# ELR 5000 (ELM 5000) register list for devices with HMI firmware from V2.04 (check the installed version in your device's MENU in item ABOUT HW, SW)

| Modbus address | Read coils (0x01) | Read holding registers (0x03) | Write single coil (0x05) | Write single register (0x06) | Write multiple registers (0x10) | Description | Access | Data type | Data length in bytes | Number of registers | Data | Example |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | | x | | | | Device class | R | uint(16) | 2 | 1 | | 41 = ELR 5000 Series |
| 1 | | x | | | | Device type | R | char | 40 | 20 | ASCII | ELR 5080-25 |
| 21 | | x | | | | Manufacturer | R | char | 40 | 20 | ASCII | |
| 41 | | x | | | | Manufacturer address | R | char | 40 | 20 | ASCII | |
| 61 | | x | | | | Manufacturer ZIP code | R | char | 40 | 20 | ASCII | |
| 81 | | x | | | | Manufacterer phone number | R | char | 40 | 20 | ASCII | |
| 101 | | x | | | | Manufacturer website | R | char | 40 | 20 | ASCII | |
| 121 | | x | | | | Nominal voltage | R | float | 4 | 2 | Floating point number IEEE754 | 80 |
| 123 | | x | | | | Nominal current | R | float | 4 | 2 | Floating point number IEEE754 | 25 |
| 125 | | x | | | | Nominal power | R | float | 4 | 2 | Floating point number IEEE754 | 320 |
| 151 | | x | | | | Article no. | R | char | 40 | 20 | ASCII | 33220430 |
| 171 | | x | | | | Serial no. | R | char | 40 | 20 | ASCII | 100010002 |
| 191 | | | | | x | User text | RW | char | 40 | 20 | ASCII | |
| 211 | | x | | | | Firmware version (HMI) | R | char | 40 | 20 | ASCII | V2.01 02.12.2015 |
| 231 | | x | | | | Firmware version (DR) | R | char | 40 | 20 | ASCII | V2.0.1 |
| 402 | x | x | x | | | Remote mode | RW | uint(16) | 2 | 1 | Coils : Remote | 0x0000 = off; 0xFF00 = on |
| 405 | x | x | x | | | DC input | RW | uint(16) | 2 | 1 | Coils : Input | 0x0000 = off; 0xFF00 = on |
| 407 | x | x | x | | | Condition of DC input after power fail alarm | RW | uint(16) | 2 | 1 | Coils : Auto-On | 0x0000 = off; 0xFF00 = auto-on |
| 408 | x | x | | x | | Condition of DC input after powering the device | RW | uint(16) | 2 | 1 | Reg : Power-On | 0xFFFF = off; 0xFFFE = restore |
| 410 | | | x | | | Restart of the device (warm start) | W | uint(16) | 2 | 1 | Coils : Restart | 0xFF00 = execute |
| 411 | | x | | | | Acknowledge alarms | W | uint(16) | 2 | 1 | Coils : Alarms | 0xFF00 = acknowledge |
| 500 | | x | | x | | Set voltage value | RW | uint(16) | 2 | 1 | 0x0000 - 0xD0E5 (0 - 102%) | Voltage value (for translation see programming guide) |
| 501 | | x | | x | | Set current value | RW | uint(16) | 2 | 1 | 0x0000 - 0xD0E5 (0 - 102%) | Current value (for translation see programming guide) |
| 502 | | x | | x | | Set power value | RW | uint(16) | 2 | 1 | 0x0000 - 0xD0E5 (0 - 102%) | Power value (for translation see programming guide) |
| 505 | | x | | | | Device state | R | uint(32) | 4 | 2 | Bit 0 - 4 : Control location | 0x00 = free; 0x01 = local; 0x06 = Ethernet |
| | | | | | | | | | | | Bit 5 : - | |
| | | | | | | | | | | | Bit 6 : - | |
| | | | | | | | | | | | Bit 7 : Input state | 0 = off; 1 = on |
| | | | | | | | | | | | Bit 8 : - | |
| | | | | | | | | | | | Bit 9-10 : Regulation mode | 00 = CV; 10 = CC; 11 = CP |
| | | | | | | | | | | | Bit 11 : Remote control | 0 = off; 1 = on |
| | | | | | | | | | | | Bit 12 : - | |
