RD6006 Protocol USB-serial: reverse engineering

This is from a reverse engineering of data protocol between the Riden Window 10 SW and a RD6006-W.
This is not complete: it is a work in progress, open to all contributions.
Riden assured me that he would like to make the protocol public in the near future... For now you can use this.

Used tools:

- ***Termite***, HEX terminal COM, free https://www.compuphase.com/products_en.htm
- ***Serialmon***, COM sniffer test-mode https://www.dunovo.com/
- ***Online CRC Calculator*** https://crccalc.com/
- ***node-red***

MODBUS Protocol

Set bits: 8N1
Set baudrate: 115200
DTR/DSR

Frame description :

Slave Address	Function Code	Data	CRC
1 byte	1 byte	0 up to 252 byte(s)	2 bytes (CRC Low, CRC Hi)

Figure 12: RTU Message Frame

Slave Address: 1..247 (0: broadcast)
Function code: see later
Data: 0..252 byte(s)
CRC16-MODBUS: see https://crccalc.com/ for code.

The **master** (the WIN SW) sends a request, the **slave** (RD6006) replies

---

Function descriptions

0x03: read registers (WORD16)

Master: (read DATA0 values)

ADDR	FUNC	DATA start Addr HI	DATA start Addr LO	DATA Word# HI	DATA Word# LO	CRC HI	CRC LO
0x01	0x03	0x00	0x50	0x00	0x04	0x44	0x18

Slave: (get DATA0 values: V-SET I-SET S-OVP S-OCP)

ADDR	FUNC	DATA byte count	DATA byte [50] HI	DATA byte [50] LO	DATA byte [51] HI	DATA byte [51] LO	DATA byte [52] HI	DATA byte [52] LO	DATA byte [53] HI	DATA byte [53] LO	CRC HI	CRC LO
0x01	0x03	0x08	0x01	0xF4	0x0B	0xC2	0x02	0xBB	0x0F	0x96	0x6D	0x7D

0x06: Set single register (WORD16)

Master: (set OUTPUT ON)

ADDR	FUNC	DATA start Addr HI	DATA start Addr LO	DATA Word HI	DATA Word LO	CRC HI	CRC LO
0x01	0x06	0x00	0x08	0x00	0x01	0xC9	0xC8

Slave: echo

ADDR	FUNC	DATA start Addr HI	DATA start Addr LO	DATA Word HI	DATA Word LO	CRC HI	CRC LO
0x01	0x06	0x00	0x08	0x00	0x01	0xC9	0xC8

0x10 Set multiple registers (WORD16)

Master: (set DATA0 values: V-SET I-SET S-OVP S-OCP)

ADDR	FUNC	DATA start Addr HI	DATA start Addr LO	DATA Word count # HI	DATA Word count # LO	DATA bytes count	DATA byte [50] HI	DATA byte [50] LO	DATA byte [51] HI	DATA byte [51] LO	DATA byte [52] HI	DATA byte [52] LO	DATA byte [53] HI	DATA byte [53] LO	CRC HI	CRC LO
0x01	0x10	0x00	0x50	0x00	0x04	0x08	0x00	0xF4	0x0B	0xC2	0x02	0xBB	0x0F	0x96	0x55	0xAA

Slave: ok

ADDR	FUNC	DATA start Addr HI	DATA start Addr LO	DATA Word count# HI	DATA Word count# LO	CRC HI	CRC LO
0x01	0x10	0x00	0x50	0x00	0x04	0xDB	0x50

MODBUS also defines other functions, but they do not seem to be used by RD6006

NOTE on RD6006 Protocol

At startup the WIN program (Master):

1) sends: “queryd” + 0x0D + 0x0A (no reply)

2) reads 0000 – 0003 registers

3) sets 000F (LOCK) register to 1

4) Reads 0048 (backliught) register

….. more user operations

…. polling loop using: 0x01 0x03 0x00 0x04 0x00 0x26 0x85 0xD1

to get registers 0x0004 …0x0029

-2) sets 0012 (OUTPUT) register to OFF

-1) sets 000F (LOCK) register to 0

RD6006 registers

This is the current list (incomplete) of registers I found.

