WA0EDA

STM32-DVM-MTR2K v2.0
I/O API Addendum

Revision: 20200103

Table Of Contents

The WA0EDA Skunkworks may be contacted at:
5505 Plymouth Drive
Lawrence, KS 66049
n0mjs@me.com
(telephone support not provided)

Disclaimers and Conditions of Use

3rd Party Software Disclaimer

The STM32-DVM-MTR2K contains various 3rd party softwares redistributed under various versions of the GPL. WA0EDA and the Regents of the K0USY Group make no claims or warranties regarding such software and we are not responsible for its use. Using the STM32-DVM-MTR2K requires acceptance of and compliance with the license terms of included 3rd party software.

Use at Your Own Risk

The STM32-DVM-MTR2K, as a whole unit, is not UL (or similarly) listed, and has not been tested for FCC Part 15 (or other) compliance. By using the STM32-DVM-MTR2K, the user agrees to indemnify the WA0EDA Club, the Regents of the K0USY Group, Cortney T. Buffington, N0MJS and other affiliated entities against liability resulting in its use. The STM32-DVM-MTR2K is intended for use in experimental and educational environments consistent with the amateur radio service.

Configuration Disclaimer

WA0EDA provides configuration advice, but is not responsible or liable for providing comprehensive or authoritative information about MMDVM or the MTR2000. Configurations presented in this manual represent how we set up MMDVM and the MTR2000 with the STM32-DVM-MTR2K. This manual does not attempt to cover alternative configurations, and is not an authoritative source of information for MMDVM or the MTR2000.

Conventions used in this manual

When illustrated in binary, bytes and bit-field ranges will always be represented from most significant bit (MSb) on the left, to least significant bit (LSb) on the right. Bit field numbering will always begin at 0, meaning the most significant bit in a byte, will be referred to as the 8th bit, and be numbered 7.

When multiple bytes are transferred, the first byte transferred will be displayed on the left, and the last on the right. Examples are a great way to wrap your head around these conventions, so let's see some:

> 0xF0 = 0b11110000 = bits 7-4 are set (1) and bits 3-0 are cleared (0).
> 0x80 = 0b10000000 = bit 7 is set (1), and bits 6-0 are cleared (0).
> 0x00 0x01 0x03 = three bytes sent: 0x00, byte 0 was first; 0x01, byte 1 was second; 0x3, byte 2 was sent last.

When discussing a bit field or range in a byte, the positions NOT being discussed will be represented by a period. If we are discussing, for example the 5th bit (bit 4), this bit may be represented as "0b...X. ...."; noting a space is left between the two nibbles of the byte because it's easier to spot the bit position at a glance this way.

Overview

The STM32-DVM-MTR2K V2.0 and newer contains hardware and software features to duplicated the functionality of the Motorola Auxiliary I/O board, Part Number CLN6698. These functions are performed by an Atmel/Microchip ATMega328P microcontroller, operating as an SPI Bus Slave to the MTR2000 SCM (System Control Module) Bus Master. As outlined in the primary manual for the STM32-DVM-MTR2K, the following GPIO are physical available in hardware:

INPUTS	OUTPUTS
GPI_3	GPO_0
GPI_4	GPO_2
GPI_7	GPO_8
GPI_10	GPO_13

This represents all of the available, programmable GPOs and half of the available GPIs. The other 4 GPIs (9, 11, 12, 13) were not connected to hardware inputs, but are available exclusively as “soft” inputs, available via API along with other features for reading the hardware GPIO pins, and receiving automatic alerts that an input or output has changed.

For hardware information, including output drive and input voltage limits, etc., see the STM32-DVM-MTR2K V2.0 manual.

API Design

Serial port parameters are 57600bps, 8-N-1.

The STM32-DVM-MTR2K I/O API is available through the asynchronous serial port (UART) of the ATMega328P I/O processor, which is available via header J3 (which may also be used via the bootloader for asynchronous serial programming), and by UART2 of the NanoPi NEO when jumpers are installed on JP7. The ATMega328P operates at 5VDC, which means the UART interface operates at 5VDC. The connection (via JP7) to the NanoPi NEO is level shifted to 3.3VDC by Q2, Q3 and Q4.

All commands and options sent to the I/O Processor from a host require only a single byte. The I/O Processor returns data to the host as either a single byte or 3 bytes, depending on the command sent. All commands and responses include an opcode, which occupies the 3 most significant bits (7, 6, 5) of the transferred byte – or the first transferred byte in multi-byte transfers. The opcodes are shown in Table 1: Opcode Definitions. The 5th bit (4) is used to indicate binary arguments, such as input (0) or output (1), or to set (1) or clear (0) a bit. Finally the least significant 4 bits (3, 2, 1, 0), referred to as data bits, are used to indicate the MTR2000 GPIO number for specific read/write operations, or to represent all four software soft GPIs (13, 12, 11, 9; respectively).

