# Modbus Interface

## *T1000-10 Natural Gas Analyser*

Revision: 16

# Table of Contents

# 1. Modbus Information

The analyser is accessible as a Modbus slave on either Modbus RTU (over RS-485) or Modbus TCP (over Ethernet). Only two functions are supported:

- Read Holding Registers (`0x03`)
- Write Multiple Holding Registers (`0x10`)

All other Modbus functions are ignored.

## 1.1. Modbus RTU

The default parameters for Modbus RTU are shown in Table 1. These can be changed from the Tunable Service Program, see MAN1014.

*Table 1. Default Modbus RTU parameters*

| | |
|---|---|
| Transport | RS-485 |
| Baud rate | 9600 |
| Data bits | 8 |
| Parity | None |
| Stop bits | 2 |
| Slave address | 0x04 |

## 1.2. Modbus TCP

The parameters in Table 2 are used for Modbus TCP. The IP can be changed from the Tunable Service Program, see MAN1014.

*Table 2. Default Modbus TCP parameters*

| | |
|---|---|
| Transport | Ethernet |
| IP | 192.168.9.2 |
| Port | 502 |

# 2. Analyser Operation

An overview of the analyser state machine is shown in Figure 1. At powerup the analyser will automatically enter the `IDLE` state and wait for commands over Modbus. Different programs can be run by writing to the `CMD` register. The current state of the analyser can be read from the `STATE` register.

Any running program can be aborted by writing `STOP` to the `CMD` register. It may take a couple of seconds for analyser to abort the program and return to the `IDLE` state. During this time the `STATE` register will read `STOPPING`.

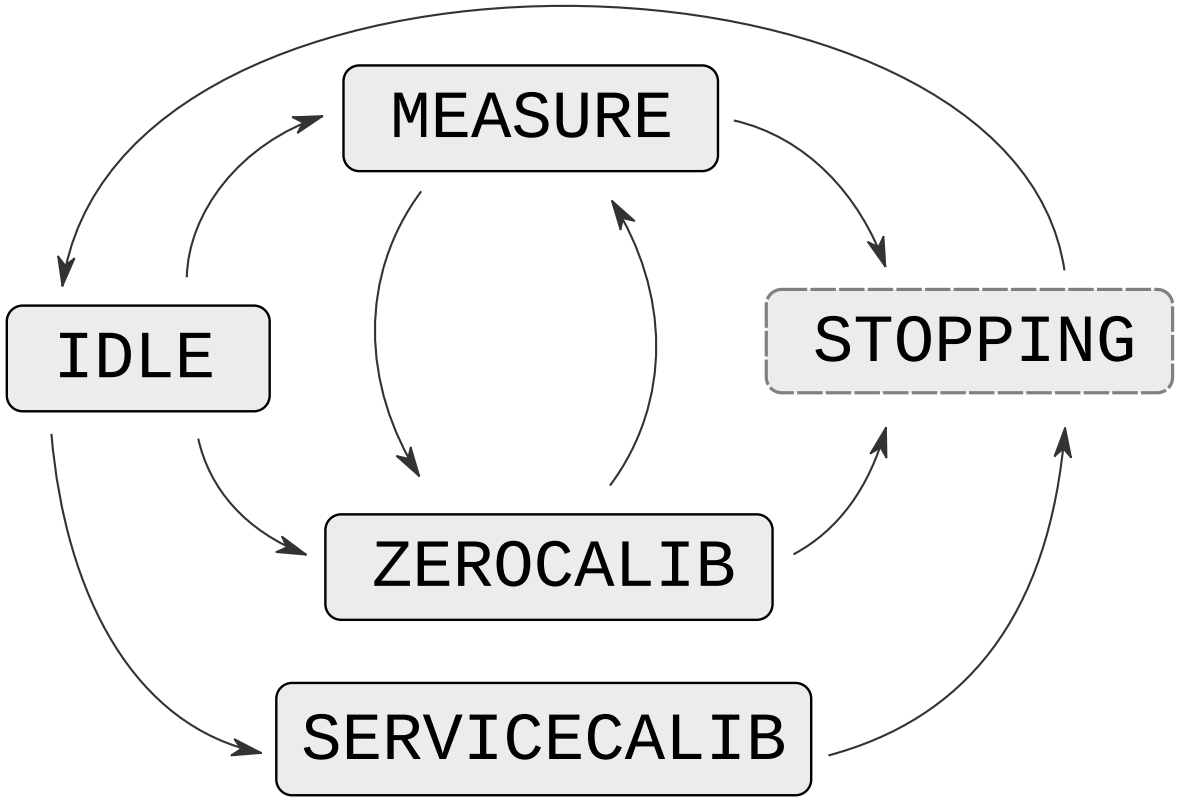


*Figure 1. State Machine*

## 2.1. Relay Control

Two relays intended for controlling soleniod valves can be controlled via the RELAY_CFG register. If the ENABLE bit in RELAY_CFG is LOW the relays are not controlled and will permanently stay OPEN. When the ENABLE bit is HIGH the relays are controlled either *manual* or *automatic* mode depending on the AUTOMATIC bit.

If the AUTOMATIC bit is LOW the relays are controlled by the RELAY_CTRL register, regardless of the analyser state.

If the AUTOMATIC bit is HIGH the relays are controlled automatically depending on the state of the analyser and the configuration in RELAY_CFG. The R1_IDLE and R2_IDLE bits control the state of the relays in the IDLE state. The R1_MEASURE and R2_MEASURE bits control the state of the relays in the MEASURE state. The R1_ZEROCALIB and R2_ZEROCALIB bits control the state of the relays in the ZEROCALIB state.

The state of each relay can always be read from the RELAY_STATE register.

## 2.2. Measure Program

The analyser will start measuring after START_MEAS is written to the CMD register. To stop the analyser, write STOP to the CMD register. The analyser will then enter the IDLE state.

