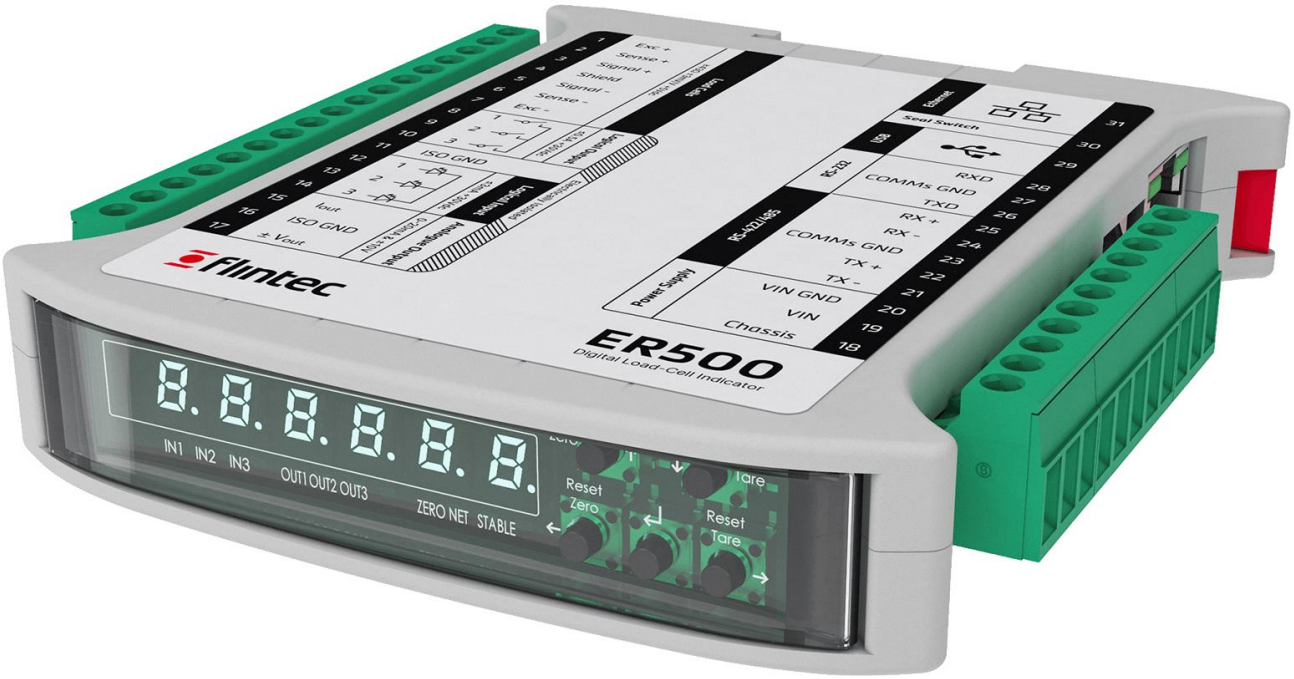
Modbus Manual



ER500

v1.03 - March 2025

Table of Contents

1. Getting Started

You will require:

- PC or PLC with RS-485 or Ethernet communication ports.
- Load-cell/scale with test weights or a load-cell simulator.
- +12V_{DC} to +24V_{DC} power supply capable of delivering approximately 300mA.
- Polling application to send data.

Note: The Flintec Device Configuration (FDC) application software does not support the Modbus protocols, there are free applications available on the internet.

1.1. How to Setup Modbus

1.1.1. Implemented functions for Modbus RTU, ASCII & TCP.

0x03 Holding Register: Used for reading 16 or 32-bit values.

0x04 Read Input Register: Used for reading 16 or 32-bit values.

0x06 Write Single Register: Used for writing 16-bit values.

0x10 Write Multiple Registers: Used for writing 32-bit values.

1.1.2. Modbus RTU or ASCII

- The Baudrate must be set using *menu 8.1* or the '*BR*' command.
- In multi-drop or 2-wire applications the user must select the duplex mode by using *menu 8.3* or the '*DX*' command.
- A Modbus address between 1 and 255 must be set in *menu 8.2* or the '*AD*' command.
- Parity is not used, please set external equipment to *No-Parity*.
- The Modbus selection can be set using the *menu 8.12* or the '*MOD*' command.

1.1.3. Modbus TCP

- The Modbus selection can be set using the *menu 8.12* or the '*MOD*' command.
    - The default IP address is 192.168.1.100 (*menu 8.9* for static IP address).
    - Port 502 needs to be assigned for Modbus traffic.

