



## Lab – Using USB Gadget drivers

Objective: Configure and compile a kernel for USB gadget usage. Test and experiment with serial, Ethernet and block USB gadget drivers.

### Setup

Go to the `/home/<user>/felabs/usb gadget/usage/` directory.

### Kernel configuration, compilation and installation

Grab a recent kernel version and configure it with the configuration file provided in the `data/` directory.

Run your favorite kernel configuration utility (`xconfig`, `gconfig` or `menuconfig`) and add support for USB Gadget. It should include:

- Core USB Gadget support, statically compiled
- AT91 USB Gadget controller support, statically compiled
- Serial, Ethernet and storage USB Gadget drivers, compiled as modules

Compile your kernel with `make`, then generate the `uImage` with `make uImage` and install the modules in the root filesystem (in the `nfsroot/` directory) using `make modules_install`. Hint: don't run `make modules_install` as root, and use the `INSTALL_MOD_PATH` variable.

### Boot the board

Boot the Calao board with the root filesystem mounted over NFS. The filesystem in `/home/<user>/felabs/usb gadget/usage/nfsroot/` should be mounted as the root filesystem. It should contain the USB gadget driver modules in `/lib/modules/<kernelversion>/`.

### Using the Ethernet gadget driver

Load the `g_ether` kernel module with the `modprobe` command. You should see a new `usb0` network interface with `ifconfig`. Set an IP address such as `192.168.42.2` to this new interface. Then connect the Calao board to your development workstation using the USB device cable. Ubuntu will automatically detect the new device, load the driver, and create the network interface. You can then use NetworkManager to configure a static IP address such as `192.168.42.1` to this new `usb0` interface.

We have now established an Ethernet over USB connection, that can be used as usual. You can for example test it with `ping`. Because our root filesystem is mounted over NFS, we cannot disconnect the Ethernet cable, but if our system was flashed into the NAND of the Calao board, then we could remove this Ethernet cable.

Using the storage gadget driver

On your host workstation, prepare a filesystem image that will contain the filesystem made visible by the Calao board through the USB connection. To do so:



- Create an empty file of 10 megabytes  
`dd if=/dev/zero of=disk.img bs=1M count=10`
- Format it as FAT  
`mkfs.vfat -F disk.img`
- Transfer it to the root directory of the Calao board filesystem  
`mv disk.img nfsroot/root/`

Now, on the Calao board, load the USB Storage gadget driver and tell it to use disk.img as the file containing the data to export:

```
modprobe g_file_storage file=/root/disk.img
```

Then connect the device to your development workstation. Ubuntu will automatically detect the device, load the driver, mount the filesystem and display a nice list of the files present in the device.

### Using the serial gadget driver

Load the `g_serial` kernel module on the Calao board. Then, create the `/dev/ttyGS0` device file that correspond to the major and minor number as given by the file in `sysfs` (see the training slides for details).

Now, connect the Calao board to your development workstation using the USB device cable. Your Linux system will automatically detect the new USB device cable and create the `/dev/ttyACM0` device file.

Now, to make a simple test:

- Run `cat /dev/ttyACM0` on your workstation
- And run `echo "hello world" > /dev/ttyGS0` on your device

You can also test the opposite communication (cat on the device and echo on your workstation). These are very simple tests, but they show that the serial communication is working. It means that we could run an application on our device that opens `/dev/ttyGS0` and makes features available to the workstation through the serial port.