



ATOLL SOLUTIONS

MODBUS CONFIGURATION AND SERVER INTEGRATION

© 2016 Atoll Solutions Pvt Ltd.

All Rights Reserved. No part of this document may be photocopied, reproduced, stored in a retrieval system, or transmitted, in any form or by any means whether, electronic, Mechanical, or otherwise without the prior written permission of Atoll Solutions. No warranty of accuracy is given concerning the contents of the information contained in this publication. To the extent permitted by law no liability (including liability to any person by reason of negligence) will be accepted by Atoll Solutions or employees for any direct or indirect loss or damage caused by omissions from or inaccuracies in this document.

Atoll Solutions reserves the right to change details in this document without notice. Product and company names herein may be the trademarks of their respective owners.

#229, Second Floor, 2A Main, 5th Cross, HAL 3rd Stage, Bangalore, India -560 075.

Web: www.AtollSolutions.com Email: info@atollsolutions.com

Introduction

This document explains how to integrate and decode the Modbus Holding registers when configured along with Atoll Modbus RTU Edge Gateway. Atoll Modbus RTU Edge Gateways support up to 6 Modbus RTU Slaves. Modbus RTU parameters are configurable using application tool "ATOLL CONFIX".

Modbus RTU Slave Configuration using Atoll Confix

If user wants to interface an external Modbus device with the ARxxx/ATxxx by using the application AtollConfix the user should have the following details of the external device (for eg: energy meter).

1. UART configuration

The user should be clear about the following properties of the UART:

- **Baud rate**(The user should configure the baud rate of the external Modbus device in Base configuration tab under Modbus UART)
- **Data bit**(The data bit of the ARxxx/ATxxx devices are hardcoded to '8', in this case please keep the data bit of the external device as '8')
- **Parity**(The user should configure the parity of the external Modbus device in Base configuration tab under Modbus UART)
- **Stop bit**(The user should configure the stop bit for the external Modbus device in Base configuration tab under Modbus UART)

2. Configure Modbus request

The user is allowed to configure 6 different types Modbus request. In order to set a Modbus request the user should be clear about the following parameter of the Modbus device.

- **Slave id/station code** for the Modbus device.
- **Function code** for the register to read.
- **Starting address of the register**(from which address onwards the user want to read the Modbus data)
- **Number of registers to read**(Number of registers to read from the starting address onwards)
- **Modbus timeout** (Modbus operation timeout in sec).

Note: The maximum number of registers can read by a single Modbus request is restricted to 60 so in case a user who needs more than 60 registers, for example if a user needs to read first 90 registers from address 0x0000 onwards, the user should read the first 60 registers using the first Modbus request and the remaining 30 registers by configuring the second Modbus request (but the starting address of the second Modbus request should be the register address for the 61th register, e.g. (0x003D)). In case of Modbus communication error instead of the Modbus data the device will send the ERROR string.

An Example: How to Integrate RS485 based Modbus (RTU) Energy Meter with Atoll's AR501RM using ATOLL CONFIX?

In this particular example we have used ELMEASURE Little Genius 5110 - 3 Phase Energy meter, which supports RS485 based Modbus RTU communication. Following are the register address details for the above mentioned Energy meter:

Sl.No.	Parameter	Data type	Address
1	Watts Total	float	40101
2	Watts R phase	float	40103
3	Watts Y phase	float	40105
4	Watts B phase	float	40107
5	VAR Total	float	40109
6	VAR R phase	float	40111
7	VAR Y phase	float	40113
8	VAR B phase	float	40115
9	PF Ave. (Inst.)	float	40117
10	PF R phase	float	40119
11	PF Y phase	float	40121

23

Sl.No.	Parameter	Data type	Address
12	PF B phase	float	40123
13	VA total	float	40125
14	VA R phase	float	40127
15	VA Y phase	float	40129
16	VA B phase	float	40131
17	VLL average	float	40133
18	Vry phase	float	40135
19	Vyb phase	float	40137
20	Vbr phase	float	40139
21	VLN average	float	40141
22	V R phase	float	40143
23	V Y phase	float	40145
24	V B phase	float	40147
25	Current Total	float	40149
26	Current R phase	float	40151
27	Current Y phase	float	40153
28	Current B phase	float	40155
29	Frequency	float	40157
30	Wh Received	float	40159
31	Load Hours Received	Unsigned long	40217

From the above mentioned register details, we can read the parameters as per customers requirement. Elmeasure Little Genius 5110 Meter addresses 32 Bit Registers, hence the decoding mechanism may vary for different meters.

For 32 bit register decoding mechanism, visit the link mentioned below:

www6.uniovi.es/~antonio/uned/ieee754/IEEE-754hex32.html

Here in this example we are going to read following 3 parameters such as:

- 1) R Phase Voltage (40142, 40143).
- 2) R Phase Current (40150, 40151).
- 3) Frequency (40156, 40157).

Configuration of the Modbus Register Address is explained below:

Slave ID and Function code of ELMEASURE Little Genius 5110 - 3 Phase Energy meter is 01 and 03 respectively.

Since we have to read the above mentioned parameters, we can split the Modbus requests into 2 [Modbus Request 1 and Modbus Request 2].