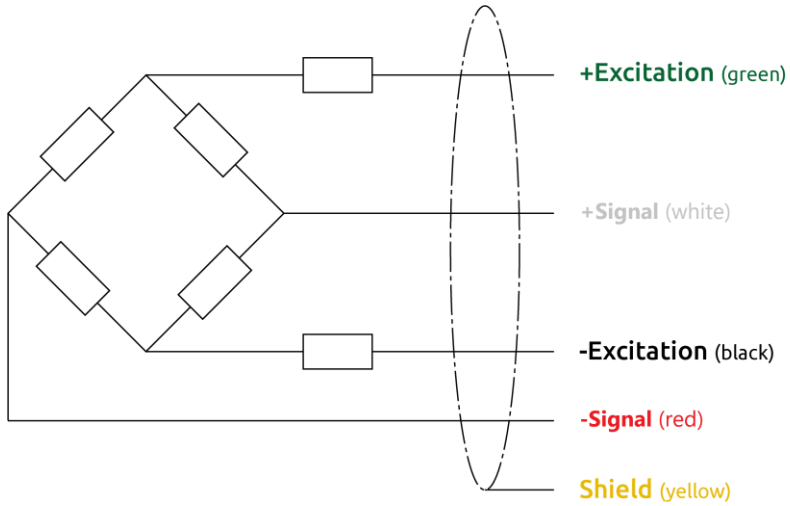
1.2. Connections

1.2.1. Load-Cell Connections

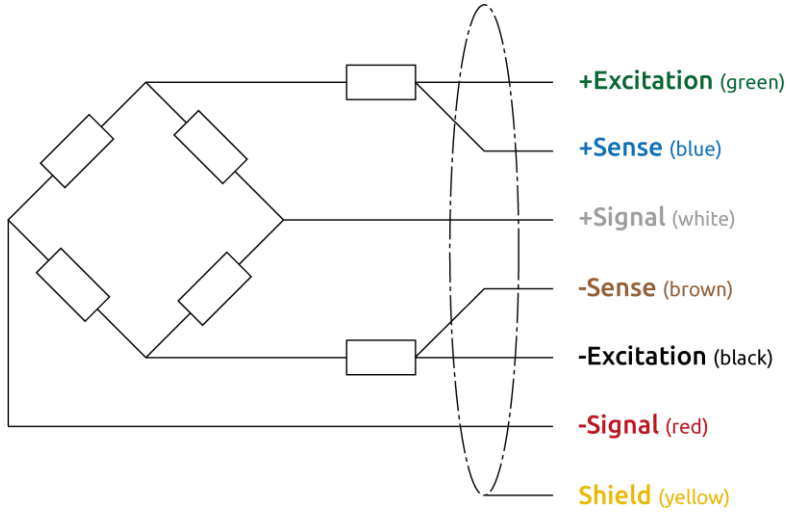
Connect the load-cell input using the reference colour coding below:

Note: If using a 4-wire configuration, tie the Excitation+ (*EXC+*) to the Sense+ & Excitation- (*EXC-*) to the Sense- for correct configuration.

4-Wire Connection:



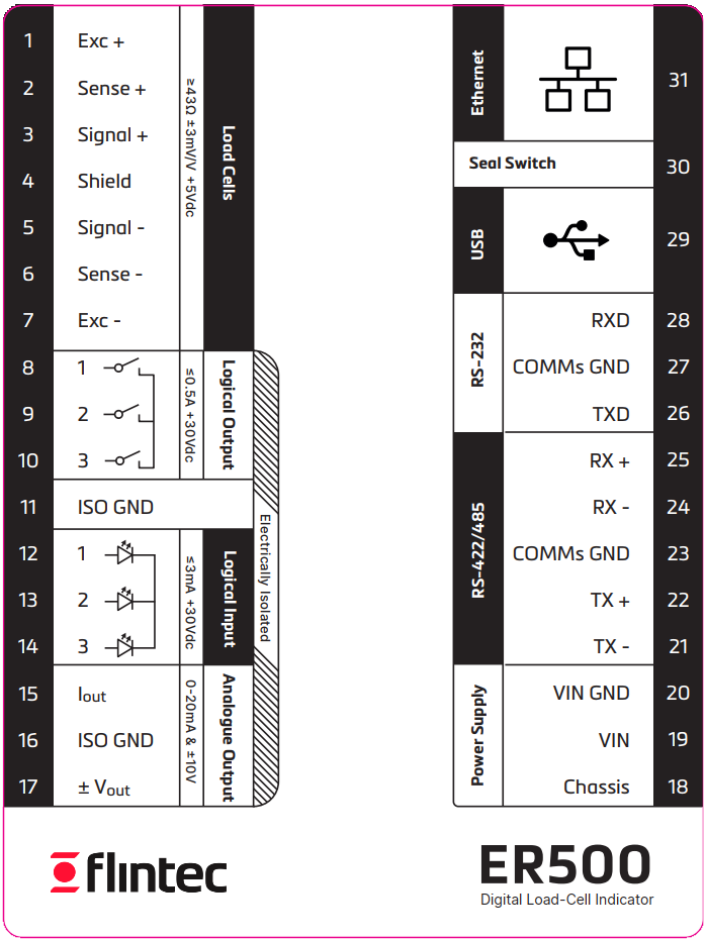
6-Wire Connection:



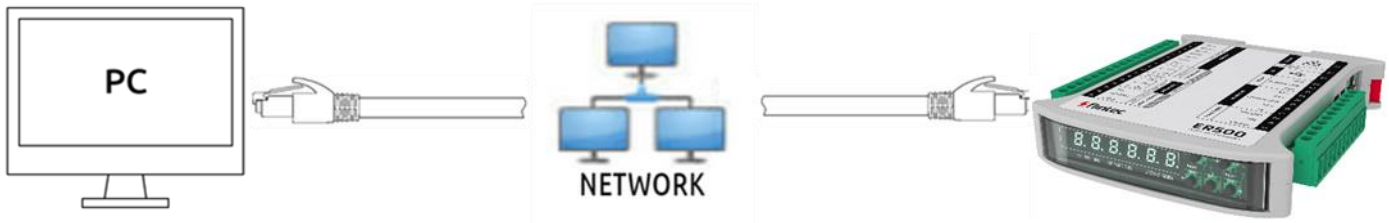
1.2.2. Power & Comms Connections

The power supply has been designed to accept a +12V_{DC} to +24V_{DC} supply. The power supply ground is not the same potential as the comms or main chassis ground pins. Only use the appropriate power and return pins for the supply, all comms and GPIOs pins to be referenced to the associated comms or main chassis ground.

Note: The ISO_GND for the analogue outputs or the GPIOs is electrically isolated from the GNDs and should not be connected to anything other than ISO_GND.



