## Procedure to Install a Kernel Module for the DHT11 Sensor

Modern kernels now use the device tree overlays to manage kernel modules. This is more efficient approach to handle the problem of multiple of drivers and interrupts, and also allow HAT or SHIELD attachments to autoconfigure themselves.

Briefly, a device tree is a description of the hardware resources of a system in a hierarchical manner. Nodes are hardware components with associated properties. For example, a register component with arrays of bytes, containing either strings or numbers, etc. The location of nodes within the device tree can be described in a C like syntax using paths with slash separators, like file systems. A hardware device, even as complex as an SoC, can be described with a dts file. However, devices like Raspberry pi have a multitude of optional components or HATs, which need to coexist sharing GPIO pins and the like. To alleviate this complexity, optional parts can be described using overlays, that is device tree fragments that can be attached to the main trunk of the entire system level device tree. More details on device trees and overlays are in the documentation.

```
https://www.raspberrypi.org/documentation
```

Kernel modules will need to be enabled by their device tree overlays before compilation and loading.
We will show this process using the DHT11 temperature and humidity sensor example.
A device driver source for DHT11 is provided by the attached file `dht11.c`
A device tree overlay for DHT11 is also provided by the attached file `dht11.dts`
This overlay file needs to be compiled into a dtb binary which then needs to be installed in the boot directory of the machine.

Please follow the following steps to reconfigure your machine using device tree overlays.

1. Compile the overlay using the built-in `dtc` compiler as follows

```
/usr/bin/dtc  -I dts -O dtb -o dht11-overlay.dtb -b 0 -@ dht11.dts
```

2. Copy the overlay file into the overlays boot subdirectory

```
sudo cp -p dht11-overlay.dtb /boot/overlays
```

3. Edit the boot configuration file `/boot/config.txt` adding the following (pin 4 can be changed)

```
dtoverlay=dht11,gpiopin=4
```

4. Connect DHT11 sensor with 3.3V, GND, and GPIO4 on Pi

5. Reboot the machine

6a. Check if the module dht11 is already loaded

```
lsmod | grep dht11
```

6b. If dht11 is not loaded, you will need to compile the `dht11.c` file using a simple Makefile

and then manually load it with the `insmod` command

```
sudo insmod dht11.ko
```

7. Test the DHT11 sensor that it works and check its accuracy

8. Perform some temperature readings using your newly loaded device driver for the DHT11 sensor.

9. Compare your kernel module approach to your previous solution in terms of timing and accuracy.

10. Your report should reflect your experience and results from the above steps.

Be careful when applying the above commands and configurations because any error may make your machine unbootable. If this occurs, you will need to restore the machine back in order. For that, you will need to extract the micro SD card, plug it in another computer, and correct the configuration files using a text editor.