

	<ul style="list-style-type: none"> <li>• Modbus RTU Datalogger</li> <li>• Acts as Modbus Master or Slave</li> <li>• Micro SD Card for Data Storage</li> <li>• Extremely Accurate Real Time Clock</li> <li>• Battery Backed RTC</li> <li>• Simple Configuration</li> <li>• Half Duplex RS485</li> <li>• Slimline DIN Rail Enclosure</li> <li>• Arduino Programmable</li> <li>• Open Source</li> </ul>
---	--

The KTA-287 Modbus Datalogger is an easy to use solution when data from a Modbus network needs to be recorded.

The device works as either a Modbus Master, polling slave devices for register data, or a Modbus Slave, with another master writing to it, in both cases the data is recorded as a plain text CSV (Comma Separated Values) file, along with an accurate ( $\pm 2$  minutes per year) timestamp to a Micro SD card.

Micro SD and Micro SDHC cards are both supported, provided they are formatted as FAT16 or FAT32.

## Connections

Name	Function
V+	Positive Voltage Input, 8-30 VDC
COM	Negative Power Supply Input
D+	RS485 Data Positive
D-	RS485 Data Negative

## Preparation

Micro SD and Micro SDHC cards are supported, provided they are formatted as either FAT16 or FAT32. Though most operating systems may have a format utility inbuilt it is recommended to use the formatting utility provided by [www.sdcard.org](http://www.sdcard.org).

Once the card is formatted in either FAT32 or FAT16, insert it into the datalogger, taking care to ensure the correct orientation (if it does not easily insert, try the other way around, don't force it), and then power the datalogger. After a few seconds the red LED should flash twice and then the green LED should flash 5 times to indicate that no configuration file was found and a new one was created. Power down the datalogger and then remove the Micro SD card and insert it into a PC.

## Configuration

Configuration of the datalogger is all done through a configuration file stored on the Micro SD card called "KTA-287.cfg". Follow the steps in the section titled "Preparation" and then open the configuration file "KTA-287.cfg" in a text editor.

It is important to note that the datalogger will look for the information in the configuration file in a specific manner, that is the line should include a colon then a space, then the parameter, then a carriage return character, this is true for all parameters except for registers to read as a master. In the case of registers to read as a master the format consists of three parameters separated by two commas and terminated with a carriage return character, without any spaces.

## Parameters

Parameter Name	Values	Description
Update Time	Y/N	When set to Y the time of the Real Time Clock (RTC) will be updated on the next power up. The value is automatically written back to N by the datalogger.
New Time	YYMMDDHHMMSS	The time to update the RTC with on the next power up, provided the "Update Time" parameter is set to Y. (HH in 24 hour time).
Master/Slave	M/S	Determines whether the datalogger is a Modbus Master or Modbus Slave.
Slave Address	1-247	The slave id the datalogger will use when in slave mode.
Baud	1200 - 57600	The baudrate the datalogger will communicate at.
Data Bits	5/6/7/8	The data bits the datalogger will communicate with.
Parity	N/O/E	The parity the datalogger will communicate with.
Stop Bits	1/2	The number of stop bits the datalogger will communicate with.
Log Rate	1-65535	The rate, in number of seconds at which the datalogger will write values to the Micro SD card.
New File Rate	H/D/M/Y	The datalogger will create a new file on the Micro SD card, either Hourly, Daily, Monthly or Yearly.
Registers to Log	Slave ID, Type (C/S//H), Reg No	Following this parameter, a list of data points to read and log. 1,H,10 will log holding register 10 from Modbus Slave 1. 3,C,1 will log coil 1 from Modbus Slave 3.

## Setting the Date and Time

Setting the time in the datalogger is done using the configuration file and timing the next power on of the datalogger to coincide with the time in the configuration file.

1. Ensure there is a configuration file on the SD card by powering the datalogger on with the SD card inserted.
2. Power off the datalogger and insert the SD card into a card reader and read it on a computer.
3. Set the "Update Time" parameter to "Y".
4. Choose a time approx 2-5 mins in the future and construct the "New Time" parameter according to the YYMMDDHHMMSS format (141231134500 is 31/12/14 13:45:00).
5. Save the configuration file and eject the SD card.
6. Ensuring power is off, insert the SD card into the datalogger and wait.
7. Monitor a clock or other accurate time source and when the time chosen in step 4 rolls about apply power to the datalogger.
8. The dataloggers green LED will flash 4 times to acknowledge that the time and date have been updated.
9. The "Update Time" parameter is automatically reset to "N" so that the time and date are not updated on the next power on.

## Using the Datalogger as a Modbus Slave

When the "Master/Slave" parameter is set to "S", the datalogger will be configured as a Modbus Slave with 12 holding registers, starting at register 40,001. The Slave ID or Modbus Address for the datalogger is set by the "Slave Address" parameter.