0000	0xEA 0x9E	*RO*	Signature = 60062
0001	0	*RO*	
0002	0x19 0x40	*RO*	Serial number (6464)
0003	0x00 0x80	*RO*	Firmware version (1.28) x 100
0004	0		
0005	TEMP SYS C	*RO*	
0006	0		
0007	TEMP SYS F	*RO*	
0008	V-SET	*R/W*	V value x 100
0009	I-SET	*R/W*	I value x 1000
000A	V-OUT	*RO*	V value x 100
000B	I-OUT	*RO*	I value x 1000
000C	0		
000D	WATT	*RO*	W value x 100
000E	V-INPUT	*RO*	V value x 100
000F	LOCK	*R/W*	0 = OPEN, 1 = LOCKED
0010	ERROR	*RO*	0 = OK, 1 = OVP, 2 = OCP
0011	0		
0012	OUTPUT ON/OFF	*R/W*	0 = OFF, 1 = ON
0013	DATA USE	*R/W*	memory ID 0..9
0014	0		
0015	0		
0016	0		
0017	0		
0018	0		
0019	0		
001A	0		
001B	0		
001C	0		
001D	0		
001E	0		
001F	0		
0020	BATTERY MODE	*RO*	0 = OFF, 1 = ON
0021	V-BATT	*RO*	V value x 100
0022	0		
0023	TEMP PROBE C	*RO*	
0024	0		
0025	TEMP PROBE F	*RO*	
0026	AMPEREH HI ?	*RO*	Ah value x 1000
0027	AMPEREH LO	*RO*	
0028	WATTH HI ?	*RO*	Wh value x 1000
0029	WATTH LO	*RO*	
002A	0		
002B	0		
002C	0		
002D	0		

002E	0		
002F	0		
0030	CLOCK YYYY	*R/W*	
0031	CLOCK M	*R/W*	
0032	CLOCK D	*R/W*	
0033	CLOCK h	*R/W*	
0034	CLOCK m	*R/W*	
0035	CLOCK s	*R/W*	
0036	0		
0037	OUTPUT V ZERO	*R/W*	Default = 21
0038	OUTPUT V SCALE	*R/W*	Default = 22872
0039	BACK V ZERO	*R/W*	Default = 21
003A	BACK V SCALE	*R/W*	Default = 17525
003B	OUTPUT I ZERO	*R/W*	Default = 210
003C	OUTPUT I SCALE	*R/W*	Default = 21451
003D	BACK I ZERO	*R/W*	Default = 76
003E	BACK I SCALE	*R/W*	Default = 17388
003F	0		
0040	0		
0041	0		
0042	TAKE OK OPZ	*R/W*	0 = OFF, 1 = ON
0043	TAKE OUT OPZ	*R/W*	0 = OFF, 1 = ON
0044	BOOT POWER OPZ	*R/W*	0 = OFF, 1 = ON
0045	BUZZER OPZ	*R/W*	0 = OFF, 1 = ON
0046	LOGO OPZ	*R/W*	0 = OFF, 1 = ON
0047	LANGUAGE	*R/W*	0 = Eng, 1 = Chinese, 2 = German, 3 = French
0048	BACKLIGHT	*R/W*	Values: 0..5
0049	0		
004A	0		
004B	0		
004C	0		
004D	0		
004E	0		
004F	0		
0050	DATA0 V-SET	*R/W*	
0051	DATA0 I-SET	*R/W*	
0052	DATA0 S-VOP	*R/W*	
0053	DATA0 S-OCP	*R/W*	
0054	DATA1 V-SET	*R/W*	
0055	DATA1 I-SET	*R/W*	
0056	DATA1 S-VOP	*R/W*	
0057	DATA1 S-OCP	*R/W*	
0058	DATA2 V-SET	*R/W*	
0059	DATA2 I-SET	*R/W*	
005A	DATA2 S-VOP	*R/W*	
005B	DATA2 S-OCP	*R/W*	
005C	DATA3 V-SET	*R/W*	
005D	DATA3 I-SET	*R/W*	
005E	DATA3 S-VOP	*R/W*	
005F	DATA3 S-OCP	*R/W*	
0060	DATA4 V-SET	*R/W*	
0061	DATA4 I-SET	*R/W*	

0062	DATA4 S-VOP	*R/W*	
0063	DATA4 S-OCP	*R/W*	
0064	DATA5 V-SET	*R/W*	
0065	DATA5 I-SET	*R/W*	
0066	DATA5 S-VOP	*R/W*	
0067	DATA5 S-OCP	*R/W*	
0068	DATA6 V-SET	*R/W*	
0069	DATA6 I-SET	*R/W*	
006A	DATA6 S-VOP	*R/W*	
006B	DATA6 S-OCP	*R/W*	
006C	DATA7 V-SET	*R/W*	
006D	DATA7 I-SET	*R/W*	
006E	DATA7 S-VOP	*R/W*	
006F	DATA7 S-OCP	*R/W*	
0070	DATA8 V-SET	*R/W*	
0071	DATA8 I-SET	*R/W*	
0072	DATA8 S-VOP	*R/W*	
0073	DATA8 S-OCP	*R/W*	
0074	DATA9 V-SET	*R/W*	
0075	DATA9 I-SET	*R/W*	
0076	DATA9 S-VOP	*R/W*	
0077	DATA9 S-OCP	*R/W*	
0078	0		
0079	0		
007A	0		
007B	0		
007C	0		
007D	0		
007E	0		
007F	0		
0080	0		