micro-DRIVE

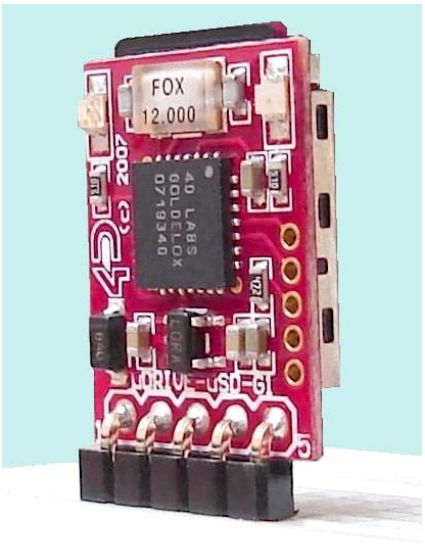
µDRIVE-uSD-G I

USERS MANUAL

Tiny "Disk Drive" Data Storage Module for Embedded Applications

Based on micro-SD Memory Card

Revision 1.0



4D Systems

PROPRIETORY INFORMATION

DISCLAIMER OF WARRANTIES & LIMITATION OF LIABILITY

Table of contents

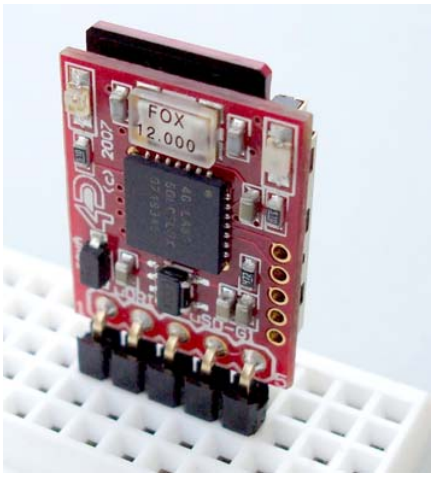
1. Description
   1.1 Introduction
   1.2 Features
2. Serial Command Set
   2.1 Command Protocol
   2.2 General Command Set
      2.2.1 Version / Device Info Request
   2.3 Disk Drive Operation Command Set
      2.3.1 initialise Disk Drive Memory Card
      2.3.2 Read Sector Block Data
      2.3.3 Write Sector Block Data
      2.3.4 read Byte Data
      2.3.5 write Byte Data
      2.3.6 Set Memory Address
   2.4 Serial Interface (TTL)
   2.5 USB Interface
   2.6 Personality-module-micro-Code (PmmC)
3. Specifications
   3.1 Host Interface & Pin-Outs
   3.2 Mechanical Details
   3.3 Power-Up Reset
4. Appendix
   4.1 Related Products
   4.2 Help and Other Information