The MTR transfers 16 bits (2 bytes) of data for each function, GPI or GPO, even though there are not 16 inputs and 16 outputs. Some of the fields are not used, some represent features (e.g. trunking controller failsoft input) not implemented by the STM32-DVM-MTR2K. A total of 8 GPIs, all of which are Wild Card (user definable) inputs are implemented. 11 GPOs are implemented, 7 of which are station status indicators (e.g. VSWR alarm) and 4 are Wild Card (user definable) outputs. When all inputs are read (GPI or GPO), all 16 bits are returned. For more information about position definitions and pin mapping, see Appendix A: MTR2000 GPI Definitions and Mapping. Unimplemented bit fields will be reported as zeros (0) and should be ignored.

Complete documentation for each opcode and arguments (if any) are fully described in the following section.

Table 1: Opcode Definitions

Opcode	Hexadecimal	Binary	Purpose
READ_ALL	0x00	0b000.	Read all GPIs
(NOT USED)	0x20	0b001.	Future expansion
SET_ALL	0x40	0b010.	Set all soft GPIs (9, 11, 12, 13) at once
IN	0x60	0b011.	Read a single GPI
OUT	0x80	0b100.	Read a single GPO
SET	0xA0	0b101.	Set a single soft GPI (9, 11, 12, 13)
AUTO	0xC0	0b110.	Enable/disable automatic I/O change alerts
ERR	0xE0	0b111.	ERROR – sent when an error condition occurs

Opcode Details

READ_ALL (0x00) (0b000. ....)

Reads all GPI or GPO depending on the state of the argument bit (4). The data bits (3-0) are not used.

Arguments:

(0b...0 ....) Request current state 8 MTR2000 GPI values, all of which represent Wild Card, or user programmable, inputs.
(0b...1 ....) Request current state 11 MTR2000 GPO values, 7 of which are station status indicators and 4 are Wild Card, or user programable, outputs.

Dat Bits:

Unused

Returns 3 Bytes:

Byte 0 (first returned) is the opcode | argument bit.
Byte1 (2nd returned) is the upper 7 GPI values 14-8. The most significant bit is always 1.
Byte2 (last returned) is the lower 8 GPI values 7-0.

Example:

Host sends: (0b0001 0000). READ_ALL | GPO argument.
Host receives: (0b0001 0000) (0b0000000) (0b00000010). In this example the only GPO set is the VSWR alarm

SET_All (0x40) (0b010. ....)

Sets all software controlled GPIs (13, 12, 11, 9). The argument bit is not used. The data bits represent the 4 software controllable GPIs, where bit 3 = GPI 13, bit 2 = GPI12 and so on.

Arguments:

Unused

Data Bits:

Data bits (3, 2, 1, 0, 0b.... xxxx) correspond to GPIs (13, 12, 11, 9) in order. Setting a bit (1) means activating the input, clearing will deactivate.

Returns Nothing

Example:

Host sends: (0b0100 1100). SET_ALL | set GPIs 13 and 12, clear GPIs 11 and 9
Host receives: Nothing

IN (0x60) (0b011. ....)

Reads a specified GPI by bit position (0-15). The argument bit is not used. Data bits specify the numeric GPI number to read. Reading a not implemented GPI will always return zero (0). The argument bit is set in the return value if the specified GPI is set.

Arguments:

Unused

Data Bits:

The numeric representation of the GPI to read (0-15) (0b.... xxxx).

Returns 1 Byte:

Opcode (bits 7-5) | Argument set/clear (depending on state) (bit 4) | numeric GPI (bits 3-0)

Example:

Host sends: (0b0110 1010). IN | (no argument) | numeric value of GPI to read (10).
Host receives: (0b0111 1010). Same as sent byte, but argument indicates whether the GPI is set or clear. In this example, GPI 10 (1010) is set (1).

OUT (0x80) (0b100. ....)

Reads a specified GPO by bit position (0-15). The argument bit is not used. Data bits specify the numeric GPO number to read. Reading a not implemented GPO will always return zero (0). The argument bit is set in the return value if the specified GPO is set.

Arguments:

Unused

Data Bits:

The numeric representation of the GPO to read (0-15) (0b.... xxxx).

Returns 1 Byte:

Opcode (bits 7-5) | Argument set/clear (depending on state) (bit 4) | numeric GPO (bits 3-0)

Example:

Host sends: (0b1000 1010). IN | (no argument) | numeric value of GPO to read.
Host receives: (0b1001 0100). Same as sent byte, but argument indicates whether the GPO is set or clear. In this example, GPO 4 (0100) is set.

