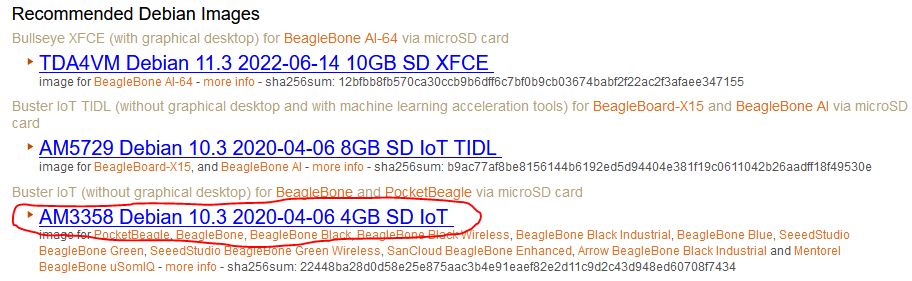
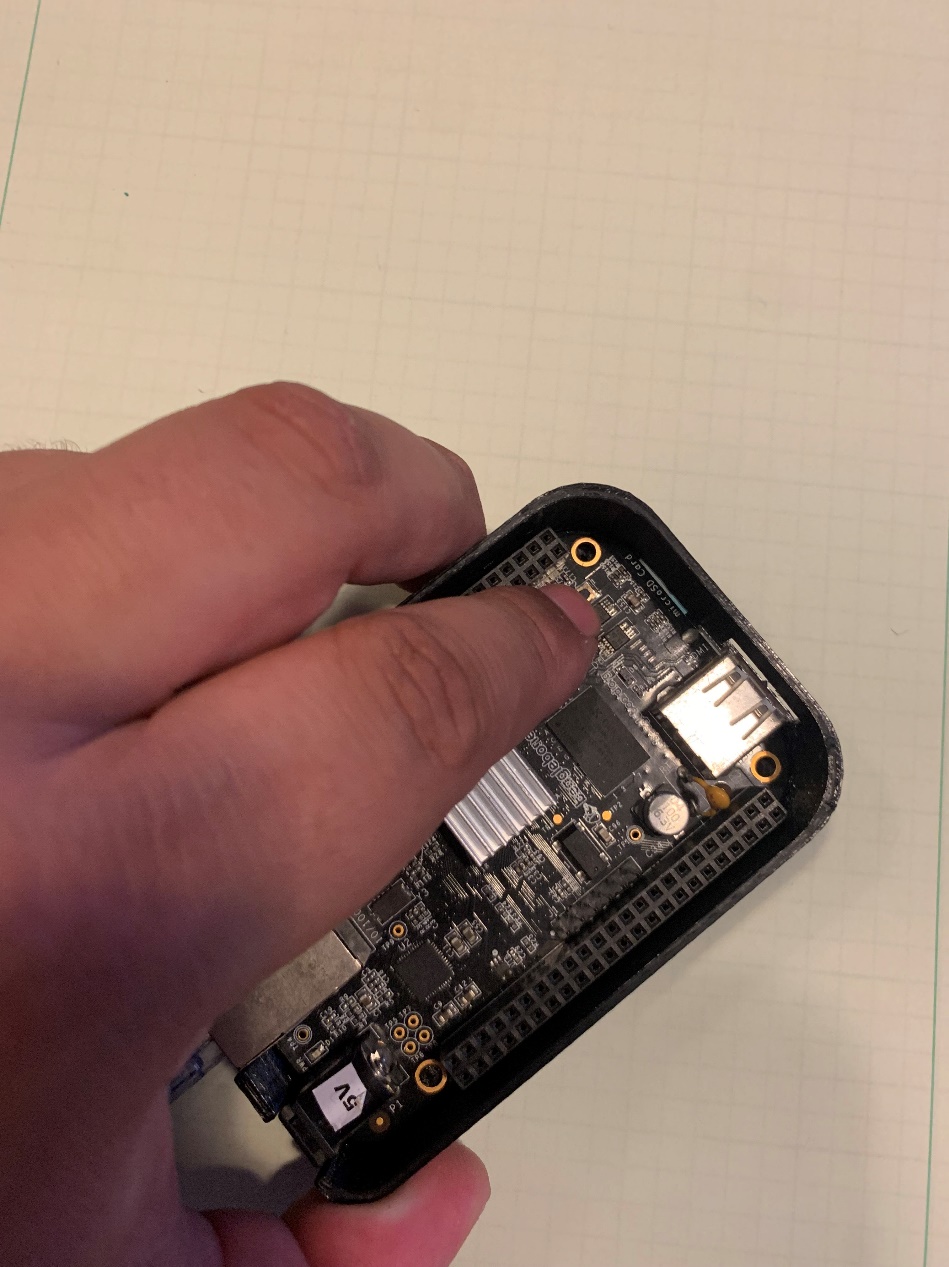
How to setup a BeagleBone Black for live video feed and sending live commands through UART