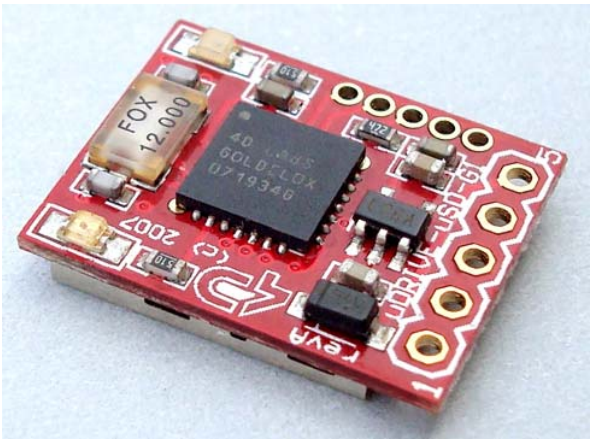
1 Description

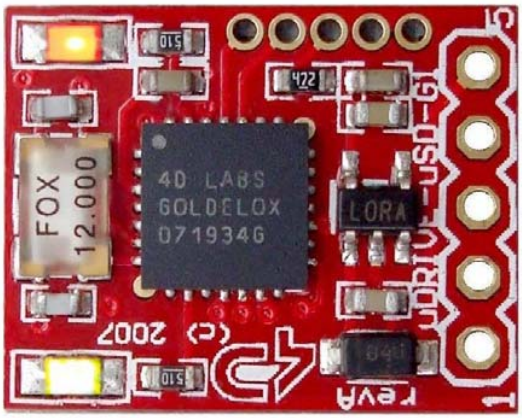
1.1 Introduction

The **µDRIVE-uSD-G1** is an extremely compact and an ideal companion module that can be easily interfaced to any microcontroller that requires a "**Disk Drive**" data storage system. Most microcontrollers are limited by the available onboard memory for those applications that require large volumes of data. The **µDRIVE-uSD-G1** takes that burden away from the host controller by allowing data transfers to-and-from via a simple serial interface.

A handful of straightforward commands provide direct access to the onboard micro-SD memory card for storing and retrieving any size or type of binary data. Applications can range from Data Logging, Program Storage, Music files, Image files, Video Clips or any general purpose data storage. So next time your PIC, AVR, ARM, STAMP, Propeller or any other microcontroller application requires data storage, the **µDRIVE-uSD-G1** maybe the ideal solution.







1.2 Features

- A general purpose data storage device with a simple serial interface that can interface to any microcontroller in a wide range of embedded applications.
- Ideal companion peripheral that can be used like a "**Disk Drive**" for your next embedded project.
- Ideal for those applications that require Data Logging, Program Storage, Music files, Image files, Video Clips or any general purpose raw Data Storage.
- Onboard micro-SD (**µSD**) memory card adaptor that can accommodate 64Mb to 2Gig memory cards. Memory cards can be purchased separately.
- A handful of simplified commands provide reads and writes of data blocks from a single byte in size to 512 bytes of sectors.
- Extremely small size: 14.9 x 18.9 x 3.5mm.
- Easy 5 pin interface to any host device: VCC, TX, RX, GND, RESET.
- Onboard Leds indicate System Status:
  - **GREEN** Led: Power and Memory Card detect indicator
  - **RED** Led: "Disk Drive" data access indicator
- Wide supply voltage range from 3.6 Volts to 6.0 Volts. Operating current @23mA nominal when using a 5.0 Volts supply source. **Note**: Due to the characteristics of some memory cards the module may require input voltages greater than 4.0 Volts.
- Serial RS-232 (0 to 3.3 Volt signal levels) with auto-baud feature from 300 to 256K baud rates. If interfacing to a system greater than 3.6 Volts supply, a series resistor (100 to 220 Ohms) maybe required on the RX line.
- Powered by the 4D-LABS **GOLDELOX** processor (also available as separate OEM chips for volume users).
- Optional USB to Serial interface via the 4D micro-USB (**µUSB-MB5** or **µUSB-CE5**) modules for Personality-module-micro-Code (**PmmC**) system firmware updates.

2 Serial Command Set

The data storage and retrieval takes place via the serial interface. With a handful of easy to learn commands that provide reads and writes of data blocks from a single byte in size to 512 bytes of sectors. The simplified command set also means that very low overheads are imposed on the host controller. The command set is grouped into 2 sections:

