SD Card Feature Implementation

1. Kernel Implementation
2. Module Insertion
3. Library for api
4. Udev rule

Kernel Implementation

1. To stop the sdcard initlization at boot time .In the file **arch/arm/mach-mx6/board-mx6sl_evk.c**

   Comment the follwoing line in the function **static void __init mx6_evk_init(void)**

```
         imx6_init_fec(fec_data);
         platform_device_register(&evk_vmmc_reg_devices);
         imx6q_add_sdhci_usdhc_imx(0, &mx6_evk_sd1_data);
        imx6q_add_sdhci_usdhc_imx(1, &mx6_evk_sd2_data);
 //       imx6q_add_sdhci_usdhc_imx(2, &mx6_evk_sd3_data);

mx6_evk_init_usb(); imx6q_add_otp();
```

2. After the function, An Export symbol should be added

```
void sdio_sd(void)
 {
         imx6q_add_sdhci_usdhc_imx(2, &mx6_evk_sd3_data);
 }

 EXPORT_SYMBOL(sdio_sd);
```

3. the core file should be changed . A device Attr should be added

- the following lines should be added at the starting of the file **drivers/mmc/core/sd.c**

```
   #define MMC_LOCK_MODE_LOCK    (1<<2)
   #define MMC_LOCK_MODE_ERASE   (1<<3)
   #define MMC_LOCK_MODE_UNLOCK  (0<<2)
   #define MMC_LOCK_MODE_CLR_PWD (1<<1)
   #define MMC_LOCK_MODE_SET_PWD (1<<0)

   char pdbuf[30]="Ca$li@p%";

   int mmc_lock_unlock(struct mmc_card *card, char *password,int mode);
   static ssize_t sdlock_status(struct device *dev, struct device_attribute *att,char
*buf,size_t len);
   static ssize_t sdlock_try(struct device *dev, struct device_attribute *att,const
char *data, size_t len);
   static ssize_t sdunlock_try (struct device *dev,struct device_attribute *att,const
char *data, size_t len);
```

- In the same file,in the function **mmc_sd_setup_card** the following line should be added

```
                    mmc_lock_unlock(card,pdbuf, MMC_LOCK_MODE_UNLOCK);
```

- In the same file, add the attributes to the card

```
 DEVICE_ATTR(unlock, S_IRUGO|S_IWUSR|S_IWUGO, NULL, sdunlock_try);
DEVICE_ATTR(lock, S_IRUGO|S_IWUSR|S_IWUGO, NULL, sdlock_try);
DEVICE_ATTR(status, S_IRUGO|S_IWUSR|S_IWUGO, NULL, sdlock_status);
```

Added the function to the respective attributes(DEVICE_ATTR)

- In the structure **static struct attribute *sd_std_attrs[]** append the following lines

```
        &dev_attr_unlock.attr,
         &dev_attr_lock.attr,
          &dev_attr_status.attr,
```

Module Insertion

A **sd.ko** should be inserted to detect a microsd card

A simple character driver is implemented for the detection of the microsd card

```
extern void sdio_sd(void);

 static int __init sd_pwr_init(void)
 {
         printk("SD Card init called");
         sdio_sd();
         return 0 ;
  }
 static void __exit sd_pwr_exit(void)
  {
  }
  module_init(sd_pwr_init);
  module_exit(sd_pwr_exit);

  MODULE_DESCRIPTION("SDCard Init Driver");
  MODULE_AUTHOR("visiontek.co.in");
  MODULE_LICENSE("GPL");
```

Library

Two api's are provided for the secure mount and unmount

```
secure_mount(void)
        return values:
                0  :success
                -1 :mount failed
                -2 :Sdcard not present

secure_umount(void)
        return values:
                0  :success
                -1 :unmount failed
                -2: sdcard not presnt
```

Udev Rules

A udev rule is added to detect the sdcard at the rum time of a device

A file **sd.rules** is added in the folder **etc/udev/rules.d/sd.rules**

```
        sd.rules

#ACTION=="add",KERNEL=="mmcblk[1-9]", SUBSYSTEM=="block", ATTRS{name}=="?*",
ATTRS{serial}=="?*", ENV{ID_NAME}="$attr{name}", ENV{ID_SERIAL}="$attr{serial}",
SYMLINK+="disk/by-id/mmc-$env{ID_NAME}_$env{ID_SERIAL}",RUN+="/root/sdlock"

ACTION=="add",KERNEL=="mmcblk[1-9]p[1-9]",SUBSYSTEM=="block",ENV{ID_NAME}=="?*",
ENV{ID_SERIAL}=="?*", SYMLINK+="disk/by-id/mmc-$env{ID_NAME}_$env{ID_SERIAL}-part
%n",RUN+="/usr/bin/mmc_sec"
```

Respective two binaries should also be added

This udev rule is used to monitor the sdcard status and lock the sdcard at the time of the insertion