# Download the latest image for Debian

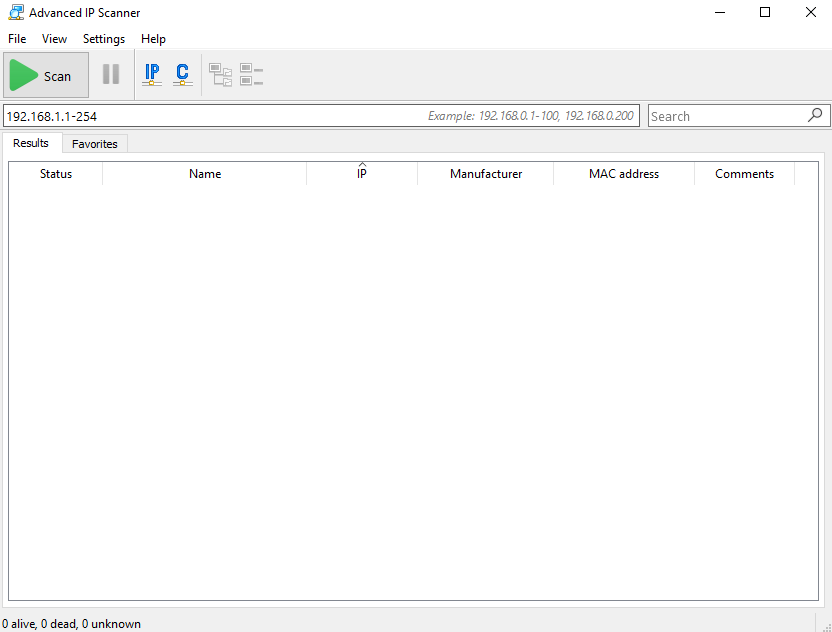
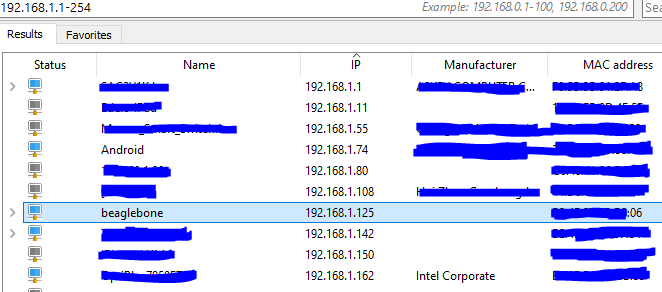
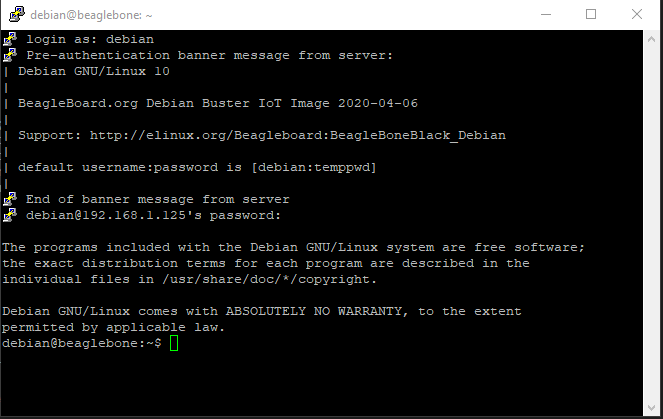
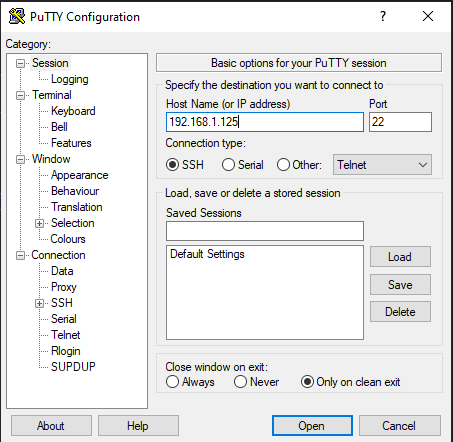
Follow this link: <https://beagleboard.org/latest-images> and download “Buster IoT” for the AM3358 under the “Recommended Debian Images” Section. (The link is usually blue, download the latest image available.