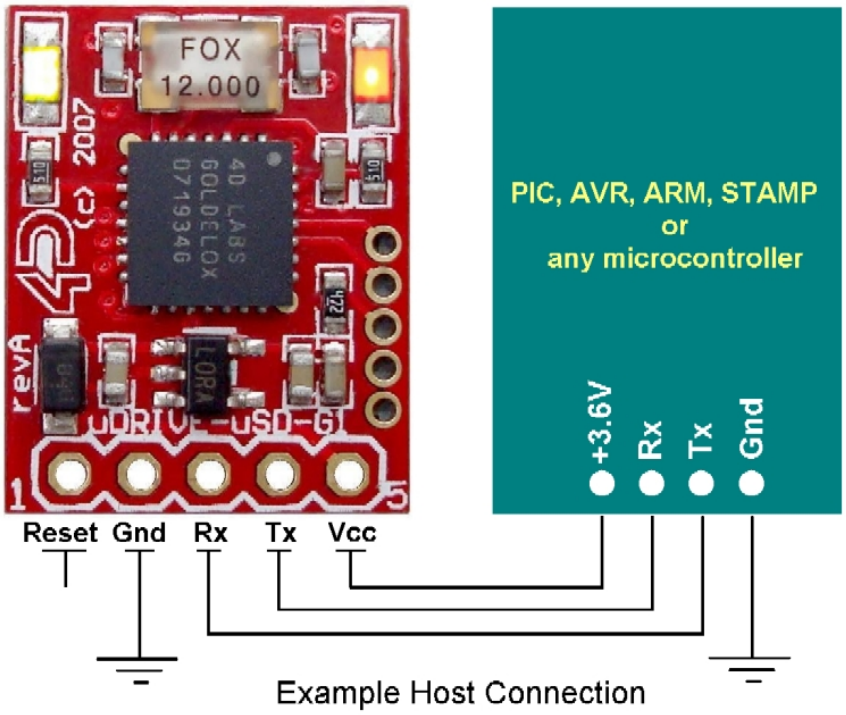
- General Command Set
- Disk Drive Operation Command Set

Each Command set is described in detail in the following sections.

NOTE!

The RX and the TX signals are at 3.3V levels. If interfacing to a host system running at voltages greater than 3.6V levels, say 5.0Volts, then a 100 to 220 Ohms series resistor must be inserted between the Host Tx and the µDRIVE-uSD-G1 Rx signals.

Serial Data Format: 8 Bits, No Parity, 1 Stop Bit.



Example Host Connection

2.1 Command Protocol

The following are each of the commands with the correct syntax. Please note that all command examples listed below are in hex (**00**hex).

NOTE: When transmitting the command and data bytes to the µDRIVE, do not include any separators such as commas ‘,’ or spaces ‘ ‘ or brackets ‘(‘ ‘)’ between the bytes. The examples show these separators purely for legibility; these must not be included when transmitting data to the module.

When a command is sent, the module will reply back with a single acknowledge byte called the **ACK** (**06**hex). This tells the host that the command was understood and the operation is completed. It will take the µDRIVE certain amount of time to reply back, depending on the command and the operation it has to perform. If the module receives a command that it does not understand it will reply back with a negative acknowledge called the **NAK** (**15**hex).

If a command that has 5 bytes but only 4 bytes are sent, the command will not be executed and the µDRIVE will wait until another byte is sent before trying to execute the command. There are no timeouts when incomplete commands are sent. The module will reply back with a **NAK** for each invalid command it receives. For correct operation make sure the command bytes are sent in the proper sequence and correct number of bytes.

2.2 General Command Set

2.2.1 Version / Device Info Request (V)

Syntax : **cmd**
Response : **device_type, hardware_rev, PmmC_rev, reserved1, reserved2**

cmd : **56**hex**, V**ascii

device_type : this response indicates the device type.
00hex = micro-OLED.
01hex = micro-LCD.
02hex = micro-VGA.
03hex = micro-DRIVE.

hardware_rev : this response indicates the device hardware version.

PmmC_rev : this response indicates the device PmmC firmware version.

reserved1: this byte is reserved for future support. If value is 0x00 then ignore it.

Reserved2: this byte is reserved for future support. If value is 0x00 then ignore it.

Description : This command requests all the necessary information from the module about its characteristics and capability.

2.3 Disk Drive Operation Command Set

The following commands are related to Disk Drive operation and they are described in this section. The **µDRIVE-uSD-G1** has an integrated micro-SD (**µSD**) memory card adaptor and can accept memory cards of any size from 64Mb up to 2Gig for data storage of any type.