To run a *Finite Measurement*, first program the number of measurement cycles in MEAS_CYCLES, then write START_FINITE_MEAS to the CMD register. The instrument will automatically return to

IDLE when the programmed number of measurement cycles are complete.

The current progression in a measurement cycle can be read through the PROGRESSION register. This register will go from 0 to 100 during each measurement cycle. When a measurement cycle finishes, the MEAS_CNT register will be increased by one.

## 2.3. Zero Calibration Program

The *Zero Calibration* program is run to remove drift due to reduced light intensity (e.g. from dirt on on lenses). The *Zero Calibration* program is started by writing START_ZEROCALIB to the CMD register while the analyser is in the IDLE state. The STATE register will read ZEROCALIB while the *Zero Calibration* program is running. The analyser will automatically return to the IDLE state after the *Zero Calibration* program completes.

*Note*

During *Zero Calibration* the analyser measurement cell must be filled with nitrogen gas.[1] Running *Zero Calibration* with a different gas may affect subsequent measurement results.

If both the ENABLE and AUTOMATIC bits in RELAY_CFG register are set, the relays are controlled according to the settings in R1_ZEROCALIB and R2_ZEROCALIB in RELAY_CFG.

## 2.4. Span Calibration Program

The *Span Calibration* program can be used to correct gain-offsets by calibrating against a known reference.

Before starting a span calibration, first check that the STATE registers reads IDLE and the ERROR_CODE registers read NONE.

Write the target values for each component in the corresponding registers (e.g. write the METHANE concentration to the SPANTARGET_METHANE register). Set the bits in SPANCALIB_INCLUDE for each component that should be calibrated. Only include components that are between 60% and 100% of full range.

Fill the measurement cell with span gas and write START_SPANCALIB to the CMD register to start the span calibration. The instrument will return to IDLE automatically when span calibration is complete. Check that the ERROR_CODE register still reads NONE to verify that the span calibration completed successfully.

When the span calibration completes, the instrument will store *span factors* for each component that is used to adjust the component accordingly in future measurements. The span factors can be read out via separate registers (e.g. span factor for METHANE is read from the SPANFACTOR_METHANE register).

*Note*

Do not include components without a set span target or with very low target concentrations. Doing so will result in poor performance in subsequent measurements. Please ensure that all components that are included in span calibration are above 60% of full range for the instrument.

Span calibration factors can be cleared (reset to 1.0) by writing `CLEAR_SPAN_FACTORS` to the `CMD` register. This will reset the calibration to factory default.

## 2.5. Stream Switching

It is possible to set the instrument up to measure multiple streams and automatically switch between them. If `AUTOMATIC_STREAM_SWITCH` is enabled, the instrument will measure *Stream 1* for the time specified in `STREAM_1_MEASURE_TIME`, then switch to *Stream 2* etc. If `AUTOMATIC_STREAM_SWITCH` is disabled, the user can manually switch streams by writing to the `MANUAL_STREAM_SELECT` register. The stream that was measured for each measurement cycle can be read from the `MEAS_STREAM` register.

## 2.6. Service Calibration Program

The service calibration program is inteded to be run after cleaning lenses. This program creates a new reference point for light intensity levels. The program should be run with nitrogen gas[1] in the measurement cell. To run the program, write `START_SERVICECALIB` while the analyser is in the `IDLE` state. The analyser automatically returns to `IDLE` after the program completes.

If both the `ENABLE` and `AUTOMATIC` bits in `RELAY_CFG` register are set, the relays are controlled according to the settings in `R1_SERVICECALIB` and `R2_SERVICECALIB` in `RELAY_CFG`.

## 2.7. Measurement Data

The measurement data registers (addresses below `0x0080`) are guaranteed to be updated atomically. This means that if these registers are read together in a single Modbus transaction, they are guaranteed to belong to the same measurement cycle.

The `MEAS_CNT` register indicates the last completed measurement cycle, and can be used to check when a new measurement cycle is finished.

*Timestamps*

Timestamps recorded in the `TIMESTAMP` register depend on persistent connection to an NTP (Network Time Protocol) server. If the instrument does not have access to an NTP server, these timestamps will be wrong.

## 2.8. Service Indicators

The service indicator registers, `ABS_TRANS` and `REL_TRANS`, give information about the current condition of the instrument. They are updated during zero calibration.

The ABS_TRANS register holds the current optical transmission relative to the level during production with N2 to in the measurement cell. The REL_TRANS register gives the current light intensity relative to the level during the last service calibration. The service calibration routine can be run by writing START_SERVICECALIB while the instrument is in the IDLE state.

*Intensity Values*

Light intensity values are affected by the gas present in the measurement cell. Make sure to analyze these values with a known gas composition.

## 2.9. Live registers

The *live sensor registers* (e.g. LIVE_GAS_PRESSURE) give out sensor readings regardless if the instrument is measuring or not. These can be used to check temperature and pressure readings when the instrument is IDLE.

# 3. Interface Description

## 3.1. Data Formats

Four different data types are used in the Modbus register map:

- uint16 (2 bytes)
- uint32 (4 bytes)
- float32 (4 bytes)
- string (n bytes)

Values labelled uint16 contain unsigned 16-bit integers.

Values labelled uint32 contain unsigned 32-bit integers. They thus span two Modbus registers. Data is stored in big-endian format (most significant byte at lowest memory address).

Values labelled float32 contain 32-bit floating point numbers. The format corresponds to the IEEE 754 standard. They thus span two Modbus registers. Data is stored in big-endian format (most significant byte at lowest memory address).

strings are stored as zero-terminated ASCII byte arrays with a prepended size field. The size field is of type uint16 and does *not* include the zero-termination character. Table 3 shows the memory layout if the string "TUNE" was stored at 0x8000 (x = undefined).

*Table 3. String Example*

| **Adress:** | 0x8000 | 0x8001 | 0x80002 | 0x8003 |
|---|---|---|---|---|
| **Value (hex):** | 0x0004 | 0x5455 | 0x4E45 | 0x00xx |
| **Value:** | 4 | "TU" | "NE" | "\0" |

## 3.2. Memory Layout

The register map is divided in four sections. Table 4 shows the layout of the Modbus memory.

