MuddLogg16 v3 User's Guide

Overview

The MuddLogg16 v3 is a fast, portable data logger that records analog measurements to a microSD card. It contains all the components and options necessary for out-of-the-box logging:

- Selectable number (2 – 16) channels of 16-bit data
- Selectable overall sampling frequency, theoretically up to 450kHz
- Simple configuration
- Supports FAT16 and FAT32 cards
- Wide power supply range: 6V – 20V
- Protection features: power polarity can be reversed up to ±20V without damaging the device, Schottky diodes on analog inputs provide protection up to 500mA

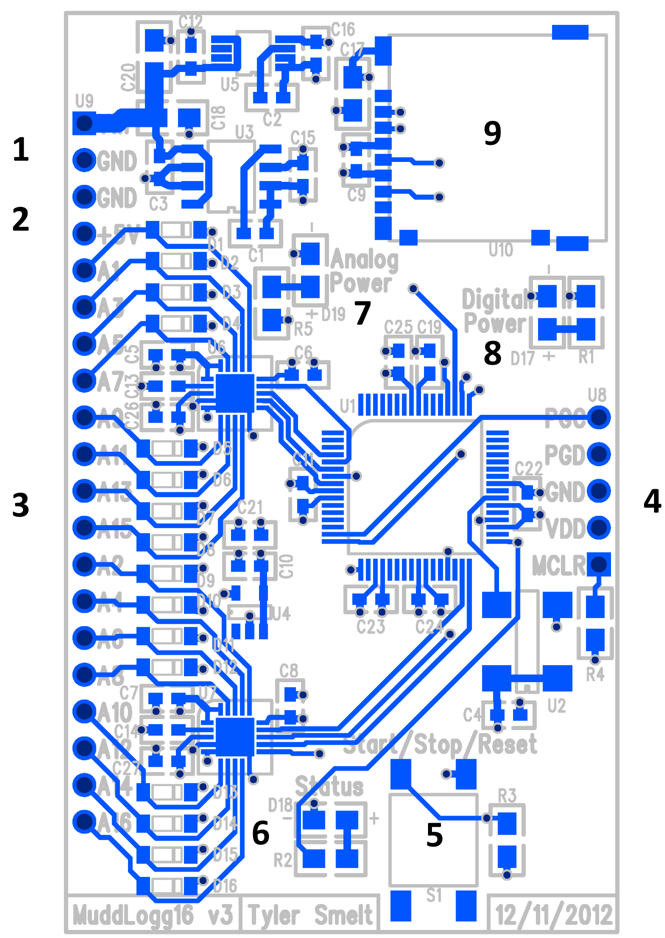

Figure 1. MuddLogg16 v3 PCB layout.

Figure 1 shows the layout of the device. Important components include:

1. Power input pins (Vin and GND)
2. Voltage regulator output pin (+5V)
3. Analog input pins (A1 – A16)
4. Programming pins, including reset pin (MCLR)
5. Start/Stop/Reset button
6. Green Status LED
7. Red Power indicator LED (5V analog section)
8. Red Power indicator LED (3.3V digital section)
9. Push-push microSD card socket

Theory of Operation

The MuddLogg16 saves data to the SD card by progressively filling and writing clusters in a 60KB data buffer. FAT16 and FAT32 file systems are supported. Files in the FAT file system are divided into large clusters, which are further divided into sectors. Sectors contain 512 bytes of data, and the number of sectors per cluster is determined when the card is formatted. Common cluster sizes are 4KB – 16KB. The MuddLogg16 does not support cards formatted with clusters of 32KB or larger, since the 60KB data buffer must be divided into at least two clusters.

Both standard-capacity and high-capacity (SDHC) cards of any speed class are supported. However, the choice of card significantly affects the maximum possible sampling rate. The main limiting factor is the cluster write time, which can vary significantly during operation. Testing during development was carried out with both a 2GB card and an 8GB Class 10 SDHC card.

Analog signals to be logged are measured between a 3.3V reference and GND. If a higher voltage is present on an analog input pin, it will be clamped at ~5.3V by a Schottky diode to protect the ADC inputs. This protection will work unless the source of the overvoltage can provide enough current (500mA) to overload the onboard 5V regulator. If this occurs, the ADCs may be permanently damaged, or may need to be power-cycled before operating correctly again.

Configuration

Configuration options are set in a file named “config.txt” located in the root directory of the card. If this file cannot be found at startup, it will be created with default values. The default contents of the file are:

```
CH = 16
FS = 64000
AUTO = N
TEST = N
```

CH

The "CH" option sets the number of ADC channels to log, from 1 to 16. Since the samples are obtained from a pair of 8-channel ADCs, the number of channels must be even. If an odd value is set, the next-highest even number will actually be used. Keep this in mind when reading the data! If the number is set to a value less than 1 or greater than 16, the default value will be used (16).

