MD46 Register API

Rev E

Mid-Continent Instruments and Avionics

Model Series MD46
Governor Controller / Data Recorder
(EMU)

CONFIDENTIAL
FOR INTERNAL USE ONLY

Revision History

Revision	API	Description	Author	Date
A	1.0.0	Initial Release	KEY	06/14/18
B	1.0.0	Update register addresses in several tables	KEY	06/19/18
C	1.0.0	Add log start/stop/increment feature	KEY	06/25/18
D	1.0.0	Modify the ordering of some of the fields, put multi-field registers in transmit order	KEY	07/06/18
E	1.0.0	Add Identifier register Clarify Register Address and Data transmit order (byte endianness)	KEY	07/23/18

Table of Contents

1 Usage 4
1.1 BLE Read Procedure 4
1.2 BLE Write Procedure 4
1.3 Data Format 4
1.3.1 Register Address 4
1.3.2 Register Data 4
2 Register Definition 4
2.1 Device Information 4
2.1.1 Software Version 4
2.1.2 API Version 5
2.1.3 Identifier 5
2.2 Timing 5
2.2.1 Flight Time 5
2.2.2 Current Date Time 5
2.3 Limits 5
2.3.1 Rotor RPM Limit 5
2.3.2 Engine RPM Limit 5
2.3.3 Cylinder Head Temperature Limit 6
2.3.4 Oil Temperature Limit 6
2.3.5 Exhaust Gas Temperature Limit 6
2.4 Exceedances 6
2.4.1 Latched Exceedances 6
2.5 Log Data 6
2.5.1 Log Control 6
2.5.2 Log Entry Count 7
2.5.3 Log Read Start Time 7
2.5.4 Log Read Stop Time 7
2.5.5 Log Read Increment Time 7
2.5.6 Log Read Control 8
2.5.7 Automated Log Dump 8

1 Usage

The MD46 allows for reading and writing various attributes via a Register + Data interface. This document will describe the register addresses, their data fields and access modes.

1.1 BLE Read Procedure

An API register can be read by first writing the 16 bit register address, 8 bit control, and optionally the 8 bit data length to the **Register Control** characteristic, then reading the **Register Data** characteristic, which will return a number of bytes up to the specified data length. See the appropriate register definition for the expected data length.
If the **Register Control** characteristic already holds the desired value, it does not need to be written before reading from the **Register Data** characteristic.

1.2 BLE Write Procedure

An API register can be written by first writing the 16 bit register address, 8 bit control, and optionally the 8 bit data length to the **Register Control** characteristic, then writing up to 20 bytes to the **Register Data** characteristic.
If the **Register Control** characteristic already holds the desired value, it does not need to be written before writing to the **Register Data** characteristic.

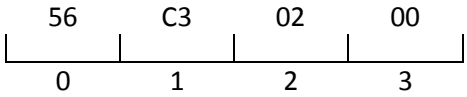
1.3 Data Format

1.3.1 Register Address

The register address is specified in big-endian format and should be transmitted in that byte order.

1.3.2 Register Data

All numerical data values are should be transmitted in byte little-endian format.
Example: A Log Entry Count (0x5010) value of 181,078 would be read in order of transmission as



Bit fields are specified in MSB to LSB, i.e. 7 to 0;
All text values are specified in readable order and should be transmitted in order from the first character to the last character.

2 Register Definition

2.1 Device Information

2.1.1 Software Version

0x0100 Software Version				
Byte	Bits	USB	BLE	Description
0	-	R	R	Major Version
1	-	R	R	Minor Version
2	-	R	R	Patch Version
3	-	R	R	Build Version

2.1.2 API Version

0x1000 API Version				
Byte	Bits	USB	BLE	Description
0	-	R	R	Major Version
1	-	R	R	Minor Version
2	-	R	R	Patch Version
3	-	R	R	Build Version

2.1.3 Identifier

0x1010 Identifier				
Byte	Bits	USB	BLE	Description
0:20	-	R/W	R	Identifier String (Max 20 characters)

2.2 Timing

2.2.1 Flight Time

0x2000 Flight Time				
Byte	Bits	USB	BLE	Description
3:0	-	R/W	R	Hours
5:4	-	R/W	R	Minutes
7:6	-	R/W	R	Seconds

2.2.2 Current Date Time

0x2010 Current Date Time				
Byte	Bits	USB	BLE	Description
3:0	-	R/W	R	UNIX Epoch time in seconds
5:4	-	R/W	R	Millisecond count

2.3 Limits

2.3.1 Rotor RPM Limit

0x3000 Rotor RPM Limit				
Byte	Bits	USB	BLE	Description
3:0	-	R/W	R	single precision floating point value

2.3.2 Engine RPM Limit

0x3004 Engine RPM Limit				
Byte	Bits	USB	BLE	Description
3:0	-	R/W	R	single precision floating point value

2.3.3 Cylinder Head Temperature Limit

0x3008 CHT Limit				
Byte	Bits	USB	BLE	Description
3:0	-	R/W	R	single precision floating point value

2.3.4 Oil Temperature Limit

0x300C OT Limit				
Byte	Bits	USB	BLE	Description
3:0	-	R/W	R	single precision floating point value

2.3.5 Exhaust Gas Temperature Limit

0x3010 EGT Limit				
Byte	Bits	USB	BLE	Description
3:0	-	R/W	R	single precision floating point value

