MODBUS-TCP PROTOCOL

DIGIRAIL OEE – V1.2x B

Recommended for devices with firmware version V 1.2x and higher.

1 INTRODUCTION

DigiRail OEE is compatible with the Modbus-TCP protocol, a data communication protocol used to connect the device to supervisory control and data acquisition (SCADA) systems. It supports up to 3 simultaneous connections and allows up to 3 Modbus-TCP clients (masters) to monitor it at the same time. **DigiRail OEE** operates both as a Modbus-TCP server (slave) and as a TCP/RTU gateway.

As a server (slave), it responds to the configured Modbus RTU address. For address that diverge from the configured address value, it will operate as a TCP/RTU gateway. In this case, the packed will be sent to the RS485 interface and, if there is a reply from any Modbus RTU slave, replied to the Modbus-RTU client (master) that generated the request.

2 REGISTERS

2.1 COMMANDS

DigiRail OEE supports the following commands:

READ HOLDING REGISTERS – 0x03:

This command can be used to read the value of one or up to a maximum of 125 consecutive registers, as shown in the table below.

WRITE HOLDING REGISTERS – 0x06:

This command can be used to write to a register, as shown in the table below.

WRITE MULTIPLE HOLDING REGISTERS – 0x16:

This command can be used to write to multiple registers, as shown in the table below.

2.2 STATUS REGISTERS TABLE

DigiRail OEE supports the following status registers:

ADDRESS	REGISTER	DESCRIPTION	MINIMUM VALUE	MAXIMUM VALUE	TYPE
1	HR_PRODUCT_CODE	Product code.	510	510	RO
2	HR_SERIAL_NUMBER_H	Serial number (32bits).	0x0000	0xFFFF	RO
3	HR_SERIAL_NUMBER_L		0x0000	0xFFFF	RO
4	HR_FIRMWARE_VERSION	Version firmware x 100.	100	65535	RO
		Reserved.			
6	HR_MAC_ADDR_0_1	MAC Address. Hexadecimal format with two digits per register. 0 : 1 :2 : 3 : 4 : 5	0x0000	0xFFFF	RO
7	HR_MAC_ADDR_2_3		0x0000	0xFFFF	RO
8	HR_MAC_ADDR_4_5		0x0000	0xFFFF	RO
		Reserved.			
10	HR_USB_STATUS	USB interface status: 0 → Disconnected 1 → Connected	0	1	RO
		Reserved.			
13	HR_NUMBER_OF_ACTIVE_CH	Number of enabled analog channels.	0	6	RO
14	HR_NUMBER_OF_ACTIVE_CHD	Number of enabled digital channels.	0	6	RO
15	HR_RESET_COUNTERS	Reset of digital channel counters. **Note:** Write 1 resets all the digital counters that are configured to be reset by Modbus-TCP and MQTT.	0	1	RW
16	HR_PWR_STATUS	Power supply status: 0 → Powered by the USB interface 1 → Powered by external supply	1	1	RO
17	HR_STATUS_OF_RECORDS	Number of registers pending submission via MQTT protocol.	0	65535	RO
		Reserved.			
20	HR_LAST_CONFIG_YEAR,	Year of the last configuration.	2016	2080	RO
21	HR_LAST_CONFIG_MONTH,	Month of the last configuration.	1	12	RO
22	HR_LAST_CONFIG_DAY,	Day of the last configuration.	1	31	RO
23	HR_LAST_CONFIG_HOUR,	Hour of the last configuration.	0	23	RO
24	HR_LAST_CONFIG_MINUTE,	Minute of the last configuration.	0	59	RO
25	HR_LAST_CONFIG_SECOND	Second of the last configuration.	0	59	RO
26	HR_CURRENT_YEAR	Current year.	2016	2080	RO
27	HR_CURRENT_MONTH	Current month.	1	12	RO
28	HR_CURRENT_DAY	Current day.	1	31	RO
29	HR_CURRENT_HOUR	Current hour.	0	23	RO
30	HR_CURRENT_MINUTE	Current minute.	0	59	RO
31	HR_CURRENT_SECOND	Current second.	0	59	RO
		Reserved.			

ADDRESS	REGISTER	DESCRIPTION	MINIMUM VALUE	MAXIMUM VALUE	TYPE
34	HR_RESET_COUNTER_CHD1	Resets the digital channel counter 1. **Note:** Write 1 resets the counter for this channel if it is configured to allow reset via Modbus-TCP and MQTT protocols.	0	1	RW
35	HR_RESET_COUNTER_CHD2	Resets the digital channel counter 2. **Note:** Write 1 resets the counter for this channel if it is configured to allow reset via Modbus-TCP and MQTT protocols.	0	1	RW
36	HR_RESET_COUNTER_CHD3	Resets the digital channel counter 3. **Note:** Write 1 resets the counter for this channel if it is configured to allow reset via Modbus-TCP and MQTT protocols.	0	1	RW
37	HR_RESET_COUNTER_CHD4	Resets the digital channel counter 4. **Note:** Write 1 resets the counter for this channel if it is configured to allow reset via Modbus-TCP and MQTT protocols.	0	1	RW
38	HR_RESET_COUNTER_CHD5	Resets the digital channel counter 5. **Note:** Write 1 resets the counter for this channel if it is configured to allow reset via Modbus-TCP and MQTT protocols.	0	1	RW
39	HR_RESET_COUNTER_CHD6	Resets the digital channel counter 6. **Note:** Write 1 resets the counter for this channel if it is configured to allow reset via Modbus-TCP and MQTT protocols.	0	1	RW
		Reserved.			
41	HR_DIGITAL_OUT1_VALUE	Digital output status and control: 0 → OFF 1 → ON Allows you to read and write to the output.	0	1	RW
42	HR_DIGITAL_OUT2_VALUE	Digital output status and control: 0 → OFF 1 → ON Allows you to read and write to the output.	0	1	RW
		Reserved.			
45	HR_CHD1_STATUS	Digital channel status: 0 → Not configured 1 → OK 2 → The configuration has an error	0	2	RO
46	HR_CHD1_VALUE_HIGH	Counting mode: Counter value in 32-bit. Event mode: Logical input level.	0	65535	RO
47	HR_CHD1_VALUE_LOW		0	65535	RO
48	HR_CHD1_TIME_STAMP_LAST_HIGH	Last event timestamp. 32-bit. Unix format.	0x0000	0xFFFF	RO
49	HR_CHD1_TIME_STAMP_LAST_LOW		0x0000	0xFFFF	RO
		Reserved.			
56	HR_CHD2_STATUS	Digital channel status: 0 → Not configured 1 → OK 2 → The configuration has an error	0	2	RO
57	HR_CHD2_VALUE_HIGH	Counting mode: Counter value in 32-bit. Event mode: Logical input level.	0	65535	RO
58	HR_CHD2_VALUE_LOW		0	65535	RO
59	HR_CHD2_TIME_STAMP_LAST_HIGH	Last event timestamp. 32-bit. Unix format.	0x0000	0xFFFF	RO
60	HR_CHD2_TIME_STAMP_LAST_LOW		0x0000	0xFFFF	RO
		Reserved.			
67	HR_CHD3_STATUS	Digital channel status: 0 → Not configured 1 → OK 2 → The configuration has an error	0	2	RO
68	HR_CHD3_VALUE_HIGH	Counting mode: Counter value in 32-bit. Event mode: Logical input level.	0	65535	RO
69	HR_CHD3_VALUE_LOW		0	65535	RO
70	HR_CHD3_TIME_STAMP_LAST_HIGH	Last event timestamp. 32-bit. Unix format.	0x0000	0xFFFF	RO

ADDRESS	REGISTER	DESCRIPTION	MINIMUM VALUE	MAXIMUM VALUE	TYPE
71	HR_CHD3_TIME_STAMP_LAST_LOW		0x0000	0xFFFF	RO
		Reserved.			
78	HR_CHD4_STATUS	Digital channel status: 0 → Not configured 1 → OK 2 → The configuration has an error	0	2	RO
79	HR_CHD4_VALUE_HIGH	Counting mode: Counter value in 32-bit. Event mode: Logical input level.	0	65535	RO
80	HR_CHD4_VALUE_LOW		0	65535	RO
81	HR_CHD4_TIME_STAMP_LAST_HIGH	Last event timestamp. 32-bit. Unix format.	0x0000	0xFFFF	RO
82	HR_CHD4_TIME_STAMP_LAST_LOW		0x0000	0xFFFF	RO
		Reserved.			
89	HR_CHD5_STATUS	Digital channel status: 0 → Not configured 1 → OK 2 → The configuration has an error	0	2	RO
90	HR_CHD5_VALUE_HIGH	Counting mode: Counter value in 32-bit. Event mode: Logical input level.	0	65535	RO
91	HR_CHD5_VALUE_LOW		0	65535	RO
92	HR_CHD5_TIME_STAMP_LAST_HIGH	Last event timestamp. 32-bit. Unix format.	0x0000	0xFFFF	RO
93	HR_CHD5_TIME_STAMP_LAST_LOW		0x0000	0xFFFF	RO
		Reserved.			
100	HR_CHD6_STATUS	Digital channel status: 0 → Not configured 1 → OK 2 → The configuration has an error	0	2	RO
101	HR_CHD6_VALUE_HIGH	Counting mode: Counter value in 32-bit. Event mode: Logical input level.	0	65535	RO
102	HR_CHD6_VALUE_LOW		0	65535	RO
103	HR_CHD6_TIME_STAMP_LAST_HIGH	Last event timestamp. 32-bit. Unix format.	0x0000	0xFFFF	RO
104	HR_CHD6_TIME_STAMP_LAST_LOW		0x0000	0xFFFF	RO
		Reserved.			
109	HR_CH1_STATUS	Analog channel 1 status: 0 → Not configured 1 → OK 2 → The configuration has an error	0	2	RO
		Reserved.			
111	HR_CH1_MV_MA_VALUE_H	Value in the unit of measurement (mA or V). Float 32-bit format.	0x0000	0xFFFF	RO
112	HR_CH1_MV_MA_VALUE_L		0x0000	0xFFFF	RO
113	HR_CH1_SENSE_USER_RANGE_H	Value in user range. Float 32-bit format. **Note:** This is the same value as the cloud publication.	0x0000	0xFFFF	RO
114	HR_CH1_SENSE_USER_RANGE_L		0x0000	0xFFFF	RO
120	HR_CH2_STATUS	Analog channel 2 status: 0 → Not configured 1 → OK 2 → The configuration has an error	0	2	RO
		Reserved.			
122	HR_CH2_MV_MA_VALUE_H	Value in the unit of measurement (mA or V). Float 32-bit format.	0x0000	0xFFFF	RO
123	HR_CH2_MV_MA_VALUE_L		0x0000	0xFFFF	RO
124	HR_CH2_SENSE_USER_RANGE_H	Value in user range. Float 32-bit format. **Note:** This is the same value as the cloud publication.	0x0000	0xFFFF	RO
125	HR_CH2_SENSE_USER_RANGE_L		0x0000	0xFFFF	RO
		Reserved.			
130	HR_MQTT_LAST_UPDATE_YEAR	Year of last sending to the MQTT Broker.	1	1	RO
131	HR_MQTT_LAST_UPDATE_MONTH	Month of the last sending to the MQTT Broker.	1	12	RO
132	HR_MQTT_LAST_UPDATE_DAY	Day of the last sending to the MQTT Broker.	1	31	RO
133	HR_MQTT_LAST_UPDATE_HOUR	Time of the last sending to the MQTT Broker.	0	23	RO