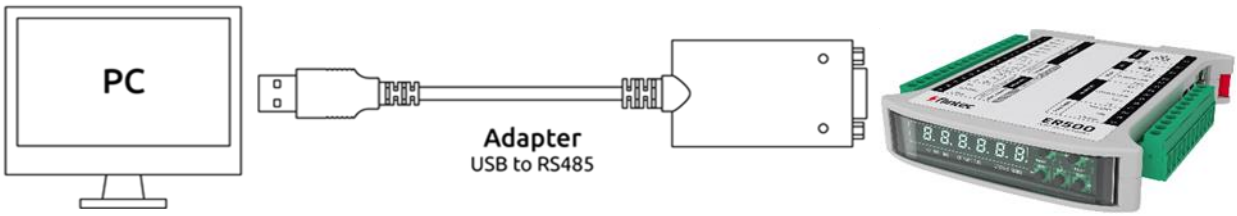
1.2.3. Ethernet Connection



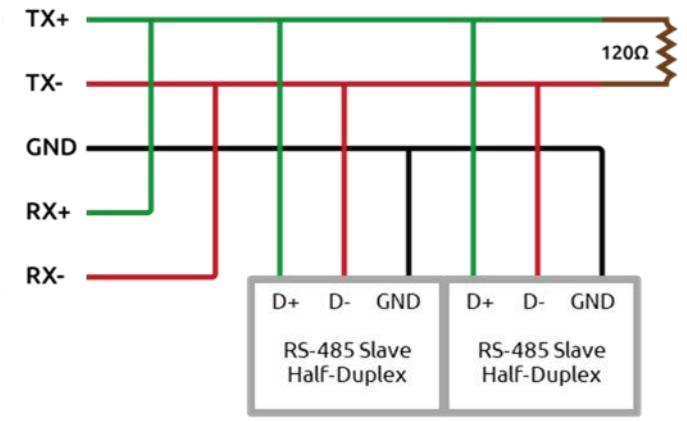
1.2.4. RS-485 Connection

Note: For '*Half-Duplex*' (RS-485) operation tie RS-485 TX+ with RS-485 RX+ and RS-485 TX- with RS-485 RX-.

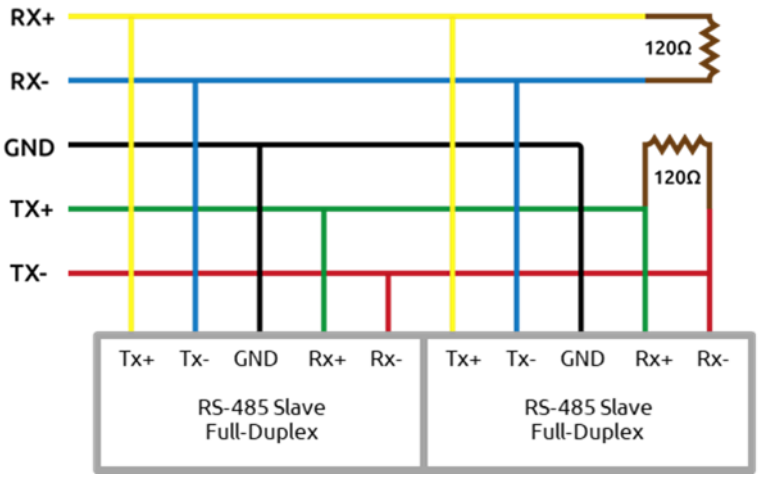
The RS-485 network requires a 120Ω termination resistor at both the host end (ER500) and at the furthest point in the network. The ER500 has a software selectable terminator built-in (see '*STR*' command).



Half-Duplex Comms



Full-Duplex Comms



2. Modbus Commands Indexes

Index (Hex)	Type	Size	Access	Function	Comment
2000	Float	2	R	Gross Weight **GG**	This index returns the latest *Gross Weight* value in divisions. The format is IEEE754 single precision floating point. The 32-bit data is obtained by reading 2, 16-bit registers from index 2000. See command description: **GG** – *Get Gross* value.
2002	Float	2	R	Net Weight **GN**	This index returns the latest *Net Weight* value in divisions. The format is IEEE754 single precision floating point. The 32-bit data is obtained by reading 2, 16-bit registers from index 2002. See command description: **GN** – *Get Net* value.
2008	Float	2	R	Average Weight **GA**	This index returns the latest average weight value in divisions. The format is IEEE754 single precision floating point. The 32-bit data is obtained by reading 2, 16-bit registers from index 2008. See command description: **GA** – *Get Average* value.
2020	Int32	2	R	Gross Weight **GG**	This index returns the *Gross Weight* value in steps. The 32-bit integer data is obtained by reading 2, 16-bit registers from index 2020. See command description: **GG** – *Get Gross* value.
2022	Int32	2	R	Net Weight	This index returns the *Net Weight* value in steps. The 32-bit integer data is obtained by reading 2, 16-bit registers from index 2022.

				GN	See command description: **GN** – *Get Net* value.
2024	Int32	2	R	mV/V Output **GMV**	This index returns the *mV/V Output* value. The 32-bit integer data is obtained by reading 2, 16-bit registers from index 2024. See command description: **GMV** – *Get mV/V Output* value.
2028	Int32	2	R	Average Weight **GA**	This index returns the latest *Average Weight* value in steps. The 32-bit integer data is obtained by reading 2, 16-bit registers from index 2028. See command description: **GA** – *Get Average* value.
202A	Int32	2	R	ADC Sample **GS**	This index returns the current filtered ADC value. The 32-bit integer data is obtained by reading 2, 16-bit registers from index 202A. See command description: **GS** – *Get ADC Sample*.
2030	Int16	1	R	Device Status **IS**	This index returns the current *Device Status*. The 16-bit integer data is obtained by reading a 16-bit register from index 2030. 0001h – Input 1 Active. 0002h – Input 2 Active. 0004h – Input 3 Active. 0100h – Stable Signal. 0200h – Zeroing Action. 0400h – Tare Active. 0800h – Warmup timer Active. 1000h – Centre Zero.

					2000h – Output 1 Active. 4000h – Output 2 Active. 8000h – Output 3 Active. See command description: **IS** – *Device Status*.
2034	Int32	2	R	Serial Number **RS**	This index returns the *Serial Number* of the ER500. The 32-bit integer data is obtained by reading 2, 16-bit registers from index 2034. See command description: **RS** – *Read Serial Number*.
2060	Int16	1	W	Set Bit Commands	This index sets the bit commands for the following functions. The 16-bit integer data is obtained by writing a 16-bit register to index 2060. The bit values are: 0001h – Reset Zero **RZ**. 0002h – Set System Zero **SZ**. 0010h – Reset Tare **RT**. 0020h – Set Tare **ST**. 0100h - Reset Peak & Valley **RM**. 0200h - Set-Hold Weight **TH**. 1000h – Correct System Zero **IZ**. 2000h – Reset mV/V Tare **RMV**. 4000h – System Reset **SR**. 8000h – Set mV/V Tare **TMV**.