*Table 4. Memory Layout*

| Section Name | Start Address | End Address |
|---|---|---|
| Data | 0x0000 | 0x0FFF |
| Control | 0x1000 | 0x1FFF |
| Configuration | 0x2000 | 0x2FFF |
| Information | 0x8000 | 0x8FFF |

*Adress Values*

Address values given in this document refers to the data address used in Modbus messages. The Modbus specification has an alternative method of specifying addresses where the first Holding Register is numbered 40001 (decimal). This notation is *not* used in this document.

The *Data Section* contains measurement results as well as status information.

The *Control Section* contains registers for controlling analyser behavior: starting/stopping measurements, performing Zero-Point Calibration, etc.

The *Configuration Section* contains configuration settings.

The *Information Section* contains information about the device.

# 4. Register Map

The register map is given in Table 5.

*Table 5. Register Map*

| Address | Name | Access | Data type |
|---|---|---|---|
| *Data Section* | | | |
| 0x0000 | METHANE | R | float32 |
| 0x0002 | ETHANE | R | float32 |
| 0x0004 | PROPANE | R | float32 |
| 0x0006 | BUTANE | R | float32 |
| 0x0008 | ISOBUTANE | R | float32 |
| 0x000A | C5TOT | R | float32 |
| 0x000C | RESERVED | R | float32 |
| 0x000E | NITROGEN | R | float32 |

| Address | Name | Access | Data type |
|---|---|---|---|
| 0x0020 | GAS_PRESSURE | R | float32 |
| 0x0022 | GAS_TEMP | R | float32 |
| 0x0024 | BOARD_TEMP | R | float32 |
| 0x0026 | FP_TEMP | R | float32 |
| 0x0028 | FP_HUMID | R | float32 |
| 0x0030 | HHV_MASS | R | float32 |
| 0x0032 | LHV_MASS | R | float32 |
| 0x0034 | HHV_VOLUME | R | float32 |
| 0x0036 | LHV_VOLUME | R | float32 |
| 0x0038 | GROSS_WOBBE | R | float32 |
| 0x003A | NET_WOBBE | R | float32 |
| 0x003C | DENSITY | R | float32 |
| 0x003E | REL_DENSITY | R | float32 |
| 0x0040 | MEAS_CNT | R | uint32 |
| 0x0042 | MEAS_FLAGS | R | uint32 |
| 0x0044 | TIMESTAMP | R | uint32 |
| 0x0046 | MEAS_OOR | R | uint32 |
| 0x0048 | MEAS_STREAM | R | uint16 |
| 0x0050 | METHANE_NUMBER | R | float32 |
| 0x0052 | COMPRESSIBILITY | R | float32 |
| 0x0200 | STATE | R | uint16 |
| 0x0202 | ERROR_CODE | R | uint32 |
| 0x0204 | PROGRESSION | R | uint16 |
| 0x0205 | RELAY_STATE | R | uint16 |
| 0x0300 | ABS_TRANS | R | uint16 |
| 0x0301 | REL_TRANS | R | uint16 |
| 0x0420 | LIVE_GAS_PRESSURE | R | float |
| 0x0422 | LIVE_GAS_TEMP | R | float |
| 0x0424 | LIVE_BOARD_TEMP | R | float |
| 0x0426 | LIVE_FP_TEMP | R | float |
| 0x0428 | LIVE_FP_HUMID | R | float |
| *Control Section* | | | |
| 0x1000 | CMD | W | uint16 |

| Address | Name | Access | Data type |
|---|---|---|---|
| 0x1001 | RELAY_CTRL | W | uint16 |
| 0x1010 | REBOOT | W | uint16 |
| *Configuration Section* | | | |
| 0x2000 | AUTOSTART | RW | uint16 |
| 0x2001 | AUTOZERO | RW | uint16 |
| 0x2002 | AUTOZERO_PERIOD | RW | float32 |
| 0x2004 | RELAY_CFG | RW | uint16 |
| 0x200A | MEAS_CYCLES | RW | uint32 |
| 0x200C | AVG_FILTER | RW | uint32 |
| 0x2030 | MANUAL_STREAM_SELECT | RW | uint16 |
| 0x2031 | AUTOMATIC_STREAM_SWITCH | RW | uint16 |
| 0x2032 | STREAM_SWITCH_INCLUDE | RW | uint16 |
| 0x2034 | STREAM_1_MEASURE_TIME | RW | float32 |
| 0x2036 | STREAM_2_MEASURE_TIME | RW | float32 |
| 0x2038 | STREAM_3_MEASURE_TIME | RW | float32 |
| 0x2100 | SPANTARGET_METHANE | RW | float32 |
| 0x2102 | SPANTARGET_ETHANE | RW | float32 |
| 0x2104 | SPANTARGET_PROPANE | RW | float32 |
| 0x2106 | SPANTARGET_BUTANE | RW | float32 |
| 0x2108 | SPANTARGET_ISOBUTANE | RW | float32 |
| 0x210A | SPANTARGET_C5TOT | RW | float32 |
| 0x2200 | SPANFACTOR_METHANE | R | float32 |
| 0x2202 | SPANFACTOR_ETHANE | R | float32 |
| 0x2204 | SPANFACTOR_PROPANE | R | float32 |
| 0x2206 | SPANFACTOR_BUTANE | R | float32 |
| 0x2208 | SPANFACTOR_ISOBUTANE | R | float32 |
| 0x220A | SPANFACTOR_C5TOT | R | float32 |
| 0x2300 | SPANCALIB_INCLUDE | RW | uint16 |
| *Information Section* | | | |
| 0x7000 | MAPTYPE | R | uint16 |
| 0x7001 | MAPREV | R | uint16 |
| 0x7002 | MANUFACTURER | R | uint16 |
| 0x7003 | DEVTYPE | R | uint16 |

| Address | Name | Access | Data type |
|---|---|---|---|
| 0x7004 | HWPLATFORM | R | uint16 |
| 0x8000 | SERIAL | R | string |
| 0x8100 | OS_VER | R | string |
| 0x8200 | FW_VER | R | string |
| 0x8300 | FPGA_VER | R | string |