FS

The "FS" option selects the overall sampling rate, which is divided evenly between the selected channels. The absolute minimum sampling rate is around 6Hz, limited by the longest period of the timers used by the microprocessor to control sampling. Setting the sampling rate anywhere near this value is not recommended, since the system will continue logging until a full cluster of data is ready to write, which will take a very long time at a low sampling rate. The absolute maximum sampling rate is 450kHz, which is the maximum combined speed of the ADCs. If a value outside these limits is specified, the default value will be used (64kHz).

The maximum sampling rate is dependent on the SD card used. Test out various rates to see what is safe. When errors do occur and data is missing, they will be of two main types: small frequent gaps in the data, and larger infrequent gaps. The appearance of frequent gaps indicates that the chosen sampling rate is near the absolute maximum data transfer rate of the system, and it is constantly struggling to keep up. With a fast card, this limit could approach 400kHz.

Some infrequent gaps may occur even at much lower frequencies, since the duration of SD write operations can vary. In the very worst case, the card may take around half a second to complete a cluster write, though this should not happen often. To guard against this kind of disturbance, it is recommended that the sampling rate should allow for around half a second of data to be held in the data buffer. The buffer holds up to 60KB of data, which corresponds to 30,720 16-bit samples. At a sampling rate around 60kHz, these samples represent data for half a second. The default sampling rate of 64kHz should be sufficiently slow to avoid almost all large gaps, but should be fast enough (4kHz/channel with 16 channels) for most applications. **Setting FS above around 100kHz is not recommended**. Test your desired sampling rate to ensure the data is sampled correctly without gaps.

Note that write operations will be more efficient with larger clusters. However, since the 60KB data buffer is divided into an integer number of clusters, 4KB is the largest cluster size that allows for full buffer utilization, while 16KB clusters only make use of 48KB. For a balance of speed and buffer utilization, a cluster size of 4KB or 8KB (56KB buffer utilization) is recommended.

AUTO

The "AUTO" option selects how the system begins logging. If AUTO = N, logging will not start until the Start/Stop/Reset button is pressed. If AUTO = Y, logging will start immediately after power-on. If there is a risk of power failure during operation, it may be a good idea to set AUTO = Y so that logging can recommence when power is restored. Note that whatever the case, power failure to the SD card will corrupt that previously logged data.

TEST

The "TEST" option selects whether sampling is active. The default value of "N" should be selected for normal operation. When TEST = Y, the system transmits a static array of data as fast as possible, instead of taking samples at the defined rate. This is used to speed-test cards, and can give an idea of a maximum possible sampling rate.

Operation Procedure

To operate the MuddLogg16, insert the SD card and connect power to the Vin (+) and GND (-) pins. The power supply must provide between 6V and 20V, and should be able to provide at least 150mA (average current draw is much lower at around 40mA). The voltage regulators on the board will survive connecting power in reverse, up to ±20V. The +5V pin provides the output of the onboard voltage regulator (up to 500mA) for use in powering other electronics like analog sensors. Do not connect a higher voltage power supply to this pin; this is one of the only ways to completely fry the onboard components.

When the system is powered on, all three LEDs should illuminate. If either red Power LED is off check the voltage regulator (U3 and U5) outputs. If the green Status LED is either constantly on or constantly off after the SD card has been inserted try power-cycling the system, and check the SD card for any messages in ".ERR" files, and re-formatting the SD card.

If AUTO = N, the green LED will begin to flash slowly, indicating that the system is idle. To start logging press the button and hold for about a second. The green LED should start flashing rapidly. If AUTO = Y, logging will begin immediately after power-up.

Press the button again to stop logging. You may need to hold the button for some time, especially if the sampling rate is low. The green LED should go back to flashing slowly, and at this time it is safe to power off or remove the SD card. Alternately, press and hold the button again to restart logging.

Data and Error Files

The MuddLogg16 will create two kinds of files in the root directory of the SD card. ".DAT" files contain binary data, with filenames numbered from DATA0000.DAT to DATA9999.DAT. New data will be placed in the lowest-numbered file that does not yet exist. If any errors occurred during setup, warning messages will be written to a ".ERR" file with the same base filename as the associated ".DAT" file.

Data in the binary files should be interpreted as 16-bit unsigned values in big-endian format, where 0x0000 (0) corresponds to 0V and 0xFFFF (65,535) corresponds to 3.3V. The order of individual samples in the data file is: CH2, CH1, CH4, CH3, …, CH(n), CH(n-1); where n is the total number of channels selected in the configuration file. The channels are actually sampled in the correct order (1, 2, …, n-1, n), but are reordered in the file due to the configuration of the serial connection between the microprocessor and the SD card.

A MATLAB script and a LabVIEW VI are provided to read the binary data files.

Modifying Hardware and Software

The PCB schematic, layout, and the C source code of the MuddLogg16 are available for modification. To program the PIC32, connect an ICD to the programming pins on the right side of the board. These pins are, from bottom to top: MCLR, VDD, GND, PGD, PGC. You can use a breadboard adapter to break-out the pins of the ICD's RJ11 connector, or strip the wires yourself. Note that much of the underlying code is part of the Microchip MDD File System Library, which is copyrighted by Microchip Technology, Inc., and is released for use only on Microchip PIC microprocessors.