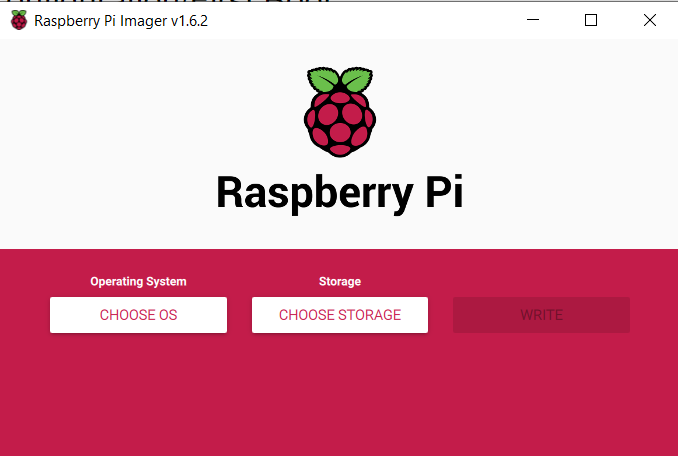
To configure the RaspberryPi Zero w for the first time, a microSD card and SD card adapter are required, as well as the official RaspberryPi Imager v1.6.2 available at:

<https://www.raspberrypi.com/software/>



With the imager downloaded and the microSD card inserted into a computer, the RaspberryPi Imager software was run with the following method:

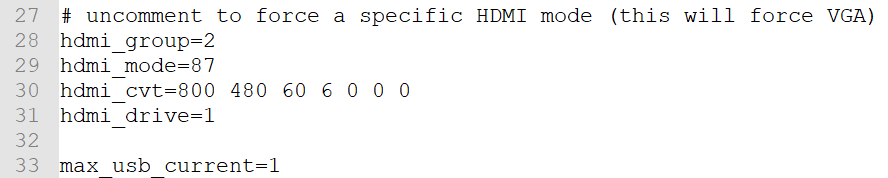
* Choose OS > Raspberry Pi OS(other) > Raspberry PI OS Full (32-bit) (Released 2021-05-07)
* Choose Storage > microSD card
* Format

The process of flashing the Raspberrian image to the microSD takes around 3-5minutes on average, but it is important to make sure that the “OS Full” option is chosen when selecting the OS - otherwise the RasPi will not have a desktop environment when connected to the HDMI display.

Once the Image is flashed to the microSD card, the Imager will ‘eject’ the card from being read by the computer. To modify the config files before interacting with the RasPi, the SD card needs to be taken out and put back into the computer. In our case, the microSD card appears as “boot (D:)”.



If using the 5” TFT HDMI screen from Adafruit, the “config.txt” file needs to be adjusted so that the screen is properly utilized by the RasPi. All adjustments to files on the SD card need to be done with an editor such as Notepad++, as standard text editors like Notepad are prone to inserting invisible characters that can and will cause errors when being read by the RasPi’s Linux-based system. The proper adjustments are shown below.



When working with the RaspberryPi Zero w (henceforth ‘RasPi’), there are two methods for interacting with the device: “headless”, and a method involving SSH. We configured the RasPi using the ‘headless’ method, which involves the usage of a Bluetooth mouse and keyboard (we utilized an android smartphone and the application “Serverless Bluetooth Keyboard & Mouse for PC/Phone” created by AppGround IO - however this specific application was priced at $4.99 before tax.

For the headless configuration, the microSD card was taken directly out from the computer with the only modifications being the TFT HDMI screen requirements. With the microSD card placed into the RasPi, the TFT HDMI screen was connected via microHDMI and microUSB to the RasPi, and the ‘PWR’ (power) microUSB port of the RasPi was connected to a power source (we used the USB port on the computer and encountered no issues). Once powered, the HDMI screen became illuminated, and 2-3 minutes were given for the RasPi to fully boot up (successive boot ups require much less time to complete).

When fully booted up, the HDMI screen will display its default background image (shown below).