2062	Int16	1	W	Software Trigger **TR**	This index is used to trigger a measurements cycle via software. The 16-bit integer data is accessed by writing a 16-bit register to index 2062. The value 0080h starts the triggered measurement. See command description: **TR** – *Software Trigger*.
2066	Int16	1	W	Save Command	This index modifies the communication *Save* settings. The 16-bit integer data is accessed by writing a 16-bit register to index 2066. See command description: 5.11 – *Save Calibration* settings. 0001h – Save DAC Calibration **AS**. 0002h – Save Calibration **CS**. 0004h – Save Setup Calibration **WP**. 0008h – Save Setpoint Calibration **SS**.
2067	Int16	1	RW	Set-point Selector	This index is used to select *Set-point*. The 16-bit integer data is accessed by writing a 16-bit register to index 2067. The values of 1 to 3 select the set-point: 2068h, 206Ah, 206Ch and 2070h. This index acts as the 'n' parameter for the **A'n'**, **H'n'**, **S'n'** and **P'n'** commands. See command description: **A'n'**– *Assignment*, **H'n'**– *Hysteresis*, **S'n'**– *Set-point*, **P'n'**– *Polarity*.

2068	Int16	1	RW	Set-point Source **A'n'**	This index is used to select *Set-point* function. The 16-bit integer data is accessed by reading or writing a 16-bit register to index 2068. 0 – *Gross Weight* as set-point source. 1 – *Net Weight* as set-point source. 2 – *Peak* value (Max) as set-point source. 3 – *Average* value as set-point source. 4 – *Hold* value as set-point source. 5 – *Peak-to-Peak* value as set-point source. 6 – *Valley* value (Min.) as set-point source. 7 – Error 4 or 5 as set-point source. See also command description: **A'n'** (n = 0, 1, 2 or, 3).
206A	Int32	2	RW	Set-point Hysteresis **H'n'**	This index reads/modifies the *Set-point Hysteresis*. The 32-bit integer data is accessed by reading or writing 2, 16-bit registers to index 206A. See command description: **H'n'** – *Set-point Hysteresis* (n = 1, 2 or 3).
206C	Int32	2	RW	Set-point Value **S'n'**	This index reads/modifies the *Set-point* limit. The 32-bit integer data is accessed by reading or writing 2, 16-bit registers to index 206C. See command description: **S'n'** – *Set-point Level* (n = 1, 2 or 3).
2070	Int16	1	RW	Set-point Polarity **P'n'**	This index reads/modifies the *Set-point Polarity*. The 16-bit integer data is accessed by reading or writing a 16-bit register to index 2070. Permitted values are: 0 or 1.

					See command description: **P'n'** – *Set-point Polarity* (n = 1, 2 or 3).
2074	Int16	1	RW	Logic Input Selector	This index reads/modifies the *Logical Input* function. The 16-bit integer data is accessed by reading or writing a 16-bit register to index 2074. Permitted values are: 1 to 3. 1 – Input 1 Selected. 2 – Input 2 Selected. 3 - Input 3 Selected.
2076	Int16	1	RW	Assign Logical Input Function **AI'n'**	This index reads/modifies the *Logic Input* function. The 16-bit integer data is accessed by reading or writing a 16-bit register to index 2076. Permitted values: 0 to 14. See command description: **AI'n'** – *Input Assignment* (n = 1, 2 or 3).
2080	Float	2	R	Peak Value **GM**	This index returns the latest *Peak* value in divisions. The format is IEEE754 single precision floating point. The 32-bit data is obtained by reading 2, 16-bit registers from index 2080. See command description: **GM** – *Get Peak* value.
2082	Int32	2	R	Peak Value **GM**	This index returns the *Peak* value in steps. The 32-bit integer data is obtained by reading 2, 16-bit registers from index 2082. See command description: **GM** – *Get Peak* value.
2084	Float	2	R	Hold Value **GH**	This index returns the latest *Hold* value in divisions. The format is IEEE754 single precision floating point.

					The 32-bit data is obtained by reading 2, 16-bit registers from index 2084. See command description: **GH** – *Get Hold* value.
2086	Int32	2	R	Hold Value **GH**	This index returns the *Hold* value in steps. The 32-bit integer data is obtained by reading 2, 16-bit registers from index 2086. See command description: **GH** – *Get Hold* value.
2088	Float	2	R	Valley Value **GV**	This index returns the latest *Valley* value in divisions. The format is IEEE754 single precision floating point. The 32-bit data is obtained by reading 2, 16-bit registers from index 2088. See command description: **GV** – *Get Valley* value.
208A	Int32	2	R	Valley Value **GV**	This index returns the latest *Valley* value in steps. The 32-bit integer data is obtained by reading 2, 16-bit registers from index 208A. See command description: **GV** – *Get Valley* value.
208C	Float	2	R	Peak-to-Peak Value **GO**	This index returns the latest *Peak-to-Peak* in divisions. The format is IEEE754 single precision floating point. The 32-bit data is obtained by reading 2, 16-bit registers from index 208C. See command description: **GO** – *Get Peak-to-Peak* value.
208E	Int32	2	R	Peak-to-Peak Value **GO**	This index returns the *Peak-to-Peak* value in steps. The 32-bit integer data is obtained by reading 2, 16-bit registers from to 208E. See command description: **GO** – *Get Peak-to-Peak* value.

