Anahit Sarao

008435583

CMPE 146

Analysis of DiskIO with FATFS Library

1. The lower level architecture for file input and output uses the diskio.c file. The file is dependent on SPI driver, SD card driver, SPI driver semaphore files. The file consists of five main functions. The functions are used to control reading and writing to the disk. The disk can be initialized in two different modes, flash or sd. Flash allows the flash memory to be accessed and SD allows the usage of the SD card port. Within the functions there is a feature known as the SPI lock which allows a mutex to lock all resources for that pertaining I/O. The only way to release the resources is to return a mutex release. This eliminates the possible problem of bus contention on the SPI bus, in other word you cannot I/O to flash and SD at the same time.
2. The numbers for the drives are designated to each SPI bus. A 0 will allow access to the flash memory where 1 will allow access to the SD card port. These parameters can be passed into the functions or used within the terminal task.
3. To be able to read and write using the FATFS library you must use to f\_open function which will reference the proper file object needed. The files containing the functionality are ff.c and ff.h.

Extra Credit

BIT Macro is a fast way to allow bit manipulation. This allows access to different pin functionality in a more efficient way. It can be seen as a different way to achieve bit masking. For the first example of BIT(var).b4=1 allows the fourth bit to be changed based on the base address of var. The two structures allow the manipulation of one bit or two bits. This is useful dependent on the bit size of the register. For the second example BIT(var).b15\_14 = 2 is used to set two bits simultaneously.