Disk Drive Operation Command Set
(@i) **i**nitialise Disk Drive Memory Card
(@R) **R**ead Sector Block Data
(@W) **W**rite Sector Block Data
(@r) **r**ead Byte Data
(@w) **w**rite Byte Data
(@A) Set Memory **A**ddress for byte wise reads/writes
Future commands may follow

2.3.1 initialise Disk Drive Memory Card (@i)

Syntax : extCmd, cmd

extCmd : 40hex**,** @ascii

cmd : 69hex**, i** ascii

Description : This command initialises the **µSD** memory card. The memory card is always initialised upon Power-Up or Reset cycle, if the card is present. If the card is inserted after the power up or a reset then this command must be used to initialise the card.

Note! There is no card insert/remove auto detect facility.

2.3.2 Read Sector Block Data (@R)

Syntax : extCmd, cmd, SectorAddress(hi:mid:lo)

extCmd : 40hex**,** @ascii

cmd : 52hex**, R**ascii

SectorAddress(hi:mid:lo)**:** A 3 byte sector address. Sector Address range from 0 to 16,777,215 depending on the capacity of the card. Each sector is 512 bytes in size. There are 2048 sectors per every 1Mb of card memory.

Description : This command provides a means of reading data back from the µDRIVE in lengths of 512 bytes. This command allows a faster and more efficient means of data retrieval. Once this command is sent, the µDRIVE will return 512 bytes of data relating to that particular sector.

2.3.3 Write Sector Block Data (@W)

Syntax : **extCmd, cmd, SectorAddress**(hi:mid:lo), **data**(1)**, .. , data**(512)

extCmd : 40hex**,** @ascii

cmd : 57hex**, W**ascii

SectorAddress(hi:mid:lo)**:** A 3 byte sector address. Sector Address range from 0 to 16,777,215 depending on the capacity of the memory card used in the module. Each sector is 512 bytes in size. There are 2048 sectors per every 1Mb of µDRIVE memory.

data(1)**, .. , data**(512)**:** 512 bytes of sector data. The data length must be 512 bytes long. Unused bytes must be padded even if not all are used.

Description : This command allows downloading and writing large amounts of data to the µDRIVE. Downloads must always be limited to 512 bytes in length. For large volumes of data such as images, the data must be broken up into multiple sectors (chunks of 512 bytes) and this command then maybe used many times until all of the data is written into the µDRIVE. If the data block to be written is less than 512 bytes in length, then make sure the rest of the remaining data are padded with 00hex or FFhex (it can be anything).

If only few bytes of data are to be written then the **Write Byte** command can be used.

Once this command message is sent, the µDRIVE will take a few milliseconds to write the data into its memory card and at the end of which it will reply back with an **ACK** (06hex) if the write cycle was successful. If there was a problem in writing the data, a **NAK** (15hex) will be sent back without any write attempts.

Only **data**(1) to **data**(512) are written to the sector. Other bytes in the command message such as Sector Address are not stored.

2.3.4 read Byte Data (@r)

Syntax : extCmd, cmd

extCmd : 40hex**,** @ascii

cmd : 72hex**, r** ascii

Description : This command provides a means of reading a single byte of data back from the µDRIVE. Before this command can be used, memory address location must be set using the **Set Memory Address** command. Once this command is sent, the µDRIVE will return 1 byte of data relating to that memory location set by the memory Address pointer. The memory Address location pointer is automatically incremented to the next address location.

2.3.5 write Byte Data (@w)

Syntax : extCmd, cmd, data

extCmd : 40hex**,** @ascii

cmd : 77hex**, w** ascii

data : 1 byte of memory card data.

Description : This command allows writing single bytes of data to the µDRIVE. This is useful for writing small chunks of data at irregular intervals quickly. For large data blocks it is more efficient to use the **Write Sector Data** command described in the previous section.

Before this command can be used, the µDRIVE memory address location must be set using the **Set Memory Address** command. Once the **Write Byte** command is sent, a single byte of data will be stored to that memory location set by the memory Address pointer. The memory Address location pointer is automatically incremented to the next address location.

Only the **data** byte is written. Other bytes in the command message are not stored.

