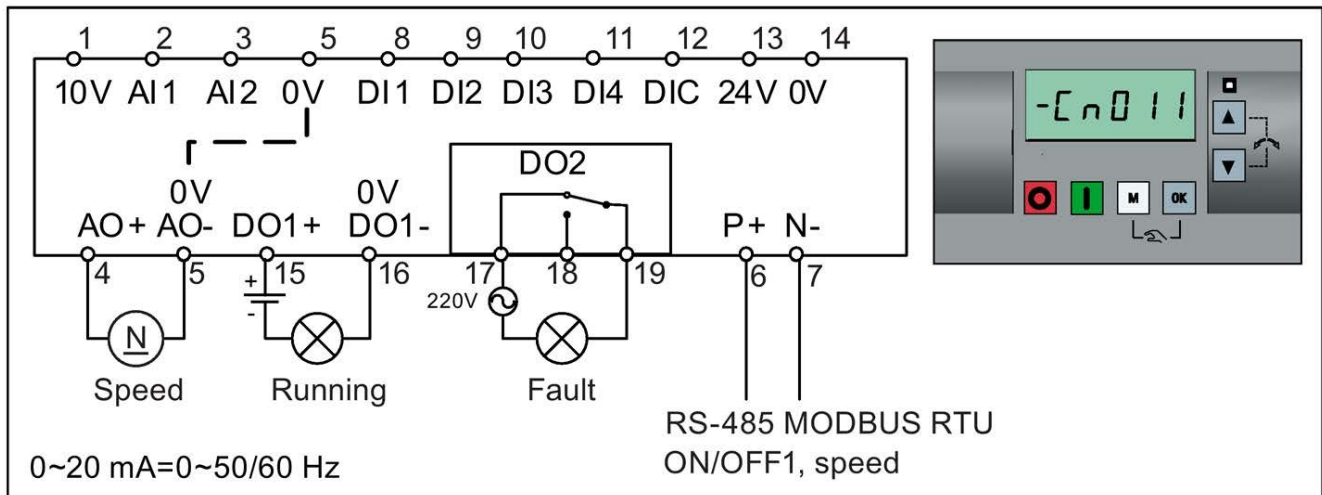


Siemens SINAMICS V20 Inverter

Inverter control through serial RS-485 MODBUS

To control the inverter from the computer we use the commissioning preset Cn011: MODBUS RTU control. This means that the inverter will take commands on a serial RS-485 line using the MODBUS protocol [1].



We found that another important step for controlling the inverter from the computer using this mode, Cn011, is to set the parameter P2014 of the inverter to 0. This makes it so that the time between receipts of two consecutive process data telegrams can be of any length [2].

Inverter setup

To setup the inverter preset Cn011:

- long press M
- short press M
- use the arrows to go to preset Cn011
- press OK
- long press M

Restart the inverter by cutting power, wait for all lights to turn off, apply power again.

To setup parameter P2014:

- short press M
- use arrows to go to P2014
- press M
- press M
- decrease value using arrows to 0
- press OK
- long press M

Serial connection and pulse motor control

USB to RS-485 adapter

The serial connection requires only two electrical lines from the computer to the inverter, a couple of terminating resistors and a USB to RS-485 adapter, since computers don't usually have a RS-485 connector.



On the type of adapter that we used the connections are marked A/D+ and B/D-. On the inverter the connections are marked 6.P+ and 7.N-. The connection should be A/D+ to 6.P+, B/D- to 7.N-

For the terminating resistors, since we do not have a complex setup with multiple devices connected, we found that two 120 Ohm resistors across each end of the lines was sufficient.

MODBUS communication software

To test the communication between the inverter and the computer we wrote a couple of simple programs in Java: one to choose the serial port for the communication, one to read the holding registers from the inverter and one to set a register to a specific value.

We used the JlibModbus [3] open source library for the MODBUS protocol implementation and the jSSC (java Simple Serial Connector) [4] open source library for serial communication.