| | | | | | | | | | | | Bit 13 : Sequence mode | 0 = stopped; 1 = running |
| | | | | | | | | | | | Bit 14 : Remote sense | 0 = internal; 1 = external |
| | | | | | | | | | | | Bit 15 : Alarms | 0 = no alarm; 1 = min. one alarm active |
| | | | | | | | | | | | Bit 16 : OVP | 0 = none; 1 = active |
| | | | | | | | | | | | Bit 17 : OCP | 0 = none; 1 = active |
| | | | | | | | | | | | Bit 18 : OPP | 0 = none; 1 = active |
| | | | | | | | | | | | Bit 19 : OT | 0 = none; 1 = active |
| | | | | | | | | | | | Bit 20 : - | |
| | | | | | | | | | | | Bit 21 : Power fail | 0 = none; 1 = active |
| | | | | | | | | | | | Bit 24-31: - | |
| 507 | | x | | | | Actual voltage | R | uint(16) | 2 | 1 | 0x0000 - 0xFFFF (0 - 125%) | Actual voltage (for translation see programming guide) |
| 508 | | x | | | | Actual current | R | uint(16) | 2 | 1 | 0x0000 - 0xFFFF (0 - 125%) | Actual current (for translation see programming guide) |
| 509 | | x | | | | Actual power | R | uint(16) | 2 | 1 | 0x0000 - 0xFFFF (0 - 125%) | Actual power (for translation see programming guide) |
| 520 | | x | | | | Count of OV alarms since power up | R | uint(16) | 2 | 1 | 0x0000 - 0xFFFF | Count |
| 521 | | x | | | | Count of OC alarms since power up | R | uint(16) | 2 | 1 | 0x0000 - 0xFFFF | Count |
| 522 | | x | | | | Count of OP alarms since power up | R | uint(16) | 2 | 1 | 0x0000 - 0xFFFF | Count |
| 523 | | x | | | | Count of OT alarms since power up | R | uint(16) | 2 | 1 | 0x0000 - 0xFFFF | Count |
| 524 | | x | | | | Count of PF alarms since power up | R | uint(16) | 2 | 1 | 0x0000 - 0xFFFF | Count |
| 550 | | x | | x | | Overvoltage protection threshold (OVP) | RW | uint(16) | 2 | 1 | 0x0000 - 0xE147 (0 - 110%) | OVP threshold (for translation see programming guide) |
| 553 | | x | | x | | Overcurrent protection threshold (OCP) | RW | uint(16) | 2 | 1 | 0x0000 - 0xE147 (0 - 110%) | OCP threshold (for translation see programming guide) |
| 556 | | x | | x | | Overpower protection threshold (OPP) | RW | uint(16) | 2 | 1 | 0x0000 - 0xE147 (0 - 110%) | OPP threshold (for translation see programming guide) |
| 850 | x | x | x | | | Sequence generator: Sequence: Start/stop | RW | uint(16) | 2 | 1 | Coils : Start/Stop | 0x0000 = stop; 0xFF00 = start |
| 859 | | x | | x | | Sequence generator: Start sequence point | RW | uint(16) | 2 | 1 | 0x0001...0x0064 | |
| 860 | | x | | x | | Sequence generator: End sequence point | RW | uint(16) | 2 | 1 | 0x0001...0x0064 | |
| 861 | | x | | x | | Sequence generator: Sequence cycles | RW | uint(16) | 2 | 1 | 0x0000...0x03E7 | 0x0000 = infinite; 0x0001...0x03E7 = Number of sequence cycles |
| 862 | | | | x | | Sequence generator: Submit sequence data | W | uint(16) | 2 | 1 | Coils : Submit | 0xFF00 = submit |
| 900 | | x | | | x | Sequence generator: Setup for sequence point 1 | RW | float | 16 | 8 | Bytes 0-3: U(DC) in V | Floating point number in IEEE754 format, see device manual for value range, chapter about sequence generator |
| | | | | | | | | float | | | Bytes 4-7: I(DC) in A | |
| | | | | | | | | float | | | Bytes 8-11: P(DC) in W | |
| | | | | | | | | float | | | Bytes 12-15: Sequence point time in ms | Integer, 1...36000000 |
| ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | | | | | ↓ | |
