

# **DRIFTER GPC SETUP PROCEDURE**

This procedure outlines a protocol for configuring the FOX D27 GPC on the Drifter platform for normal use.

[https://www.acmesystems.it/doc\\_foxd27](https://www.acmesystems.it/doc_foxd27)

1. Copy the image from a working image.
  - a. `sudo dd if=/dev/sdb of=/home/eric/Desktop/fox_card.img bs=4M status=progress`
  - b. `sudo dd if=/home/eric/Desktop/fox_card.img of=/dev/sdb bs=4M status=progress`
2. Plug in TTL UART into serial console port header. 115200, 8,N,1
3. Set shutdown switch to 5V and plug in USB port to power board.
4. Login is user: acme/Password: acmesystems
5. repartition to fill card
  - a. `sudo fdisk -l`
  - b. `sudo fdisk -l /dev/mmcblk0`
  - c. `sudo apt update`
  - d. `sudo apt install parted`
  - e. `sudo parted /dev/mmcblk0`
    - i. (parted) `resizepart 2`
      1. 100%
    - ii. 'q' to quit
  - f. resize the filesystem
    - i. `sudo resize2fs /dev/mmcblk0p2`
  - g. `sudo reboot`
6. Install python
  - a. `sudo apt update`
  - b. `sudo apt install python3-libgpiod`
7. Set a fixed IP address
  - a. edit /etc/network/interfaces.d/eth0 with "`sudo nano /etc/network/interfaces.d/eth0`"
  - b. Comment out the code that's there with '#' on the first four lines.
  - c. Add the following code:

```
auto eth0
iface eth0 inet static
address 192.168.9.1
netmask 255.255.255.0
gateway 192.168.9.90
```
  - d. reboot by running 'sudo halt' and pulling power once shut down followed by repowering.
  - e. verify ip address is 192.168.9.1 with 'sudo ifconfig'
    - i. Note sometimes I set to 192.168.8.207 (default gateway 192.168.8.1) to get on internet and be on the same subnet as the computer.
8. Set up an ssh account
  - a. `sudo adduser ssh`
  - b. `sudo passwd ssh` (set to Jameswebb18-)

- c. `sudo usermod -a -G sudo,users ssr`
- d. `sudo halt` and log back in as `ssr`
- 9. Install Samba
  - a. `mkdir Share`
  - b. `sudo apt update`
  - c. `sudo apt upgrade`
  - d. `sudo apt install samba`
  - e. `whereis samba` (check if successful) should output `samba: /usr/sbin/samba`
    - i. Create a directory that samba will point to eg. `sudo mkdir /home/ssr/Share`
  - f. edit configuration file in `/etc/samba/smb.conf` with "`sudo nano /etc/samba/smb.conf`"
  - g. Add the following at the beginning of the file after `[global]`

```
[global]
    oplocks = no
    level2 oplocks = no
```

```
[smbashare]
    comment = Samba on Ubuntu
    path = /home/username/smbashare
    read only = no
    browsable = yes
```

```
[smbashare]
    comment = Samba on Ubuntu
    path = /home/ssr/Share
    read only = no
    browsable = yes
    oplocks = no
    level2 oplocks = no
```

- h. restart samba for configuration to take effect "`sudo service smbd restart`"
- i. UPdate firewall rules to allow samba traffic "`sudo ufw allow samba`" (probably don't need this'
- j. give read write access to all users for the mapped directory using `chmod 777` eg. "`sudo chmod 777 /home/ssr/Share`"
- k. set the samba username and password to the system account "`sudo smbpasswd -a ssr`"
- l. reboot with '`sudo halt`' and pull power and restart
- m. Test Samba
  - i. With an ethernet cable connect a laptop
  - ii. Disable WiFi
  - iii. On Laptop Configure ethernet settings IPV4