Pulse control commands

To initialize the pulse control of the motor on the inverter [5] we have to set the STW register 40100 (index 99 in the MODBUS table). This register contains 16 bit flags that setup the pulse motor control. We have the enable (set to 1) the OFF2, OFF3, PULSE ENABLE, RFG ENABLE, RFG START, SETPOINT ENABLE, CONTROL BY PLC. The rest of the flags

are set to 0, including the OFF1 (motor start flag). When assembled in the correct order this 16 bit field gives us the value 0x047E in hexadecimal and 1150 in decimal.

So the first step is to write 1150 to the register with index 99.

The second step is to **set the initial motor pulse**. This is done by writing a value representing the pulse frequency to the HSW register 40101 (MODBUS index 100). The value is a number between 0 and 0x4000 hexadecimal, representing 0% to 100% of the maximum motor pulse frequency set in the motor setup parameters of the inverter. A value of 500 (decimal) is what we found that starts the motor very slowly and doesn't cause an under-power whining sound.

Then the next step is to **start the motor** by flipping the start motor OFF1 flag on the STW register. This means the value 0x047F in hexadecimal or 1151 in decimal to the register with MODBUS index 99.

After this we can write any value we want to register 100 to change the speed of the motor, and 1150 to register 100 to stop or 1151 to start the motor.

Earthquake simulation

Reading the SAC file

The code we are using to read the horizontal motion (Y) is taken from the SeisFile open source library written as part of the “Lithospheric Seismology program” at the University of South Carolina [6]. This gives us an array of samples of the Y horizontal “counts”, the sample period, total number of samples and, depending on the file, a couple of metadata information from the header of the SAC file: time of the start of the recording, station information, scale of the recording etc.

Frequency extraction algorithm

Based on the data extracted from the SAC file we have implemented a very simple algorithm that takes periods of time (by default 3 seconds) and determines the number of changes of direction in that interval and divides them by the interval length to get the frequency. In the algorithm the threshold can be set to ignore changes that have an amplitude difference lower than a set value (by default 2000 counts).

```
int tDelta = 0;
int tChnq = 0;
boolean goingDown = true;
int prevY = 0;
int chnqY = 0;
boolean started = false;
for (int i=0; i<data.header.getNpts(); i++) {
    tDelta = tDelta + sDelta;
    if(tDelta>=3000) {
        if(started || tChnq > 3) {
            while(true) {
                try {
                    m.writeSingleRegister(slaveId, 100, (tChnq*500)/(tDelta/1000));
                    break;
                } catch (ModbusIOException miex) {}
            }
            Thread.sleep(2960);
            started = true;
        }
        tDelta = 0;
        tChnq = 0;
    }
    if(i>0) {
        float deltaY = prevY - (int)data.q[i];
        if(deltaY<0 && goingDown) {
            goingDown = false;
            if(chnqY - prevY > 2000) {
                chnqY = (int)data.q[i];
                tChnq = tChnq + 1;
            }
        }
        if(deltaY>0 && !goingDown) {
            goingDown = true;
            if(prevY - chnqY > 2000) {
                chnqY = (int)data.q[i];
                tChnq = tChnq+1;
            }
        }
        prevY = (int)data.q[i];
    }
}
```

loop over the total number of sample points

each 3 seconds interval check the number of changes in amplitude direction

if more than 3 changes detected start calculating the frequency and write it to the motor

frequency to motor pulse scaling factor

40 ms to calculate frequency and write to the motor, sleep for the remaining time to reach 3 seconds

detect changes in amplitude direction and add them

but only if they have a delta > 2000 counts

References

1. Siemens SINAMICS V20 Inverter Operating Instructions, https://cache.industry.siemens.com/dl/files/484/67267484/att_61458/v1/v20_operating_instructions_complete_en-US_en-US.pdf, pag 74-75
2. Idem, pag 145, "Fault" section
3. JLibModbus Java library, <https://github.com/kochedykov/jlibmodbus>
4. jSSC java Simple Serial Connector library, <https://github.com/scream3r/java-simple-serial-connector>
5. Siemens SINAMICS V20 Inverter Operating Instructions, https://cache.industry.siemens.com/dl/files/484/67267484/att_61458/v1/v20_operating_instructions_complete_en-US_en-US.pdf, pag 138-146 and pag 232-234
6. SeisFile, <https://www.seis.sc.edu/seisFile.html>