# 4.1. Register Descriptions

## 4.1.1. METHANE Register

*Table 6. METHANE Register (0x0000, float32, R)*

| Unit | Description |
|---|---|
| mol-% | METHANE concentration |

## 4.1.2. ETHANE Register

*Table 7. ETHANE Register (0x0002, float32, R)*

| Unit | Description |
|---|---|
| mol-% | ETHANE concentration |

## 4.1.3. PROPANE Register

*Table 8. PROPANE Register (0x0004, float32, R)*

| Unit | Description |
|---|---|
| mol-% | PROPANE concentration |

## 4.1.4. BUTANE Register

*Table 9. BUTANE Register (0x0006, float32, R)*

| Unit | Description |
|---|---|
| mol-% | BUTANE concentration |

## 4.1.5. ISOBUTANE Register

*Table 10. ISOBUTANE Register (0x0008, float32, R)*

| Unit | Description |
|---|---|
| mol-% | ISOBUTANE concentration |

### 4.1.6. C5TOT Register

*Table 11. C5TOT Register (0x000A, float32, R)*

| Unit | Description |
|---|---|
| mol-% | C5TOT concentration |

### 4.1.7. NITROGEN Register

*Table 12. NITROGEN Register (0x000E, float32, R)*

| Unit | Description |
|---|---|
| mol-% | NITROGEN concentration |

### 4.1.8. GAS_PRESSURE Register

*Table 13. GAS_PRESSURE register (0x0020, float32, R)*

| Unit | Description |
|---|---|
| bar | Gas pressure |

### 4.1.9. GAS_TEMP Register

*Table 14. GAS_TEMP register (0x0022, float32, R)*

| Unit | Description |
|---|---|
| C | Gas temperature |

### 4.1.10. BOARD_TEMP Register

*Table 15. BOARD_TEMP register (0x0024, float32, R)*

| Unit | Description |
|---|---|
| C | Board temperature |

### 4.1.11. FP_TEMP Register

*Table 16. FP_TEMP register (0x0026, float32, R)*

| Unit | Description |
|---|---|
| C | Fabry-Perot filter temperature |

### 4.1.12. FP_HUMID Register

*Table 17. FP_HUMID register (0x0028, float32, R)*

| Unit | Description |
|---|---|
| RH | Fabry-Perot filter humidity |

### 4.1.13. HHV_MASS Register

*Table 18. HHV_MASS register (0x0030, float32, R)*

| Unit | Description |
|---|---|
| MJ/kg | Higher Heating Value (Calorific Value), mass basis |

### 4.1.14. LHV_MASS Register

*Table 19. LHV_MASS register (0x0032, float32, R)*

| Unit | Description |
|---|---|
| MJ/kg | Lower Heating Value (Calorific Value), mass basis |

### 4.1.15. HHV_VOLUME Register

*Table 20. HHV_VOLUME register (0x0034, float32, R)*

| Unit | Description |
|---|---|
| MJ/m3 | Higher Heating Value (Calorific Value), volume basis |

### 4.1.16. LHV_VOLUME Register

*Table 21. LHV_VOLUME register (0x0036, float32, R)*

| Unit | Description |
|---|---|
| MJ/m3 | Lower Heating Value (Calorific Value), volume basis |

### 4.1.17. GROSS_WOBBE Register

*Table 22. GROSS_WOBBE register (0x0038, float32, R)*

| Unit | Description |
|---|---|
| MJ/m3 | Gross Wobbe Index |

### 4.1.18. NET_WOBBE Register

*Table 23. NET_WOBBE register (0x003A, float32, R)*

| Unit | Description |
|---|---|
| MJ/m3 | Net Wobbe Index |

### 4.1.19. DENSITY Register

*Table 24. DENSITY register (0x003C, float32, R)*

| Unit | Description |
|---|---|
| kg/m3 | Gas density |

### 4.1.20. REL_DENSITY Register

*Table 25. REL_DENSITY register (0x003E, float32, R)*

| Unit | Description |
|---|---|
| - | Relative density |

### 4.1.21. MEAS_CNT Register

*Table 26. MEAS_CNT register (0x0040, uint32, R)*

| Unit | Description |
|---|---|
| - | Counter increased by 1 each time a new measurement cycle finishes. |

### 4.1.22. MEAS_FLAGS Register

*Table 27. MEAS_FLAGS register (0x0042, uint32, R)*

| Bit | Name | Description |
|---|---|---|
| 0 | DATAREADY | Indicates if data registers are ready (contains measurement data).<br>0: Data registers are not ready.<br>1: Data registers are ready. |
| 1 | FP_TEMP_HIGH | Fabry Perot filter temperature high warning<br>0: No warning.<br>1: Fabry Perot filter temperature too high. |
| 2 | FP_TEMP_LOW | Fabry Perot filter temperature low warning.<br>0: No warning.<br>1: Fabry Perot filter temperature too low. |
| 3 | GAS_TEMP_HIGH | Gas temperature high warning.<br>0: No warning.<br>1: Gas temperature too high. |
| 4 | GAS_TEMP_LOW | Gas temperature high warning. 0: No warning.<br>1: Gas temperature too low. |
| 5 | GAS_PRESSURE_HIGH | Gas pressure high warning.<br>0: No warning.<br>1: Gas pressure too high. |
| 6 | GAS_PRESSURE_LOW | Gas pressure low warning.<br>0: No warning.<br>1: Gas pressure too low warning |
| 7 | ZERO_FLUSHING | Indicates whether instrument is flushing with zero gas.<br>0: No flushing in progress.<br>1: Flushing in progress |

| Bit | Name | Description |
|---|---|---|
| 8 | PROCESS_FLUSHING | Indicates whether instrument is flushing with process gas.<br>0: No flushing in progress.<br>1: Flushing in progress |
| 9 | MEASUREMENT_OUT_OF_RANGE | One or more measurements outside calibrated range warning.<br>0: No warning.<br>1: One or more measured quantities outside of calibrated ranges, check the MEAS_OOR register for per-quantity warnings. |