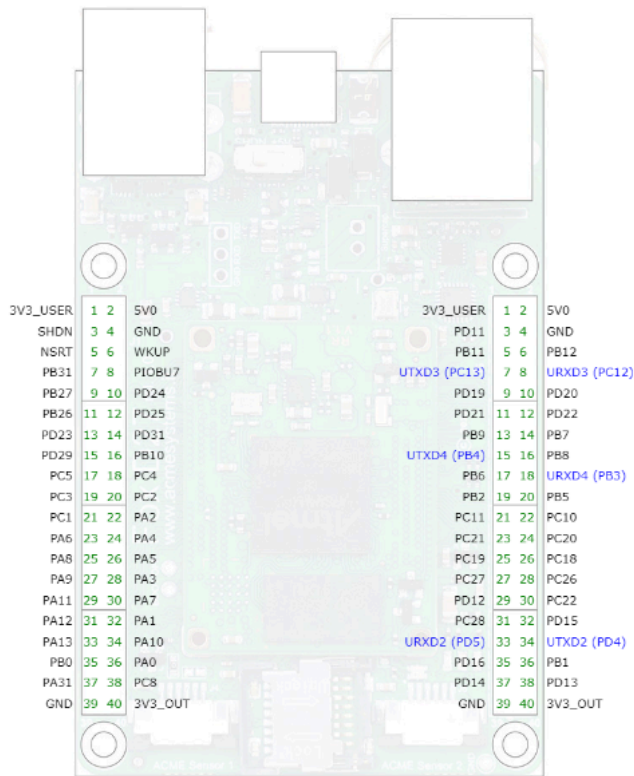
1. Use Following
    - a. IP Address 192.168.9.9
    - b. Subnet Mask: 255.255.255.0
    - c. Default Gateway: 192.168.9.90
  - iv. In Putty connect to 192.168.9.1 (Default Drifter IP address) or in linux ssh  
ssr@192.168.9.1
  - v. Map Network Drive \\192.168.9.1\sambashare
10. Install libgpiod
  - a. `sudo apt update`
  - b. `sudo apt install python3-libgpiod`
11. Install minicom
  - a. `sudo apt update`
  - b. `sudo apt install -y minicom`
  - c. `sudo minicom -s`
12. Reconfigure the Device Tree
13. Install the Arm Crossplatform Toolchain ON A PC with UBUNTU or in VirtualBox
  - a. `sudo apt update`
  - b. `sudo apt install libc6-armel-cross libc6-dev-armel-cross binutils-arm-linux-gnueabi libncurses5-dev build-essential bison flex libssl-dev bc`
  - c. `sudo apt install gcc-arm-linux-gnueabi g++-arm-linux-gnueabi`
  - d. Install SSH client to allow copying files to board
    - i. `sudo apt update`
    - ii. `sudo apt install openssh-client`
    - iii. Test connection with `sudo ssh ssr@192.168.8.217`
    - iv. 'exit' to close ssh
      1. to copy files to the linux device via ssh use
        - a. `scp hello ssr@[your_board_ip]:/home/ssr/hello`
  - e. Install the kernel resources
    - i. `wget https://www.kernel.org/pub/linux/kernel/v4.x/linux-4.19.134.tar.xz`
    - ii. `tar xvfJ linux-4.19.134.tar.xz`
    - iii. `cd linux-4.19.134`
  - f. Install git
    - i. `sudo apt update`
    - ii. `sudo apt install git`
  - g. Install a local git repository for acme files
    - i. `git init; git add .; git commit -m "Linux vanilla"; git branch acme; git checkout acme`
    - ii. If git credentials not set up set them up (eg. eberkenpas...)
    - iii. `wget https://raw.githubusercontent.com/AcmeSystems/acmepatches/master/linux-4.19.x.patch`
    - iv. `patch -p1 < linux-4.19.x.patch`
  - h. `make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi-acme-roadrunner-10uart.dtb`

