

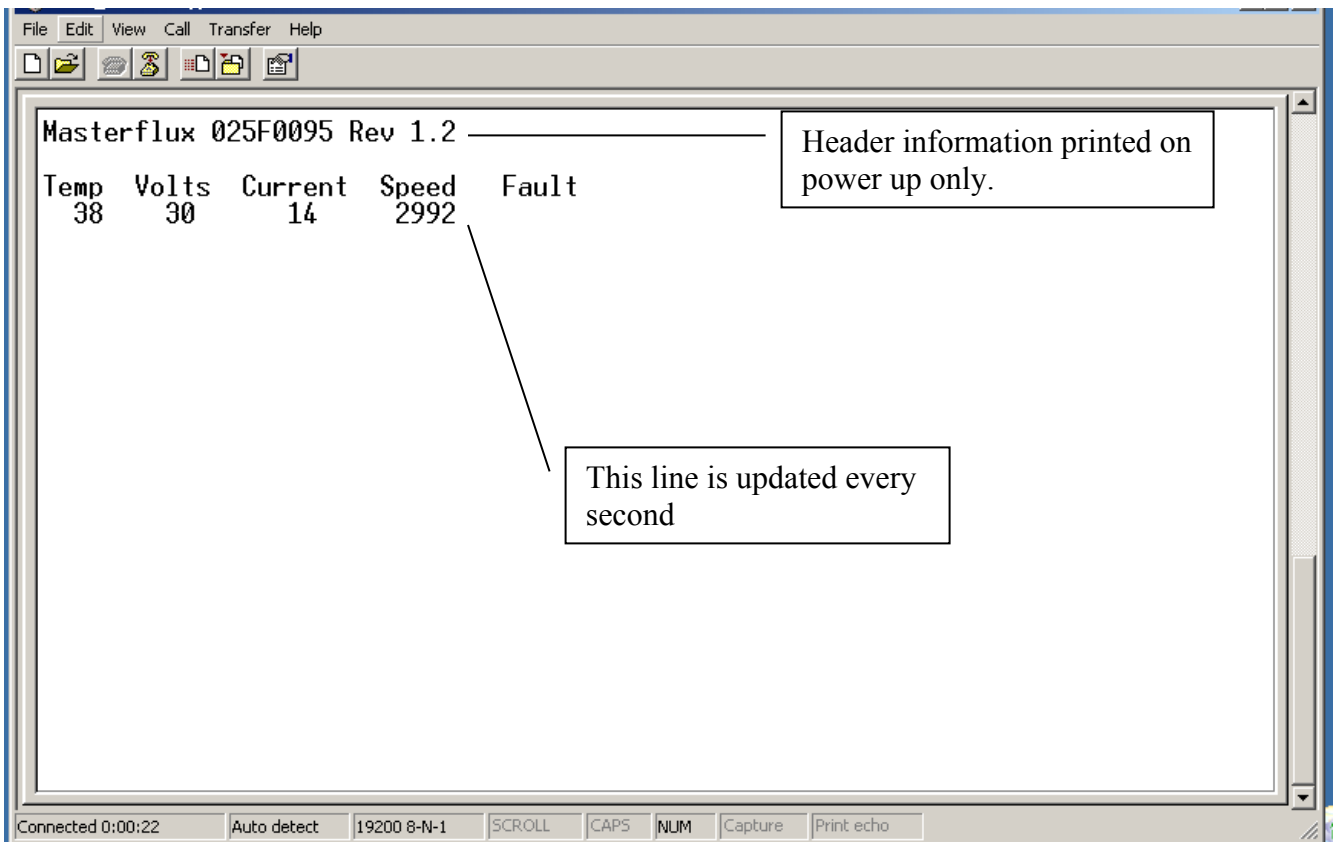
## Serial Port Interface

The serial interface is configured for 19.2 K baud, 8 data bits, 1 stop bit, no parity, and no flow control. The controller will display the header information only on power up then update the following operating parameters once per second over the serial interface. The serial interface is 0 -5V TTL non isolated output.

The possible faults that are displayed are shown below:

OVER CURRENT  
OVER VOLTAGE  
UNDER VOLTAGE  
CONTROLLER OVERHEAT  
MOTOR OVERHEAT  
STALLED  
LOW SPEED  
STARTUP FAILED

Below is a print out of the controller message using hyper terminal. The controller is not faulted in this illustration but would display one of the fault messages above under the fault text below if faulted. Once the fault condition is cleared the fault message will also be cleared.



Below is the #C function that is used to display the controller status.  
The format can be derived from this.

```
void DisplayInputs(void)
{
    int i = 0;

    printf("%4d",Get_ModuleTemperature());
    printf("%6d",Get_DC_BusVolts());
    printf("%8d", Get_DC_BusCurrent());
    printf("%9d", HZ_TO_RPM(Tach));

    // print fault status
    printf(" ");
    if(ValBit(Power_Motor_Status, OVER_CURRENT))
        i = printf("OVER CURRENT");
    else if (ValBit(Power_Motor_Status, OVER_VOLTAGE))
        i = printf("OVER VOLTAGE");
    else if (ValBit(Power_Motor_Status, UNDER_VOLTAGE))
        i = printf("UNDER VOLTAGE");
    else if (ValBit(Power_Motor_Status, OVER_HEAT_MODULE))
        i = printf("MODULE OVERHEAT");
    else if (ValBit(Power_Motor_Status, OVER_HEAT_MOTOR))
        i = printf("MOTOR OVERHEAT");
    else if(ValBit(LatchedFault, MOTOR_STALLED))
        i = printf("STALLED");
    else if(ValBit(LatchedFault, LOW_SPEED_FAULT))
        i = printf("LOW SPEED");
    else if (ValBit(LatchedFault, START_UP_FAILED))
        i = printf("STARTUP FAILED");

    for(;i < 20; i++) // overwrite any previous message
        putchar(' ');

    putchar('\r');
}
```