For Modbus Request 1, we are going to read R Phase Voltage and R Phase Current that is from register address 40142 to 40152 (10 Registers).

For Modbus Request 2, we are going to read Frequency that is 40156 to 40157(2 registers).

Following are the steps which shows as to how a user can Integrate AR501RM with Modbus Based Energy Meter [RTU]?

Step 1:

Connect the Atoll Confix board with the AR501RM Gateway with the help of 4 Pin micro fit connector and 20 Pin micro fit connector.

Step 2:

Connect the Atoll Confix board to the Local PC with the help of micro USB cable and resume the application tool "Atoll Confix".

Step 3:

Power up the Energy Meter as well as the Atoll Confix Board.

Step 4:

Select the desired COM port on the application tool and click on the “Connect” button. Please note that after clicking on the “connect” button this tool will display Device Summary details.

Step 5:

Click on the Base configuration tab. Configure and update the server parameters as well as the Modbus Requests.

For more information: www.youtube.com/watch?v=x1JDCA2ultg

AtollConfix

Serial | **Base Configuration** | Parameter Configuration

BASIC PARAMETERS

Message Header: msgeheade

Device id: DEV01

Updation Interval: 0060 sec

Server Selection: ☐ TCP/IP ☒ HTTP ☐ HTTPS

☒ HTTP GET ☐ HTTP POST

url Path: /temp/atollservice.p

Authentication KEY1: keyyyy1

Authentication KEY2: keyyyyy2

IP address: 54.186.113.71

Port: 80

APN: airtelgprs.com

GPS: ☐

Modbus UART (RS485)

Baud: 009600

DataBit: 8

Parity: none

StopBit: 1

Mobile no.1: +910000000001

Mobile no.2: +910000000002

Mobile no.3: +919945002993

☐ Data Backup

	Slave id	Functnod	Start_add	No.Reg.read	Timeout
Modbus Request 1	01	03	00142	10	0100 ms
Modbus Request 2	01	03	00156	002	0100 ms
Modbus Request 3	01	03	00100	002	0100 ms
Modbus Request 4	01	03	00100	002	0100 ms
Modbus Request 5	01	03	00100	002	0100 ms
Modbus Request 6	01	03	00100	002	0100 ms

STATUS

Step 6:

Click on the Parameter configuration tab. Configure and update the required parameters that needs to be pushed to server along with Modbus Reply for the configured requests in base configuration.

For more information: www.youtube.com/watch?v=x1JDCA2ultg

The screenshot shows the AtollConfX software interface with the 'Parameter Configuration' tab selected. The interface is divided into several sections for configuring various parameters.

Serial | **Base Configuration** | **Parameter Configuration**

Message Header ☒ **MNC** ☒ **GPS**

Device Id ☒ **Cell Id** ☒ Latitude ☐

IMEI number ☒ **Device firmware version** ☒ Longitude ☐

Date and Time ☒ **Module firmware version** ☒ Altitude ☐

MCC ☒ **Internal Battery Voltage** ☒ Speed Overground ☐

Main Power Status ☒ **GSM SignalStrength** ☒ Course Overground ☐

Number of Satellites ☐

Modbus Reply

Modbus Reply1 ☒ Modbus Reply2 ☒ Modbus Reply3 ☐

Modbus Reply4 ☐ Modbus Reply5 ☐ Modbus Reply6 ☐

Digital input1 ☒ **Digital output1** ☒ **Analog input1** ☒

Digital input2 ☒ **Digital output2** ☒ **Analog input2** ☒

Digital input3 ☒ **Digital output3** ☒ **4-20MA input1** ☒

Digital input4 ☒ **Digital output4** ☒ **4-20MA input2** ☒

Update

STATUS

Data received at the server:

```

graph LR
    A[ ] --> B[R phase Voltage]
    B --> C[R Phase Current]
    C --> D[Frequency]
    style A fill:none,stroke:none

```

```
header=ELMEASURE&id=DEV01&imei=353173066847441&time=13:06:2016,16:44:23&di1=1  
&di2=1&di3=1&di4=1&do1=0&do2=0&do3=0&do4=0&ai1=11&ai2=11&ai3=11&ai4=11&modbus  
1=6507594300000000000000000000000000000000&modbus2=5e024842&modbus3=MODBUS3_ER  
ROR&modbus4=MODBUS4_ERROR&modbus5=MODBUS5_ERROR&modbus6=MODBUS6_ERROR&mcc=404&  
mnc=45&cellid=1F87&ver=001&battery=3.97&pwrstatus=1&gsmver=10.00.188&gsmisig=30&end=  
0*
```

header=ELMEASURE&id=DEV01&imei=353173066847441&time=13:06:2016,16:44:23&dil=1
&di2=1&di3=1&di4=1&do1=0&do2=0&do3=0&do4=0&ai1=11&ai2=11&ai3=11&ai4=11&modbus
1=**65075943**000000000000000000000000**00000000**&modbus2=**5e024842**&modbus3=MODBUS3_ER
ROR&modbus4=MODBUS4_ERROR&modbus5=MODBUS5_ERROR&modbus6=MODBUS6_ERROR&mcc=404&
mnc=45&cellid=1F87&ver=001&battery=3.97&pwrstatus=1&gsmver=10.00.188&gmsig=30&end=
0