### 4.1.23. TIMESTAMP Register

*Table 28. TIMESTAMP register (0x0044, uint32, R)*

| Unit | Description |
|---|---|
| s | Unix timestamp (seconds since 1 Jan 1970 00:00) |

### 4.1.24. MEAS_OOR Register

*Table 29. MEAS_OOR register (0x0046, uint32_t, R)*

| Bit | Name | Description |
|---|---|---|
| 0 | METHANE_OOR | Indicates if the METHANE concentration is within calibrated range.<br>0: Concentration within calibrated range.<br>1: Concentration outside of calibrated range. |
| 1 | ETHANE_OOR | Indicates if the ETHANE concentration is within calibrated range.<br>0: Concentration within calibrated range.<br>1: Concentration outside of calibrated range. |
| 2 | PROPANE_OOR | Indicates if the PROPANE concentration is within calibrated range.<br>0: Concentration within calibrated range.<br>1: Concentration outside of calibrated range. |
| 3 | BUTANE_OOR | Indicates if the BUTANE concentration is within calibrated range.<br>0: Concentration within calibrated range.<br>1: Concentration outside of calibrated range. |
| 4 | ISOBUTANE_OOR | Indicates if the ISOBUTANE concentration is within calibrated range.<br>0: Concentration within calibrated range.<br>1: Concentration outside of calibrated range. |

| Bit | Name | Description |
|---|---|---|
| 5 | C5TOT_OOR | Indicates if the C5TOT concentration is within calibrated range.<br>0: Concentration within calibrated range.<br>1: Concentration outside of calibrated range. |
| 7 | NITROGEN_OOR | Indicates if the NITROGEN concentration is within calibrated range.<br>0: Concentration within calibrated range.<br>1: Concentration outside of calibrated range. |
| 16 | METHANE_NUMBER_OOR | Indicates if the methane number calculation is within calibrated range.<br>0: Methane number calulation valid according to ISO23306.<br>1: Methane number calculation not valid. |

### 4.1.25. MEAS_STREAM Register

*Table 30. MEAS_STREAM register (0x0048, uint16, R)*

| Value | Name | Description |
|---|---|---|
| 0 | N/A | N/A |
| 1 | STREAM_1 | Measure Stream 1 |
| 2 | STREAM_2 | Measure Stream 2 |
| 3 | STREAM_3 | Measure Stream 3 |

### 4.1.26. METHANE_NUMBER Register

*Table 31. METHANE_NUMBER register (0x0050, float32, R)*

| Description |
|---|
| Methane number calculated according to ISO 23306:2020 Annex A |

### 4.1.27. COMPRESSIBILITY Register

*Table 32. COMPRESSIBILITY register (0x0052, float32, R)*

| Description |
|---|
| Compressibility factor calculated according to ISO 6976:2016 section 6.2. |

### 4.1.28. STATE Register

*Table 33. STATE Register (0x0200, uint16, R)*

| Value | Name | Description |
|---|---|---|
| 0 | IDLE | Analyser is idle |

| Value | Name | Description |
|---|---|---|
| 1 | MEASURE | Analyser is measuring gas |
| 2 | ZEROCALIB | Analyser is performing Zero-Point Calibration |
| 7 | STOPPING | Analyser is in process of stopping (returning to IDLE state) |
| 13 | SERVICECALIB | Analyser is performing service calibration |
| 16 | SPANCALIB | Analyser is performing Span Calibration |
| 18 | ZERO_FLUSHING | Analyser is flushing with zero gas |
| 19 | PROCESS_FLUSHING | Analyser is flushing with process gas |
| 24 | STARTING | Analyzer in process of starting a measurement |

### 4.1.29. ERROR_CODE Register

*Table 34. ERROR_CODE register (0x0202, uint32, R)*

| Value | Name | Description |
|---|---|---|
| 0 | NONE | No error |
| 1 | GENERIC | Unspecified error |
| 2 | CONFIG | Configuration error |
| 3 | DATABASE | Database error |
| 4 | FPGA | FPGA error |
| 5 | IRDRV | IR driver error |
| 6 | IRDET | IR detector error |
| 7 | FILTER | IR filter error |
| 10 | DMA | DMA error |
| 11 | FILTERTEMP | Filter temperature sensor error |
| 12 | GASTEMP | Gas temperature sensor error |
| 13 | GASPRESSURE | Gas pressure sensor error |
| 14 | FILTERHUMID | Filter humidity sensor error |
| 15 | SELFTEST | Selftest failed |
| 17 | WATCHDOG | Watchdog error |
| 18 | MODBUS | Modbus error |
| 19 | FILTERCALIB | Filter calibration error |
| 20 | RELAYCTRL | Error in relay ctrl |
| 22 | HEALTHCHECK | Errur during healthcheck |
| 24 | IRDETCLIP | IR detector clipping |
| 26 | COMPUTE | Internal computation error |

| Value | Name | Description |
|---|---|---|
| 29 | SPANCALIB | Span calibration failed |
| 33 | FPDACCLIP | FP DAC clip (internal error) |
| 34 | HIGHFPTEMP | Filter temperature too high |
| 35 | LOWFPTEP | Filter temperature too low |
| 36 | HIGHGASTEMP | Gas temp too high |
| 37 | LOWGASTEMP | Gas temp too low |
| 38 | HIGHGASPRESS | Gas pressure too high |
| 39 | LOWGASPRESS | Gas pressure too low |
| 40 | HIGHREFCALIBTEMP | Temperature above calibration |
| 41 | LOWREFCALIBTEMP | Temperature below calibration |
| 42 | CVFAIL | Failed to calculate Calorific Value |
| 44 | MNFAIL | Failed to calculate Methane Number |
| 45 | MODELERR | Error in estimation model |
| 63 | ZEROFAIL | Zero caliberation failed |