2.3.6 Set Memory Address (@A)

Syntax : extCmd, cmd, Address(Umsb:Ulsb:Lmsb:Llsb)

extCmd : 40hex**,** @ascii

cmd : 41hex**, A**ascii

Address(Umsb:Ulsb:Lmsb:Llsb)**:** A 4 byte µDRIVE address for byte wise access.

Description : This command sets the µDRIVE memory Address pointer for byte wise reads and writes. After a byte read or write, the memory Address pointer is automatically incremented internally to the next Address location.

2.4 Serial Interface (TTL)

The µDRIVE needs to be connected via a serial link to a host controller. The host uses this serial link to send commands to the module for data storage and retrieval. Use the signal pin-outs as well as the application example shown in the following section for correct connection to the host.

NOTE!

The RX and the TX signals are at 3.3V levels. If interfacing to a host system running at voltages greater than 3.6V levels, say 5.0Volts, then a 100 to 220 Ohms series resistor must be inserted between the Host Tx and the µDRIVE-uSD-G1 Rx signals.

Serial Data Format: 8 Bits, No Parity, 1 Stop Bit.

Auto Baud Detect:

The µDRIVE module has an auto-baud detect function which can operate from **300 baud to 256K baud**. Prior to any commands being sent to the module, it must first be initialized by sending the ASCII character **'U' (55h)** after power-up. This will allow the module to determine and lock on to the baud rate of the host controller automatically without needing any further setup.
This must be done every time the module is powered up or reset.

If the host needs to change the baud rate, the µDRIVE must be powered down and powered back up again. The "U" command cannot be used to change the baud rate during the middle of normal usage.

Serial Timing:

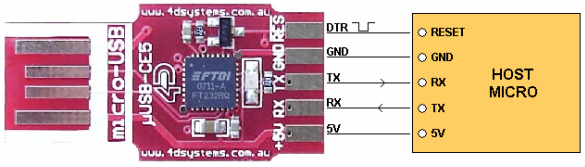
Each command is made up of a sequence of data bytes. When a command is sent to the module and the operation is completed, the µDRIVE will reply back with a single acknowledge byte called the **ACK** (06h). This tells the host that the command was understood and the operation is completed. It will take the module a certain amount of time to reply back with an **ACK**, depending on the command and the operation that has to be performed. If the µDRIVE module receives a command that it does not understand it will reply back with a negative acknowledge called the **NAK** (15h).

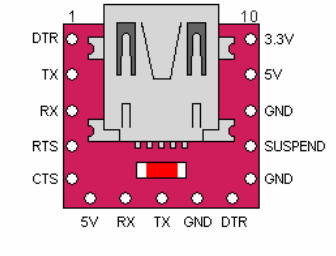
For example, if a command has 5 bytes but only 4 bytes are sent, the command will not be executed and when the next following command bytes are sent the module will reply back with a **NAK** for each and every byte it receives. For correct operation make sure the command bytes are sent in the correct sequence.

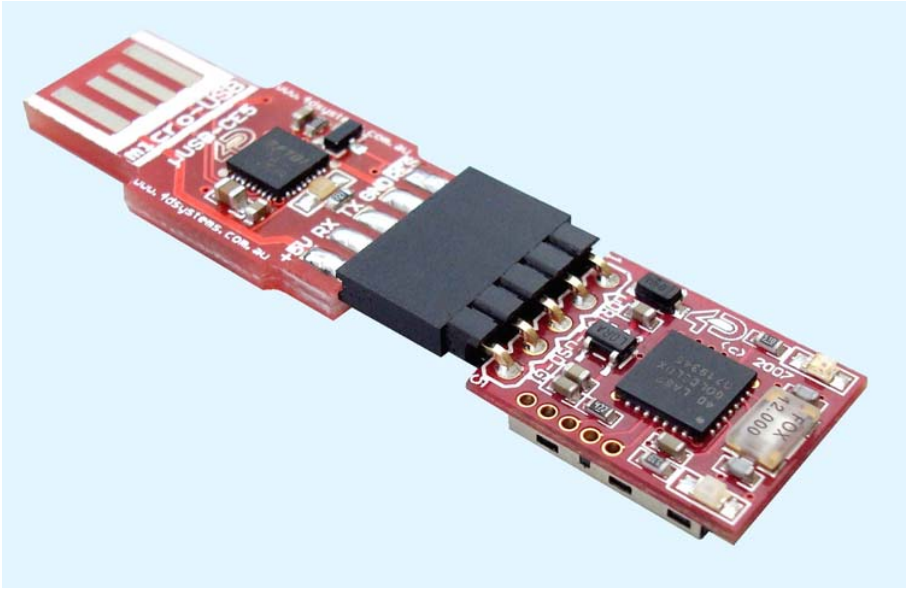
Note: No termination character is to be sent at the end of the command sequence. i.e. don't send any CR, or Null, or any other end of command bytes.

