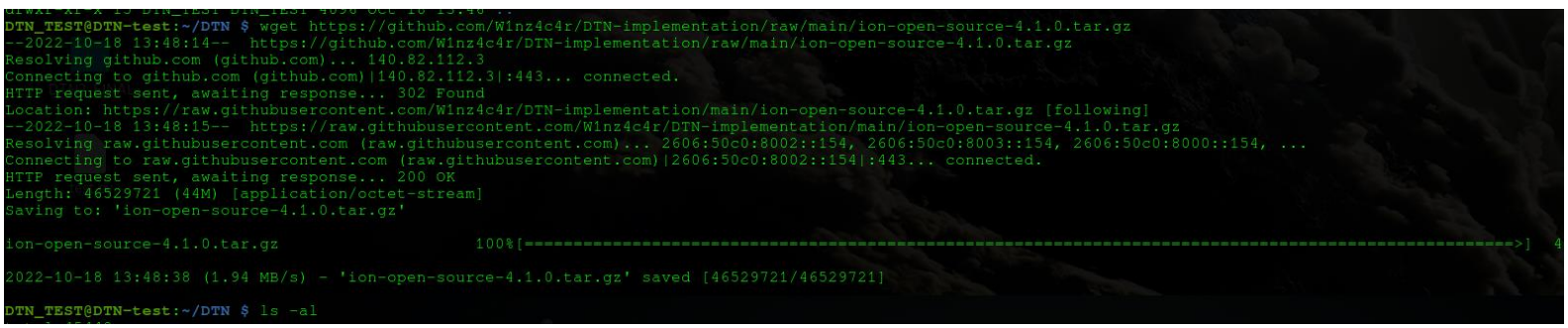
    I. ssh DTN_TEST@192.168.0.22



DTN and ION installation process.

All the installation steps and file can also be found at https://github.com/W1nz4c4r/DTN-implementation

2. Create a folder in where you are going to safe all the DTN/Bundle protocol files
3. download the ion.tar
    - wget https://github.com/W1nz4c4r/DTN-implementation/raw/main/ion-open-source-4.1.0.tar.gz



4. Unzip the .tar file

   - tar -xf ion-open-source-4.1.0.tar.gz

```
DTN_TEST@DTN-test:~/DTN $ tar -xf ion-open-source-4.1.0.tar.gz
DTN_TEST@DTN-test:~/DTN $ ls -la
total 45452
drwxr-xr-x  3 DTN_TEST DTN_TEST     4096 Oct 18 13:52 .
drwxr-xr-x 15 DTN_TEST DTN_TEST     4096 Oct 18 13:46 ..
drwxr-xr-x 27 DTN_TEST DTN_TEST     4096 May 31  2021 ion-open-source-4.1.0
-rw-r--r--  1 DTN_TEST DTN_TEST 46529721 Oct 18 13:48 ion-open-source-4.1.0.tar.gz
DTN_TEST@DTN-test:~/DTN $
```

5. cd ion-open-source-4.1.0
6. autoheader
   - **I.** If it shows the error ***"-bash: autoheader: command not found"*** please follow this steps:

```
DTN_TEST@DTN-test:~/DTN/ion-open-source-4.1.0 $ autoheader
-bash: autoheader: command not found
DTN_TEST@DTN-test:~/DTN/ion-open-source-4.1.0 $
```