# Image SD card with a Linux distribution for the BeagleBone

1. Download and install balenaEtcher from this link: <https://www.balena.io/etcher/> and install it on your computer.
2. Insert your microSD card to your computer.
3. On balenaEtcher:
4. Click on: “Flash from file”, a window will open, look for the image you downloaded on Step 1.
   1. select your SD card. **MAKE SURE YOU SELECT THE CORRECT DRIVE BECAUSE ALL THE PARTITIONS AND DATA WILL BE ERASED!**
   2. Click on: “Flash!”, you will get a “Permission” dialog window. Accept in order for the flashing to occur. This step will take about 5 minutes depending on the speed of your computer, card reader, and card speed. It will take an additional 3 minutes to verify the image on the SD card. Once it finishes successfully eject the card safely.
5. With the power DISCONNECTED, insert the SD card into the BeagleBone SD slot.
6. In order to boot into Debian from the SD card and not it’s onboard flash memory, the USER/BOOT button must be pressed while the power is connected.
7. After all the lights turn blue and then off, you may stop pressing the button. After it has fully booted, the first LED will be blinking in a “heartbeat” fashion.

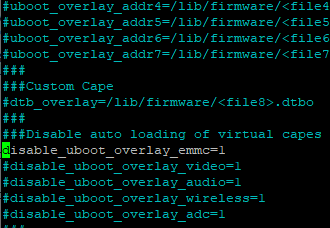
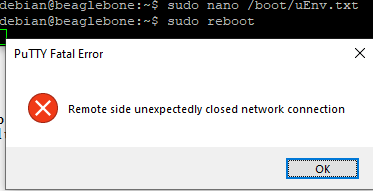
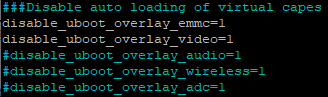
# Connect to network and logging in into Debian

Since we won’t be using a monitor for this setup, we will have to connect to the BeagleBone through SSH. Meaning that we will connect the board to the network through the ethernet port. After connecting to the ethernet port, and successfully booting into Linux, we need to figure the board’s private IP address. The easiest way for me to find the BeagleBone private address is by using “Advanced IP Scanner” on a computer that is on the same network as the BeagleBone, it can be downloaded from the following link: <https://www.advanced-ip-scanner.com/>