ADDRESS	REGISTER	DESCRIPTION	MINIMUM VALUE	MAXIMUM VALUE	TYPE
134	HR_MQTT_LAST_UPDATE_MINUTE	Minute of the last sending to the MQTT Broker.	0	59	RO
135	HR_MQTT_LAST_UPDATE_SECOND	Second of the last sending to the MQTT Broker.	0	59	RO
136	HR_MQTT_STATUS_BROKER	Communication status with the MQTT Broker: 0 → Broker disconnected 1 → Broker connected 2 → DNS problem 3 → Broker error 4 → Connecting to the Broker	0	4	RO
		Reserved.			
139	HR_WIFI_RSSI	Signal quality between the device and the Wi-Fi Gateway displayed in percent. The higher the value, the better the signal.	0	65535	RO
		Reserved.			
141	HR_LAN_GATEWAY_COM_STATUS	ETH communication status: 0 → Gateway disconnected 1 → Gateway connected 2 → Wi-Fi provisioning error 3 → Obtaining IP via DHCP 4 → Error obtaining IP via DHCP	0	4	RO
142	HR_LAN_IP_ADDR_0_1	IPv4 address. Two octets per register. Dec 0 . Dec 1 . Dec 2 . Dec 3	0	65535	RO
143	HR_LAN_IP_ADDR_2_3		0	65535	RO
144	HR_LAN_MASK_ADDR_0_1	Mask. Two octets per register. Dec 0 . Dec 1 . Dec 2 . Dec 3	0	65535	RO
145	HR_LAN_MASK_ADDR_2_3		0	65535	RO
146	HR_LAN_GATEWAY_ADDR_0_1	Gateway. Two octets per register. Dec 0 . Dec 1 . Dec 2 . Dec 3	0	65535	RO
147	HR_LAN_GATEWAY_ADDR_2_3		0	65535	RO
148	HR_LAN_DNS_ADDR_0_1	DNS server IP. Two octets per register. Dec 0 . Dec 1 . Dec 2 . Dec 3	0	65535	RO
149	HR_LAN_DNS_ADDR_2_3		0	65535	RO
		Reserved.			
151	HR_LAN_IPV6_ADDR_0_1,	IPv6 address – Local. Hexadecimal format. 0_1 : 2_3 : 4_5 : 6_7 : 8_9 : 10_11 : 12_13 : 14_15	0	65535	RO
152	HR_LAN_IPV6_ADDR_2_3,		0	65535	RO
153	HR_LAN_IPV6_ADDR_4_5,		0	65535	RO
154	HR_LAN_IPV6_ADDR_6_7,		0	65535	RO
155	HR_LAN_IPV6_ADDR_8_9,		0	65535	RO
156	HR_LAN_IPV6_ADDR_10_11,		0	65535	RO
157	HR_LAN_IPV6_ADDR_12_13,		0	65535	RO
158	HR_LAN_IPV6_ADDR_14_15,		0	65535	RO
159	HR_LAN_IPV6_GLOBAL_ADDR_0_1,	IPv6 address – Global. Hexadecimal format. 0_1 : 2_3 : 4_5 : 6_7 : 8_9 : 10_11 : 12_13 : 14_15	0	65535	RO
160	HR_LAN_IPV6_GLOBAL_ADDR_2_3,		0	65535	RO
161	HR_LAN_IPV6_GLOBAL_ADDR_4_5,		0	65535	RO
162	HR_LAN_IPV6_GLOBAL_ADDR_6_7,		0	65535	RO
163	HR_LAN_IPV6_GLOBAL_ADDR_8_9,		0	65535	RO
164	HR_LAN_IPV6_GLOBAL_ADDR_10_11,		0	65535	RO
165	HR_LAN_IPV6_GLOBAL_ADDR_12_13,		0	65535	RO
166	HR_LAN_IPV6_GLOBAL_ADDR_14_15,		0	65535	RO
167	HR_CHD1_LEVEL,	Logical level of digital input 1.	0	1	RO
168	HR_CHD2_LEVEL,	Logical level of digital input 2.	0	1	RO
169	HR_CHD3_LEVEL,	Logical level of digital input 3.	0	1	RO
170	HR_CHD4_LEVEL,	Logical level of digital input 4.	0	1	RO
171	HR_CHD5_LEVEL,	Logical level of digital input 5.	0	1	RO
172	HR_CHD6_LEVEL,	Logical level of digital input 6.	0	1	RO
		Reserved.			
174	HR_CHD1_SETVALUE_H	Changes the value of the 32-bit counter for channel 1.	0	65535	RW

ADDRESS	REGISTER	DESCRIPTION	MINIMUM VALUE	MAXIMUM VALUE	TYPE
175	HR_CHD1_SETVALUE_L		0	65535	RW
176	HR_CHD2_SETVALUE_H	Changes the value of the 32-bit counter for channel 2.	0	65535	RW
177	HR_CHD2_SETVALUE_L		0	65535	RW
178	HR_CHD3_SETVALUE_H	Changes the value of the 32-bit counter for channel 3.	0	65535	RW
179	HR_CHD3_SETVALUE_L		0	65535	RW
180	HR_CHD4_SETVALUE_H	Changes the value of the 32-bit counter for channel 4.	0	65535	RW
181	HR_CHD4_SETVALUE_L,		0	65535	RW
182	HR_CHD5_SETVALUE_H	Changes the value of the 32-bit counter for channel 5.	0	65535	RW
183	HR_CHD5_SETVALUE_L		0	65535	RW
184	HR_CHD6_SETVALUE_H	Changes the value of the 32-bit counter for channel 6.	0	65535	RW
185	HR_CHD6_SETVALUE_L		0	65535	RW
		Reserved.			
187	HR_SS_COLLECT_RECORD_MAX_QTTY	Maximum number of downloads supported by memory.	1824	7096	RO
188	HR_SS_COLLECT_LAST_RECORD	Position of the last download added to memory.	0	7096	RO
189	HR_SS_COLLECT_FIRST_RECORD	Position of the first download added to memory.	0	7096	RO
190	HR_SS_COLLECT_REQUESTED_RECORD	Position of the download requested for reading.	0	7096	RW
191	HR_SS_COLLECT_TIMESTAMP_UNIX_H	Timestamp of the requested download in Unix format.	0	65535	RO
192	HR_SS_COLLECT_TIMESTAMP_UNIX_L		0	65535	RO
193	HR_SS_COLLECT_TIMESTAMP_MS	Timestamp of the requested download in milliseconds.	0	65535	RO
194	HR_SS_COLLECT_CHD_EVENT_INDEX	When an event occurs in the requested download, returns the index of the digital channel: 0 → No event. It is a periodic log 1 → Event on channel 1 2 → Event on channel 2 3 → Event on channel 3 4 → Event on channel 4 5 → Event on channel 5 6 → Event on channel 6	0	6	RO
195	HR_SS_COLLECT_CHD_EVENT_TYPE	When an event occurs in the requested download, returns the event type: 0 → No event. It is a periodic log 1 → Falling edge event of the digital channel 2 → Rising edge event of the digital channel	0	2	RO
196	HR_SS_COLLECT_CHD1_VALUE_H	Value of digital channel 1 in the requested download.	0	65535	RO
197	HR_SS_COLLECT_CHD1_VALUE_L		0	65535	RO
198	HR_SS_COLLECT_CHD2_VALUE_H	Value of digital channel 2 in the requested download.	0	65535	RO
199	HR_SS_COLLECT_CHD2_VALUE_L		0	65535	RO
200	HR_SS_COLLECT_CHD3_VALUE_H	Value of digital channel 3 in the requested download.	0	65535	RO
201	HR_SS_COLLECT_CHD3_VALUE_L		0	65535	RO
202	HR_SS_COLLECT_CHD4_VALUE_H	Value of digital channel 4 in the requested download.	0	65535	RO
203	HR_SS_COLLECT_CHD4_VALUE_L		0	65535	RO
204	HR_SS_COLLECT_CHD5_VALUE_H	Value of digital channel 5 in the requested download.	0	65535	RO
205	HR_SS_COLLECT_CHD5_VALUE_L		0	65535	RO
206	HR_SS_COLLECT_CHD6_VALUE_H	Value of digital channel 6 in the requested download.	0	65535	RO
207	HR_SS_COLLECT_CHD6_VALUE_L		0	65535	RO
208	HR_SS_COLLECT_CH1_SENSE_USER_ RANGE_H	Displays the sensor value in the user range of analog channel 1 (in Float).	0	65535	RO
209	HR_SS_COLLECT_CH1_SENSE_USER_ RANGE_L		0	65535	RO
210	HR_SS_COLLECT_CH2_SENSE_USER_ RANGE_H	Displays the sensor value in the user range of analog channel 2 (in Float).	0	65535	RO
211	HR_SS_COLLECT_CH2_SENSE_USER_ RANGE_L		0	65535	RO
		Reserved.			

ADDRESS	REGISTER	DESCRIPTION	MINIMUM VALUE	MAXIMUM VALUE	TYPE
216	HR_SS_WIFI_RSSI_MIN	Minimum value indicated by the HR_WIFI_RSSI register. You can reset the value using the HR_RESET_DIAG_RSSI register.	0	65535	RO
217	HR_SS_WIFI_RSSI_MAX	Maximum value indicated by the HR_WIFI_RSSI register. You can reset the value using the HR_RESET_DIAG_RSSI register.	0	65535	RO
218	HR_SS_WIFI_RSSI_AVERAGE	Average value indicated by the HR_WIFI_RSSI register. You can reset the value using the HR_RESET_DIAG_RSSI register.	0	65535	RO
		Reserved.			
221	HR_RESET_COUNTER_WDT	Resets the system Watchdog diagnostic counters.	0	1	RW
222	HR_RESET_COUNTER_LOGS	Resets the system logs diagnostic counters.	0	1	RW
223	HR_RESET_DIAG_RSSI	Resets the minimum, maximum and average signal quality (RSSI) measurement	0	1	RW

Table 1 – Registers table

2.3 CONFIGURATION REGISTERS TABLE

DigiRail OEE supports the following configuration registers:

ADDRESS	REGISTER	DESCRIPTION	MINIMUM VALUE	MAXIMUM VALUE	TYPE
1000	HR_CS_SETTING_RESTORE_DEFAULT	Allows you to return all configuration parameters to the factory defaults.	0	1	RW
1001	HR_CS_SETTING_PUBLISH_INTERVAL_S	Allows you to configure the publishing interval of the device.	1	65535	RW
1002	HR_CS_APPLY_CONFIGURATION	Allows you to instantly apply the settings.	0	1	RW
1003	HR_CS_LOCATION_0	Allows you to define the location of the device. Each register corresponds to two characters	0x0000	0xFFFF	RW
1004	HR_CS_LOCATION_1		0x0000	0xFFFF	RW
1005	HR_CS_LOCATION_2		0x0000	0xFFFF	RW
1006	HR_CS_LOCATION_3		0x0000	0x00FF	RW
1007	HR_CS_LOCATION_4		0x0000	0xFFFF	RW
1008	HR_CS_LOCATION_5		0x0000	0xFFFF	RW
1009	HR_CS_LOCATION_6		0x0000	0xFFFF	RW
1010	HR_CS_LOCATION_7		0x0000	0xFFFF	RW
1011	HR_CS_LOCATION_8		0x0000	0xFFFF	RW
1012	HR_CS_LOCATION_9		0x0000	0xFFFF	RW
1013	HR_CS_LOCATION_10		0x0000	0xFFFF	RW
1014	HR_CS_LOCATION_11		0x0000	0xFFFF	RW
1015	HR_CS_LOCATION_12		0x0000	0xFFFF	RW
1016	HR_CS_LOCATION_13		0x0000	0xFFFF	RW
1017	HR_CS_LOCATION_14		0x0000	0xFFFF	RW
1018	HR_CS_LOCATION_15		0x0000	0xFFFF	RW
1019	HR_CS_LOCATION_16		0x0000	0xFFFF	RW
1020	HR_CS_LOCATION_17		0x0000	0xFFFF	RW
1021	HR_CS_LOCATION_18		0x0000	0xFFFF	RW
1022	HR_CS_LOCATION_19		0x0000	0xFFFF	RW
1023	HR_CS_SETTING_ENABLE_ ALTERNATIVE_PUB_INTERVAL	Allows you to enable an alternative publishing interval whenever there are connection problems.	0	1	RW
1024	HR_CS_SETTING_ALTERNATIVE_PUB_ INTERVAL_S	Allows you to configure an alternative publishing interval whenever there are connection problems.	60	65535	RW
1025	HR_CS_SETTING_FORCE_RESET_ COUNTERS	Allows you to reset the digital channel counters, regardless of the channels permission.	0	1	RW
		Reserved.			
1027	HR_CS_SETTING_TITLE_1	Allows you to configure a name for the device.	0x0000	0xFFFF	RW