### 4.1.30. PROGRESSION Register

*Table 35. PROGRESSION register (0x0204, uint16, R)*

| Unit | Description |
|---|---|
| % | Measurement cycle progression |

### 4.1.31. RELAY_STATE Register

*Table 36. RELAY_STATE register (0x0205, uint16, R)*

| Bit | Name | Description |
|---|---|---|
| 0 | R1 | 0: R1 is OPEN.<br>1: R1 is CLOSED. |
| 1 | R2 | 0: R2 is OPEN.<br>1: R2 is CLOSED. |

### 4.1.32. ABS_TRANS Register

*Table 37. ABS_TRANS Register (0x0300, uint16, R)*

| Unit | Description |
|---|---|
| % | Absolute transmission |

### 4.1.33. REL_TRANS Register

*Table 38. REL_TRANS Register (0x0301, uint16, R)*

| Unit | Description |
|---|---|
| % | Relative transmission |

### 4.1.34. LIVE_GAS_PRESSURE Register

*Table 39. LIVE_GAS_PRESSURE register (0x0420, float32, R)*

| Unit | Description |
|---|---|
| bar | Live gas pressure |

### 4.1.35. LIVE_GAS_TEMP Register

*Table 40. LIVE_GAS_TEMP register (0x0422, float32, R)*

| Unit | Description |
|---|---|
| C | Live gas temperature |

### 4.1.36. LIVE_BOARD_TEMP Register

*Table 41. LIVE_BOARD_TEMP register (0x0424, float32, R)*

| Unit | Description |
|---|---|
| C | Live board temperature |

### 4.1.37. LIVE_FP_TEMP Register

*Table 42. LIVE_FP_TEMP register (0x0426, float32, R)*

| Unit | Description |
|---|---|
| C | Live Fabry-Perot filter temperature |

### 4.1.38. LIVE_FP_HUMID Register

*Table 43. LIVE_FP_HUMID register (0x0428, float32, R)*

| Unit | Description |
|---|---|
| RH | Fabry-Perot filter humidity |

### 4.1.39. CMD Register

*Table 44. CMD Register (0x1000, uint16, W)*

| Value | Name | Description |
|---|---|---|
| `0` | `NOOP` | No operation. This command is ignored. |

| Value | Name | Description |
|---|---|---|
| 1 | START_MEAS | Start measurements |
| 2 | STOP | Stop measurement activity. Return to idle state. |
| 3 | START_ZEROCALIB | Start Zero Calibration. |
| 8 | START_SERVICECALIB | Start Service Calibration |
| 9 | START_SPANCALIB | Start Span Calibration |
| 10 | CLEAR_SPAN_FACTORS | Reset Span Calibration Factors to 1. |
| 11 | CLEAR_ERROR | Clears the ERROR_CODE register. |
| 12 | START_FINITE_MEAS | Starts a *Finite Measurement*. See also MEAS_CYCLES. |

### 4.1.40. RELAY_CTRL Register

*Table 45. RELAY_CTRL register (0x1001, uint16, W)*

| Bit | NAME | Description |
|---|---|---|
| 0 | R1 | 0: R1 is OPEN.<br>1: R1 is CLOSED. |
| 1 | R2 | 0: R2 is OPEN.<br>1: R2 is CLOSED. |

### 4.1.41. REBOOT Register

*Table 46. REBOOT register (0x1010, uint16, W)*

| Value | Name | Description |
|---|---|---|
| 0xDEAD | REBOOTCMD | Triggers a reboot of the instrument. All other values are ignored. |

### 4.1.42. AUTOSTART Register

*Table 47. AUTOSTART register (0x2000, uint16, RW)*

| Value | Name | Description |
|---|---|---|
| 0 | FALSE | Enter IDLE state on start-up |
| 1 | TRUE | Start measurements on start-up |

### 4.1.43. AUTOZERO Register

*Table 48. AUTOZERO register (0x2001, uint16, RW)*

| Value | Name | Description |
|---|---|---|
| 0 | FALSE | Do not run automatic zero-point calibrations |
| 1 | TRUE | Run zero-point calibrations automatically |