- i. copy acme-roadrunner-10uart.dtb to /boot/ directory on the FOX D27
    - i. scp acme-roadrunner-10uart.dtb:/boot/acme-roadrunner-10uart.dtb
  - j. backup acme-roadrunner.dtb on the Roadrunner platform
    - i. ssh ssr@192.168.8.217
    - ii. cd /boot
    - iii. sudo cp acme-roadrunner.dtb acme-roadrunner-original.dtb
  - k. overwrite acme-roadrunner.dtb with the new one
    - i. sudo cp acme-roadrunner-10uart.dtb acme-roadrunner.dtb
  - l. reboot platform
    - i. sudo reboot
  - m. log back into platform
- 14. Check comms with drifter controller
  - a. sudo minicom -o -b 57600 -D /dev/ttyS2
- 15. Install serial python libraries
  - a. Download pyserial
    - i. curl -O <https://files.pythonhosted.org/packages/1e/7d/ae3f0a63f41e4d2f6cb66a5b57197850f919f59e558159a4dd3a818f5082/pyserial-3.5.tar.gz>
    - ii. tar -xzf pyserial-3.5.tar.gz
    - iii. cd pyserial-3.5
    - iv. sudo python3 setup.py install
    - v. Check version python3
      - 1. import serial
      - 2. serial.VERSION
    - vi. install GPIOD library for GPIOs
      - 1. sudo apt update
      - 2. sudo apt install python3-libgpiod
  - b. Add current user to dialout to give serial permission
    - i. sudo usermod -a -G dialout \$USER
    - ii. reboot for new settings to take effect
      - 1. sudo reboot
- 16. Make ssr a root user that doesn't need to enter a password for a sudo call
  - a. sudo visudo
  - b. Add the following to the last line:
    - i. ssr ALL=(ALL) NOPASSWD: ALL
  - c. Check if settings are good by using:
    - i. sudo -u ssr sudo whoami
- 17. Configure GPCstart.py to run on boot.
  - a. copy the 'gpc\_start.service' file to the /etc/systemd/system/ folder
    - i. go to the driftersystem folder
    - ii. sudo cp gpc\_start.service /etc/systemd/system/gpc\_start.service
  - b. Enable and Start the service
    - i. Reload the systemd manager to recognize the new service
      - 1. sudo systemctl daemon-reload

- ii. Enable the service so that it starts on boot
    - 1. `sudo systemctl enable gpc_start.service`
    - 2. `sudo systemctl disable gpc_start.service` (to disable)
  - iii. Start the service to immediately test it
    - 1. `sudo systemctl start gpc_start.service`
    - 2. `sudo systemctl stop gpc_start.service`
  - iv. Check the status of your service to verify it is actively running
    - 1. `sudo systemctl status gpc_start.service`
18. Create the "Data" folder to save all data files.
- a. `cd /home/ssr/Share`
  - b. `sudo mkdir Data`
  - c. `sudo chmod -R 777 /home/ssr/Share/Data`
19. Configure "drifterterm" for execution (allows user to enter terminal mode on the drifter controller).
- a. Go to driftersystem directory
    - i. `cd /home/ssr/Share/driftersystem`
  - b. Make the drifterterm file executable
    - i. `sudo chmod +x drifterterm`
  - c.

**ttyS3**

# Pinout FOX Board D27



- ☐ UART0 IOSet1
- ☐ UART1 IOSet1 (Debug port)
- ☒ UART2 IOSet1 - ☐ IOSet2 - ☐ IOSet3
- ☒ UART3 IOSet1 | ☐ UART3 IOSet2 | ☐ UART3 IOSet3
- ☒ UART4 IOSet1
- ☐ FLEXCOM0 IOSet1
- ☐ FLEXCOM1 IOSet1
- ☐ FLEXCOM2 IOSet1
- ☐ FLEXCOM3 IOSet1
- ☐ FLEXCOM4 IOSet1
- ☐ TW0 (PD21,PD22)
- ☐ TW0 (PB31,PC0)
- ☐ TW0 (PC27,PC28)
- ☐ TW0 FLEXCOM0 (PC29,PC30)
- ☐ TW1 (PD4,PD5)
- ☐ TW1 (PD19,PD20)
- ☐ TW1 (PC6,PC7)
- ☐ SPIO (PA15,PA16,PA14,PA17,PA18,PA19,PA20)
- ☐ SPIO (PB0,PA31,PB1,PA30,PA29,PA27,PA28)