   - **i.** sudo apt update
   - **ii.** sudo apt install autoconf

```
DTN_TEST@DTN-test:~/DTN/ion-open-source-4.1.0 $ sudo apt install autoconf
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  libfuse2
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  automake autotools-dev libsigsegv2 m4
Suggested packages:
  autoconf-archive gnu-standards autoconf-doc libtool gettext m4-doc
The following NEW packages will be installed:
  autoconf automake autotools-dev libsigsegv2 m4
0 upgraded, 5 newly installed, 0 to remove and 0 not upgraded.
Need to get 1424 kB of archives.
After this operation, 4307 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

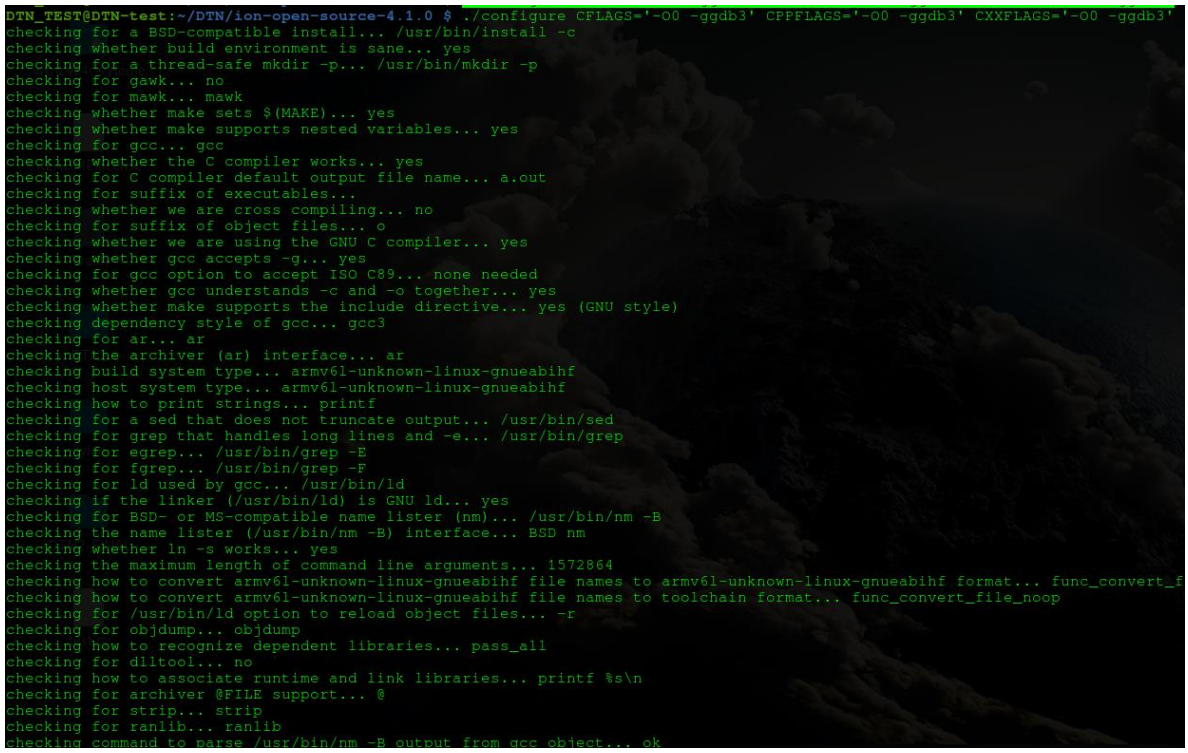
   - **iii.** sudo localectl set-locale LANG=en_IN.UTF-8
   - **iv.** sudo reboot
   - **v.** wait for one or two minutes while the Raspberry reboot. Connect again with ssh and go to the folder you where previously on.

vi. Autoheader

7. Aclocal
8. Autoconf
9. Automake

