DDSU666 (Direct chint meter) + mbpoll Command Reference

Version: 1.0
Prepared For: Field & Deployment Teams
Platform: Raspberry Pi / Linux
Tool: mbpoll

1. Purpose

This document explains how to communicate with the CHINT DDSU666 Direct Smart Meter using Modbus RTU and the mbpoll tool.

It covers:
• Reading electrical parameters
• Setting meter addresses
• Verifying communication
• Preparing for OpenEMS integration

2. Hardware & Software Requirements

Hardware
• DDSU666 (Direct Version)
• RS485 → USB Converter
• Raspberry Pi / Linux PC
• Correct RS485 wiring (A(converter)↔24(meter com-port terminal), B(converter)↔25(meter com-port terminal), GND recommended)

Software
Install mbpoll:

> “
> sudo apt update
> sudo apt install mbpoll

Check serial port:

```
ls /dev/ttyUSB*
```

Example output:

```
/dev/ttyUSB0
```

3. Communication Parameters (Confirmed)

Parameter	Value
Protocol	Modbus RTU
Baud Rate	9600
Data Bits	8
Parity	None
Stop Bits	2
Format	8N2
Float Order	Big Endian
Addressing	0-Based

These parameters must always be used.

4. Standard mbpoll Format

All commands follow this format:

```
mbpoll -m rtu -b 9600 -P none -s 2 -t 4:float -B -0 -r <register> -c <count>
/dev/ttyUSB0 -a <id> -1
```

Where:

- <register> = Modbus register
- <count> = Number of values
- <id> = Meter address (NO.)

5. Electrical Parameters

Summary:

```
```

> Voltage: 0x2000
> Current: 0x2002
> Active Power: 0x2004
> Power Factor: 0x200A
> Frequency: 0x200E
> Energy: 0x4000

5.1 Voltage (V) — Register 0x2000

> “
> **Meter ID = 1**
> mbpoll -m rtu -b 9600 -P none -s 2 -t 4:float -B -0 \
> -r 0x2000 -c 1 /dev/ttyUSB0 -a 1 -1
> **Meter ID = 2**
> mbpoll -m rtu -b 9600 -P none -s 2 -t 4:float -B -0 \
> -r 0x2000 -c 1 /dev/ttyUSB0 -a 2 -1

5.2 Current (A) — Register 0x2002

> “
> **Meter ID = 1**
> mbpoll -m rtu -b 9600 -P none -s 2 -t 4:float -B -0 \
> -r 0x2002 -c 1 /dev/ttyUSB0 -a 1 -1
> **Meter ID = 2**
> mbpoll -m rtu -b 9600 -P none -s 2 -t 4:float -B -0 \
> -r 0x2002 -c 1 /dev/ttyUSB0 -a 2 -1

5.3 Active Power — Register 0x2004

> “
> **Meter ID = 1**
> mbpoll -m rtu -b 9600 -P none -s 2 -t 4:float -B -0 \
> -r 0x2004 -c 1 /dev/ttyUSB0 -a 1 -1
>
> **Meter ID = 2**
> mbpoll -m rtu -b 9600 -P none -s 2 -t 4:float -B -0 \
> -r 0x2004 -c 1 /dev/ttyUSB0 -a 2 -1

5.4 Power Factor — Register 0x200A

> **Meter ID = 1**
> mbpoll -m rtu -b 9600 -P none -s 2 -t 4:float -B -0 \
> -r 0x200A -c 1 /dev/ttyUSB0 -a 1 -1
> **Meter ID = 2**
> mbpoll -m rtu -b 9600 -P none -s 2 -t 4:float -B -0 \
> -r 0x200A -c 1 /dev/ttyUSB0 -a 2 -1

5.5 Frequency (Hz) — Register 0x200E

> **Meter ID = 1**
> mbpoll -m rtu -b 9600 -P none -s 2 -t 4:float -B -0 \
> -r 0x200E -c 1 /dev/ttyUSB0 -a 1 -1
> **Meter ID = 2**
> mbpoll -m rtu -b 9600 -P none -s 2 -t 4:float -B -0 \
> -r 0x200E -c 1 /dev/ttyUSB0 -a 2 -1

5.6 Energy (kWh) — Register 0x4000

> **Meter ID = 1**
> mbpoll -m rtu -b 9600 -P none -s 2 -t 4:float -B -0 \
> -r 0x4000 -c 1 /dev/ttyUSB0 -a 1 -1
> **Meter ID = 2**
> mbpoll -m rtu -b 9600 -P none -s 2 -t 4:float -B -0 \
> -r 0x4000 -c 1 /dev/ttyUSB0 -a 2 -1

6. Reading Multiple Values

Use **-c** option to read multiple registers.

Example (Voltage + Current Together)

>

```
mbpoll -m rtu -b 9600 -P none -s 2 -t 4:float -B -0 \
-r 0x2000 -c 2 /dev/ttyUSB0 -a 1 -1
```

7. Meter Address

> **Register: 0x0006**
>
> ```
> mbpoll -m rtu -b 9600 -P none -s 2 -t 4:int16 -0 \
> -r 0x0006 -c 1 /dev/ttyUSB0 -a 2 -1
> ```
>
> **Example Output:**
> [6]: 2

8. Changing Address

- Only one meter connected.
- Power cycle after change.

> **Change Address (2 → 1)**
> ```
> mbpoll -m rtu -b 9600 -P none -s 2 -t 4:int16 -0 \
> -r 0x0006 /dev/ttyUSB0 -a 2 -W -- 1
> ```
> **Power Cycle**
> Turn OFF → ON the meter.
> **Verify**
> ```
> mbpoll -m rtu -b 9600 -P none -s 2 -t 4:int16 -0 \
> -r 0x0006 -c 1 /dev/ttyUSB0 -a 1 -1
> ```

9. Health Check

Voltage should be 220–240V.

> ```
> mbpoll -m rtu -b 9600 -P none -s 2 -t 4:float -B -0 \
> -r 0x2000 -c 1 /dev/ttyUSB0 -a 1 -1
> ```

10. Common Issues

Timeout Errors

- Duplicate IDs
- Wrong wiring
- Wrong stop bits
- Multiple meters during setup

Incorrect Float Values

Use -B option.

Current / Power = 0

Normal when:

- No load
- Bench testing

11. Deployment Workflow

For large installations:

- Connect one meter
- Convert to Modbus
- Set address
- Test voltage
- Install
- Repeat
- Never deploy duplicate addresses.

> “
> Connect → Configure → Test → Install

12. System Status

✔ Modbus Enabled
✔ 9600 8N2 Confirmed
✔ Big-Endian Floats
✔ Multi-meter Bus Working
✔ Ready for OpenEMS

Revision #4
Created 2026-01-26 01:15:27 UTC by hillary.arinda
Updated 2026-03-28 11:17:11 UTC by hillary.arinda