ADDRESS	REGISTER	DESCRIPTION	MINIMUM VALUE	MAXIMUM VALUE	TYPE
1028	HR_CS_SETTING_TITLE_2	Each register corresponds to two characters.	0x0000	0xFFFF	RW
1029	HR_CS_SETTING_TITLE_3		0x0000	0xFFFF	RW
1030	HR_CS_SETTING_TITLE_4		0x0000	0xFFFF	RW
1031	HR_CS_SETTING_TITLE_5		0x0000	0xFFFF	RW
1032	HR_CS_SETTING_TITLE_6		0x0000	0xFFFF	RW
1033	HR_CS_SETTING_TITLE_7		0x0000	0xFFFF	RW
1034	HR_CS_SETTING_TITLE_8		0x0000	0xFFFF	RW
1035	HR_CS_SETTING_TITLE_9		0x0000	0xFFFF	RW
1036	HR_CS_SETTING_TITLE_10		0x0000	0xFFFF	RW
		Reserved.			
1038	HR_CS_SETTING_GMT	Allows you to configure GMT.	0x0000	0xFFFF	RW
1039	HR_CS_SETTING_YEAR	Allows you to configure the year (RTC in GMT 0).	2016	2080	RW
1040	HR_CS_SETTING_MONTH	Allows you to configure the month (RTC in GMT 0).	1	12	RW
1041	HR_CS_SETTING_DAY	Allows you to configure the day (RTC in GMT 0).	1	31	RW
1042	HR_CS_SETTING_HOUR	Allows you to configure the hour (RTC in GMT 0).	0	23	RW
1043	HR_CS_SETTING_MINUTE	Allows you to configure the minute (RTC in GMT 0).	0	59	RW
1044	HR_CS_SETTING_SECOND	Allows you to configure the second (RTC in GMT 0).	0	59	RW
		Reserved.			
1046	HR_CS_CHD_ADD_COUNTER_AT_EVENT	Allows you to add the digital channel count value to the event log.	0	1	RW
1047	HR_CS_CHD_RESET_TYPE	Allows you to define the reset mode of the digital counters: 0 → Daily 1 → Weekly 2 → Monthly	0	2	RW
1048	HR_CS_CHD_RESET_DAY	Allows you to configure the reset day for the digital counters according to what was configured in register 1060.	1	31	RW
1049	HR_CS_CHD_RESET_HOUR	Allows you to configure the reset hour for the digital counters according to what was configured in register 1060.	0	23	RW
1050	HR_CS_CHD_RESET_MINUTE	Allows you to configure the reset minute for the digital counters according to what was configured in register 1060.	0	59	RW
1051	HR_CS_CHD_RESET_SECOND	Allows you to configure the reset second for the digital counters according to what was configured in register 1060.	0	59	RW
1052	HR_CS_CHD_RESET_WEEK_DAY	Allows you to configure the reset week for the digital counters according to what was configured in register 1060.	1	7	RW
1053	HR_CS_CHD1_ENABLED	Allows you to enable digital channel 1.	0	1	RW
1054	HR_CS_CHD1_COUNTING_MODE	Allows you to configure the counting mode for digital channel 1: 0 → Not defined 1 → Counter 2 → Event	0	2	RW
1055	HR_CS_CHD1_SENSOR_TYPE	Allows you to configure the sensor type of digital channel 1: 0 → Not configured 1 → PNP 2 → NPN 3 → Dry contact	0	3	RW
1056	HR_CS_CHD1_COUNTING_EDGE	Allows you to configure the counting edge of digital channel 1: 1 → Rising edge 2 → Falling edge 3 → Both edges	1	3	RW
1057	HR_CS_CHD1_DEBOUNCE_TIME_ms	Allows you to configure the digital channel 1 Debounce time for the Dry Contact sensor type (in milliseconds).	0	60000	RW
		Reserved.			
1060	HR_CS_CHD1_RESET_MODE	Allows you to configure the reset mode of the accumulators of digital channel 1: Bit 0 → Overflow Bit 1 → Calendar Bit 2 → Protocol	0	2	RW
1061	HR_CS_CHD1_DEBOUNCE_ENABLE	Allows you to enable Debounce for digital channel 1.	0	1	RW
1062	HR_CS_CHD1_CHANGE_VALUE_ENABLE	Allows you to enable the permission to change the digital channel 1 count value via Modbus-TCP or MQTT protocols.	0	1	RW

ADDRESS	REGISTER	DESCRIPTION	MINIMUM VALUE	MAXIMUM VALUE	TYPE
		Reserved.			
1067	HR_CS_CHD2_ENABLED	Allows you to enable digital channel 2.	0	1	RW
1068	HR_CS_CHD2_COUNTING_MODE	Allows you to configure the counting mode for digital channel 2: 0 → Not defined 1 → Counter 2 → Event	0	2	RW
1069	HR_CS_CHD2_SENSOR_TYPE	Allows you to configure the sensor type of digital channel 2: 0 → Not configured 1 → PNP 2 → NPN 3 → Dry contact	0	3	RW
1070	HR_CS_CHD2_COUNTING_EDGE	Allows you to configure the counting edge of digital channel 2: 1 → Rising edge 2 → Falling edge 3 → Both edges	1	3	RW
1071	HR_CS_CHD2_DEBOUNCE_TIME_ms	Allows you to configure the digital channel 2 Debounce time for the Dry Contact sensor type (in milliseconds).	0	60000	RW
		Reserved.			
1074	HR_CS_CHD2_RESET_MODE	Allows you to configure the reset mode of the accumulators of digital channel 2: Bit 0 → Overflow Bit 1 → Calendar Bit 2 → Protocol	0	3	RW
1075	HR_CS_CHD2_DEBOUNCE_ENABLE	Allows you to enable Debounce for digital channel 2.	0	1	RW
1076	HR_CS_CHD2_CHANGE_VALUE_ENABLE	Allows you to enable the permission to change the digital channel 2 count value via Modbus-TCP or MQTT protocols.	0	1	RW
		Reserved.			
1081	HR_CS_CHD3_ENABLED	Allows you to enable digital channel 3.	0	1	RW
1082	HR_CS_CHD3_COUNTING_MODE	Allows you to configure the counting mode for digital channel 3: 0 → Not defined 1 → Counter 2 → Event	0	2	RW
1083	HR_CS_CHD3_SENSOR_TYPE	Allows you to configure the sensor type of digital channel 3: 0 → Not configured 1 → PNP 2 → NPN 3 → Dry contact	0	3	RW
1084	HR_CS_CHD3_COUNTING_EDGE	Allows you to configure the counting edge of digital channel 3: 1 → Rising edge 2 → Falling edge 3 → Both edges	1	3	RW
1085	HR_CS_CHD3_DEBOUNCE_TIME_ms	Allows you to configure the digital channel 3 Debounce time for the Dry Contact sensor type (in milliseconds).	0	60000	RW
		Reserved.			
1088	HR_CS_CHD3_RESET_MODE	Allows you to configure the reset mode of the accumulators of digital channel 3: Bit 0 → Overflow Bit 1 → Calendar Bit 2 → Protocol	0	2	RW
1089	HR_CS_CHD3_DEBOUNCE_ENABLE	Allows you to enable Debounce for digital channel 3.	0	1	RW
1090	HR_CS_CHD3_CHANGE_VALUE_ENABLE	Allows you to enable the permission to change the digital channel 3 count value via Modbus-TCP or MQTT protocols.	0	1	RW
		Reserved.			
1095	HR_CS_CHD4_ENABLED	Allows you to enable digital channel 4.	0	1	RW
1096	HR_CS_CHD4_COUNTING_MODE	Allows you to configure the counting mode for digital channel 4: 0 → Not defined 1 → Counter 2 → Event	0	2	RW
1097	HR_CS_CHD4_SENSOR_TYPE	Allows you to configure the sensor type of digital channel 4: 0 → Not configured 1 → PNP 2 → NPN	0	3	RW

ADDRESS	REGISTER	DESCRIPTION	MINIMUM VALUE	MAXIMUM VALUE	TYPE
		3 → Dry contact			
1098	HR_CS_CHD4_COUNTING_EDGE	Allows you to configure the counting edge of digital channel 4: 1 → Rising edge 2 → Falling edge 3 → Both edges	1	3	RW
1099	HR_CS_CHD4_DEBOUNCE_TIME_ms	Allows you to configure the digital channel 4 Debounce time for the Dry Contact sensor type (in milliseconds).	0	60000	RW
		Reserved.			
1102	HR_CS_CHD4_RESET_MODE	Allows you to configure the reset mode of the accumulators of digital channel 4: Bit 0 → Overflow Bit 1 → Calendar Bit 2 → Protocol	0	2	RW
1103	HR_CS_CHD4_DEBOUNCE_ENABLE	Allows you to enable Debounce for digital channel 4.	0	1	RW
1104	HR_CS_CHD4_CHANGE_VALUE_ENABLE	Allows you to enable the permission to change the digital channel 4 count value via Modbus-TCP or MQTT protocols.	0	1	RW
		Reserved.			
1109	HR_CS_CHD5_ENABLED	Allows you to enable digital channel 5.	0	1	RW
1110	HR_CS_CHD5_COUNTING_MODE	Allows you to configure the counting mode for digital channel 5: 0 → Not defined 1 → Counter 2 → Event	0	2	RW
1111	HR_CS_CHD5_SENSOR_TYPE	Allows you to configure the sensor type of digital channel 5: 0 → Not configured 1 → PNP 2 → NPN 3 → Dry contact	0	3	RW
1112	HR_CS_CHD5_COUNTING_EDGE	Allows you to configure the counting edge of digital channel 5: 1 → Rising edge 2 → Falling edge 3 → Both edges	1	3	RW
1113	HR_CS_CHD5_DEBOUNCE_TIME_ms	Allows you to configure the digital channel 5 Debounce time for the Dry Contact sensor type (in milliseconds).	0	60000	RW
		Reserved.			
1116	HR_CS_CHD5_RESET_MODE	Allows you to configure the reset mode of the accumulators of digital channel 5: Bit 0 → Overflow Bit 1 → Calendar Bit 2 → Protocol	0	2	RW
1117	HR_CS_CHD5_DEBOUNCE_ENABLE	Allows you to enable Debounce for digital channel 5.	0	1	RW
1118	HR_CS_CHD5_CHANGE_VALUE_ENABLE	Allows you to enable the permission to change the digital channel 5 count value via Modbus-TCP or MQTT protocols.	0	1	RW
		Reserved.			
1123	HR_CS_CHD6_ENABLED	Allows you to enable digital channel 6.	0	1	RW
1124	HR_CS_CHD6_COUNTING_MODE	Allows you to configure the counting mode for digital channel 6: 0 → Not defined 1 → Counter 2 → Event	0	2	RW
1125	HR_CS_CHD6_SENSOR_TYPE	Allows you to configure the sensor type of digital channel 6: 0 → Not configured 1 → PNP 2 → NPN 3 → Dry contact	0	3	RW
1126	HR_CS_CHD6_COUNTING_EDGE	Allows you to configure the counting edge of digital channel 6: 1 → Rising edge 2 → Falling edge 3 → Both edges	1	3	RW
1127	HR_CS_CHD6_DEBOUNCE_TIME_ms	Allows you to configure the digital channel 6 Debounce time for the Dry Contact sensor type (in milliseconds).	0	60000	RW
		Reserved.			
1130	HR_CS_CHD6_RESET_MODE	Allows you to configure the reset mode of the accumulators of digital channel 6:	0	2	RW

ADDRESS	REGISTER	DESCRIPTION	MINIMUM VALUE	MAXIMUM VALUE	TYPE
		Bit 0 → Overflow Bit 1 → Calendar Bit 2 → Protocol			
1131	HR_CS_CHD6_DEBOUNCE_ENABLE	Allows you to enable Debounce for digital channel 6.	0	1	RW
1132	HR_CS_CHD6_CHANGE_VALUE_ENABLE	Allows you to enable the permission to change the digital channel 6 count value via Modbus-TCP or MQTT protocols.	0	1	RW
		Reserved.			
1138	HR_CS_CH1_ENABLE	Allows you to enable analog channel 1.	0	1	RW
		Reserved.			
1140	HR_CS_CH1_SENSOR_TYPE	Allows you to configure the sensor type of analog channel 1: 0 → Not defined 0 → 0-5 V 2 → 0-10 V 3 → 0-20 mA 4 → 4-20 mA	0	4	RW
		Reserved.			
1142	HR_CS_CH1_RANGE_MIN_HIGH	Allows you to configure the minimum limit of analog channel 1.	0	0xFFFF	RW
1143	HR_CS_CH1_RANGE_MIN_LOW		0	0xFFFF	RW
1144	HR_CS_CH1_RANGE_MAX_HIGH	Allows you to configure the maximum limit of analog channel 1.	0	0xFFFF	RW
1145	HR_CS_CH1_RANGE_MAX_LOW		0	0xFFFF	RW
1146	HR_CS_CH1_DECIMAL_POINT	Allows you to configure the decimal place of analog channel 1 (fixed point for display and memory register): 0 → No decimal places 1 → One decimal place 2 → Two decimal places	0	2	RW
1147	HR_CS_CH1_ERROR_VALUE_HIGH	Allows you to configure a value for error indication on analog channel 1.	0	0xFFFF	RW
1148	HR_CS_CH1_ERROR_VALUE_LOW		0	0xFFFF	RW
		Reserved.			
1158	HR_CS_CH2_ENABLE	Allows you to enable analog channel 2.	0	1	RW
		Reserved.			
1160	HR_CS_CH2_SENSOR_TYPE	Allows you to configure the sensor type of analog channel 2: 0 → Not defined 0 → 0-5 V 2 → 0-10 V 3 → 0-20 mA 4 → 4-20 mA	0	4	RW
		Reserved.			
1162	HR_CS_CH2_RANGE_MIN_HIGH	Allows you to configure the minimum limit of analog channel 2.	0	0xFFFF	RW
1163	HR_CS_CH2_RANGE_MIN_LOW		0	0xFFFF	RW
1164	HR_CS_CH2_RANGE_MAX_HIGH	Allows you to configure the maximum limit of analog channel 2.	0	0xFFFF	RW
1165	HR_CS_CH2_RANGE_MAX_LOW		0	0xFFFF	RW
1166	HR_CS_CH2_DECIMAL_POINT	Allows you to configure the decimal place of analog channel 1 (fixed point for display and memory register): 0 → No decimal places 1 → One decimal place 2 → Two decimal places	0	2	RW
1167	HR_CS_CH2_ERROR_VALUE_HIGH	Allows you to configure a value for error indication on analog channel 2.	0	0xFFFF	RW
1168	HR_CS_CH2_ERROR_VALUE_LOW		0	0xFFFF	RW
		Reserved.			
1172	HR_CS_MODBUS_TCP_ENABLE	Allows you to enable the Modbus-TCP protocol.	0	1	RW
1173	HR_CS_MODBUS_TCP_PORT	Allows you to configure the Modbus-TCP protocol communication port.	0	0xFFFF	RW
1174	HR_CS_MODBUS_TCP_ADDRESS	Allows you to configure the Modbus address for the device. For different values, it will forward the data to RS485, acting as a Gateway.	1	255	RW
1175	HR_CS_RS485_BAUD_RATE,	Allows you to configure the RS485 interface Baud Rate: 0 → 1200 1 → 2400	0	7	RW

