DM2000 User Manual

Fabio Dalla Libera

May 11, 2023

1 Hardware

Figure 1 shows a picture of the DM2000. It consists of four serial peripherals which can freely connected

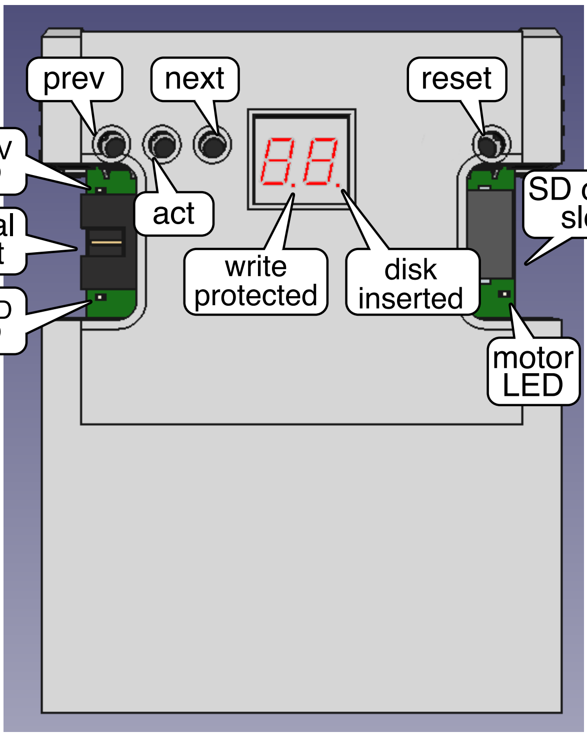

Figure 1: DM2000 top side.

to each other. The four serial peripherals are

- `S`: a 3.3V serial port, with pinout compatible to DL FEN, an **S**F-7000 emulator. See Section TTL serial for further details.
- `U`: a **U**SB to serial adapter. See Section USB for further details.
- `D`: a ±12V RS-232 serial on a **D**E-9 connector. See Section RS-232 for further details.
- `E`: An **E**SP8266. The intended usage of the ESP is simulating an Hayes modem. See Section ESP for further details.

The receive line (RxD) of any of these peripherals is denoted by `XR`, where `X` is one of `S`, `U`, `D`, or `E`. Similarly the transmit line (TxD) of any of these peripherals is denoted by `XT`, where `X` is one of `S`, `U`, `D`, or `E`. Activity on each of these lines is shown by the corresponding activity LEDs.

The switching matrix consists of four pin headers. Each pin header allows connecting one receive line `XR` to the transmit line `YT` of any of the other devices by the use of a jumper. See Section Switch matrix for further details.

Power is provided through the USB-C connector, controlled using the power switch, and indicated by the `PWR` LED. Two push buttons, allow to program (`FLASH` button) and reset (`RST` button) the ESP8266. Refer to Section Updating Zimodem firmware for further details.

1.1 TTL serial

The TTL serial is provided as a 6 pin IDC connector. Pin associations are as follows:

Pin	Function
1	reserved
2	TX
3	not connected
4	RX
5	GND
6	GND

1.2 USB

The USB connector is intended for providing both power and a data connection. It can be connected either to a power adapter, such as a phone charger, if a serial over USB is not needed. Both USB-C to USB-C and USB-C to USB-A cables can be used to connect the device to a PC. The usb is connected to a CH340. When the device is connected to a PC, a new serial port is shown. Drivers should not be necessary on recent operating systems. Refer to `https://learn.sparkfun.com/tutorials/how-to-install-ch340-drivers/all` for further information on the installation of drivers for the CH340.

1.3 RS-232

The DE9 connector is wired as a standard RS-232 serial

Pin	Function
2	RX
3	TX
5	GND

and the other pins are left as not connected. *When connecting to a PC use a Simple Null Modem Cable.* When connecting to a physical modem a straight through serial cable should be used.

1.4 ESP

The socket located on the top of the device is intended for the insertion of an ESP-01 WiFi Module. See `https://docs.ai-thinker.com/_media/esp8266/esp8266_series_modules_user_manual_en.pdf` for further details on the device. The device should be plugged in as shown in Figure 1. ***Inserting the ESP-01 in the wrong direction leads to damage of the ESP-01, the DM2000 and possibly of the devices connected to the DM2000***.

1.5 Switch matrix

Figure 2 shows the switch matrix configuration.

Each device is associated to a direction, corresponding to the approximate position of the corresponding connector:

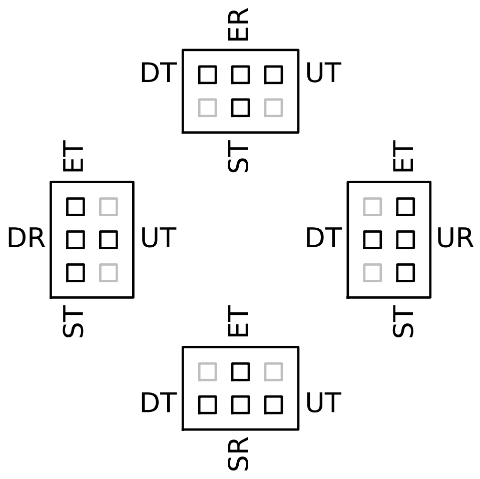

Figure 2: Switch matrix configuration.