SET (0xA0) (0b101. ....)

Sets a specified software GPI by bit position (0-15), though only the software GPIs are allowed (13, 12, 11, 9). The argument bit indicates whether the GPI should be set (1) or cleared (0). Attempting to set any other GPI results in no action taken.

Arguments:

Set (0b...1 ....) indicates the GPI will be set.
Clear (0b..0 ....) indicates the GPI will be cleared.

Data Bits:

The numeric representation of the GPI to read (0-15) (0b.... xxxx).

Returns Nothing

Example:

Host sends: (0b1010 1011). SET (0b101. ...) | clear (0b...0 ....) | GPI 11 (0b.... 1011)
Host receives: Nothing

AUTO (0xC0) (0b110. ....)

Turns on automatic reporting of GPIO changes. On reset, this features is disabled. When enabled, all GPI updates sent to the MTR2000 will be sent to the host and all GPO updates from the MTR2000 will be sent to the host. Messages to the host are 3 bytes long and include the IN or OUT opcodes to indicate GPI or GPO respectively, and the full 16 bit (two byte) values of the GPI or GPO registers.

Arguments:

Set (0b...1 ....) enables auto reporting.

Clear (0b...0 ....) disables auto reporting (default state).

Data Bits:

Unused

Returns Nothing

Example:

Host sends: (0b1101 ....). AUTO | set (0b...1 ....) enables automatic reporting.
Host receives: Nothing

ERR (0xE0) (0b111. ....)

This opcode is currently only sent to the host from the I/O processor. It indicates a received command from the host was not understood.

Arguments:

Unused

Data Bits:

Unused

Returns:

N/A

Example

Host receives: 0xE0 (0b1110 0000). Indicates an error. This implies the host's previous attempt to interact created a problem and should be tried again. Consistent reception of the ERR opcode indicates a problem with serial communications or the I/O processor firmware.

Appendix A: MTR2000 GPI Definitions and Mapping

Table 2: MTR2000 GPI Definitions

Bit Position	MTR2000 Definition	System Connector (J5)	SMT32-DVM-MTR2K	MTR2000 Function
0	GPI_0	B7	Not Implemented	Dedicated to Ext_Repeat Input
1	GPI_1	A8	N/A	Not Supported (Do Not Use)
2	GPI_2	A9	N/A	Not Supported (Do Not Use)
3	GPI_3	A5	Hardware Input	Wild Card Input
4	GPI_4	C5	Hardware Input	Wild Card Input
5	GPI_3	B6	N/A	Not Supported (Do Not Use)
6	GPI_4	A7	Not Implemented	Dedicated to Ext_Failsoft input
7	GPI_7	A22	Hardware Input	Wild Card Input
8	GPI_8	B5	N/A	Not Supported (Do Not Use)
9	GPI_9	A28	Software Input	Wild Card Input
10	GPI_10	C12	Hardware Input	Wild Card Input
11	GPI_11	B12	Software Input	Wild Card Input
12	GPI_12	B11	Software Input	Wild Card Input
13	GPI_13	B9	Software Input	Wild Card Input
14	GPI_14	B26/A29+A26	Not Implemented	Fast Ext_PTT or Not Supported
15	GPI_15	C7+A6	N/A	Not Supported (Do Not Use)

Table 3: MTR2000 GPO Definitions

Bit Position	MTR2000 Definition	System Connector (J5)	SMT32-DVM-MTR2K	Notes
0	GPO_0	A12	Hardware Output	Wild Card Output
1	GPO_1	A10	Software Only	VSWR_Fail output
2	GPO_2	A11	Hardware Output	Wild Card Output
3	GPO_3	A28	N/A	Not Supported (Do Not Use)
4	GPO_4	C12	N/A	Not Supported (Do Not Use)
5	GPO_5	B12	N/A	Not Supported (Do Not Use)
6	GPO_6	B11	N/A	Not Supported (Do Not Use)
7	GPO_7	B9	N/A	Not Supported (Do Not Use)
8	GPO_8	A1	Hardware Output	Wild Card Output
9	GPO_9	B1	Software Only	Rx_Lock output
10	GPO_10	C1	Software Only	Tx_Lock output
11	GPO_11	A2	Software Only	PA_Fail output
12	GPO_12	C4	Software Only	FailSoft output
13	GPO_13	B2	Hardware Output	Wild Card Output
14	GPO_14	A30/B29	Software Only	AC_Fail output or Not supported
15	GPO_15	C3/B3	Not Implemented	RdStat