

```

// ResetProg.cpp - RESET CONTROLLER PROGRAM
//
// MODULE INDEX
// NAME          CONTENTS
// main          MAIN LINE
//
// MAINTENANCE HISTORY
// DATE          PROGRAMMER AND DETAILS
// 04-10-12     JS   Original
//
//-----

#include <ctime>          // C-style operating system time functions
using namespace std;     // Expand the standard namespace
#include "ResetIO.h"     // Reset I/O class declaration

//-----

// MAIN LINE

int
main ()
{
    struct timespec millisec; // A millisecond
    int    milliCnt; // Millisecond count

    // Turn off the power and turn on the power fail warning

    ResetIO_c::SetPowerSwitch (0);
    ResetIO_c::SetPowerWarn (1);

    // Load the millisecond time interval

    millisec.tv_sec = 0;
    millisec.tv_nsec = 1000000;

    // Loop through power failure cycles

    for (;;) {

        // Cycle through power failures during the startup sequence

        do {
            // Wait for the power to come good

            while ( ! ResetIO_c::IsPowerGood() )
                nanosleep (&millisec, 0);

            // Wait for the power to remain good for 3s

            milliCnt = 0;
            while (ResetIO_c::IsPowerGood() && milliCnt < 3000) {
                nanosleep (&millisec, 0);
                milliCnt ++ ;
            }

            // If the power fails, wait for the power to come

```

```
    // good again

} while (milliCnt < 3000);

// Switch the power on and the warning off

ResetIO_c::SetPowerSwitch (1);
ResetIO_c::SetPowerWarn (0);

// Loop until we get a real power failure

do {
    // Wait for a power failure

    while (ResetIO_c::IsPowerGood())
        nanosleep (&millisec, 0);

    // Activate the warning

    ResetIO_c::SetPowerWarn (0);

    // Wait for the power to fail for 20ms

    milliCnt = 0;
    while ( ! ResetIO_c::IsPowerGood() && milliCnt < 20) {
        nanosleep (&millisec, 0);
        milliCnt ++ ;
    }

    // If the power came good, wait for another power
    // failure

} while (milliCnt < 20);

// If the power really failed turn off the power to the
// data logger and restart the power up cycle

ResetIO_c::SetPowerSwitch (0);
}
return 0;
}
```