- up for device `E`
- down for device `S`
- left for device `D`
- right for device `U`

The pin header for the receive line of a certain device is the one in the direction corresponding to the direction associated to the device. For instance, the receive line of device `S`, associated to the bottom direction, can be connected using the bottom pin header. Inside a pin header, the transmission line of each of the other devices is in the associated direction. For instance, to connect the receive line of `S` (down) to the transmission line of `D` (left), a jumper should be connected as shown in Fig. 3.

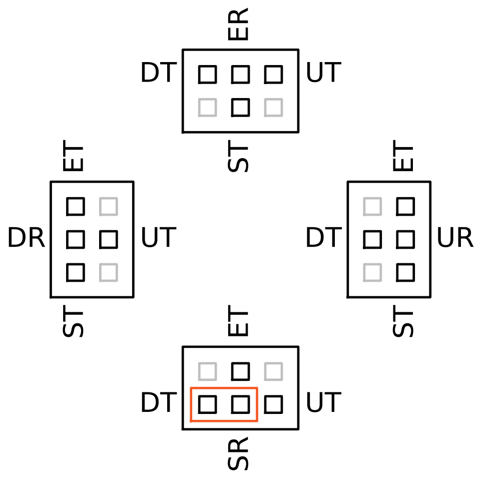

Figure 3: Jumper connection of SR to DT..

2 Connecting the SC-3000 to a Linux machine

Make the connections `UR-ST` and `SR-UT`, as shown in Fig. 4.

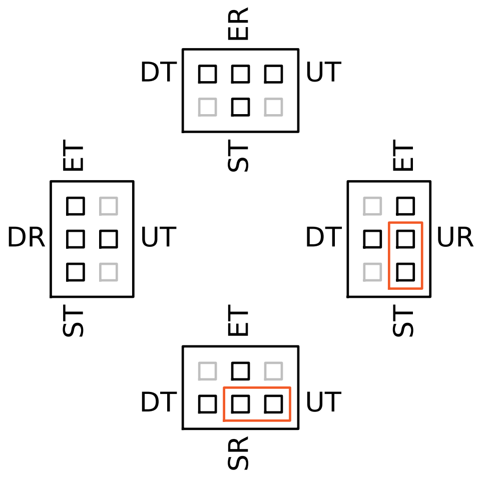

Figure 4: Jumper configuration for connecting FEN to a PC.

Connect the USB port to a Linux machine. A new USB tty should appear. Using `dmesg` you should see something like

```
[26436.778832] usb 3-2.4.3: new full-speed USB device number 67 using xhci_hcd
[26436.886319] usb 3-2.4.3: New USB device found, idVendor=1a86, idProduct=7523, bcdDevice= 2.64
[26436.886325] usb 3-2.4.3: New USB device strings: Mfr=0, Product=2, SerialNumber=0
[26436.886327] usb 3-2.4.3: Product: USB Serial
[26436.941831] ch341 3-2.4.3:1.0: ch341-uart converter detected
[26436.948388] usb 3-2.4.3: ch341-uart converter now attached to ttyUSB0
```

On the Linux machine type

```
sudo stty -F /dev/ttyUSB0 1200
TERM=vt52 sudo -E systemctl start serial-getty@ttyUSB0.service
```

where`ttyUSB0` is the serial port created by attaching the DM2000.

Once logged in, on the SC-3000 type (or add it to `.bashrc`)

```
export TERM=vt52
stty rows 24 cols 40
```

3 Connecting the SC-3000 to a BBS

The ESP-01 emulates an Hayes modem. The first step to connect to a BBS is thus making the connections `ER-ST` and `SR-ET`, as shown in Fig. 5.

A terminal emulator program must be started on the SC-3000. A possible choice is `DL VT52 terminal emulator`. The following section provides an example of the commands to be typed on the SC-3000.

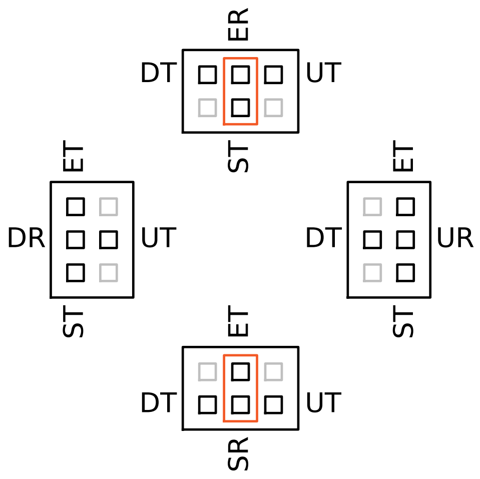

Figure 5: Jumper configuration for connecting FEN to the ESP-01.

4 Zimodem

The ESP-01 of the DM2000 comes preloaded with Zimodem https://github.com/bozimmerman/Zimodem. Refer to Zimodem documentation for details. This is intended as a minimal start guide.