2.5 USB Interface

The µDRIVE can be interfaced to a PC using a standard USB cable and the 4D Systems micro-USB module (**µUSB-MB5** or **µUSB-CE5**) as shown below. The micro-USB module (optional extra), simply connects to the µDRIVE 5 pin header and captures the USB data and converts it into serial TTL data. The micro-USB modules and drivers are available from your local 4D distributor. This is an optional extra product and is not included with the µDRIVE module.







2.6 Personality-module-micro-Code (PmmC)

One of the important features of the µDRIVE module is the ability to upload its onboard **GOLDELOX** processor with a micro-Code firmware which allows the module to take on a new personality. This is referred to as Personality-module-micro-Code (**PmmC**). One of the greatest benefits is that it allows the module to be easily upgraded by the user at any time with PmmC files as further enhancements are made in the future. This allows the user to benefit from those latest features.

The latest **PmmC** system file for the µDRIVE-uSD-G1 can be downloaded from:
www.4dsystems.com.au/prod.php?id=22

The latest version of **PmmCLoader.exe** PC software tool and the User Guide can be downloaded from:
www.4dsystems.com.au/prod.php?id=22

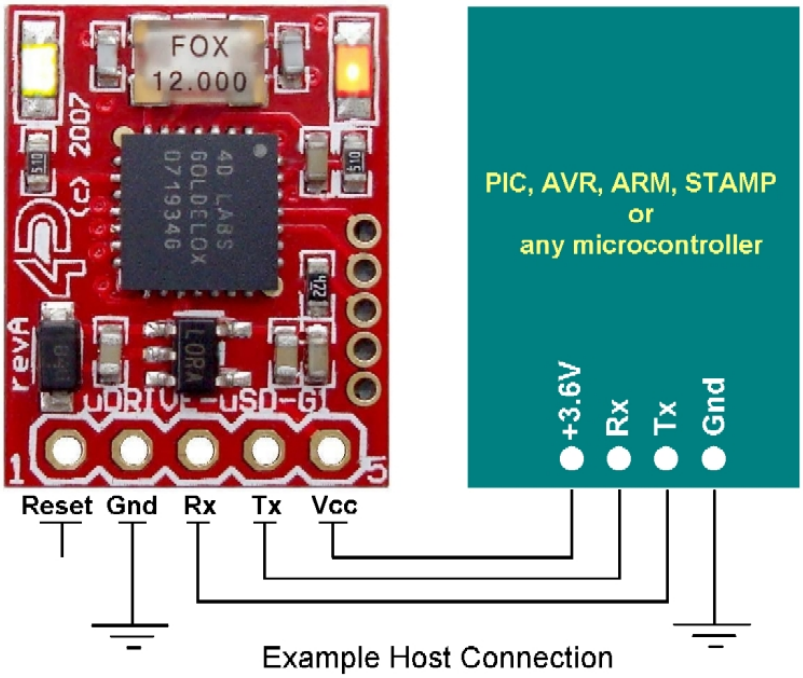
3. Specifications

The µDRIVE-uSD-G1 has the following electrical specifications which must be adhered to at all times to prevent damage to the device.