ADDRESS	REGISTER	DESCRIPTION	MINIMUM VALUE	MAXIMUM VALUE	TYPE
		2 → 4800 3 → 9600 4 → 19200 5 → 38400 6 → 57600 7 → 115200			
1176	HR_CS_RS485_STOP_BITS,	Allows you to configure the RS485 interface Stop Bits: 0 → 1 Stop Bit 1 → 2 Stop Bits	0	1	RW
1177	HR_CS_RS485_PARITY,	Allows you to configure the RS485 interface parity: 0 → No parity 1 → Odd parity 2 → Even parity	0	2	RW
1178	HR_CS_RS485_TIMEOUT	Allows you to configure a timeout value for the connection (in milliseconds).	0	65535	RW
		Reserved.			
1180	HR_CS_ETH_IP_GET_ADRESS	Allows you to define the way in which the device will get its IP: 0 → Static 1 → DHCP	0	1	RW
		Reserved.			
1182	HR_CS_ETH_IPV4_ADDR_0_1	Allows you to configure the device IPv4 address. Two octets per register. Dec 0 . Dec 1 . Dec 2 . Dec 3	0	65535	RW
1183	HR_CS_ETH_IPV4_ADDR_2_3		0	65535	RW
1184	HR_CS_ETH_IPV4_MASK_ADDR_0_1	Allows you to define the network mask. Two octets per register. Dec 0 . Dec 1 . Dec 2 . Dec 3	0	65535	RW
1185	HR_CS_ETH_IPV4_MASK_ADDR_2_3		0	65535	RW
1186	HR_CS_ETH_IPV4_GATEWAY_ADDR_0_1	Allows you to configure the network Gateway address. Two octets per register. Dec 0 . Dec 1 . Dec 2 . Dec 3	0	65535	RW
1187	HR_CS_ETH_IPV4_GATEWAY_ADDR_2_3		0	65535	RW
1188	HR_CS_ETH_IPV4_DNS_ADDR_0_1	Allows you to configure the DNS server IP. Two octets per register. Dec 0 . Dec 1 . Dec 2 . Dec 3	0	65535	RW
1189	HR_CS_ETH_IPV4_DNS_ADDR_2_3		0	65535	RW
1190	HR_CS_ETH_IPV6_PREFIX	Allows you to configure the IPv6 prefix.	0	128	RW
1191	HR_CS_ETH_IPV6_ADDR_0_1,	Allows to configure the IPv6 - Local address. Hexadecimal format. 0_1 : 2_3 : 4_5 : 6_7 : 8_9 : 10_11 : 12_13 : 14_15T	0	65535	RW
1192	HR_CS_ETH_IPV6_ADDR_2_3,		0	65535	RW
1193	HR_CS_ETH_IPV6_ADDR_4_5,		0	65535	RW
1194	HR_CS_ETH_IPV6_ADDR_6_7,		0	65535	RW
1195	HR_CS_ETH_IPV6_ADDR_8_9,		0	65535	RW
1196	HR_CS_ETH_IPV6_ADDR_10_11,		0	65535	RW
1197	HR_CS_ETH_IPV6_ADDR_12_13,		0	65535	RW
1198	HR_CS_ETH_IPV6_ADDR_14_15,		0	65535	RW
1199	HR_CS_ETH_IPV6_DNS_ADDR_0_1,	Allows to configure the IPv6 - Global. Hexadecimal format. 0_1 : 2_3 : 4_5 : 6_7 : 8_9 : 10_11 : 12_13 : 14_15	0	65535	RW
1200	HR_CS_ETH_IPV6_DNS_ADDR_2_3,		0	65535	RW
1201	HR_CS_ETH_IPV6_DNS_ADDR_4_5,		0	65535	RW
1202	HR_CS_ETH_IPV6_DNS_ADDR_6_7,		0	65535	RW
1203	HR_CS_ETH_IPV6_DNS_ADDR_8_9,		0	65535	RW
1204	HR_CS_ETH_IPV6_DNS_ADDR_10_11,		0	65535	RW
1205	HR_CS_ETH_IPV6_DNS_ADDR_12_13,		0	65535	RW
1206	HR_CS_ETH_IPV6_DNS_ADDR_14_15,		0	65535	RW
1207	HR_CS_WIFI_ROUTER_TITLE_1	Allows you to define the Wi-Fi network SSID. Each register is equivalent to 2 characters.	0x0000	0xFFFF	RW
1208	HR_CS_WIFI_ROUTER_TITLE_2		0x0000	0xFFFF	RW
1209	HR_CS_WIFI_ROUTER_TITLE_3		0x0000	0xFFFF	RW
1210	HR_CS_WIFI_ROUTER_TITLE_4		0x0000	0xFFFF	RW
1211	HR_CS_WIFI_ROUTER_TITLE_5		0x0000	0xFFFF	RW
1212	HR_CS_WIFI_ROUTER_TITLE_6		0x0000	0xFFFF	RW

ADDRESS	REGISTER	DESCRIPTION	MINIMUM VALUE	MAXIMUM VALUE	TYPE
1213	HR_CS_WIFI_ROUTER_TITLE_7		0x0000	0xFFFF	RW
1214	HR_CS_WIFI_ROUTER_TITLE_8		0x0000	0xFFFF	RW
1215	HR_CS_WIFI_ROUTER_TITLE_9		0x0000	0xFFFF	RW
1216	HR_CS_WIFI_ROUTER_TITLE_10		0x0000	0xFFFF	RW
1217	HR_CS_WIFI_ROUTER_TITLE_11		0x0000	0xFFFF	RW
1218	HR_CS_WIFI_ROUTER_TITLE_12		0x0000	0xFFFF	RW
1219	HR_CS_WIFI_ROUTER_TITLE_13		0x0000	0xFFFF	RW
1220	HR_CS_WIFI_ROUTER_TITLE_14		0x0000	0xFFFF	RW
1221	HR_CS_WIFI_ROUTER_TITLE_15		0x0000	0xFFFF	RW
1222	HR_CS_WIFI_ROUTER_TITLE_16		0x0000	0xFFFF	RW
		Reserved.			
1224	HR_CS_WIFI_ROUTER_PASSWORD_1		0x0000	0xFFFF	RW
1225	HR_CS_WIFI_ROUTER_PASSWORD_2		0x0000	0xFFFF	RW
1226	HR_CS_WIFI_ROUTER_PASSWORD_3		0x0000	0xFFFF	RW
1227	HR_CS_WIFI_ROUTER_PASSWORD_4		0x0000	0xFFFF	RW
1228	HR_CS_WIFI_ROUTER_PASSWORD_5		0x0000	0xFFFF	RW
1229	HR_CS_WIFI_ROUTER_PASSWORD_6		0x0000	0xFFFF	RW
1230	HR_CS_WIFI_ROUTER_PASSWORD_7		0x0000	0xFFFF	RW
1231	HR_CS_WIFI_ROUTER_PASSWORD_8		0x0000	0xFFFF	RW
1232	HR_CS_WIFI_ROUTER_PASSWORD_9		0x0000	0xFFFF	RW
1233	HR_CS_WIFI_ROUTER_PASSWORD_10		0x0000	0xFFFF	RW
1234	HR_CS_WIFI_ROUTER_PASSWORD_11	Allows you to define the Wi-Fi network password. Each register is equivalent to 2 characters.	0x0000	0xFFFF	RW
1235	HR_CS_WIFI_ROUTER_PASSWORD_12		0x0000	0xFFFF	RW
1236	HR_CS_WIFI_ROUTER_PASSWORD_13		0x0000	0xFFFF	RW
1237	HR_CS_WIFI_ROUTER_PASSWORD_14		0x0000	0xFFFF	RW
1238	HR_CS_WIFI_ROUTER_PASSWORD_15		0x0000	0xFFFF	RW
1239	HR_CS_WIFI_ROUTER_PASSWORD_16		0x0000	0xFFFF	RW
1240	HR_CS_WIFI_ROUTER_PASSWORD_17		0x0000	0xFFFF	RW
1241	HR_CS_WIFI_ROUTER_PASSWORD_18		0x0000	0xFFFF	RW
1242	HR_CS_WIFI_ROUTER_PASSWORD_19		0x0000	0xFFFF	RW
1243	HR_CS_WIFI_ROUTER_PASSWORD_20		0x0000	0xFFFF	RW
1244	HR_CS_WIFI_ROUTER_PASSWORD_21		0x0000	0xFFFF	RW
		Reserved.			
1246	HR_CS_MQTT_ENABLE	Allows you to enable the MQTT protocol.	0	1	RW
1247	HR_CS_MQTT_QOS	Allows you to configure the QoS for sending messages.	0	1	RW
1248	HR_CS_MQTT_PORT	Allows you to configure the port to be used for the MQTT protocol.	0	65535	RW
1249	HR_CS_MQTT_ROUTER_TITLE_1		0x0000	0xFFFF	RW
1250	HR_CS_MQTT_ROUTER_TITLE_2		0x0000	0xFFFF	RW
1251	HR_CS_MQTT_ROUTER_TITLE_3		0x0000	0xFFFF	RW
1252	HR_CS_MQTT_ROUTER_TITLE_4		0x0000	0xFFFF	RW
1253	HR_CS_MQTT_ROUTER_TITLE_5		0x0000	0xFFFF	RW
1254	HR_CS_MQTT_ROUTER_TITLE_6		0x0000	0xFFFF	RW
1255	HR_CS_MQTT_ROUTER_TITLE_7	Allows you to enter the Broker user. Each register is equivalent to 2 characters.	0x0000	0xFFFF	RW
1256	HR_CS_MQTT_ROUTER_TITLE_8		0x0000	0xFFFF	RW
1257	HR_CS_MQTT_ROUTER_TITLE_9		0x0000	0xFFFF	RW
1258	HR_CS_MQTT_ROUTER_TITLE_10		0x0000	0xFFFF	RW
1259	HR_CS_MQTT_ROUTER_TITLE_11		0x0000	0xFFFF	RW
1260	HR_CS_MQTT_ROUTER_TITLE_12		0x0000	0xFFFF	RW