### 4.1.44. AUTOZERO_PERIOD Register

*Table 49. AUTOZERO_PERIOD register (0x2002, float32, RW)*

| Unit | Description |
|---|---|
| h | Time between zero-point calibrations |

### 4.1.45. RELAY_CFG Register

*Table 50. RELAY_CFG register (0x2004, uint32, RW)*

| Bit | NAME | Description |
|---|---|---|
| 0 | ENABLE | 0: Relay control is DISABLED.<br>1: Relay controls ENABLED. |
| 1 | AUTOMATIC | 0: Relay control is MANUAL.<br>1: Relay control is AUTOMATIC. |
| 2 | R1_IDLE | 0: R1 is OPEN in IDLE state.<br>1: R1 is CLOSED in IDLE state. |
| 3 | R2_IDLE | 0: R2 is OPEN in IDLE state.<br>1: R2 is CLOSED in IDLE state. |
| 4 | R1_MEASURE | 0: R1 is OPEN in MEASURE state<br>1: R1 is CLOSED in MEASURE state. |
| 5 | R2_MEASURE | 0: R2 is OPEN in MEASURE state.<br>1: R2 is CLOSED in MEASURE state. |
| 6 | R1_ZEROCALIB | 0: R1 is OPEN in ZEROCALIB state.<br>1: R1 is CLOSED in ZEROCALIB state. |
| 7 | R2_ZEROCALIB | 0: R2 is OPEN in ZEROCALIB state.<br>1: R2 is CLOSED in ZEROCALIB state. |
| 8 | R1_SERVICECALIB | 0: R1 is OPEN in SERVICECALIB state.<br>1: R1 is CLOSED in SERVICECALIB state. |
| 9 | R2_SERVICECALIB | 0: R2 is OPEN in SERVICECALIB state.<br>1: R2 is CLOSED in SERVICECALIB state. |
| 10 | R1_MEASURE2 | 0: R1 is OPEN in MEASURE STREAM 2 state<br>1: R1 is CLOSED in MEASURE STREAM 2 state. |
| 11 | R2_MEASURE2 | 0: R2 is OPEN in MEASURE STREAM 2 state.<br>1: R2 is CLOSED in MEASURE STREAM 2 state. |
| 12 | R1_MEASURE3 | 0: R1 is OPEN in MEASURE STREAM 3 state<br>1: R1 is CLOSED in MEASURE STREAM 3 state. |
| 13 | R2_MEASURE3 | 0: R2 is OPEN in MEASURE STREAM 3 state.<br>1: R2 is CLOSED in MEASURE STREAM 3 state. |
| 14 | R1_SPANCALIB | 0: R1 is OPEN in SPANCALIB state<br>1: R1 is CLOSED in SPANCALIB state. |

| Bit | NAME | Description |
|---|---|---|
| 15 | R2_SPANCALIB | 0: R2 is OPEN in SPANCALIB state.<br>1: R2 is CLOSED in SPANCALIB state. |

### 4.1.46. MEAS_CYCLES Register

*Table 51. MEAS_CYCLES Register (0x200A, uint32, RW)*

| Unit | Description |
|---|---|
| - | Number of measurement cycles to run for a *Finite Measurement*. |

### 4.1.47. AVG_FILTER Register

*Table 52. AVG_FILTER Register (0x200C, uint32, RW)*

| Unit | Description |
|---|---|
| - | Size of averaging filter in number of measurement cycles. |

### 4.1.48. MANUAL_STREAM_SELECT Register

*Table 53. MANUAL_STREAM_SELECT register (0x2030, uint16, RW)*

| Value | Name | Description |
|---|---|---|
| 1 | STREAM_1 | Select stream 1 |
| 2 | STREAM_2 | Select stream 2 |
| 3 | STREAM_3 | Select stream 3 |

### 4.1.49. AUTOMATIC_STREAM_SWITCH Register

*Table 54. AUTOMATIC_STREAM_SWITCH register (0x2031, uint16, RW)*

| Value | Name | Description |
|---|---|---|
| 0 | FALSE | Disable automatic switch of measurement streams |
| 1 | TRUE | Enable automatic switch of measurement streams |

### 4.1.50. STREAM_SWITCH_INCLUDE Register

*Table 55. STREAM_SWITCH_INCLUDE register (0x2032, uint16, RW)*

| Bit | Name | Description |
|---|---|---|
| 0 | INCLUDE_STREAM_1 | Indicates if measurement stream 1 will be included on the automatic stream switching.<br>0: Do not include 1: Include |
| 1 | INCLUDE_STREAM_2 | Indicates if measurement stream 2 will be included on the automatic stream switching.<br>0: Do not include. 1: Include. |

| Bit | Name | Description |
|---|---|---|
| 2 | INCLUDE_STREAM_3 | Indicates if measurement stream 3 will be included on the automatic stream switching.<br>0: Do not include. 1: Include. |

### 4.1.51. STREAM_1_MEASURE_TIME Register

*Table 56. STREAM_1_MEASURE_TIME register (0x2034, float32, RW)*

| Unit | Description |
|---|---|
| h | Time to measure stream 1 when automatic stream switching is enabled |

### 4.1.52. STREAM_2_MEASURE_TIME Register

*Table 57. STREAM_2_MEASURE_TIME register (0x2036, float32, RW)*

| Unit | Description |
|---|---|
| h | Time to measure stream 2 when automatic stream switching is enabled |

### 4.1.53. STREAM_3_MEASURE_TIME Register

*Table 58. STREAM_3_MEASURE_TIME register (0x2038, float32, RW)*

| Unit | Description |
|---|---|
| h | Time to measure stream 3 when automatic stream switching is enabled |