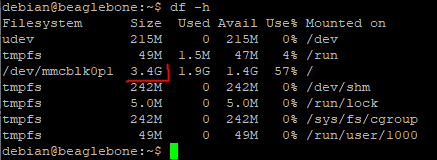
1. Once the BeagleBone is running, connect it to your network by using an ethernet cable (a wifi card can be added later). Having the green and orange light in the ethernet connector is a sign of communication.
2. Open “Advanced IP Scanner”, make sure that in the IP address bar you are in the correct range. I will have mine to scan from 192.168.1.1 – 192.168.1.254. Depending on your router your IP address range might be different. You can try 192.168.2.1 – 192.168.2.254 if the first range didn’t work. The length of your listed devices after scan will depend on how many devices are connected to your network (you might even find devices that are getting “free” internet from you 😉).
3. The IP address that my BeagleBone was automatically assigned by the router is 192.168.1.125
4. In order to connect through SSH, PuTTy will be used. It can be downloaded from the following link: <https://www.putty.org/>
5. In PuTTY, I’ll be connecting using the SSH protocol, port 22. And entering the IP address of the board. For quick connection, under “Saved Sessions”, a quick profile can be saved.
6. Depending if this device has been previously connected or not, a warning message might pop-up, there is no reason for concern in this case, we’ll just continue. You will be asked for the login user name, the default username is “debian” and then you will be asked for the password, which the default is “temppwd”. It can be changed later and it’s recommended. I will keep the default values for the rest of the document.

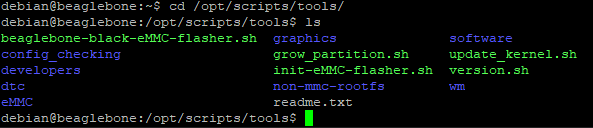
# Optional: Disable emmc (internal flash memory) and HDMI

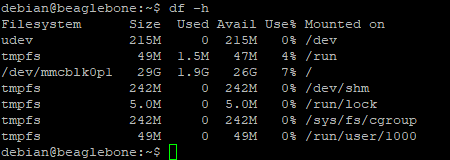
Since we’ll be working off the SD card, we don’t need the internal memory to be enabled since they take up I/O that can be used for other purposes. Also, since we’ll be working off the network and won’t be connecting a screen to the HDMI port of the board, we can disable that and have more I/O available for more devices. We need to modify the uEnv.txt file as a root. We use “nano” as the text editor in order to modify the file. Let’s do the following in order to disable these modules.

1. “sudo nano /boot/uEnv.txt” <- “sudo” is used to run “nano” as a super user or “root”. Otherwise, we won’t have permission to modify the file. Be careful when running command using sudo.
2. Remove the “#” symbol on the line “#disable\_uboot\_overlay\_emmc=1”
3. To save the changes: On your keyboard press “Ctrl” + “x”. You will be prompted if you want to save the changes. Press “y” and then “enter” to confirm. Reboot the system by typing: “sudo reboot”, enter the password if asked for it.
4. After the system reboots, you will get an error that the connection is lost. This is normal, open another session as done before using PuTTY. Once back in the board, repeat the steps above, but this time remove the “#” symbol on the line “#disable\_uboot\_overlay\_video=1” in order to disable the HDMI port and free up additional I/O in the header.
5. Again, save the changes. Enter the password when prompted and reboot the BeagleBone again.
6. At this point both the emmc and the HDMI port will be disabled.

# Expand system on SD card

The SD card that I’m using is 32GB of storage, but when I run the command: “df -h” I can see that I can only use 3.4GB. In the latest image of Debian, there is a script that will expand the partition in order to use the whole SD card, space might be needed to install, applications, updates, and other user files.

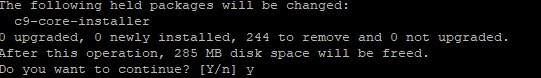
In the folder “tools” of Debian, we can see the included scripts that can help us automate several things. We’ll be running “grow\_partition.sh”. In order to go to this folder, we need to run the following command: “cd /opt/scripts/tools/” and then type the command “ls” in order to see all the files and folder on “tools” directory. “cd” is the command to “change directory” it is used when you want to go to a different folder. “ls” is the command used to list files or directories in Linux.