2.4 Exceedances

2.4.1 Latched Exceedances

0x4000 Exceedances				
Byte	Bits	USB	BLE	Description
0	7:6	-	-	Unused
	5	R	R	Current Manifold Exceedance State
		W	-	0 – Clear Manifold Exceedance
	4	R	R	Current EGT Exceedance State
		W	-	0 – Clear EGT Exceedance
	3	R	R	Current OT Exceedance State
		W	-	0 – Clear OT Exceedance
	2	R	R	Current CHT Exceedance State
		W	-	0 – Clear CHT Exceedance
	1	R	R	Current Engine RPM Exceedance State
		W	-	0 – Clear Engine RPM Exceedance
	0	R	R	Current Rotor RPM Exceedance State
		W	-	0 – Clear Rotor RPM Exceedance
3:1	-	-	-	Unused

2.5 Log Data

2.5.1 Log Control

0x5000 Log Control				
Byte	Bits	USB	BLE	Description
0	7:1	-	-	Unused
	0	W	-	1 - Erases the log
3:1	-	-	-	Unused

This register provides some general control inputs for the log reading functionality.

- Writing a 1 to bit 0 of byte 0 will erase the log if communicating via the USB interface.

2.5.2 Log Entry Count

0x5010 Log Entry Count				
Byte	Bits	USB	BLE	Description
3:0	-	R	R	Current Log Entry Count

This register returns the current total number of Log Entries stored in the EMU.
Log entries are made at least once every second, with the timestamp stored in the first 6 bytes of the record.

2.5.3 Log Read Start Time

0x5020 Log Read Start Time				
Byte	Bits	USB	BLE	Description
3:0	-	R/W	R/W	UNIX Epoch time in seconds
5:4	-	R	R	Milliseconds

This register determines the time for the starting log entry to read.
Writing a value larger than the **Current Date Time** will set the **Log Read Start Time** to the last available record.
Writing a value smaller than the first log entry timestamp will set the **Log Read Start Time** to the first available record.
If the written time value corresponds to a log entry written during an exceedance period, the **Log Read Start Time** will be set to the first log entry in that exceedance period.

- *Initial value on power up is the time of the first log entry.*

2.5.4 Log Read Stop Time

0x5030 Log Read Stop Time				
Byte	Bits	USB	BLE	Description
3:0	-	R/W	R/W	UNIX Epoch time in seconds
5:4	-	R	R	Milliseconds

This register determines the log entry time to stop reading log entries at.
Writing a value larger than the **Current Date Time** will set the **Log Read Stop Time** to the last available record.
If the written time value corresponds to a log entry written during an exceedance period, the **Log Read Stop Time** will be set to the last log entry in that exceedance period.

- *Initial value on power up is the time of the last written log entry.*

2.5.5 Log Read Increment Time

0x5040 Log Read Increment Time				
Byte	Bits	USB	BLE	Description
3:0	-	R/W	R/W	Increment time in seconds

This register determines the time increment to step between log entries read out.
Writing a 0 to this register will read all entries – normal and exceedance

- *Initial value on power up is 0.*

2.5.6 Log Read Control

0x5100 Log Read Control				
Byte	Bits	USB	BLE	Description
0	7:1	-	-	Unused
	0	W	W	1 – Enable Automated Log Dump 0 – Disable Automated Log Dump
3:1	-	-	-	Unused
-	-	R	R	Get Next Log Entry

This register is the main transfer mechanism for Log Entries.

- Writing a 1 to bit 0 of byte 0 will enable the automated log dump feature
- Writing a 0 to bit 0 of byte 0 will disable the automated log dump feature and interrupt any current log dump in progress.

Issuing a read to the **Log Read Control** register returns:

- If the Automated Log Dump feature is disabled - the currently configured log entry.
- If the Automated Log Dump feature is enabled – all of the log entries from the start and stop time with the given time interval.

> *Automated Log Dump reads are done via notifications for the Bluetooth Interface.*
>
> *Changing the register address via the Bluetooth Interface will halt the log dump until a new read is issued to the **Log Read Control** register.*

A single log entry consists of 35 bytes of data in the following format:

Byte	Type	Units	Description
3:0	uint32	seconds	UNIX Epoch time in seconds
5:4	uint16	millisecond	Millisecond count
7:6	uint16	Hz	Rotor RPM
9:8	uint16	Hz	Engine RPM
11:10	int16	°C	Cylinder Head Temperature
13:12	int16	°C	Oil Temperature
15:14	int16	°C	EGT #1
17:16	int16	°C	EGT #2
19:18	int16	°C	EGT #3
21:20	int16	°C	EGT #4
23:22	int16	°C	EGT #5
25:24	int16	°C	EGT #6
27:26	int16	°C	Outside Air Temperature
29:28	int16	°C	Ambient Air Temperature
31:30	int16	kPa	Manifold Pressure
33:32	uint16	kPa	Ambient Pressure
34	uint8	-	Exceedance Flags

> *Note: the data for this register is longer than the BLE limit of 20 bytes, so reading via Bluetooth may require a Bluetooth Stack that supports long reads.*

2.5.7 Automated Log Dump

The Automated Log Dump feature can be used to extract a series of log entries from the MD46.

The entries transmitted are determined by the following parameters:

- Log Read Start Time
- Log Read End Time
- Log Read Increment Time

Once the Automated Log Dump feature is triggered, the MD46 will continuously transmit logs in sequence, starting with the timestamp identified in **Log Read Start Time** and stepping using the **Log Read Increment Time** value, until one of the following conditions are met:

- the time in **Log Read Stop Time** is reached
- The Automated Log Dump bit is cleared in **Log Read Control**
- For Bluetooth Only – The Register Control value is changed.

During this process, the **Log Read Start Time** register will be updated to reflect the next log entry to be read. This allows halting the Automated Log Dump feature and then resuming after the last transmitted location without having to rewrite the start and stop times.