2100	Int16	1	RW	Analogue Source **AP**	This index is used to select *Analogue Output Source*. The 16-bit integer data is accessed by reading or writing a 16-bit register to index 2100. Permitted values are: 0 to 8. See command description: **AP** – *Analogue Output Source*.
2102	Int32	2	RW	Analogue High **AH**	This index defines the weight value assigned to the *Analogue High* setting. The 32-bit integer data is accessed by reading or writing 2, 16-bit registers to index 2102. See command description: **AH** – *Analogue High*.
2104	Int32	2	RW	Analogue Low **AL**	This index defines the weight value assigned to the *Analogue Low* setting. The 32-bit integer data is accessed by reading or writing 2, 16-bit registers to index 2104. See command description: **AL** – *Analogue Low*.
2106	Int16	1	RW	Filter Setting **FL**	This index is used to select the *Filter Cut-off* setting. The 16-bit integer data is accessed by reading or writing a 16-bit register to index 2106. Permitted values are: 0 to 14. See command description: **FL** – *Filter Cut-off* value.
210A	Int16	1	RW	Logic Output **IO**	This index reads/modifies the status of the *Logical Output* signals. The 16-bit integer data is accessed by reading or writing a 16-bit register to index 210A. See command description: **IO** – *Logical Output* status.

210C	Int16	1	R	Logic Input **IN**	This index reads the status of the *Logical Input* signals. The 16-bit integer data is accessed by reading a 16-bit register from index 210C. See command description: **IN** – *Logical Input* status.
210E	Int32	2	RW	Measuring Time **MT**	This index reads/modifies the time over which the Average value will be built. The 32-bit integer data is accessed by reading or writing 2, 16-bit registers to index 210E. See command description: **MT** – *Measure Time*.
2110	Int16	1	RW	Filter Mode **FM**	This index reads/modifies the *Filter Mode*. The 16-bit integer data is accessed by reading or writing a 16-bit register to index 2110. Permitted values are: 0 to 1. See command description: **FM** – *Filter Mode*.
2112	Int32	2	RW	No-Motion Range **NR**	This index reads/modifies the *No-Motion Range* in divisions. The 32-bit integer data is accessed by reading or writing 2, 16-bit registers to index 2112. See command description: **NR** – *No-Motion Range*.
2114	Int32	2	RW	No-Motion Time **NT**	This index reads/modifies the *No-Motion Time*. The 32-bit integer data is accessed by reading or writing 2, 16-bit registers to index 2114. See command description: **NT** – *No-Motion Time*.
2116	Int32	2	RW	Logical Output Mask **OM**	This index reads/modifies the *Logical Outputs*. Mask. The 32-bit integer data is accessed by reading or writing 2, 16-bit registers to index 2116. See command description: **OM** – *Logical Outputs/Set-point Selection*.

2118	Int32	2	R	Tare Value **GT**	This index reads the *Tare* value in steps. The 32-bit integer data is accessed by reading 2, 16-bit registers from index 2118. See command description: **GT** – *Get Tare*.
2120	Int32	2	RW	Update Rate **UR**	This index reads/modifies the *Average*. The 32-bit integer data is accessed by reading or writing 2, 16-bit registers to index 2120. Permitted values are: 0 to 7. See command description: **UR** – *Update Rate*.
2122	Int32	2	RW	Zero Tracking **ZT**	This index reads/modifies the *Zero-Tracking* The 32-bit integer data is accessed by reading or writing 2, 16-bit registers to index 2122. Permitted values are: 0 to 255. See command description: **ZT** – *Zero-Tracking*.
2128	Int16	1	RW	Analogue Output Mode **AM**	This index reads/modifies the *Analogue Output Mode*. The 16-bit integer data is accessed by reading or writing a 16-bit register to index 2128. Permitted values are: 0 to 130. See also command description: **AM** – *Analogue Output Mode*.
2200	Int32	4	RW	Absolute Gain Calibration **AG**	This index reads/modifies the *Absolute Gain* point. The 32-bit integer data is accessed by reading or writing 4, 16-bit registers to index 2200. Send mV/V value as 2, 16-bit registers and number of divisions as 2, 16-bit registers. See command description: **AG** – *Absolute Gain*.

2202	Int32	2	RW	Absolute Zero Calibration **AZ**	This index reads/modifies the *Absolute Zero* point. The 32-bit integer data is accessed by reading or writing 2, 16-bit registers to index 2202. See command description: **AZ** – *Absolute Zero*.
2204	Int16	1	RW	Enable Calibration **CE**	This index reads/modifies the calibration counter. The 32-bit integer data is accessed by reading or writing a 16-bit register to index 2204. See command description: **CE** – *Calibration Enable*.
2206	Int32	2	RW	Calibrate Gain **CG**	This index sets the *Calibration Gain* (span) value. The 32-bit integer data is accessed by reading or writing 2, 16-bit registers to index 2206. See command description: **CG** – *Calibrate Gain*.
220A	Int32	2	RW	Calibrate Gain **GC**	This index reads/modifies the *Calibration Gain* (span) value. The 32-bit integer data is accessed by reading or writing 2, 16-bit registers to index 220A. See command description: **CG** – *Calibrate Gain*.
220C	Int32	2	RW	Calibrate Max. **CM1**	This index reads/modifies the *Maximum* value. The 32-bit integer data is accessed by reading or writing 2, 16-bit registers to index 220C. See command description: **CM'n'** – *Maximum Output* (n = 1, 2 or 3).
220E	Int32	2	RW	Calibrate Min. **CI**	This index reads/modifies the *Minimum* value. The 32-bit integer data is accessed by reading or writing 2, 16-bit registers to index 220E. See command description: **CI** – *Minimum Calibration*.