Symbol	Characteristic	Min	Typ	Max	Units
Vcc	Supply voltage	3.6*	5.0	6.0	**V**
I	Current	--	23	--	**mA**
Deg C	Operating temp	0	--	70	**C**
Tpu	Power-up delay	300	400	--	**mS**

***** Due to characteristics of certain micro-SD memory cards, the µDRIVE module may require supply voltages greater than 4.0 Volts.

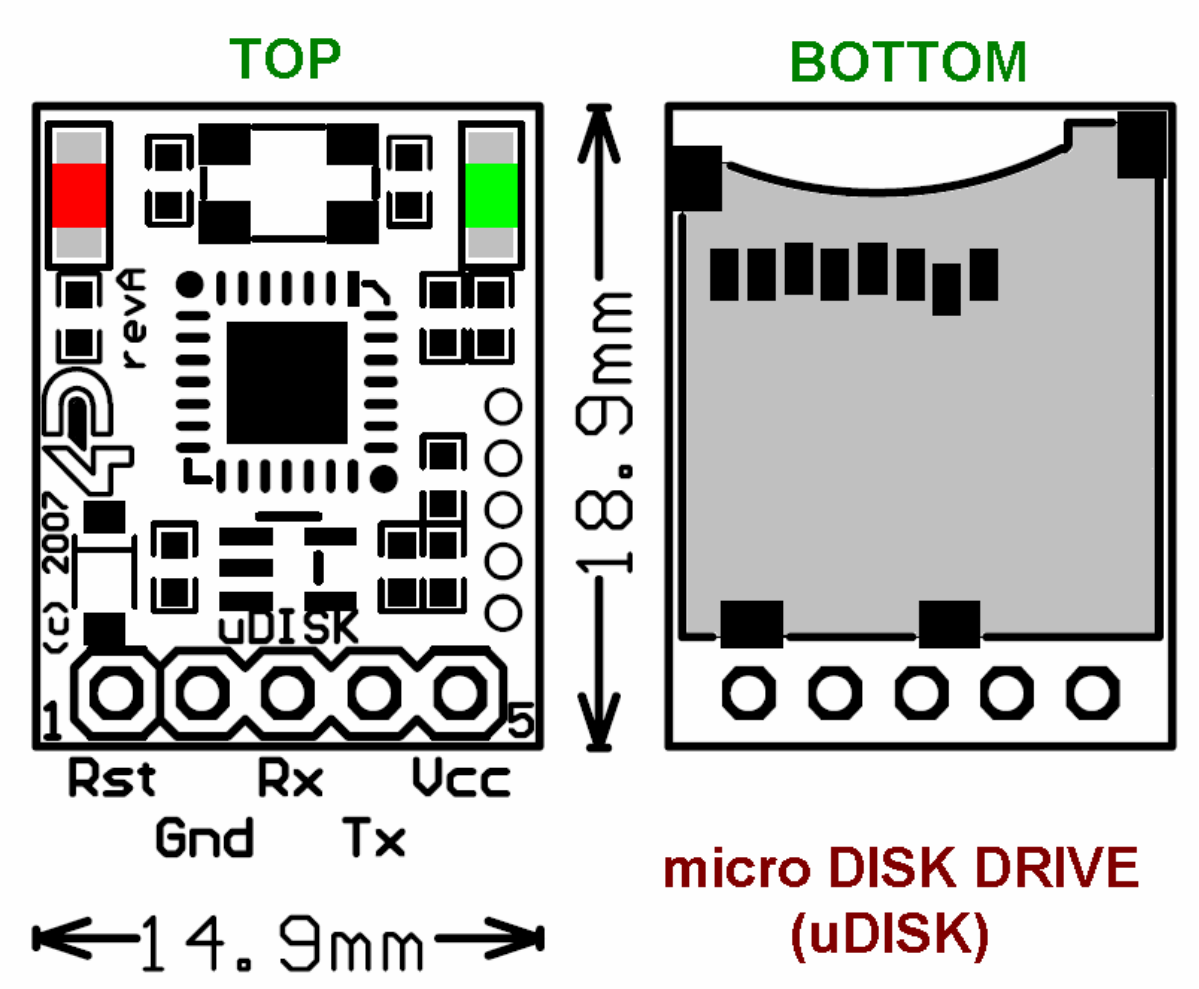
3.1 Host Interface & Pin-Outs



Example Host Connection

Pin	Function
5	+ve Power Supply input: 3.6 to 6.0 Volts D.C
4	Serial Transmit Data Pin: 0 to 3.3 Volt signal levels.
3	Serial Receive Data Pin: 0 to 3.3 Volt signal levels.
2	-ve Power Supply input: connect to GND
1	Reset Pin: Active Low pulse > 10 micro seconds will reset the module. Not necessary for normal operation usage. Internally pulled up to 3.3Volts.

3.2 Mechanical Details



The **µDRIVE-uSD-G1** module footprint is **14.9**mm x **18.9**mm x **3.5**mm.

3.3 Power-Up Reset

When the µDRIVE-uSD-G1 module comes out of a power up reset, allow up to 400ms before attempting to communicate with the module. The power up sequence of events should be as follows:

- Allow 400ms after power-up or reset for µDRIVE-uSD-G1 to settle. Do not attempt to communicate with the module during this period. The module may send garbage on its Tx Data line during this period, the host should disable its Rx Data reception.
- Within 100ms of powering up, the host should make sure it has its Tx line pulled HIGH. If the host Tx (µDRIVE-uSD-G1 Rx) is LOW or floating after the 100ms period, the module may misinterpret this as the START bit and lock onto some unknown Baud Rate. If the host has a slow wake up time, i.e. less than 100ms, its Tx line maybe floating. This can be easily resolved by adding a pull up resistor on the host Tx line which will ensure the µDRIVE-uSD-G1 does not encounter a false START bit. The pull up resistor can be any value within 10K to 100K.