In Modbus Slave mode, the contents of the 12 slave registers are saved to the Micro SD card for every log period.

## Using the Datalogger as a Modbus Master

When the “Master/Slave” parameter is set to “M”, the datalogger will be configured as a Modbus Master and can interrogate up to 12 tags, which can be holding registers, input registers, coils or status bits, from various modbus slaves. The 12 tags can be any combination of the register types and any combination of slave addresses. The tags are configured at the end of the configuration file and are saved to the Micro SD card for every log period. If there is a communication problem and the datalogger can not receive a valid response from a slave for a particular tag, the reading in the log file will be replaced with the text “FAIL”.

## Notes on Log Rate and New File Rate

Choosing a logging rate and new file creation rate must be done with a little thought of the application and the data processing after the data has been logged and is be analysed or graphed on a computer.

In the simplest instance the data can be imported to a spreadsheet program, such as Microsoft Excel or Open Office Calc. Please note that Open Office Calc and older versions of Excel have a limit of 65535 rows in a single file, therefore if logging files will have more than 65535 loggings it is wise to either decrease the logging rate or shorten the new file creation period.

It should also be noted that the more loggings there are, the longer it will take to open and analyse files.

Depending on the New File Rate setting, filenames of the logged data will take different forms.

“YY.csv” when in yearly file creation mode, where YY is the 2 digit representation of the year.

“YYMM.csv” when in monthly file creation mode, where YY is the 2 digit representation of the year and MM is the 2 digit representation of the month.

“YYMMDD.csv” when in daily file creation mode, where YY is the 2 digit representation of the year, MM is the 2 digit representation of the month and DD is the 2 digit representation of the day of the month.

“YYMMDDHH.csv” when in hourly file creation mode, where YY is the 2 digit representation of the year, MM is the 2 digit representation of the month, DD is the 2 digit representation of the day of the month and HH is the 2 digit representation of the hour, in 24 hour time.

The table below can be used as a guide to choose a good logging rate and new file creation rate combination.

Cells highlighted in Yellow indicate logging rates with over 65535 rows per file, which is unsuitable for some spreadsheet programs						
Log Rate (sec)	Log Rate (min)	Logs/Hour	Logs/Day	Logs/Month	Logs/Year	
1	0.0	3600	86400	2592000	31536000	
2	0.0	1800	43200	1296000	15768000	
5	0.1	720	17280	518400	6307200	
10	0.2	360	8640	259200	3153600	
15	0.3	240	5760	172800	2102400	
20	0.3	180	4320	129600	1576800	
30	0.5	120	2880	86400	1051200	
40	0.7	90	2160	64800	788400	
60	1.0	60	1440	43200	525600	
120	2.0	30	720	21600	262800	
300	5.0	12	288	8640	105120	
400	6.7	9	216	6480	78840	
482	8.0	7	179	5378	65427	
600	10.0	6	144	4320	52560	
700	11.7	5	123	3703	45051	
800	13.3	5	108	3240	39420	
900	15.0	4	96	2880	35040	
1800	30.0	2	48	1440	17520	
3600	60.0	1	24	720	8760	

### LED Error Indication

#### Red LED

Permanently lit – Card failure or card not present

Blink 2 times – Card present on power up

#### Green LED

Blink 5 times – Configuration file not found, new configuration file written to SD card.

Blink 4 times – New time found and updated

Blink 2 times – No need to update time

Toggle once per second – Normal operation

#### Blue LED

Blinking Periodically – Normal operation, lit when writing to SD card

### Replacing the Battery

The internal timekeeping circuit requires a back up battery. A fresh battery should keep the timekeeping circuit accurate for over a year without external power applied to the datalogger.

If the datalogger is creating files with incorrect time and date then the battery will need to be replaced.

A 12mm 3V coin cell battery, CR1225 or equivalent is needed. To replace the battery, open the enclosure, remove the old battery from it's holder and then insert the new battery, taking care of the polarity, ensuring it is as marked on the battery holder. Once the battery is installed replace the enclosure and update the time of the datalogger.

### Reprogramming the Datalogger

The KTA-287 runs the Arduino bootloader and therefore can be reprogrammed to suit a particular purpose or application. A programming adapter such as the 3.3V FTDI Basic is required available from Ocean Controls as ARD-010 <http://oceancontrols.com.au/ARD-010.html>.

When programming, the board type in the Arduino environment needs to be set to “*Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega328*”.

Ocean Controls can not provide support for reprogramming the KTA-287 datalogger with custom firmware.

### Open Source License

The firmware for the KTA-287 Modbus Datalogger is licensed under GNU GPL V3.0 see <http://www.gnu.org/licenses/> for details of the license