| 1692 | | x | | | x | Sequence generator: Setup for sequence point 100 | RW | float | 16 | 8 | Bytes 0-3: U(DC) in V | Floating point number in IEEE754 format, see device manual for value range, chapter about sequence generator |
| | | | | | | | | float | | | Bytes 4-7: I(DC) in A | |
| | | | | | | | | float | | | Bytes 8-11: P(DC) in W | |
| | | | | | | | | float | | | Bytes 12-15: Sequence point time in ms | Integer, 1...36000000 |
| 9000 | | x | | x | | Upper limit of voltage set value (U-max) | RW | uint(16) | 2 | 1 | 0x0000 - 0xD0E5 (0 - 102%) | Voltage value (for translation see programming guide) |
| 9001 | | x | | x | | Lower limit of voltage set value (U-min) | RW | uint(16) | 2 | 1 | 0x0000 - 0xD0E5 (0 - 102%) | Voltage value (for translation see programming guide) |
| 9002 | | x | | x | | Upper limit of current set value (I-max) | RW | uint(16) | 2 | 1 | 0x0000 - 0xD0E5 (0 - 102%) | Current value (for translation see programming guide) |
| 9003 | | x | | x | | Lower limit of current set value (I-min) | RW | uint(16) | 2 | 1 | 0x0000 - 0xD0E5 (0 - 102%) | Current value (for translation see programming guide) |
| 9004 | | x | | x | | Upper limit of power set value (P-max) | RW | uint(16) | 2 | 1 | 0x0000 - 0xD0E5 (0 - 102%) | Power value (for translation see programming guide) |
| 10007 | x | x | x | | | Ethernet: TCP Keep-Alive | RW | uint(16) | 2 | 1 | Coils : TCP keep-alive | 0x0000 = off; 0xFF00 = on |
| 10008 | x | x | x | | | Ethernet: DHCP | RW | uint(16) | 2 | 1 | Coils : DHCP on/off | 0x0000 = off; 0xFF00 = on |
| 10010 | x | x | x | | | Protocol: Modbus | RW | uint(16) | 2 | 1 | Coils : MODBUS on/off | 0x0000 = off; 0xFF00 = on |
| 10011 | x | x | x | | | Protocol: SCPI | RW | uint(16) | 2 | 1 | Coils : SCPI on/off | 0x0000 = off; 0xFF00 = on |
| 10502 | | x | | | x | Ethernet: IP address | RW | uint(8) | 4 | 2 | Bytes 0 - 3: 0..255 | 192.168.0.2 (default) |
| 10504 | | x | | | x | Ethernet: Subnet mask | RW | uint(8) | 4 | 2 | Bytes 0 - 3: 0..255 | 255.255.255.0 (default) |
| 10506 | | x | | | x | Ethernet: Gateway | RW | uint(8) | 4 | 2 | Bytes 0 - 3: 0..255 | 192.168.0.1 (default) |
| 10508 | | x | | | x | Ethernet: Host name | RW | char | 54 | 27 | ASCII | "Client" (default) |
| 10535 | | x | | | x | Ethernet: Domain name | RW | char | 54 | 27 | ASCII | "Workgroup" (default) |
| 10562 | | x | | | x | Ethernet: DNS | RW | uint(8) | 4 | 2 | Bytes 0 - 3: 0..255 | 0.0.0.0 (default) |
| 10567 | | x | | | | Ethernet: MAC | R(W) | uint(8) | 6 | 3 | Bytes 0-6: 0..255 | 00:50:C2:C3:12:34 or 00-50-C2-C3-12-34 |
| 10572 | | x | | x | | Ethernet: Port | RW | uint(16) | 2 | 1 | 0..65536 (except 80) | 5025 (default) |
| 10573 | | x | | x | | Ethernet: Interface connection timeout in seconds | RW | uint(16) | 2 | 1 | 0..65535 | 5 sec (default) |
| 11000 | | x | | x | | MPP Tracking: MPP-Mode | RW | uint(16) | 2 | 1 | 0..4 | 0 (default), 1 (MPP1), 2 (MPP2), 3 (MPP3), 4 (MPP4) |
| 11001 | | x | | x | | MPP Tracking: Uoc (Setup) | RW | uint(16) | 2 | 1 | 0x0000 - 0xCCCC (0 - 100%) | Voltage value in % of Unom (for translation see programming guide) |
| 11002 | | x | | x | | MPP Tracking: Isc (Setup) | RW | uint(16) | 2 | 1 | 0x0000 - 0xCCCC (0 - 100%) | Current value in % of Inom (for translation see programming guide) |