1. Enter the command: “cd /opt/scripts/tools/”
2. Then enter: “sudo ./grow\_partition.sh” <- in order to run the file as an executable/script
3. Reboot by using: sudo reboot and log back again
4. Double check how much space is available for the mmcblk0p1 partition by using “df -h” again.

Now that we have all the space available, we can continue to download and install updates. But before that we should remove three systems that take up space and will slow down installing and downloading updates.

# Remove Bonescript, Cloud9 and NodeRed

I won’t be using Bonescript, Cloud9 and NodeRed to develop on the BeagleBone, I followed the steps on this page: <https://www.diozero.com/internals/bbbsetup.html> in order to remove them. This page has additional tips and steps that can help in the development and maintenance of the board. Type the command on the snipped bellow in order to free up space in the SD card and makes updating the board more efficient.

for the commands: “sudo apt -y remove c9-core…” and “sudo apt -y purge c9-core…” I removed the “-y” in order for them to work. When prompted if “you want to continue?” I press the letter “y” and then “enter”

After running all the commands, reboot the BeagleBone. Let’s get ready to install updates.

# Install updates

As in all OS we should install the latest updates, updates can either add features, fix bugs and/or add security features. In Debian based OS (we installed Debian in this guide), we run two commands.

* First: “sudo apt update”, this command will download an updated list of the packages you currently have installed. You can think of it as updating a catalog magazine with its newest additions from the current applications you have installed.
* Second: “sudo apt upgrade”, this command will download and install all available packages that are on the list you just updated with the previous command. You can think of it as buying all the items in the catalog magazine.

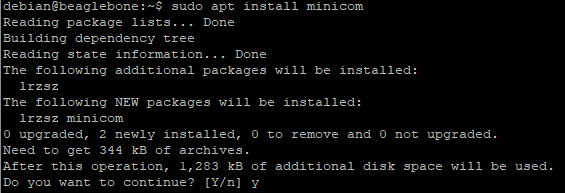
Remember, “apt upgrade” will download and install whatever is in the saved catalog. If you want to install the newest packages and apps you must “apt update” your catalog first.

1. Run the command: “sudo apt update”. Enter the password if requested. Depending on internet speed and what’s the current state of your OS, this might take a few minutes. Wait until it finishes.
2. Run the command: “sudo apt upgrade”. Enter the password if requested. Depending on internet speed and what’s the current state of your OS, this might take a few minutes. Eventually, you will get a message with a small report. At the end of the report, you are told how much of additional storage will be needed to complete the upgrade. Type “y” and then “enter”. Again, this might take a few minutes.
3. For safe measure, “reboot” the board and log back in.

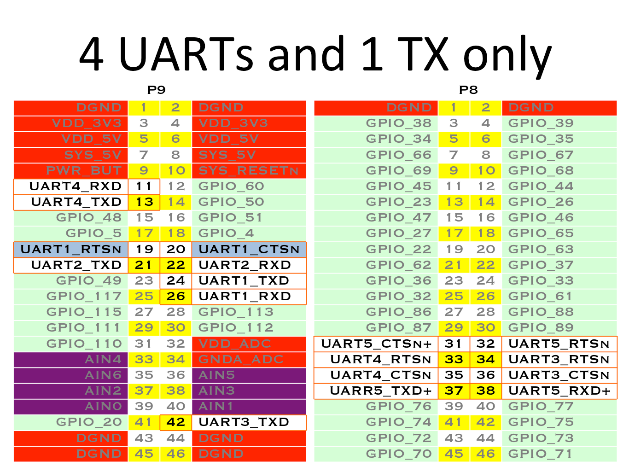
# Install minicom and use minicom

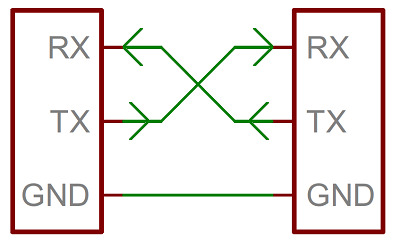
In order to use the console to send commands, characters, strings, etc through one of the UART ports, we need an application. The application I chose is called: minicom.