The connection can be checked by typing

```
ATI
```

(followed by carriage return). The modem should return something like

```
Zimodem Firmware v4.0.0
sdk=2.2.2-dev(38a443e) chipid=1327304 cpu@80
totsize=1024k ssize=381k fsize=51k speed=40m
INITIALIZED
READY.
OK
```

The SSIDs of the accessible wireless access points can be listed by the command

```
ATW
```

and the result should be something like

```
NETWORK SSID (-71)*
OK
```

To connect you should provide the command

```
ATW"NETWORK_SSID,NETWORK_PASSWORD"
```

which on success returns

```
OK
```

The command

```
ATI
```

should now return a message like

```
Zimodem Firmware v4.0.0
sdk=2.2.2-dev(38a443e) chipid=1327304 cpu@80
totsize=1024k ssize=381k fsize=51k speed=40m
CONNECTED TO NETWORK_SSID (192.168.1.10)
READY.
OK
```

The command to dial up to a BBS is of the type

```
ATD"bbs.fozztexx.com:23"
```

which should greet you with the BBS login message, in case of the above example,

```
CONNECT 1


Welcome to the *NEW* Level 29 BBS!
916 965 1701 - bbs.fozztexx.com


              --
  ___________/__\___________
 /   _  _  -__ -   ___  -   \
 \__________________________/
             |  |
             |  |
        _____|  |______
    ___|     |  |      |___
   |   |     |  |      |   |
___|___|____/____\_____|___|___
                        --   --
|     __        __  |  /  \ /  \
|    /__\ |  | /__\ |   __/ \__/
|___ \__   \/  \__  |  /___   /

The official BBS of
RetroBattlestations.com


Enter your username or NEW or VISITOR
User:
```

The command

```
AT&W
```

can be used to save the current configuration.

4.1 Updating Zimodem firmware

Download the code from https://github.com/bozimmerman/Zimodem. Open the `zimodem.ino` sketch.

At the moment of writing, Zimodem requires using old versions of the Arduino libraries to properly compile. Install Arduino IDE rev 1.8.10. Go to `Tools`,`Board`,`Board Manager...`. Search for "esp8266". Install version 2.7.4. If another version is installed, remove it and install version 2.7.4.

In the `Tools` menu,make the following selections

```
Board: Generic ESP8266 Module
Builtin Led "1"
CPU Frequency "80 Mhz"
Crystal Frequency "26 Mhz"
Flash size "1Mb (FS:512KB OTA:~246KB)"
Flash Mode "DIO"
Flash Frequency "40MHz"
Reset Method "no dtr (aka ck)"
```

and select the correct Port (usually the last one).

While holding the FLASH button pressed down, push and release the RST button. Press the upload button. The Arduino message area should report something similar to the following

```
Warning: Board breadboard:avr:atmega328bb doesn't define a 'build.board' preference.
Auto-set to: AVR_ATMEGA328BB
Executable segment sizes:
IROM   : 347980          - code in flash         (default or ICACHE_FLASH_ATTR)
IRAM   : 27472   / 32768 - code in IRAM          (ICACHE_RAM_ATTR, ISRs...)
DATA   : 1316  )         - initialized variables (global, static) in RAM/HEAP
RODATA : 7240  ) / 81920 - constants             (global, static) in RAM/HEAP
BSS    : 31520 )         - zeroed variables      (global, static) in RAM/HEAP
Sketch uses 384008 bytes (88%) of program storage space. Maximum is 434160 bytes.
Global variables use 40076 bytes (48%) of dynamic memory, leaving 41844 bytes for
local variables. Maximum is 81920 bytes.
esptool.py v2.8
Serial port /dev/ttyUSB1
Connecting...
Chip is ESP8266EX
Features: WiFi
Crystal is 26MHz
MAC: 18:fe:34:a6:a2:01
Uploading stub...
Running stub...
Stub running...
Configuring flash size...
Auto-detected Flash size: 1MB
Flash params set to 0x0220
Compressed 388160 bytes to 281532...

Writing at 0x00000000... (5 %)
Writing at 0x00004000... (11 %)
Writing at 0x00008000... (16 %)
Writing at 0x0000c000... (22 %)
Writing at 0x00010000... (27 %)
Writing at 0x00014000... (33 %)
Writing at 0x00018000... (38 %)
Writing at 0x0001c000... (44 %)
Writing at 0x00020000... (50 %)
Writing at 0x00024000... (55 %)
Writing at 0x00028000... (61 %)
Writing at 0x0002c000... (66 %)
Writing at 0x00030000... (72 %)
Writing at 0x00034000... (77 %)
Writing at 0x00038000... (83 %)
Writing at 0x0003c000... (88 %)
Writing at 0x00040000... (94 %)
Writing at 0x00044000... (100 %)
Wrote 388160 bytes (281532 compressed) at 0x00000000 in 24.8 seconds (effective 125.4 kbit/s)...
Hash of data verified.

Leaving...
Soft resetting...
```