| 11003 | | x | | x | | MPP Tracking: Umpp (Setup) | RW | uint(16) | 2 | 1 | 0x0000 - 0xCCCC (0 - 100%) | Voltage value in % of Unom (for translation see programming guide) |
| 11004 | | x | | x | | MPP Tracking: Impp (Setup) | RW | uint(16) | 2 | 1 | 0x0000 - 0xCCCC (0 - 100%) | Current value in % of Inom (for translation see programming guide) |
| 11005 | | x | | x | | MPP Tracking: Pmpp (Setup) | RW | uint(16) | 2 | 1 | 0x0000 - 0xCCCC (0 - 100%) | Power value in % of Pnom (for translation see programming guide) |
| 11006 | | x | | x | | MPP Tracking: DeltaP (Setup) | RW | uint(16) | 2 | 1 | 0x0000 - 0x6666 (0 - 50%) | Power value in % of Pnom (for translation see programming guide) |
| 11007 | | x | | | | MPP Tracking: Umpp (Result in MPP1/2/4) | R | uint(16) | 2 | 1 | 0x0000 - 0xCCCC (0 - 100%) | Voltage value in % of Unom (for translation see programming guide) |
| 11008 | | x | | | | MPP Tracking: Impp (Result in MPP1/2/4) | R | uint(16) | 2 | 1 | 0x0000 - 0xCCCC (0 - 100%) | Current value in % of Inom (for translation see programming guide) |
| 11009 | | x | | | | MPP Tracking: Pmpp (Result in MPP1/2/4) | R | uint(16) | 2 | 1 | 0x0000 - 0xCCCC (0 - 100%) | Power value in % of Pnom (for translation see programming guide) |
| 11010 | x | x | x | | | MPP Tracking: Start/Stop | RW | uint(16) | 2 | 1 | Coils: Start/Stop | 0x0000 = stop; 0xFF00 = start |
| 11011 | x | x | | | | MPP Tracking: Finished (Function status for MPP1/2/4) | R | uint(16) | 2 | 1 | Coils: | 0x0000 = running; 0xFF00 = finished |
| 11012 | x | x | | | | MPP Tracking: Error during function | R | uint(16) | 2 | 1 | Coils: | 0x0000 = no error; 0xFF00 = error |
| 11013 | | x | | x | | MPP-Tracking : Interval (Setup) | RW | uint(16) | 2 | 1 | 0x0005 - 0xEA60 | Regulation & measuring interval in milliseconds, either for tracking in modes 1 and 2 or for user curve progression in mode 3 |
| 11014 | | x | | x | | MPP4 : Start | RW | uint(16) | 2 | 1 | 0x0001 - 0x0064 | Start voltage value out of 100 (related to registers 11100-11199) for use in MPP4 mode |
| 11015 | | x | | x | | MPP4 : End | RW | uint(16) | 2 | 1 | 0x0001 - 0x0064 | End voltage value out of 100 (related to registers 11100-11199) for use in MPP4 mode |
| 11016 | | x | | x | | MPP4 : Repetitions | RW | uint(16) | 2 | 1 | 0x0000 - 0xFFFF | 0x0000 = no repetitions |
| 11100 | | x | | | x | MPP Tracking: User curve (MPP4 mode) voltage values 1-20 | RW | uint(16) | 40 | 20 | 0x0000 - 0xCCCC (0 - 100%) | Voltage value in % of Unom (for translation see programming guide) |
| 11120 | | x | | | x | MPP Tracking: User curve (MPP4 mode) voltage values 21-40 | RW | uint(16) | 40 | 20 | 0x0000 - 0xCCCC (0 - 100%) | Voltage value in % of Unom (for translation see programming guide) |
| 11140 | | x | | | x | MPP Tracking: User curve (MPP4 mode) voltage values 41-60 | RW | uint(16) | 40 | 20 | 0x0000 - 0xCCCC (0 - 100%) | Voltage value in % of Unom (for translation see programming guide) |
| 11160 | | x | | | x | MPP Tracking: User curve (MPP4 mode) voltage values 61-80 | RW | uint(16) | 40 | 20 | 0x0000 - 0xCCCC (0 - 100%) | Voltage value in % of Unom (for translation see programming guide) |
| 11180 | | x | | | x | MPP Tracking: User curve (MPP4 mode) voltage values 81-100 | RW | uint(16) | 40 | 20 | 0x0000 - 0xCCCC (0 - 100%) | Voltage value in % of Unom (for translation see programming guide) |