2212	Int16	1	W	Calibrate Zero **CZ**	This index sets the *Calibration Zero* point. The 16-bit integer data is accessed by writing a 16-bit register to index 2212. Accessed by writing any 16-bit integer value to the register. See command description: **CZ** – *Calibrate Zero*.
2208	Int16	1	W	Calibrate Zero **ZC**	This index reads the *Calibration Zero* point. The 16-bit integer data is accessed by writing a 16-bit register to index 2208. See command description: **CZ** – *Calibrate Zero*.
2214	Int32	2	RW	Decimal Point **DP**	This index reads/modifies the *Decimal Point Position*. The 32-bit integer data accessed by reading or writing 2, 16-bit registers to index 2214. See command description: **DP** – *Decimal Point Position*.
2216	Int16	1	RW	Display Step Size **DS**	This index reads/modifies the *Display Step Size*. The 16-bit integer data is accessed by reading or writing a 16-bit register to index 2216. See command description: **DS** – *Display Step Size*.
221A	Int32	2	RW	**CM2**	This index reads/modifies the Maximum value. The 32-bit integer data is accessed by reading or writing 2, 16-bit registers to index 221A. See command description: CM'n' – Maximum Output (n = 1, 2 or 3).
221C	Int32	2	RW	**CM3**	This index reads/modifies the Maximum value. The 32-bit integer data is accessed by reading or writing 2, 16-bit registers to index 221C. See command description: CM'n' – Maximum Output (n = 1, 2 or 3).
221E	Int32	2	RW	Initial Zero	This index reads/modifies the *Initial Zero* function @ power ON.

				ZI	The 32-bit integer data is accessed by reading or writing 2, 16-bit registers to index 221E. See command description: **ZI** – *Initial Zero*.
2220	Int32	2	RW	Zero Range **ZR**	This index reads/modifies the *Zero Range*. The 32-bit integer data is accessed by reading or writing 2, 16-bit registers to index 2220. A value of 0 enables the standard zero range of ±2% of maximum. See command description: **ZR** – *Zero Range*.
2224	Int32	2	RW	Store Tare **TN**	This index reads/modifies the *Tare* storing in NVM. The 32-bit integer data is accessed by reading or writing 2, 16-bit registers to index 2224. See command description: **TN** – *Store Zero* value.
2226	Int32	2	RW	Store Zero **ZN**	This index reads/modifies the *Zero* storing in NVM. The 32-bit integer data is accessed by reading or writing 2, 16-bit registers to index 2226. See command description: **ZN** – *Store Zero* value.
2228	Int32	2	RW	Maximum Tare Level **MTL**	This index reads/modifies the *Maximum Tare Level*. The 32-bit integer data is accessed by reading or writing 2, 16-bit registers to index 2228. See command description: **MTL** – *Maximum Tare Level*.
2230	Int16	1	RW	DHCP Allocation **DH**	This index reads/modifies the *DHCP Allocation*. The 16-bit integer data is accessed by reading or writing a 16-bit register to index 2230. See command description: **DH** – *DHCP Allocation*.
211A	Int32	2	RW	Start Delay **SD**	This index Reads/Modifies the *Start Delay* time. The 32-bit integer data accessed by reading or writing 2, 16-bit registers to index 211A.

					See command description: **SD** – *Start Delay*.
211C	Int32	2	RW	Trigger Edge **TE**	This index reads/modifies the trigger slope edge detection. The 32-bit integer data is accessed by reading or writing 2, 16-bit registers to index 211C. See command description: **TE** – *Trigger Edge*.
211E	Int32	2	RW	Trigger Level **TL**	This index reads/modifies the *Trigger Level*. The 32-bit integer data is accessed by reading or writing 2, 16-bit registers to index 211E. See command description: **TL** – *Trigger Level*.
2408	Int32	2	RW	Hold Time **HT**	This index reads/modifies the *Hold Time* of set-point. The 32-bit integer data is accessed by reading or writing 2, 16-bit registers to index 2408. See command description: **HT** – *Hold Time*.
300C	Int32	2	RW	IP Address **NA**	This index reads/modifies the static *IP Address*. The 32-bit integer data accessed by reading or writing 2, 16-bit registers to index 300C. Byte 1 Group 1 – AA (hex). Byte 2 Group 2 – BB (hex). Byte 3 Group 3– CC (hex). Byte 4 Group 4– DD (hex). See command description: **NA** – *Network Address*. (AA:BB:CC:DD).
3300	Int16 Int8	5	R	Combined Result Integer **GW**	This index reads the *Net Weight, Gross Weight* & Status Byte – in steps. The 2x 32-bit integer data accessed by reading 4, 16-bit registers from index 3300. 2x 32-bit Integer Data Bytes.

					1x 8-bit Integer Status Flag (upper byte). 1x 8-bit Integer Checksum (lower byte) – Not inserted as the Modbus message already has a checksum applied to the message. See command description: **GW** – Get data string '*Net, Gross & Status*'.
3500	Float, Int8	5	R	Combined Result Float **GW**	This index reads the *Net Weight, Gross Weight* & Status Byte – in divisions. The 2x 32-bit Floating point data is accessed by reading 4, 16-bit registers from index 3500. The format for floating points variables is IEEE754 single precision floating point. 2x 32-bit Float Data Bytes. 1x 8-bit Integer Status Flag (upper byte). 1x 8-bit Integer – Checksum (lower byte) – Not inserted as the Modbus checksum message already has a checksum applied. See command description: **GW** – Get data string '*Net, Gross & Status*'.
243A	Int16 Int8	5	R	Combined Result Integer **GL**	This index reads the *Average Weight, Gross Weight* & Status Byte – in steps. The 2x 32-bit integer data accessed by reading 4, 16-bit registers from index 243A. 2x 32-bit Integer Data Bytes. 1x 8-bit Integer Status Flag (upper byte). 1x 8-bit Integer Checksum (lower byte) – Not inserted as the Modbus message already has a checksum applied to the message. See command description: **GL** – Get data string '*Average, Gross & Status*'.

