Device Driver Assignment

**Consist of three parts:**

**Part 1 – Device Driver Source Code name motor1.c**

**Part 2 – Steps to prepare Device Driver motor1.ko**

**Part 3 – C Program testMotor.c**

## Part 1 – Device Driver Source Code name motor1.c

Supports these services:

static ssize\_t motor\_stop(struct file \*file,)

static ssize\_t motor\_rotate(struct file \*file, int direction)

static int motor\_open(struct inode \*inode, struct file \*file)

static int motor\_release(struct inode \*inode, struct file \*file)

static ssize\_t motor\_read(struct file \*file, char \*buf, size\_t count, loff\_t \*ptr)

static ssize\_t motor\_write(struct file \*file, const char \*buf, size\_t count, loff\_t \* ppos)

static int motor\_ioctl(struct inode \*inode, struct file \*file, unsigned int cmd, unsigned long arg)

## Part 2 – Steps to prepare Device Driver motor1.ko

1. Create a directory at and add what?
2. Configure the Kconfig file so that he device module can be loaded (statically, dynamically, or not at all) – identify which kconfig file and what you would place in it and where?
3. Which Makefile to modify and how?
4. How to compile the driver?
5. Where to place the built driver, either manually or during the make process
6. To load the module once in resides in the modules/linux-3.8 directory and offload:
7. Manually use:
8. Automatically as part of init process runlevel 3
9. Make device node for device driver so that the application binds its request to a specific driver.

mknod /dev/motor1 c 234 0

## Part 3 – C Program testMotor.c

Open the device Motor1, issue a start command to rotate in the left direction, stop the motor and close the device.