| 11200 | | x | | | | MPP Tracking: User curve (MPP4 mode) results 1-10 (10x Umon, Imon, Pmon) | R | uint(16) | 60 | 30 | 0x0000 - 0xCCCC (0 - 100%) / 0x0000 - 0xCCCC (0 - 100%) / 0x0000 - 0xCCCC (0 - 100%) | Voltage value in % of Unom / Current value in % of Inom / Power value in % of Pnom (for translation see programming guide) |
| 11230 | | x | | | | MPP Tracking: User curve (MPP4 mode) results 11-20 (10x Umon, Imon, Pmon) | R | uint(16) | 60 | 30 | 0x0000 - 0xCCCC (0 - 100%) / 0x0000 - 0xCCCC (0 - 100%) / 0x0000 - 0xCCCC (0 - 100%) | Voltage value in % of Unom / Current value in % of Inom / Power value in % of Pnom (for translation see programming guide) |
| 11260 | | x | | | | MPP Tracking: User curve (MPP4 mode) results 21-30 (10x Umon, Imon, Pmon) | R | uint(16) | 60 | 30 | 0x0000 - 0xCCCC (0 - 100%) / 0x0000 - 0xCCCC (0 - 100%) / 0x0000 - 0xCCCC (0 - 100%) | Voltage value in % of Unom / Current value in % of Inom / Power value in % of Pnom (for translation see programming guide) |
| 11290 | | x | | | | MPP Tracking: User curve (MPP4 mode) results 31-40 (10x Umon, Imon, Pmon) | R | uint(16) | 60 | 30 | 0x0000 - 0xCCCC (0 - 100%) / 0x0000 - 0xCCCC (0 - 100%) / 0x0000 - 0xCCCC (0 - 100%) | Voltage value in % of Unom / Current value in % of Inom / Power value in % of Pnom (for translation see programming guide) |
| 11320 | | x | | | | MPP Tracking: User curve (MPP4 mode) results 41-50 (10x Umon, Imon, Pmon) | R | uint(16) | 60 | 30 | 0x0000 - 0xCCCC (0 - 100%) / 0x0000 - 0xCCCC (0 - 100%) / 0x0000 - 0xCCCC (0 - 100%) | Voltage value in % of Unom / Current value in % of Inom / Power value in % of Pnom (for translation see programming guide) |
| 11350 | | x | | | | MPP Tracking: User curve (MPP4 mode) results 51-60 (10x Umon, Imon, Pmon) | R | uint(16) | 60 | 30 | 0x0000 - 0xCCCC (0 - 100%) / 0x0000 - 0xCCCC (0 - 100%) / 0x0000 - 0xCCCC (0 - 100%) | Voltage value in % of Unom / Current value in % of Inom / Power value in % of Pnom (for translation see programming guide) |
| 11380 | | x | | | | MPP Tracking: User curve (MPP4 mode) results 61-70 (10x Umon, Imon, Pmon) | R | uint(16) | 60 | 30 | 0x0000 - 0xCCCC (0 - 100%) / 0x0000 - 0xCCCC (0 - 100%) / 0x0000 - 0xCCCC (0 - 100%) | Voltage value in % of Unom / Currient value in % of Inom / Power value in % of Pnom (for translation see programming guide) |
| 11410 | | x | | | | MPP Tracking: User curve (MPP4 mode) results 71-80 (10x Umon, Imon, Pmon) | R | uint(16) | 60 | 30 | 0x0000 - 0xCCCC (0 - 100%) / 0x0000 - 0xCCCC (0 - 100%) / 0x0000 - 0xCCCC (0 - 100%) | Voltage value in % of Unom / Current value in % of Inom / Power value in % of Pnom (for translation see programming guide) |
| 11440 | | x | | | | MPP Tracking: User curve (MPP4 mode) results 81-90 (10x Umon, Imon, Pmon) | R | uint(16) | 60 | 30 | 0x0000 - 0xCCCC (0 - 100%) / 0x0000 - 0xCCCC (0 - 100%) / 0x0000 - 0xCCCC (0 - 100%) | Voltage value in % of Unom / Current value in % of Inom / Power value in % of Pnom (for translation see programming guide) |
| 11470 | | x | | | | MPP Tracking: User curve (MPP4 mode) results 91-100 (10x Umon, Imon, Pmon) | R | uint(16) | 60 | 30 | 0x0000 - 0xCCCC (0 - 100%) / 0x0000 - 0xCCCC (0 - 100%) / 0x0000 - 0xCCCC (0 - 100%) | Voltage value in % of Unom / Current value in % of Inom / Power value in % of Pnom (for translation see programming guide) |