### 4.1.54. SPANTARGET_METHANE Register

*Table 59. SPANTARGET_METHANE register (0x2100, float32, RW)*

| Description |
|---|
| Span calibration target for METHANE |

### 4.1.55. SPANTARGET_ETHANE Register

*Table 60. SPANTARGET_ETHANE register (0x2102, float32, RW)*

| Description |
|---|
| Span calibration target for ETHANE |

### 4.1.56. SPANTARGET_PROPANE Register

*Table 61. SPANTARGET_PROPANE register (0x2104, float32, RW)*

| Description |
|---|
| Span calibration target for PROPANE |

### 4.1.57. SPANTARGET_BUTANE Register

*Table 62. SPANTARGET_BUTANE register (0x2106, float32, RW)*

| Description |
| --- |
| Span calibration target for BUTANE |

### 4.1.58. SPANTARGET_ISOBUTANE Register

*Table 63. SPANTARGET_ISOBUTANE register (0x2108, float32, RW)*

| Description |
| --- |
| Span calibration target for ISOBUTANE |

### 4.1.59. SPANTARGET_C5TOT Register

*Table 64. SPANTARGET_C5TOT register (0x210A, float32, RW)*

| Description |
| --- |
| Span calibration target for C5TOT |

### 4.1.60. SPANFACTOR_METHANE Register

*Table 65. SPANFACTOR_METHANE register (0x2200, float32, R)*

| Description |
| --- |
| Span calibration factor for METHANE |

### 4.1.61. SPANFACTOR_ETHANE Register

*Table 66. SPANFACTOR_ETHANE register (0x2202, float32, R)*

| Description |
| --- |
| Span calibration factor for ETHANE |

### 4.1.62. SPANFACTOR_PROPANE Register

*Table 67. SPANFACTOR_PROPANE register (0x2204, float32, R)*

| Description |
| --- |
| Span calibration factor for PROPANE |

### 4.1.63. SPANFACTOR_BUTANE Register

*Table 68. SPANFACTOR_BUTANE register (0x2206, float32, R)*

| Description |
| --- |
| Span calibration factor for BUTANE |

### 4.1.64. SPANFACTOR_ISOBUTANE Register

*Table 69. SPANFACTOR_ISOBUTANE register (0x2208, float32, R)*

| Description |
|---|
| Span calibration factor for ISOBUTANE |

### 4.1.65. SPANFACTOR_C5TOT Register

*Table 70. SPANFACTOR_C5TOT register (0x220A, float32, R)*

| Description |
|---|
| Span calibration factor for C5TOT |

### 4.1.66. SPANCALIB_INCLUDE Register

*Table 71. SPANCALIB_INCLUDE register (0x2300, uint32, RW)*

| Bit | Name | Description |
|---|---|---|
| 0 | METHANE | Indicates if next span calibration should include METHANE.<br>0: Do not include 1: Include |
| 1 | ETHANE | Indicates if next span calibration should include ETHANE.<br>0: Do not include 1: Include |
| 2 | PROPANE | Indicates if next span calibration should include PROPANE.<br>0: Do not include 1: Include |
| 3 | BUTANE | Indicates if next span calibration should include BUTANE.<br>0: Do not include 1: Include |
| 4 | ISOBUTANE | Indicates if next span calibration should include ISOBUTANE.<br>0: Do not include 1: Include |
| 5 | C5TOT | Indicates if next span calibration should include C5TOT.<br>0: Do not include 1: Include |

### 4.1.67. MAPTYPE Register

Identifies the register map type.

*Table 72. MAPTYPE register (0x7000, uint16, R)*

| Value | Description |
|---|---|
| 12 | Always reads this value. Indicates that the register map corresponds to the description in this document. |

### 4.1.68. MAPREV Register

Identifies the register map revision.

*Table 73. MAPREV register (0x7001, uint16, R)*

| Value | Description |
|---|---|
| `10` | Always reads this value. Indicates that the register map matches this documentation. |

### 4.1.69. MANUFACTURER Register

Identifies the manufacturer.

*Table 74. MANUFACTURER register (0x7002, uint16, R)*

| Value | Description |
|---|---|
| `0x5455` | Always reads this value. Indicates that this is a Tunable AS instrument. |

### 4.1.70. DEVTYPE Register

Identifies the device type.

*Table 75. DEVTYPE register (0x7003, uint16, R)*

| Value | Description |
|---|---|
| `2` | Always reads this value. Indicates that this is a T1000-10 instrument. |

### 4.1.71. HWPLATFORM Register

Identifies the hardware platform.

*Table 76. HWPLATFORM register (0x7004, uint16, R)*

| Description |
|---|
| Number representing the hardware platform |

### 4.1.72. SERIAL Register

*Table 77. SERIAL Register (0x8000, string, R)*

| Description |
|---|
| Serial number of device. |

### 4.1.73. OS_VER Register

*Table 78. OS_VER Register (0x8100, string, R)*

| Description |
|---|
| OS version |

### 4.1.74. FW_VER Register

*Table 79. FW_VER Register (0x8200, string, R)*

| Description |
|---|
| FW version |

### 4.1.75. FPGA_VER Register

*Table 80. FPGA_VER Register (0x8300, string, R)*

| Description |
|---|
| FPGA version |

# 5. Revisions

*Table 81. Document Revisions*

| Rev. | Date | Description |
|---|---|---|
| 16 | 07.11.2024 | Fixed wrong address for `NITROGEN` register. |
| 15 | 21.08.2024 | Updated documentation for `ERROR_CODE` register. |
| 14 | 25.04.2024 | Split manual for T1000-10 and T1000-20 in two documents. Added `MANUFACTURER`, `DEVTYPE` and `HWPLATFORM` registers. Increased `MAPREV` to 10. |
| 13 | 13.02.2024 | Added `Rx_SPANCALIB` bits to `RELAY_CFG` register. Increased `MAPREV` to 9. |
| 12 | 15.11.2023 | Fixed wrong values for `SERVICECALIB` and `SPANCALIB` in `STATE` register. Added `STARTING` state. |
| 11 | 27.10.2023 | Added `LIVE_` registers. Added *Finite Measurement* command. Changed `MAPREV` to 8. |
| 10 | 21.09.2023 | Added stream switch registers. Changed `MAPREV` to 7. |
| 09 | 30.05.2023 | Added `MEASUREMENT_OUT_OF_RANGE` flag to `MEAS_FLAGS` register. Added `MEAS_OOR` register. Added `COMPRESSIBILITY` register. Changed `MAPREV` to 6. |
| 08 | 02.08.2022 | Added `NOOP` command to `CMD` register. Changed `MAPREV` to 5. |
| 07 | 06.07.2022 | Renamed `FLUSHING` flag to `ZERO_FLUSHING`. Added `PROCESS_FLUSHING` flag. Added `ZERO_FLUSHING` and `PROCESS_FLUSHING` states to the `STATE` register. Changed `MAPREV` to 4. |
| 06 | 14.05.2022 | Added `FLUSHING` flag to `MEAS_FLAGS`. Added `DENSITY` and `REL_DENSITY` registers. Added `CLEAR_ERROR` command to `CMD` register. Changed `MAPREV` to 3. |
| 05 | 21.01.2022 | Added methane number. Added span calibration registers. Removed `FACTORY_RESET` register. |

| Rev. | Date | Description |
|---|---|---|
| 04 | 04.10.2021 | Fixed wrong revision in header |
| 03 | 26.04.2021 | Fixed wrong data type for `RELAY_CFG` |
| 02 | 11.01.2021 | Changed document number |
| 01 | 26.11.2020 | Initial release |

[1] A suitable inert gas like helium or argon may be used instead of nitrogen.