Minicom should not be able to be installed before the “apt update” command. But since we have done that step we should be able to install minicom by running the command “sudo apt install minicom”. “app install” will look for an application (in this case “minicom” and if it’s in the catalog, it will download it and install it.

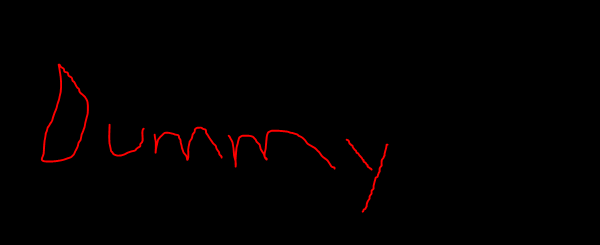
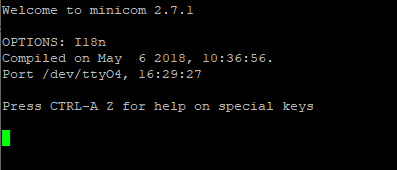
1. Enter the command: “sudo apt install minicom”
2. You will get alerted on the space required, and what additional apps will be installed to install minicom. Press “y” and then “enter” to accept.

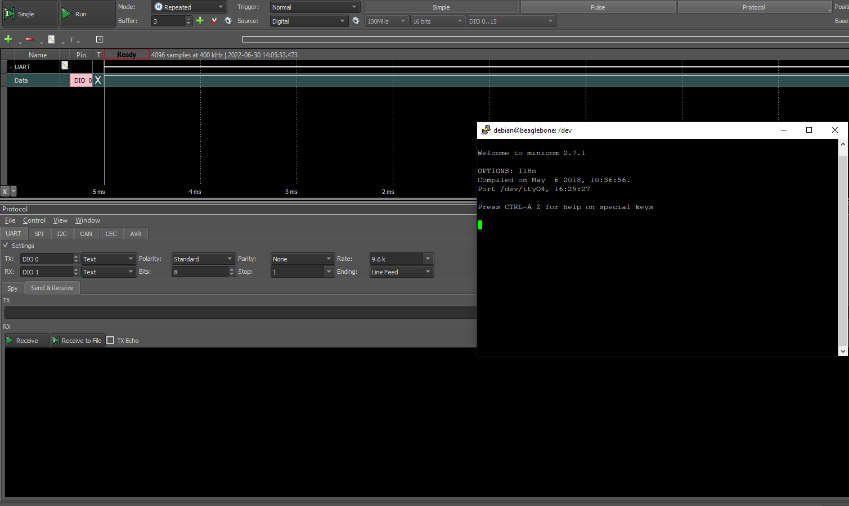
# UART explanation and using minicom

To test minicom, we need to select a UART port to use on the BeagleBone. The board has 4 Full duplex UARTs plus one that only has the TX pin available. For this test I will be using UART4. The header pin order and location can be seen as follows:

