

目的：

- 每一個Thread都有一個唯一的變數，這個變數用來儲存測試過程中所有相關的訊息
- 這個訊息包含Device、Fixture和相關測量儀器的紀錄

方法：

- Equipment裡有4個Singleton Instance:
 1. folderPath->用來儲存Log存放資料夾的路徑，預設路徑是/vault/Equipment_Log ·
 2. thread1EchoColectionStr->用來儲存thread1的Log ·
 3. thread2EchoColectionStr->用來儲存thread2的Log ·
 4. thread3EchoColectionStr->用來儲存thread3的Log ·
 5. thread4EchoColectionStr->用來儲存thread4的Log ·

- Equipment將會有以下幾個Class Method去存取這4個Singleton Instance：

1. 設定存放資料夾的路徑:

```
+(void)setLogFolderPath:(NSString*)path
```

2. 將訊息儲存到指定Thread的Log:

```
+(NSString *)saveLogWithThread:(int)num withFileName:(NSString *) name;
```

3. 儲存指定Thread的Log:

```
+(NSString *)saveLogWithThread:(int)num withFileName:(NSString *) name;
```

4. 清空指定Thread的Log:

```
+(void)clearLogFileWithThread:(int)num
```

5. 讓指定的Thread睡眠，並將睡眠的秒數紀錄在Log:

```
+(void)delayWithThread:(int) num withSecond:(int) second
```

6. 同上，只是單位變成微秒:

```
+(void)delayWithThread:(int) num withMicorSecond:(int) mSecond
```

- 在Equipment宣告一個Property “myThreadIndex”，用於儲存Thread_index：
 1. 如果是1v4，在ini.plist中，“EQUIPMENTS”欄位裡所有item的“USEDFOR”欄位須符合規則“THRD(Num)_(Equipment Name)”，ex:THRD1_FixUART、THRD3_DevUART ·
 2. 透過PlistIO Class的“equipmentsInit” Method抓取“THRD(Num)_(Equipment Name)”字串當中的Num作為myThreadIndex的值，假如是1up的話，myThreadIndex設為1 ·

- 將上述1、2、3的Class Method 包成 Instance Method 方便繼承Equipment的Class使用：

1. 設定存放資料夾的路徑:

```
-(void)setLogFolderPath:(NSString*)path
{
    [Equipments setLogFolderPath:path];
}
```

2. 將訊息儲存到Log:

```
-(void)attachLogFileWithTitle:(NSString*)title withDate:(NSString*)date with
Message:(NSString*)content
{
    [Equipments attachLogFileWithThread:myThreadIndex withTitle:title with
Date:date withMessage:content];
}
```

3. 儲存Log:

```
-(NSString *)saveLogWithFileName:(NSString *) name
{
    return [Equipments saveLogWithThread:myThreadIndex withFileName:name];
}
```

- 取代Device、Fixture...等繼承Equipment的Class當中的writeToDevice和readFromDevice的echoCollectionStr:

1. Device裡writeToDevice的Method:

```
-(BOOL)writeToDevice:(NSString *)uartCmd
{
    if([uart TX:uartCmd])
    {
        [self attachLogFileWithTitle:[NSString stringWithFormat:@"%d",[
self className],[self myThreadIndex]]
            withDate:[Utility getTimeBy24hr]
            withMessage:[NSString stringWithFormat:@"SEND:%@",
uartCmd]];
        usleep(DELAYTIME);
        return TRUE;
    }
    return FALSE;
}
```

2. Device裡readFromDevice的Method:

```
-(NSString *)readFromDevice
{
    NSString *echo=[uart RX];
    if ([echo length]>0)
    {
        [self attachLogFileWithTitle:[NSString stringWithFormat:@"%@" %d",[
self className],[self myThreadIndex]]
            withDate:[Utility getTimeBy24hr]
            withMessage:[NSString stringWithFormat:@"READ:%@",
echo]];
        return echo;
    }
    return @"";
}
```