ADDRESS	REGISTER	DESCRIPTION	MINIMUM VALUE	MAXIMUM VALUE	TYPE
1261	HR_CS_MQTT_ROUTER_TITLE_13		0x0000	0xFFFF	RW
1262	HR_CS_MQTT_ROUTER_TITLE_14		0x0000	0xFFFF	RW
1263	HR_CS_MQTT_ROUTER_TITLE_15		0x0000	0xFFFF	RW
1264	HR_CS_MQTT_ROUTER_TITLE_16		0x0000	0xFFFF	RW
		Reserved.			
1266	HR_CS_MQTT_BROKER_PASSWORD_1		0x0000	0xFFFF	RW
1267	HR_CS_MQTT_BROKER_PASSWORD_2		0x0000	0xFFFF	RW
1268	HR_CS_MQTT_BROKER_PASSWORD_3		0x0000	0xFFFF	RW
1269	HR_CS_MQTT_BROKER_PASSWORD_4		0x0000	0xFFFF	RW
1270	HR_CS_MQTT_BROKER_PASSWORD_5		0x0000	0xFFFF	RW
1271	HR_CS_MQTT_BROKER_PASSWORD_6		0x0000	0xFFFF	RW
1272	HR_CS_MQTT_BROKER_PASSWORD_7		0x0000	0xFFFF	RW
1273	HR_CS_MQTT_BROKER_PASSWORD_8		0x0000	0xFFFF	RW
1274	HR_CS_MQTT_BROKER_PASSWORD_9		0x0000	0xFFFF	RW
1275	HR_CS_MQTT_BROKER_PASSWORD_10		0x0000	0xFFFF	RW
1276	HR_CS_MQTT_BROKER_PASSWORD_11	Allows you to configure the Broker password. Each register is equivalent to 2 characters.	0x0000	0xFFFF	RW
1277	HR_CS_MQTT_BROKER_PASSWORD_12		0x0000	0xFFFF	RW
1278	HR_CS_MQTT_BROKER_PASSWORD_13		0x0000	0xFFFF	RW
1279	HR_CS_MQTT_BROKER_PASSWORD_14		0x0000	0xFFFF	RW
1280	HR_CS_MQTT_BROKER_PASSWORD_15		0x0000	0xFFFF	RW
1281	HR_CS_MQTT_BROKER_PASSWORD_16		0x0000	0xFFFF	RW
1282	HR_CS_MQTT_BROKER_PASSWORD_17		0x0000	0xFFFF	RW
1283	HR_CS_MQTT_BROKER_PASSWORD_18		0x0000	0xFFFF	RW
1284	HR_CS_MQTT_BROKER_PASSWORD_19		0x0000	0xFFFF	RW
1285	HR_CS_MQTT_BROKER_PASSWORD_20		0x0000	0xFFFF	RW
1286	HR_CS_MQTT_BROKER_PASSWORD_21		0x0000	0xFFFF	RW
		Reserved.			
1288	HR_CS_MQTT_BROKER_MQTT_IP_URL_1		0x0000	0xFFFF	RW
1289	HR_CS_MQTT_BROKER_MQTT_IP_URL_2		0x0000	0xFFFF	RW
1290	HR_CS_MQTT_BROKER_MQTT_IP_URL_3		0x0000	0xFFFF	RW
1291	HR_CS_MQTT_BROKER_MQTT_IP_URL_4		0x0000	0xFFFF	RW
1292	HR_CS_MQTT_BROKER_MQTT_IP_URL_5		0x0000	0xFFFF	RW
1293	HR_CS_MQTT_BROKER_MQTT_IP_URL_6		0x0000	0xFFFF	RW
1294	HR_CS_MQTT_BROKER_MQTT_IP_URL_7		0x0000	0xFFFF	RW
1295	HR_CS_MQTT_BROKER_MQTT_IP_URL_8		0x0000	0xFFFF	RW
1296	HR_CS_MQTT_BROKER_MQTT_IP_URL_9		0x0000	0xFFFF	RW
1297	HR_CS_MQTT_BROKER_MQTT_IP_URL_10		0x0000	0xFFFF	RW
1298	HR_CS_MQTT_BROKER_MQTT_IP_URL_11	Allows you to define the IP or URL of the Broker. Each register is equivalent to 2 characters.	0x0000	0xFFFF	RW
1299	HR_CS_MQTT_BROKER_MQTT_IP_URL_12		0x0000	0xFFFF	RW
1300	HR_CS_MQTT_BROKER_MQTT_IP_URL_13		0x0000	0xFFFF	RW
1301	HR_CS_MQTT_BROKER_MQTT_IP_URL_14		0x0000	0xFFFF	RW
1302	HR_CS_MQTT_BROKER_MQTT_IP_URL_15		0x0000	0xFFFF	RW
1303	HR_CS_MQTT_BROKER_MQTT_IP_URL_16		0x0000	0xFFFF	RW
1304	HR_CS_MQTT_BROKER_MQTT_IP_URL_17		0x0000	0xFFFF	RW

ADDRESS	REGISTER	DESCRIPTION	MINIMUM VALUE	MAXIMUM VALUE	TYPE
1305	HR_CS_MQTT_BROKER_MQTT_IP_URL_18		0x0000	0xFFFF	RW
1306	HR_CS_MQTT_BROKER_MQTT_IP_URL_19		0x0000	0xFFFF	RW
1307	HR_CS_MQTT_BROKER_MQTT_IP_URL_20		0x0000	0xFFFF	RW
1308	HR_CS_MQTT_BROKER_MQTT_IP_URL_21		0x0000	0xFFFF	RW
1309	HR_CS_MQTT_BROKER_MQTT_IP_URL_22		0x0000	0xFFFF	RW
1310	HR_CS_MQTT_BROKER_MQTT_IP_URL_23		0x0000	0xFFFF	RW
1311	HR_CS_MQTT_BROKER_MQTT_IP_URL_24		0x0000	0xFFFF	RW
1312	HR_CS_MQTT_BROKER_MQTT_IP_URL_25		0x0000	0xFFFF	RW
1313	HR_CS_MQTT_BROKER_MQTT_IP_URL_26		0x0000	0xFFFF	RW
1314	HR_CS_MQTT_BROKER_MQTT_IP_URL_27		0x0000	0xFFFF	RW
1315	HR_CS_MQTT_BROKER_MQTT_IP_URL_28		0x0000	0xFFFF	RW
1316	HR_CS_MQTT_BROKER_MQTT_IP_URL_29		0x0000	0xFFFF	RW
1317	HR_CS_MQTT_BROKER_MQTT_IP_URL_30		0x0000	0xFFFF	RW
		Reserved.			
1319	HR_CS_MQTT_SECURITY,	Allows you to configure the protocol and data encryption for communication with the MQTT Broker: 0 → None 1 → TLS V1.2 – Server signed 2 → TLS V1.2 – CA only 3 → TLS V1.2 – Self signed	0	3	RW
1320	HR_CS_MQTT_DEVICE_ID_1	Allows you to configure a device ID. Each register is equivalent to 2 characters.	0x0000	0xFFFF	RW
1321	HR_CS_MQTT_DEVICE_ID_2		0x0000	0xFFFF	RW
1322	HR_CS_MQTT_DEVICE_ID_3		0x0000	0xFFFF	RW
1323	HR_CS_MQTT_DEVICE_ID_4		0x0000	0xFFFF	RW
1324	HR_CS_MQTT_DEVICE_ID_5		0x0000	0xFFFF	RW
1325	HR_CS_MQTT_DEVICE_ID_6		0x0000	0xFFFF	RW
1326	HR_CS_MQTT_DEVICE_ID_7		0x0000	0xFFFF	RW
1327	HR_CS_MQTT_DEVICE_ID_8		0x0000	0xFFFF	RW
1328	HR_CS_MQTT_DEVICE_ID_9		0x0000	0xFFFF	RW
1329	HR_CS_MQTT_DEVICE_ID_10		0x0000	0xFFFF	RW
		Reserved.			
1331	HR_CS_MQTT_PRIMARY_KEY_1	Allows you to define a primary key for the device (Only for Microsoft Azure).	0x0000	0xFFFF	RW
1332	HR_CS_MQTT_PRIMARY_KEY_2		0x0000	0xFFFF	RW
1333	HR_CS_MQTT_PRIMARY_KEY_3		0x0000	0xFFFF	RW
1334	HR_CS_MQTT_PRIMARY_KEY_4		0x0000	0xFFFF	RW
1335	HR_CS_MQTT_PRIMARY_KEY_5		0x0000	0xFFFF	RW
1336	HR_CS_MQTT_PRIMARY_KEY_6		0x0000	0xFFFF	RW
1337	HR_CS_MQTT_PRIMARY_KEY_7		0x0000	0xFFFF	RW
1338	HR_CS_MQTT_PRIMARY_KEY_8		0x0000	0xFFFF	RW
1339	HR_CS_MQTT_PRIMARY_KEY_9		0x0000	0xFFFF	RW
1340	HR_CS_MQTT_PRIMARY_KEY_10		0x0000	0xFFFF	RW
1341	HR_CS_MQTT_PRIMARY_KEY_11		0x0000	0xFFFF	RW
1342	HR_CS_MQTT_PRIMARY_KEY_12		0x0000	0xFFFF	RW

ADDRESS	REGISTER	DESCRIPTION	MINIMUM VALUE	MAXIMUM VALUE	TYPE
1343	HR_CS_MQTT_PRIMARY_KEY_13		0x0000	0xFFFF	RW
1344	HR_CS_MQTT_PRIMARY_KEY_14		0x0000	0xFFFF	RW
1345	HR_CS_MQTT_PRIMARY_KEY_15		0x0000	0xFFFF	RW
1346	HR_CS_MQTT_PRIMARY_KEY_16		0x0000	0xFFFF	RW
1347	HR_CS_MQTT_PRIMARY_KEY_17		0x0000	0xFFFF	RW
1348	HR_CS_MQTT_PRIMARY_KEY_18		0x0000	0xFFFF	RW
1349	HR_CS_MQTT_PRIMARY_KEY_19		0x0000	0xFFFF	RW
1350	HR_CS_MQTT_PRIMARY_KEY_20		0x0000	0xFFFF	RW
1351	HR_CS_MQTT_PRIMARY_KEY_21		0x0000	0xFFFF	RW
1352	HR_CS_MQTT_PRIMARY_KEY_22		0x0000	0xFFFF	RW
1353	HR_CS_MQTT_PRIMARY_KEY_23		0x0000	0xFFFF	RW
1354	HR_CS_MQTT_PRIMARY_KEY_24		0x0000	0xFFFF	RW
1355	HR_CS_MQTT_PRIMARY_KEY_25		0x0000	0xFFFF	RW
1356	HR_CS_MQTT_PRIMARY_KEY_26		0x0000	0xFFFF	RW
1357	HR_CS_MQTT_PRIMARY_KEY_27		0x0000	0xFFFF	RW
1358	HR_CS_MQTT_PRIMARY_KEY_28		0x0000	0xFFFF	RW
1359	HR_CS_MQTT_PRIMARY_KEY_29		0x0000	0xFFFF	RW
1360	HR_CS_MQTT_PRIMARY_KEY_30		0x0000	0xFFFF	RW
		Reserved.			
1362	HR_CS_MQTT_PROJECT_ID_1		0x0000	0xFFFF	RW
1363	HR_CS_MQTT_PROJECT_ID_2		0x0000	0xFFFF	RW
1364	HR_CS_MQTT_PROJECT_ID_3		0x0000	0xFFFF	RW
1365	HR_CS_MQTT_PROJECT_ID_4		0x0000	0xFFFF	RW
1366	HR_CS_MQTT_PROJECT_ID_5	Allows you to configure an ID for the project (Only for Google Cloud).	0x0000	0xFFFF	RW
1367	HR_CS_MQTT_PROJECT_ID_6		0x0000	0xFFFF	RW
1368	HR_CS_MQTT_PROJECT_ID_7		0x0000	0xFFFF	RW
1369	HR_CS_MQTT_PROJECT_ID_8		0x0000	0xFFFF	RW
1370	HR_CS_MQTT_PROJECT_ID_9		0x0000	0xFFFF	RW
1371	HR_CS_MQTT_PROJECT_ID_10		0x0000	0xFFFF	RW
		Reserved.			
1373	HR_CS_MQTT_GGL_REGION_1		0x0000	0xFFFF	RW
1374	HR_CS_MQTT_GGL_REGION_2		0x0000	0xFFFF	RW
1375	HR_CS_MQTT_GGL_REGION_3		0x0000	0xFFFF	RW
1376	HR_CS_MQTT_GGL_REGION_4		0x0000	0xFFFF	RW
1377	HR_CS_MQTT_GGL_REGION_5		0x0000	0xFFFF	RW
1378	HR_CS_MQTT_GGL_REGION_6	Allows you to define a region for the connection (Only for Google Cloud).	0x0000	0xFFFF	RW
1379	HR_CS_MQTT_GGL_REGION_7		0x0000	0xFFFF	RW
1380	HR_CS_MQTT_GGL_REGION_8		0x0000	0xFFFF	RW
1381	HR_CS_MQTT_GGL_REGION_9		0x0000	0xFFFF	RW
1382	HR_CS_MQTT_GGL_REGION_10		0x0000	0xFFFF	RW
1383	HR_CS_MQTT_TOPIC_RETAIN	Allows you to enable data retention in the cloud.	0	1	RW
1384	HR_CS_MQTT_TOPIC_DATA_PUB_1		0x0000	0xFFFF	RW
1385	HR_CS_MQTT_TOPIC_DATA_PUB_2		0x0000	0xFFFF	RW
1386	HR_CS_MQTT_TOPIC_DATA_PUB_3		0x0000	0xFFFF	RW
1387	HR_CS_MQTT_TOPIC_DATA_PUB_4	Allows the device to publish data to the Channels and Events publishing topics.	0x0000	0xFFFF	RW
1388	HR_CS_MQTT_TOPIC_DATA_PUB_5		0x0000	0xFFFF	RW
1389	HR_CS_MQTT_TOPIC_DATA_PUB_6		0x0000	0xFFFF	RW
1390	HR_CS_MQTT_TOPIC_DATA_PUB_7		0x0000	0xFFFF	RW