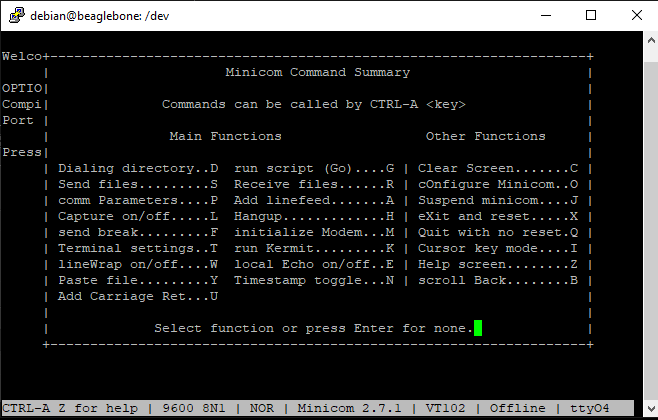
UART4 pins are located on header P9 on pins 11 and 13. RXD (receiving channel) is on pin 11. And TXD (transmitting channel) is on pin 13. In order for the beagle bone to talk to other development boards or devices, both must have a common reference voltage, which we call “Ground”. Ground is labeled as “DGND” (digital ground) and can be found on several pins: 1, 2, 43, 44, 45, and 46 of header “P9”. And on pins: 1 and 2 of header “P8”. I will be testing the UART4 port and minicom by using a logic analyzer which can see the digital levels on the pins and record them in a wave form. One of the functions of this logic analyzer, is to check on protocol in order to debug developments. I will be using both. A UART-to-USB adapter may be used to test the system. On UART communication, the TX pin of one of the devices must be connected to the RX pin of the other device and vice-versa. IMPORTANT!!! The BeagleBone logic level is from 0V to 3.3V, applying voltage higher than 3.3V on any pin (1.8V on analog pins) will damage the board. If the other device you are communicating with is higher than 3.3V, use shift level converter or use a voltage divider to convert the higher voltage to 3.3v.

Since the way we’ll be using the UART protocol is in an asynchronous form, both devices communicating must agree on several specification which are: Baud Rate (speed of transfer), number of bits to transfer per transaction, if there will be a parity bit, how many stop bits will be present (bit that tell the receiving device that the last bit has been sent and transaction is over; sometimes polarity can be chosen, meaning if the line will be logic “high” or logic “low” while idling.

* 1. Connect ground signals together. Connect the BeagleBone’s UART4 TX line to the logic analyzer receiving pin. And connect UART4 RX to the sending pin of the logic analyzer.
  2. Run the command: “minicom -b 9600 -D /dev/ttyO4” to open minicom and setup the communication settings. “minicom” is the application we are running for UART communication. “-b 9600” sets the baudrate to 9600. And “-D /dev/ttyO4” set that the device we are to UART4 on the BeagleBone.
  3. Once minicom is running try typing letter or number, we won’t be able to see anything we type unless we enable the “echo” function.

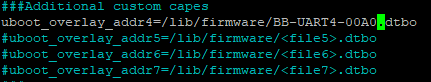


* 1. NOTHING HAPPENED! Let’s fix it. Exit minicom by pressing “Ctrl” + “a” on your keyboard and then “z”. This will open up a menu, we need to select “eXit and reset”, let’s press “x” on the keyboard. Then press “y” to confirm the exit.



# Enable the Device tree overlay for UART communication port

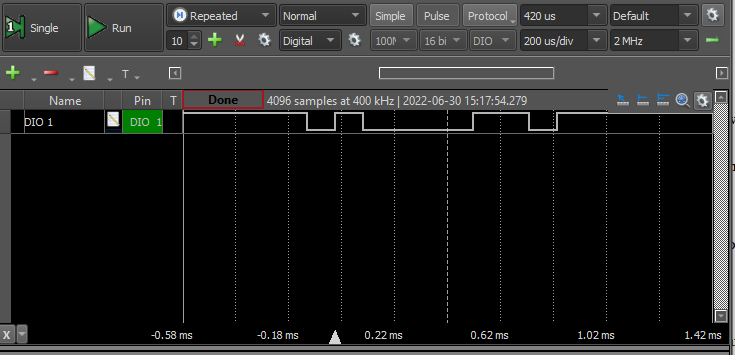
In order to use the UART4 communication, we need to enable it by editing the uEnv.txt. Just like we disabled the HDMI and the emmc uboot overlay. We need to enable UART4 uboot overlay. Device Tree is a binary file that is read at boot time and sets the functions of each pin. This has to be done since each pin can have more than one function. In order to change the functionality of each pin without recompiling the whole Device Tree, Device Tree Overlays have been created. These overlays will modify the Device Tree at boot time, and configure the pins appropriately and load the driver for such function if necessary. There are currently several overlays already compiled and available in the BeagleBone’s directory: /lib/firmware/.