2438	Float, Int8	5	R	Combined Result Float **GL**	This index reads the *Average Weight, Gross Weight* & Status Byte – in divisions. The 2x 32-bit Floating point data is accessed by reading 4, 16-bit registers from index 2438. The format for floating points variables is IEEE754 single precision floating point. 2x 32-bit Float Data Bytes. 1x 8-bit Integer Status Flag (upper byte). 1x 8-bit Integer – Checksum (lower byte) – Not inserted as the Modbus checksum message already has a checksum applied. See command description: **GL** – Get data string '*Average, Gross & Status*'.
241C	Int16	1	RW	Device Address **AD**	This index reads/modifies the communication *Device Address* setting. The 16-bit integer data is accessed by reading or writing a 16-bit register to index 241C. See command description: **AD** – *Device Address*.
2420	Int32	2	RW	Baud-Rate **BR**	This index reads/modifies the communication *Baud-Rate* setting. The 32-bit integer data is accessed by reading or writing 2, 16-bit registers to index 2420. See command description: **BR** – *Baud-Rate*.
2422	Int16	1	W	Close Address **CL**	This index sends the *Close Address* command. The 16-bit integer data is accessed by writing a 16-bit register to index 2422. See command description: **CL** – *Close Address*.
2424	Int32	2	R	Current IP Address **DA**	This index reads the network *Current IP Address*. The 32-bit integer data is accessed by reading 2, 16-bit registers from index 2424. Byte 1 Group 1 – AA (hex).

					Byte 2 Group 2 – BB (hex). Byte 3 Group 3– CC (hex). Byte 4 Group 4– DD (hex). See command description: **DA** – *Current IP Address*.
242A	Int32	2	RW	Current Display Format **DISP**	This index reads/modifies the current *Display Format* setting of the front panel. The 32-bit integer data is accessed by reading or writing 2, 16-bit registers to index 242A. See command description: **DISP** – *Display Format*.
2432	Int16	1	RW	Duplex Setting **DX**	This index reads/modifies the communication *Duplex Setting*. The 32-bit integer data is accessed by reading or writing a 16-bit register to index 2432. See command description: **DX** – *Duplex Setting*.
2232	Int16	1	W	Restore Defaults **FD**	This index modifies the *Factory Default*. The 16-bit integer data is accessed by writing a 16-bit register to index 2232. Accessed by writing any 16-bit integer value to the register. See command description: **FD** – *Factory Default*.
2434	Int32	2	R	Display Firmware Version **FFD**	This index returns the current *Display Firmware Version* of the ER500. The 32-bit integer data is obtained by reading 2, 16-bit registers from index 2434. Byte 1 Major Revision – AA (hex). Byte 2 Minor Revision – BB (hex). Byte 3 Not Used.

					Byte 4 Not Used. See command description: FFD – *Flintec Display Firmware Version*. (AA:BB).
2436	Int32	2	R	Firmware Version **FFV**	This index returns the current *Firmware Version* of the ER500. The 32-bit integer data is obtained by reading 2, 16-bit registers from index 2436. Byte 1 Legally Relevant – AA (hex). Byte 2 Major Revision – BB (hex). Byte 3 Minor Revision – CC (hex). Byte 4 Not Used. See command description: FFV – *Flintec Firmware Version*. (AA:BB.CC).
2414	Int16	1	RW	Modbus Select **MOD**	This index modifies the *Modbus Mode*. The 16-bit integer data is accessed by reading or writing a 16-bit register to index 2414. 0 – OFF. 1 – ASCII. 2 – RTU. 3 – TCP. See command description: **MOD** – *Modbus Mode*.
300A	Int16	3	R	MAC Address **MA**	This index returns the *MAC Address*. The 3x, 16-bit integer data is accessed by reading 3x, 16-bit registers from index 300A/B/C. Flintec Assigned Address Range 04:C3:E6:Bx:xx:xx [Set in Factory – Unique Address].

					See command description: **MA** – *MAC Address.*
2218	Int32	2	RW	*Multi-Range /Multi Interval* **MR**	This index reads/modifies the *Multi-Range /Multi Interval* setting. The 32-bit integer data is accessed by reading or writing 2, 16-bit registers to index 2218. See command description: **MR** – *Multi-Range /Multi Interval*.
243C	Int32	2	R	RAW ADC Sample **GRS**	This index reads the current *RAW ADC Sample* data. The 32-bit integer data is accessed by reading 2, 16-bit registers from index 243C. See command description: **GRS** – *Get RAW ADC Sample*.
2998	Int32	2	RW	Gateway Address **NG**	This index reads/modifies the *Network Gateway Address*. The 32-bit integer data accessed by reading or writing 2, 16-bit registers to index 2998. Byte 1 Group 1 – AA (hex). Byte 2 Group 2 – BB (hex). Byte 3 Group 3– CC (hex). Byte 4 Group 4– DD (hex). See command description: **NG** – *Network Gateway Address*. (AA:BB:CC:DD).
2996	Int32	2	RW	Network Mask **NM**	This index reads/modifies the *Network Mask*. The 32-bit integer data accessed by reading or writing 2, 16-bit registers to index 2996. Byte 1 Group 1 – AA (hex). Byte 2 Group 2 – BB (hex). Byte 3 Group 3– CC (hex). Byte 4 Group 4– DD (hex).

					See command description: **NM** – *Network Mask*. (AA:BB:CC:DD).
2440	Int32	2	RW	Output Format **OF**	This index reads/modifies the *Output Format*. The 32-bit integer data is accessed by reading or writing 2, 16-bit registers to index 2440. See command description: **OF** – *Output Format*.
2442	Int16	1	RW	Open Device **OP**	This index reads/modifies the *Open Device Address*. The 16-bit integer data is accessed by reading or writing a 16-bit register to index 2442. See command description: **OP** – *Open Device Address*.
2448	Int32	2	RW	Preset Tare **SP**	This index reads/modifies the *Preset Tare*. The 32-bit integer data is accessed by reading or writing 2, 16-bit registers to index 2448. See command description: **SP** – *Save Preset Tare Setup*.
244A	Int32	2	RW	Set Termination **STR**	This index reads/modifies the *Termination* impedance. The 32-bit integer data is accessed by reading or writing 2, 16-bit registers to index 244A. See command description: **STR** – *Set Termination*.
244C	Int32	2	RW	Transmission Delay **TD**	This index reads/modifies the *Transmission Delay* time. The 32-bit integer data is accessed by reading or writing 2, 16-bit registers to index 244C. See command description: **TD** – *Transmission Delay*.
2450	Int16	1	RW	Tare Mode **TM**	This index reads/modifies the *Tare Mode*. The 16-bit integer data is accessed by reading or writing a 16-bit register to index 2450. See command description: **TM** – *Tare Mode*.
2452	Int32	2	RW	Preset Tare	This index reads/modifies the *Preset Tare Setup*.