ADDRESS	REGISTER	DESCRIPTION	MINIMUM VALUE	MAXIMUM VALUE	TYPE
1391	HR_CS_MQTT_TOPIC_DATA_PUB_8		0x0000	0xFFFF	RW
1392	HR_CS_MQTT_TOPIC_DATA_PUB_9		0x0000	0xFFFF	RW
1393	HR_CS_MQTT_TOPIC_DATA_PUB_10		0x0000	0xFFFF	RW
1394	HR_CS_MQTT_TOPIC_DATA_PUB_11		0x0000	0xFFFF	RW
1395	HR_CS_MQTT_TOPIC_DATA_PUB_12		0x0000	0xFFFF	RW
1396	HR_CS_MQTT_TOPIC_DATA_PUB_13		0x0000	0xFFFF	RW
1397	HR_CS_MQTT_TOPIC_DATA_PUB_14		0x0000	0xFFFF	RW
1398	HR_CS_MQTT_TOPIC_DATA_PUB_15		0x0000	0xFFFF	RW
1399	HR_CS_MQTT_TOPIC_DATA_PUB_16		0x0000	0xFFFF	RW
1400	HR_CS_MQTT_TOPIC_DATA_PUB_17		0x0000	0xFFFF	RW
1401	HR_CS_MQTT_TOPIC_DATA_PUB_18		0x0000	0xFFFF	RW
1402	HR_CS_MQTT_TOPIC_DATA_PUB_19		0x0000	0xFFFF	RW
1403	HR_CS_MQTT_TOPIC_DATA_PUB_20		0x0000	0xFFFF	RW
1404	HR_CS_MQTT_TOPIC_DATA_PUB_21		0x0000	0xFFFF	RW
1405	HR_CS_MQTT_TOPIC_DATA_PUB_22		0x0000	0xFFFF	RW
1406	HR_CS_MQTT_TOPIC_DATA_PUB_23		0x0000	0xFFFF	RW
1407	HR_CS_MQTT_TOPIC_DATA_PUB_24		0x0000	0xFFFF	RW
1408	HR_CS_MQTT_TOPIC_DATA_PUB_25		0x0000	0xFFFF	RW
1409	HR_CS_MQTT_TOPIC_DATA_PUB_26		0x0000	0xFFFF	RW
1410	HR_CS_MQTT_TOPIC_DATA_PUB_27		0x0000	0xFFFF	RW
1411	HR_CS_MQTT_TOPIC_DATA_PUB_28		0x0000	0xFFFF	RW
1412	HR_CS_MQTT_TOPIC_DATA_PUB_29		0x0000	0xFFFF	RW
1413	HR_CS_MQTT_TOPIC_DATA_PUB_30		0x0000	0xFFFF	RW
		Reserved.			
1415	HR_CS_MQTT_TOPIC_ACK_PUB_1		0x0000	0xFFFF	RW
1416	HR_CS_MQTT_TOPIC_ACK_PUB_2		0x0000	0xFFFF	RW
1417	HR_CS_MQTT_TOPIC_ACK_PUB_3		0x0000	0xFFFF	RW
1418	HR_CS_MQTT_TOPIC_ACK_PUB_4		0x0000	0xFFFF	RW
1419	HR_CS_MQTT_TOPIC_ACK_PUB_5		0x0000	0xFFFF	RW
1420	HR_CS_MQTT_TOPIC_ACK_PUB_6		0x0000	0xFFFF	RW
1421	HR_CS_MQTT_TOPIC_ACK_PUB_7		0x0000	0xFFFF	RW
1422	HR_CS_MQTT_TOPIC_ACK_PUB_8		0x0000	0xFFFF	RW
1423	HR_CS_MQTT_TOPIC_ACK_PUB_9		0x0000	0xFFFF	RW
1424	HR_CS_MQTT_TOPIC_ACK_PUB_10		0x0000	0xFFFF	RW
1425	HR_CS_MQTT_TOPIC_ACK_PUB_11		0x0000	0xFFFF	RW
1426	HR_CS_MQTT_TOPIC_ACK_PUB_12	Allows the device to publish data to the **Ack Config** publishing topic.	0x0000	0xFFFF	RW
1427	HR_CS_MQTT_TOPIC_ACK_PUB_13		0x0000	0xFFFF	RW
1428	HR_CS_MQTT_TOPIC_ACK_PUB_14		0x0000	0xFFFF	RW
1429	HR_CS_MQTT_TOPIC_ACK_PUB_15		0x0000	0xFFFF	RW
1430	HR_CS_MQTT_TOPIC_ACK_PUB_16		0x0000	0xFFFF	RW
1431	HR_CS_MQTT_TOPIC_ACK_PUB_17		0x0000	0xFFFF	RW
1432	HR_CS_MQTT_TOPIC_ACK_PUB_18		0x0000	0xFFFF	RW
1433	HR_CS_MQTT_TOPIC_ACK_PUB_19		0x0000	0xFFFF	RW
1434	HR_CS_MQTT_TOPIC_ACK_PUB_20		0x0000	0xFFFF	RW
1435	HR_CS_MQTT_TOPIC_ACK_PUB_21		0x0000	0xFFFF	RW
1436	HR_CS_MQTT_TOPIC_ACK_PUB_22		0x0000	0xFFFF	RW
1437	HR_CS_MQTT_TOPIC_ACK_PUB_23		0x0000	0xFFFF	RW
1438	HR_CS_MQTT_TOPIC_ACK_PUB_24		0x0000	0xFFFF	RW

ADDRESS	REGISTER	DESCRIPTION	MINIMUM VALUE	MAXIMUM VALUE	TYPE
1439	HR_CS_MQTT_TOPIC_ACK_PUB_25		0x0000	0xFFFF	RW
1440	HR_CS_MQTT_TOPIC_ACK_PUB_26		0x0000	0xFFFF	RW
1441	HR_CS_MQTT_TOPIC_ACK_PUB_27		0x0000	0xFFFF	RW
1442	HR_CS_MQTT_TOPIC_ACK_PUB_28		0x0000	0xFFFF	RW
1443	HR_CS_MQTT_TOPIC_ACK_PUB_29		0x0000	0xFFFF	RW
1444	HR_CS_MQTT_TOPIC_ACK_PUB_30		0x0000	0xFFFF	RW
		Reserved.			
1446	HR_CS_MQTT_TOPIC_CONFIG_SUB_1		0x0000	0xFFFF	RW
1447	HR_CS_MQTT_TOPIC_CONFIG_SUB_2		0x0000	0xFFFF	RW
1448	HR_CS_MQTT_TOPIC_CONFIG_SUB_3		0x0000	0xFFFF	RW
1449	HR_CS_MQTT_TOPIC_CONFIG_SUB_4		0x0000	0xFFFF	RW
1450	HR_CS_MQTT_TOPIC_CONFIG_SUB_5		0x0000	0xFFFF	RW
1451	HR_CS_MQTT_TOPIC_CONFIG_SUB_6		0x0000	0xFFFF	RW
1452	HR_CS_MQTT_TOPIC_CONFIG_SUB_7		0x0000	0xFFFF	RW
1453	HR_CS_MQTT_TOPIC_CONFIG_SUB_8		0x0000	0xFFFF	RW
1454	HR_CS_MQTT_TOPIC_CONFIG_SUB_9		0x0000	0xFFFF	RW
1455	HR_CS_MQTT_TOPIC_CONFIG_SUB_10		0x0000	0xFFFF	RW
1456	HR_CS_MQTT_TOPIC_CONFIG_SUB_11		0x0000	0xFFFF	RW
1457	HR_CS_MQTT_TOPIC_CONFIG_SUB_12		0x0000	0xFFFF	RW
1458	HR_CS_MQTT_TOPIC_CONFIG_SUB_13		0x0000	0xFFFF	RW
1459	HR_CS_MQTT_TOPIC_CONFIG_SUB_14		0x0000	0xFFFF	RW
1460	HR_CS_MQTT_TOPIC_CONFIG_SUB_15	Allows the device to receive data from the **Config** subscription topic.	0x0000	0xFFFF	RW
1461	HR_CS_MQTT_TOPIC_CONFIG_SUB_16		0x0000	0xFFFF	RW
1462	HR_CS_MQTT_TOPIC_CONFIG_SUB_17		0x0000	0xFFFF	RW
1463	HR_CS_MQTT_TOPIC_CONFIG_SUB_18		0x0000	0xFFFF	RW
1464	HR_CS_MQTT_TOPIC_CONFIG_SUB_19		0x0000	0xFFFF	RW
1465	HR_CS_MQTT_TOPIC_CONFIG_SUB_20		0x0000	0xFFFF	RW
1466	HR_CS_MQTT_TOPIC_CONFIG_SUB_21		0x0000	0xFFFF	RW
1467	HR_CS_MQTT_TOPIC_CONFIG_SUB_22		0x0000	0xFFFF	RW
1468	HR_CS_MQTT_TOPIC_CONFIG_SUB_23		0x0000	0xFFFF	RW
1469	HR_CS_MQTT_TOPIC_CONFIG_SUB_24		0x0000	0xFFFF	RW
1470	HR_CS_MQTT_TOPIC_CONFIG_SUB_25		0x0000	0xFFFF	RW
1471	HR_CS_MQTT_TOPIC_CONFIG_SUB_26		0x0000	0xFFFF	RW
1472	HR_CS_MQTT_TOPIC_CONFIG_SUB_27		0x0000	0xFFFF	RW
1473	HR_CS_MQTT_TOPIC_CONFIG_SUB_28		0x0000	0xFFFF	RW
1474	HR_CS_MQTT_TOPIC_CONFIG_SUB_29		0x0000	0xFFFF	RW
1475	HR_CS_MQTT_TOPIC_CONFIG_SUB_30		0x0000	0xFFFF	RW
		Reserved.			
1477	HR_CS_MQTT_TOPIC_CMD_SUB_1		0x0000	0xFFFF	RW
1478	HR_CS_MQTT_TOPIC_CMD_SUB_2		0x0000	0xFFFF	RW
1479	HR_CS_MQTT_TOPIC_CMD_SUB_3		0x0000	0xFFFF	RW
1480	HR_CS_MQTT_TOPIC_CMD_SUB_4		0x0000	0xFFFF	RW
1481	HR_CS_MQTT_TOPIC_CMD_SUB_5	Allows the device to receive data from the **Command** subscription topic.	0x0000	0xFFFF	RW
1482	HR_CS_MQTT_TOPIC_CMD_SUB_6		0x0000	0xFFFF	RW
1483	HR_CS_MQTT_TOPIC_CMD_SUB_7		0x0000	0xFFFF	RW
1484	HR_CS_MQTT_TOPIC_CMD_SUB_8		0x0000	0xFFFF	RW
1485	HR_CS_MQTT_TOPIC_CMD_SUB_9		0x0000	0xFFFF	RW
1486	HR_CS_MQTT_TOPIC_CMD_SUB_10		0x0000	0xFFFF	RW