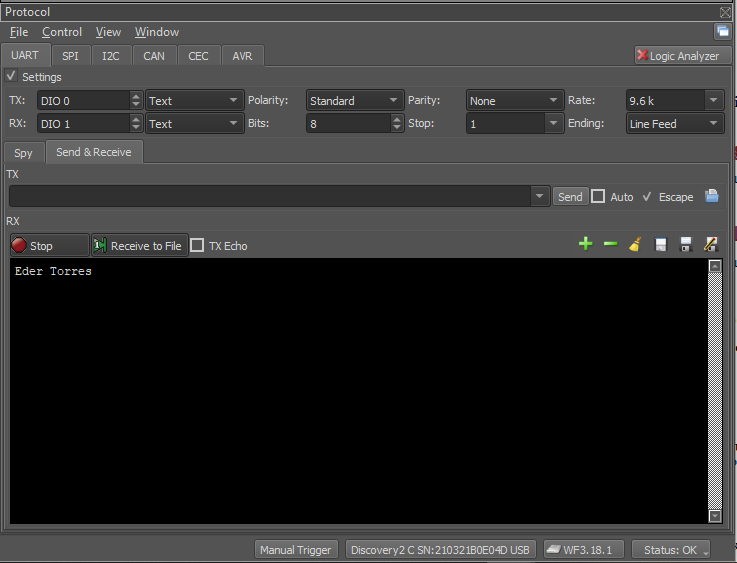
* 1. Type the command: “sudo nano /boot/uEnv.txt” to edit the uEnv.txt file with “nano”.
  2. Modify the line that says “#uboot\_overlay\_addr4==lib/firmware/<file4>.dtbo” to “uboot\_overlay\_addr4=/lib/firmware/BB-UART1-00A0.dtbo”
  3. Press “Ctrl” + “x” on your keyboard. Then “y” and then “enter” to save the changes to the same file.
  4. Reboot the board.

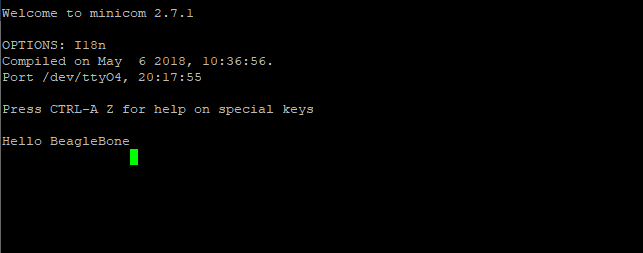
# Sending data with minicom through UART4

Now we are ready to use minicom to send data through UART. Let’s open minicom with the settings to use the serial communication at a baud of 9600 and use the UART4 which is the “TTYO4”.

1. Type: “minicom -b 9600 -D /dev/ttyO4”
2. As you type in the, the logic analyzer can see the pulses coming out of the TX pin of UART4.
3. To exit minicom press “Ctrl” + “A”, then “z” to enter the menu. Then press “x” to exit. And then “enter” to confirm the exit.

When we pressed “a” while in minicom, the logic analyzer saw the following pulses.

When pressing keys using the Protocol analyzer function and specifying UART. We can see all the data received from the TX pin. I typed “Eder Torres”.

If we type something in the TX section of the protocol analyzer, then press “enter”. We can finally see something showing up the screen of minicom. Notice how the pointer didn’t go back to the beginning of a new line, it just jumped the line.

# References and additional reading

* <https://beagleboard.org/blog/2022-02-15-using-device-tree-overlays-example-on-beaglebone-cape-add-on-boards>
* <http://www.ofitselfso.com/BeagleNotes/Disabling_The_EMMC_Memory_On_The_Beaglebone_Black.php>
* http://www.ofitselfso.com/BeagleNotes/Beaglebone\_Black\_And\_Device\_Tree\_Overlays.php