				TP	The 32-bit integer data is accessed by reading or writing 2, 16-bit registers to index 2452. See command description: **TP** – *Save Preset Tare Setup*.
2458	Int32	2	RW	Warm-Up Time **WT**	This index reads/modifies the *Warm-up Time*. The 32-bit integer data is accessed by reading or writing 2, 16-bit registers to index 2458. See command description: **WT** – *Warm-up Time*.
245A	Int32	2	RW	DAC Slew Rate **SLEW**	This index reads/modifies the *Slew Rate* of the Analogue Output DAC. The 32-bit integer data is accessed by reading or writing 2, 16-bit registers to index 245A. See command description: **SLEW** – *Slew Rate*.
245C	Int32	2	RW	Slew Rate Filter Cut-off **SRC**	This index reads/modifies the DAC *Slew Rate Filter Cut-off*. The 32-bit integer data is accessed by reading or writing 2, 16-bit registers to index 245C. See command description: **SRC** – DAC *Slew Rate Filter Cut-off*.
2460	Int32	2	RW	Slew Rate Setting **SRS**	This index reads/modifies the DAC *Slew Rate Setting*. The 32-bit integer data is accessed by reading or writing 2, 16-bit registers to index 2460. See command description: **SRS** – *Slew Rate Setting*.

3. Check-Weigher Modbus Command Indexes

240A	Int32	2	RW	Tare Window **TW**	This index reads/modifies the re-trigger *Tare Window*. The 32-bit integer data is accessed by reading or writing 2, 16-bit registers to index 240A. See command description: **TW** – *Tare Window*.
240C	Int32	2	RW	Auto Tare Time **TI**	This index reads/modifies the re-trigger auto *Tare Time*. The 32-bit integer data is accessed by reading or writing 2, 16-bit registers to index 240C. See command description: **TI** – *Tare Time*.
2454	Int32	2	RW	Re-Trigger Stop **TS**	This index reads/modifies the *Re-Trigger Stop* value. The 32-bit integer data is accessed by reading or writing 2, 16-bit registers to index 2454. See command description: **TS** –*Re-Trigger Stop* value.
2446	Int32	2	RW	Re-Trigger Window **RW**	This index reads/modifies the *Re-Trigger Window* function. The 32-bit integer data is accessed by reading or writing 2, 16-bit registers to index 2446. See command description: **RW** –*Re-Trigger Window*.
2456	Int32	2	RW	Average Time for Re-Trigger **TT**	This index reads/modifies the *Average Time for Re-Trigger* function. The 32-bit integer data is accessed by reading or writing 2, 16-bit registers to index 2456. See command description: **TT** – *Average Time for Re-Trigger*.

242C	Int32	2	RW	Short-Time Averaging **DT** (Check-Weigher)	This index reads/modifies the *Short-Time Averaging* period. The 32-bit integer data is accessed by reading or writing 2, 16-bit registers to index 242C. See command description: **DT** – *Short-Time Averaging* period.

4. Bottle/Filling Modbus Command Indexes

2026	Int32	2	R	Get Last Dosed Weight **GD**	This index reads the *Last Dosed Weight*. The 32-bit integer data is accessed by reading 2, 16-bit registers from index 2026. See command description: **GD** – *Last Dosed Weight*.
2416	Int16	1	RW	Filling Register Selector	This index is used to select the Filling configuration/control register. The 16-bit integer data is accessed by writing a 16-bit register to index 2416. Permitted values are 1 to 21. See command description: **PD'n'** (n = 1, 2 to 21).
2418	Int32	2	RW	Filling Data Register	This index is used to read/modify the current Filling register. The 32-bit integer data is accessed by reading or writing 2, 16-bit registers to index 2418.
246A	Int32	2	R	Filling Status **DI** (Filling)	This index reads the sequence *Filling Status*. The 32-bit integer data is accessed by reading 2, 16-bit registers from index 246A. See command description: **DI** – *Filling Status*.
2466	Int16	1	W	Save Dosing Setup **SDD**	This index writes the *Save Dosing Setup*. The 16-bit integer data is accessed by writing a 16-bit register to index 2466. See command description: **SDD** – *Save Dosing Setup*.
2464	Int16	1	W	Start Cycle **SC**	This index initiates the *Start Cycle* command (Filling/Bottling). The 16-bit integer data is accessed by writing a 16-bit register to index 2464. See command description: **SC** – *Start Cycle*.
2462	Int16	1	W	Abort Cycle	This index initiates a filling sequence, *Abort Cycle* command.

				AC (Filling)	The 16-bit integer data is accessed by writing a 16-bit register to index 2462. See command description: **AC** – *Abort Cycle*.
246C	Int32	2	R	Get Last Tare Weight **DT** (Filling)	This index reads the *Last Tare Weight*. The 32-bit integer data is accessed by reading 2, 16-bit registers from index 246C. See command description: **DT** – *Last Tare Weight*.

5. Modbus Exceptions

MODBUS Exceptions	Explanation for Return Code
01 – MODBUS ILLEGAL FUNCTION	Generic Failure.
02 – MODBUS ILLEGAL DATA ADDRESS	Out-of-range value Incorrect message command parameters Incorrect command quantity Unknown command index
03 – MODBUS ILLEGAL DATA VALUE	Message length incorrect. ER500 Conditions are not correct. Tac counter not enabled. Unstable Command data values out of range. Incorrect ER500 model. Unexpected data value. ER500 configuration is not correct.
04 – MODBUS SLAVE DEVICE FAILURE	Device communication values are incorrect. Incorrect multidrop communication address
05 – MODBUS ACKNOWLEDGE	Message processed successfully

flintec
Where Quality Meets Precision

www.flintec.com