```
DTN_TEST@DTN-test:~/DTN/ion-open-source-4.1.0 $ aclocal
DTN_TEST@DTN-test:~/DTN/ion-open-source-4.1.0 $ autoconf
DTN_TEST@DTN-test:~/DTN/ion-open-source-4.1.0 $ automake
Makefile.am:485: warning: '%'-style pattern rules are a GNU make extension
Makefile.am:488: warning: '%'-style pattern rules are a GNU make extension
Makefile.am:491: warning: '%'-style pattern rules are a GNU make extension
Makefile.am:685: warning: '%'-style pattern rules are a GNU make extension
Makefile.am:688: warning: '%'-style pattern rules are a GNU make extension
Makefile.am:691: warning: '%'-style pattern rules are a GNU make extension
Makefile.am:825: warning: '%'-style pattern rules are a GNU make extension
Makefile.am:828: warning: '%'-style pattern rules are a GNU make extension
Makefile.am:1146: warning: '%'-style pattern rules are a GNU make extension
Makefile.am:1149: warning: '%'-style pattern rules are a GNU make extension
Makefile.am:1152: warning: '%'-style pattern rules are a GNU make extension
Makefile.am:1724: warning: '%'-style pattern rules are a GNU make extension
Makefile.am:1727: warning: '%'-style pattern rules are a GNU make extension
Makefile.am:2006: warning: '%'-style pattern rules are a GNU make extension
Makefile.am:2009: warning: '%'-style pattern rules are a GNU make extension
Makefile.am:2012: warning: '%'-style pattern rules are a GNU make extension
Makefile.am:2483: warning: '%'-style pattern rules are a GNU make extension
Makefile.am:2486: warning: '%'-style pattern rules are a GNU make extension
Makefile.am:2489: warning: '%'-style pattern rules are a GNU make extension
Makefile.am:2709: warning: '%'-style pattern rules are a GNU make extension
Makefile.am:2712: warning: '%'-style pattern rules are a GNU make extension
Makefile.am:2789: warning: '%'-style pattern rules are a GNU make extension
Makefile.am:2792: warning: '%'-style pattern rules are a GNU make extension
Makefile.am:2795: warning: '%'-style pattern rules are a GNU make extension
Makefile.am:2934: warning: '%'-style pattern rules are a GNU make extension
Makefile.am:2937: warning: '%'-style pattern rules are a GNU make extension
Makefile.am:2940: warning: '%'-style pattern rules are a GNU make extension
Makefile.am:3032: warning: '%'-style pattern rules are a GNU make extension
Makefile.am:3035: warning: '%'-style pattern rules are a GNU make extension
Makefile.am:3112: warning: '%'-style pattern rules are a GNU make extension
Makefile.am:3115: warning: '%'-style pattern rules are a GNU make extension
Makefile.am:3118: warning: '%'-style pattern rules are a GNU make extension
Makefile.am:3219: warning: '%'-style pattern rules are a GNU make extension
Makefile.am:3222: warning: '%'-style pattern rules are a GNU make extension
Makefile.am:3225: warning: '%'-style pattern rules are a GNU make extension
Makefile.am:3318: warning: '%'-style pattern rules are a GNU make extension
Makefile.am:3480: warning: ':='-style assignments are not portable
Makefile.am:3480: warning: filter %.1 %.3 %.5,$(man_MANS: non-POSIX variable name
Makefile.am:3480: (probably a GNU make extension)
Makefile.am:3482: warning: ':='-style assignments are not portable
Makefile.am:3482: warning: patsubst %,%.ps,$(justmans: non-POSIX variable name
Makefile.am:3482: (probably a GNU make extension)
Makefile.am:3483: warning: ':='-style assignments are not portable
Makefile.am:3483: warning: sort $(filter %.1.ps,$(mans_as_ps: non-POSIX variable name
Makefile.am:3483: (probably a GNU make extension)
Makefile.am:3484: warning: ':='-style assignments are not portable
Makefile.am:3484: warning: sort $(filter %.3.ps,$(mans_as_ps: non-POSIX variable name
Makefile.am:3484: (probably a GNU make extension)
Makefile.am:3485: warning: ':='-style assignments are not portable
Makefile.am:3485: warning: sort $(filter %.5.ps,$(mans_as_ps: non-POSIX variable name
Makefile.am:3485: (probably a GNU make extension)
Makefile.am:3486: warning: ':='-style assignments are not portable
Makefile.am:3489: warning: '%'-style pattern rules are a GNU make extension
```

10. ./configure CFLAGS='-O0 -ggdb3' CPPFLAGS='-O0 -ggdb3' CXXFLAGS='-O0 -ggdb3'



The following instructions will take some time to finish

11. Make
12. sudo make install
13. sudo ldconfig

If there is no error display the installation of the DTN-ION software has finished successfully

Installing python3 library (pyion)

this python library pyion will be the in charge of handling all the bundle protocol usage and its all facilitated on pre-defined classes.

1. sudo apt-get install autotools-dev automake python3-dev
2. add the ion folder to the environment
    a. export ION_HOME=/"your-ion-folder-path"
    b. you can verify it was correctly adding by entering:
        - echo $ION_HOME

    the path of the ion folder should show up

```
DTN_TEST@DTN-test:~/DTN/ion-open-source-4.1.0 $ pwd
/home/DTN_TEST/DTN/ion-open-source-4.1.0
DTN_TEST@DTN-test:~/DTN/ion-open-source-4.1.0 $ export ION_HOME=/home/DTN_TEST/DTN/ion-open-source-4.1.0
DTN_TEST@DTN-test:~/DTN/ion-open-source-4.1.0 $ echo $ION_HOME
/home/DTN_TEST/DTN/ion-open-source-4.1.0
DTN_TEST@DTN-test:~/DTN/ion-open-source-4.1.0 $
```

3. git clone --branch v4.1.0 https://github.com/msancheznet/pyion.git

4. cd pyion

5. sudo -E python3 setup.py install

Installing rlwrap

We will use the program rlwrap to run our main DTN code because it will make its navigation a lot smother

- sudo apt install rlwrap