ADDRESS	REGISTER	DESCRIPTION	MINIMUM VALUE	MAXIMUM VALUE	TYPE
1487	HR_CS_MQTT_TOPIC_CMD_SUB_11		0x0000	0xFFFF	RW
1488	HR_CS_MQTT_TOPIC_CMD_SUB_12		0x0000	0xFFFF	RW
1489	HR_CS_MQTT_TOPIC_CMD_SUB_13		0x0000	0xFFFF	RW
1490	HR_CS_MQTT_TOPIC_CMD_SUB_14		0x0000	0xFFFF	RW
1491	HR_CS_MQTT_TOPIC_CMD_SUB_15		0x0000	0xFFFF	RW
1492	HR_CS_MQTT_TOPIC_CMD_SUB_16		0x0000	0xFFFF	RW
1493	HR_CS_MQTT_TOPIC_CMD_SUB_17		0x0000	0xFFFF	RW
1494	HR_CS_MQTT_TOPIC_CMD_SUB_18		0x0000	0xFFFF	RW
1495	HR_CS_MQTT_TOPIC_CMD_SUB_19		0x0000	0xFFFF	RW
1496	HR_CS_MQTT_TOPIC_CMD_SUB_20		0x0000	0xFFFF	RW
1497	HR_CS_MQTT_TOPIC_CMD_SUB_21		0x0000	0xFFFF	RW
1498	HR_CS_MQTT_TOPIC_CMD_SUB_22		0x0000	0xFFFF	RW
1499	HR_CS_MQTT_TOPIC_CMD_SUB_23		0x0000	0xFFFF	RW
1500	HR_CS_MQTT_TOPIC_CMD_SUB_24		0x0000	0xFFFF	RW
1501	HR_CS_MQTT_TOPIC_CMD_SUB_25		0x0000	0xFFFF	RW
1502	HR_CS_MQTT_TOPIC_CMD_SUB_26		0x0000	0xFFFF	RW
1503	HR_CS_MQTT_TOPIC_CMD_SUB_27		0x0000	0xFFFF	RW
1504	HR_CS_MQTT_TOPIC_CMD_SUB_28		0x0000	0xFFFF	RW
1505	HR_CS_MQTT_TOPIC_CMD_SUB_29		0x0000	0xFFFF	RW
1506	HR_CS_MQTT_TOPIC_CMD_SUB_30		0x0000	0xFFFF	RW
		Reserved.			
1508	HR_CS_MQTT_TOPIC_CMD_ACK_PUB_1		0x0000	0xFFFF	RW
1509	HR_CS_MQTT_TOPIC_CMD_ACK_PUB_2		0x0000	0xFFFF	RW
1510	HR_CS_MQTT_TOPIC_CMD_ACK_PUB_3		0x0000	0xFFFF	RW
1511	HR_CS_MQTT_TOPIC_CMD_ACK_PUB_4		0x0000	0xFFFF	RW
1512	HR_CS_MQTT_TOPIC_CMD_ACK_PUB_5		0x0000	0xFFFF	RW
1513	HR_CS_MQTT_TOPIC_CMD_ACK_PUB_6		0x0000	0xFFFF	RW
1514	HR_CS_MQTT_TOPIC_CMD_ACK_PUB_7		0x0000	0xFFFF	RW
1515	HR_CS_MQTT_TOPIC_CMD_ACK_PUB_8		0x0000	0xFFFF	RW
1516	HR_CS_MQTT_TOPIC_CMD_ACK_PUB_9		0x0000	0xFFFF	RW
1517	HR_CS_MQTT_TOPIC_CMD_ACK_PUB_10		0x0000	0xFFFF	RW
1518	HR_CS_MQTT_TOPIC_CMD_ACK_PUB_11		0x0000	0xFFFF	RW
1519	HR_CS_MQTT_TOPIC_CMD_ACK_PUB_12		0x0000	0xFFFF	RW
1520	HR_CS_MQTT_TOPIC_CMD_ACK_PUB_13		0x0000	0xFFFF	RW
1521	HR_CS_MQTT_TOPIC_CMD_ACK_PUB_14	Allows the device to publish data to the **Ack Command** publishing topic.	0x0000	0xFFFF	RW
1522	HR_CS_MQTT_TOPIC_CMD_ACK_PUB_15		0x0000	0xFFFF	RW
1523	HR_CS_MQTT_TOPIC_CMD_ACK_PUB_16		0x0000	0xFFFF	RW
1524	HR_CS_MQTT_TOPIC_CMD_ACK_PUB_17		0x0000	0xFFFF	RW
1525	HR_CS_MQTT_TOPIC_CMD_ACK_PUB_18		0x0000	0xFFFF	RW
1526	HR_CS_MQTT_TOPIC_CMD_ACK_PUB_19		0x0000	0xFFFF	RW
1527	HR_CS_MQTT_TOPIC_CMD_ACK_PUB_20		0x0000	0xFFFF	RW
1528	HR_CS_MQTT_TOPIC_CMD_ACK_PUB_21		0x0000	0xFFFF	RW
1529	HR_CS_MQTT_TOPIC_CMD_ACK_PUB_22		0x0000	0xFFFF	RW
1530	HR_CS_MQTT_TOPIC_CMD_ACK_PUB_23		0x0000	0xFFFF	RW
1531	HR_CS_MQTT_TOPIC_CMD_ACK_PUB_24		0x0000	0xFFFF	RW
1532	HR_CS_MQTT_TOPIC_CMD_ACK_PUB_25		0x0000	0xFFFF	RW
1533	HR_CS_MQTT_TOPIC_CMD_ACK_PUB_26		0x0000	0xFFFF	RW
1534	HR_CS_MQTT_TOPIC_CMD_ACK_PUB_27		0x0000	0xFFFF	RW

ADDRESS	REGISTER	DESCRIPTION	MINIMUM VALUE	MAXIMUM VALUE	TYPE
1535	HR_CS_MQTT_TOPIC_CMD_ACK_PUB_28		0x0000	0xFFFF	RW
1536	HR_CS_MQTT_TOPIC_CMD_ACK_PUB_29		0x0000	0xFFFF	RW
1537	HR_CS_MQTT_TOPIC_CMD_ACK_PUB_30		0x0000	0xFFFF	RW
		Reserved.			
1539	HR_CS_MQTT_REGISTER_ID_1	Allows you to configure an ID for the register (Only for Google Cloud).	0x0000	0xFFFF	RW
1540	HR_CS_MQTT_REGISTER_ID_2		0x0000	0xFFFF	RW
1541	HR_CS_MQTT_REGISTER_ID_3		0x0000	0xFFFF	RW
1542	HR_CS_MQTT_REGISTER_ID_4		0x0000	0xFFFF	RW
1543	HR_CS_MQTT_REGISTER_ID_5		0x0000	0xFFFF	RW
1544	HR_CS_MQTT_REGISTER_ID_6		0x0000	0xFFFF	RW
1545	HR_CS_MQTT_REGISTER_ID_7		0x0000	0xFFFF	RW
1546	HR_CS_MQTT_REGISTER_ID_8		0x0000	0xFFFF	RW
1547	HR_CS_MQTT_REGISTER_ID_9		0x0000	0xFFFF	RW
1548	HR_CS_MQTT_REGISTER_ID_10		0x0000	0xFFFF	RW
1549	HR_CS_MQTT_SEND_DIAGCOUNTERS	Adds system counters in MQTT diagnostics and enables periodic diagnostic publishing in the command confirmation topic.	0	1	RW
1550	HR_CS_MQTT_CLOUD_SEL	Allows you to configure the cloud to be used by the MQTT protocol: 0 → Generic cloud 1 → Google Cloud 2 → Amazon AWS 3 → Microsoft Azure 4 → NOVUS Cloud 5 → LiveMES.	0	5	RW
1551	HR_CS_MQTT_TOKEN_EXP	Allows you to configure the token expiration time (in minutes). Valid for connections to Google Cloud and Amazon AWS clouds.	0	65535	RW
		Reserved.			
1553	HR_CS_SNTP_SERVER_ENABLE	Allows you to enable the NTP server.	0	1	RW
1554	HR_CS_SNTP_SERVER_MIN_DIFF	Allows you to configure the minimum difference between the DigiRail OEE clock and the information received via the NTP server to update (in seconds).	1	0xFFFF	RW
1555	HR_CS_SNTP_SERVER_IP_URL_1	Allows you to define the IP or URL of the NTP server.	0x0000	0xFFFF	RW
1556	HR_CS_SNTP_SERVER_IP_URL_2		0x0000	0xFFFF	RW
1557	HR_CS_SNTP_SERVER_IP_URL_3		0x0000	0xFFFF	RW
1558	HR_CS_SNTP_SERVER_IP_URL_4		0x0000	0xFFFF	RW
1559	HR_CS_SNTP_SERVER_IP_URL_5		0x0000	0xFFFF	RW
1560	HR_CS_SNTP_SERVER_IP_URL_6		0x0000	0xFFFF	RW
1561	HR_CS_SNTP_SERVER_IP_URL_7		0x0000	0xFFFF	RW
1562	HR_CS_SNTP_SERVER_IP_URL_8		0x0000	0xFFFF	RW
1563	HR_CS_SNTP_SERVER_IP_URL_9		0x0000	0xFFFF	RW
1564	HR_CS_SNTP_SERVER_IP_URL_10		0x0000	0xFFFF	RW
1565	HR_CS_SNTP_SERVER_IP_URL_11		0x0000	0xFFFF	RW
1566	HR_CS_SNTP_SERVER_IP_URL_12		0x0000	0xFFFF	RW
1567	HR_CS_SNTP_SERVER_IP_URL_13		0x0000	0xFFFF	RW
1568	HR_CS_SNTP_SERVER_IP_URL_14		0x0000	0xFFFF	RW
1569	HR_CS_SNTP_SERVER_IP_URL_15		0x0000	0xFFFF	RW
1570	HR_CS_SNTP_SERVER_IP_URL_16		0x0000	0xFFFF	RW
1571	HR_CS_SNTP_SERVER_IP_URL_17		0x0000	0xFFFF	RW
1572	HR_CS_SNTP_SERVER_IP_URL_18		0x0000	0xFFFF	RW
1573	HR_CS_SNTP_SERVER_IP_URL_19		0x0000	0xFFFF	RW
1574	HR_CS_SNTP_SERVER_IP_URL_20		0x0000	0xFFFF	RW
1575	HR_CS_SNTP_SERVER_IP_URL_21		0x0000	0xFFFF	RW
1576	HR_CS_SNTP_SERVER_IP_URL_22		0x0000	0xFFFF	RW

ADDRESS	REGISTER	DESCRIPTION	MINIMUM VALUE	MAXIMUM VALUE	TYPE
1577	HR_CS_SNTP_SERVER_IP_URL_23		0x0000	0xFFFF	RW
1578	HR_CS_SNTP_SERVER_IP_URL_24		0x0000	0xFFFF	RW
1579	HR_CS_SNTP_SERVER_IP_URL_25		0x0000	0xFFFF	RW
1580	HR_CS_SNTP_SERVER_IP_URL_26		0x0000	0xFFFF	RW
1581	HR_CS_SNTP_SERVER_IP_URL_27		0x0000	0xFFFF	RW
1582	HR_CS_SNTP_SERVER_IP_URL_28		0x0000	0xFFFF	RW
1583	HR_CS_SNTP_SERVER_IP_URL_29		0x0000	0xFFFF	RW
1584	HR_CS_SNTP_SERVER_IP_URL_30		0x0000	0xFFFF	RW

Table 2 – Configuration registers table

3 ACCESS TO CIRCULAR MEMORY

DigiRail OEE has a circular memory to log events and periodic downloads. In periodic downloads, the data from all enabled channels will be saved. In Event mode, only the data from the event that originated the log will be saved. Through the configuration, it is possible to add the count value even when the channel is set as Event.

Three Modbus registers indicate the download positions and allow you to read the circular memory: HR_SS_COLLECT_RECORD_MAX_QTTY, HR_SS_COLLECT_LAST_RECORD, and HR_SS_COLLECT_FIRST_RECORD. The application must rely on these three registers to calculate the position to be requested (see function *u16GetNextIndex* from the example code below).

After calculating the position, the application must write the value to the Modbus register HR_SS_COLLECT_REQUESTED_RECORD. After writing the value, the application can read the registers, which have already been updated with the requested download values. To monitor and download the registers from memory, see the *TaskReadMem* function from the example code below.

The first valid read position is at address 1.

If you request a position with no data to download yet, all registers with values intended for download will be filled with the value 0xFFFF.