The HDMI screen will then present the user with basic configuration options such as selecting the country for WiFi connections (we picked “US”), and an option to configure a connection to a WPA/WPA2 internet - but this option should be skipped if connecting to the “TAMU\_WiFi” oncampus network because the default networking capabilities present on the vanilla Image are not easily configurable to work on the WPA-Enterprise network. More information on this topic will follow later in this document. Interacting with the RasPi was possible due to the touchscreen capabilities of the specific 5” TFT HDMI screen we utilized from Adafruit.

Once the basic configuration settings have been input, the user should make use of the Bluetooth functionality of the RasPi by clicking the Bluetooth symbol, followed by “Add Device”. In the method we used to connect our android smartphone to the RasPi, we set the smartphone to be detectable by Bluetooth - and the phone’s name appeared momentarily as a selectable option on the HDMI screen. When selected, the devices will attempt to connect, and a successful communication will cause the same random number to appear on both devices, followed by a dialog box confirming that the numbers are the same. If both numbers are the same, then confirm the dialogue box. For our purposes, the smartphone served as a mouse and keyboard and this was achieved via “Serverless Bluetooth Keyboard & Mouse for PC/Phone” created by AppGround IO found on the Google Play Store (premium version was $4.99). This connection allows the RasPi to be supplied with simulated keyboard input and more responsive cursor control.

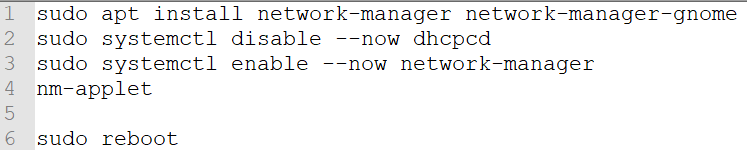
*(The default bluetooth capabilities on the freshly flashed image do not support BLE (Bluetooth Low Energy), but the documentation to get around this is included in the “Raspberry Pi BLE Setup” document).*

The next step for configuration is to establish a WiFi connection to the TAMU\_WiFi network. The default networking abilities on the installed image are not easily adjustable to allow connection to WPA-E networks, so a different network manager application was installed to the RasPi. One thing to note however is that mobile hotspots and non WPA-E networks are easily accessible from the default network manager, so we connected the RasPi to the mobile hotspot of the android smartphone so that an internet connection could be temporarily utilized to install packages.

Once connected, the terminal icon (image below) was selected so that commands could be used to acquire, install, and utilize the ‘network manager’ module.



The code is entered as follows:



These lines of code will first install the standard linux network manager instead of the pre-packaged version installed with Raspbian, then the system is told to stop using the dhcpcd module (as it causes conflicts). With the dhcpcd module disabled, the linux network manager is enabled, and the installed packages are ‘executed’ as an applet. In simple terms and a brief, generalized explanation - applets are visible on the ‘desktop’ display of the HDMI screen in the same way that the terminal icon, internet icon, clock, etc., appear. Lastly, the RasPi will reboot to that all of the configurations up to this point will be put into effect.

Once the device reboots and the smartphone is reconnected (reconnecting is done by just hitting “connect” on the smartphone), the newly created icon for the linux network manager will appear next to the Bluetooth icon. Click this icon, and select “TAMU\_WiFi”, which will bring up an overly-large dialogue box with numerous settings. Before explaining which settings to modify, it is important to note that the dialogue box is too large for the display and will present issues in ‘scrolling’ up and down - this is circumvented by holding the ALT key then clicking and dragging.

The configuration is as follows:

* Wi-Fi security: WPA & WPA2 Enterprise
* Authentication: Protected EAP (PEAP)
* Anonymous identity: (leave blank)
* Domain: net.tamu.edu
* CA certificate: Select from file…
  + - Navigate to /etc/ssl/certs and select the “USERTrust\_RSA\_Certification\_Authority.perm” file
* PEAP version: Version 1
* Inner authentication: MSCHAPv2
* Username: <user NetID>
* Password: <user NetID password>
  + - Username and password values are the same values that students use to sign-in to school computers, Howdy, Canvas, etc.

If the connection fails or other issues arise, enter the command terminal and execute the following command:

* sudo service network-manager restart

If everything went successfully, the RaPi will have internet access (our model had no issues that weren’t resolved by executing the above command).