- The host transmits the ASCII 'U' (capital **U**, **55**hex) as the first command so the µDRIVE-uSD-G1 can lock onto the host's serial baud rate. This is called "**Auto Bauding**". The module will respond with an 'ACK' (06h). See section 2.4
- The µDRIVE-uSD-G1 is now ready to accept Disk Drive operation commands from the host.

4. Appendix

4.1 Related Products:

- **µUSB-MB5**
  - micro-USB module, USB to Serial Bridge
  - Standard USB miniB connector
  - 10 pin header provides the following signals:
    - 5V, 3.3V, GND, Tx, Rx, Suspend,
    - DTR, CTS, RTS, GND
  - 5 Volts supply @ 500mA, 3.3 Volts supply @ 100mA
  - Additional flow control signals, DTR, CTS, RTS
  - Available with an additional 5 pin header for the µOLED interface
    www.4dsystems.com.au/prod.php?id=18

- **µUSB-CE5**
  - micro-USB module, USB to Serial Bridge, FTDI Chipset
  - Plugs directly into USB port
  - 5 pin header provides the following signals:
    - 5V, Rx, Tx, GND, Reset
  - 5 Volts supply @ 500mA
    www.4dsystems.com.au/prod.php?id=19

- **µSD-64Mb**
  - 64Mb micro-SD Memory Card
  - Extremely small footprint.
  - Measuring only 15mm x 11mm x 0.8mm
  - The uSD-64Mb memory card can be used for all general purpose data storage.

- **PmmC System File for the µDRIVE-uSD-G1**
  - The latest PmmC system files can be downloaded from:
    www.4dsystems.com.au/prod.php?id=22
- **PmmC Loader PC Software Tool**
  - Latest version of **PmmC-Loader** software tool can be downloaded from:
    www.4dsystems.com.au/prod.php?id=22

4.2 Help and Other Information:

- Assistance with latest information and downloads visit the **µDRIVE-uSD-G1** product web-page of your distributor.
- Questions and technical support please email support@4dsystems.com.au
- All related product information can be downloaded from www.4dsystems.com.au/prod.php?id=22