3.1 CIRCULAR MEMORY RELATED REGISTERS TABLE

	ADDRESS	NAME	DESCRIPTION	PERMISSION
DOWNLOAD POSITION CALCULATION	187	HR_SS_COLLECT_RECORD_MAX_QTTY	Maximum number of downloads supported by memory.	RO
	188	HR_SS_COLLECT_LAST_RECORD	Position of the last download added to memory.	RO
	189	HR_SS_COLLECT_FIRST_RECORD	Position of the first download added to memory.	RO
REQUESTS DOWNLOAD	190	HR_SS_COLLECT_REQUESTED_RECORD	Position of the download requested for reading.	RW
REQUESTED REGISTRATION DATA	191	HR_SS_COLLECT_TIMESTAMP_UNIX_H	Timestamp of the requested download in Unix format.	RO
	192	HR_SS_COLLECT_TIMESTAMP_UNIX_L		RO
	193	HR_SS_COLLECT_TIMESTAMP_MS	Timestamp of the requested download in milliseconds.	RO
	194	HR_SS_COLLECT_CHD_EVENT_INDEX	Index of the digital channel when an event occurs.	RO
	195	HR_SS_COLLECT_CHD_EVENT_TYPE	Event type (when it occurs).	RO
	196	HR_SS_COLLECT_CHD1_VALUE_H	Value of digital channel 1 in the requested download.	RO
	197	HR_SS_COLLECT_CHD1_VALUE_L		RO
	198	HR_SS_COLLECT_CHD2_VALUE_H	Value of digital channel 2 in the requested download.	RO
	199	HR_SS_COLLECT_CHD2_VALUE_L		RO
	200	HR_SS_COLLECT_CHD3_VALUE_H	Value of digital channel 3 in the requested download.	RO
	201	HR_SS_COLLECT_CHD3_VALUE_L		RO
	202	HR_SS_COLLECT_CHD4_VALUE_H	Value of digital channel 4 in the requested download.	RO
	203	HR_SS_COLLECT_CHD4_VALUE_L		RO
	204	HR_SS_COLLECT_CHD5_VALUE_H	Value of digital channel 5 in the requested download.	RO
	205	HR_SS_COLLECT_CHD5_VALUE_L		RO
	206	HR_SS_COLLECT_CHD6_VALUE_H	Value of digital channel 6 in the requested download.	RO
	207	HR_SS_COLLECT_CHD6_VALUE_L		RO
	208	HR_SS_COLLECT_CH1_SENSE_USER_RANGE_H	Displays the sensor value in the user range of analog channel 1 (in Float).	RO
	209	HR_SS_COLLECT_CH1_SENSE_USER_RANGE_L		RO
	210	HR_SS_COLLECT_CH2_SENSE_USER_RANGE_H	Displays the sensor value in the user range of analog channel 2 (in Float).	RO
	211	HR_SS_COLLECT_CH2_SENSE_USER_RANGE_L		RO

Table 3 – Registers table

3.2 CIRCULAR MEMORY AVAILABILITY TABLE

This table allows you to evaluate the maximum number of downloads performed by the enabled channels if the digital channel count is in event mode or if it is not in event mode:

DIGITAL CHANNELS	ANALOG CHANNELS	MAXIMUM AMOUNT (WITHOUT EVENT COUNTING)	MAXIMUM AMOUNT (WITH EVENT COUNTING)
0	1	7096	4913
0	2	5806	4913
1	0	5806	4913
2	0	4258	4258
3	0	3361	3361
4	0	2777	2777

5	0	2365	2365
6	0	2060	2060
6	1	1935	1935
6	2	1824	1824

Table 4 – Circular memory availability

3.3 EXAMPLE CODE

```
typedef enum e_oee_collect_memmap
{
   ADDR_MAX_RECORDS_QTTY,
   ADDR_LAST_RECORD,
   ADDR_FIRST_RECORD,
   ADDR_REQUESTED_RECORD,
   ADDR_TIMESTAMP_UNIX_HIGH,
   ADDR_TIMESTAMP_UNIX_LOW,
   ADDR_TIMESTAMP_MS,
   ADDR_DIGITAL_CHANNEL_EVENT_INDEX,
   ADDR_EVENT_TYPE,
   ADDR_DIGITAL_CHANNEL_1_HIGH,
   ADDR_DIGITAL_CHANNEL_1_LOW,
   ADDR_DIGITAL_CHANNEL_2_HIGH,
   ADDR_DIGITAL_CHANNEL_2_LOW,
   ADDR_DIGITAL_CHANNEL_3_HIGH,
   ADDR_DIGITAL_CHANNEL_3_LOW,
   ADDR_DIGITAL_CHANNEL_4_HIGH,
   ADDR_DIGITAL_CHANNEL_4_LOW,
   ADDR_DIGITAL_CHANNEL_5_HIGH,
   ADDR_DIGITAL_CHANNEL_5_LOW,
   ADDR_DIGITAL_CHANNEL_6_HIGH,
   ADDR_DIGITAL_CHANNEL_6_LOW,
   ADDR_ANALOG_CHANNEL_1_HIGH,
   ADDR_ANALOG_CHANNEL_1_LOW,
   ADDR_ANALOG_CHANNEL_2_HIGH,
   ADDR_ANALOG_CHANNEL_2_LOW
} oee_collect_memmap_t;

typedef enum e_digital_channels
{
   DIGITAL_CHANNEL_1,
   DIGITAL_CHANNEL_2,
   DIGITAL_CHANNEL_3,
   DIGITAL_CHANNEL_4,
   DIGITAL_CHANNEL_5,
   DIGITAL_CHANNEL_6,
   DIGITAL_CHANNELS_TOTAL
} digital_channels_t

typedef enum e_analog_channels
{
   ANALOG_CHANNEL_1,
   ANALOG_CHANNEL_2,
   ANALOG_CHANNELS_TOTAL
} analog_channels_t

typedef enum e_channel_digital_event_index
{
    DIGITAL_CHANNEL_EVENT_INDEX_NONE, // Periodic log, no event associated to digital channel
    DIGITAL_CHANNEL_EVENT_INDEX_CH1, // Event - channel 1
    DIGITAL_CHANNEL_EVENT_INDEX_CH2, // Event - channel 2
    DIGITAL_CHANNEL_EVENT_INDEX_CH3, // Event - channel 3
   DIGITAL_CHANNEL_EVENT_INDEX_CH4, // Event - channel 4
   DIGITAL_CHANNEL_EVENT_INDEX_CH5, // Event - channel 5
   DIGITAL_CHANNEL_EVENT_INDEX_CH6, // Event - channel 6
} channel_digital_event_index_t;

typedef enum e_event_type
{
   EVENT_TYPE_NONE,
   EVENT_TYPE_FALLING_EDGE,
   EVENT_TYPE_RISING_EDGE,
} event_type_t;

#define COLLECTED_DATA_SIZE 21

/*****************************************************************//**
 * @brief        Gets the next record index to be requested based on the last record already collected
 * @param[in]    actualIndex    Record index from register already collected
 * @return       uint16_t    Next record index to be collected
 ********************************************************************/
uint16_t u16GetNextIndex(uint16_t actualIndex)
{
   uint16_t lastRecord = FncReadSingleRegisterModbus(ADDR_LAST_RECORD);
   uint16_t firstRecord = FncReadSingleRegisterModbus(ADDR_FIRST_RECORD_REGISTER);
   uint16_t recordsQtty = FncReadSingleRegisterModbus(ADDR_MAX_RECORDS_QTTY);

   // when the index of collected record is different from the index of last record in memory
```

```
    if (actualIndex != lastRecord)
    {
        // no record has been overwritten
        if (lastRecord > firstRecord)
        {
            // collected record index is less than the index of last record in memory
            if (actualIndex < lastRecord)
            {
                return actualIndex + 1;
            }
        }
        // records circulated the memory
        else if (lastRecord < firstRecord)
        {
            // collected record index is less than the index of last record in memory
            if (actualIndex < lastRecord)
            {
                return actualIndex + 1;
            }
            // collected record index is higher than the most recent record and LESS than memory capacity
            else if (actualIndex < recordsQtty)
            {
                return actualIndex + 1;
            }
            // collected record index is higher than the most recent record and HIGHER than memory capacity
            else
            {
                return 1; // first record address
            }
        }
    }

    return actualIndex;
}

/*****************************************************************//**
 * @brief         Thread to monitor new records and collect when needed
 * @param[in]     None
 * @return        None
 *****************************************************************/
void TaskReadMem (void)
{
    uint16_t actualIndex = 0, nextIndex, lastRecord, buf[COLLECTED_DATA_SIZE];

    while (1)
    {
        // reads the index of the last record in memory
        lastRecord = FncReadSingleRegisterModbus(ADDR_LAST_RECORD);

        // if the index of collected record is different from the index of last record in memory
        if (lastRecord != actualIndex)
        {
            nextIndex = u16GetNextIndex(actualIndex);

            // requests a record by writing the index through a modbus register
            FncWriteSingleRegisterModbus(ADDR_REQUESTED_RECORD, nextIndex);

            // collects record data from requested index
            FncReadBufferModbus(buf, ADDR_TIMESTAMP_UNIX_HIGH, COLLECTED_DATA_SIZE);

            // after app uses the record, should update the index
            if (FncUseColletedData(buf) == FNC_SUCCESS)
            {
                actualIndex = nextIndex;
            }
        }
        threadSleep(100);
    }
}

/*
 * Functions that require user implementation
 *
 * FncReadSingleRegisterModbus (uint16_t registerAddr)
 * FncReadBufferModbus (uint16_t* buffer, uint16_t registerInitAddr, uint16_t size)
 * FncWriteSingleRegisterModbus  (uint16_t registerAddr, uint16_t value)
 * FncUseColletedData (uint16_t* collectedData)
 *
 */
```

3.4 EXAMPLES OF CIRCULAR MEMORY USAGE

EXAMPLE 1

In the example below, there are not yet enough logs to circulate in the device memory:

Position	Memory
1	Log 1
2	Log 2
3	Log 3
4	Log 4
5	Log 5
6	Log 6
7	Log 7
8	
9	
10	
	Position
First log	1
Last log	7

Figure 1 - Example 1

EXAMPLE 2

In the example below, the new logs have already circulated in the device memory:

Position	Memory
1	Log 11
2	Log 12
3	Log 13
4	Log 4
5	Log 5
6	Log 6
7	Log 7
8	Log 8
9	Log 9
10	Log 10
	Position
First Log	4
Last Log	3

Figure 2 - Example 2

EXAMPLE 3

In the example below, the memory already circulated has advanced:

Position	Memory
1	Log 11
2	Log 12
3	Log 13
4	Log 14
5	Log 15
6	Log 6
7	Log 7
8	Log 8
9	Log 9
10	Log 10
	Position
First Log	6
Last Log	5

Figure 3 - Example 3

3.5 DOWNLOAD EXAMPLE

EXAMPLE 4

Rise event on digital channel 2:

Register	Value
TIMESTAMP_UNIX_HIGH	0x607F
TIMESTAMP_UNIX_LOW	0x540A
TIMESTAMP_MS	300
DIGITAL_CHANNEL_EVENT_INDEX	2
EVENT_TYPE	1
DIGITAL_CHANNEL_1_ACC_HIGH	0xFFFF
DIGITAL_CHANNEL_1_ACC_LOW	0xFFFF
DIGITAL_CHANNEL_2_ACC_HIGH	0x001A
DIGITAL_CHANNEL_2_ACC_LOW	0x5648
DIGITAL_CHANNEL_3_ACC_HIGH	0xFFFF
DIGITAL_CHANNEL_3_ACC_LOW	0xFFFF
DIGITAL_CHANNEL_4_ACC_HIGH	0xFFFF
DIGITAL_CHANNEL_4_ACC_LOW	0xFFFF
DIGITAL_CHANNEL_5_ACC_HIGH	0xFFFF
DIGITAL_CHANNEL_5_ACC_LOW	0xFFFF
DIGITAL_CHANNEL_6_ACC_HIGH	0xFFFF
DIGITAL_CHANNEL_6_ACC_LOW	0xFFFF
ANALOG_CHANNEL_1_USER_RANGE_HIGH	0xFFFF
ANALOG_CHANNEL_1_USER_RANGE_LOW	0xFFFF
ANALOG_CHANNEL_2_USER_RANGE_HIGH	0xFFFF
ANALOG_CHANNEL_2_USER_RANGE_LOW	0xFFFF

Figure 4 - Example 4

EXAMPLE 5

Periodic log with digital channels 3 and 6 disabled, as well as analog channel 1:

Register	Value
TIMESTAMP_UNIX_HIGH	0x607F
TIMESTAMP_UNIX_LOW	0x5511
TIMESTAMP_MS	889
DIGITAL_CHANNEL_EVENT_INDEX	0
EVENT_TYPE	0
DIGITAL_CHANNEL_1_ACC_HIGH	0x0000
DIGITAL_CHANNEL_1_ACC_LOW	0x0001
DIGITAL_CHANNEL_2_ACC_HIGH	0x001A
DIGITAL_CHANNEL_2_ACC_LOW	0x5648
DIGITAL_CHANNEL_3_ACC_HIGH	0xFFFF
DIGITAL_CHANNEL_3_ACC_LOW	0xFFFF
DIGITAL_CHANNEL_4_ACC_HIGH	0x0000
DIGITAL_CHANNEL_4_ACC_LOW	0x0841
DIGITAL_CHANNEL_5_ACC_HIGH	0x7800
DIGITAL_CHANNEL_5_ACC_LOW	0x1566
DIGITAL_CHANNEL_6_ACC_HIGH	0xFFFF
DIGITAL_CHANNEL_6_ACC_LOW	0xFFFF
ANALOG_CHANNEL_1_USER_RANGE_HIGH	0xFFFF
ANALOG_CHANNEL_1_USER_RANGE_LOW	0xFFFF
ANALOG_CHANNEL_2_USER_RANGE_HIGH	0xC1BC
ANALOG_CHANNEL_2_USER_RANGE_LOW	0x0000

Figure 5 - Example 5